

Rochester Institute of Technology

RIT Digital Institutional Repository

Theses

1999

IP and ATM integration: A New paradigm in multi-service internetworking

Remesh Shanmuganathan

Follow this and additional works at: <https://repository.rit.edu/theses>

Recommended Citation

Shanmuganathan, Remesh, "IP and ATM integration: A New paradigm in multi-service internetworking" (1999). Thesis. Rochester Institute of Technology. Accessed from

This Thesis is brought to you for free and open access by the RIT Libraries. For more information, please contact repository@rit.edu.

IP and ATM Integration

A new paradigm in multi-service internetworking

By

Ramesh Shanmuganathan

BSc.Eng (Hons) MBA CEng (UK) MIEEE(USA) AMIEE(UK) AMBCS(UK)

Thesis submitted in Partial fulfillment of the requirements for the degree of
Master of Science in Telecommunications and Data Networks

**Department of Information Technology
Rochester Institute of Technology**

May 1999

© Copyright 1999 Ramesh Shanmuganathan

All rights reserved

**Rochester Institute of Technology
Department of Information Technology**

**Master of Science in Information Technology
Thesis Approval Form**

Student Name: Ramesh Shanmuganathan

Student Number: _____

Thesis Title: IP and ATM Integration:
A New Paradigm in Multi-Service Internetworking

Thesis Committee

Name

Signature

Date

Prof. A'isha Ajayi
Chair

5/28/99

Prof. Daryl Johnson
Committee Member

5/7/99

Prof. Rayno Niemi
Committee Member

5/7/99

Thesis Reproduction Permission Statement

Permission from Author Required

Title of thesis: IP and ATM integration:
 A new paradigm in multi-service internetworking

I, **Ramesh Shanmuganathan**, prefer to be contacted each time a request for reproduction is made. If permission is granted, any reproduction will not be for commercial use or profit. I can be reached at the following address:

255/13 Galle Road
Bambalapitiya
Colombo 4
SRI LANKA
Tel: 94-1-589290
Email: ramesh@computer.org, s.ramesh-rit@ieee.org

Date: 7th May 1999 Signature of Author: _____

Dedication

To my parents, Shan & Param

To my sweetheart, Karen

and

To my siblings, Shyama & Suresh

Acknowledgements

This research study was undertaken in partial fulfillment of the requirements for the Master of Science in Data & Telecommunications Networks (Information Technology) at Rochester Institute of Technology, New York. I am greatly indebted to Professor Niemi (Graduate Chair), Professor A'isha Ajayi (Primary thesis supervisor) and Professor Daryl Johnson (Secondary thesis supervisor) for their valued guidance, advise and unstinted support in fulfilling my research work, thesis writing and defense amidst a very tight schedule.

I would like to extend my gratitude to the Hayes-Fulbright Commission in the USA and in Sri Lanka for the opportunity provided to me, in terms of the Hayes-Fulbright scholarship, which facilitated this enriching and rewarding experience for me in the United States to explore new vistas and test my limits doing it. I would like to extend my gratitude to the staff members at United States Education Foundation, Sri Lanka and Institute of International Education, New York for their continued support and assistance during my stay in the United States.

Further, I would also like to thank all the staff members at the School of Computer Science and Information Technology, my colleagues at University Telecommunications Division, University of Rochester and to all my friends and colleagues for their moral support and encouragement in all my endeavors.

Thank you.

Ramesh Shanmuganathan

Rochester, New York

May 7, 1999

Table of Contents

List of Figures	iv
List of Tables	vii
Abstract	viii
1. Introduction	1
2. Scope, Methodology and Limitations of research	3
2.1 <i>Scope of Research</i>	3
2.2 <i>Research Model</i>	4
2.3 <i>Methodology of research</i>	5
2.4 <i>Limitations of research</i>	5
3. Research Findings – Part I	
Asynchronous Transfer Mode – Architecture & internetworking	6
3.1 <i>ATM –An overview</i>	6
3.2 <i>ATM Architecture</i>	8
3.2.1 <i>Physical Layer of ATM architectures</i>	9
3.2.2 <i>ATM adaptation layers</i>	13
3.2.3 <i>ATM Physical Interfaces</i>	17
3.2.4 <i>Cell structure</i>	17
3.2.5 <i>ATM virtual connection</i>	19
3.2.6 <i>ATM Connection setup</i>	22
3.3 <i>Switching Architectures</i>	24
3.3.1 <i>Transport vs. Control</i>	25
3.3.2 <i>ATM switch functions</i>	25
3.3.3 <i>Queuing Methods</i>	26
3.3.4 <i>General Structure of an ATM switch</i>	28
3.3.5 <i>Switching Element performance requirements</i>	28
3.3.6 <i>Switch architectures</i>	31
3.4 <i>ATM Signaling and Addressing</i>	36
3.5 <i>ATM Routing protocols</i>	41
3.5.1 <i>PNNI requirements</i>	42
3.5.2 <i>PNNI concepts</i>	42
3.5.3 <i>PNNI routing</i>	43
3.5.4 <i>PNNI – QoS support</i>	44
3.5.5 <i>PNNI – Scalability and reachability</i>	47
3.5.6 <i>Crankback and alternate routing</i>	51
3.5.7 <i>The IISP protocol</i>	52
3.5.8 <i>Multicast routing</i>	53
3.5.9 <i>PNNI routing extensions</i>	53
3.6 <i>ATM and OSI Model</i>	54
3.7 <i>Summary</i>	55

4. Research Findings – Part II	
Internet protocol (IP) – Architecture and Internetworking	56
4.1 Architecture and protocols	56
4.1.1 Architectural Models	57
4.1.2 Addressing	60
4.1.3 Internet Protocol (IP)	69
4.1.4 Internet Control Message Protocol (ICMP)	75
4.1.5 Internet Group Management Protocol (IGMP)	77
4.1.6 Address Resolution Protocol (ARP)	78
4.1.7 Reverse Address Resolution Protocol (RARP)	83
4.2 IP Routing protocol	84
4.2.1 Basic IP routing	84
4.2.2 Routing Architecture	86
4.2.3 Interior Routing Protocols	88
4.2.4 Exterior Routing protocols	98
4.2.5 New Routing Protocols	102
4.3 IP – The Next generation (IPng)	104
4.3.1 Limitations of IPv4	104
4.3.2 The IPv6 challenge	105
4.3.3 The technical case for IPv6	106
4.3.4 IPv6 Addresses – unicast, anycast and multicast	107
4.3.5 Exceptional extension headers	109
4.3.6 The IPv6 architecture	113
4.3.7 Other protocols and services	118
4.3.8 Transition mechanisms from IPv4 to IPv6	118
4.3.9 Transition scenarios	123
4.4 RSVP and integrated services on IP networks	125
4.4.1 Integrated services	126
4.4.2 Resource reSerVation Protocol(RSVP)	126
4.5 Summary	128
5. Research Findings – Part III	
IP over ATM or IP/ATM Integration – A new paradigm shift?	130
5.1 The challenges of interworking IP and ATM	130
5.2 Framework for IP and ATM internetworking	133
5.3 Models for IP and ATM internetworking architecture	135
5.3.1 The Classical/Overlay Model	135
5.3.2 The conventional Model	136
5.3.3 The Peer Model	136
5.3.4 The PNNI and the integrated model	137
5.4 Techniques for internetworking IP and ATM: Intra-subnet	137
5.4.1 IP and ARP over ATM – A classical approach	138
5.4.2 IP Multicast over ATM – AN extension	140
5.4.3 LAN Emulation (LANE) – An Overlay approach	142
5.5 Techniques for internetworking IP and ATM: Inter – subnet	147
5.5.1 Next Hop Address Resolution protocol (NHRP)	147
5.5.2 Integrated PNNI – A new dimension	149
5.5.3 Multi Protocol over ATM – A futuristic approach	150
5.5.4 IP Switching – An alternative approach	155
5.5.5 Multi Protocol Label Switching – An evolutionary approach	158
5.5.6 IP integrated service over ATM – An integrated approach	159
5.5.7 IPv6 and ATM – extending the boundaries	161
5.6 Summary	162

6. Discussion – Is IP.ATM integration the answer?	163
6.1 <i>IP and ATM – What are the issues?</i>	164
6.2 <i>What are our goals?</i>	165
6.3 <i>What must we achieve?</i>	166
6.4 <i>Class of service or Quality of service?</i>	166
6.5 <i>Options for High speed networks of tomorrow</i>	169
6.6 <i>A case based review</i>	172
6.6.1 <i>vBNS – A precursor to Next generation Internet initiative</i>	172
6.6.2 <i>Internet 2 – the Next generation Internet initiative</i>	178
6.6.3 <i>Sprint's ION – Future directions of multi-service providers</i>	183
6.6.4 <i>University of Rochester</i>	185
6.7 <i>Is efficiency an issue in the overall picture?</i>	189
6.8 <i>How efficient are the transport protocols associated with IP at higher wire speeds?</i>	191
6.9 <i>Is Packet over Sonet a viable alternative?</i>	192
6.10 <i>Adaptation Layers for IP in an ATM world</i>	195
6.11 <i>Summary</i>	198
7. Final thoughts and recommendations	201
Appendices	206
Appendix 1	206
Appendix 2	209
References and Bibliography	210

List of Figures

Figure 1 - ATM Cell Structure	6
Figure 2 - The ATM Protocol Reference Model	9
Figure 3 - The Physical Layer of ATM architecture	10
Figure 4 - The ATM Protocol Reference Model Sublayers	10
Figure 5 - OSI Flow into ATM	13
Figure 6 - ATM Adaptation Layer 1 Structure	14
Figure 7 - ATM Adaptation Layer 2 Structure	15
Figure 8 - ATM Adaptation Layer 3/4 Structure	15
Figure 9 - ATM Adaptation Layer 5 Structure	16
Figure 10 - Public UNI and Private NNI Illustration	17
Figure 11 - UNI and NNI frame formats	18
Figure 12 - Virtual Channel/Virtual Path Illustration	20
Figure 13 - Virtual Path and Channel illustration	21
Figure 14 - Virtual Path Switching	21
Figure 15 - Virtual Path and Virtual Channel Switching	22
Figure 16 - An Example of VPI routing	24
Figure 17 - Architecture of an input-queued packet switch	26
Figure 18 - Output queuing diagram	27
Figure 19 - An 8x8 Banyan switch	32
Figure 20 - Knockout Switch overall architecture	33
Figure 21 - Tandem Banyan Switch block diagram	34
Figure 22 - Shared Memory Switch block diagram	35
Figure 23 - Peer Model of ATM Addressing	36
Figure 24 - Overlay Model of ATM Addressing	38
Figure 25 - ATM Private Network Address Formats	39
Figure 26 - UNI and NNI Signaling	41
Figure 27 - Connection Admission Control	45
Figure 28 - The PNNI Network Hierarchy Model	48
Figure 29 - Operation of Crankback	51
Figure 30 - IP Architectural Model	57
Figure 31 - IP Detailed Architectural Model	58
Figure 32 - Internet Router - The router function is performed by the IP protocol	60
Figure 33 - Assigned Classes of Internet Addresses	60

Figure 34 - IP Routing without Subnets	63
Figure 35 - IP Routing with Subnets	64
Figure 36 - Internet Protocol (IP)	69
Figure 37 - Base IP Datagram	69
Figure 38 - IP Datagram Format	70
Figure 39 - Direct and Indirect IP Routes	73
Figure 40 - Example IP Routing Table	74
Figure 41 - IP Routing Algorithm	75
Figure 42 - Internet Control Message Protocol (ICMP)	75
Figure 43 - ICMP Message Format	76
Figure 44 - Address Resolution Protocol (ARP)	78
Figure 45 - Frame Formats for Ethernet and IEEE 802.3	79
Figure 46 - ARP Packet Reception	81
Figure 47 - Hosts Interconnected by a Router	82
Figure 48 - Proxy-ARP Router	83
Figure 49 - Reverse Address Resolution Protocol (RARP)	83
Figure 50 - Router Operation of IP	84
Figure 51 - The ARPANET Backbone	86
Figure 52 - The Counting to Infinity Problem	89
Figure 53 - IPv4 and IPv6 Header Formats	106
Figure 54 - IPv6 address formats	107
Figure 55 - Source Routing Extension Header	110
Figure 56 - MTU Discovery Process	111
Figure 57 - Tunnel Mode and Transport Mode of IPv6 Encryption	112
Figure 58 - Firewalls and Steel Pipe	113
Figure 59 - IPv4 Address Classes	117
Figure 60 - Aggregation-based Allocation Structures	114
Figure 61 - Aggregation-based IPv6 Addresses	115
Figure 62 - Subdividing the NLA Address Space	115
Figure 63 - ND Message Exchange	116
Figure 64 - Forwarded IP Traffic	117
Figure 65 - Net-Structure & Packet-Structure	121
Figure 66 - Net-Structure Packet-Structure	121
Figure 67 - IPv6 Unites Private Address Spaces	123
Figure 68 - Islands of IPv6	124

Figure 69 - RSVP Operational Model	127
Figure 70 - Receiver oriented reservation	127
Figure 71 - Instability in the Internet	129
Figure 72 - Classical Model	139
Figure 73 - VC Model for IP over ATM multicast	141
Figure 74 - Multicast server model for IP over ATM multicast	142
Figure 75 - Overlay Models	143
Figure 76 - LANE protocol model	144
Figure 77 - LAN Emulation Components and Protocol Interfaces	145
Figure 78 - LANE Control Connections	145
Figure 79 - LANE Data Connections	148
Figure 80 - NHRP Model	150
Figure 81 - MPOA in operation	154
Figure 82 - Ipsilon's IP switch implementation	155
Figure 83 - Elements of QoS support	166
Figure 84 - Q depth v Loading graph	167
Figure 85 - Desirable spectrum of Quality for multi-service delivery	170
Figure 86 - Delta of best-fit for IP/ATM integration	171
Figure 87 - Internet 2 network layout	180
Figure 88 - University of Rochester ATM backbone	186
Figure 89 - LANE topology and SSRP topology	187
Figure 90 - RFC-1483 encapsulation of IP packet	189
Figure 91 - Efficiency of IP packets on AAL5 versus the packet size.	190
Figure 92 - IP over ATM and IP over SONET performance	193
Figure 93 - Multi-service networks of the future	205

List of Tables

Table 1 - IP Routing Algorithm	75
Table 2 - Internet Control Message Protocol (ICMP)	75
Table 3 - ICMP Message Format	76
Table 4 - The counting to infinity problem	89
Table 5 - Type of service values in OSPF	95
Table 6 - BGF OSPF Attribute field mapping	101
Table 7 - Aspects of service and traffic types	168
Table 8 - IP vs. ATM features	170
Table 9- Models and generality	171
Table 10 - MPOA vs. MPLS	172
Table 11 - IP over ATM over SONET overhead	192
Table 12 - IP over PPP over SONET overhead	192
Table 13 - ATM and MPLS principles	199
Table 14 - QoS delivery in ATM and MPLS	199

Abstract

ATM is a widespread technology adopted by many to support advanced data communication, in particular efficient Internet services provision. The expected challenges of multimedia communication together with the increasing massive utilization of IP-based applications urgently require redesign of networking solutions in terms of both new functionalities and enhanced performance. However, the networking context is affected by so many changes, and to some extent chaotic growth, that any approach based on a structured and complex top-down architecture is unlikely to be applicable. Instead, an approach based on finding out the best match between realistic service requirements and the pragmatic, intelligent use of technical opportunities made available by the product market seems more appropriate. By following this approach, innovations and improvements can be introduced at different times, not necessarily complying with each other according to a coherent overall design.

With the aim of pursuing feasible innovations in the different networking aspects, we look at both IP and ATM internetworking in order to investigating a few of the most crucial topics/ issues related to the IP and ATM integration perspective. This research would also address various means of internetworking the **Internet Protocol (IP)** and **Asynchronous Transfer Mode (ATM)** with an objective of identifying the best possible means of delivering **Quality of Service (QoS)** requirements for multi-service applications, exploiting the meritorious features that IP and ATM have to offer.

Although IP and ATM often have been viewed as competitors, their complementary strengths and limitations form a natural alliance that combines the best aspects of both the technologies. For instance, one limitation of ATM networks has been the relatively large gap between the speed of the network paths and the control operations needed to configure those data paths to meet changing user needs. IP's greatest strength, on the other hand, is the inherent flexibility and its capacity to adapt rapidly to changing conditions. These complementary strengths and limitations make it natural to combine IP with ATM to obtain the best that each has to offer.

Over time many models and architectures have evolved for IP/ATM internetworking and they have impacted the fundamental thinking in internetworking IP and ATM. These technologies, architectures, models and implementations will be reviewed in greater detail in addressing possible issues in integrating these architectures in a multi-service, enterprise network. The objective being to make recommendations as to the best means of interworking the two in exploiting the salient features of one another to provide a faster, reliable, scalable, robust, QoS aware network in the most economical manner. How IP will be carried over ATM when a commercial worldwide ATM network is deployed is not addressed and the details of such a network still remain in a state of flux to specify anything concrete.

Our research findings culminated with a strong recommendation that the best model to adopt, in light of the impending integrated service requirements of future multi-service environments, is an ATM core with IP at the edges to realize the best of both technologies in delivering QoS guarantees in a seamless manner to any node in the enterprise.

1 Introduction

The **Internet protocol (IP)** suite provides the foundation for the current data communications infrastructure in the United States and much of the rest of the world. The IP protocols have proven to be very flexible and have been deployed widely over the past two decades. As the technology makes it possible to communicate at gigabit speeds, it is essential to create scalable, high-performance networks that implement IP protocols. In the past ten years, **Asynchronous Transfer Mode (ATM)** technology has emerged as a key component of the next generation networks. ATM offers unprecedented scalability and cost/performance, as well as the ability to reserve network resources for the real-time oriented traffic and support for multipoint communication.

ATM stands for "Asynchronous Transfer Mode". It is a suite of communication protocols designed to support integrated voice and data networks. It was initially developed as a standard for wide-area broadband networks. The fact that local ATM networks are appearing in advance of long-haul ATM networks makes ATM an attractive alternative to traditional LANs. It is primarily driven by telecommunications companies and is a proposed telecommunications standard for **Broadband ISDN (BISDN)**.

Asynchronous Transfer Mode (ATM) is a network technology used to send information at extremely high speeds. ATM is a cell-switching and multiplexing technology designed to combine the benefits of circuit switching (constant transmission delay, guaranteed capacity) with those of packet switching (flexibility, efficiency of intermittent traffic). ATM carries with it many benefits to network users and administrators. It uses small fixed length cells and is capable of carrying different types of network traffic (voice, video, or data) over the same network.

Asynchronous Transfer Mode (ATM) technology is emerging as an important worldwide standard for the transmission of information. Rapidly being deployed by telephone companies and enterprise customers, ATM represents for many of the next generation for **Local Area Network (LAN) / Wide Area Network (WAN)** switching & internetworking. ATM can accommodate simultaneous transmission of data (includes imagery), fax, voice, and real-time video. Unique features such as **Quality of Service (QoS)** further enhance the support of mixed media (voice, video, and data) and associated delivery requirements. ATM-only (Native ATM) enterprise network is a router-less system, switched environment.

On the other hand, with the prevalence of the multitude of technologies we are faced with issues of internetworking them in making them inter-operable across the board so we have a seamless integration. Internetworking is sharing of computer resources by connecting the computers through a number of data communications networks. The network can be a private or public network; they can be local or wide area networks. Internetworking allows the users of different networks to exchange information with each other.

ATM has the potential to remove the performance bottlenecks in today's LANs and WANs. It redefines the basic unit of LAN data transportation. Short, fixed length cells that can carry voice, video, and data at very high speeds replace the variable length packets. ATM has the potential to redefine the networking industry and cause a literal paradigm shift in the way networks are built and used. The deployment of ATM to support other networks is consistent with the trend towards increased use of ATM in multi-service networks. The carriers and service providers can accommodate the continuing growth of Frame Relay and LAN services; ATM offers high-speed trunks that permit the negotiation of **Quality of Service (QoS)** features, such as delay, throughput, and peak burst rate. These features are not supported by most other technologies. This multitude of features is what makes ATM the fastest growing area of the networking industry.

The explosive growth of the Internet has given IP routing new challenges. Innovative solutions have recently been proposed to solve the current router bottleneck by evolving to a tighter integration of IP and ATM. However, the challenge is to closely integrate the IP and ATM routing and forwarding functions whilst differentiating them according to type of traffic towards an IP over ATM dual-mode routing model.

Although IP and ATM often have been viewed as competitors, their complementary strengths and limitations form a natural alliance that combines the best aspects of both the technologies. For instance, one limitation of ATM networks has been the relatively large gap between the speed of the network paths and the control operations needed to configure those data paths to meet changing user needs. IP's greatest strength, on the other hand, is the inherent flexibility and its capacity to adapt rapidly to changing conditions. These complementary strengths and limitations make it a natural choice to combine IP with ATM to obtain the best that each has to offer. It is the objective of the author to research the existing and future methodologies/techniques of transporting IP over ATM. This would address implementation alternatives, associated issues and will scale to address possible integration perspectives in the Enterprise networks.

In unraveling the intricacies involved in the IP/ATM integration perspective, we take a layered approach in looking at ATM and IP architecture and internetworking models in their own merit before dwelling into the integration of both. The thesis report per se follows this logically in initially addressing the research scope, methodology and limitation (Section 2); the material relevant to ATM architecture, internetworking models, associated implementations and issues with a view of integrating it with higher layer protocols such as IP and this relevant findings are presented in Part I of the research findings (Section 3); the wider aspects of Internet protocol (IP) architecture and internetworking models and their limitations in light of growing requirements based on **Quality of service (QoS)** with the multitude of technologies associated with these are presented in Part II of the research findings (Section 4); the integration models and perspectives of IP and ATM are presented in Part III of the research findings (Section 5).

The trends in integrating IP and ATM are reviewed in light of global trends in terms of QoS, performance and complexity with the current drive to integrate them into preferably, one platform. A future safe IP and ATM routing paradigm must satisfy all the requirements of existing networks in terms of both unicast and multicast traffic. The major trade-off for providing high-performance solutions is complexity. The key to efficient IP and ATM routing is a good balance between complexity and performance. The final thoughts and recommendations are formulated considering these factors basing them on the authors research as well as on his own exposure/experience gained in working at the University of Rochester as a Telecommunications Research Engineer, primarily focusing on the two infrastructural projects such as the Internet2 (IP over ATM) and the switching project (Voice over ATM). These are used as case reviews in drawing conclusions and recommendations, further drawing on materials made available by NEC Business Communications Systems, Cisco Systems Inc. as the prime solution partners associated with these projects at University of Rochester.

2. Scope, Methodology and Limitations of this Research

2.1 Scope of research

ATM as a data link/network layer networking technology presents unique challenges to the IP layer if it is desired to make the most effective use of the ATM layer.

The first challenge is that the existing Internet model assumes that there is a one-to-one mapping between a particular network layer and a **Logical IP Subnet (LIS)**, and that any communication between IP hosts on different LISs will require an IP router(s) to forward packets between those LISs. Since ATM is capable of supporting many distinct LISs on the same physical network, strictly following the existing Internet model requires traffic between two IP/ATM hosts that are on the same ATM network but in different LISs, to direct their traffic through an intermediate router even though there is a direct pathway between the IP/ATM hosts at the ATM network layer. This effectively doubles the traffic on the ATM network unnecessarily, reduces the network performance, and increases delay. The second challenge that ATM presents is that it is unique as a data link/network layer technology in that plans are being made and actual network infrastructure is being implemented to create a global ATM infrastructure as a single logical ATM network. The scope of such a global ATM network will be similar to that of the existing Internet. As such, it will have to deal with the same problems of scaling the addressing and routing that the current Internet is facing, and that the next generation of IP (IPng/IPv6) has been designed to address.

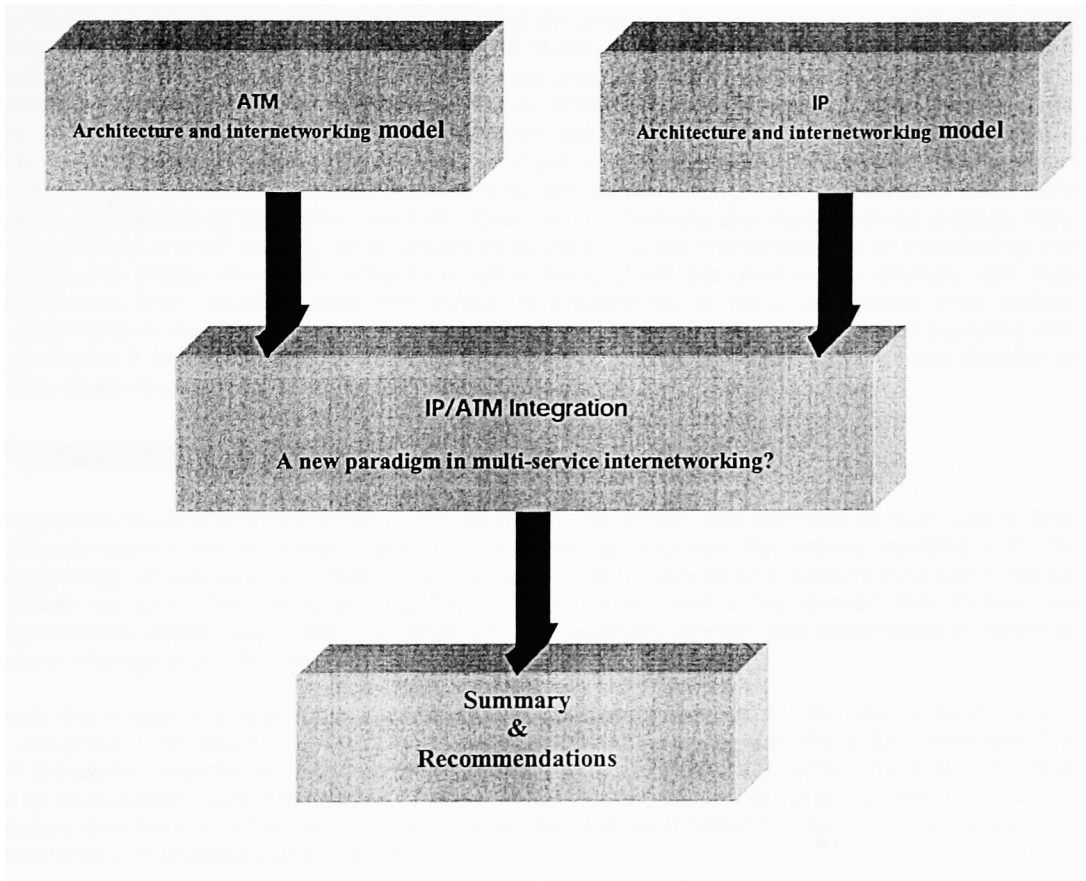
From the IP perspective, all the previous data link/network layers it was required to deal with were of a fairly limited scope, such as a **Local Area Network (LAN)** like Ethernet or a **Metropolitan Area Network (MAN)** like **Switched Multimegabit Data Services (SMDS)**. ATM is the first network layer that IP has to interface with that is designed and intended to have a global scope and the scale of the Internet. One of the main challenges here is the mapping between the IP and ATM layer addresses and the integration of the IP and ATM layer routing, when both are global in scope. The real challenge in confronting these difficulties is devising a solution that will allow the advantages and capabilities of both ATM and IP to be combined and fully integrated with one another, rather than having them compete against one another. Users should be presented with a seamless interface to these dual technologies, such that the underlying technologies, whether IP or ATM, are for the most part transparent.

We need to end the die-hard IP camp and the die-hard ATM camp approach. Nothing productive is gained by those within the IP camp who say that ATM will never fly nor by those within the ATM camp who say that ATM is the world's solution for networking which will make IP obsolete. IP and ATM should not only peacefully coexist with one another, but they should be fully integrated so that they can both benefit from each other's strengths in a symbiotic fashion, with the user being the ultimate beneficiary, getting the best of both worlds.

The proposed research would address the broader issues associated with this integration perspectives of ATM and IP towards a new paradigm shift in internetworking and integration perspectives. This would highlight and identify key features in each technology, which facilitate a deeper and broader understanding of issues that may have to be addressed in carrying IP over ATM or integrating them with related technologies. As such the research will basically focus on the following.

- ↳ **ATM Architecture and Internetworking**
- ↳ **IP Architecture and Internetworking**
- ↳ **IP Over ATM or IP/ATM integration? – A paradigm shift**
- ↳ **Summary and recommendations**

2.2 Research Model



2.3 Methodology of research

The methodology adopted for this specific research is inductive in nature, with an extensive literature review and associated study bordering on previous and on going research on IP, ATM technologies and implementations with the goal of integration the two to exploit the features and benefits offered by both. The literature review is not only limited to published research material, but covers the broader spread of magazines, forums, notes conference, white papers, etc. Specific data and case studies used herein those made available to me by the Telecommunications Division at University of Rochester and their principal technology partners, namely Cisco Systems and NEC Business Communications Systems, who are implementing their Voice and IP over ATM network at University of Rochester. Since this relates to my firsthand exposure on these projects they will be used as one of the key case studies in drawing our recommendations in concluding our research, but where necessary data from other sources will be used in the analysis with due reference to their sources. With the future developments in mind we would also review developments in the areas of IPng, 6bone and Mbone in order to be exhaustive in covering the requirements it will place on the backbone technology such as ATM in light of the ***Quality of Service (QoS)*** requirements in integrating IP with it.

2.4 Limitations of research

This research/study is limited by the scope as alluded to earlier and will base its conclusions and recommendations on available data and material as obtained by means described in the methodology of research and will be limited by the accuracy of the sources and data made available as such. The prime limiting factor in this study being the desired lab facilities to independently design, build, test and measure the necessary models and parameters in verifying previous findings and in formulating new ones.

Further, the research scope is limited to the IP and ATM integration in the Enterprise networks which are assumed to be private networks and does not extend it's purview to the public networks. The term Enterprise networks are used to refer private network infrastructure, which facilitates the free flow of multimedia applications such as data, video and voice. Geographically, the same would include Local Area and Metropolitan area networks. One such network, which shall be alluded to in here, is that of University of Rochester.

3. Research Findings – Part I

Asynchronous Transfer Mode (ATM) – Architecture and internetworking

As a precursor to our research on IP and ATM, based on our research model, we will review architectural and internetworking models associated with IP and ATM. We do so, since we feel an in depth understanding of ATM and IP architectural and internetworking models is essential for successful evaluation and implementation. of such models for merging or integrating the two technologies in building an network infrastructure in order to cater for QoS and bandwidth requirements of various multimedia applications

ATM is a widespread technology adopted by many operators to support advanced data communication, in particular efficient Internet services provision. The expected challenges of multimedia communication together with the increasing massive utilization of IP-based applications urgently require redesign of networking solutions in terms of both new functionalities and enhanced performance. However, the networking context is affected by so many changes, and to some extent chaotic growth, that any approach based on a structured and complex top-down architecture is unlikely to be applicable. Instead, an approach based on finding out the best match between realistic service requirements and the pragmatic, intelligent use of technical opportunities made available by the product market seems more appropriate. By following this approach, innovations and improvements can be introduced at different times, not necessarily complying with each other according to a coherent overall design.

With the aim of pursuing feasible innovations in the different networking aspects, we are investigating a few of the most crucial topics related to the IP and ATM integration perspective. Towards this end in this chapter we will review the ATM model, the protocol layering, it's operations and how it scales into larger networks and the protocols which facilitate ATM to be a key underlay network for carrying diverse traffic ranging from data, voice and video across the broader spread of network topologies and architectures. This will help us understand how well we could integrate IP and ATM in future, enterprise networks.

3.1. ATM – An overview

ATM is an advanced implementation of packet switching that provides high-speed data transmission rates to send fixed-size packets over broadband and baseband **Local Area Networks (LAN)** or **Wide Area Networks (WAN)**. ATM is an extension of **Broadband Integrated Services Digital Network (B-ISDN)**¹ concept. ATM is flexible and scalable. At one extreme, ATM is capable of operating over multiple satellite hops (256Kbps). At the other extreme, ATM can operate at **Synchronous Optical Network (SONET)**² rates from OC-3 (155.52 Mbps) to OC-48 (2.49 Gbps) and even scaling as high as OC-96 (9.6 Gbps). ATM is a broadband cell relay method that transmits data in 53 bytes (octets) cells rather than in variable-length frames. These cells consist of 48 bytes (octets) of application information (payload) with five additional bytes of ATM header data. For example, ATM would divide a 2000-byte packet into 42 data frames and put each data frame into a cell. ATM's fixed cell size can be quickly processed and switched.

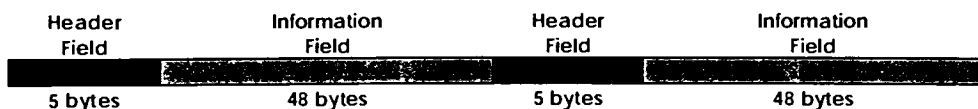


Figure 1. ATM Cell Structure³

¹ William Stallings, *ISDN and Broadband ISDN with Frame Relay and ATM*, Prentice Hall, 1995

² Daniel Minoli, *Enterprise Networking*, Artech House, 1993

³ ATM Tutorial, <http://juggler.lanl.gov/lanp/am.tutorial.html>

ATM is a cell-based protocol that exists at the physical/data link layers and therefore resides in the hardware of device. It is a switched protocol that will operate in either a connection oriented (similar to WAN data link) or connectionless method (broadcast medium like Ethernet). ATM's small fixed size cells maintain a high level of consistency throughout the network in the amount of delay that is encountered. ATM's protocol is self-sufficient; it does not rely on anything in particular from the upper layer. This allows ATM to not be terribly particular about the type of traffic it carries. ATM provides cells in a bit format; therefore it will theoretically run across any media, at any speed. ATM provides very low latency because ATM has no error control on payload and the payload will not be checked for transmission error. Only the cell header will be checked for error.

ATM is not an asynchronous transmission technique, but a switching and multiplexing process. In the ATM context, the term "asynchronous" refers to the method by which cells are transferred across a switching element.⁴ The transfer of cells is asynchronous in the sense that the recurrence of cells containing information from an individual channel is not necessarily periodic. Technologies that fragment data into small pieces can have disastrously low performance if any of the pieces are lost in transit. Because there usually is no mechanism in the cell network to detect and retransmit lost cells, the only way to recover the damaged packet is to retransmit the entire large packet again. Thus two copies of the packet will be sent because a single cell was lost. The throughput of the network is considerably reduced. One alternative is to retransmit the entire packet and the other is to retransmit the individual cells that are lost. However, as noted previously, cell network designs assume that cells will not be retransmitted. This is because retransmission scheme requires that each cell be uniquely tagged, typically with a sequence number, so the receiver can request a particular cell be retransmitted and the transmitter can properly identify the cell its is to retransmit. Unfortunately, at gigabit rates, a sequence number would need to be about 8 bytes long.

ATM is the complement of "**Synchronous Transfer Mode (STM)**".⁵ In STM, time-division multiplexing is employed to pre-assigned users to time slots. ATM time slots are made available on demand with labels identifying the source of transmission in each cell. **Time Division Multiplexing (TDM)** is inefficient relative to ATM because, if a station has nothing to transmit when its time slot comes up, that time slot is wasted. The converse situation, where one station has lots of information to transmit, is also less efficient. In this case, the station can only transmit when its turn comes up, even though all the other time slots may be empty. STM services dedicate a physical path to a voice call for the duration of the call. No other call can use this facility. Once the call is completed, everything is torn down and made available for use by the next call. ATM - asynchronous transfer mode is a sophisticated well conceived technology.

An ATM logical connection defines a path through the network from a transmitter to a receiver. This requires setting up appropriate entries in routing tables (a look up table which converts header-in to port number-out and replacement header-out) of each switch along the way. It is in the setting up of these tables that one of the problems of ATM appears. There was no standard way of doing this a few years back, but since then ATM Forum has made progress in this area in terms of PNNI and related techniques for auto-configurations of switches and intermediate nodes. But it is also done manually at each switch on the network, or by 'vendor unique' software. This has some implications in terms of scalability whilst an ATM network is almost infinitely flexible, it makes it very difficult to access that flexibility. Standards based techniques which being developed which scales well with the current scope of ATM and there are various methods by which this can be accomplished and they are reviewed in detail later on.

Implicit in the definition of ATM is that both transmitting and receiving connection have some buffering capability, since if packets addressed to one output link arrive simultaneously on several input links, they cannot all be forwarded instantaneously but must be held until they can be forwarded. However, this buffering must be finite, so it is possible to overload an output link if

⁴ Koichi Asatani et al, *Introduction to ATM networks and B-ISDN*, John Wiley & Sons, 1997

⁵ William Stallings, *ISDN and Broadband ISDN with Frame Relay and ATM*, Prentice Hall, 1995

several input links all try to send data to it, at a total rate greater than its capacity. ATM has no flow control mechanism, and will therefore simply discard the excess packets. Therefore, when setting up a logical link, the maximum rate of sending packets must be negotiated so that it will not exceed the capacity of any of the links along its route. If a certain capacity is negotiated, but not used, it cannot reliably be grabbed by other users. This, of course, is not a problem in the telecommunications market, where fixed audio and video channels provide a steady stream of data at a known transfer rate. However, it is less suitable for computer transfers or video postproduction transfers, where the data flow is bursty, because (unlike computer-originated networks such as Ethernet) the bandwidth cannot be shared on an as-needed basis. It is very difficult to say whether ATM will be suitable for transferring live non-compressed video streams until significantly more data bandwidth is available.

ATM is based upon the concept of two end terminals communicating to each other via a set of intermediate switches. The ATM protocol functions similar to a phone conversation⁶. In order for a session to begin, person A sets up a call with person B. Once person B picks up the phone, a connection is established, and the conversation begins (or in the case of ATM, data transfer commences).

If a network user wishes to communicate with another terminal it must first determine its address (phone number). Then the system must determine the best route for that information to take. Once that is determined, the system transmitting the information sends a connection request to the destination system. When setting up this connection, the user also specifies other information about the upcoming transmission. For instance the speed of transmission and the quality of service. Its like sending mail, you can choose 1st class, 4-day delivery, or FedEx.

Using ATM, information is segmented into 53 byte cells. With a 5 byte header (containing address and file information). And a 48 byte payload, which holds the information (voice, video, or data). This cell structure of fixed sized packets, allows for voice, video, and data to travel over the same network while making sure that no single type of data hogs the line. In order for ATM to be active over a network, the information must be formatted in a certain way (i.e. chopped up into 53 byte cells for transportation, given address information, etc.) this process is accomplished as the data travels through different "layers" of the ATM protocol⁷.

3.2. ATM Architecture

The ATM protocol is made up of 3 layers. Each responsible for handling information in their own way. The information (voice, video, or data) must travel through these layers from the top down, in order to be sent out. Once the information arrives at its destination, it must go back through these layers from the bottom up, so the information can be translated back to its original form.

There are 3 Main layers⁸,

- ☞ ATM Adaptation layer
- ☞ ATM Layer
- ☞ Physical Layer.

The Top Layer, is the ATM Adaptation Layer. It's responsible for mapping information into and out of the ATM cells. The middle layer is the ATM Layer. It provides the addressing information for each cell. It uses a system called **Virtual Path Identifier (VPI)** and **Virtual Channel Identifier (VCI)**⁹ labeling,

⁶ Thomas M. Chen, Stephen S. Liu, *ATM Switching Systems*, Artech House, 1995

⁷ I.J. Duffy Hines, *ATM - The key to High-speed Broadband Networking*, M&T Books, 1996

⁸ Koichi Asatani et al, *Introduction to ATM networks and B-ISDN*, John Wiley & Sons, 1997

⁹ William Stallings, *ISDN and Broadband ISDN with Frame Relay and ATM*, Prentice Hall, 1995

to identify the destination of each cell. The bottom layer is the Physical Layer. It puts the bits on the wire. This layer is responsible for formatting the information into the proper physical format, so it can be sent from one ATM node to the next. ATM is very flexible because it can travel over a variety of physical formats (coax, fiber optics, or twisted pair wiring).

Next, we will look deeper into the mechanics of each layer through the exploration of sub-layers. Figure 2 illustrates ATM protocol reference model for ATM communication. The ATM communication will take place through each layer. The function of each layer will be discussed in the later sections. The ATM protocol reference model consists of three planes, as shown in figure 2.

- ↳ **User plane** for transporting user information.
- ↳ **Control plane** is responsible for call control and connection control functions and it contains mainly signaling information.
- ↳ **Management plane**, which contains layer management functions and plane management, functions.

There is no defined (or standardized) relationship between OSI layers and ATM protocol model layers. But the following relations can be found. The Physical layer of ATM is almost equivalent to layer 1 (Physical Layer) of the OSI model and it performs bit level functions. The ATM layer can be equivalent of the lower edge of the layer 2 of the OSI model. The ATM Adaptation Layer performs the adaptation of OSI higher layer protocols.

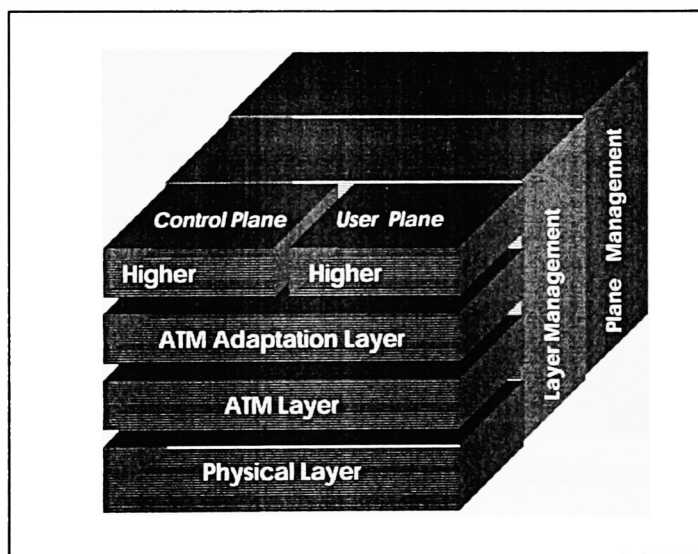


Figure 2. The ATM Protocol Reference Model

3.2.1 The Physical Layer of ATM Architecture

ATM is a physical layer architecture and hardware based, not software. The ATM layer is broken into three primary sublayers: The **ATM Adaptation Layer (AAL)**, **ATM Cell Switching Layer (ATML)** and the **Physical Layer (PHY)**. There is no standard in upper layer protocol that always produces a 48-octet packet of data, voice, or video. Therefore, ATM incorporates the process of adapting the upper layer protocols to fit into cell structure through adaptation layer. Figure 3 illustrates the physical layer of ATM architecture. Note that AAL is implemented only in ATM endpoint devices (Source or destination computer), not in ATM switches.

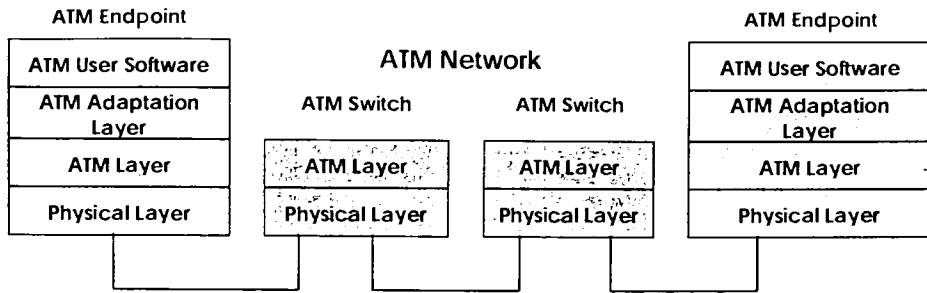


Figure 3. The Physical Layer of ATM architecture ¹⁰

The ATM Protocol Reference Model sub layers and their respective functions are shown in figure 4.

Convergence Sublayer (CS)		AAL
Segmentation and Reassembly (SAR)		
Generic Flow Control		ATM
Cell VPI/VCI translation		
Cell multiplex and demultiplex		
Cell header generation/extraction		PHY
Cell rate decoupling	TC	
HEC header sequence generation and verification		
Cell delineation		
Transmission frame adoption		
Transmission frame generation/recovery	PM	
Bit timing		
Physical Medium		

Figure 4. The ATM Protocol Reference Model Sublayers

3.2.1.1 Physical layer Functions

As shown in figure 4, Physical Layer is divided into two sublayers¹¹.

- **Physical Medium (PM)** sublayer
- **Transmission Convergence (TC)** sublayer

The **PM sublayer** contains only the Physical Medium dependent functions. It provides bit transmission capability including bit alignment. It performs line coding and also electrical/optical conversion if necessary. Optical fiber will be the physical medium and in some cases, coaxial and twisted pair cables are also used. It includes bit-timing functions such as the generation and reception of waveforms suitable for the medium and also insertion and extraction of bit timing information.

¹⁰ Koichi Asatani et al, *Introduction to ATM networks and B-ISDN*. John Wiley & Sons, 1997

¹¹ J.J. Duffly Hines, *ATM - The key to High-speed Broadband Networking*. M&T Books, 1996

The **TC sublayer** mainly does five functions as shown in the figure 4. The lowest function is generation and recovery of the transmission frame. The next function i.e. transmission frame adaptation takes care of all actions to adapt the cell flow according to the used payload structure of the transmission system in the sending direction. It extracts the cell flow from the transmission frame in the receiving direction. The frame can be a **Synchronous Digital Hierarchy (SDH)**¹² envelope or an envelope according to **International Telecommunications Union (ITU)** Recommendation.

Cell delineation function enables the receiver to recover the cell boundaries. Scrambling and Descrambling are to be done in the information field of a cell before the transmission and reception respectively to protect the cell delineation mechanism. The **Header Error Control (HEC)** sequence generation is done in the transmit direction and its value is recalculated and compared with the received value and thus used in correcting the header errors. If the header errors can not be corrected, the cell will be discarded. Cell rate decoupling inserts the idle cells in the transmitting direction in order to adapt the rate of the ATM cells to the payload capacity of the transmission system. It suppresses all idle cells in the receiving direction. Only assigned and unassigned cells are passed to the ATM layer.

3.2.1.2 ATM layer functions

ATM layer is the layer above the physical layer. As shown in the figure 4, it does the four functions that can be explained as follows.

Cell header generation/extraction: This function adds the appropriate ATM cell header (except for the HEC value) to the received cell information field from the AAL in the transmit direction. VPI/VCI values are obtained by translation from the **Service Access Point (SAP)**¹³ identifier. It does opposite (i.e. removes cell header) in the receive direction. Only cell information field is passed to the AAL.

Cell multiplexing and demultiplexing: This function multiplexes cells from individual VPs and VCs into one resulting cell stream in the transmit direction. It divides the arriving cell stream into individual cell flows VC or VP in the receive direction.

VPI and VCI translation: these functions are performed at the ATM switching and/or cross connect nodes. At the VP switch, the value of the VPI field of each incoming cell is translated into a new VPI value of the outgoing cell. The values of VPI and VCI are translated into new values at a VC switch. Detailed description of VPI and VCI discussed in the following section.

Generic Flow Control (GFC)¹⁴: This function supports control of the ATM traffic flow in a customer network. This is defined at the ATM **User-to-network interface (UNI)**¹⁵.

3.2.1.3 ATM Adaptation Layer (AAL)¹⁶

The major function of AAL is to prepare the upper layer data for transmission into the network, on receiving end, check to ensure that it has been correctly received. To perform this task AAL is broken into two sublayers, the **Convergence Sublayer (CS)** and the **Segmentation and Reassembly Sublayer (SAR)** as shown in figure 4. The CS is responsible for preparation of data for segmentation and the checking after reassembly. The CS is broken down into two sublayers, **Service Specific Convergence Sublayer (SSCS)** and **Common Part Convergence Sublayer (CPCS)**. The two CS sublayers were created to handle different traffic types. There are three primary uses

¹² Byeong Gi Lee, Minho Kang, Jonghee Lee, *Broadband Telecommunications Technology*, Artech House, 1996

¹³ Andrew S. Tanenbaum, *Computer Networks*, Prentice Hall, 1996

¹⁴ Ulyess Black, *ATM: Foundation for Broadband Networks*, Prentice Hall, 1995

¹⁵ Andrew S. Tanenbaum, *Computer Networks*, Prentice Hall, 1996

¹⁶ Koichi Asatani et al, *Introduction to ATM networks and B-ISDN*, John Wiley & Sons, 1997

for SSCS, Frame Relay over ATM, *LAN Emulation (LANE)*¹⁷ and voice over ATM; other than these three, this layer is skipped or inactive. When Frame Relay frame goes through this layer, SSCS defines how to encapsulate the frame. For LANE, SSCS defines an additional addressing mechanism that is used. The SSCS will provide the clock recovery mechanism for voice and video over ATM. If there is a timing gap between the cells of voice and video stream due to traffic, the clock recovery will correct the timing gap such that the sequence of video and voice stream cells arrive exactly the way it was sent. This is done by time stamping the voice and video cells.

The CPCS and SAR sublayers are together responsible for getting user traffic in and out of cells. These layers are utilized in different manners depending on the type of traffic being carried. The primary function of CPCS is to ensure that the frame delivered out of ATM is the same as the frame originally sent into ATM. The SAR does the slicing and dicing of the frames into 48 octets payload to fit into ATM cell format. The 48 octets payload piece is passed to ATM layer. This layer adds a 5-octet cell header, which includes addressing information for the cell to be routed in the network through Physical Layer.

3.2.1.3.1 Service Specific Convergence Sublayer (SSCS)¹⁸ Flow

The frames from the Data Link Layer flows into Convergence Sublayer with all of the headers and trailers from the above layers. The frame first travel through the SSCS, if this layer is in use a header will be added to the frame. The header is typically two octets. For Frame Relay over ATM and LANE this header contains addressing information, and for voice it contains clocking information. If SSCS is not in use the frame will travel from Data Link Layer directly into the CPCS sublayer, a header and/or a trailer will be added to the frame.

3.1.3.2 Common Part Convergence Sublayer (CPCS)¹⁹ Flow

The purpose of CPCS sublayer is to add headers and/or trailers depending on the type of traffic passing through ATM. The information contained in these headers and trailers are items such as buffer allocation information and *Cyclic Redundancy Check (CRC)*²⁰ information for error detection and correction of ATM cell headers.

3.1.3.3 Segmentation and Reassembly Sublayer (SAR)²¹ Flow

The function of SAR is to segment the frame into 48 octet pieces and reassemble the frame at the receiving end of the network. If SAR need to communicate with SAR on receiving end of the network, a header and/or a trailer may be added. SAR will chop the frame from upper layer into small pieces and one by one add headers and/or trailers. This piece is passed to ATM layer (ATM Cell Switching).

3.2.1.4 ATM Layer (ATM Cell Switching)²²

In this layer a 5 octets ATM cell header is added to the 48 octets of payload. The cell header is the only header read by the ATM switching devices inside of the network. The rest of the headers and trailers are read and interpreted at the edge devices. Cell header includes addressing information so that the cell can be routed through the network.

¹⁷ I.J. Duffy Hines, *ATM - The key to High-speed Broadband Networking*, M&T Books, 1996

¹⁸ Ulysses Black, *ATM: Foundation for Broadband Networks*, Prentice Hall, 1995

¹⁹ Ulysses Black, *ATM: Foundation for Broadband Networks*, Prentice Hall, 1995

²⁰ Andrew S. Tanenbaum, *Computer Networks*, Prentice Hall, 1996

²¹ Ulysses Black, *ATM: Foundation for Broadband Networks*, Prentice Hall, 1995

²² I.J. Duffy Hines, *ATM - The key to High-speed Broadband Networking*, M&T Books, 1996

3.2.1.5 Physical Layer (Physical Path)²³

The Physical layer takes the 53 octets ATM cell from ATM layer and converts it to signals representing bits and transmits them to ATM switches and/or endpoint devices. It consists of the *Physical Medium sublayer (PM)* and the *Transmission Convergence sublayer (TC.)*

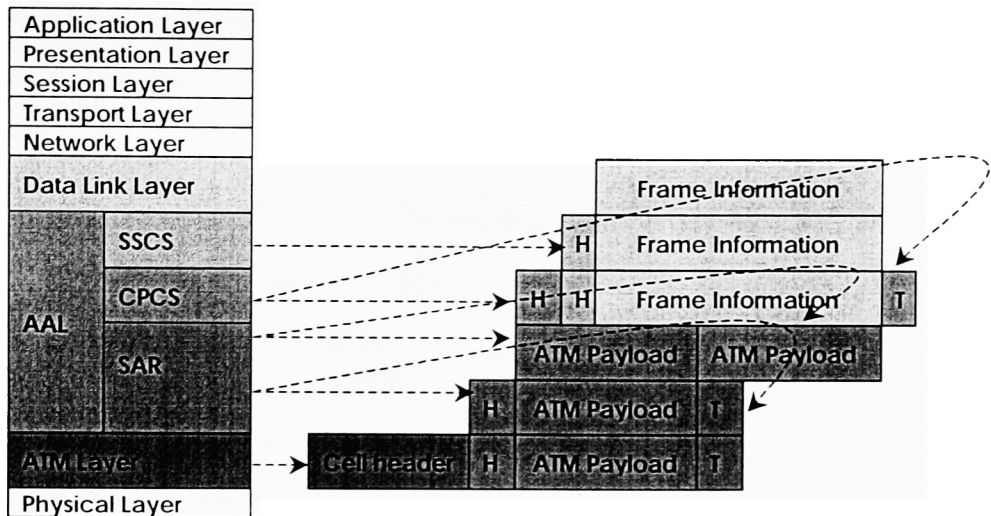


Figure 5. OSI Flow into ATM

3.2.1.6 Open Systems Interconnect (OSI)²⁴ Flow into ATM

Figure 5 illustrates the data flow into ATM. At each of the layers of the OSI model, headers and possibly trailers are added so that the layer communicate with its peer layer at the other side of the network or link.

3.2.2 ATM Adaptation Layer Types

In order for ATM to support many kinds of services with different traffic characteristics and system requirements, it is necessary to adapt the different classes of applications to the ATM layer. This function is performed by the AAL, which is service dependent. Four types of AAL were originally introduced. Two of these have now been merged into one. Also, within the past year a fifth type of AAL has been proposed. The following is the description of the four *ATM adaptation layers (AAL)*²⁵ defined up to date.

AAL1:

AAL1 is the abbreviation for *ATM Adaptation Layer 1 and is a communications service intended for continuous data flowing* from a source over an ATM network to a destination without interruption. Source and destination must use the same timing information, or the AAL1 service must carry it. ALL1 supports connection-oriented services that require constant bit rates and have specific timing and delay requirements.

Applications that are suited are:

- 25 Circuit transport to support synchronous (e.g. 64Kbit/s) and asynchronous (e.g. 1.5, 2 Mbps) circuits. Video signal transport for interactive and distributive services.

²³ Koichi Asatani et al, *Introduction to ATM networks and B-ISDN*, John Wiley & Sons, 1997

²⁴ Andrew S. Tanenbaum, *Computer Networks*, Prentice Hall, 1996

²⁵ William Stallings, *ISDN and Broadband ISDN with Frame Relay and ATM*, Prentice Hall, 1995

- ⌘ Voice band signal transport.
- ⌘ High quality audio transport

The following is the structure of AAL1 as shown in figure 6:

- ⌘ 47 bytes of payload per cell
- ⌘ Sequence Number used to detect lost or misinserted cells
- ⌘ Sequence Number Protection - Maybe used to provide error-detection and error-correction capabilities for the sequence number field

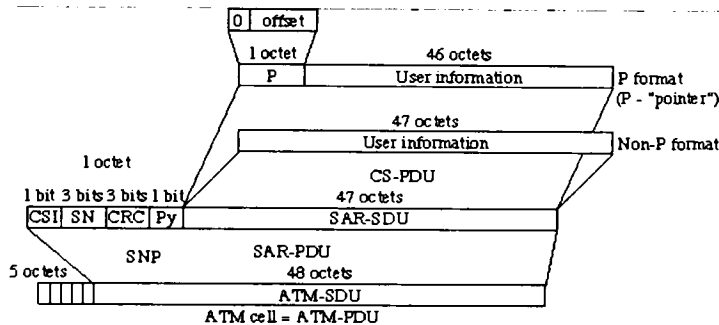


Figure 6. ATM Adaptation Layer 1 Structure²⁶

- P = octet offset of data block over 2 cells (= 0111 111 if not required)
- CSI = Convergence Sublayer Indication
- CRC = Cyclic Redundancy Check
- SN = Sequence Number (non-P; CSI = 0 P format; CSI = 1 only if SN = 0,2,4 or 6)
- SNP = Sequence Number Protection
- Py = Parity (even, 1 bit)
- SAR = Segmentation And Reassembly
- PDU = Protocol Data Unit

Speech data are suited for transmission using AAL1. As an example encoded voice continuously generates one 8-bit sample every 125 microseconds. This is **Constant Bit Rate (CBR)** data, sometimes also described as **Constant Bit rate Oriented (CBO)** data and is thus, in principle, suited to AAL1. Delay may be a problem here. If 47 samples are collected before a cell is transmitted, this means a collection delay of $47 \cdot 125 \text{ micros} = 5.875 \text{ milliseconds}$. The same amount of time may be required at the receive side, making a minimum total delay of 11.75 ms. If this delay is significantly added to by the network, then echo may start to become a problem. If several signals can share an ATM cell, then this problem is reduced e.g. 30 voice channels would each suffer only one thirtieth of the delay, which can then be neglected.

AAL2:

AAL 2 has not currently well defined, but services for this type may include:

- ⌘ Supports connection-oriented services that do not require constant bit rates. In other words, transfer of service data units with a variable source bit-rate.
- ⌘ Transfer of timing information between source & destination.

The following is the structure of AAL2 as shown in figure 7:

- ⌘ 45 bytes of payload per cell
- ⌘ Item Type - Used to indicate beginning of message, continuation of message, or end of message and also a component of the video or audio signal
- ⌘ Length Indicator - Indicates the number of octets of the **CS (Convergence Sublayer) PDU** that are included in the **SAR (Segmentation And Reassembly) payload**

²⁶ George C. Sackett, Christopher Y. Metz, *ATM and Multiprotocol Networking*, McGraw-Hill, 1996

- ↳ Cyclic-Redundancy-check code - Used to detect errors up to two correlated bit errors in the SAR (Segmentation And Reassembly) **PDU (Protocol Data Unit)**

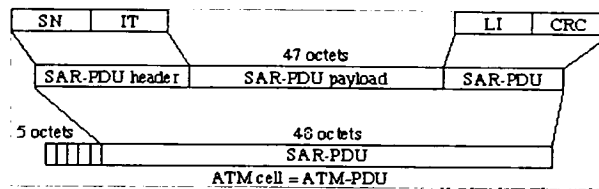


Figure 7. ATM Adaptation Layer 2 Structure²⁷

CRC = Cyclic Redundancy Check
 IT = Information Type
 LI = Length Indication
 PDU = Protocol Data Unit
 SAR = Segmentation And Reassembly
 SN = Sequence Number

AAL3/4:

is AAL is intended for both connectionless and connection oriented variable bit rate services. Originally two distinct adaptation layers AAL3 and 4, they have been merged into a single AAL which name is AAL3/4 for historical reasons.

The following is the structure of AAL3/4 as shown in figure 8:

- ↳ 44 bytes of payload per cell
- ↳ Multiplexing Identification - Provides for the multiplexing and demultiplexing of multiple CS PDUs concurrently over a single ATM connection. All SAR PDUs of a given CS PDU will have the same MID value.

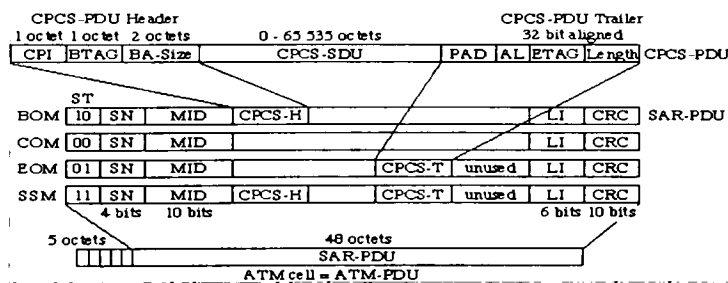


Figure 8. ATM Adaptation Layer 3/4 Structure²⁸

AL = 32 bit Alignment (using 0)
 Btag = Beginning Tag field per CPCS-PDU must be identical to Etag
 BA-size = Buffer Allocation size
 BOM = Beginning Of Message
 COM = Continuation Of Message
 CPI = Common Part Indicator (0 = octet as unit for BA-size, all other values for further study)
 CPCS = Common Part Convergence Sublayer
 CPCS-H = CPCS-Header
 CPCS-T = CPCS-Trailer
 CRC = Cyclic Redundancy Check
 EOM = End Of Message
 Etag = End Tag field - see Btag

²⁷ George C. Sackett, Christopher Y. Mei: *ATM and Multiprotocol Networking*, McGraw-Hill, 1996

²⁸ Koichi Asatani et al, *Introduction to ATM networks and B-ISDN*, John Wiley & Sons, 1997

IT = Information Type
Length = Length of CS-SDU
LI = Length Indication (max 44 octets)
MID = Multiplex Identification
PAD = Padding (0)
PDU = Protocol Data Unit
SAR = Segmentation And Reassembly
SDU = Service Data Unit
SN = Sequence Number
SSM = Single Segment Message
ST = Segment Type

AAL5:

Supports connection-oriented variable bit rate data services. It is a substantially lean AAL compared with AAL3/4 at the expense of error recovery and built in retransmission. This tradeoff provides a smaller bandwidth overhead, simpler processing requirements, and reduced implementation complexity. Some organizations have proposed AAL5 for use with both connection-oriented and connectionless services.

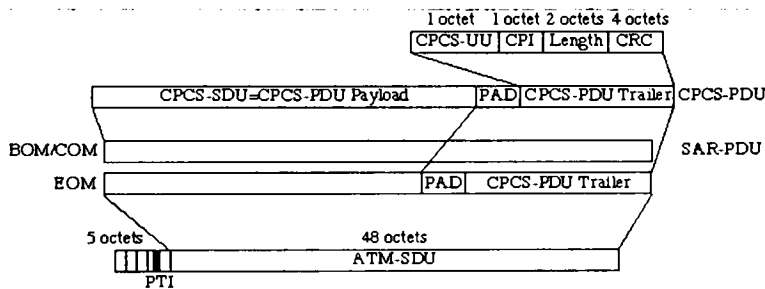


Figure 9. ATM Adaptation Layer 5 Structure²⁹

BOM = Beginning Of Message (PTI=0)
COM = Continuation Of Message(PTI=0)
CPCS = Common Part Convergence Sublayer
CPCS-UU = CPCS User-to-User indication
CPI = Common Part Indicator
CRC = Cyclic Redundancy Check
EOM = End Of Message(PTI=1)
Length = Length of CPCS-SDU (Length=0 ->Abort function)
PAD = Padding (0) to 47 octets
PDU = Protocol Data Unit
Reserved = 32 bit alignment of CPCS-PDU trailer
PTI = Payload Type Indication
SDU = Service Data Unit

The following table summarizes the AAL services classes³⁰.

²⁹ Andrew S. Tanenbaum. *Computer Networks*, Prentice Hall, 1996

	Class A	Class B	Class C	Class D
Timing relation between source and destination	Required		Not required	
Bit rate	Constant	Variable		
Connection mode	Connection oriented			Connectionless
AAL protocol type	Type 1	Type 2	Type 3/4, 5	Type 3/4

Table 1 – Summary of AAL service classes

3.2.3 ATM Physical Interfaces

There are two physical interfaces that are important to functions operating in the ATM layer. One is **User-Network Interface (UNI)** and the other is **Network-Network Interface (NNI)**. Figure 10 illustrates the interpretation of the ATM standards adopted by ATM Forum with respect to UNI and NNI. The UNI is the connection to the network from the endpoint devices (Phone, PBX, workstation, and router) to an ATM switch (intermediate system). The NNI is the interface that exists between the two switches.

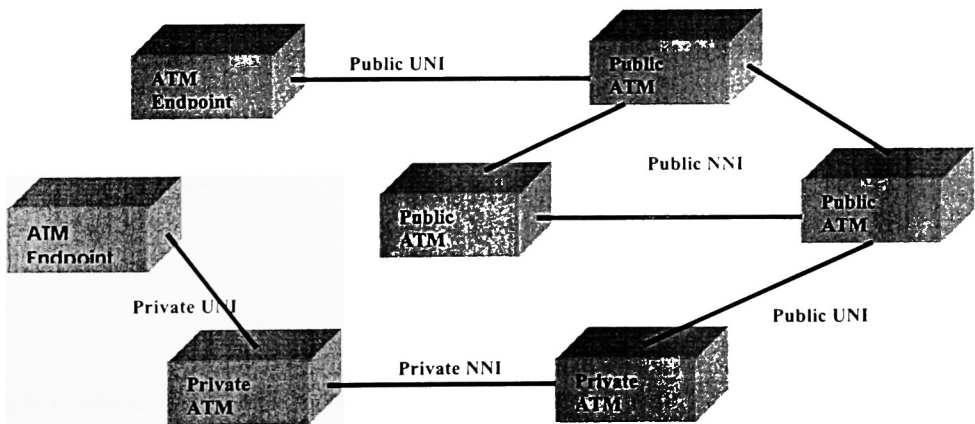


Figure 10. Public UNI and Private NNI Illustration

3.2.4 Cell Structure

The structure of the cell is important for the overall functionality of the ATM network. A large cell gives a better payload to overhead ratio, but at the expense of longer, more variable delays. Shorter packets overcome this problem, however the amount of information carried per packet is reduced. A compromise between these two conflicting requirements was reached, and a standard cell format chosen. The ATM cell consists of a 5-octet header and a 48-octet information field after the header. This is shown below.

¹⁰ William Stallings. *ISDN and Broadband ISDN with Frame Relay and ATM*. Prentice Hall, 1995

3.2.4.1 ATM Cell Header

The ATM layer provides the capability for switching the cells around the network based on the routing information, or label, in the header field. This label consists of a **Virtual Path Identifier (VPI)**, a **Virtual Channel Identifier (VCI)**, an error control field, and other cell management functions. Error detection at the ATM layer is confined to the header field. Figure 11 depicts an ATM cell with each component in the header field identified and briefly defined. Both the **User-Network Interface (UNI)** and the **Network-Network Interface (NNI)** are shown. The VPI occupies the entire first byte in the NNI, but only half the first byte in the UNI, which is the only difference.

Generic Flow Control (GFC) Field: A 4-bit field, which allows encoding of 16 states for flow control.
Routing Field (VPI/VCI): Sixteen bits are available for the VCI and 8 bits for the VPI, for a total of 24 routing bits. An additional 4 bits are available for the VPI in the NNI header.

Payload Type Field (PT): A 2-bit field is available to identify the payload type. It provides an indication of whether the cell payload consists of user information or network information.

Cell Loss Priority Field (CLP): If CLP is set (CLP value = 1) the cell is subject to discard, depending on network conditions. If not set, the cell has higher priority.

Header Error Control Field (HEC): Eight-bit field used for error management of the cell header.

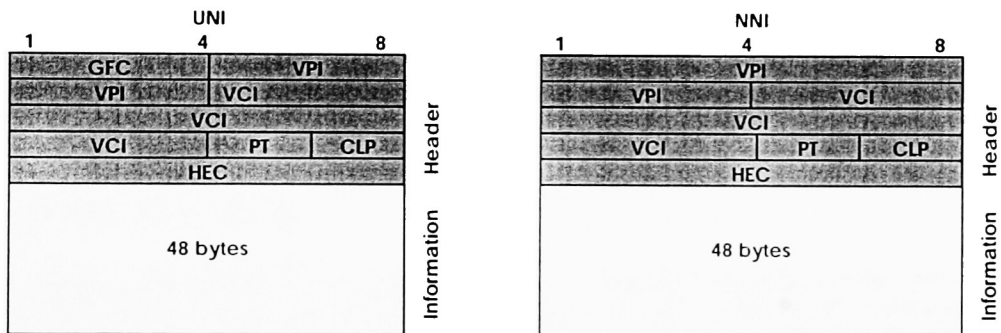


Figure 11: UNI and NNI frame formats

3.2.4.2 ATM Cell Header at User-to-Network Interface (UNI)

Routing: ATM is a connection-oriented mode. The header values (i.e. VCI and VPI etc.) are assigned during the connection set up phase and translated when switched from one section to other. Signaling information is carried on a separate virtual channel than the user information. In routing, there are two types of connections i.e. **Virtual Channel Connection (VCC)** and **Virtual Path Connection (VPC)**. A VPC is an aggregate of VCCs. Switching on cells is first done on the VPC and then on the VCC.

ATM Resources: ATM is connection-oriented and the establishment of the connections includes the allocation of a VCI and/or VPI and also includes the allocation of the required resources on the user access and inside the network. These resources, expressed in terms of throughput and quality of service, can be negotiated between user and network either before the call set up or during the call.

ATM Cell Identifiers: ATM cell identifiers, i.e. Virtual Path Identifier, Virtual Channel Identifier and **Payload Type Identifier (PT)** are used to recognize an ATM cell on a physical transmission medium. VPI and VCI are same for cells belonging to the same virtual connection on a shared transmission medium.

Throughput: **Peak Cell Rate (PCR)** can be defined as a Throughput parameter which in turn is defined as the inverse of the minimum inter arrival time T between two consecutive basic events and T is the peak emission interval of the ATM connection. PCR applies to both **Constant Bit Rate (CBR)** and **Variable Bit Rate (VBR)** services for ATM connections. It is an upper bound of the cell rate of an ATM connection and there is another parameter **Sustainable Cell Rate (SCR)** allows the ATM network to allocate resources more efficiently.

Quality Of Service: **Quality of Service (QOS)** parameters include cell loss, the delay and the delay variation incurred by the cells belonging to the connection in an ATM network. QOS parameters can be either specified explicitly by the user or implicitly associated with specific service requests. A limited number of specific QOS classes will be standardized in practice.

Usage Parameter Control: In ATM, excessive reservation of resources by one user affects traffic for other users. So the throughput must be policed at the user-network interface by a Usage Parameter Control function in the network to ensure that the negotiated connection parameters per VCC or VPC between network and subscriber is maintained by each other user. Traffic parameters describe the desired throughput and QOS in the contract. The traffic parameters are to be monitored in real time at the arrival of each cell. CCITT recommends a check of the peak cell rate (PCR) of the high priority cell flow (CLP = 0) and a check of the aggregate cell flow (CLP = 1), per virtual connection.

Flow Control: In order to control the flow of traffic on ATM connections from a terminal to the network, a **General Flow Control (GFC)** mechanism is proposed at the User to Network Interface (UNI). This function is supported by GFC field in the ATM cell header. Two sets of procedures are associated with the GFC field i.e. Uncontrolled Transmission which is for use in point-to-point configurations and Controlled Transmission which can be used in both point-to-point and shared medium configurations.

3.2.5 ATM Virtual Connection

The *ATM layer is responsible for transporting information across the network*. ATM uses virtual connections for information transport. The connections are deemed virtual because although the users can connect end-to-end, connection is only made when a cell needs to be sent. The connection is not dedicated to the use of one conversation. As mentioned earlier the two types of ATM connections are:

- ♣ The **Virtual Path (VP)**³¹
- ♣ The **Virtual Channel (VC)** or **Virtual Circuits (VC)**³²

VC is a general term that signifies a logical unidirectional connection between two endpoints. It is the properties of the VP and VC that allow cell multiplexing. Cell switching requires only the value of the **VP Identifier (VPI)** to be known.

3.2.5.1 Virtual Channels

The *connection between two endpoints is called a Virtual Channel Connection, VCC*. It is made up of a series of Virtual channel links that extend between VC switches. A **Virtual Channel Identifier (VCI)** identifies the VC. The value of the VCI will change as it enters a VC switch, due to routing translation tables. Within a virtual channel link the value of the VCI remains constant. The VCI (and VPI) are used in the switching environment to ensure that channels and paths are routed

³¹ Daniel Minoli, *Enterprise Networking*, Artech House, 1993

³² Daniel Minoli, *Enterprise Networking*, Artech House, 1993

correctly. They provide a means for the switch to distinguish between different types of connection. There are many types of virtual channel connections, these include:

- ⌚ User-to-user applications. Between customer equipment at each end of the connection.
- ⌚ User-to-network applications. Between customer equipment and network node.
- ⌚ Network-to-network applications. Between two network nodes and includes traffic management and routing.

3.2.5.2 Virtual channel connections have the following properties:

- ⌚ A VCC user is provided with a quality of service, QoS, specifying parameters such as **cell-loss ratio (CLR)** and **cell-delay variation (CDV)**.
- ⌚ VCCs can be switched or semi-permanent.
- ⌚ Cell sequence integrity is maintained within a VCC.
- ⌚ Traffic parameters can be negotiated, using the **Usage Parameter Control (UPC)**.

A detailed diagram showing the relationship between virtual channels and paths is shown below.

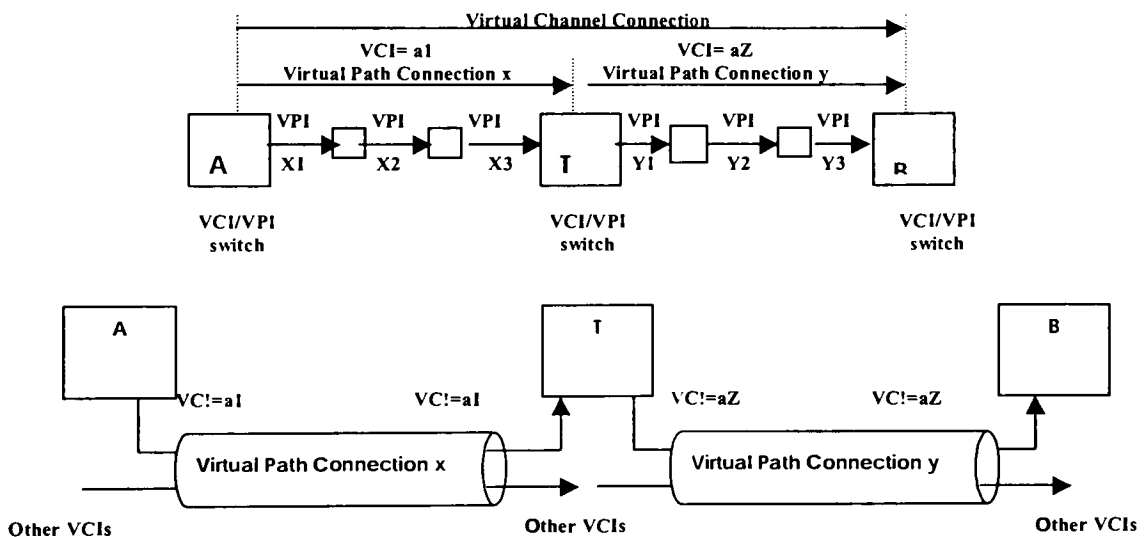


Figure 12 - Virtual Channel/Virtual Path Illustration

3.2.5.3 Virtual Paths

A virtual path, VP, is a term for a bundle of virtual channel links that all have the same endpoints. As with VCs, virtual path links can be strung together to form a virtual path connection, VPC. A VPC endpoint is where it's related VPIs are originated, terminated or translated.

Virtual paths are used to simplify the ATM addressing structure. VPs provide logical direct routes between switching nodes via intermediate cross-connect nodes. A virtual path provides the logical equivalent of a link between two switching nodes that are not necessarily directly connected on a physical link. It therefore allows a distinction between logical and physical network structure and provides the flexibility to rearrange the logical structure according to traffic requirements. This is best shown in the figure 12 above.

Virtual Path Switching

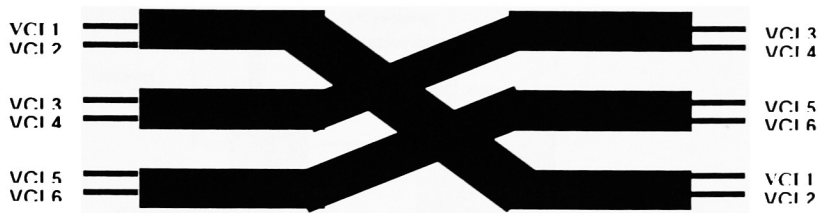


Figure 13. Virtual Path and Channel illustration

As with VCs, virtual paths are identified in the cell header with the VPI. Within an ATM cross-connect, information about individual virtual channels within a virtual path is not required, as all VCs within one path follow the same route as that path. A virtual channel may be described as a unidirectional pipe made up from the concatenation of a sequence of connection elements. A virtual path consists of a set of these channels.

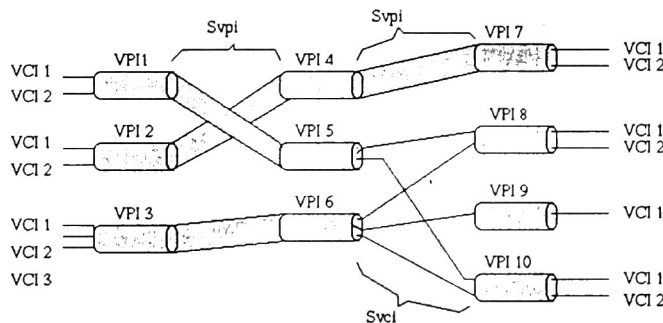


Figure 14. Virtual Path Switching³³

Each channel and path has an identifier associated with it. All channels within a single path must have distinct channel identifiers but may have the same channel identifier as channels in different virtual paths. Its virtual channel and virtual path number can therefore uniquely identify an individual channel. The virtual channel and path numbers of a connection may differ from source to destination if the connection is switched at some point within the network. Virtual channels, which remain within a single virtual path throughout the connection, will have identical virtual channel identifiers at both ends. Cell sequence is maintained through a virtual channel connection. Each virtual channel and virtual path has negotiated QOS associated with it. This parameter includes values for cell loss and cell delay.

³³ Byeong Gi Lee, Minho Kang, Jonghee Lee, *Broadband Telecommunications Technology*, Artech House, 1996

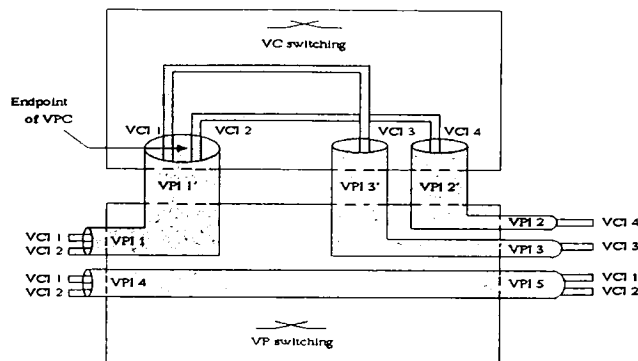


Figure 15. Virtual Path and Virtual Channel Switching³⁴

3.2.6 ATM Connection Setup³⁵

ATM connections are established as either **Permanent Virtual Circuits (PVC)** or **Switched Virtual Circuits (SVCs)**³⁶. As their name implies, PVCs are always present, whereas SVCs must be established each time a connection is set up.

To set up a connection, a signaling circuit is used first. A signaling circuit is a pre-defined circuit (with VPI = 0 and VCI = 5) that is used to transfer signaling messages, which are in turn used for making and releasing calls or connections. If a connection request is successful, a new set of VPI and VCI values are allocated on which the parties that set up the call can send and receive data.

Six message types are used to establish virtual circuits, each message occupying one or more cells and containing the message type, length, and parameters. The following table lists these message types.

- ⚡ **SETUP** Requests that a call be established Indicates an incoming call
- ⚡ **CALL PROCEEDING** Acknowledges the incoming call Indicates the call request will be attempted
- ⚡ **CONNECT** Indicates acceptance of the call Indicates the call was accepted
- ⚡ **CONNECT ACK** Acknowledges acceptance of the call Acknowledges making the call
- ⚡ **RELEASE** Requests that the call be terminated Terminates the call
- ⚡ **RELEASE ACK** Acknowledges releasing the call Acknowledges releasing the call

The sequence for establishing and releasing a call is as follows:

1. The host sends a **SETUP** message on the signaling circuit.
2. The network responds by sending a **CALL PROCEEDING** message to acknowledge receiving the request.
3. Along the route to the destination, each switch receiving the **SETUP** message acknowledges it by sending the **CALL PROCEEDING** message.
4. When the **SETUP** message reaches its final destination, the receiving host responds by sending the **CONNECT** message to accept the call.

³⁴ Byeong Gi Lee, Minho Kang, Jonghee Lee, *Broadband Telecommunications Technology*, Artech House, 1996

³⁵ Mahammed A Rahman, *ATM Systems and Technology*, Artech House, 1998

³⁶ Ulyess Black, *ATM: Foundation for Broadband Networks*, Prentice Hall, 1995

5. The network sends a **CONNECT ACK** message to acknowledge receiving the **CONNECT** message.
6. Along the route back to the sender, each switch that receives the **CONNECT** message acknowledges it by sending **CONNECT ACK**.
7. To terminate the call, a host (either the caller or the receiver) sends a **RELEASE** message, causing the message to propagate to the other end of the connection, and then releasing the circuit. Again, the message is acknowledged at each switch along the way.

3.2.6.1 Sending Data to Multiple Receivers

In ATM networks, users can set up *Point-to-Multipoint (P/MP)* calls, with one sender and multiple receivers. A P/MP VC allows an endpoint called the root node to exchange data with a set of remote endpoints called leaves. To set up a point-to-multipoint call, a connection to one of the destinations is set up in the usual way. Once the connection is established, users can send the **ADD PARTY** message to attach a second destination to the VC returned by the previous call. To add receivers, users can then send additional ADD PARTY messages.

This process is similar to a user dialing multiple parties to set up a telephone conference call. One difference is that an *ATM P/MP call doesn't allow data to be sent by parties towards the root (or the originator of the call). This is because the ATM Forum Standard UNI 3.1 restricts data flow on P/MP VCs to be from the root towards the leaves only. But UNI 4.0 rectifies this limitation.*

3.2.6.2 ATM Switching³⁷

An ATM switch transports cells from the incoming links to the outgoing links, using information from the cell header and information stored at the switching node by the connection setup procedure. The connection setup procedure performs the following tasks:

- z_o Defines a unique connection identifier (a VPI and VCI) for each connection at the incoming link and link identifier, and a unique connection identifier at the outgoing link.
- z_o Sets up routing tables at each switching node to provide an association between the incoming and outgoing links for each connection.

As mentioned previously, VPI and VCI are the connection identifiers used in ATM cells. To uniquely identify each connection, VPIs are uniquely defined at each link, and VCIs are uniquely defined at each VP. To establish an end-to-end connection, a path from source to destination has to be determined first. Once the path has been established, the sequence of links to be used for the connection and their identifiers are known.

VPIs are used to reduce the processing of an ATM switch by routing on the VPI field only. For example, VPI routing is useful when many VCs share a common physical route (similar to all phone connections between Los Angeles and Washington D.C.). Figure below provides an example of VPI routing.

Switch A				Switch B			
Input Port	Incoming VPI/VCI	Output Port	Outgoing VPI/VCI	Input Port	Incoming VPI/VCI	Output Port	Outgoing VPI/VCI
4	3/16	2	9/16	4	9/16	3	7/16

Switch C			
Input Port	Incoming VPI/VCI	Output Port	Outgoing VPI/VCI
4	7/16	2	20/16

³⁷ **ATM cell switching technologies**, <http://www.sims.berkeley.edu/resources/infoecon/FAQs>

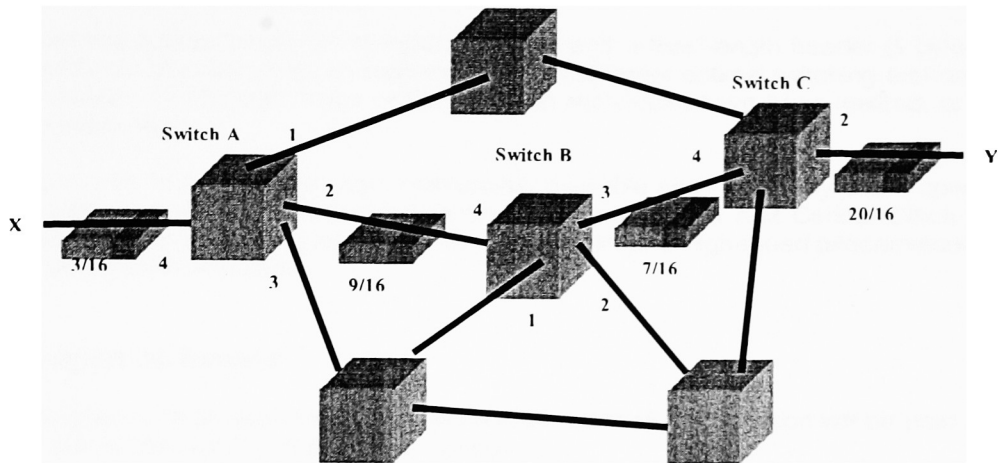


Figure 16. An Example of VPI routing

A common unique identifier referred to as a virtual circuit identifier (VCI) associates ATM cells transported on a particular VC. When a network connection is established, a route from the source computer to the destination computer is selected as part of the connection setup, and that route is used for all traffic flowing over the connection. An example of an end-to-end connection between two users, using only the VCI field of the ATM header, is presented next.

Suppose a virtual circuit consisting of a four-hop path is selected between two users, X and Y, by a routing algorithm. After the network finds a path between the two nodes, it assigns the VCI values to be used at each node along the path and sets up routing tables at nodes N1 to N5. When the transmission starts, all cells of the connection follow the same path in the network. Once the communication is completed, one of the two end users releases the connection, and the VC is also terminated. The end-to-end connection defined by the concatenation of VC links is called a **Virtual Circuit Connection (VCC)**.

Virtual path (VP) is a generic term for a bundle of virtual circuit connections, all having the same VPI value and terminating at the same pair of endpoints. In ATM cells, a virtual path identifier (VPI) is assigned for each VP. When a VP is routed, all the VCs belonging to that VP are routed together. The term **Virtual Path Connection (VPC)** is used to refer to a sequential collection of VPs-VPC defines a route between the source and destination nodes.

3.3 Switching Architectures³⁸

Various switching architectures were developed in the past for different application such as voice and data, based on modes like **STM (Synchronous Transfer Mode)**³⁹ and packet switching. The switching architectures that were previously developed for STM and for conventional packet switching like X.25 are not directly applicable for broadband ATM. Three major factors have a large impact on the implementation of the ATM switching architecture⁴⁰:

- ⌚ The high speed at which the switch has to operate (from 150Mbit/sec up to 600Mbit/sec).
- ⌚ The statistical behavior of the ATM stream passing through the ATM switching system.

³⁸ R Y Awdeh & H T Mouftah, *Survey of ATM Switching Architectures*, Computer Networks Vol. 27, 1995

³⁹ Thomas M. Chen, Stephen S. Liu, *ATM Switching Systems*, Artech House, 1995

⁴⁰ Byeong Gi Lee, Minho Kang, Jonghee Lee, *Broadband Telecommunications Technology*, Artech House, 1996

- ⚡ ATM is connection oriented. Therefore, the switching elements have pre-defined routing tables to minimize the complexity of single switch routing.

The ATM cell has a fixed length (of 48 bytes) payload and a fixed length header (5 bytes) with limited header functionality allows to implementation of different optimal switching architectures, queuing functions for example. Some of the switching techniques have been realized, or are in stage of implementation.

A growing number of ATM switches are commercially available and installed by public operations to offer a public, wide area broadband service, sometimes called ATM Central Office. Other switches are deployed by private users and are used in an internal high-speed telecommunication networks, often called ATM LAN.

3.3.1 Transport vs. Control

In the description of ATM switching in the following introduction the attention will be paid to the "transport" part of the switch and not to the "control" part.

The transport network is defined as all the physical means, which are responsible to the current transportation of the information from the ATM inlet to the ATM outlet. The transport network in the ATM network mainly performs functions located in the user plan of the ATM protocol reference model.

The control part of the switch is that which controls the transport network. It decides for instance, which inlet to connect to which outlet. The decision is based on incoming signaling information. The control network mainly performs functions located in the control plane of the ATM protocol reference model. The quality of service parameters for the transport network is the cell loss rate, the error rate, cell delay and cell jitter.

3.3.2 ATM Switch Functions⁴¹

ATM is connection oriented. All cells belong to a virtual connection pre-established by the transport network. All traffic is segmented into cells for transmission across the network. The sequence integrity of all the cells in the virtual connection is preserved across each ATM switch to simplify reconstruction of the original traffic at the destination (allows smaller total delay on the net). The ATM cell is 53 bytes long, built of 48 payload bytes and a 5-byte header. Each cell's header contains a VCI (virtual channel identifier) that identifies the virtual connection to which the cell belongs.

The ATM switch has several main tasks:

3.3.2.1 VCI translation.

The established connection on the ATM network defines the virtual path through different switches across the network. The VCI is local to each switch port. As each cell travels across an ATM switch, the VCI is translated into a new value. The switch has to build the new cell header containing the new VCI (and possibly new VPI - virtual path identifier) and calculate the new HEC value.

⁴¹ Uyles Black, *ATM: Foundation for Broadband Networks*, Prentice Hall, 1995

3.3.2.2 Switching - Cell transport from its input to its output.

The transportation of the information (cell) from an incoming logical ATM channel (inlet) to an outgoing logical ATM channel (outlet) is also the responsibility of the ATM switch. The logical ATM channel is characterized by two identifiers:

1. The physical inlet/outlet which is characterized by a physical port number.
2. The logical channel on the physical port which is identified by the VCI and/or the VPI.

In order to provide the switching function, both physical and logical identifiers of the incoming cell have to be related to physical and logical identifiers of the outgoing cell. Two functions have to be implemented in the ATM switching system.

The first function is the space switching function. The space switching function is the one, which allows the connection between every input and every output. An important aspect of space switching is the internal routing. This means how the information is routed internally in the switch. The internal structure of the switch must allow connections between every input to every output.

The second function is time switching. Since ATM is working in an asynchronous mode, cells which had arrived in various time slots from the different inputs can be delivered from different outputs in different time slots (there is no time identifier in ATM as it is in STM). Since there is no pre-assigned time slot connection, a contention problem arises if more than two logical channels are connected to the same output at the same time slot. Implementing a queuing function in the ATM switch system solves this problem in the ATM switch. In this work the functions of routing and queuing will be discussed in more detail.

3.3.3 Queuing Methods

Having established the need for smoothing queues in packet switches, it remains to be seen how these queues can be incorporated into the switch. The two basic techniques are:

- Input Queuing
- Output Queuing

3.3.3.1 Input Queuing

Input queuing is illustrated in the figure below.

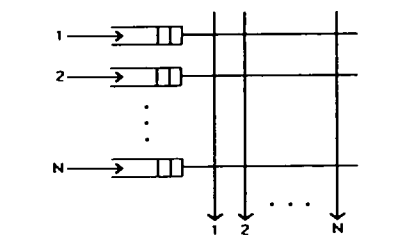


Figure 17 - Architecture of an input-queued packet switch⁴²

Cells arriving on each input line are placed into smoothing buffers, prior to placement on the correct output lines. In any given time slot, leading cells in all the buffers are sent to their output lines. If several buffer-leading cells contend for the same output, only one of them is selected

⁴² Byeong Gi Lee, Minho Kang, Jonghee Lee, *Broadband Telecommunications Technology*, Artech House, 1996

according to a contention scheme (a plausible scheme may be round-robin), while the rest remain in the buffers and contend again in the very next time slot.

An input-queued $N \times N$ switch requires N Packet routing elements and N^2 Contention elements, which usually can all be synthesized on one VLSI chip. Each of the buffers, on the other hand, normally requires a separate chip, since RAM technology normally allows the contents of only one memory location to be read or written at a time. Therefore, the switch complexity, measured as the number of chips required for implementation, grows linearly with N .

The input queuing technique suffers from an unfortunate phenomenon known as head-of-queue blocking. The phenomenon occurs when a cell in any given queue is denied access to its output line, even though there are no other cells requiring the same output line, simply because the cell in the head of that queue was blocked in contention for a totally different output. If the leading cell output line is under heavy contention (this may be the case when one LAN provides some kind of service required by all other LANs; then, the server LAN output line will normally be contended for more often than others), other cells in that queue might have to wait for a relatively long period. In fact, the delay for a given cell may grow unbounded even for an offered load less than 100 percent. A simple calculation shows that the maximum offered load for which the average cell delay time remains bounded is only 63 percent if the cells in each queue are served on a random basis. Paradoxically perhaps, if the cells in the queue are kept in first-in first-out rather than in random order, this figure drops further to only about 58 percent.

The input-queued switch has some trouble in supporting cells intended for more than one output, i.e. broadcasting and multicasting. If a leading cell is of this kind, it has to contend simultaneously for all the outputs it is intended for; head-of-queue blocking can be aggravated many times if the contention schemes are independent. Alternatively, the round-robin contention policy can be implemented in such a way that the rotating pointer is the same for all outputs at any given time; such an implementation guarantees that a cell does not stay in the queue more time just because it is a multicast cell, but it leads to an even larger delay for the more normal single-cast cells.

3.3.3.2 Output Queuing

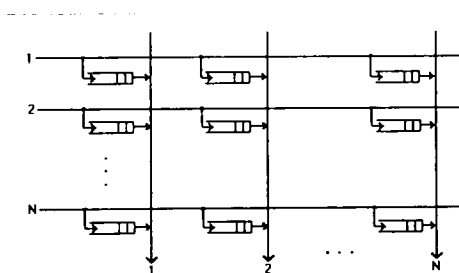


Figure 18 - Output queuing diagram⁴³

There are N^2 buffers, one for each input-output combination. For every time slot, all cells active during that time slot enter the queue in the intersection of their input line and output line. The leading cells in all the buffers with a common output contend for that output; a plausible contention scheme may be, for example, round-robin.

⁴³ *Byeong Gi Lee, Minho Kang, Jonghee Lee, Broadband Telecommunications Technology, Artech House, 1996*

Because the routing of a cell is done before its queuing, the head-of-queue blocking effect, described in the section on input queuing, cannot occur: a cell cannot be denied access to its output by the presence of cells headed for other outputs, but rather only by other cells contending for that same output. Hence, the switch throughput is, in theory, as high as 100%; in other words, even with the percentage of occupied time slots approaching unity, the mean delay time for any cell remains finite (assuming infinite buffers).

The price paid for solving the head-of-queue blocking problem is evident: assuming a chip is required for every buffer, the switch complexity is now quadratic in N , rather than linear as was the case for input queuing.

In passing, let us note that the output-queued switch readily supports broadcasting and multicasting; the only difference is that an arriving cell is copied to buffers corresponding to all the outputs it is intended for. The price that has to be paid to support multicasting is the need to keep a conversion table from a multicast address to the corresponding outputs and the additional logic to apply it, but then again, such a table cannot be spared in any implementation of multicasting.

3.3.4 General Structure of an ATM switch

The ATM switch has to handle a minimum of several hundred thousand cells in a second at every switch port. A switch has to connect from a few ports to thousands of ports. In principle, a switch fabric can be implemented by a single switching element. But from practical reasons the switch fabric has to be built of basic switching building blocks - switching elements.

A switching element is the basic unit of the switch fabric. It can be implemented in a single integrated circuit element. At the **input port (inlet)** the routing information of the incoming cell is analyzed and the cell is then directed to the correct **output port (outlet)**. In general the switching element consists of an interconnection network, and **IC (input controller)** for each incoming line and an **OC (output controller)** for each outgoing line. The IC will synchronize arriving cells to the internal clock. The OC transport cells, which have been received from the interconnection network toward the destination. The interconnection network couples the IC and OC.

3.3.5 Switching Element Performance Requirements ⁴⁴

An ATM network has to support a wide range of applications using various kinds of information in, a wide range of speeds from telecontrol to high quality video. These services define different requirements in terms of bit rate, behavior in time (constant bit rate or variable bit rate), semantic transparency (cell loss rate, bit error rate) and time transmission (over all delay, jitter). The ATM switch architectures have to be considered in these requirements.

Connection Blocking

Since ATM is defined to be connection oriented, after connection set-up a logical connection must be found between the logical inlet and the logical outlet. Connection blocking is defined as the probability that not enough resources can be found to allow all the required physical connections between inlets and outlets at any time.

Cell Loss, Cell Insertion

In an ATM switch it is possible that temporarily too many cells in the switch have to be transmitted through the same link (switch internal or external link). In optimal operational conditions there is an available entry in a queue to hold all the cells. But if the queue is currently full, another cell that will require the same queue will be lost.

⁴⁴ Raif. O Onvural, *Asynchronous Transfer Mode Networks - Performance Issues*, Artech House, 1995

The probability of a cell lost must be kept in specified limits to assure high semantic transparency. Typical values for cell loss in ATM are in the range of 10^{-8} to 10^{-10} . Some switching architectures are designed such that they will not suffer from cells competing for the same resources internally, but only at their inlets and/or outlets.

It is also possible that from some internal routing error a cell will be sent to the wrong logical connection. If such an error occurs, error impact is doubled by the fact that one destination will miss a cell and that a second destination will accept an additional cell. The switch element has to be designed so that cell insertion error probability will be about 1000 times better than a cell loss.

Switching Delay

To allow support of different real time services in an ATM network, a maximal delay has to be guaranteed and a low values of jitter.

Typical delay values are between 10 and 1000 usec, with jitter of 100 nsec or less. The delay and the jitter in the cell are strongly related to the queuing in the switching element. A small queue will assure better delays but will increase the cell loss probability.

Information Rates

A large number of information rates have to be switched in the same ATM switch. The maximal bit rate, which a future ATM switch has to be able to switch, lies around 150 Mbps. For such fast services, the switching element can be implemented as several switching elements in parallel. Or, several 150Mbit/sec switching elements can be multiplexed on a single link. That will require a switching rate in the order of Gbps.

Broadcast

In classical connection oriented packet switching services, only point to point connections are available, because the information (cell) can be switched from one logical inlet to one logical outlet only. In future broadband networks broadcast and multicast services are required for different applications from electronic mail to network TV services.

Queuing Methods

There are many queuing problems in an ATM switch because actually the ATM switch performs statistical multiplexing in the switch inputs and de-multiplexing in the switch outputs. Suppose two ATM cells arrived at two inlets at the same time and are aiming for the same outlet. Some arbitration mechanism and queue of waiting cells has to be implemented in the switch. There are several queuing possibilities. It is possible to add a queue at the switch element inputs, add a queue at the switch output, or add a queue between the inputs on the outputs of the switch.

The different possibilities are described briefly.

Input Buffers

In this configuration the buffers are located at the input controller (IC). When using a **first-in-first-out (FIFO)** buffer, a collision occurs if two or more head-of-the-queue cells compete simultaneously for the same output. Then all but one of the cells is blocked. The cells behind the blocking head-of-the-queue cell are also blocked even if they are destined for another available (currently not in use) output. In this method the switch interconnection network will transfer the cell from the input buffer to the output buffer without internal conditions. Arbitration logic is needed to determine which of the cells held in different inlet buffers destined to the same output will be transferred in the interconnection network. The arbitration logic can be from very simple logic (e.g. simple round robin) to more complex arbitration methods (aiming to keep the same queue length in all the buffers).

To overcome this disadvantage, the FIFO buffer can be replaced by a random access memory (RAM). If the first cell in the queue is blocked, the next cell, which is destined for an idle output (or internal switch interconnection network link), will be selected for transmission. The disadvantage of this solution is that a complex buffering control is required to find a cell destined to an idle connection and also to guarantee a correct cell sequence of cells destined for the same output.

The input buffer approach achieves the worst performance in the sense of the queue length required to achieve a given cell-loss rate in various switch loads in comparison to the other two queuing methods.

Output Buffers

In this technique, the buffers are located at the OC of the switch element. The assumption is that many cells from the IC can cross the internal interconnection network and arrive to the outlets. This solution requires use of a very fast internal pass. In order to allow a non-blocking switch, the interconnection network and the output buffer have to be capable of handling N cells at one cell time (when N is the number of ICs). When output buffers are in use, no arbitration has to be used. The control of the output is based on a simple FIFO logic.

Central Queuing

In the central queuing approach, the queuing buffers are not dedicated to a single inlet (as in the input buffer approach) or to a single outlet (as in the output buffer approach), but shared between all inlets and outlets. Each coming cell will be directly stored in the central storing element. Every outlet will identify the cells destined to it in a FIFO discipline.

From the queuing point of view, this method is the most efficient and required the smallest total storage to allow minimal cell loss in heavy load conditions. Since the available memory on an integrated circuit switching element is limited, it is possible to achieve low cell-loss probabilities when using the central queuing approach. The disadvantages of this approach are that very fast memory elements are required to allow all the coming cells and outgoing cells access to the memory ports at the same time, and big complexity in the queue management.

Technical Aspects

The realization of the three queuing approaches (input buffer, output buffer, central queuing) are very different. The differences are in three main aspects:

Queue size

The size of the queue depends on the performance requirements of the system (cell loss ratio, load, delay) and the queuing method in use. The queue size is reflected in the number of cell buffers, which are supported by the switching element.

Memory speed

The access time of the queuing element depends on the queuing method in use, the number of inlets and the number of outlets of the switching element and the rate of the incoming and outgoing cells.

Memory control

In order to control the queues of the switching element, additional control logic is required.

3.3.6 Switch Architectures⁴⁵

Let us restate once more the functions required from a packet switch:

1. Routing of the applied input packets to the appropriate outputs
2. Resolving output port contention by buffering.

As we have shown in the section about queuing methods, one way to construct a packet switch is by separating these two functions altogether. Thus, a switch would consist of a routing network, devoid of buffers, which would only have to be able to provide a connection between every input and every output, and a set of buffers either at the input or the output of that routing network.

Recalling, the advantages and disadvantages of a solution of this kind were:

- z_o For input queuing⁴⁶:
 1. Linear complexity (number of required buffers proportional to N);
 2. Low permitted input load (only 63% for a large switch)
 3. Difficulty supporting multicasting.
- z_o For output queuing⁴⁷:
 1. Full link utilization (100% permitted input load);
 2. Easy support of multicasting and broadcasting;
 3. Quadratic complexity (number of required buffers proportional to N²)

As for the routing network itself, it can be proved that the minimum number of beta-elements required for a rearrangeable, non blocking network (i.e. a network that can concurrently route any permutation of paths from distinct inputs to distinct outputs) is $N \log N$.

We shall define a network as self-routing if the state of each & beta-element can be deduced only from the information in the cells at its inputs. The minimum rearrangeably nonblocking network (with $N \log N$ elements), also known as a Benes network, is not self-routing; rather, a sophisticated controller is required to set the states of all the switching elements correctly. An implementation of a rearrangeably nonblocking self-routing network requires, therefore, even more switching elements.

The switch architectures presented below attempt to trade between link utilization, packet loss rate, and design complexity.

- z_o The Banyan Switch
- z_o The Knockout Switch
- z_o The Tandem Banyan Switch
- z_o The Shared Memory Switch

3.3.6.1 The Banyan Switch

The Banyan switch is a multistage self-routing architecture which uses fewer & beta-elements than the minimum number required for a rearrangeably nonblocking design. More specifically, a $N \times N$ Banyan switch uses $(N/2) \log N$ elements. Consequently, the switch cannot be nonblocking; input-to-output permutations can be constructed that cannot be concurrently routed with the switch. Therefore, smoothing buffers must lie inside the switch to achieve a reasonably low packet loss rate.

⁴⁵ ATM Switching, <http://cne.gmu.edu/~sreddiva/ATMswitch.html>

⁴⁶ Thomas M. Chen, Stephen S. Liu, *ATM Switching Systems*, Artech House, 1995

⁴⁷ Thomas M. Chen, Stephen S. Liu, *ATM Switching Systems*, Artech House, 1995

The structure of an 8x8 Banyan switch is depicted in the figure below

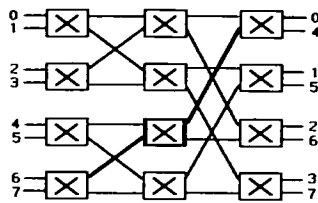


Figure 19 - An 8x8 Banyan switch⁴⁸

We see that the β-elements are arranged in three columns of four elements each, in a pattern that resembles a grid of butterflies. The inputs to the switch are the inputs to the elements in the first column, and the outputs of the last column are the outputs from the switch. In each β-element, one output is connected to the input of the element just horizontally on its right, and the other goes to an element whose line number, represented in binary, differs in precisely the j 's bit, where j is the column number of the element (counting from 0). For example, the outputs of element (2,1) (bold in the diagram) are connected to the inputs of elements (2,2) (horizontal connection) and (0,2) (diagonal connection), as the numbers 2 and 0 differ in bit # 1 of their binary representation. This simple rule also tells how to construct a path from any input to any output: in each column j , an appropriate β-element should be set in the "bar" state if the j 's bits of the input and the output numbers equal, and in the "cross" state if those bits differ. The path shown in bold in the diagram illustrates how to connect input 7 to output 0. Since all the bits in the binary representations of the input and the output differ, all elements along the path are set to "cross". Note that every such path is unique. Obviously, several paths cannot be routed concurrently unless they happen to require the same states of the β-elements. Thus, in our case, once input 7 is connected to output 0, input 6 cannot be connected to outputs 2, 4, and 6, because any of these connections would require the element in the first column to be set to "bar".

Several remedies can be employed to attempt resolving this type of routing conflict: (1) Provide buffers within the beta-elements, so that cells that cannot be immediately delivered are stored and their routing deferred according to some contention resolution policy; (2) Run the internal links at a rate that is a multiple of the cell arrival rate, sequentially establishing several paths within the duration of one cell. To provide an insight to how good these techniques can be in reducing packet loss rate, it suffices to quote the results of a computer simulation for a large (1024x1024) Banyan switch run at full input load. With the internal links running at twice the cell rate (hence capable of establishing two subsequent paths within one time slot) and a buffer size of 5 cells in each β-element, as many as 92% of the input cells were delivered, compared to about 25% for a simple unbuffered switch, and about 75% for a double-rate unbuffered switch. Still, to achieve reasonable packet loss rates (such as one packet per million), the input load would have to be reduced considerably.

3.3.6.2 The Knockout Switch

The knockout switch is a fully connected architecture which attempts to combine the implementation simplicity of input queuing (buffer complexity is linear in the number of ports) with the throughput performance of output queuing (permitted input load and saturation throughput both approaching 100%). The knockout switch architecture achieves this goal by intentionally introducing a new source of packet loss, known as buffer blocking, in addition to packet loss mechanisms present in any switch architecture, namely buffer overflow and noise-induced random channel errors. The rate of loss from buffer blocking can be readily controlled and kept low, to reduce significantly the complexity of a switch based in principle on the output queuing idea. The knockout switch architecture is explained in the following set of diagrams.

⁴⁸ R Y Awdeh & H T Moufiah. *Survey of ATM Switching Architectures*. Computer Networks Vol. 27. 1995

As in the output queuing model, each fixed-length cell arriving at one of the input ports is placed on a broadcast bus from which each of the output modules taps the cells intended for it. It is obvious that multicast and broadcast cells are readily supported. The output module acts as statistical multiplexer, deferring cells that cannot be immediately placed onto the output link because of contention.

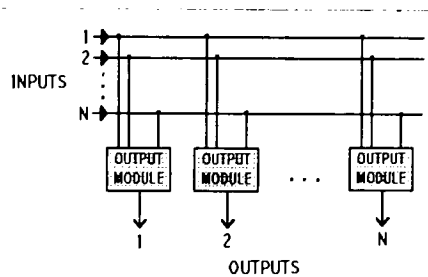


Figure 20 - Knockout Switch overall architecture⁴⁹

Each input to an output module, receives the fixed-length cells broadcasted on the corresponding input bus. The job of each packet filter is simply to pass the cell to the concentrator if the cell is destined for that output, and to mark the cell as inactive otherwise. Such a filter can be easily implemented by a β-element, only one input and one output of which is used. The role of the concentrator is to identify among its inputs those cells that are active and route them to its leftmost outputs, one cell per output line. Note that the concentrator has only $L < N$ outputs. Should $L+1$ or more cells arrive simultaneously, only L of them will be processed via the concentrator; all others will be lost. This is the extra packet loss source in the knockout switch. By properly choosing L , the loss rate induced by the concentrator can be controlled and maintained at a reasonably low level.

Furthermore, the value of L required to maintain a given loss rate is relatively small, independent of the number of inputs when the latter is large, and grows only logarithmically in the loss rate. For example, $L = B$ is sufficient to maintain the packet loss rate in the concentrator at one packet per million, for large N and full input load, and it only grows by one per every order of magnitude reduced in the loss rate (i.e. $L = 11$ is enough for a loss rate of one packet per billion). This effect is the key to maintaining linear complexity of the knockout switch, as the number of buffers is proportional to $L \times N$ rather than N^2 .

The concentrator inputs receive cells which have already been passed by the packet filters and are known to be intended for the switch output port served by the concentrator. There are four (generally, L) stages in the concentrator shown in the diagram. Each stage is designed to operate like an elimination tournament. Specifically, each β-element is programmed to set itself to the "bar" state if there is an active cell on its left input, and to "cross" otherwise. Whenever there is only one active cell at the inputs of a β-element, it is allowed to pass downward. If both cells are active, the right-hand one is "knocked out" to the next stage and contends there. Each stage produces one "winner" among the active cells that enter it, and each subsequent stage receives one less active cell than the previous one. Therefore, when there are k active cells, they are guaranteed to come out on the outputs of the first k stages.

If the packet buffers in the output module diagram were to be loaded directly from the concentrator outputs, then the leftmost buffers would tend to fill up faster, and might even overflow despite the presence of empty buffer entries on the right. The shifter prevents that from happening by spreading each bulk of cells arriving at its input continuously to the right; in other

⁴⁹ Byeong Gi Lee, Minho Kang, Jonghee Lee. *Broadband Telecommunications Technology*, Artech House, 1996

words, if the last buffer to receive a packet happened to be m , then the next k cells arriving at the shifter's input will be directed to buffers $m+1, m+k$ (modulo L). Physically, the shifter can be implemented with a $L \times L$ Banyan network.

Because of the round-robin nature of the shifter and the fact that the buffers are filled cyclically, they can also be emptied cyclically. At each time slot, the output line fetches a cell from a buffer just right (cyclically) of the buffer last fetched from, beginning with buffer 1. Moreover, if the output circuitry encounters an empty buffer, the round-robin policy of buffer filling guarantees that all buffers are empty at that point, and the one just reached is precisely the next one to be filled again. The output pointer can then just stop there and wait for that buffer to receive a cell, after which the circular emptying of buffers can restart from that point.

3.3.6.3 The Tandem Banyan Switch

A block diagram of the Tandem Banyan switch is presented.

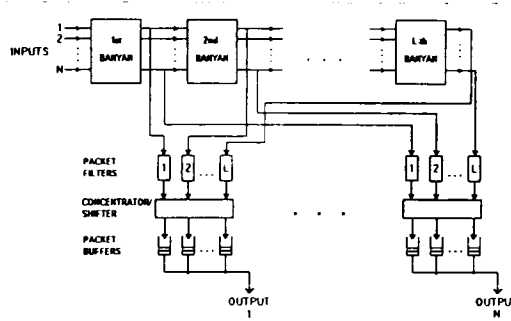


Figure 21 - Tandem Banyan Switch block diagram⁵⁰

A brief glance at the diagram reveals that the only difference between the Tandem Banyan switch and the knockout switch is the replacement of the N input broadcast lines by a cascade of L Banyans. Consequently, there are only L packet filters per each output. Otherwise, the switches are identical. The jobs of the concentrator, the shifter, and the shared buffers remain the same as for a knockout switch, and the reader is referred to that section for details. In particular, if more than L cells simultaneously arrive for the same output, all except L of them are lost.

The Tandem Banyan switch does not use a knockout concentrator to find those inputs intended for a given output. Instead, it uses a cascade of memoryless Banyan routing networks. The first Banyan does the best it can to route its inputs to its correct outputs. However, since the path from every input to any output inside the Banyan is unique, contention may arise if two inputs either request the same output or need to use the same internal link. In both cases, one of the cells is routed correctly, while the other one is misrouted and appears on a wrong line at the output. In order to minimize the damage caused by misrouted cells, any cell that has been misrouted once (and therefore cannot reach its correct output within the Banyan) is marked so as not to contend for internal links in later stages of the Banyan, which can be needed for correctly routed cells. The first packet filter of that output module intercepts every cell that has reached its correct output by the end of the first Banyan. Only the cells misrouted by the first Banyan enter the second one, and so on; each subsequent Banyan receives only cells misrouted by previous Banyans in the cascade. Cells still misrouted after L stages are irrecoverably lost.

⁵⁰ Uyles Black, *ATM: Foundation for Broadband Networks*, Prentice Hall, 1995

In passing, note that the Tandem Banyan architecture cannot support multicasting as easily as the knockout switch does. The Banyans have to know a single destination for each input cell, and no packet filter can just intercept a cell appearing on the correct line, because that cell may have to be read by other outputs' packet filters as well.

Contrary to the knockout switch, in the Tandem Banyan, even if not more than L cells appear concurrently for a given output, there is no guarantee that they will all enter that output's concentrator. As a consequence, the value of L required to maintain a given packet loss rate resulting from buffer blocking is generally larger than for the knockout. Moreover, it does not become independent for large N , as was the case there. Specifically, given an input load of 100% and $N = 1024$, $L = 15$ stages are required as opposed to only $L = 8$ in the knockout switch, and it grows further (approximately logarithmically) for larger switches.

The major advantage of the Tandem Banyan architecture over the knockout is in the number of β -elements required in its implementation. In the knockout switch, there are N output modules, each containing N packet filters and a concentrator whose implementation requires an order of LN β -elements, for a total of (LN^2) switching elements. In the Tandem Banyan, an output module contains only L packet filters, and its concentrator, being of lower dimensions, needs only about L^2 β -elements, for a total of (L^2N) . As for the Banyan packet routing networks - there are L of them, each having $(N/2) \log N$ switching elements, for a total of $(LN \log N)$. Since, as previously mentioned, L itself grows approximately as a logarithm of N , we get the Tandem Banyan switch complexity to be $(LN \log N) < (LN^2)$. For very large N , the Tandem Banyan switch therefore requires a substantially smaller amount of β -elements.

3.3.6.4 The Shared Memory Switch

The distinctive feature of a $N \times N$ shared memory switch is its use of a high-speed internal bus, with a bit rate N times as large as the rate on each individual input/output line. For a time slot of length F , the internal bus is capable of transferring a cell in a mini-slot of length F/N . All cells received during a time slot are transferred to the shared memory, albeit with a very short delay (probably during the very next time slot, if double-buffering technique is used within the serial-to-parallel converter). Conversion from serial to parallel is required to maintain acceptable clock rates; the bus clock rate only needs to be N/W times higher than the incoming bit rate, with W the bus (and memory) width. The shared memory switch architecture is described in the diagram.

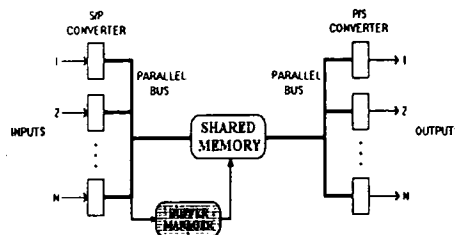


Figure 22 - Shared Memory Switch block diagram⁵¹

In the shared memory, the cells intended for each output is kept in separate partitions. During an output cycle, the cells are discharged to their outputs, again with only a minislot of length F/N being required on the output bus to discharge a single cell. Thus in a full time slot one cell is read from the shared memory for each output. If the memory partition for an output is empty, the corresponding minislot remains unfilled.

It is a matter of implementation whether the partitions in the shared memory are of rigid or flexible sizes. Obviously, flexible-size partitions require more sophisticated hardware to manage, but the cell loss rate performance is improved, because a memory partition does not suffer from overflow until no free memory remains at all; outputs idle at a given time can "lend" some memory to other outputs that happen to be heavily used at the moment.

⁵¹ Byeong Gi Lee, Minho Kang, Jonghee Lee, *Broadband Telecommunications Technology*, Artech House, 1996

The design simplicity appeal of the shared memory architecture makes it popular for small-sized fast switches, particularly for interconnecting a small number of LANs. However, for even moderately sized switches, the clock rate required of the internal bus becomes intolerable. As an example, at a cell arrival rate of 155 Mbps, $N = 32$ and $W = 16$, the internal bus has to operate under a clock rate of 310 MHz.

3.4. ATM Signaling and Addressing⁵²

The current and planned ATM signaling protocols and their associated ATM addressing models are reviewed herein. *ATM signaling protocols vary by the type of ATM link. ATM UNI signaling is used between an ATM end-system and an ATM switch across an ATM UNI; ATM NNI signaling is used across NNI links.* Standard do exist for ATM UNI and NNI signaling, and with progressive work they have gone through few phases of evolution. The current standard for ATM UNI signaling is described in the ATM Forum UNI 4.0 specification, which has added new features to the earlier UNI 3.1 specification. UNI signaling requests are carried across the UNI in a well-known default connection: VPI=0, VCI=5.

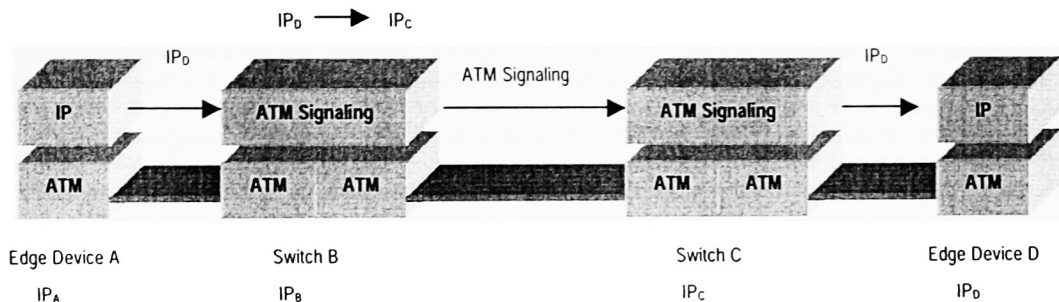


Figure 23 - Peer Model of ATM Addressing

The UNI 3.1 specification⁵³ is based upon Q 2931 (Q series standards⁵⁴), a public network signaling protocol developed by the International Telecommunications Union-Telecommunications Sector (ITU-T), which, in turn, was based upon the Q.931 signaling protocol used with **Narrowband ISDN (N-ISDN)**. The ATM signaling protocols run on top of a **Service Specific Convergence Protocol (SSCOP)**, defined by the ITU-T Recommendations Q.2100, Q.2110, and Q.2130. This is a data link protocol that guarantees delivery through the use of windows and retransmissions.

Note that in general, ATM does not offer an assured service. Cells are not retransmitted by ATM devices upon loss, for instance, since it is assumed that higher layers (such as TCP) will handle reliable delivery, if this is what the application requires. This also makes ATM devices much simpler, faster, and cheaper. ATM signaling requires the assured delivery guarantees of SSCOP since it does not run on any standard higher layer protocol like **Transmission Control Protocol (TCP)**, and the signaling state machines can be made much simpler if assured delivery can be assumed.

ATM signaling uses the 'one-pass' method of connection set-up, which is the model used in all common telecommunications networks (e.g. the telephone network). That is, a connection request from the source end-system is propagated through the network, setting up the connection as it goes, until it reaches the final destination end-system. The routing of the connection request and hence of any subsequent data flow is governed by the ATM routing protocols.

⁵² George C. Sackett & Christopher Y Mertz, *ATM and Multiprotocol Networking*, McGraw Hill, 1997

⁵³ *ATM interface specifications*, <http://cell-relay.indiana.edu/cell-relay/docs/atmforum/ps.html>

⁵⁴ Koichi Asatani et al, *Introduction to ATM networks and B-ISDN*, John Wiley & Sons, 1997

Such protocols route the connection request based upon both the destination address, and the traffic and QoS parameters requested by the source end-system. The destination end-system may choose to accept or reject the connection request, but since the call routing is based purely on the parameters in the initial connection request message, the scope for negotiation of connection parameters between source and destination, which may in turn affect the connection routing. A number of message types are defined in the UNI specification, together with a number of state machines defining the operation of the protocol, cause error codes defining reasons for connection failure, and so forth. Data elements used in the signaling protocol addresses, for instance are carried within *Information Elements (IE)* within the signaling packets.

In overview, a source end-system wishing to set up a connection will formulate and send into the network, across its UNI, a Setup message, containing the destination end-system address, desired traffic and QoS parameters, various IEs defining particular desired higher layer protocol bindings and so forth. This Setup message is sent to the first switch, across the UNI, which responds with a local Call Proceeding acknowledgment. The switch will then invoke an ATM routing protocol, as discussed later on, to propagate the signaling request across the network, to the switch which is attached the destination end-system.

This switch will then forward the Setup message to the end-system, across its UNI. The latter may choose to either accept or reject the connection request; in the former case, it returns a Connect message, back through the network, along the same path, to the requesting source end-system. Once the source end-system receives and acknowledges the Connect message, either node can then start transmitting data on the connection. If the destination end-system rejects the connection request, it returns a Release message, which is also sent back to the source end-system, clearing the connection (e.g. any allocated connection identifiers) as it proceeds. Release messages are also used by either of the end-systems, or by the network, to clear an established connection.

The ATM Forum greatly simplified the Q.2931 protocol, but also extended it to add support for point-to-multipoint connection set up. In particular, UNI 3.1 allows for a root node to set up a point-to-multipoint connection, and to subsequently add a leaf node. While a leaf node can autonomously leave such a connection, it cannot add itself. The ATM Forum has added new capabilities to UNI signaling with the release of UNI 4.0 specification. UNI 4.0 will add support for, amongst other things, leaf-initiated joins to a multipoint connection. While some would like to use this to allow for true multipoint-to-multipoint connections, it should be noted that signaling support for such connections does not imply the existence of a suitable mechanism for such connections. At this time, it is not clear that UNI 4.0 will have any better solution for multicast within ATM than what exists today.

The most important contribution of UNI 3.0/3.1 in terms of internetworking across ATM was its addressing structure. Any signaling protocol, of course, requires an addressing scheme to allow the signaling protocol to identify the sources and destination of connections. The ITU-T has long settled upon the use of telephone number-like E.164 addresses as the addressing structure for public ATM (B-ISDN) networks. Since E.164 addresses are a public (and expensive) resource, and cannot typically be used within private networks, the ATM Forum extended ATM addressing to include private networks. In developing such a private network addressing scheme for UNI 3.0/3.1, the ATM Forum evaluated two fundamentally different models for addressing.

These two models differed in the way in which the ATM protocol layer was viewed in relation to existing protocol layers, in particular, existing network layer protocols such as IP, IPX, and so on. These existing protocols all have their own addressing schemes and associated routing protocols. One proposal was to also use the same addressing schemes within ATM networks. Hence existing network layer addresses (such as Internet Protocol addresses) would identify ATM endpoints, and ATM signaling requests would carry such addresses. Existing network layer routing protocols (such as IGRP and OSPF) would also be used within the ATM network to route the ATM signaling requests, since these requests, using existing network layer addresses, would look essentially like connectionless packets.

This model was known as the **peer model**, since it essentially treats the ATM layer as a peer of existing network layers. An alternate model sought to decouple the ATM layer from any existing protocol, defining for it an entirely new addressing structure. By implication, all existing protocols would operate over the ATM network. For this reason, the model is known as the **subnetwork or overlay model**. This mode of operation is, in fact, the manner in which such protocols as IP operate over such protocols like X.25 or over dial-up lines. The overlay model requires the definition of both a new addressing structure, and an associated routing protocol. All ATM systems would need to be assigned an ATM address in addition to any higher layer protocol addresses it would also support.

The ATM addressing space would be logically disjoint from the addressing space of whatever protocol would run over the ATM layer, and typically would not bear any relationship with it. Hence, all protocols operating over an ATM subnet would also require some form of ATM address resolution protocol to map higher layer addresses (such as IP addresses) to their corresponding ATM addresses. Note that the peer model does not require such address resolution protocols. By using existing routing protocols, the peer model also may have precluded the need for the development of a new ATM routing protocol.

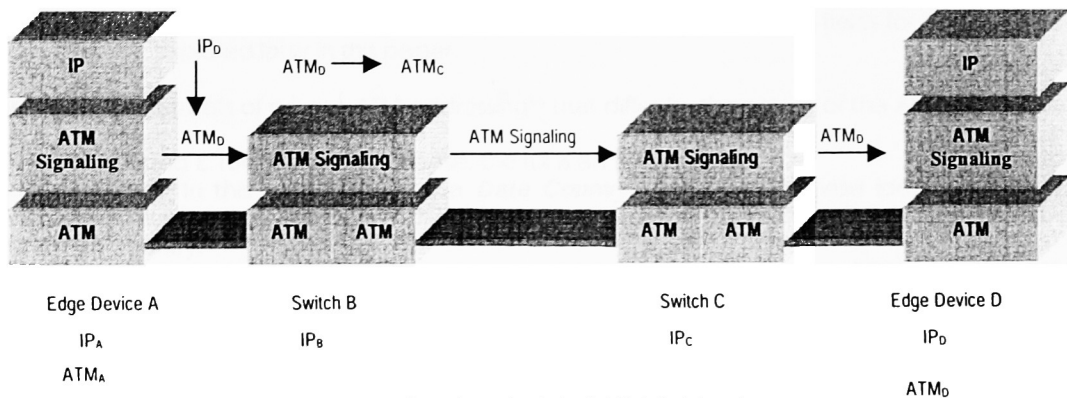


Figure 24 - Overlay Model of ATM Addressing

Nonetheless, it was the overlay model that was finally chosen by the ATM Forum for use with UNI 3.0/3.1 signaling. Among other reasons, the peer model, while simplifying end-system address administration, greatly increases the complexity of ATM switches, since they must essentially act like multiprotocol routers and support address tables for all current protocols, as well as all of their existing routing protocols. Current routing protocols, being originally developed for current LAN and WAN networks, also do not map well into ATM or allow use of ATM's unique QoS properties.

Perhaps most importantly, the overlay model, by decoupling ATM from other higher protocol layers, allows each to be developed independently of the other. This is very important from a practical engineering viewpoint as will be seen, both ATM and evolving higher layer protocols are extremely complex and coupling their development would likely have slowed the deployment of ATM quite considerably. Though there is a price to pay for such layering, in the need for disjoint address spaces and routing protocols, and in possibly sub optimal end-to-end routing, the practical benefits arguably greatly exceed the theoretical costs.

This may happen in large, meshed networks consisting of both packet routers and ATM switches because the higher layer packet routing protocols operate independently of the ATM level routing protocol. Hence once a path is chosen, crossing the ATM network, a change in the topology or characteristics of the ATM layer would not become known to the higher layer routing protocol, even if that change would result in a different, more optimal end-to-end path, bypassing the ATM network, being chosen. While this is indeed a potential drawback of the overlay model, in practice it is unlikely to be a major problem since it is likely that in any practical network the ATM network would always remain the preferred path.

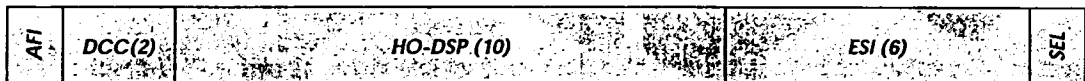
Given the choice of the overlay model, the ATM Forum then defined an address format for private networks based on the syntax of an OSI **Network Service Access Point (NSAP)** address. Note, however, that an ATM address is not an NSAP, despite the similar structure; while in common usage such addresses are often referred to as "NSAP addresses," they are better described as ATM private network addresses, or ATM end-point identifiers, and identify not NSAPs, but subnetwork points of attachment.

The 20-byte NSAP format ATM addresses are designed for use within private ATM networks, while public networks typically use E.164 addresses that are formatted as defined by ITU-T. The Forum did specify, however, an NSAP encoding for E.164 addresses. This will be used for encoding E.164 addresses within private networks but may also be used by some private networks. Such networks may base their own (NSAP format) addressing on the E.164 address of the public UNI to which they are connected and take the address prefix from the E.164 number, identifying local nodes by the lower order bits. All NSAP format ATM addresses consist of three components: an **Authority and Format Identifier (AFI)**, which identifies the type and format of the **Initial Domain Identifier (IDI)**; the IDI, which identifies the address allocation and administration authority; and the **Domain Specific Part (DSP)**, which contains actual routing information. The Q.2931 protocol defines source and destination address fields for signaling requests, and also defines subaddress fields for each; the use of the latter are explored later in this paper.

There are three formats of private ATM addressing⁵⁵ that differ by the nature of the AFI and IDI:

- z NSAP Encoded E.164 format: In this case, the IDI is an E.164 number.
- z DCC Format: In this case, the IDI is a **Data Country Code (DCC)**; these identify particular countries, as specified in ISO 3166. The ISO National Member Body administers such addresses in each country.
- z ICD Format: In this case, the IDI is an **International Code Designator (ICD)**; the ISO 6523 registration authority (the British Standards Institute) allocates these. ICD codes identify particular international organizations.

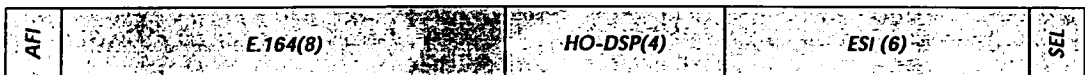
The ATM Forum recommends that organizations or private network service providers use either the DCC or ICD formats to form their own numbering plan. Organizations that want to obtain ATM addresses would do so through the same mechanism used to obtain NSAP addresses (for example, through a local address administration body -- in the US, this is ANSI). Once obtained, such addresses can be used for both ATM addresses and also, if desired, for NSAP addressing.



DCC ATM Format



ICD ATM Format



NSAP ATM Format

Figure 25 - ATM Private Network Address Formats⁵⁶

⁵⁵ UNI & protocols, <http://cne.gmu.edu/~sreddiva/unip.html>

⁵⁶ George C. Sackett, Christopher Y. Metz, *ATM and Multiprotocol Networking*, McGraw-Hill, 1996, page 181

In real NSAPs, the DSP is typically subdivided into a fixed hierarchy that consists of a **Routing Domain (RD)**, an **Area identifier (AREA)**, and an **End System Identifier (ESI)**. The ATM Forum, however, has combined the RD and AREA fields into a single **High-Order DSP (HO-DSP)** field, which is then used to support flexible, multi-level addressing hierarchies for prefix-based routing protocols. No rigid boundary exists within the HO-DSP; instead, a range of addressing hierarchies will be supported, using prefix masks, as with IP subnets.

The ESI field is specified to be a 48-bit **Media Access Control (MAC)** address, as administered by the IEEE. This facilitates the support of both LAN equipment, which is typically hardwired with such addresses, and of such LAN protocols as IPX, which rely on MAC addresses. The final, one octet, **Selector (SEL)** field is meant to be used for local multiplexing within end-stations and has no network significance.

To facilitate the administration and configuration of ATM addresses into ATM end systems across UNI, the ATM Forum defined an address registration mechanism using the **Interim Local Management Interface (ILMI)**. This allows an ATM end-system to inform an ATM switch across the UNI, of its unique MAC address, and to receive the remainder of the node's full ATM address in return. This mechanism not only facilitates the auto configuration of a node's ATM addressing, but may also be extended, in the future, to allow for the auto configuration of other types of information (such as higher layer addresses and server addresses).

Note that the addressing formats defined in UNI 3.0/3.1 identify only single end-points. These can also be used to set up point-to-multipoint connections because in UNI 3.0/3.1 such connections are set up a leaf at a time, using unicast addressing. UNI 4.0 will add support for group addresses, and will permit point-to-multipoint connections to be set up to multiple leaves in one request.

The notion of an anycast address will also be supported in UNI 4.0. A well known anycast address, which may be shared by multiple end systems, is used to route a request to a node providing a particular service, and not to identify the particular node per se. A call made to an anycast address is routed to the "nearest" end-system that registered itself with the network to provide the associated service. Anycast is a powerful mechanism for auto configuration and operation of networks since it precludes the need for manual configuration or service locations protocols. While few details of ATM group addressing have yet been determined, the ATM Forum has decided that anycast will be addressed as a special case of group addressing.

Specifically, nodes will use an extension of the ILMI address registration mechanism to inform the network that they support a particular group address (note that this is the opposite of the normal address registration mechanism). As part of this registration, the node also informs the network of the desired scope of registration, that is, the extent of the network to which the existence of the multicast node should be advertised (as part of the ATM routing protocols). This scope is administrative (such as within a single building, within the local site, or within the enterprise network). The network must map this information through administrative policy to the ATM routing protocol's own hierarchy. Once a node has registered its membership within a multicast group, other nodes may set up connections to these nodes.

If the requesting node initiates a point-to-multipoint connection to the group address, the network will connect all nodes that are registered with that particular ATM address. Conversely, if the requesting node specifies a point-to-point connection, the network will set up a connection to the "nearest" registered node. In this way, anycast can be supported as a special case of group addressing, and a new addressing format is not required.

3.5 ATM Routing Protocols⁵⁷

We now turn to the **Network Node Interface (NNI)** protocols used within ATM networks to route ATM signaling requests between ATM switches. Since ATM is connection oriented, a connection request needs to be routed from the requesting node through the ATM network and to the destination node, much as packets are routed within a packet-switched network. The NNI protocols are hence to ATM networks, what routing protocols (such as OSPF or IGRP) are to current routed networks.

The ATM Forum has an ongoing effort to define a **Private NNI (PNNI)** protocol. The goal is to define NNI protocols for use within private ATM networks, more specifically, within networks that use NSAP format ATM addresses. Public networks that use E.164 numbers for addressing will be interconnected using a different NNI protocol stack based upon the ITU-T B-ISUP signaling protocol and the ITU-T MTP Level 3 routing protocol. This work, being carried out by the **Broadband Inter-Carrier Interface (B-ICI)** subnetworking group of the ATM Forum [Forum4], and other international standards bodies, is not discussed in detail herein.

The PNNI protocol consists of two components: the first is a PNNI signaling protocol used to relay ATM connection requests within the networks, between the source and destination UNI. The UNI signaling request is mapped into NNI signaling at the source (ingress) switch. The NNI signaling is remapped back into UNI signaling at the destination (egress) switch.

The PNNI protocols operate between ATM switching systems (which can represent either physical switches or entire networks operating as a single PNNI entity), which are connected by PNNI links. PNNI links can be physical links or virtual, "multi-hop" links. A typical example of a virtual link is a virtual path that connects two nodes together. Since all virtual channels, including the connection carrying the PNNI signaling, would be carried transparently through any intermediate switches between these two nodes on this virtual path, the two nodes are logically adjacent in relation to the PNNI protocols.

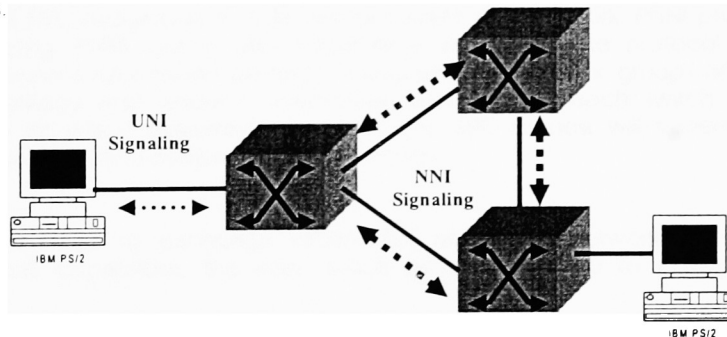


Figure 26: UNI and NNI Signaling

The ILMI protocol, first defined for use across UNI links, will also be used across both physical and virtual NNI links; enhancements to the ILMI MIBs allow for automatic recognition of NNI versus UNI links, and of private versus public UNI.

The current PNNI signaling protocol being developed by the ATM Forum is an extension of UNI signaling and incorporates additional **Information Elements (IE)** for such NNI-related parameters as **Designated Transit Lists (DTL)**. PNNI signaling is carried across NNI links on the same virtual channel, VCI=5, which is used for signaling across the UNI. The VPI value depends on whether the NNI link is physical or virtual.

The second component of the PNNI protocol is a virtual circuit routing protocol. This is used to route the signaling request through the ATM network. This is also the route on which the ATM connection is set up, and along which the data will flow. The operation of routing a signaling request through

⁵⁷ George C. Sackett, Christopher Y. Metz. ATM and Multiprotocol Networking. McGraw-Hill. 1996

an ATM network, somewhat paradoxically, given ATM's connection oriented nature, is superficially similar to that of routing connectionless packets within existing network layer protocols (such as IP). This is due to the fact that prior to connection set up, there is, of course, no connection for the signaling request to follow.

As such, a VC routing protocol can use some of the concepts underlying many of the connectionless routing protocols that have been developed over the last few years. However, the P-NNI protocol is much more complex than any existing routing protocol. This complexity arises from two goals of the protocol: to allow for much greater scalability than what is possible with any existing protocol, and to support true QoS-based routing.

The current state of the PNNI protocols will be examined by looking at the manner in which the protocol tackles these challenges. It should be noted, however, that the ATM Forum did define an interim protocol whilst it was working on the specification of the "PNNI Phase 1" protocol, which was carried under the specification called "P-NNI Phase 0" protocol, the *Interim Inter-Switch Signaling Protocol (IISP)*⁵⁸. This protocol will be examined after examining Phase 1 of PNNI. Finally, multicast routing, how private and public ATM networks interconnect, and implementation considerations for PNNI are discussed.

Both the PNNI Phase 1 protocol, and the IISP protocol, currently only will interface with, and support the capabilities of, UNI 3.0/3.1 signaling. In particular, neither of these protocols will support such aspects of UNI 4.0 signaling as leaf-initiated joins, group addressing, or ABR connection parameter negotiation. Such functionality will be added to the PNNI protocols as part of the current PNNI Phase 2 protocol specification, which is being worked on.

3.5.1 PNNI requirements

PNNI is a switch to switch protocol developed by ATM forum to support efficient, dynamic, and scalable routing of SVC requests in a multi-vendor private ATM network. PNNI phase I consists of routing and signaling. PNNI routing uses hierarchical topology state protocol to disseminate topology and resources information among participating switches or groups of switches. PNNI signaling uses topology and resource information available at each switch to construct a designated transit list which determines the path the SVC request will traverse to meet the requested QoS objectives and complete the connection.

Since ATM switches had to exchange information about the network topology, available resources, and QoS capabilities, the inter switch protocol needed to possess the following properties:

- ☛ **Scalable** – must be able to support small or large network of ATM switches
- ☛ **Simple to install and configure** – configurations should be minimal
- ☛ **Efficient routing of SVCs** – use the best path that will meet the QoS objective
- ☛ **Administrations of source and transit policies** – effect security, usage and traffic policies for different domains
- ☛ **Multi-vendor** – must support a network of multi vendor switches

Designing such a protocol to meet the said objectives was the goal of ATM forum and the concepts which went into engineering the protocol called PNNI are reviewed in the next section.

3.5.2 PNNI Concepts

PNNI is a switch to switch protocol developed to support efficient, dynamic, and scalable routing of SVC requests in a multi vendor environment. PNNI phase I consists of two protocols.

⁵⁸ Ulyess Balck, *ATM: Foundation for Broadband Networks*, Prentice Hall, 1995

➤ Routing

PNNI routing is used to distribute information on the topology of the ATM network between switches and groups of switches. This information is used by the switch closest to the SVC requestor to compute a path to the destination that will satisfy QoS objectives. PNNI supports a hierarchical routing structure that allows it to scale to large networks.

➤ Signaling

PNNI signaling uses the topology and resource information available at each switch to construct a source route path called **Designated Transit List (DTL)**. This lists the specific nodes and links the SVC request will have to traverse to meet QoS objectives and complete the connection. Crankback and alternate routing are also supported to route around a failed path.

PNNI has benefited greatly from previous research on routing mechanisms and an important design criteria was to implement where appropriate, existing mechanisms to provide a desired function or operation. Hence, we see PNNI using several of the existing techniques that have been previously implemented in other internetworking protocols and some of them are,

- Link state routing
- Hierarchical routing
- Source routing

PNNI uses link-state vector as the primary algorithm as opposed to distance vector. The choice was obvious since distance vector had associated issues with regard to scalability, slow convergence, routing loops and overhead traffic associated with its operation. Thus, Link state has proven to be more scalable, converges rapidly, generates less overhead traffic and is also extensible. Extensible in the sense it can be used to exchange information other than link status between nodes and readily incorporated into the topology database.

PNNI is a topology state protocol. ATM switches use this in exchanging information on the links and nodes in the status update messages sent to neighboring switches. The nodal information may include data about switch capacity, QoS and transit time. This information is vital in routing SVC requests over a path that meets its QoS objectives.

PNNI must also be scalable and to achieve this it implements a hierarchical routing structure. In a hierarchical routing structure topology and addressing information about a group of nodes are summarized and presented as a single node in the next level up the hierarchy. Topology aggregation as this process is called reduces complexity and simplifies the tables at the expense of some accuracy about the information being presented.

Source routing is used by PNNI to enable the first switch in the SVC request path to compute the entire path, based on its knowledge of the network. Because QoS metrics are advertised and contained in topology databases, the first switch has a good idea about which path to take. Also, source routing helps prevent routing loops.

3.5.3 PNNI Routing

PNNI specifications define the following,

- **Peer Group** - A collection of nodes that maintains an identical topology database and exchange topology and resource information with each other. Members of the peer group discover their neighbours using a hello protocol.
- **Peer Group Identifier** - Members of the same peer group are identified by a common peer group identifier. The peer group identifier is derived from a unique 20 byte ATM address that is manually configured on the switch.
- **Logical node** - A logical node is any switch or group of switches that runs PNNI routing protocol.

- ⌚ **Logical group node (LGN)** – An LGN is an abstract representation of a lower level peer group for the purposes of representing that peer group in the next higher level peer group.
- ⌚ **Parent peer group** – This contains the LGN representing the peer group below it.
- ⌚ **Child peer group** – This contains the node that is part of a LGN in the next higher level peer group.
- ⌚ **Peer group leader (PGL)** – Within the peer group, a PGL is elected to represent the peer group as a logical group node in the next higher level peer group.
- ⌚ **Hello protocol** – This is a standard link state procedure used by neighbor nodes to discover the existence and identity of each other.
- ⌚ **Border nodes** – A border node is a logical node which has a neighbor that belongs to a different peer group.
- ⌚ **Uplinks** – This is a logical connection from a border node to a higher level LGN.
- ⌚ **Logical link** – This is a connection between two nodes.
- ⌚ **Routing control channel** – VPI=0 and VCI=1B is reserved as the VC used to exchange routing information between logical nodes.
- ⌚ **Topology Aggregation** – This is the process of summarizing and compressing information at one peer group to advertise into the next higher level peer group.
- ⌚ **PNNI topology state element (PTSE)** – This is used by nodes to build and synchronize a topology database within the same peer group.

The process of building PNNI peer group is recursive. It is used at each level of the hierarchy except the lowest and the highest level peer groups due to the lack of a child and parent nodes respectively. As discussed before PNNI is a topology-state protocol. This means that logical nodes will advertise link state and nodal state parameters where the former describes the characteristics of the link and the latter describes the characteristics of the node.

3.5.4 PNNI - QoS Support⁵⁹

One of the great advantages of ATM is its support for guaranteed QoS in connections. Hence, a node requesting a connection set up can request a certain QoS from the network and can be assured that the network will deliver that QoS for the life of the connection. Such connections are categorized into various types of ATM QoS types: CBR, VBR, ABR, and UBR, depending upon the nature of the QoS guarantee desired and the characteristics of the expected traffic types (see Appendix A). Depending upon the type of ATM service requested, the network is expected to deliver guarantees on the particular mix of QoS elements that are specified at the connection set-up (such as cell loss ratio, cell delay, and cell delay variation).

To deliver such QoS guarantees, ATM switches implement a function known as **Connection Admission Control (CAC)**. Whenever the switch receives a connection request, the switch performs the CAC function. That is, based upon the traffic parameters and requested QoS of the connection, the switch determines whether setting up the connection violates the QoS guarantees of established connections (for example, by excessive contention for switch buffering). The switch accepts the connection only if violations of current guarantees are not reported. CAC is a local switch function, and is dependent on the architecture of the switch and local decisions on the strictness of QoS guarantees.

The VC routing protocol must ensure that a connection request is routed along a path that leads to the destination and has a high probability of meeting the QoS requested in the connection set up, that is, of traversing switches whose local CAC will not reject the call.

⁵⁹ *ATM Quality of Service*, <http://www.telecoms-mag.com/marketing/articles/feb97/bennet.html>

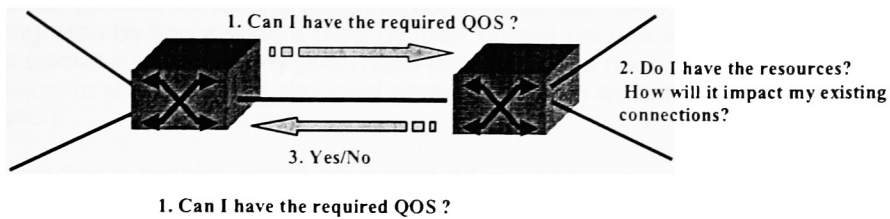


Figure 27 - Connection Admission Control⁶⁰

To do this, the protocol uses a topology state routing protocol in which nodes flood QoS and reachability information so that all nodes obtain knowledge about reachability within the network and the available traffic resources within the network. Such information is passed within **PNNI topology state packets (PTSP)**, which contain various **type-length-value (TLV)** encoded **PNNI topology state elements (PTSE)**. This is similar to current link state routing protocols such as OSPF. Unlike these, however, which only have rudimentary support for QoS, the PNNI protocol supports a large number of link and node state parameters that are transmitted by nodes to indicate their current state at regular intervals, or when triggered by particular events.

There are two types of link parameters: non-additive link attributes used to determine whether a given network link or node can meet a requested QoS; and additive link metrics that are used to determine whether a given path, consisting of a set of concatenated links and nodes (with summed link metrics), can meet the requested QoS.

The current set of link metrics are⁶¹:

- ⌚ **Maximum cell transfer delay (MCTD)** per traffic class
- ⌚ **Maximum cell delay variation (MCDV)** per traffic class
- ⌚ **Maximum cell loss ratio (MCLR)** for CLP=0 cells, for the CBR and VBR traffic classes
- ⌚ **Administrative Weight:** This is a value set by the network administrator and is used to indicate the desirability or otherwise of a network link.

The current set of link attributes is:

- ⌚ **Available Cell Rate (ACR):** A measure of the available bandwidth in cells per second, per traffic class
- ⌚ **Cell Rate Margin (CRM):** A measure of the difference between the effective bandwidth allocation per traffic class, and the allocation for sustainable cell rate; this is a measure of the safety margin allocated above the aggregate sustained rate
- ⌚ **Variance Factor (VF):** A relative measure of CRM margin normalized by the variance of the aggregate cell rate on the link

All network nodes can obtain an estimate of the current state of the entire network through flooded PTSPs that contain such information as listed above. Unlike most current link state protocols, the PNNI protocol advertises not only link metrics, but also nodal information. Typically, PTSPs include bidirectional information about the transit behavior of particular nodes based upon entry and exit port, and current internal state. This is particularly important in cases where the node represents an aggregated network. In such a case, the node metrics must attempt to approximate the state of the entire aggregated network. This internal state is often at least as important as that of the connecting links for QoS routing purposes.

⁶⁰ Whay C Lee, Michael G Hluchyj & Pierre A Himblet, *Routing subject to QoS service constraints*, IEEE Networks, July 1995

⁶¹ Uyless Black, *Internetworking with ATM*, Prentice Hall, 1995

The need to aggregate network elements and their associated metric information also has important consequences on the accuracy of such information, as discussed below. Two approaches are possible for routing a connection through the network: hop-by-hop routing and source routing. Hop-by-hop routing is used by most current network layer protocols such as IP or IPX, where a packet is routed at any given node only to another node, the "next hop" closer to the final destination. In source routing, the initial node in the path determines the entire route to the final destination.

Hop-by-hop routing is a good match for current connectionless protocols because they impose little packet processing at each intermediate node. The PNNI protocol, however, uses source routing for a number of reasons. For instance, it is very difficult to do true QoS-based routing with a hop-by-hop protocol since each node needs to perform local CAC and evaluate the QoS across the entire network to determine the next hop. Hop-by-hop routing also requires a standard route determination algorithm at each hop to preclude the danger of looping.

However, in a source-routed protocol, only the first node would ideally need to determine a path across the network, based upon the requested QoS and its knowledge of the network state, which is gained from the PTSPs. It could then insert a full source routed path into the signaling request that would route it to the final destination. Ideally, intermediate nodes would only need to perform local CAC before forwarding the request. Also, since it is easy to preclude loops when calculating a source route, a particular route determination algorithm does not need to be standardized, leaving this as another area for vendor differentiation.

This description is only ideal, however. In practice, the source routed path that is determined by a node can only be a best guess. This is because in any practical network, any node can have only an imperfect approximation to the true Network State because of the necessary latencies and periodicity. As discussed in the next section, the need for hierarchical summarization of reachability information also means that link parameters must also be aggregated. Aggregation is a "lossy" process, and necessarily leads to inaccuracies. Furthermore, as noted above, CAC is a local matter. In particular, this means that the CAC algorithm performed by any given node is both system dependent and open to vendor differentiation.

The PNNI protocol tackles these problems by defining a **Generic CAC (GCAC)** algorithm. This is a standard function that any node can use to calculate the expected CAC behavior of another node, given that node's advertised additive link metrics, described above, and the requested QoS of the new connection request. The GCAC is an algorithm that was chosen to provide a good prediction of a typical node-specific CAC algorithm, while requiring a minimum number of link state metrics. Individual nodes can control the degree of stringency of the GCAC calculation involving the particular node by controlling the degree of laxity or conservativeness in the metrics advertised by the node.

The GCAC actually uses the additive metrics described above; indeed these metrics were selected to support the GCAC algorithm chosen for the PNNI protocol. Individual nodes (physical or logical) will need to determine and then advertise the values of these parameters for themselves, based upon their internal structure and loading. Note, however, that the PNNI Phase 1 GCAC algorithm is primarily designed for CBR and VBR connections; variants of the GCAC are used depending upon the type of QoS guarantees requested and the types of link metrics available, yielding greater or lesser degrees of accuracy.

The only form of GCAC done for UBR connections is to determine whether a node can support such connections. For ABR connections, a check is made to determine whether the link or node is authorized to carry any additional ABR connections and to ensure that the ACR for the ABR traffic class for the node is greater than the Minimum Cell Rate specified by the connection. The details of the GCAC are described in Forum5. Using the GCAC, a node presented with a connection request (which passes its own CAC) processes the request as follows:

- ⌘ All links that cannot provide the requested ACR, and those whose CLR exceeds that of the requested connection, are "pruned" from the set of all possible paths using the GCAC.
- ⌘ From this reduced set, along with the advertised reachability information, a shortest path computation is performed to determine a set of one or more possible paths to the destination.
- ⌘ These possible paths are further pruned by using the additive link metrics, such as delay, and possibly other constraints. One of the acceptable paths would then be chosen. If multiple paths are found, the node may optionally perform tasks such as load balancing.
- ⌘ Once such a path is found (note that this is only an "acceptable" path to the destination, not the "best" path, the protocol does not attempt to be optimal), the node constructs a designated transit list (DTL) that describes the complete route to the destination (the structure of the DTL is described below) and inserts this into the signaling request. The request is then forwarded along this path.

This, however, is not the end of the story. Each node in the path still performs its own CAC on the routed request because its own state may have changed since it last advertised its state within the PTSP used for the GCAC at the source node. Its own CAC algorithm is also likely to be somewhat more accurate than the GCAC. Hence, notwithstanding the GCAC, there is always the possibility that a connection request may fail CAC at some intermediate node. This becomes even more likely in large networks with many levels of hierarchy, since QoS information cannot be accurately aggregated in such cases. To allow for such cases, without excessive connection failures and retries, the P-NNI protocol also supports the notion of crankback.

3.5.5 PNNI - Scalability and Reachability⁶²

In addition to providing true QoS support, the ATM Forum has also set the goal of universal scalability for the PNNI Phase 1 protocol. The P-NNI Phase 1 protocol is being designed to be capable of being applied both to small networks of a few switches and to a possible future global ATM Internet comprising millions of switches. Such scalability is well beyond that of any single routing protocol today. The Internet, for instance, supports many different types of routing protocols, intra-domain routing protocols, such as *Interior Gateway Routing Protocol (IGRP)* or *Open Shortest Path First (OSPF)*, which scale to large enterprise networks, and inter-domain protocols, such as *Border Gateway Protocol (BGP)* or IDRP, which interconnect such lower level networks. By building upon the many years of experience gained in the development of such current protocols, however, the ATM Forum hopes to build a single protocol that could perform at all levels within a network.

The key to such a scalable protocol is hierarchical network organization, with summarization of reachability information between levels in the hierarchy. Protocols such as OSPF implement such mechanisms, but only implement two level of hierarchy, which is inadequate for very large networks. The PNNI protocol, however, uses the 20-byte *Network Services Access Point (NSAP)* addresses to identify levels in the network hierarchy to support an almost limitless number of levels: a maximum of 105 (the number of bits in the 13 high-order bytes of the NSAP address, excluding the ESI and SEL fields), though no more than a half dozen or so will likely ever need to be used, and even then only within the very largest, global networks.

To support this hierarchy, the PNNI model defines a uniform network model at each level of the hierarchy. The PNNI hierarchical model explains how each level of the hierarchy operates, how multiple devices or nodes at one level can be summarized into the higher level, and how information is exchanged between levels. The model is recursive in that the same mechanisms used at one level are also used at the next level.

⁶² *PNNI specifications*, [ftp://ftp.atmforum.com/pub/approved-specs/af-pnni-0055.000.pdf](http://ftp.atmforum.com/pub/approved-specs/af-pnni-0055.000.pdf)

Each level in the hierarchy consists of a set of logical nodes, interconnected by logical links. At the lowest level, each logical node represents a physical switching system consisting of a single physical switch, or a network of switches that internally operate a proprietary NNI protocol and support the PNNI protocol for external connectivity. At this lowest level, each switching system must be assigned a unique ATM NSAP address.

Nodes within a given level are grouped into sets known as a peer group. The definition of a peer group is a collection of nodes that all obtain the identical topological database and exchange full link state information with each other. While all nodes within a peer group have complete state information on each other, peer groups cannot be extended too widely since this would lead to excessive PTSP traffic and processing. Hence, peer groups are organized hierarchically and are associated with a higher level parent peer group.

Within its parent peer group, each peer group is represented, by default, as a single logical node, known as the logical group node. Within the parent peer group, the logical group node acts as a normal node, exchanging PTSPs with the other nodes within the parent peer group. The peer groups represented by logical group nodes within a parent group are known as the child peer groups of that group.

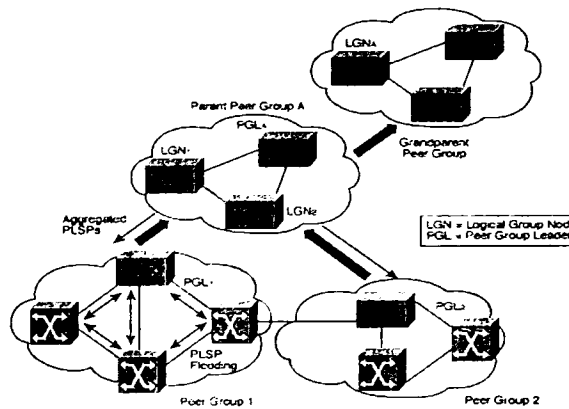


Figure 28 - The PNNI Network Hierarchy Mode⁶³

Normally, peer groups are identified by strict prefixes of private ATM addresses. At the lowest level, where switching systems consist of actual switches, and where by default, all end systems connected to a switch obtain their network address prefix from that of the switch (which implies that end system reachability defaults to switch reachability), the default peer group ID is the high 12 order bytes of the switch NSAP address. This allows for up to 256 switches within this lowest level peer group, without requiring any manual configuration of peer group IDs of the switches or configuration of the end systems.

At higher levels, the default for a peer group ID is a prefix on a lower level peer group ID. The peer group ID of a parent must be shorter than the prefix of its child peer group ID; this makes it easy to determine the relationship between two peer groups, and precludes the formation of a peer group hierarchy loop. Hence, the peer group ID becomes smaller as the hierarchical level becomes larger.

A 22-byte node identifier identifies nodes within a peer group. At the lowest level, this is essentially the same as the switching system's ATM address. At higher levels, the node ID (which now identifies

⁶³ George C. Sackett, Christopher Y. Metz, *ATM and Multiprotocol Networking*, McGraw-Hill, 1996

logical group nodes) includes two level indicators that indicate the hierarchical level (that is, prefix length) of both the associated peer group and the child peer group, plus the peer group ID.

In addition to nodes, the PNNI protocol also requires that links be identified since links between peer groups need to be identified in PTSPs and may also be optionally specified in DTLs. Since ATM link attributes can be asymmetrical (since connections may be asymmetrical), links are identified by a combination of a transmitting node ID and a locally assigned port ID. Nodes exchange such port IDs between themselves (using the Hello protocol discussed below) and hence together identify particular links. In practice, link identification is somewhat more complex, since multiple physical or virtual links may need to be aggregated. (Refer to [Forum5] for more details.)

Each peer group elects a single node within the group to perform the functions of the logical group node. This node, known as the **peer group leader (PGL)**⁶⁴, is selected through an election mechanism and is based upon a "leadership priority" and the switches' node ID. Each PGL is identified by a unique ATM address; if a node acts as a PGL within multiple levels of peer groups, then it must have a unique ATM address at each of those levels.

PGLs within each peer group have the responsibility of formulating and exchanging PTSPs with their peer nodes within the parent peer group to inform those nodes of the child group's reachability and attributes. Similarly, recursive information obtained by the PGL about the parent group and that group's parent groups are then fed down by the PGL into the child group. The child nodes can then obtain knowledge about the full network hierarchy, in order to construct full source routes.

At higher levels of the PNNI hierarchy, multiple outside links may be aggregated together into fewer logical uplinks, but information about the binding between logical uplinks and their constituent outside links must be advertised so that nodes can map a logical inter-peer group link into a physical link.

Border nodes also exchange information about the PGLs of their own peer groups. This allows the PGLs of groups that discover that they are within the same parent peer group to set up connections to each other, across the identified uplinks, and start exchanging their own Hellos and PTSPs. They then discover the existence of yet higher level peer groups, until all nodes discover their entire network hierarchy. Through fed-down PTSPs, containing summarized reachability and uplink information, the PGLs discover full network state.

Once full state information is obtained by all nodes, they can then use this to route signaling requests. When a signaling request is received across a UNI by a switch, the DTL originator, the switch will use a shortest path algorithm, such as a Dijkstra calculation, to determine one or more paths that connect the source node to the desired destination, using the algorithm described in the previous section. This calculation will create a hierarchically complete source route, that is, a set of DTLs, which will have: a full, detailed path within the source node's own peer group; a less detailed path within the parent peer group; and even less detail on higher level peer groups, terminating in the lowest level peer group which is an ancestor of both the source and destination nodes.

These DTLs are arranged in a stack within the PNNI signaling request where each DTL contains the path elements for one level in the hierarchy. This comprises a list of node and, optionally, link IDs, together with a pointer that indicates which element in the list is to be processed next. Within a given peer group, that peer group's DTL is processed by nodes until it reaches a node that is a border node to the next peer group on the path. At this point, the DTL of that peer group is exhausted, since the final element in that DTL is the ID of the border node. The border node then removes that DTL, notes that the next DTL points to the neighbor peer group (possibly at a different level in the hierarchy), and forwards it to its peer border node within that neighbor peer group.

⁶⁴ George C. Sackett, Christopher Y. Metz, *ATM and Multiprotocol Networking*, McGraw-Hill, 1996

Once the request arrives at that border node within that neighbor peer group, that node discovers that the request must be routed through that node's peer group. Typically, however, the original DTL only has aggregated information about this neighbor peer group. The border node then constructs one or more new DTLs, describing how to route the request through its peer group and "pops" it onto the top of the stack of DTLs. In this way, the request is forwarded to a border node within this peer group, which performs a similar function for the next peer group in the path, and so on, until the final destination peer group is reached.

At this point, the border node will construct a DTL that routes the request to the switch on which the destination end system is attached. There, the final switch which is the DTL terminator re-maps the request into UNI signaling and forwards it across the appropriate UNI link. DTLs are hence only created by the source node and by border nodes. Other intermediate nodes only process DTLs and move the DTL pointer forward and pass the request to the next node on the path.

Crankback (discussed below) works within this same mechanism to make the previous description more precise, connections can only be cranked back to nodes that actually create and insert DTLs into a request from the original source node, or border nodes. Such nodes maintain state information about all requests that they have forwarded until the connection set up is confirmed, or a connection reject is received from the destination end system. If, however, an intermediate node rejects the call (for example, due to local CAC), then the call is rerouted back along the path that it followed to that node to the last node to insert a DTL. If possible, this node then recalculates a new path across its own peer group, avoiding the node that rejected the call, and re-forwards the request.

Good examples of the operation of both PNNI routing and crankback are given in [Forum5] and are highly recommended, since a proper description of the PNNI procedures is outside the scope of this research. While the procedures outlined here can be scaled to very large networks, it should be noted that the aggregation used to ensure such scalability also fundamentally works against the QoS routing properties of ATM. This is because the QoS metrics discussed in the previous section must also be aggregated to match the aggregation of network topology inherent in the network hierarchy; aggregation, however, is a fundamentally "lossy" process. At the lowest level, such metrics might yield information about the state of particular switch and link combinations. At higher levels, the same metrics must attempt to approximate the "average" state of entire networks, which consists of many individual switches.

Clearly such aggregated information will be much less accurate than information about individual switches. This problem is exacerbated by the fact that at higher levels entire peer groups are represented by single nodes (that is, logical group nodes). Advertising metrics about such nodes imply an assumption about the symmetry and compactness of the topology of the child peer group and its traffic flows, which is very unlikely to be accurate in practice.

To ameliorate this problem, the PNNI protocol allows a peer group to be modeled at higher levels, for advertising purposes, not as a single node but as a "complex node," with an internal structure. The Phase 1 PNNI protocol allows complex nodes to be modeled as a star of nodes that consists of a "pseudo-node" connected to a group of border nodes across "pseudo-links," each with an identical radius for each link parameter. These nodes need not necessarily correspond to any actual physical node, but the hope is that the "radius" advertised for this abstract network better represents the metrics across the actual peer network, than by modeling it by a single node. Modeling peer groups in this fashion require much more information to be advertised and modeled within PTSPs. There are more complex and possibly more accurate ways to model a peer group other than a star (such as a mesh or spanning tree). Future phases of the PNNI protocol might allow for these alternate models of complex nodes.

In addition to summarized addresses, a number of other elements of reachability information are also carried within PTSP. Routes to external networks, reachable across exterior links, are advertised as external addresses. Peer groups may also include nodes with non-aggregatable addresses, which must also be advertised, as must registered group and anycast addresses. Generally none of

these types of information can be summarized, since they fall outside the scope of the default PNNI address hierarchy.

Note that the scope of advertisement of the group addresses is a function of how the network administrator maps the administrative scope of a registered node to the corresponding PNNI hierarchy. The PNNI protocol also has support for "soft permanent virtual connection" set-up. The latter is a means of setting up PVCs and **Permanent Virtual Paths (PVP)** using PNNI procedures. Through network management, a PVC or PVP is established only across the source and destination UNI, but not across the entire network. Then, through network management the first switch is instructed to route a connection across the network to the destination switch using PNNI. This is done with the usual P-NNI procedures, but hooks in the signaling instruct the destination switch to terminate the connection on the pre-established PVC/PVP, rather than forwarding a UNI signaling request to the destination end-system.

Given the need to use permanent connections (because end-systems do not support signaling, for instance), soft connection set-up is a much more convenient and reliable way to set up such connections rather than using hop-by-hop configuration. This also allows permanent connections to be set up with a specific QoS using the PNNI procedures. While the procedures outlined here can be scaled to very large networks, it should be noted that the aggregation used to ensure such scalability also fundamentally works against the QoS routing properties of ATM. This is because the QoS metrics discussed in the previous section must also be aggregated to match the aggregation of network topology inherent in the network hierarchy; aggregation, however, is a fundamentally "lossy" process. At the lowest level, such metrics might yield information about the state of particular switch and link combinations. At higher levels, the same metrics must attempt to approximate the "average" state of entire networks, which consists of many individual switches.

3.5.6 Crankback and alternate routing

Crankback is where a connection which is blocked along a selected path is rolled back to an intermediate node, earlier in the path. This intermediate node attempts to discover another path to the final destination, using the same procedure as the original node, but uses newer, or hopefully more accurate network state. This is another mechanism that can be much more easily supported in a source-routed protocol than in a hop-by-hop protocol.

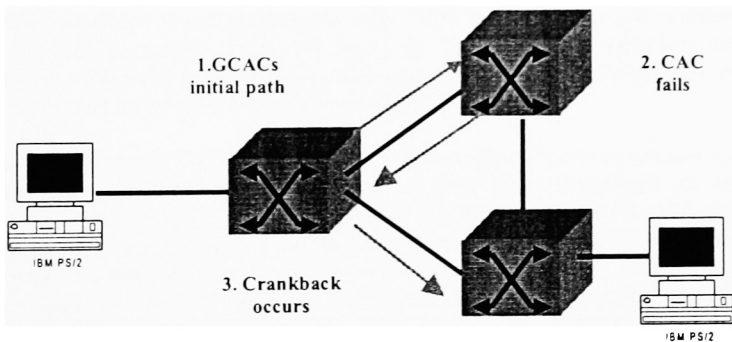


Figure 29: Operation of Crankback

One of the concerns with PNNI route generation is that most commonly used routing algorithms (such as Dijkstra calculations) were designed for single, cumulative metrics such as link weightings or counts. Since PNNI uses a number of complex link parameters for link pruning, path selection may often not generate any acceptable paths. In such cases, sophisticated algorithms may use a technique known as fallback, where particular attributes (such as delay) are selectively relaxed, and paths are recalculated in order to find a path that meets some minimal set of desired attributes. In general, path selection, like CAC, is an area with considerable scope for vendor differentiation.

3.5.7 The IISP Protocol⁶⁵

While the PNNI Phase 1 protocol is extremely powerful, it is also quite complex. For this reason, the ATM Forum's work on the protocol went beyond the second half of 1995 and the actual interoperable implementations were only widely deployed by 1997. Unfortunately, without a PNNI protocol, there is no standard way for users to build interoperable multivendor ATM networks. Many users were not willing to wait until 1996 for such interoperability since they have pressing needs to test multiple vendor's switches within the ATM test beds that they are currently running. To solve this short-term protocol, Cisco Systems proposed to the ATM Forum that it develop a very simple, UNI-based signaling protocol for switch interoperability.

Originally designated the PNNI Phase 0 protocol, this was later renamed the Interim **Inter-Switch Signaling Protocol (IISP)** to avoid confusion with the P-NNI Phase 1 protocol. This protocol was completed and approved by the ATM Forum. The IISP, as the name suggests, is essentially a signaling protocol for inter-switch communication. Given the fact that the UNI 3.0/3.1 signaling procedures are essentially symmetrical, it uses UNI signaling for switch-to-switch communication, with nodes arbitrarily taking the role of the network and user side across particular switch-to-switch links (known as IISP links).

Signaling requests are routed between switches using configured address prefix tables within each switch, which precludes the need for a VC routing protocol. These tables are configured with the address prefixes that are reachable through each port on the switch. When a switch receives a signaling request, either across a UNI or an IISP link, the switch checks the destination ATM address against the prefix table and notes the port with the longest prefix match. It then forwards the signaling request across that port using UNI procedures.

The IISP protocol is very simple and does not require modification to UNI 3.0/3.1 signaling or any new VC routing protocol. It can leverage current development efforts on UNI signaling and hence can be deployed very quickly. The IISP, however, does not have anywhere near the same scalability as the Phase 1 protocol. For instance, manually configuring prefix tables limits its applicability to networks with only a small number of nodes. This is adequate for now, given that most ATM switches today are deployed in small test beds and not in large scale production networks.

IISP implementations will not be interoperable with PNNI Phase 1 implementations because IISP only uses UNI and not NNI signaling. Users will need to upgrade their switches when P-NNI Phase 1 becomes available. This was deliberately done to simplify the specification and accelerate the deployment of IISP, and to emphasize its interim nature.

The IISP also does not support QoS-based routing, although nodes may implement CAC; it does not support crankback, though nodes can be configured with redundant or alternate paths (the selection of such paths being a local matter). These limitations of the IISP, however, are not as restrictive as might first be imagined. While the Phase 1 protocol has extensive support for QoS routing, this is required only for routing VBR and CBR connections, where end systems can request a specific QoS. End systems that request either **Unspecified Bit Rate (UBR)** or **Available Bit Rate (ABR)** connections, however, can specify only very limited QoS capabilities. As such, the P-NNI protocol metrics do not apply to such connections and must be routed using some other criteria -- such as shortest path.

Most data traffic on ATM networks will likely use UBR or ABR connections in the short to medium term, since higher layer protocols cannot specify QoS (and hence use VBR connections). Given these factors, it is likely that IISP will be widely deployed prior to the final specification and deployment of the P-NNI Phase 1 protocol, though it will certainly be supplanted by the latter as it becomes available.

⁶⁵ **UNI & protocols**, <http://cne.gmu.edu/~sreddiva/unip.html>

3.5.8 Multicast Routing⁶⁶

In the first instance, with UNI 3.0/3.1, point-to-point connections will be set up a leaf at a time, with each add-leaf request addressed by the leaf's unicast ATM address. Hence IISP and the PNNI Phase 1 protocol will rout such connection requests in the same manner as point-to-point connections.

The only difference is that the signaling procedures will ensure that no new connections are set up across a link for a particular add-leaf request if a branch of the point-to-multipoint connection already exists across that link. Ideally, a new branch of the tree will be added only at the point "closest" to the new leaf, where the connection must branch off to the new leaf. In terms of the PNNI Phase 1 operation, this may impact the selection of possible routes during the route-pruning phase.

Through this support of point-to-multipoint connections, the PNNI Phase 1 and IISP protocols will support existing UNI 3.0/3.1 multicast mechanisms such as multicast servers and overlaid point-to-multipoint connections.

With UNI 4.0, support will need to be added for group addressing. Reachability information about registered group addresses can be advertised within PTSP in the Phase 1 protocol, and can be configured within the IISP protocol. This does not address, however, the support of such new UNI 4.0 mechanisms as leaf-initiated joins and the addition of multiple leaves in a single point-to-multipoint connection request. The PNNI group deferred such issues to a possible Phase 2 effort.

This effort may tackle ways to automatically configure groups of ATM end-points into some form of multicast group, based upon their registration of membership within the multicast group. Support will also be needed for a multicast routing protocol to allow for point-to-multipoint connections to group addresses, since the PNNI protocols will then need to generate a source rooted tree linking the source to each of the leaves. Such a protocol may build upon such existing multicast protocols as *Protocol Independent Multicast (PIM)*.

3.5.9 PNNI Routing extensions

PNNI was designed to support signaling and routing of SVC requests through a network of ATM switches. However, PNNI can be extended to support not only the routing of VSC requests but also the routing of layer 3 packets such as IP. Two such proposals which have made it through to the drawing boards. They are;

- z5 **PNNI Augmented Routing (PAR)** – With PAR, IP routers and ATM switches separate routing protocols. OSPF for IP and PNNI for ATM. However, routers attached to the ATM network would also run PNNI. This would enable routers attached to the same ATM network to locate each other and setup SVCs. PAR enables ATM attached routers to bootstrap a series of inter router SVCs over the ATM network, thus removing the requirement to preconfigure ATM information in the router. It would also provide ATM attached routers, with information about QoS capabilities of the ATM network.
- z5 **Integrated-PNNI (I-PPNI)** – IPNNI is a single routing protocol that is used between IP routers and ATM switches. Routers and switches exchange topology information about the entire network. This enables end to end routing to be based on a single network topology, supporting QoS routing and requires only one routing protocol to be configured and managed.

⁶⁶ *Multicast Architectures for ATM networks*. <http://www.eantc.de/Documents/Standards/IETF/internet-drafts/draft-ietf-ion-marsmcs-02.txt>

3.6 ATM and the OSI Model

An issue that often causes great confusion is that of to which layer in the OSI 7-layer model ATM corresponds. The adoption of the overlay model by the ATM Forum, as described in the previous section, sometimes cause some to describe ATM as a layer 2 protocol that is, a data link protocol, akin to a MAC protocol like Ethernet or Token Ring. Yet this description is often contested by others who note that ATM possesses most if not all of the characteristics of a layer 3 or network layer protocol, such as IP or IPX. These characteristics include a hierarchical address space and, as will be described in the next section, a complex routing protocol.

In practice, much of the controversy arises both from limitations of the OSI model, and from an incomplete understanding of the complexities of practical network operation. The basic OSI model did not incorporate the concept of overlay networks, where one network layer must overlay another, though such concepts were later added as addenda to the model. As we discussed in the previous section, such a model is often used where one type of network protocol must be carried transparently across another. Today, for instance, such layer 3 protocols as IP and IPX are often carried (tunneled) across other network layer protocols like X.25 or the telephone network, for instance since this is generally much simpler than attempting to interoperate the protocols through a protocol gateway.

As noted in the previous section, the ATM overlay model was chosen so as to separate and hence facilitate the engineering efforts involved in both completing the ATM layer protocols, as well the efforts needed to modify existing protocols to operate with ATM. The overlay model also simplifies switch operation, at the arguable cost of redundancy in protocol functions and sub optimality in routing. As we will discuss later, the overlay model also leverages the existing installed application base, and facilitates future application portability, since it builds upon and extends today's ubiquitous network layer protocol infrastructure. Such trade offs were felt by the Forum to be defensible, but in no way detract from the fact that ATM is indeed a full fledged network layer protocol that is perhaps at least as complex as any that exists today.

What makes ATM a network layer protocol is indeed the very complexity of its addressing and routing protocols? This is independent of the fact that other network layer protocols are run over ATM -- indeed, as we will discuss later, the LAN Emulation protocols actually operate a MAC layer protocol over ATM, but this does not make ATM a physical layer.

A related issue that also causes confusion is the notion of "flat addressing" and whether or not ATM can be used to build a "simpler" network, in some sense, than today's network layer protocol based routed internetworks. This issue is coupled to the layering issue discussed above because some, as noted, draw a correspondence between ATM and layer 2 MAC protocols. As the latter do indeed have a flat address space, (i.e. a 48 bit MAC addresses) and it is true that MAC layer internetworking devices such as MAC bridges do offer "plug and play" capabilities, and do not require the complex configuration of layer 3 internetworking devices (that is, routers).

This simplicity comes from the fact that since MAC addresses are indeed flat, they have no logical hierarchy and packets must be flooded throughout the network, using bridging protocols. While this requires no network configuration, it also greatly reduces the scalability and stability of such bridged networks. A hierarchical address space, together with address assignment policies that minimize (flat) host routes, permit the use of address aggregation, where reachability for entire sets of end systems can be summarized by a single address prefix (or, equivalently, by subnet masks). Coupled with a routing protocol that disseminates such address prefixes, hierarchical addressing precludes the need for flooding, and greatly reduces the amount of reachability information that must be exchanged.

Protocols with hierarchical address spaces do indeed generally require more configuration for address and subnet assignment, but by the same token this very hierarchy permits the operation of routing protocols, and hence the deployment of much more scalable and stable networks. Flat addressing, by definition, precludes routing and requires bridging, with consequent lack of scalability.

Indeed, very few networks, outside of bridged LANs, actually have a truly flat address space. The telephone network for instance, is often thought of as a flat network, which actually incorporates a very structured hierarchy within its address space. That is, country code, area code, and so on), and it is only this rigid hierarchy that has permitted the telephone network to scale globally as it has. ATM networks certainly do not have a flat address space, but as discussed in the previous section, the ATM address space has scope for an unprecedented level of hierarchical structure, and this structure is exploited in the ATM routing protocols. We discuss below how to support greater degrees of scalability within ATM networks than is possible within any other network.

Much of the discussion about flat addressing and ATM actually revolves around the perception that ATM networks can be made easier to administer than existing layer 3 networks. It is true that, for historical reasons, few efforts were made in the development of many current network layer protocols to facilitate ease of administration, though many such efforts are being made today, for instance as with the **Dynamic Host Configuration Protocol (DHCP)**, in the case of IP. Ease of administration argues not for flat addressing, however, but for a systematic focus on supporting auto-configuration within protocols, as is now being done for the **IP Next Generation (IPng or IPv6)** protocol. This has been a prime focus for the ATM Forum from its inception, and by building on such mechanisms as the ILM1, most of the protocols developed for ATM do incorporate such support.

3.7 Summary

The emerging networked computing environments require scalability of bandwidth, isochronous capabilities, and lower cost to provide better performance from legacy applications as well as a suitable environment for the future. The day-to-day administration of the network must be adapted to an expanding environment that is changing in both size and diversity. Therefore, network management must improve and become simplified so that it is both affordable and usable by any organization.

New ways of working are already stretching networks to their limits. Although extensions of shared-media LANs and switched Ethernet and Token Ring offer some tactical relief, *only ATM offers a fundamentally different, strategic solution to the problem.* ATM moves away from frame-based, shared-media LANs and multiprotocol enterprise networks toward a cell-based, switched environment that blends the LAN and WAN in a seamless, end-to-end protocol.

Although we recognize that all of the LAN environments will coexist and flourish for many years, we believe that ATM's inherent characteristics make it a superior choice for all application environments. Its scalability of bandwidth and lower cost of operation is attractive attributes that make ATM the optimum choice for backbones and server access. ATM's ability to support isochronous traffic, to set negotiated qualities of service, and to provide predictable connections make it the best environment for new desktop applications. We will go on to highlight the spectrum of benefits and attributes of ATM in order to help understand how the other networks based on other technologies such as IP can be integrated effectively in enterprise networks for greater flexibility and scalability.

This chapter covered the research material on architectural and internetworking features with regard to the diverse models available to meet varying QoS requirements, associated signaling requirements and implementations with diverse switching architectures available to achieve varying end objectives. In light of these factors, we shall now present our research finding on the architectural and internetworking aspects of the Internet Protocol (IP) in the next chapter.

4. Research Findings – Part II

Internet Protocol (IP) – Architecture and Internetworking

IP is a network-layer protocol supporting both point-to-point(unicast) and multipoint-to-multipoint (multicast and broadcast) data communication services. Unicast, direct communications between terminals is allowed within a logical IP subnetwork, universally identified by the protocol addressing scheme and deployed over a single data link network based on either a single hardware technology or a set of bridged networks with a single or multiple hardware technologies. The communications between terminals belonging to different IP logical subnetworks may take place only through IP nodes or routers, which operate as switching elements at the IP layer. Routers are the gateways between multiple IP subnetworks and are used to switch variable-length IP packets. IP offers a unique service class that is best effort. This means each IP packet is treated the same way and has the same possibility as all the others to reach the destination. Packets are discarded only in case of congestion.

Further, with an exponential growth of the Internet and associated IP networks', routing has become one of the fundamentals of the current Internet. Dynamic distributed routing protocols have given the Internet a high degree of reliability and flexibility. However, new ambitions to carry time-sensitive and broadband traffic mean that the current routing model is facing new challenges. First, the amazing growth rates of the Internet have translated into a proportional increase in the number of routers. As a result, route computations through this enlarging flat topology map become more complex. This scalability issue can be resolved by introducing more hierarchy in the network. In hierarchical networks, information is aggregated at the cost of some routing accuracy. The challenge is to design efficient algorithms for routing and topology information condensation in order to minimize any loss of accuracy. Second, routing of traffic with **Quality of Service (QoS)** requirements, or the combination of shortest-path computations with additional policy or cost constraints, require routing algorithms that take more than one parameter (e.g., hop count) into account. Multiparameter routing is not only computationally more complex, but also requires protocols that are able to perform more dynamic monitoring of network resources (e.g., link load, available bandwidth, latency), while keeping the inserted overhead traffic under control. Third, the exploding number of users combined with the increasing number of multimedia applications is significantly pumping up overall traffic volume. Forwarding performance needs to keep pace with these growing bandwidth requirements. A way to relax this requirement is to forward more layer 3 traffic directly at layer 2, thus shifting to a more connection-oriented hardware-driven forwarding mode.

In order to understand the implications of the above, how an IP network functions to meet its end objective of best effort delivery is prime. Therefore, this chapter unfolds our research findings on the IP architecture and the related protocols and their interactions in the Internet cloud to make it a functionally whole and seemingly one, big transparent network.

4.1 Architecture and Protocols

The TCP/IP protocol suite is named for two of its most important protocols: **Transmission Control Protocol (TCP)** and **Internet Protocol (IP)**. Another name for it is the Internet Protocol Suite, and this is the phrase used in official Internet standards documents. We shall use the more common term TCP/IP to refer to the entire protocol suite.

4.1.1 Architectural Model

4.1.1.1 Internetworking ⁶⁷

The first design goal of TCP/IP was to build an interconnection of networks that provided universal communication services: an *internetwork*, or *Internet*. Each physical network has its own technology-dependent communication interface, in the form of a programming interface that provides basic communication functions (primitives). Communication services are provided by software that runs between the physical network and the user applications and that provides a common interface for these applications, independent of the underlying physical network. The architecture of the physical networks is hidden from the user.

The second aim is to *interconnect* different physical networks to form what appears to the user to be one large network. Such a set of interconnected networks is called an *internetwork* or an *Internet*. To be able to interconnect two networks, we need a computer that is attached to both networks and that can forward packets from one network to the other; such a machine is called a *router*. The term **IP router** is also used because the routing function is part of the IP layer of the TCP/IP protocol suite.

The basic properties of a router are:

- From the network standpoint, a router is a normal host.
- From the user standpoint, routers are invisible. The user sees only one large internetwork.

To be able to identify a host on the internetwork, each host is assigned an address, the **IP address**. When a host has multiple network adapters, each adapter has a separate IP address.

The IP address consists of two parts: **<network number><host number>**

The *network number* part of the IP address is assigned by a central authority and is unique throughout the Internet. The authority for assigning the *host numbers* part of the IP address resides with the organization, which controls the network, identified by the network number.

4.1.1.2 Internet Architecture ⁶⁸

The TCP/IP protocol suite has evolved over a time period of some 25 years and the most important aspects of the protocol suite is discussed herein to better understand the TCP/IP suite and its applicability to internetworking.

I. Layered Protocols

TCP/IP, like most networking software, is modeled in layers. This layered representation leads to the term *protocol stack*, which is synonymous with protocol suite. It can be drawn in parallel to others, such as **Systems Network Architecture (SNA)** and **Open System Interconnection (OSI)** layering. Functional comparisons cannot easily be extracted from this, as there are basic differences in the layered models used by the different protocol suites. The Internet protocols are modeled in four layers as shown in figure 1 and they are,

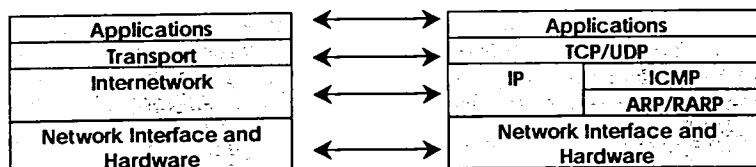


Figure 30: Architectural Model

⁶⁷ Dr. Sidnie Feit, **TCP/IP - Architecture, Protocols & Implementation**, McGraw-Hill, 1997

⁶⁸ **TCP/IP Networking**. <http://www.lmu.edu/admin/IS/training/protected/tcpip.html>

Application is a user process cooperating with another process on the same or a different host. Examples are TELNET (a protocol for remote terminal connections), **FTP (File Transfer Protocol)** and **SMTP (Simple Mail Transfer Protocol)**. **Transport** provides the end-to-end data transfer. Example protocols are **TCP (connection-oriented)** and **UDP (connectionless)**.

Internetwork also called the *Internet layer* or the *network layer*, the internetwork provides the "virtual network" image of Internet (that is, this layer shields the higher levels from the typical network architecture below it). Internet Protocol (IP) is the most important protocol in this layer. It is a *connectionless* protocol, which doesn't assume reliability from the lower layers. IP does not provide reliability, flow control or error recovery. These functions must be provided at a higher level, either at the Transport layer by using TCP as the transport protocol, or at the Application layer if UDP is used as the transport protocol. A message unit in an IP network is called an *IP datagram*. This is the basic unit of information transmitted across TCP/IP networks.

Network Interface also called the *link layer* or the *data-link layer*, the network interface layer is the interface to the actual network hardware. This interface may or may not provide reliable delivery, and may be packet or stream oriented. In fact, TCP/IP does not specify any protocol here, but can use almost any network interface available, which illustrates the flexibility of the IP layer. Examples are IEEE 802.2, X.25, ATM, FDDI, Packet Radio Networks and even SNA. The arrows in figure 30 show the actual interactions between the layers, but a more detailed "layering model" is shown in figure 31 below.

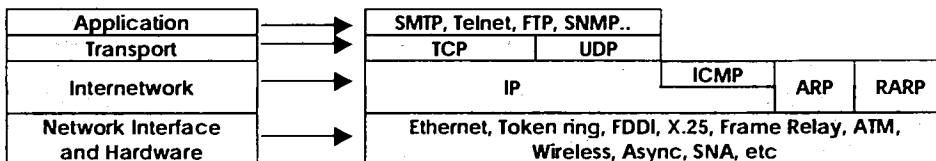


Figure 31: Detailed Architectural Model

II. Bridges, Routers and Gateways ⁶⁹

Forming an internetwork by interconnecting multiple networks is done by *routers*. It is important to distinguish between a router, a bridge and a gateway.

Bridge interconnects LAN segments at the Network Interface layer level and forwards frames between them. A bridge performs the function of a **Media Access Control (MAC)** relay, and is independent of any higher layer protocol (including the Logical Link protocol). It provides MAC layer protocol conversion, if required.

A bridge can be said to be *transparent* to IP. That is, when a host sends an IP datagram to another host on a network connected by a bridge, it sends the datagram directly to the host and the datagram "crosses" the bridge without the sending host being aware of it.

Router Interconnects networks at the internetwork layer level and routes packets between them. The router must understand the addressing structure associated with the networking protocols it supports and take decisions on whether, or how, to forward packets. Routers are able to select the best transmission paths and optimal packet sizes. The basic routing function is implemented in the IP layer of the TCP/IP protocol stack. Therefore any host or workstation running TCP/IP may be used as a router. However, dedicated routers provide much more sophisticated routing than the minimum function implemented by IP. Because IP provides this basic routing function, the term "IP router", is often used. Other, older, terms for router are "IP gateway", "Internet gateway" and "gateway". The term *gateway* is now normally used for connections at a higher level than the router level.

⁶⁹ James Martin, TCP/IP Networking, Prentice Hall, 1994

A router can be said to be *visible* to IP. That is, when a host sends an IP datagram to another host on a network connected by a router, it sends the datagram to the router and not directly to the target host.

Gateway interconnects networks at higher levels than bridges or routers. A gateway usually supports address mapping from one network to another, and may also provide transformation of the data between the environments to support end-to-end application connectivity. Gateways typically limit the interconnectivity of two networks to a subset of the application protocols supported on either one. The term "gateway", when used in this sense is *not* synonymous with "IP gateway".

A gateway can be said to be *opaque* to IP. That is, a host cannot send an IP datagram through a gateway: it can only send it to a gateway. The higher-level protocol information carried by the datagrams is then passed on by the gateway using whatever networking architecture is used on the other side of the gateway.

Closely related to routers and gateways is the concept of a *firewall* or *firewall gateway* which is used to restrict access from the Internet to a network or a group of networks controlled by an organization for security reasons.

III. IP Routing

Incoming datagrams will be checked to see if the local host is the IP destination host. If "yes" then the datagram is passed to the higher-level protocols, else if "no" then the datagram is for a different host. The action depends on the value of the *ipforwarding* flag. If "true" then the datagram is treated as an outgoing datagram and is routed to the *next hop* according to the algorithm described below; else if "false" then the datagram is discarded.

In the internet protocol, outgoing IP datagrams pass through the *IP routing* algorithm, which determines where to send the datagram according to the destination IP address.

1. If the host has an entry in its *IP routing table*, which matches the destination IP address, the datagram is sent to the address in the entry.
2. If the network number of the destination IP address is the same as the network number for one of the host's network adapters (that is, the destination and the host are on the same network) the datagram is sent to the physical address of the host matching the destination IP address.
3. Otherwise, the datagram is sent to a *default router*.

This base algorithm, needed on all IP implementations, is sufficient to perform the base routing function.

As noted above, a TCP/IP host has basic router functionality included in the IP protocol. Such a router is adequate for simple routing, but not for complex networks. The protocols needed in complex cases are described Later. The IP routing mechanism combined with the "layered" view of the TCP/IP protocol stack is represented in figure 3 below. This shows an IP datagram, going from one IP address (network number X, host number A) to another (network number Y, host number B), through two physical networks. Note that at the intermediate router, only the lower part of the TCP/IP protocols stack (the internetwork and the network interface layers) are involved.

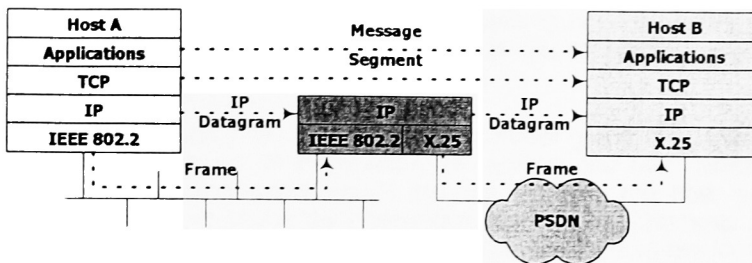


Figure 32: Internet Router - The router function is performed by the IP protocol.

4.1.2 Addressing ⁷⁰

Internet addresses can be symbolic or numeric. The symbolic form is easier to read, for example: ramesh@rit.edu. The numeric form is a 32-bit unsigned binary value, which is usually expressed, in a dotted decimal format. For example, 9.167.5.8 is a valid Internet address. The IP software uses the numeric form. The mapping between the two is done by the *Domain Name System* discussed in Domain Name System (DNS). We shall first look at the numeric form, which is called the IP address.

4.1.2.1 The IP Address ⁷¹

The standards for IP addresses are described in *RFC 1166 - Internet Numbers*. To be able to identify a host on the Internet, each host is assigned an address, the *IP address*, or *Internet Address*. When the host is attached to more than one network, it is called *multi-homed* and it has one IP address for each network interface. The IP address consists of a pair of numbers:

IP address = <network number><host number>. The *network number* part of the IP address is centrally administered by the Internet Network Information Center (the InterNIC) and is unique throughout the Internet.

IP addresses are used by the IP protocol to uniquely identify a host on the internet. IP datagrams (the basic data packets exchanged between hosts) are transmitted by some physical network attached to the host and each IP datagram contains a *source IP address* and a *destination IP address*. To send a datagram to a certain IP destination, the target IP address must be translated or mapped to a physical address. This may require transmissions on the network to find out the destination's physical network address (for example, on LANs the Address Resolution Protocol, discussed in **Address Resolution Protocol (ARP)**, is used to translate IP addresses to physical MAC addresses).

The first bits of the IP address specify how the rest of the address should be separated into its network and host part. The terms *network address* and *netID* are sometimes used instead of network number, but the formal term, used in RFC 1166, is network number. Similarly, the terms *host address* and *hostID* are sometimes used instead of host number. There are five classes of IP addresses. These are shown in figure 33 below.

Class A	0	Network ID	Host ID
Class B	10	Network ID	Host ID
Class C	110	Network ID	Host ID
Class D	1110	Multicast address	
Class E	1111	Reserved	

Figure 33: Assigned Classes of Internet Addresses

⁷⁰ K. Washburn, J.T. Evans, *TCP/IP - Running a successful Network*, Addison-Wesley, 1993

⁷¹ RFC 791 - Internet Protocol

Two numbers out of each of the class A, class B and class C network numbers, and two host numbers out of every network are pre-assigned: the "all bits 0" number and the "all bits 1" number. These are discussed below in Special IP Addresses.

- Class A addresses use 7 bits for the network number giving 126 possible networks (we shall see below that out of every group of network and host numbers, two have a special meaning). The remaining 24 bits are used for the host number, so each network can have up to $2^{24} - 2$ (16,777,214) hosts.
- Class B addresses use 14 bits for the network number, and 16 bits for the host number giving 16382 networks each with a maximum of 65534 hosts.
- Class C addresses use 21 bits for the network number and 8 for the host number giving 2,097,150 networks each with up to 254 hosts.
- Class D addresses are reserved for multicasting, which is used to address groups of hosts in a limited area.
- Class E addresses are reserved for future use.

It is clear that a class A address will only be assigned to networks with a huge number of hosts, and that class C addresses are suitable for networks with a small number of hosts. However, this means that medium-sized networks (those with more than 254 hosts or where there is an expectation that there may be more than 254 hosts in the future) must use Class B addresses. The number of small- to medium-sized networks has been growing very rapidly in the last few years and it was feared that, if this growth had been allowed to continue unabated, all of the available Class B network addresses would have been used by the mid-1990s. This is termed the IP Address Exhaustion Problem. The problem and how it is being addressed are discussed in the IP Address Exhaustion Problem.

One point to note about the split of an IP address into two parts is that this split also splits the responsibility for selecting the IP address into two parts. The network number is assigned by the InterNIC, and the host number by the authority which controls the network. As we shall see in the next section, the host number can be further subdivided: this division is controlled by the authority which owns the network, and *not* by the InterNIC.

4.1.2.2 Subnets ⁷²

Due to the explosive growth of the Internet, the use of assigned IP addresses became too inflexible to allow easy changes to local network configurations. These changes might occur when:

- A new physical network is installed at a location.
- Growth of the number of hosts requires splitting the local network into two or more separate networks.

To avoid having to request additional IP network addresses in these cases, the concept of *subnets* was introduced. The host number part of the IP address is sub-divided again into a network number and a host number. This second network is termed a *subnetwork* or *subnet*. The main network now consists of a number of subnets and the IP address is interpreted as:

<network number><subnet number><host number>

The combination of the subnet number and the host number is often termed the "local address" or the "local part". "Subnetting" is implemented in a way that is transparent to remote networks. A host within a network, which has subnets, is aware of the subnetting but a host in a different network is not; it still regards the local part of the IP address as a host number.

The division of the local part of the IP address into subnet number and host number parts can be chosen freely by the local administrator; any bits in the local part can be used to form the subnet accomplished. The division is done using a *subnet mask*, which is a 32-bit number. Zero bits in the

⁷² RFC 950 - IP Subnet extensions

subnet mask indicate bit positions ascribed to the host number, and ones indicate bit positions ascribed to the subnet number. The bit positions in the subnet mask belonging to the network number are set to ones but are not used. Subnet masks are usually written in dotted decimal form, like IP addresses.

The special treatment of "all bits zero" and "all bits one" applies to each of the three parts of a subnetted IP address just as it does to both parts of an IP address which has not been subnetted. For example, a subnetted Class B network, which has a 16-bit local part, could use one of the following schemes:

- ❶ The first byte is the subnet number, the second the host number. This gives us 254 (256 minus 2 with the values 0 and 255 being reserved) possible subnets, each having up to 254 hosts. The subnet mask is 255.255.255.0.
- ❷ The first 12 bits are used for the subnet number and the last four for the host number. This gives us 4094 possible subnets (4096 minus 2) but only 14 hosts per subnet (16 minus 2). The subnet mask is 255.255.255.240. There are many other possibilities.

While the administrator is completely free to assign the subnet part of the local address in any legal fashion, the objective is to assign a *number* of bits to the subnet number and the remainder to the local address. Therefore, it is normal to use a contiguous block of bits at the beginning of the local address part for the subnet number because this makes the addresses more readable (this is particularly true when the subnet occupies 8 or 16 bits). With this approach, either of the subnet masks above are "good" masks, but masks like 255.255.252.252 or 255.255.255.15 are not.

I. Types of Subnetting

There are two types of subnetting: static and variable length. Variable length is the more flexible of the two. Which type of subnetting is available depends upon the routing protocol being used; native IP routing supports only static subnetting, as does the widely used RIP protocol. However, RIP Version 2 supports variable length subnetting as well. These are reviewed in detail later on.

Static subnetting means that all subnets in the subnetted network use the same subnet mask. This is simple to implement and easy to maintain, but it implies wasted address space for small networks. For example, a network of four hosts that use a subnet mask of 255.255.255.0 wastes 250 IP addresses. It also makes the network more difficult to reorganize with a new subnet mask. Currently, almost every host and router supports static subnetting.

Variable length subnetting facilitates the use of different subnet masks by the subnets that make up the network. A small subnet with only a few hosts needs a subnet mask that accommodates only these few hosts. A subnet with many hosts attached may need a different subnet mask to accommodate the large number of hosts. The possibility to assign subnet masks according to the needs of the individual subnet will help conserve network addresses. Also, a subnet can be split into two parts by adding another bit to the subnet mask. Other subnets in the network are unaffected by the change. Not every host and router supports variable length subnetting.

Only networks need will determine the network size allocated and isolating networks with routers that support variable subnetting will solve routing problems. A host that does not support this kind of subnetting would have to route to a router that supports variable subnetting.

Mixing Static and Variable Length Subnetting . At first sight, it appears that the presence of a host which only supports static subnetting would prevent variable length subnetting from being used anywhere in the network. Fortunately this is not the case. Provided that the routers between subnets with different subnet masks are using variable length subnetting, the routing protocols employed are able to hide the difference between subnet masks from the hosts in a subnet. Hosts can continue to use basic IP routing and offload all of the complexities of the subnetting to dedicated routers.

II. IP Routing with Subnets ⁷³

Some implications of this algorithm (shown below) are:

- ☞ It is a change to the general IP algorithm. Therefore, to be able to operate this way, the particular gateway must contain the new algorithm. Some implementations may still use the general algorithm, and will not function within a subnetted network, although they can still communicate with hosts in other networks, which are subnetted.
- ☞ As IP routing is used in all of the hosts (and not just the routers), all of the hosts in the subnet must:
 1. Have an IP algorithm that supports subnetting.
 2. Have the same subnet mask (unless subnets are formed within the subnet).
- ☞ If the IP implementation on any of the hosts does not support subnetting, that host will be able to communicate with any host in its own subnet but not with any machine on another subnet within the same network. This is because the host sees only one IP network and its routing cannot differentiate between an IP datagram directed to a host on the local subnet and a datagram that should be sent via a router to a different subnet.

In case one or more hosts do not support subnetting, an alternative way to achieve the same goal exists in the form of *proxy-ARP*, which doesn't require any changes to the IP routing algorithm for single-homed hosts, but does require changes on routers between subnets in the network. This is explained in more detail in *Proxy-ARP* or *Transparent Subnetting*. To route an IP datagram on the network, the general IP routing algorithm as shown in figure is applied to each address.

III. Obtaining a Subnet Mask ⁷⁴

Usually, hosts will store the subnet mask to be used in a configuration file. However, sometimes this cannot be done, as for example in the case of a diskless workstation. The ICMP protocol includes two messages, address mask request and address mask reply, which allow hosts to obtain the correct subnet mask from a server.

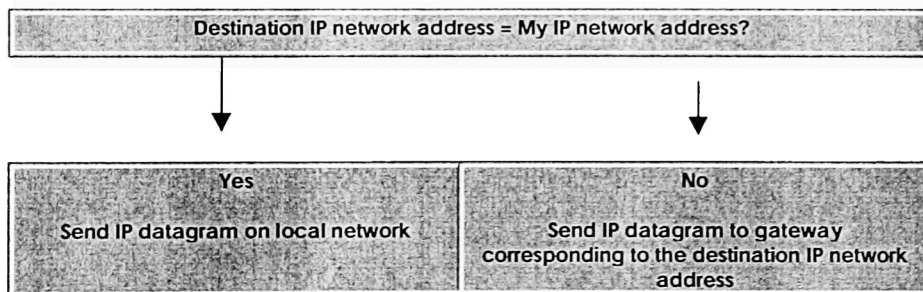


Figure 34: IP Routing without Subnets

⁷³ John T. Moy, *OSPF - Anatomy of an Internet Routing Protocol*, Addison-Wesley, 1998

⁷⁴ John T. Moy, *OSPF - Anatomy of an Internet Routing Protocol*, Addison-Wesley, 1998

To be able to differentiate between subnets, the IP routing algorithm changes and has the following form:

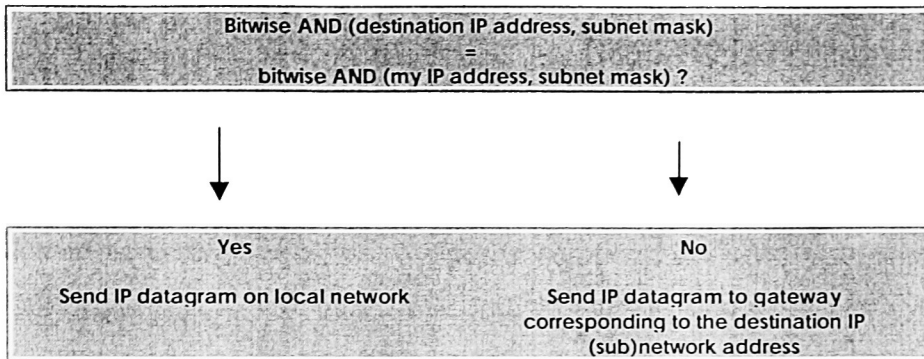


Figure 35: IP Routing with Subnets

IV. Addressing Routers and Multi-homed Hosts

Whenever a host has a physical connection to multiple networks or subnets, it is described as being *multi-homed*. All routers are multi-homed since their purpose is to join networks or subnets. A multi-homed host always has different IP addresses associated with each network adapter, since each adapter is in a different subnet or network. There is one apparent exception to this rule: with some systems (for example VM and MVS) it is possible to specify the same IP address for multiple point-to-point links (such as channel-to-channel adapters) if the routing protocol used is limited to the basic IP routing algorithm. For example, a VM "hypervisor" system running TCP/IP connected to a token-ring LAN. A very cost-effective solution for connecting these guests to the token-ring is to connect them to the hypervisor's TCP/IP with virtual channel-to-channel connections.

The IP addresses can be chosen so that the channel-to-channel connected systems constitute their own subnet, in which case the hypervisor acts as a router. Because the routing protocols available on VM and MVS only support static subnetting, it may be difficult to find an additional subnet number if the address space is constrained. Therefore, the channel-to-channel connected systems may be given IP addresses in the same subnet as the token ring, in which case the hypervisor is taking the place of a bridge, and is *not* multi-homed. One disadvantage of this configuration is that other hosts on the LAN need static definitions to route datagrams for the hosts on the far side of the "bridge" via the hypervisor because the controller is not aware that it has this bridging responsibility.

4.1.2.3 Special IP Addresses

As noted above, any component of an IP addresses with a value "all bits 0" or all "all bits 1" has a special meaning.

all bits 0 : stands for "this": "this" host (IP address with <host number>=0) or "this" network (IP address with <network number>=0) and is only used when the real value is not known. This form is only used in source addresses when the host is trying to determine its IP addresses from a remote server. The host may know include its host number if known, but not its subnet or network number..

all bits 1 : stands for "all": "all" networks or "all" hosts. For example, 128.2.255.255 (a class B address with a host number of 255.255) means all hosts on network 128.2. These are used in broadcast messages, as described below.

There is another address of special importance: the "all bits 1" class A network number 127 is reserved for the *loopback address*. Anything sent to an address with 127 as the value of the high

order byte, for example 127.0.0.1, must not be routed via a network but must be routed directly from the IP implementation's output driver to its input driver.

4.1.2.4 Unicasting, Broadcasting and Multicasting ⁷⁵

The majority of IP addresses refer to a single recipient: these are called *unicast* addresses. However, as noted above, there are two special types of IP address, which are used for addressing multiple recipients: broadcast addresses and multicast addresses. These addresses are used for sending messages to multiple recipients. Any protocol, which is connectionless, may send broadcast or multicast messages as well as unicast messages. A protocol, which is connection-oriented, can only use unicast addresses because the connection exists between a specific pair of hosts.

I. Broadcasting

There are a number of addresses, which are used for IP broadcasting: all use the convention that "all-bits 1" indicates "all". Broadcast addresses are never valid as source addresses, only as destination addresses. The different types of broadcast addresses are listed here:

limited broadcast address: The address 255.255.255.255 (all bits 1 in all parts of the IP address) is used on networks which support broadcasting, such as token rings, and it refers to all hosts on the subnet. It does not require the host to know any IP configuration information at all. All hosts on the local network will recognize the address, but routers will never forward it.

There is one exception to this rule, called *BOOTP forwarding*. The BOOTP protocol uses the limited broadcast address to allow a diskless workstation to contact a boot server. BOOTP forwarding is a configuration option available on some routers. Without this facility, a separate BOOTP server would be required on each subnet. However, this is not simple forwarding because the router also plays a part in the BOOTP protocol.

network-directed broadcast address: If the network number is a valid network number, the network is not subnetted and the host number is all ones (for example, 128.2.255.255), then the address refers to all hosts on the specified network. Routers should forward these broadcast messages unless configured otherwise. This is used in ARP requests on unsubnetted networks.

subnet-directed broadcast address : If the network number is a valid network number, the subnet number is a valid subnet number and the host number is all ones, then the address refers to all hosts on the specified subnet. Since the sender's subnet and the target subnet may have different subnet mask, the sender must somehow find out the subnet mask in use at the target. The actual broadcast is performed by the router, which receives the datagram into the subnet.

all-subnets-directed broadcast address : If the network number is a valid network number, the network is subnetted and the local part is all ones (for example, 128.2.255.255), then the address refers to all hosts on all subnets in the specified network. In principal routers may propagate broadcasts for all subnets but are not required to do so. In practice, they do not; there are few circumstances where such a broadcast would be desirable, and it can lead to problems, particularly if a host has been incorrectly configured with no subnet mask. Consider the wasted resource involved if a host 9.180.214.114 in the subnetted Class A network 9 thought that it was not subnetted and used 9.255.255.255 as a "local" broadcast address instead of 9.180.214.255 and all of the routers in the network respected the request to forward the request to all clients.

If routers do respect all-subnets-directed broadcast address they use an algorithm called *Reverse Path Forwarding* to prevent the broadcast messages from multiplying out of control.

⁷⁵ RFC 919 -IP Broadcast Datagrams

II. Multicasting ⁷⁶

Broadcasting has a major disadvantage: its lack of selectivity. If an IP datagram is broadcast to a subnet, every host on the subnet will receive it, and have to process it to determine whether the target protocol is active. If it is not, the IP datagram is discarded. Multicasting avoids this overhead by using groups of IP addresses. A 28-bit number represents each group, which is included in a Class D address. Recall that a class D address has the format:

Class D	1110	<i>xxxxxx</i>	Multicast address	<i>xxxxxx</i>
---------	-------------	---------------	--------------------------	---------------

So *multicast group addresses* are IP addresses in the range 224.0.0.0 to 239.255.255.255. For each multicast address there is a set of zero or more hosts which are listening to it. This set is called the *host group*. There is no requirement for any host to be a member of a group to send to that group. There are two kinds of host group:

permanent : The IP address is permanently assigned by IANA. The membership of a host group is not permanent: a host may leave or join the group at will. The list of IP addresses assigned permanent host groups is included in *STD 2 - Assigned Internet Numbers*.

Important ones are:

- 224.0.0.0 Reserved base address
- 224.0.0.1 All systems on this subnet
- 224.0.0.2 All routers on this subnet

Some other examples used by the OSPF routing protocol

- 224.0.0.5 All OSPF routers
- 224.0.0.6 OSPF Designated Routers

An application may also retrieve a permanent host group's IP address from the domain name system using the domain *mcast.net*, or determine the permanent group from an address by using a pointer query in the domain *224.in-addr.arpa*. A permanent group exists even if it has no members.

transient : Any group, which is not permanent, is transient and is available for dynamic assignment as needed. Transient groups cease to exist when their membership drops to zero.

Multicasting on a single physical network, which supports the use of multicasting, is simple. To join a group, a process running on a host must somehow inform its network device drivers that it wishes to be a member of the specified group. The device driver software itself must map the multicast address to a physical multicast address and enable the reception of packets for that address. The device driver must also ensure that the receiving process does not receive any spurious datagrams by checking the destination address in the IP header before passing it to the IP layer.

Despite this requirement for software filtering of multicast packets, multicasting still causes much less overhead for hosts that are not interested. In particular, those hosts that are not in any host group are not listening to any multicast addresses and all multicast messages are filtered by the network interface hardware.

Multicasting is not limited to a single physical network. There are two aspects to multicasting across physical networks:

⁷⁶ RFC 1112 - Internet Group Multicast Protocol

- A mechanism for deciding how widespread the multicast is (remember that unlike unicast addresses and broadcast addresses) multicast addresses cover the entire Internet.
- A mechanism for deciding whether a multicast datagram needs to be forwarded to a particular network.

The first problem is easily solved: the multicast datagram has a *Time To Live (TTL)* value like every other, which is decremented with each hop to a new network. When the Time to Live field is decremented to zero, the datagram can go no further. The mechanism for deciding whether a router should forward a multicast datagram is called *Internet Group Management Protocol (IGMP) or Internet Group Multicast Protocol*. IGMP and multicasting are defined in *RFC 1112 - Host extensions for IP multicasting*.

4.1.2.5 The IP Address Exhaustion Problem

The number of networks on the Internet has been approximately doubling annually for a number of years. However, the usage of the Class A, B and C networks differs greatly: nearly all of the new networks assigned in the late 1980s were Class B, and in 1990 it became apparent that if this trend continued, the last Class B network number would be assigned during 1994. On the other hand, Class C networks were hardly being used.

The reason for this trend was that most potential users found a Class B network to be large enough for their anticipated needs, since it accommodates up to 65534 hosts, whereas a class C network, with a maximum of 254 hosts, severely restricts the potential growth of even a small initial network. Furthermore, most of the class B networks being assigned were small ones. There are relatively few networks that would need as many as 65,534 host addresses, but very few for which 254 hosts would be an adequate limit. In summary, although the Class A, Class B and Class C divisions of the IP address are logical and easy to use (because they occur on byte boundaries), with hindsight they are not the most practical because Class C networks are too small to be useful for most organizations while Class B networks are too large to be densely populated by any but the largest organizations.

4.1.2.6 Private Internets ⁷⁷

Another approach to conservation of the IP address space is described in *RFC 1597 - Address Allocation for Private Internets*. Briefly, it relaxes the rule that IP addresses are globally unique by reserving part of the address space for networks which are used exclusively within a single organization and which do not require IP connectivity to the Internet. There are three ranges of addresses, which have been reserved by IANA for this purpose:

- 10 A single Class A network
- 172.16 through 172.31 16 contiguous Class B networks
- 192.168.0 through 192.168.255 256 contiguous Class C networks

Any organization may use any addresses in these ranges without reference to any other organization. However, because these addresses are not globally unique, hosts in another organization cannot reference them and they are not defined to any external routers. Routers in networks not using private addresses, particularly those operated by Internet service providers, are expected to quietly discard all routing information regarding these addresses. Routers in an organization using private addresses are expected to limit all references to private addresses to internal links; they should neither advertise routes to private addresses to external routers nor forward IP datagrams containing private addresses to via external routers. Hosts having only a private IP address do not have IP-layer connectivity to the Internet. This may be desirable and may even be a reason for using private addressing. All connectivity to external Internet hosts must be provided with application gateways.

⁷⁷ *Intranets and Virtual Private Networks*, <http://www.riscpa.org/seminar/intvnpn.htm>

4.1.2.7 Classless Inter-Domain Routing (CIDR) ⁷⁸

There is a major problem with the use of a range of Class C addresses instead of a single Class B addresses: each network must be routed separately. Standard IP routing understands only the class A, B and C network classes. Within each of these types of network, subnetting can be used to provide better granularity of the address space within each network, but there is no way to specify that multiple class C networks are actually related. The result of this is termed the *routing table explosion* problem: a Class B network of 3000 hosts requires one routing table entry at each backbone router, but if the same network is addressed as a range of Class C networks, it requires 16 entries.

The solution to this problem is a scheme called *Classless Inter-Domain Routing (CIDR)*. CIDR is a *proposed standard protocol* with a status of *elective*. CIDR does not route according to the class of the network number (hence the term *classless*) but solely according to the high order bits of the IP address which are termed the *IP prefix*. Each CIDR routing entry contains a 32-bit IP address and a 32-bit network mask, which together give the length and value of the IP prefix. This can be represented as <IP_address network_mask>. For example <194.0.0.0 254.0.0.0> represents the 7 bit IP prefix B'1100001'.

CIDR handles the routing for a group of networks with a common prefix with a single routing entry. This is the reason why multiple Class C network numbers assigned to a single organization have a common prefix. This process of combining multiple networks into a single entry is termed *address aggregation* or *address summarization*. It is also called *supernetting* because routing is based upon network masks, which are shorter than the *natural* network mask of the IP address, in contrast to *subnetting* where the network masks are longer than the natural mask.

Unlike subnet masks, which are normally contiguous but may have a discontinuous local part, supernet masks are *always* contiguous. If IP addresses are represented with a tree showing the routing topology, with each leaf of the tree representing a group of networks which are considered as a single unit (called a *routing domain*) and the IP addressing scheme is chosen so that each fork in this tree corresponds to an increase in the length of the IP prefix, then CIDR allows address aggregation to be performed very efficiently.

For example, if a router in North America routes all European traffic via a single link, then a single routing entry for <194.0.0.0 254.0.0.0> includes the group of Class C network addresses assigned to Europe as described above. This single entry takes the place of all the entries for all of the assigned network numbers in this range, which is a possible maximum of 2^7 (superscript 7), or 131,072 numbers. At the European end of this link, there are routing entries with longer prefixes, which map to the European network topology but this routing information is not needed at the American end of the link.

CIDR uses a *longest match is best* approach, so if the router in the US needs to make an exception for one range of addresses, such as the 64 network range <195.1.64.0 255.555.192.0> it needs just one additional entry, since this entry overrides the more general (shorter) one for those networks it contains. It is apparent from this example that as the usage of the IP address space increases, particularly the Class C address space, the benefits of CIDR increase as well, provided that the assignment of addresses follows the network topology. The existing state of the IP address space does not follow such a scheme since it pre-dates the development of CIDR.

However, new Class C addresses are being assigned in such a way as to enable CIDR, and this should have the effect of alleviating the routing table explosion problem in the near term. In the longer term, a restructuring of the IP address space along topological lines may be necessary. This would involve the re-numbering of a large number of networks, implying an enormous amount of implementation effort, and so would be a gradual process.

⁷⁸ RFC 1517 - *Applicability Statement for the Implementation of Classless Inter-Domain Routing (CIDR)*

It is an over-simplification to assume that routing topology can be represented as a simple tree; although most routing domains have a single attachment which provides access to the rest of the Internet, there are also many domains which have multiple attachments. Routing domains of these two types are called *single-homed* and *multi-homed*. Furthermore, the topology is not static. Not only are new organizations joining the Internet at an ever-increasing rate, but existing organizations may change places within the topology, for example if they change between service providers for commercial or other reasons. Although such cases complicate the practical implementation of CIDR-based routing and reduce the efficiency of address aggregation that can be achieved, they do not invalidate the approach.

4.1.3 Internet Protocol (IP) ⁷⁹

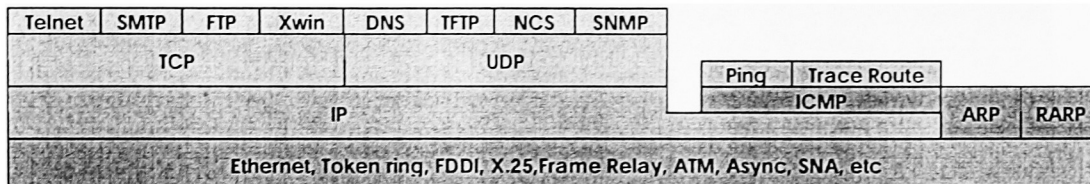


Figure 36: Internet Protocol (IP)

IP is a *standard protocol* with STD number 5 which also includes ICMP and IGMP. Its status is *required*. The current IP specification can be found in RFCs 791, 950, 919 and 922, with updates in RFC 1349. IP is the protocol that hides the underlying physical network by creating a *virtual network* view. It is an unreliable, best-effort, connectionless packet delivery protocol.

It adds no reliability, flow control or error recovery to the underlying network interface protocol. Packets (*datagrams*) sent by IP may be lost, out of order, or even duplicated, and IP will not handle these situations. It is up to higher layers to provide these facilities. IP also assumes little from the underlying network mechanisms, only that the datagrams will "probably" (best-effort) be transported to the addressed host.

4.1.3.1 IP Datagram ⁸⁰

The *Internet datagram (IP datagram)* is the base transfer packet in the Internet protocol suite. It has a header containing information for IP, and data that is relevant only to the higher level protocols.

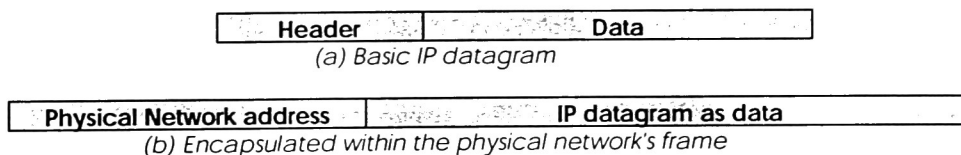


Figure 37: Base IP Datagram

The IP datagram is encapsulated in the underlying network's frame, which usually has a maximum length or frame limitation, depending on the hardware used. For Ethernet, this will typically be 1500 bytes. Instead of limiting the IP datagram length to some maximum size, IP can deal with **fragmentation and re-assembly** of its datagrams. In particular, the IP standard does not impose a maximum size, but states that all subnetworks should be able to handle datagrams of at least 576 bytes. Fragments of a datagram all have a header, basically copied from the original datagram, and data following it. They are treated as normal IP datagrams while being transported to their destination. Note, however, that if one of the fragments gets lost, the complete datagram is

⁷⁹ Floyd Wilder, *A guide to the TCP/IP Protocol suite*. Artech House, 1993

⁸⁰ RFC 791 - Internet Protocol

considered lost since IP does not provide any acknowledgment mechanism, so the remaining fragments will simply be discarded by the destination host.

I. IP Datagram Format

The IP datagram header is a minimum of 20 bytes long:

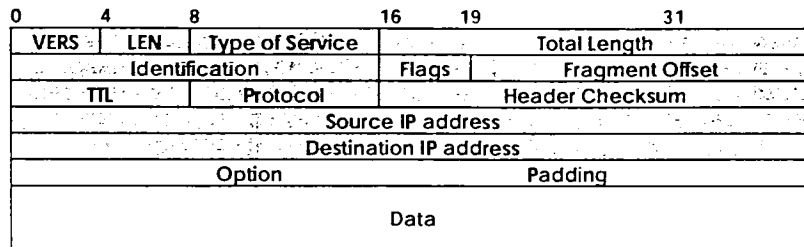


Figure 38: IP Datagram Format

Where:

VERS: The version of the IP protocol. The current version is 4. 5 is experimental and 6 is IPng.

LEN: The length of the IP header counted in 32-bit quantities. This does not include the data field.

Type of Service: The type of service is an indication of the quality of service requested for this IP datagram.

Total Length: The total length of the datagram, header and data, specified in bytes.

Identification: A unique number assigned by the sender to aid in reassembling a fragmented datagram. Fragments of a datagram will have the same identification number.

Flags: Various control flags.

Fragment Offset: Used with fragmented datagrams, to aid in reassembly of the full datagram. The value is the number of 64-bit pieces (header bytes are not counted) that are contained in earlier fragments. In the first (or only) fragment, this value is always zero.

Time to Live: Specifies the time (in seconds) this datagram is allowed to travel. Each router where this datagram passes is supposed to subtract from this field its processing time for this datagram. Actually a router is able to process a datagram in less than 1 second; thus it will subtract one from this field, and the TTL becomes a hop-count metric rather than a time metric. When the value reaches zero, it is assumed that this datagram has been traveling in a closed loop and it is discarded. The initial value should be set by the higher-level protocol, which creates the datagram.

Protocol Number: Indicates the higher-level protocol to which IP should deliver the data in this datagram. Some important values are:

0	Reserved
1	Internet Control Message Protocol (ICMP)
2	Internet Group Management Protocol (IGMP)
3	Gateway-to-Gateway Protocol (GGP)
4	IP (IP encapsulation)
5	Stream
6	Transmission Control (TCP)
8	Exterior Gateway Protocol (EGP)
9	Private Interior Routing Protocol
17	User Datagram (UDP)
89	Open Shortest Path First

The full list can be found in *STD 2 - Assigned Internet Numbers*.

Header Checksum: Is a checksum on the header only. It does not include the data. The checksum is calculated as the 16-bit one's complement of the one's complement sum of all 16-bit words in the header. For the purpose of this calculation, the checksum field is assumed to be zero. If the

header checksum does not match the contents, the datagram is discarded because at least one bit in the header is corrupt, and the datagram may even have arrived at the wrong destination.

Source IP Address: The 32-bit IP address of the host sending this datagram.

Destination IP Address: The 32-bit IP address of the destination host for this datagram.

Options: Variable length. An IP implementation is not required to be capable of generating options in the datagrams it creates, but all IP implementations are required to be able to process datagrams containing options. The Options field is variable in length. There may be zero or more options.

Length: counts the length (in bytes) of the option, including the type and length fields.

option data: contains data relevant to the option.

Padding: If an option is used, the datagram is padded with all-zero bytes up to the next 32-bit boundary.

Data: The data contained in the datagram is passed to a higher-level protocol, as specified in the *protocol* field.

II. Fragmentation

When an IP datagram travels from one host to another, it can cross different physical networks. Physical networks have a maximum frame size, called the **Maximum Transmission Unit (MTU)**, which limits the length of a datagram that can be placed in one physical frame. Therefore, a scheme has been put in place to fragment long IP datagrams into smaller ones, and to reassemble them at the destination host. IP requires that each link has an MTU of at least 68 bytes, so if any network provides a lower value than this, fragmentation and re-assembly must be implemented in the network interface layer in a way that is transparent to IP. 68 is the sum of the maximum IP header length of 60 bytes and the minimum possible length of data in a non-final fragment (8 bytes). IP implementations are not required to handle unfragmented datagrams larger than 576 bytes, but most implementations will handle larger values, typically slightly more than 8192 bytes or higher, and rarely less than 1500.

An unfragmented datagram has all-zero fragmentation information. That is, the more fragments flag bit is zero and the fragment offset is zero. When fragmentation is to be done, the following steps are performed:

- ✎ The DF flag bit is checked to see if fragmentation is allowed. If the bit is set, the datagram will be discarded and an error will be returned to the originator using ICMP.
- ✎ Based on the MTU value, the data field is split into two or more parts. All newly created data portions must have a length, which is a multiple of 8 bytes, with the exception of the last data portion.
- ✎ All data portions are placed in IP datagrams. The header of these datagrams are copies of the original one, with some modifications:
 - ↻ The more fragments flag bit is set in all fragments except the last.
 - ↻ The fragment offset field in each is set to the location this data portion occupied in the original datagram, relative to the beginning of the original unfragmented datagram. The offset is measured in 8-byte units.
 - ↻ If options were included in the original datagram, the high order bit of the option type byte determines whether or not they will be copied to all fragment datagrams or just to the first one. For instance, source route options have to be copied in all fragments and therefore they have this bit set.
 - ↻ The header length field of the new datagram is set.
 - ↻ The total length field of the new datagram is set.
 - ↻ The header checksum field is re-calculated.
- ✎ Each of these fragmented datagrams is now forwarded as a normal IP datagram. IP handles each fragment independently, that is, the fragments may traverse different routers to the intended destination, and they may be subject to further fragmentation if they pass through networks that have smaller MTUs.

At the destination host, the data has to be reassembled into one datagram. The identification field of the datagram was set by the sending host to a unique number (for the source host, within the limits imposed by the use of a 16-bit number). As fragmentation doesn't alter this field, incoming fragments at the receiving side can be identified, if this ID field is used together with the Source and Destination IP addresses in the datagram. The Protocol field is also checked for this identification.

In order to reassemble the fragments, the receiving host allocates a buffer in storage as soon as the first fragment arrives. A timer routine is then started. When the timer timeouts and not all of the fragments have been received, the datagram is discarded. The initial value of this timer is called the IP datagram **time-to-live (TTL)** value. It is implementation dependent, and some implementations allow it to be configured; for example AIX Version 3.2 provides an *ipfragttl* option with a default value of 60 seconds.

When subsequent fragments of the datagram arrive, before the timer expires, the data is simply copied into the buffer storage, at the location indicated by the fragment offset field. As soon as all fragments have arrived, the complete original unfragmented datagram is restored, and processing continues, just as for unfragmented datagrams. Note: IP does not provide the reassembly timer. It will treat each and every datagram, fragmented or not, the same way, that is, as individual messages. It is up to the higher layer to implement a timeout and to look after any missing fragments. The higher layer could be TCP for a connection-oriented transport network or the application for connectionless transport networks based upon UDP and IP. The netstat command may be used on some TCP/IP hosts to list details of fragmentation that is occurring.

II. IP Datagram Routing Options ⁸¹

The IP datagram Options field allows two methods for the originator of an IP datagram to explicitly provide routing information and one for an IP datagram to determine the route that it travels.

Loose Source Routing option, also called the **Loose Source and Record Route (LSRR)** option, provides a means for the source of an IP datagram to supply explicit routing information to be used by the routers in forwarding the datagram to the destination, and to record the route followed.

Strict Source Routing option, also called the **Strict Source and Record Route (SSRR)** option, uses the same principle as loose source routing except that the intermediate router *must* send the datagram to the next IP address in the source route via a directly connected network and not via an intermediate router. If it cannot do so it reports an error with an ICMP Destination Unreachable message.

III. Internet Timestamp

A timestamp is an option forcing some (or all) of the routers on the route to the destination to put a timestamp in the option data. The timestamps are measured in seconds and can be used for debugging purposes.

4.1.3.2 IP Routing

An important function of the IP layer is *IP routing*. It provides the basic mechanism for routers to interconnect different physical networks. This means that an internet host can function as a normal host and a router simultaneously.

⁸¹ John T. Moy, *OSPF - Anatomy of an Internet Routing Protocol*, Addison-Wesley, 1998

A basic router of this type is referred to as a *router with partial routing information*, because the router only has information about four kinds of destination:

- Hosts which are directly attached to one of the physical networks to which the router is attached
- Hosts or networks for which the router has been given explicit definitions
- Hosts or networks for which the router has received an ICMP redirect message
- A default destination for everything else

The last two items allow a basic router to begin with a very limited amount of information and to increase its information because a more sophisticated router will issue an ICMP redirect message if it receives a datagram and it knows of a better router on the same network for the sender to use. This process is repeated each time a basic router of this type is restarted.

Additional protocols are needed to implement a full-function router that can exchange information with other routers in remote network. Such routers are essential except in small networks.

I. Direct and Indirect Destinations

If the destination host is attached to a network to which the source host is also attached, an IP datagram can be sent directly, simply by encapsulating the IP datagram in the physical network frame. This is called *direct routing*. *Indirect routing* occurs when the destination host is not on a network directly attached to the source host. The only way to reach the destination is via one or more routers. The address of the first of these routers (the *first hop*) is called an *indirect route*. The first hop address is the only information needed by the source host: the router that receives a datagram has responsibility for the second hop and so on.

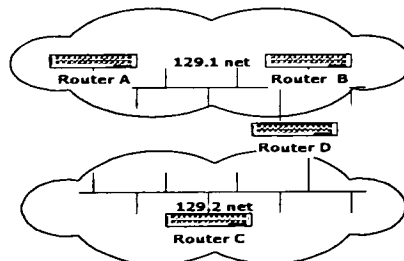


Figure 39: Direct and Indirect IP Routes

Host A has a *direct* route to hosts B and D, and an *indirect* route to host C. Host D is a router between the 129.1 and 129.2 networks. A host can tell whether a route is direct or indirect by examining the network number and subnet number parts of the IP address.

1. If they match one of the IP addresses of the source host, the route is a direct one. The host needs to be able to address the target correctly using a lower-level protocol than ARP. This can either be done automatically using a network-specific protocol, such as ARP, which is used on broadcast LANs, or by statically configuring the host, for example when an MVS host has a TCP/IP connection over an SNA link.
2. For "indirect" routes, the only knowledge required is the IP address of a router leading to the destination network.

IP implementations may also support explicit host routes, that is, a route to a specific IP address. This is common for dial-up connections using Serial Line Internet Protocol (SLIP) which does not provide a mechanism for two hosts to inform each other of their IP addresses. Such routes may even have the same network number as the host, for example on subnets composed of point-to-point links. In general, however, routing information is based on network number and subnet number only.

II. IP Routing Table

Each host keeps the set of mappings between destination IP addresses and the IP addresses of the next hop routers for those destinations in a table called the *IP routing table*.

Three types of mappings can be found in this table:

1. Direct routes, for locally attached networks
2. Indirect routes, for networks reachable via one or more routers
3. A default route, which contains the IP address of a router to be used for all IP addresses which are not covered by the direct and indirect routes.

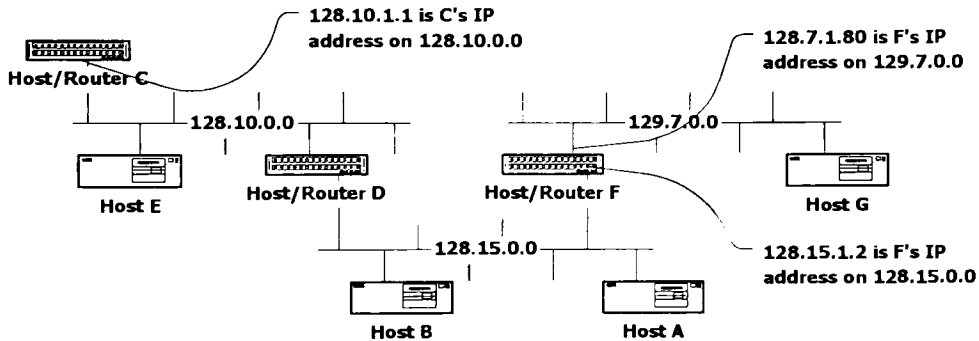


Figure 40: Example IP Routing Table

The routing table of host D will contain the following entries

Destination	route via
128.10	direct attachment
128.15	direct attachment
129.7	128.15.1.2
default	128.10.1.1

III. IP Routing Algorithm ⁸²

From the foregoing discussion, one can easily derive the steps that IP must take in order to determine the route for an outgoing datagram. This is called the *IP routing algorithm* and it is shown schematically in figure 41 below.

⁸² Merilee Ford, *Internetworking Technologies Handbook*. Cisco Press. 1997

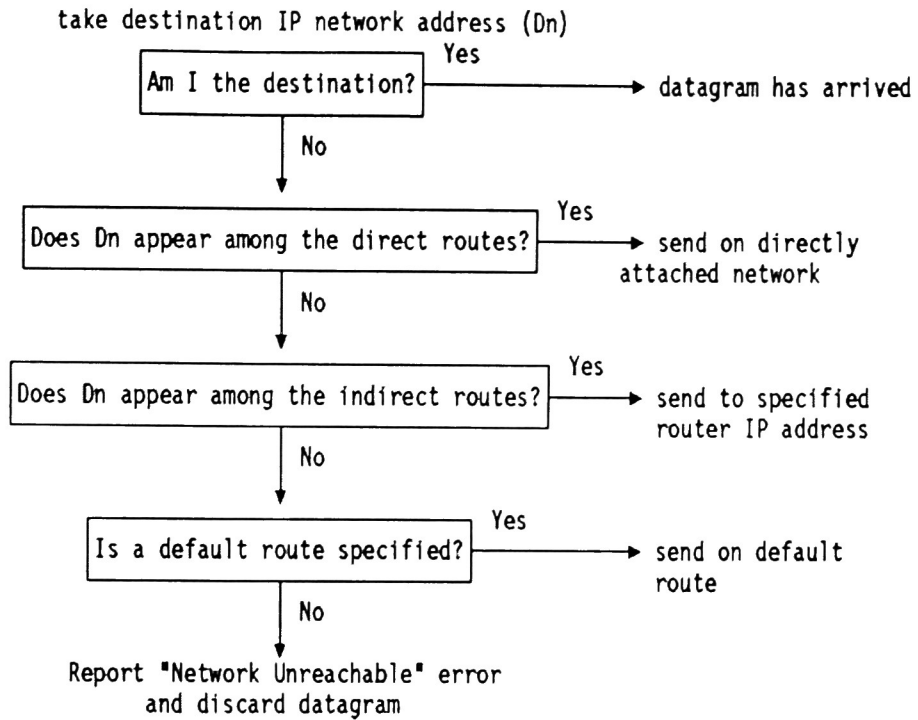


Figure 41: IP Routing Algorithm

Note that this is an iterative process. It is applied by every host handling a datagram, except for the host to which the datagram is finally delivered.

4.1.4 Internet Control Message Protocol (ICMP) ⁸³

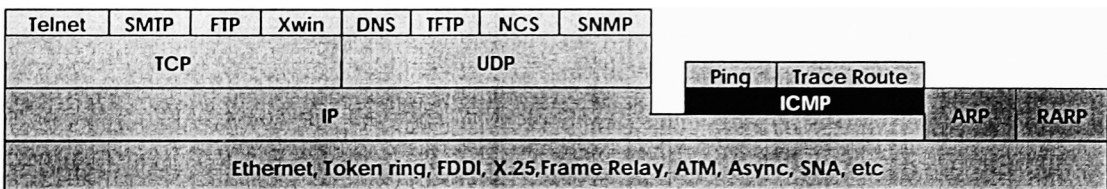


Figure 42: Internet Control Message Protocol (ICMP)

ICMP is a *standard protocol* with STD number 5 which also includes IP and IGMP. Its status is *required*. It is described in RFC 792 with updates in RFC 950. It is part of STD 5, which also includes IP. Path MTU Discovery is a *draft standard protocol* with a status of *elective*. It is described in RFC 1191. ICMP Router Discovery is a *proposed standard protocol* with a status of *elective*. It is described in RFC 1256. When a router or a destination host must inform the source host about errors in datagram processing, it uses the **Internet Control Message Protocol (ICMP)**. ICMP can be characterized as follows:

⁸³ RFC 792 - Internet Control Message Protocol

- ICMP uses IP as if ICMP were a higher-level protocol (that is, ICMP messages are encapsulated in IP datagrams). However, ICMP is an integral part of IP and must be implemented by every IP module.
- ICMP is used to report some errors, *not* to make IP reliable. Datagrams may still be undelivered without any report on their loss. Reliability must be implemented by the higher-level protocols that use IP.
- ICMP can report errors on any IP datagram with the exception of ICMP messages, to avoid infinite repetitions.
- For fragmented IP datagrams, ICMP messages are only sent about errors on fragment zero. That is, ICMP messages never refer to an IP datagram with a non-zero fragment offset field.
- ICMP messages are never sent in response to datagrams with a destination IP address that is a broadcast or a multicast address.
- ICMP messages are never sent in response to a datagram, which does not have a source IP address that represents a unique host. That is, the source address cannot be zero, a loopback address, a broadcast address or a multicast address.
- ICMP messages are never sent in response to ICMP error messages. They may be sent in response to ICMP query messages (ICMP types 0, 8, 9, 10 and 13 through 18).
- RFC 792 states that ICMP messages “may” be generated to report IP datagram processing errors, *not* “must”. In practice, routers will almost always generate ICMP messages for errors, but for destination hosts, the number of ICMP messages generated is implementation dependent.

4.1.4.1 ICMP Messages

ICMP messages are described in RFC 792 and RFC 950, belong to STD 5 and are mandatory. ICMP messages are sent in IP datagrams. The IP header will always have a Protocol number of 1, indicating ICMP and a type of service of zero (routine). The IP data field will contain the actual ICMP message in the format shown in figure 43 below.

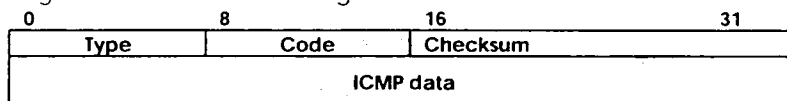


Figure 43: ICMP Message Format

Where:

- | | |
|------|------------------------------------|
| Type | Specifies the type of the message: |
| 0 | Echo reply |
| 3 | Destination unreachable |
| 4 | Source quench |
| 5 | Redirect |
| 8 | Echo |
| 9 | Router advertisement |
| 10 | Router solicitation |
| 11 | Time exceeded |
| 12 | Parameter problem |
| 13 | Timestamp request |
| 14 | Timestamp reply |
| 15 | Information request (obsolete) |
| 16 | Information reply (obsolete) |
| 17 | Address mask request |
| 18 | Address mask reply |

4.1.4.2 ICMP for IP Version 6

The ICMP implementation above is specific to IP Version 4 (IPv4). IP Version 6 (IPv6) will require a new version of ICMP. The definitions of both new versions of ICMP and IP are currently under test and review. Important features already known are:

- **ICMP for IP Version 6 (ICMPv6)** will use a new protocol number to distinguish it from **ICMP Version 4 (ICMP)**.
- The ICMP header format will remain the same.
- Field lengths in messages will change to accommodate longer IPv6 messages.
- The Type and code values will be changed. Certain little used values will be removed.
- The size of ICMP messages will be increased to exploit the increased size of packets, which IPv6 guarantees will be transmitted without fragmentation.
- The Fragmentation Required variant of the ICMP Destination unreachable message will be replaced by a Packet Too Big ICMP message which will include the outgoing link Maximum Transmission Unit (MTU) where the problem is identified.
- IGMP will be merged with ICMP.

4.1.5 Internet Group Management Protocol (IGMP) ⁸⁴

IGMP is a *standard protocol* with STD number 5 which also includes IP and ICMP. Its status is *recommended* and it is described in RFC 1112. IGMP is best regarded as an extension to ICMP and occupies the same place in the IP protocol stack.

4.1.5.1 IGMP Operation

Systems participating in IGMP fall into two types: hosts and multicast routers. As described in Multicasting, in order to receive multicast datagrams, a host must join a host group. When a host is multi-homed, it may join any group on one or more of its interfaces (attached subnets). The multicast messages that the host receives from the same group on two different subnets may be different. For example 244.0.0.1 is the group for "all hosts on this subnet", so the messages received on one subnet will always be different for this group from those on another. Multiple processes on a single host may be listening for messages for a multicast group on a subnet. If this is the case, the host joins the group once only, and keeps track internally of which processes are interested in that group.

To join a group, the host sends a report on an interface. The report is addressed to the multicast group of interest. Multicast routers on the same subnet receive the report and set a flag to indicate that at least one host on that subnet is a member of that group. No host has to join the all hosts group (224.0.0.1); membership is automatic. Multicast routers have to listen to all multicast addresses (that is, all groups) in order to detect such reports. The alternatives would be to use broadcasts for reports or to configure hosts with unicast addresses for multicast routers.

Multicast routers regularly, but infrequently (RFC 1112 mentions one-minute intervals), send out a query to the all hosts multicast address. Each host, which still wishes to be a member of one or more groups, replies once for each group of interest (but never the all hosts group, since membership is automatic). Each reply is sent after a random delay to ensure that IGMP does not cause bursts of traffic on the subnet. Since routers do not care how many hosts are members of a group and since all hosts which are members of that group can hear each other replying, any host which hears another claim membership of a group will cancel any reply that it is due to send in order to avoid wasting resources. If no hosts claim membership of a group within a specified interval, the multicast router decides that no host is a member of the group. When a host or a multicast router receives a multicast datagram, its action is dependent upon the TTL value and the destination IP address.

⁸⁴ RFC 1112 - *Internet Group Multicast Protocol*

- 0 A datagram sent with a TTL value of zero is restricted to the source host.
- 1 A datagram with a TTL value of one reaches all hosts on the subnet which are members of the group. Multicast routers decrement the value to zero, but unlike unicast datagrams, they do not report this with an ICMP Time Exceeded message. Expiration of a multicast datagram is a normal occurrence.
- 2+ All hosts which are members of the group and all multicast routers receive the datagram. The action of the routers depends on the multicast group address.

224.0.0.0 - 224.0.0.255 : This range is intended for single-hop multicasting applications only. Multicast routers will not forward datagrams with destination IP addresses in this range. It may seem at first as though a host need not bother reporting its membership of a group in this range since multicast routers will not forward datagrams from other subnets. However, the report also informs other hosts on the subnet that the reporting host is a member of the group. The only group, which is never reported, is 224.0.0.1 because all hosts know that the group consists of all hosts on that subnet.

Other : Datagrams with other values for the destination address are forwarded as normal by the multicast router: it decrements the TTL value by at least one second as usual. This allows a host to locate the nearest server, which is listening on a multicast address using what is called an *expanding ring search*. The host sends out a datagram with a TTL value of 1 (same subnet) and waits for a reply. If none is received, it tries a TTL value of 2, then 3, and so on. Eventually it will find the closest server.

4.1.6 Address Resolution Protocol (ARP) ⁸⁵

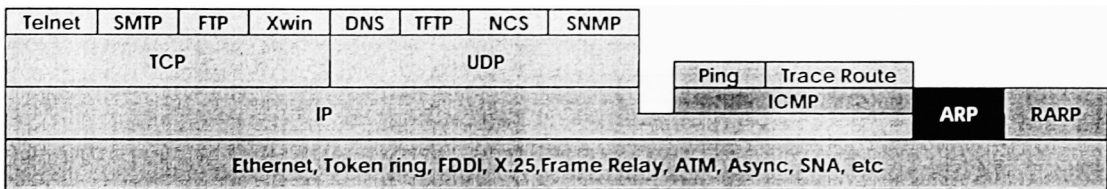
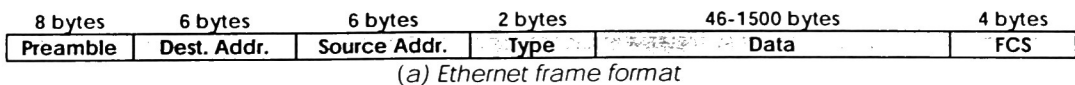


Figure 44: Address Resolution Protocol (ARP)

The **Address Resolution Protocol (ARP)** is a *network-specific standard protocol*. Its status is *elective*. The address resolution protocol is responsible for converting the higher-level protocol addresses (IP addresses) to physical network addresses. First, let's consider some general topics on Ethernet.

1.6.1 Ethernet versus IEEE 802.3 ⁸⁶

Two frame formats can be used on the Ethernet coaxial cable: (1) The standard issued in 1978 by Xerox Corporation, Intel Corporation and Digital Equipment Corporation, usually called *Ethernet* (or *DIX Ethernet*); (2) The international IEEE 802.3 standard, a more recently defined standard. The difference between the two standards is in the use of one of the header fields, which contains a protocol-type number for Ethernet and the length of the data in the frame for IEEE 802.3.



⁸⁵ RFC 826 - Address Resolution Protocol

⁸⁶ Fred HalSall, *Data Communications, Computer Networks and Open Systems*, Addison Wesley, 1993

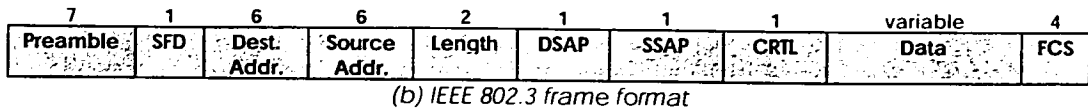


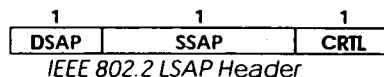
Figure 45: Frame Formats for Ethernet and IEEE 802.3

- ☛ The type field in Ethernet is used to distinguish between different protocols running on the coaxial cable, and allows their coexistence on the same physical cable.
- ☛ The maximum length of an Ethernet frame is 1526 bytes. This means a data field length of up to 1500 bytes. The length of the 802.3 data field is also limited to 1500 bytes for 10 Mbps networks, but is different for other transmission speeds.
- ☛ In the 802.3 MAC frame, the length of the data field is indicated in the 802.3 header. The type of protocol it carries is then indicated in the 802.2 header (higher protocol level).
- ☛ In practice however, both frame formats can coexist on the same physical coax. This is done by using protocol type numbers (type field) greater than 1500 in the Ethernet frame. However, different device drivers are needed to handle each of these formats.

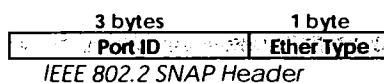
Thus, for all practical purposes, the Ethernet physical layer and the IEEE 802.3 physical layer are compatible. However, the Ethernet data link layer and the IEEE 802.3/802.2 data link layer are incompatible.

I. The 802.2 Logical Link Control (LLC) layer

The IEEE 802.3 uses a concept known as *Link Service Access Point (LSAP)* which uses a 3-byte header:



where DSAP and SSAP stand for Destination and Source Service Access Point respectively. An IEEE committee assigns Numbers for these fields. Due to a growing number of applications using IEEE 802 as lower protocol layers, an extension was made to the IEEE 802.2 protocol in the form of the *Sub-Network Access Protocol (SNAP)*. It is an extension to the LSAP header above, and its use is indicated by the value 170 in both the SSAP and DSAP fields of the LSAP frame above.



II. The evolution of TCP/IP ⁸⁷

Three standards were established, which describe the encapsulation of IP and ARP frames on these networks:

- ☛ 1984: *RFC 894 - Standard for the Transmission of IP Datagrams over Ethernet Networks* specifies only the use of Ethernet type of networks. The values assigned to the type field are:
 - 2048 (hex 0800) for IP datagrams
 - 2054 (hex 0806) for ARP datagrams
- ☛ 1985: *RFC 948 - Two Methods for the Transmission of IP Datagrams over IEEE 802.3 Networks* specifies two possibilities:
 - ☞ The Ethernet compatible method: the frames are sent on a real IEEE 802.3 network in the same fashion as on an Ethernet network, that is, using the IEEE 802.3 data-length field as the Ethernet type field, thereby violating the IEEE 802.3 rules, but compatible with an Ethernet network.

⁸⁷ *RFC 1042 - Standard for the Transmission of IP Datagrams over IEEE 802 Networks*

⇒ IEEE 802.2/802.3 LLC type 1 format: using 802.2 LSAP header with IP using the value 6 for the SSAP and DSAP fields.

The RFC indicates clearly that the IEEE 802.2/802.3 method is the preferred method, that is, that all future IP implementations on IEEE 802.3 networks are supposed to use the second method.

- 1987: *RFC 1010 - Assigned Numbers* (now obsolete by RFC 1700 dated 1994) notes that as a result of IEEE 802.2 evolution and the need for more internet protocol numbers, a new approach was developed based on practical experiences exchanged during the August 1986 TCP Vendors Workshop. It states in an almost completely overlooked part of this RFC that from now on all IEEE 802.3, 802.4 and 802.5 implementations should use the Sub-Network Access Protocol (SNAP) form of the IEEE 802.2 LLC: DSAP and SSAP fields set to 170 (indicating the use of SNAP) and then SNAP assigned as follows:

⇒ 0 (zero) as organization code.

⇒ Ether Type field:

2048 (hex 0800) for IP datagrams

2054 (hex 0806) for ARP datagrams

32821 (hex 8035) for RARP datagrams

These are the same values as used in the Ethernet type field.

- 1988: *RFC 1042 - Standard for the Transmission of IP Datagrams over IEEE 802 Networks*. As this new approach (very important for implementations) passed almost unnoticed in a little note of an unrelated RFC, it became quite confusing, and finally, in February 1988, it was repeated in an RFC on its own: *RFC 1042*, which obsoletes *RFC 948*.

However, in practical situations, there are still TCP/IP implementations that use the older LSAP method (RFC 948 or 1042). Such implementations will not communicate with the more recent implementations.

4.1.6.2 ARP Overview⁸⁸

On a single physical network, individual hosts are known on the network by their physical hardware address. Higher-level protocols address destination hosts in the form of a symbolic address (IP address in this case). When such a protocol wants to send a datagram to destination IP address w.x.y.z, the device driver does not understand this address.

Therefore, a module (ARP) is provided that will translate the IP address to the physical address of the destination host. It uses a lookup table (sometimes referred to as the *ARP cache*) to perform this translation. When the address is not found in the ARP cache, a broadcast is sent out on the network, with a special format called the *ARP request*. If one of the machines on the network recognizes its own IP address in the request, it will send an *ARP reply* back to the requesting host. The reply will contain the physical hardware address of the host and source route information (if the packet has crossed bridges on its path). Both this address and the source route information are stored in the ARP cache of the requesting host. All subsequent datagrams to this destination IP address can now be translated to a physical address, which is used by the device driver to send out the datagram on the network. ARP was designed to be used on networks that support hardware broadcast. This means, for example, that ARP will not work on an X.25 network.

4.1.6.3 ARP Detailed Concept

ARP is used on IEEE 802 networks as well as on the older DIX Ethernet networks to map IP addresses to physical hardware addresses. To do this, it is closely related to the device driver for that network. In fact, the ARP specifications in RFC 826 only describe its functionality, not its implementation. The implementation depends to a large extent on the device driver for a network type and they are usually coded together in the *adapter microcode*.

⁸⁸ Timothy Parker et al, *TCP/IP Unleashed*, SAMS Publishing, 1996

I. ARP Packet Generation

If an application wishes to send data to a certain IP destination address, the IP routing mechanism first determines the IP address of the "next hop" of the packet (it can be the destination host itself, or a router) and the hardware device on which it should be sent. If it is an IEEE 802.3/4/5 network, the ARP module must be consulted to map the *<protocol type, target protocol address>* to a physical address.

The ARP module tries to find the address in this ARP cache. If it finds the matching pair, it gives the corresponding 48-bit physical address back to the caller (the device driver) which then transmits the packet. If it doesn't find the pair in its table, it *discards the packet* (assumption is that a higher-level protocol will retransmit) and generates a network broadcast of an ARP request.

II. ARP Packet Reception

When a host receives an ARP packet (either a broadcast request or a point-to-point reply), the receiving device driver passes the packet to the ARP module, which treats it as shown in figure.

The requesting host will receive this ARP reply, and will follow the same algorithm to treat it. As a result of this, the triplet *<protocol type, protocol address, hardware address>* for the desired host will be added to its lookup table (ARP cache). The next time a higher-level protocol wants to send a packet to that host, the ARP module will find the target hardware address and the packet will be sent to that host. Note that because the original ARP request was a broadcast on the network, all hosts on that network will have updated the sender's hardware address in their table (only if it was already in the table).

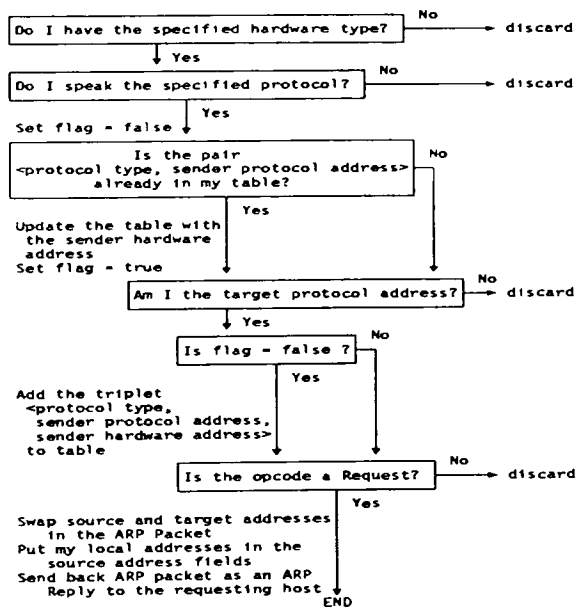


Figure 46: ARP Packet Reception

4.1.6.4 ARP and Subnets

The ARP protocol remains unchanged in the presence of subnets. Remember that each IP datagram first goes through the IP routing algorithm. This algorithm selects the hardware device driver that should send out the packet. Only then, the ARP module associated with that device driver is consulted.

4.1.6.5 Proxy-ARP or Transparent Subnetting

Proxy-ARP is described in *RFC 1027 - Using ARP to Implement Transparent Subnet Gateways*, which is in fact a subset of the method proposed in *RFC 925 - Multi-LAN Address Resolution*. It is another method to construct local subnets, without the need for a modification to the IP routing algorithm, but with modifications to the routers, which interconnect the subnets.

I. Proxy-ARP Concept

Consider one IP network, which is divided into subnets, interconnected by routers. We use the "old" IP routing algorithm, which means that no host knows about the existence of multiple physical networks. Consider hosts A and B which are on different physical networks within the same IP network, and a router R between the two subnetworks:

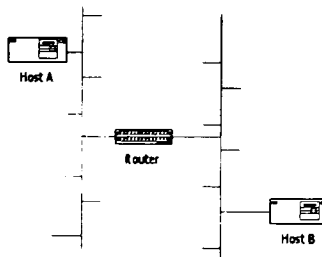


Figure 47: Hosts Interconnected by a Router

When host A wants to send an IP datagram to host B, it first has to determine the physical network address of host B through the use of the ARP protocol. As host A cannot differentiate between the physical networks, his IP routing algorithm thinks that host B is on the local physical network and sends out a broadcast ARP request. Host B doesn't receive this broadcast, but router R does. Router R understands subnets, that is, it runs the "subnet" version of the IP routing algorithm and it will be able to see that the destination of the ARP request (from the target protocol address field) is on another physical network. If router R's routing tables specify that the next hop to that other network is through a different physical device, it will reply to the ARP *as if it were host B*, saying that the network address of host B is that of the router R itself.

Host A receives this ARP reply, puts it in his cache and will send future IP packets for host B to the router R. The router will forward such packets to the correct subnet. The result is transparent subnetting:

- ✎ Normal hosts (such as A and B) don't know about subnetting, so they use the "old" IP routing algorithm.
- ✎ The routers between subnets have to:
 - ∞ Use the "subnet" IP algorithm.
 - ∞ Use a modified ARP module, which can reply on behalf of other hosts.

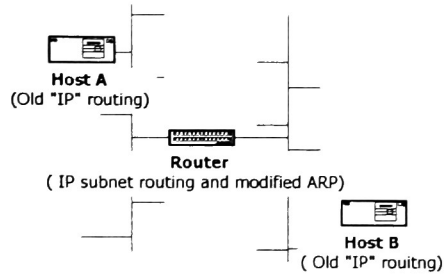


Figure 48: Proxy-ARP Router

4.1.7 Reverse Address Resolution Protocol (RARP) ⁸⁹

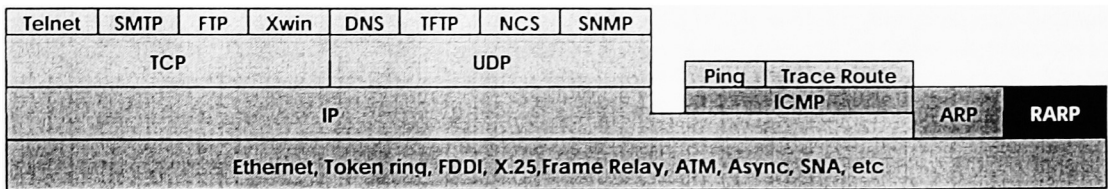


Figure 49: Reverse Address Resolution Protocol (RARP)

4.1.7.1 RARP Overview

The RARP protocol is a *network-specific standard protocol*. Its status is *elective*. Some network hosts, such as diskless workstations, do not know their own IP address when they are booted. To determine their own IP address, they use a mechanism similar to **ARP (Address Resolution Protocol)**, but now the hardware address of the host is the known parameter, and the IP address the queried parameter. It differs more fundamentally from ARP in the fact that a "RARP server" must exist on the network that maintains a database of mappings from hardware address to protocol address.

4.1.7.2 RARP Concept

The reverse address resolution is performed the same way as the ARP address resolution. The same packet format is used as for ARP. An exception is the "operation code" field which now takes the following values:

3 for the RARP request

4 for the RARP reply

And of course, the "physical" header of the frame will now indicate RARP as the higher-level protocol (8035 hex) instead of ARP (0806 hex) or IP (0800 hex) in the *EtherType* field. Some differences arise from the concept of RARP itself:

- ☞ ARP only assumes that every host knows the mapping between its own hardware address and protocol address. RARP requires one or more server hosts on the network to maintain a database of mappings between hardware addresses and protocol addresses so that they will be able to reply to requests from client hosts
- ☞ Due to the size this database can take, part of the server function is usually implemented outside the adapter's microcode, with optionally a small cache in the microcode. The microcode part is then only responsible for reception and transmission

⁸⁹ RFC 903 - Reverse Address resolution Protocol

- of the RARP frames, the RARP mapping itself being taken care of by server software running as a normal process in the host machine.
- ✎ The nature of this database also requires some software to create and update the database manually.
- ✎ In case there are multiple RARP servers on the network, the RARP requester only uses the first RARP reply received on its broadcast RARP request, and discards the others.

4.2. IP Routing Protocols ⁹⁰

One of the basic functions of IP is its ability to form connections between different physical networks. This is due to the flexibility of IP to use almost any physical network below it, and to the IP routing algorithm. A system which does this is termed a *router*, although the older term *IP gateway* is also used.

The routing function is part of the internetwork layer, but the primary function of a routing protocol is to *exchange* routing information with other routers, and in this respect the protocols behave more like application protocols. The routing protocols described here use all three approaches to data transport: using UDP (for example **Routing Information Protocol (RIP)**), TCP (for example **Border Gateway Protocol (BGP)**) and providing its own transport layer on top of IP (for example **Open Shortest Path First (OSPF)**). Therefore, we shall not attempt to represent the position of these protocols in the protocol stack with a diagram as we do with the other protocols.

4.2.1 Basic IP Routing

The fundamental function for routers is present in *all* IP implementations: *An incoming IP datagram that specifies a "destination IP address" other than the local host's IP address, is treated as a normal outgoing IP datagram.*

This outgoing IP datagram is subject to the IP routing algorithm of the local host, which selects the *next hop* for the datagram (the next host to send it to). This new destination can be located on any of the physical networks to which the intermediate host is attached. If it is a physical network other than the one on which the host originally received the datagram, then the net result is that the intermediate host has *forwarded* the IP datagram from one physical network to another.

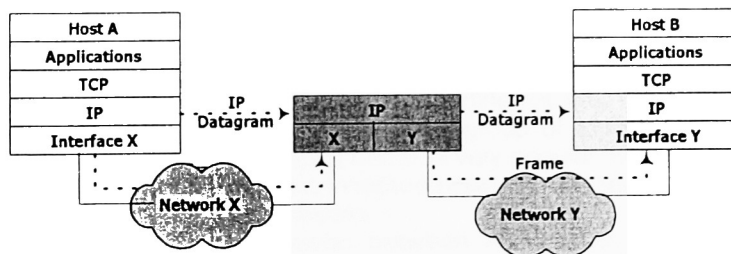


Figure 50: Router Operation of IP

The normal IP routing table contains information about the locally attached networks and the IP addresses of other routers located on these networks, plus the networks they attach to. It can be extended with information on IP networks that are farther away, and can also contain a default route, but it still remains a table with limited information; that is, it represents only a part of the whole internet. That is why this kind of router is called a *router with partial routing information*.

Some considerations apply to these routers with partial information:

- ✎ They do not know about all internet networks.

⁹⁰ John T. Moy, *OSPF - Anatomy of an Internet Routing Protocol*, Addison-Wesley, 1998

- ☞ They allow local sites autonomy in establishing and modifying routes.
- ☞ A routing entry error in one of the routers may introduce inconsistencies, thereby making part of the network unreachable.

Routers should implement some error reporting with partial information via the **Internet Control Message Protocol (ICMP)**. They should be able to report the following errors back to the source host:

- ☞ Unknown IP destination network by an ICMP *Destination Unreachable* message.
- ☞ Redirection of traffic to more suitable routers by sending ICMP *Redirect* messages.
- ☞ Congestion problems (too many incoming datagrams for the available buffer space) by an ICMP *Source Quench* message.
- ☞ The "Time-to-Live" field of an IP datagram has reached zero. This is reported with an ICMP *Time Exceeded* message.
- ☞ Also, the following base ICMP operations and messages should be supported:
 - ☞ Parameter problem
 - ☞ Address mask
 - ☞ Time stamp
 - ☞ Information request/reply
 - ☞ Echo request/reply

A more intelligent router is required if:

- ☞ The router has to know routes to *all* possible IP networks, as was the case for the ARPANET core gateways.
- ☞ The router has to have dynamic routing tables, which are kept up-to-date with minimal or no manual intervention.
- ☞ The router has to be able to advertise local changes to other routers.

These more advanced forms of routers use additional protocols to communicate with each other. A number of protocols of this kind exist, and descriptions of the important ones will be given in the following sections. The reasons for this multiplicity of different protocols are basically fourfold:

- ☞ Using Internet terminology, there is a concept of a group of networks, called an **Autonomous System (AS)**, which is administered as a unit. The AS concept arose because the TCP/IP protocols were developed with the ARPANET already in place. Routing within an AS and routing outside an AS are treated as different issues and are addressed by different protocols.
- ☞ Over two decades several routing protocols were tested in the Internet. Some of them performed well, others had to be abandoned.
- ☞ The emergence of Autonomous Systems of different sizes called for different routing solutions. For small to medium sized ASs a group of routing protocols based upon Distance Vector (RIP, for example) became very popular. However, such protocols do not perform well for large interconnected networks. Link State protocols like OSPF are much better suited for such networks.
- ☞ To exchange routing information between ASs border gateway protocols were developed.

Before discussing the various routing protocols, we will review the routing architectures used in the early Internet, since this will help in understanding the role played by the differing routing protocols. This overview will also show the difference between *Interior* and *Exterior* routing. We will then discuss the various protocols used for the two types of routing.

4.2.1.1 Routing Daemons

The routing protocols are often implemented using one of two daemons:

routed : Pronounced "route D". This is a basic routing daemon for interior routing supplied with the majority of TCP/IP implementations. It uses the RIP protocol.

gated : Pronounced "gate D". This is a more sophisticated daemon on UNIX-based systems for interior and exterior routing. It can employ a number of additional protocols such as OSPF and BGP.

4.2.2 Routing Architecture ⁹¹

Initially, the ARPANET played a central role in the development of the Internet, particularly in the area of routing. Although it was replaced in its role as the backbone of the Internet by the NSFNET in the late 1980s, the experience gained from its routing architecture had a direct effect on the later development of the current set of routing protocols. At its heart was the concept of an *Autonomous System (AS)*: a collection of networks controlled by a single authority. Each AS is registered with the NIC (now the InterNIC) and has a 16-bit identification number called the *autonomous system number* or *AS number*. These are listed in *RFC 1166 - Internet Numbers*. The ARPANET core system was itself considered an autonomous system.



Figure 51: The ARPANET Backbone

In keeping with the nomenclature used at the time, we shall refer to the routers between Autonomous Systems as *gateways*. All routing between gateways can be categorized as *intra-AS* (also termed *Interior*) if the gateways belong to the same AS or *inter-AS* (also termed *Exterior*) if they belong to different ones. Intra-AS routing uses an *Interior Gateway Protocol (IGP)* and inter-AS routing uses an *Exterior Gateway Protocol (EGP)*. The ARPANET architecture did not specify which protocol should be used as an IGP, but it did specify a protocol to be used as an EGP. Confusingly, this protocol was also named *Exterior Gateway Protocol*. To avoid confusion, we shall use the term "EGP" to refer specifically to the EGP protocol, and "an EGP" to refer to a protocol belonging to the EGP group of protocols.

I. Core and Non-Core Gateways

In the ARPANET system, a central authority, the Internet Network Operations Center, maintained the core gateways, which make up the backbone. They provided reliable and authoritative routes for *all* possible Internet networks, and connected the ARPANET to the other Internet networks.

The core gateways (shown as CGx, CGy and so on in figure) had to know about all possible destinations in order to optimize the ARPANET traffic. A datagram travelling from one local network to another via the core system passed through exactly two core gateways. The ARPANET routing architecture specified that the core gateways communicated with the *Gateway-to-Gateway Protocol (GGP)*.

Non-core gateways (shown as G in figure) were maintained by the organizations responsible for the individual Autonomous Systems and forwarded information about networks in their areas to the core gateways using EGP.

II. Core Gateways

In addition to the simple ICMP error-reporting messages, an ARPANET core gateway also implemented:

- *Gateway-to-Gateway Protocol (GGP)* to exchange connectivity information between core gateways.

⁹¹ TCP/IP Networking, <http://www.lmu.edu/admin/IS/training/protected/tcpip.html>

- **Exterior Gateway Protocol (EGP)** to collect connectivity information from non-core gateways.
- **Cross-Network Debugging Protocol (XNET)**, used to load the gateway and to create and examine the gateway's data.
- **Host Monitoring Protocol (HMP)** used to collect measurements and statistics information from the gateways (RFC 869).

III. Non-Core Gateways

Local internetworks created by individual groups can span multiple physical networks, tied together through gateways (non-core gateways). Such a group of networks is called an **Autonomous System (AS)**. Among its responsibilities, an AS must:

- Collect reachability information for all of its connected networks.
- Advertise reachability information to the core system using a standard protocol.
- Have a single administrative and technical point of contact.

An autonomous system must collect routing and reachability information about its own internal networks. Selected machines must forward that information to other autonomous systems and to the core gateways. As noted above, EGP must be used for this inter-AS communication.

For inter-AS communication any suitable IGP may be used, the two most common being:

- The Hello protocol
- Routing Information Protocol (RIP)

IV. Gateway-to-Gateway Protocol (GGP)

GGP is an *historic protocol*. Its status is *not recommended*. It is described in detail in *RFC 823 - The DARPA Internet Gateway*. As mentioned previously, the original ARPANET core gateways used the *Gateway-to-Gateway Protocol*, to exchange routing information. In addition to this role, it had to route datagrams that were passing through the core system. Any datagram in transit through the core system should pass through two core gateways. The basic principles of GGP follow:

When a core gateway comes up, it is assigned core *neighbors*. A gateway only needs to propagate information about the networks it can reach to its neighbors. The neighbors will update their routing information with the received information and will send the changes to their assigned neighbors.

The information consists of sets (N,C) where:

N is a network that is reachable by this gateway

C is the cost of reaching that network. The cost is expressed in gateway hops (number of gateways to pass). A cost of zero corresponds to a network that is directly attached to the core gateway. Maximum cost corresponds to unreachable networks.

GGP messages are carried in IP datagrams, and typically contain a list of (N,C) pairs. They are sent by a gateway to its neighbors whenever one of the following occurs:

- A new network becomes reachable from the gateway.
- A network becomes unreachable.
- Routing data is changed due to reception of GGP messages from neighbor gateways.

Upon receipt of a GGP message from gateway G, the neighbor gateway A will compare an incoming (N,C) pair to the (N,C) pair in its local tables. If the cost to reach network N would be smaller by using the gateway G (originator of the GGP message) than when using the routing information in the local table, the routing path for network N is updated to point to gateway G,

and as this is a route change, the gateway A will generate a GGP message to inform its neighbors of the change. Eventually, the information on network N will reach all the core gateways.

4.2.3 Interior Routing Protocols

Interior routing protocols or **Interior Gateway Protocols (IGPs)** are used to exchange routing information between routers within a single autonomous system. They are also used by routers, which run *exterior routing protocols* to collect network-reachability information for the autonomous system. The term interior routing protocol has no abbreviation in common use, so we shall use the abbreviation IGP as is usual in TCP/IP literature.

The most widely used IGPs are:

- The Hello protocol
- Routing Information Protocol
- The Open Shortest Path First protocol

Before discussing these three protocols in detail, we shall look at two important groups of routing algorithm used in IGPs.

4.2.3.1 Routing Algorithms

In this section, we discuss the Vector-Distance and Link-State, Shortest Path First routing algorithms.

I. Vector-Distance ⁹²

The term *Vector-Distance* refers to a class of algorithms that gateways use to update routing information. Each router begins with a set of routes for those networks or subnets to which it is directly attached, and possibly some additional routes to other networks or hosts if the network topology is such that the routing protocol will be unable to produce the desired routing correctly. This list is kept in a *routing table*, where each entry identifies a destination network or host and gives the "distance" to that network. The distance is called a *metric* and is typically measured in "hops".

Periodically, each router sends a copy of its routing table to any other router it can reach directly. When a report arrives at router B from router A, B examines the set of destinations it receives and the distance to each. B will update its routing table if:

- A knows a shorter way to reach a destination.
- A lists a destination that B does not have in its table.
- A's distance to a destination, already routed through A from B, has changed.

This kind of algorithm is easy to implement, but it has a number of disadvantages:

- When routes change rapidly, that is, a new connection appears or an old one fails, the routing topology may not stabilize to match the changed network topology because information propagates slowly from one router to another and while it is propagating, some routers will have incorrect routing information.
- Another disadvantage is that each router has to send a copy of its entire routing table to every neighbor at regular intervals. Of course, one can use longer intervals to reduce the network load but that introduces problems related to how well the network responds to changes in topology.
- Vector-distance algorithms using hop counts as a metric do not take account of the link speed or reliability. Such an algorithm will use a path with hop count 2 that crosses two slow-speed lines, instead of using a path with hop count 3 that crosses three token-rings and may be substantially faster.

⁹² Andrew S. Tanenbaum, *Computer Networks*, Prentice Hall, 1996

The most difficult task in a vector-distance algorithm is to prevent instability. Different solutions are available:

Counting to infinity: Let us choose a value of 16 to represent infinity. Suppose a network becomes inaccessible; all the immediately neighboring routers time out and set the metric to that network to 16. We can consider that all the neighboring routers have a piece of hardware that connects them to the vanished network, with a cost of 16. Since that is the only connection to the vanished network, all the other routers in the system will converge to new routes that go through one of those routers with a direct but unavailable connection. Once convergence has happened, all the routers will have metrics of 16 for the vanished network. Since 16 indicates infinity, all routers then regard the network as unreachable. The question with vector distance algorithms is not *whether* convergence will occur but *how long* will it take? Let us consider the configuration shown in figure.

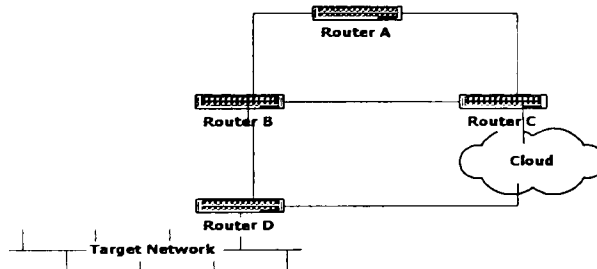


Figure 52: The Counting to Infinity Problem

All links have a metric of 1 except for the indirect route from C to D, which has a metric of 10. Let us consider only the routes from each gateway to the target network. Now, consider that the link from B to D fails. The routes should now adjust to use the link from C to D. The routing changes start when B notices that the route to D is no longer usable. For RIP this occurs when B does not receive a routing update on its link to D for 180 seconds.

Gateway	First Hop	Metric
D	-	1
B	D	2
C	B	3
A	B	3

The following picture shows the metric to the target network, as it appears in the routing table of each gateway.

Gateway	1		2		3		9		10	
	FH	M	FH	M	FH	M	FH	M	FH	M
D	-	1	-	1	-	1	-	1	-	1
B	x	x	C	4	C	5	-	-	-	-
C	B	3	A	4	A	5	C	11	C	12
A	B	3	C	4	C	5	A	11	D	11
							C	11	C	12

Table 4: The Counting to Infinity Problem

The problem is that B can get rid of its route to D (using a timeout mechanism), but vestiges of that route persist in the system for a long time (time between iterations is 30 seconds using RIP). Initially, A and C still think they can reach D via B, so they keep sending updates listing metrics of 3. B will receive these updates and, in the next iteration, will claim that it can get to D via either A or C. Of course, it can't because the routes claimed by A and C (D reachable via B with a metric of 3) are now gone, but they have no way of knowing that yet. Even when they discover that their routes via B have gone away, they each think there is a route available via the other. Eventually the system will converge, when the direct link from C to D has a lower cost than the one received (by C) from B and A.

The worst case is when a network becomes completely inaccessible from some part of the system: in that case, the metrics may increase slowly in a pattern like the one above until they finally reach "infinity". For this reason, the problem is called *counting to infinity*. Thus

the choice of infinity is a trade off between network size and speed of convergence in case counting to infinity happens. This explains why we chose as low a value as 16 to represent infinity. 16 is the value used by RIP.

II. Link-State, Shortest Path First ⁹³

The growth in networking over the past few years has pushed the currently available *Interior Gateway Protocols*, which use vector-distance algorithms, past their limits. The primary alternative to vector-distance schemes is a class of protocols known as *Link State, Shortest Path First*.

The important features of these routing protocols are:

- ✎ A set of physical networks is divided into a number of areas.
- ✎ All routers within an area have an identical database.
- ✎ Each router's database describes the complete topology (which routers are connected to which networks) of the routing domain. The topology of an area is represented with a database called a *Link State Database* describing all of the links that each of the routers in the area has.
- ✎ Each router uses its database to derive the set of optimum paths to all destinations from which it builds its routing table. The algorithm used to determine the optimum paths is called a **Shortest Path First (SPF)** algorithm.

In general, a link state protocol works as follows. Each router periodically sends out a description of its connections (the state of its links) to its neighbors (routers are neighbors if they are connected to the same network). This description, called a **Link State Advertisement (LSA)**, includes the configured cost of the connection. The LSA is flooded throughout the router's domain. Each router in the domain maintains an identical synchronized copy of a database composed of this link state information. This database describes both the topologies of the router's domain and routes to networks outside of the domain such as routes to networks in other autonomous systems. Each router runs an algorithm on its topological database resulting in a shortest-path tree. This shortest-path tree contains the shortest path to every router and network the gateway can reach. From the shortest-path tree, the cost to the destination and the next hop to forward a datagram is used to build the router's routing table.

Link-state protocols, in comparison with vector-distance protocols, send out updates when there is news, and may send out regular updates as a way of ensuring neighbor routers that a connection is still active. More importantly, the information exchanged is the state of a router's links, not the contents of the routing table. This means that link-state algorithms use fewer network resources than their vector-distance counterparts, particularly when the routing is complex or the autonomous system is large. They are, however, compute-intensive. In return, users get faster response to network events, faster route convergence, and access to more advanced network services.

4.2.3.2 The Hello Protocol ⁹⁴

This was used in the "Fuzzball" software for LSI/11 minicomputers, which were widely used in Internet experimentation. The Hello protocol is described in *RFC 891 - DCN Local-Network Protocols*. It is not an Internet standard. OSPF includes a quite separate protocol for negotiation between routers which is also called the Hello protocol.

The communication in the Hello protocol is via Hello messages, which are carried via IP datagrams. Hello uses protocol number 63 (reserved for "any local network"). The Hello protocol is significant partly because of its wide deployment during the early expansion of the Internet and partly because it provides an example of a vector-distance algorithm that does not use hop counts like RIP but, instead, network delays as a metric for the distance.

⁹³ Andrew S. Tanenbaum, *Computer Networks*, Prentice Hall, 1996

⁹⁴ *RFC 891 - DCN Local-Network Protocols*

4.2.3.3 Routing Information Protocol (RIP) ⁹⁵

There are two versions of RIP. Version 1 (RIP-1) is a widely deployed protocol with a number of known limitations. Version 2 (RIP-2) is an enhanced version designed to alleviate the limitations of RIP while being highly compatible with it. The term RIP is used to refer to Version 1, while RIP-2 refers to Version 2. Whenever the reader encounters the term RIP in TCP/IP literature, it is safe to assume that it is referring to Version 1 unless explicitly stated otherwise. We shall use this nomenclature in this section except when the two versions are being compared, when we shall use the term RIP-1 to avoid possible confusion.

I. Routing Information Protocol Version 1 (RIP, RIP-1)

RIP is a *standard protocol* (STD 34). Its status is *elective*. It is described in RFC 1058, although many RIP implementations pre-date this RFC by a number of years. RIP is generally implemented with a daemon named *routed*. RIP is also supported by *gated* daemons. RIP was based on the Xerox PUP and XNS routing protocols. It is widely used, as the code is incorporated in the routing code of Berkeley BSD UNIX, which provides the basis for many UNIX implementations.

RIP is a straightforward implementation of vector-distance routing for local networks. RIP communication uses UDP as a transport protocol, with port number 520 as the destination port for a description of UDP and ports). RIP operates in one of two modes: *active* (normally used by routers) and *passive* (normally used by hosts).

Limitations - RIP is not designed to solve every possible routing problem. RFC 1720 (STD 1) describes these technical limitations of RIP as "serious" and the IETF is evaluating candidates for a new standard "open" protocol to replace RIP. Possible candidates include OSPF and OSI IS-IS. However, RIP is widely deployed and therefore is unlikely to be completely replaced for some time. RIP has the following specific limitations:

- ✎ The maximum cost allowed in RIP is 16, which means that the network is unreachable. Thus RIP is inadequate for large networks (that is, those in which legitimate hop counts approach 16).
- ✎ RIP does not support variable length subnet masks (*variable subnetting*). There is no facility in a RIP message to specify a subnet mask associated with the IP address.
- ✎ RIP has no facilities to ensure that routing table updates come from authorized routers. It is an unsecure protocol.
- ✎ RIP only uses fixed metrics to compare alternative routes. It is not appropriate for situations where routes need to be chosen based on real-time parameters such as measured delay, reliability, or load.
- ✎ The protocol depends upon *counting to infinity* to resolve certain unusual situations. As described earlier, the resolution of a loop would require either much time (if the frequency of updates was limited) or much bandwidth (if updates were sent whenever changes were detected). As the size of the routing domain grows, the instability of the vector-distance algorithm in the face of changing topology becomes apparent. RIP specifies mechanisms to minimize the problems with counting to infinity (these are described below) which allows RIP to be used for larger routing domains, but eventually RIP will be unable to cope. There is no fixed upper limit, but the practical maximum depends upon the frequency of changes to the topology, the details of the network topology itself, and what is deemed as an acceptable maximum time for the routing topology to stabilize.

Using the split horizon, poisoned reverse and triggered updates techniques, one can solve the counting to infinity problem.

⁹⁵ RFC 1058 - Routing Information Protocol

Split horizon with poisoned reverse - Let's consider our example network again. All links have a metric of 1 except for the indirect route from C to D, which has a metric of 10.

As described in Vector-Distance, the problem was caused by the fact that A and C are engaged in a pattern of mutual deception. Each claims to be able to reach D via the other. This can be prevented by being more careful about where information is sent. In particular, it is never useful to claim reachability for a destination network to the neighbor from which the route was learned (reverse routes). The *split horizon with poisoned reverse* scheme includes routes in updates sent to the router from which they were learned, but sets their metrics to infinity. If two routers have routes pointing at each other, advertising reverse routes with a metric of 16 will break the loop immediately. If the reverse routes are simply not advertised (this scheme is called *simple split horizon*), the erroneous routes will have to be eliminated by waiting for a timeout. Poisoned reverse does have a disadvantage: it increases the size of the routing messages.

Triggered updates - Split horizon with poisoned reverse will prevent any routing loop that involves only two gateways. However, it is still possible to end up with patterns in which three routers are engaged in mutual deception. For example, A may believe it has a route through B, B through C, and C through A. This cannot be solved using split horizon. This loop will only be resolved when the metric reaches infinity and the network or host involved is then declared unreachable. Triggered updates are an attempt to speed up this convergence. Whenever a router changes the metric for a route, it is required to send update messages almost immediately, even if it is not yet time for one of the regular update messages (RIP specifies a small time delay, between 1 and 5 seconds, in order to avoid having triggered updates generate excessive network traffic).

II. Routing Information Protocol Version 2 (RIP-2) ⁹⁶

RIP-2 is a *draft standard protocol*. Its status is *elective*. It is described in RFC 1723. RIP-2 extends RIP-1. It is less powerful than other recent IGPs such as OSPF and IS-IS, but it has the advantages of easy implementation and lower overheads. The intention of RIP-2 is to provide a straightforward replacement for RIP, which can be, used on small to medium-sized networks, can be employed in the presence of variable subnetting or superneting (CIDR) and importantly, can interoperate with RIP-1.

RIP-2 takes advantage of the fact that half of the bytes in a RIP-1 message are reserved (must be zero) and that the original RIP-1 specification was well designed with enhancements in mind, particularly in the use of the version field. One notable area where this is not the case is in the interpretation of the metric field. RIP-1 specifies it as being a value between 0 and 16 stored in a four-byte field. For compatibility, RIP-2 preserves this definition, meaning that it agrees with RIP-1 that 16 is to be interpreted as infinity, and wastes most of this field. Neither RIP-1 nor RIP-2 are properly suited for use as an IGP in an AS where a value of 16 is too low to be regarded as infinity, because high values of infinity exacerbate the counting to infinity problem. The more sophisticated Link-State protocol used in OSPF and IS-IS provides a much better routing solution when the AS is large enough to have a legitimate hop count close to 16. Provided that a RIP-1 implementation obeys the specification in RFC 1058, RIP-2 can interoperate with RIP-1.

To ensure safe interoperation with RIP, RFC 1723 specifies the following restrictions for RIP-2 routers sending over a network interface where a RIP-1 router may hear and operate on the RIP messages.

3. Information internal to one network must never be advertised into another network.
4. Information about a more specific subnet may not be advertised where RIP-1 routers would consider it a host route.
5. *Supernet* routes (routes with a subnet mask shorter than the natural or "unsubnetted" network mask) must not be advertised where they could be misinterpreted by RIP-1 routers.

⁹⁶ RFC 1721 - RIP Version 2 Protocol Analysis

RIP-2 also supports the use of multicasting rather than simple broadcasting. This can reduce the load on hosts, which are not listening for RIP-2 messages. This option is configurable for each interface to ensure optimum use of RIP-2 facilities when a router connects mixed RIP-1/RIP-2 subnets to RIP-2-only subnets. Similarly, the use of authentication in mixed environments can be configured to suit local requirements.

RIP-2 is implemented in recent versions of the *gated* daemon, often termed *gated Version 3*. Since the draft standard is new at the time of writing, many implementations will comply with the earlier version described in RFC 1388. Such implementations will interoperate with those adhering to RFC 1723.

4.2.3.4 Open Shortest Path First Protocol (OSPF) Version 2 ⁹⁷

The term OSPF is invariably used to refer to *OSPF Version 2 (OSPF-2)*. OSPF Version 1, which is described in RFC 1131, is obsolete. OSPF is a *draft standard protocol*. Its status is *elective*, but RFC 1370 contains an *applicability statement* for OSPF, which says that any router implementing a protocol other than simple IP-based routing must implement OSPF (this does not preclude a router implementing other protocols as well, of course). OSPF is described in RFC 1583, which obsoletes RFC 1247. OSPF implementations based on RFC 1583 are backward-compatible with implementations based on RFC 1247 and will interoperate with them. Readers interested in the development of OSPF Version 2 from Version 1 should refer to Appendix F of RFC 1247 and Appendix E of RFC 1583.

OSPF is an interior routing protocol, but it is designed to operate with a suitable exterior protocol, such as BGP. OSPF is a complex standard when compared to RIP: RFC 1583 runs to 216 pages, whereas RIP, specified in RFC 1058 has 33 pages and RIP-2 (RFC 1723) adds only another 9. Much of the complexity of OSPF is directed towards a single purpose: ensuring that the topological databases are the same for all of the routers within an area. Because the database is the basis for all routing choices, if routers were to have independent databases, they could make mutually conflicting decisions.

OSPF communicates using IP (its protocol number 89). It is a *Link-State, Shortest Path First* protocol. OSPF supports different kinds of networks such as point-to-point networks, broadcast networks, such as Ethernet and token-ring, and non-broadcast networks, such as X.25. The OSPF specification makes use of *state machines* to define the behavior of routers complying with the protocol. Aspects of a router's operation which are important to OSPF, such as its network interfaces and its neighboring routers, are described as being in one of a finite number of *states* (for example, a neighbor may be in the down state). There is a separate state machine for each separate component (for example, two network interfaces have separate state machines) and the state of one is independent of the state of another. The possible states are sufficient to describe all possible conditions relevant to the protocol, so a state machine is always in one, and only one, of its possible states. State changes occur only as a result of *events*. There is a finite set of events for each type of state machine, which is sufficient to describe all possible occurrences relevant to the protocol.

The behavior of the state machine in response to an event is defined for all possible combinations of state and event. For example, if the state machine for a network interface experiences an InterfaceDown event, the state machine changes to the down state unconditionally. The InterfaceDown event is generated by the OSPF implementation whenever it receives an indication from a lower-level protocol that the interface is not functioning. See RFC 1583 for a complete description of each of the state machines, their possible states and events and the changes associated with them. (*Please refer Appendix 1 for definition related to OSPF operations*)

⁹⁷ RFC 1583 - OSPF Version 2

I. Overview of OSPF Operation ⁹⁸

The basic sequence of operations performed by OSPF routers is:

- Discovering OSPF neighbors
- Electing the Designated Router
- Forming adjacencies
- Synchronizing databases
- Calculating the routing table
- Advertising Link States

Routers will go through these steps when they first come up, and will repeat these steps in response to events, which occur in the network. Each router must perform each of these steps for each network it is connected to, except for the calculation of the routing table. Each router generates and maintains a single routing table for all networks.

Each of these steps is described in the following sections.

Discovering OSPF Neighbors : When OSPF routers start, they initiate and sustain relationships with their neighbors using the Hello protocol. The Hello protocol also ensures that communication between neighbors is bi-directional. Hello packets are sent periodically out to all router interfaces. Bi-directional communication is indicated when the router sees itself in the neighbor's Hello packet. On a broadcast network, Hello packets are sent using multicast; neighbors are then discovered dynamically. On non-broadcast networks each router that may potentially become a Designated Router has a list of all routers attached to the network and will send Hello packets to all other potential Designated Routers when its interface to the non-broadcast network first becomes operational.

Determining the Designated Router: This is done using the Hello protocol. A brief description of the process is given here. See RFC 1583 for a full description. The router examines the list of its neighbors, discards any with which it does not have bi-directional communication or which have a Router Priority of zero, and records the Designated Router, Backup Designated Router and Router Priority declared by each one. The router adds itself to the list, using the Router Priority configured for the interface and zero (unknown) for the Designated Router and Backup Designated Router values, if the calculating router has just come up.

Forming adjacencies: After a neighbor has been discovered, bi-directional communication ensured, and (on a multi-access network) a Designated Router elected, a decision is made regarding whether or not an adjacency should be formed with the neighbor:

- ✎ On multi-access networks, all routers become adjacent to both the Designated Router and the Backup Designated Router.
- ✎ On point-to-point links (or virtual links), the router always forms an adjacency with the router at the other end.

If the decision is made to not form an adjacency, the state of the neighbor communication remains in the 2-way state. Adjacencies are established using **Database Description** packets. These contain a summary of the sender's link state database. Multiple packets may be used to describe the database: for this purpose a poll-response procedure is used. The router with the higher router ID will become the master, the other will become the slave.

⁹⁸ RFC 1246 - Experience with the OSPF Protocol

Synchronization of databases :After the Database Exchange Process is over, each router has a list of those link advertisements for which the neighbor has more up-to-date instances. These are then requested in **Link State Request** packets. The response to a Link State Request packet is a Link State Update packet, which contains some or all of the link state advertisements requested. At most one Link State Request can be outstanding: if no response is received, the requester must retry the request. Link state advertisements come in five formats.

Type of Service : IP type of service that this metric refers to. RFC 1349 defines the possible TOS values in an IP header using a 4-bit sequence. OSPF encodes these by treating the sequence as a number and doubling it (there is a reserved bit of 0 immediately following the TOS value field in the IP datagram, so OSPF allows for its future inclusion in the TOS value). There are five defined values:

OSPF	RFC 1349	Type of service
0	0000	Normal service
2	0001	Minimize monetary cost
4	0010	Maximize reliability
8	0100	Maximize throughput
16	1000	Minimize delay

Table 5 : Type of Service Values

Calculating the routing table :Using its attached areas' link state databases as input, a router runs the SPF algorithm to build its routing table. The routing table is always built from scratch: updates are never made to an existing routing table. An old routing table is not discarded until changes between the two tables have been identified.

Advertising Link States : A router periodically advertises its link state, so the absence of a recent advertisement indicates to a router's neighbors that the router is down. All routers, which have established bi-directional communication with a neighbor, run an inactivity timer to detect such an occurrence. If the timer is not reset, it will eventually pop, and the associated event places the state machine corresponding to that neighbor in the down state. This means that communication must be re-established from the beginning, including re-synchronization of databases. A router also re-issues its advertisements when its state changes.

A router can issue several link state advertisements into each area. These are propagated throughout the area by the flooding procedure. Each router issues a Router Links Advertisement. If the router is also the Designated Router for one or more of the networks in the area, then it will originate Network Links Advertisements for those networks. Area border routers issue one Summary Link Advertisement for each known inter-area destination. AS boundary routers originate one AS External Link Advertisement for each known external destination. Destinations are advertised one at a time so that the change in any single route can be flooded without reflooding the entire collection of routes. During the flooding procedure, a single Link State Update packet can carry many link state advertisements.

II. Summary of Features for OSPF

OSPF is a complex routing protocol, as will be clear from the preceding sections. The benefits of this complexity (over RIP) are as follows:

- Because of the synchronized Link State databases, OSPF routers will converge much faster than RIP routers after topology changes. This effect becomes more pronounced as autonomous systems get larger.

- It includes *Type of Service (TOS)* routing that is designed to compute separate routes for each type of service. For any destination, multiple routes can exist, each route supporting one or more different Types of Service.
- It uses weighted metrics for different speed links. For example, a T1 (1.544 Mbps) link might be assigned a metric of 1 and a 9600 bps SLP link might be assigned a metric of 10.
- It provides load balancing since an OSPF gateway can use more than one equal minimum cost path to a destination.
- A subnet mask is associated with each route, allowing variable-length subnetting and superneting (CIDR).
- All exchanges between routers may be authenticated by the use of passwords.
- OSPF supports host-specific routes, network-specific routes as well as subnet routes.
- OSPF allows contiguous networks and hosts to be grouped together into areas within an AS, simplifying the topology and reducing the amount of routing information which must be exchanged. Knowledge of an area's topology remains hidden from other areas.
- It minimizes broadcasts by allowing a more complex graph topology in which multi-access networks have a *Designated Router*, which is responsible for describing that network to the other networks in the area.
- It allows external routing information exchange, that is, routing information learned from another autonomous system.
- It allows routing within the AS to be configured according to a virtual topology rather than just the physical interconnections. Areas can be joined using virtual links, which cross other areas without requiring complicated routing.
- It allows the use of point-to-point links without IP addresses, which can save scarce resources in the IP address space.

III. OSI Intermediate System to Intermediate System (IS-IS) ⁹⁹

Intermediate System to Intermediate System (IS-IS) is a similar protocol to OSPF: it also uses a Link State, Shortest Path First algorithm. However, IS-IS is an OSI protocol used for routing **Connectionless Network Protocol (CLNP)** packets within a routing domain. CLNP is the OSI protocol most comparable to IP.

Integrated IS-IS extends IS-IS to encompass TCP/IP. Integrated IS-IS is described in RFC 1195. Its goal is to provide a *single* (and efficient) routing protocol for TCP/IP *and* for OSI. Its design makes use of the OSI IS-IS routing protocol, augmented with IP-specific information, and provides explicit support for IP subnetting, variable subnet masks, TOS-based (type of service) routing, and external routing. It provides a provision for authentication information. Integrated IS-IS is based on the same SPF routing algorithm as OSPF.

Integrated IS-IS does not employ mutual encapsulation of IP and CLNP packets: both types are forwarded "as-is", nor does it change the behavior of the router as expected by either protocol suite. Integrated IS-IS behaves like an IGP in a TCP/IP network and in an OSI network. The only change is the addition of additional IP-related information. IS-IS uses the term **Intermediate System (IS)** to refer to an IS-IS router, but we shall use the term router, since this is freely used in the Integrated IS-IS standard. IS-IS groups networks into domains in a fashion that is analogous to OSPF. A *routing domain* is analogous to an Autonomous system, and it is subdivided into areas just like OSPF.

Here is an overview of some of the more important aspects of IS-IS routing. Where possible, comparisons are made with equivalent concepts used in OSPF but it is dangerous to draw too close a parallel, since there are fundamental differences between the two protocols.

⁹⁹ RFC 1195 - *Integrated IS-IS*

- ✎ Routers are divided into Level 1 routers, which know nothing of the topology outside their areas, and Level 2 routers, which do know about the higher-level topology, but know nothing about the topology inside the areas unless they are also Level 1 routers.
- ✎ A Level 1 router may belong to more than one area, but unlike OSPF this is not for routing purposes but for ease of management of the domain, and would normally be short term. A Level 1 router recognizes another as a neighbor if they are both in the same area.
- ✎ A Level 2 router recognizes all other Level 2 routers as neighbors. A Level 2 router may also be a Level 1 router in one area, but not more.
- ✎ A Level 1 router in IS-IS cannot have a link to an external router (in OSPF an internal router can be an AS border router).
- ✎ There is a Level 2 backbone containing all Level 2 routers, but unlike in OSPF, it must be physically connected.
- ✎ The OSI addressing scheme explicitly identifies the target area for a packet, allowing simple selection of routing choices as follows:
 - ∞ Level 2 routers route towards the area without regard to its internal structure.
 - ∞ Level 1 routers route towards the destination if it is within their area, or to the nearest Level 2 router if it is not.
 - ✎ Multi-access networks use a Designated Router concept. To avoid the " $n(n-1)/2$ " problem described under OSPF, IS-IS implements a *pseudonode* for the LAN. Each LAN-attached router is regarded as having a link to the pseudonode but no link to any of the other routers on the LAN. The Designated Router then acts on behalf of the pseudonode.

Integrated IS-IS permits considerable mixing of the two protocol suites, subject to certain restrictions on the topology. Three types of router are defined:

IP-only : A router which uses IS-IS as the routing protocol for IP and does not otherwise support OSI protocols (for example, such routers would not be able to forward OSI CLNP packets).

OSI-only : A router, which uses IS-IS as the routing protocol for OSI but not IP.

Dual : A router, which uses IS-IS as a single integrated routing protocol for both IP and OSI.

It is possible to have a mixed domain containing IS-IS routers, some of which are IP-only, some OSI-only and some are dual. Each area within a mixed domain is configured to be OSI, IP or dual. Areas that are to carry mixed traffic must have dual routers for all of the Level 1 routers. Similarly, the Level 2 routers in a mixed domain must all be dual routers if mixed traffic is to be routed between areas.

IV. Co-existence of TCP/IP and OSI Routing Protocols without IS-IS

As its name suggests, Integrated IS-IS offers an integrated routing solution for multi-protocol networks. OSPF, like other TCP/IP routing protocols, uses an approach termed ***Ships In the Night (SIN)*** to handle coexistence issues. In the SIN approach, each multiprotocol router runs a separate process for each network layer (IP and OSI). A SIN router allows network managers to insert new SIN-based routing protocols, such as OSPF, one by one in the network, but the protocols exist independently of one another, and their frames pass each other like ships in the night.

Since the customer base of independent router vendors remains largely TCP/IP-focused, most of these vendors are choosing, for now, to stick with SIN even if it means their routers will not be able to work in OSI networks. A few of them have announced that they will support Integrated IS-IS in the future.

4.2.4 Exterior Routing Protocols ¹⁰⁰

Exterior Routing Protocols or **Exterior Gateway Protocols (EGPs)** are used to exchange routing information between routers in different autonomous systems. The term Exterior Routing Protocol has no abbreviation in common use, so we shall use the abbreviation EGP as is usual in TCP/IP literature.

Two EGPs are in common use

- Exterior Gateway Protocol.
- Border Gateway Protocol.

EGP is being replaced progressively by BGP. We shall discuss the two protocols in turn.

4.2.4.1 Exterior Gateway Protocol (EGP) ¹⁰¹

EGP is a *standard protocol*. Its status is *recommended*. The **Exterior Gateway Protocol (EGP)** is the protocol used for exchange of routing information between exterior gateways (not belonging to the same autonomous system).

EGP gateways may only forward reachability information for networks within their autonomous system. This routing information must be collected by this EGP gateway, usually via an **Interior Gateway Protocol (IGP)**, used to exchange information between gateways within an autonomous system. EGP is based on periodic polling using *Hello/I Hear You* message exchanges, to monitor neighbor reachability and poll requests to solicit update responses. EGP restricts exterior gateways by allowing them to advertise only those destination networks reachable entirely within that gateway's autonomous system. Thus, an exterior gateway using EGP passes along information to its EGP neighbors but does not advertise reachability information about its EGP neighbors (gateways are neighbors if they exchange routing information) outside the autonomous system. It has three main features:

- It supports a *neighbor acquisition protocol*. Two gateways may be regarded as neighbors if they are connected by an *internet*, which is transparent to them. EGP does not specify the way in which one gateway initially decides that it wants to become a neighbor of another. To become a neighbor it must send an *Acquisition confirm* message as a response to an *Acquisition Request* message. This step is necessary to obtain routing information from another gateway.
- It supports a *neighbor reachability protocol*. It is used by a gateway to keep real-time information as to the reachability of its neighbors. The EGP protocol provides two message types for that purpose: a *Hello* message and an *I Hear You* message (response to Hello).
- It supports update messages (also called **Network Reachability (NR) messages**) that carry routing information. No gateway is required to send NR messages to any other gateway, except as a response to a *poll request*.

As indicated above, the EGP routing information messages associate a *distance* qualifier to each route. But, EGP *does not interpret these distance values*. They merely serve as an indication of the reachability or unreachability of a network (a value of 255 means that the network is unreachable). The value cannot be used to compute the shorter of two routes unless those routes are both contained within a single autonomous system. For this reason, EGP cannot be used as a routing algorithm. As a result there will be only one path from the exterior gateway to any network.

EGP is gradually being replaced by the more functional Border Gateway Protocol (BGP). For a more detailed description of BGP refer to the next section.

¹⁰⁰ John T. Moy, *OSPF - Anatomy of an Internet Routing Protocol*, Addison-Wesley, 1998

¹⁰¹ RFC 904 - Exterior Gateway Protocol - Specification

4.2.4.2 Border Gateway Protocol (BGP) ¹⁰²

There are four different versions of BGP defined. Where BGP is specified without a version number, it normally refers to BGP Version 3 unless the document pre-dates the publication of the BGP-3 definition. BGP-3 is described in this section and BGP-4. BGP-1 and BGP-2, described in RFC 1105 and RFC 1163, are obsolete. Changes from BGP-1 and BGP-2 to BGP-3 are documented in Appendix 2 and 3 of RFC 1267.

I. Border Gateway Protocol Version 3 (BGP-3)

BGP-3 is a *draft standard protocol*. Its status is *elective*. It is described in RFC 1267. BGP-3 is an **inter-Autonomous System (inter-AS)** routing protocol based on experience gained from EGP. Unlike other routing protocols which communicate via packets or datagrams, BGP-3 is *connection oriented*; it uses TCP as a transport protocol. The well-known port number is 179.

Recall that the EGP was designed as a protocol to exchange *reachability* information between autonomous systems, rather than a true *routing* protocol. Because inter-AS routing information is not available, EGP cannot detect the presence of a loop caused by a set of EGP routers all believing that one of the others can reach another AS to which none of them is connected. A further problem with EGP has to do with the amount of information exchanged; as the number of IP networks known to the NSFNET increased, the size of the EGP Neighbor Reachability (NR) messages became quite large and the amount of time it took to process them became significant.

BGP-3 has replaced EGP in the NSFNET backbone for these reasons. However, BGP-3 as described in RFC 1268 does not require the NSFNET or any other backbone to play any central role. Compare this to the Core System role played by the ARPANET in the early days of the Internet. Instead, BGP-3 views the Internet as an arbitrary collection of autonomous systems, and it does not take account of the internal topology of an AS nor of the IGP (or possibly multiple IGPs) used within an AS. (*Please refer appendix 2 for BGP definitions*)

Routing Policy

A set of rules constraining routing to conform to the wishes of the authority which administers the AS. Routing policies are not defined in the BGP-3 protocol, but are selected by the AS authority and presented to BGP-3 in the form of implementation-specific configuration data. Routing policies may be selected by the AS authority in whatever way that authority sees fit. For example:

- ☛ A multihomed AS can refuse to act as a transit AS. It does this by not advertising routes to networks other than those directly connected to it.
- ☛ A multihomed AS can limit itself to being a transit AS for a restricted set of adjacent ASs. It does this by advertising its routing information to this set only.
- ☛ An AS can select which outbound AS should be used for carrying transit traffic. An AS can also apply performance-related criteria when selecting outbound paths:
- ☛ An AS can optimize traffic to use short AS paths rather than long ones.
- ☛ An AS can select transit routes according to the service quality of the intermediate hops. This service quality information could be obtained using mechanisms external to BGP-3.

It can be seen from the definitions above that a stub AS or a multihomed AS has the same topological properties as an AS in the ARPANET architecture: that is it never acts as an intermediate AS in an inter-AS route. In the ARPANET architecture, EGP was sufficient for such an AS

¹⁰² RFC 1265 - BGP Protocol Analysis

to exchange reachability information with its neighbors, and this remains true with BGP-3. Therefore, a stub AS or a multihomed AS may continue to use EGP (or any other suitable protocol) to operate with a transit AS. However, RFC 126B recommends that BGP-3 is used instead of EGP for these types of AS because it provides an advantage in bandwidth and performance. Additionally, in a multihomed AS, BGP-3 is more likely to provide an optimum inter-AS route than EGP, since EGP only addresses reachability and not "distance".

Path Selection

Each BGP speaker must evaluate different paths to a destination from the border router(s) for an AS connection, select the best one that complies with the routing policies in force and then advertise that route to all of its BGP neighbors at that AS connection. BGP-3 is a vector-distance protocol but, unlike traditional vector-distance protocols such as RIP where there is a single metric, BGP-3 determines a preference order by applying a function mapping each path to a preference value and selects the path with the highest value. The function applied is generated by the BGP-3 implementation according to configuration information. Where there are multiple viable paths to a destination, BGP-3 maintains all of them but only advertises the one with the highest preference value. This approach allows a quick change to an alternate path should the primary path fail.

Routing Policies

RFC 126B includes a recommended set of policies for all implementations:

- ✎ A BGP-3 implementation should be able to control which routes it announces. The granularity of this control should be at least at the network level for the announced routes and at the AS level for the recipients. For example, BGP-3 should allow a policy of announcing a route to a specific network to a specific adjacent AS.
- ✎ BGP-3 should allow a weighting policy for paths. Each AS can be assigned a weight and the preferred path to a destination is then the one with the lowest aggregate weight.
- ✎ BGP-3 should allow a policy of excluding an AS from all possible paths. This can be done with a variant of the previous policy; each AS to be excluded is given an "infinite" weight and the route selection process refuses to consider paths of infinite weight.

AS Consistency

BGP-3 requires that a transit AS present the same view to every AS using its services. If the AS has multiple BGP speakers, they must agree on two aspects of topology: intra-AS and inter-AS. Since BGP-3 does not deal with intra-AS routing at all, a consistent view of intra-AS topology must be provided by the interior routing protocol(s) employed in the AS. Naturally, a protocol such as OSPF or Integrated IS-IS, which implement synchronization of router databases, lends itself well to this role. Consistency of the external topology *may* be provided by all BGP speakers in the AS having BGP sessions with each other, but BGP-3 does not require that this method be used, only that consistency be maintained.

Routing Information Exchange

BGP-3 only advertises routes that it uses itself to its neighbors. That is, BGP-3 conforms to the normal Internet hop-by-hop paradigm, even though it has additional information in the form of AS paths and theoretically could be capable of informing a neighbor of a route it would not use itself.

When two BGP speakers form a BGP session, they begin by exchanging their entire routing tables. Routing information is exchanged via UPDATE messages (see below for the format of these messages). Since the routing information contains the complete AS path to each listed destination in the form of a list of AS numbers in addition to the usual reachability and next hop information used in traditional vector distance protocols, it can be used to suppress routing loops and to eliminate the *counting-to-infinity* problem found in RIP. After BGP neighbors have performed their initial exchange of their complete routing databases, they only exchange updates to that information.

II. BGP OSPF Interaction ¹⁰³

There is a *proposed standard protocol* with a status of *elective* defining how BGP-3 (an exterior routing protocol) should interact with OSPF (an interior routing protocol). Any host or router, which dynamically exchanges information between BGP-3 and OSPF, should adhere to this standard. It is described in *RFC 1654 - BGP OSPF Interaction*.

BGP OSPF interaction covers the conversion from OSPF fields in an External Links Advertisement to BGP path attributes, and vice versa, for three properties of a route definition.

OSPF Field	BGP Attribute
Type and Metric	INTER-AS METRIC
External Tag	ORIGIN and AS_PATH
Forwarding Address	NEXT HOP

Table 6: BGP OSPF Attribute-Field Mapping

The standard defines how these mappings should be done and what restrictions there are on what may be done automatically. Please refer to the RFC for more information.

III. Border Gateway Protocol Version 4 (BGP-4) ¹⁰⁴

BGP-4 is a *proposed standard protocol*. Its status is *elective*. It is described in RFC 1654. The main changes are to support *supernetting* or **Classless Inter-Domain Routing (CIDR)** which is described in CIDR. In particular BGP-4 supports IP prefixes and path aggregation. Because CIDR is radically different from the normal Internet routing architecture, BGP-4 is incompatible with BGP-3. However, BGP does define a mechanism for two BGP speakers to negotiate a version, which they both understand. This is done using the OPEN message. Therefore, it is possible to implement "multi-lingual" BGP speakers, which will allow inter-operation of BGP-3 and BGP-4.

The following items identify the major changes between BGP-3 and BGP-4.

- The BGP Version Number in the header field is 4.
- CIDR removes the concept of a network class from inter-domain routing, replacing it with the concept of an IP prefix.
- The list of networks in an UPDATE message is replaced by *network layer reachability information (NLRI)*.
- BGP-4 introduces the aggregation of multiple routes or AS paths into single entries or *aggregates*. The use of aggregates can dramatically reduce the amount of routing information required.
- A new attribute for an AS path (ATOMIC_AGGREGATE) can be used to insure that certain aggregates are not de-aggregated. Another new attribute (AGGREGATOR) can be added to aggregate routes in order to advertise which AS and which BGP speaker within that AS caused the aggregation.
- BGP-4 conceptually models the data held by a BGP speaker into three sets of **Routing Information Bases (RIB)**: one set (Adj-RIBs-In) for data obtained from BGP neighbors; one RIB (Loc-RIB) for local data obtained by the operation of the local routing policies on the Adj-RIBs-In; and one set (Adj-RIBs-Out) for data that is to be advertised in update messages.
- BGP-4 allows negotiation of Hold Times on a per-connection basis so that both ends of a connection are using the same value.
- BGP-4's format of the UPDATE message has changed from that of BGP-3

¹⁰³ RFC 1654 - A Border Gateway Protocol 4 (BGP-4)

¹⁰⁴ RFC 1654 - A Border Gateway Protocol 4 (BGP-4)

4.2.4.3 Inter Domain Routing Protocol

The *Inter-Domain Routing Protocol (IDRP)* was first designed for use with the OSI family of protocols developed by the ISO, and is defined in ISO standard 10747. Since the most designers of BGP were also involved with IDRP design (making IDRP a descendant of BGP), the widespread opinion among IPv6 developers was that adopting IDRP was wiser than developing a new version of BGP for IPv6 use. Other reasons for the choice are :

- It does not have any OSI networking dependencies, even though it was developed for OSI network use.
- It was designed from the ground up for *multiprotocol* routing and can compute routing information from different address families.
- It is based on the same (well-tested) path-vector design as BGP, making it technically safe.

The main differences between BGP and IDRP are [Huit96]:

- BGP messages are exchanged over a TCP connection, IDRP messages are exchanged using bare datagram services.
- BGP is a single-address-family protocol. IDRP supports multiple simultaneous protocol families.
- BGP uses 16-bit autonomous system IDs, IDRP uses variable-length prefixes.
- BGP always describes the full list of autonomous systems that a path passes through, IDRP can use the concept of Routing Domain Confederations to aggregate this information.

The changes needed in IDRP to support IPv6 are more a matter of defining the content of certain fields than changing the protocol [Rek96]

4.2.5 New Routing Protocols ¹⁰⁵

Some new routing protocols are being developed that attempt to build on current knowledge of routing protocols. The most prominent of these is Nimrod, and its ATM offshoot called PNNI.

4.2.5.1 Nimrod

The Nimrod routing architecture is a new scalable routing architecture still in the design stage. It is geared towards IPv6 while not being tied to it, and supports dynamic internetworking with arbitrary network sizes, provides service-specific routing, and allows incremental deployment within an internetwork. The design philosophy of the Nimrod effort is "maximize the lifetime and flexibility of the architecture", with a secondary philosophy of specifying all field lengths as somewhat larger than can conceivably be used. Past history shows that large changes in numeric magnitude are not exceptional in computer science, with microprocessor address size and IPv4 address size serving as good examples [RFC1753].

The main goals of Nimrod are:

- Support a dynamic internetwork of arbitrary size by limiting the amount of routing information that must be known throughout the internetwork.
- Provide service-specific routing in the presence of multiple constraints imposed by service providers and users.
- Admit incremental deployment throughout an internetwork.

o meet these goals, Nimrod represents internetwork connectivity and services with *maps* which have different abstraction levels. It supports user-controlled route generation and selection based on these maps and on general traffic service requirements, and it also supports user-directed packet forwarding along established paths. Nimrod is a scalable architecture, that is to say it can perform routing either within a single domain or between multiple domains in effect, is it both an

¹⁰⁵ <http://www.pub.ro/~cmatei/network>

IGP and an EGP. The technology is not tied to IP, and can function equally well in an OSI environment.

Nimrod sees the internetwork as clusters of "entities" at various levels of abstraction. The entities can be hosts, routers, or other equipment, and the method of clustering them together is not mandated by Nimrod - a cluster can represent a local network, a larger collection of networks that are managed by a single authority, etc. Clusters can be grouped into other clusters, giving abstraction levels to the network "map". All elements of a cluster must satisfy one condition: connectivity. Any two entities within a cluster must be connected by at least one route that lies entirely within the cluster. Once a cluster is formed, connectivity and service information about it is stored (this is statistical information that describes the cluster "as a whole").

Each cluster selects which portion of the available routing information it advertises to the "outside world" and which portion it wants to receive. This is one of the key elements of Nimrod's scalability, since it allows portions of an internetwork to retain only the necessary amount of information. At the rate the Internet is growing, it is becoming more and more impractical to store *all* available routing information. Nimrod allows route generation according to specific constraints, but since this is a computation-intensive procedure it calculates such information only for entities that request it. Entities can also select their own route selection algorithms. To limit the amount of forwarding information that must be maintained in each router, Nimrod multiplexes multiple traffic flows with similar requirements over a single path. To meet the same goal, Nimrod retains information only for active traffic flows.

On the lowest level, Nimrod handles traffic between *endpoints*, which are identified by **endpoint identifiers (EID)**. These are globally unique bit strings with no topological significance. Regions of the network (clusters) are represented as *nodes*, with stored adjacency information telling which nodes connect to which. Network topology is stored in *maps*, which consists of nodes and adjacency information. These maps are used to calculate routes for messages. It is not expected that routers on the network have consistent maps (due to the information hiding nature of Nimrod and to delays in routing information propagation). Nimrod has been designed to prevent loops even in the case of inconsistent router maps.

"Addresses" are expressed as *locators* in Nimrod. Each node and each endpoint has a locator, with endpoints potentially having multiple locators. All calculations are performed on locators, EIDs are not used in routing decisions. A node is said to own the locators, which have the node's locator as prefix. A node can have a more detailed internal map, which a router can request from it if it wants to make detailed routing decisions. This internal map can recursively contain other internal maps, allowing the mapping of a whole internetwork into a single node. There is no defined "lowest level" of maps ("*it's turtles all the way down!*"). Nimrod is a fairly complex architecture, and a complete description of its operating modes (let alone protocols) is outside the scope of this document. A high-level description of Nimrod functionality can be found in.

4.2.5.2 PNNI

Private Network to Network Interface (PNNI) is an evolving routing architecture for use with ATM networks. PNNI is based on the Nimrod protocol suite being developed by the IETF, but it is a distinct protocol and not just a version of Nimrod.

PNNI has similar goals to Nimrod, namely scalability, support for policy-based routing, and user-level determination of the desired **Quality of Service (QoS)**. The basic units are *switches*, which are recursively grouped into *peer groups*. Like Nimrod, PNNI is map-based, and performs routing decisions based on internal maps which are kept as up-to-date as possible. Peer group leaders receive maps from the peer groups that it "contains", and forward their own maps to their (possible) parent peer groups. Like Nimrod, the maps contain resource information, which is used in route determination.

PNNI is still an evolving protocol. As of this writing, the packet formats and protocols have largely been settled but there are still lots of work to be done, especially in the support for policy routing

and authentication. While ATM networks currently feature very little in the structure of the Internet, the high-speed virtual circuits they offer will probably find use somewhere in the Internet community, and PNNI routing may have to be taken into account. As such, ATM is quite different from TCP/IP, with the former being a circuit-oriented design and the latter being a packet-oriented strategy. While some opponents see ATM as being based on multimedia needs as conceived 15 years ago, instead of the current philosophy of "lots bandwidth is all you need", the powerful financial investments made by major telecommunications companies into ATM technology cannot be overlooked.

4.3 IP - The Next Generation (IPng) ¹⁰⁶

The Internet has evolved and changed over the years from being an academic network to one, which is the nerve center of many business networks. With the changing nature of the Internet and business networks, the current *Internet Protocol (IP)* is rapidly becoming obsolete.

Until late 80s, the Internet was primarily an academic network providing support for rather simple applications such as file transfer, remote access and electronic mail. But, today we see the Internet has evolved through the years and is fast becoming a multimedia based, application rich environment with many corporate networks using it as the infrastructure for their day-to-day client/server environments by way of Intranets, Extranets and Globenet (may be the future). So, where is the Internet headed? Is it geared to cater to the growing demand?

All these developments have obviously outstripped the capability of the IP-based network, originally designed and engineered to survive the cold war, to supply the much desired functionality and services. The demand on the other hand was for an internetworked environment, which could support real time traffic, flexible congestion control schemes, and security features. None of these requirements are easily met with the existing IP. But, the impending force behind the new version of IP is that we are running out of IP addresses and the fixed 32-bit address is inadequate for the explosive growth of networks since early 90s.

4.3.1 Limitations of IPv4

Each TCP/IP network interface requires a unique IP address, which IP routers use to forward packets as needed across the various network cables that connect communicating systems. An IPv4 address, 32 bits long is capable of accommodating several million such interconnections, more than enough to handle the needs of the TCP/IP community throughout eternity - at least, as far ahead into eternity as anyone could have foreseen twenty years ago.

Yesterday's anticipated networks of tens of thousands of computers, however, have since blossomed into today's potential for *hundreds of millions* of connections. The Internet is fast becoming the whole world's backbone network, and as the world scrambles to connect to it, the IPv4 addressing scheme is gradually running out of gas.

One may wonder how, given that IPv4 can mathematically handle up to 2^{32} (over 4 billion) possible values, the community can possibly be running out of addresses. The problem, in fact, isn't that there aren't enough bits; it's in the way the bits are grouped under IPv4's simple network/host numbering scheme. IPv4 addresses suffer from two soon-to-be-fatal inefficiencies: wasteful address assignment, and excessive routing overhead.

First of all, the standard IPv4 address assignment system is inherently very wasteful. Following the IPv4 rules as they were originally conceived, any medium-sized company with more than 256 computers would apply for Class B addresses, and consequently tie up 64 thousand values. Large companies claiming Class A addresses tie up (and grossly underutilized) about 16 million of IPv4's

¹⁰⁶ Stephen A. Thomas, *IPng and the TCP/IP Protocols*, John Wiley & Sons, Inc. 1998

available addresses. Were certain emergency measures not taken a few years ago, the TCP/IP community would already be out of host numbers, with most claimed addresses going unused! As it stands, the stopgap introduction of the ***Classless Interdomain Routing Protocol (CIDR)*** a few years ago, which (among other things) permits the assignment of Class C addresses in consecutive blocks (aggregates) to build mini-Class-B networks, has bought some time for the IPv4 address pool.

Secondly, classifying millions of computers with just two hierarchical levels results in very large routing tables – incurring enormous processing overhead – in the Internet's interior routers. As originally conceived, an IP packet's network number identifies the organization that reserved it, and its host number identifies a specific network interface within that organization. This was a workable idea in the 70's, when relatively few network numbers identified the paths to most of the world's computers, and Internet routers needed to advertise and maintain only a few thousand entries in their routing tables. If the original IPv4 architecture were in place today, the Internet's routers would be melting under the strain of maintaining millions of paths to Class C networks in individual households! Again, CIDR to the rescue: its address aggregation scheme flattened the growth of the Internet's routing tables, and bought some more time for IPv4.

So, how much time do we have left? Not much. Even with CIDR in place, various estimates now foresee IPv4's collapse occurring somewhere between the years 2000 and 2018. None of IP's founding mothers and fathers ever expected to outlive their 32-bit creation, but there's no longer any doubt about it: the future of the Internet demands nothing less than the immediate and fundamental reconsideration of the Internet Protocol.

4.3.2 The IPv6 Challenge: Change while Staying the Same ¹⁰⁷

Where IPv4's two-level addressing hierarchy failed to foresee the demographics of modern networks, IPv6 begins there.

There are three kinds of network users:

- ☞ People who work within an organization that is connected to the Internet, and are considered members of their organization's network (an organization's internal network is today referred to as an *intranet*). There can be millions of such organizations worldwide, each with potentially millions of members. As Internet members, each intranet must have its own unique address. Within an intranet, each interface address (host number) must also be unique.
- ☞ People who work within an organization that are not connected to the Internet, and can be considered members of their organization's network. There can likewise be millions of people in this situation, scattered among millions of organizations, with an added wrinkle: by simply signing a service contract and installing some network hardware, they can all be made part of the Internet community instantly. How many Intranet addresses will they need to change to make them unique within the Internet community?
- ☞ Individual network users, working from home or while traveling, connected through telephone lines or packet radio links. What network and host numbers should they use, and where? If they're ordinary citizens, how can they reasonably be expected to administer their addresses as they move from place to place?

Clearly, IPv4's three Unicast address classes don't "map" very neatly to any of these users. Developed over many years of careful design and exhaustive review, the IPv6 addressing scheme is radically new, based on the demographic nature of the community it will serve. At the same time, it includes provisions for upward compatibility from and interoperability with today's IPv4 network architecture.

IPv6 addresses are 128 bits long. It's the IPv4 addressing scheme *squared*. It is large enough address space to assign a unique address to several billion per square meter of the Earth's surface.

¹⁰⁷ William Stallings, *IPv6: The New Internet Protocol*, IEEE Communications, July 1996

4.3.3 The Technical Case for IPv6 ^{108 109}

4.3.3.1 Comparison of IPv4 and IPv6 Headers

A good way to start an in-depth investigation of IPv6 is to compare the new streamlined IPv6 header with the current IPv4 header. Both headers carry version numbers and source/destination addresses, but as Figure 6 shows, the IPv6 header is considerably simplified, which makes for more efficient processing by routing nodes. Whereas the IPv4 headers are variable in length, all IPv6 headers have a fixed length of 40 bytes. This allows router software designers to optimize the parsing of IPv6 headers along fixed boundaries. Additional processing efficiencies have been realized by reducing the number of required header fields in IPv6. The classic IPv4 header contains 14 fields, whereas IPv6 only requires 8 fields.

One of the first IPv4 components to be discarded was the header length field, which is clearly no longer required due to the fixed header length of all IPv6 packets. The total length field of IPv4 has been retained in the guise of the IPv6 payload length field. But this field does not include the length of the IPv6 header, which is always assumed to be 40 bytes. The new payload length field can accommodate packets up to 64 KB in length. IPv6 routers will forward even larger packets, called "jumbograms", if the payload length field is set to zero and a special extension header is added, as discussed below.

The time-to-live field of IPv4 has been given a face-lift in the form of the IPv6 hop limit field. Although the names are different, both fields are used by routers to decrement a maximum hop value by 1 for each hop of the end-to-end route. The hop-limit field is set to the appropriate value by the source node. When the value in the hop limit field is decremented to zero, the packet is discarded. The IPv6 hop-count field will store a value of up to 8 bits or 255 hops, which should be more than adequate for even the largest of networks for the foreseeable future.

Version	IHL	Type of Service	Total Length	
Identification		Flag	Fragment Offset	
Time to Live	Protocol		Header Checksum	
Source Address				
Destination Address				
Options			Padding	

(a) IPv4 Packet Header

Version	Priority	Flow Label		
Payload length		Next Header	Hop Limit	
Source Address				
Destination Address				

(b) IPv6 Packet Header

Figure 53: IPv4 and IPv6 Header Formats

¹⁰⁸ David Lee, Danial L Lough, Scott Midkoff, *The next Generation of the Internet: Aspects of the IPv6*, IEEE Networks, February 1998

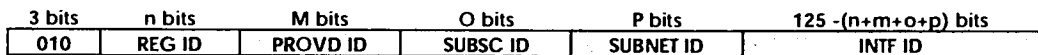
¹⁰⁹ RFC 1883 - Internet Protocol Version 6

In addition to the header length field, a number of basic IPv4 fields were eliminated from IPv6: type-of-service, fragment offset, identification, flags, checksum, and header length. The functionality of the IPv4 type-of-service field has been transferred to the two new IPv6 fields: flow control and priority. The IPv4 fragmentation fields (offset, identification, and flags) have been made optional headers in IPv6, as is discussed below. Finally, the IPv4 checksum fields have been abandoned in IPv6, due to the prevalence of error checking at other levels of the protocol stack. It is assumed that bad packets will be detected below, at the link layer, or above, at transport or higher layers.

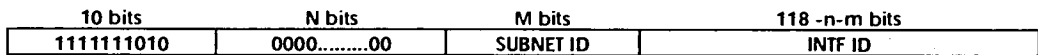
4.3.4 IPv6 Addresses - Unicast, Anycast and Multicast

The IPv6 addresses are assigned to individual interfaces on nodes, not to nodes themselves. A single interface may have multiple unique Unicast addresses and any of the Unicast address associated with a node's interface may be used to uniquely identify that node. IPv6 recognizes three types of addresses:

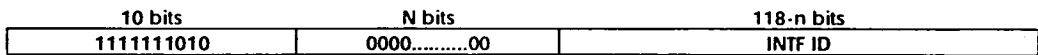
- ☛ **Unicast:** An identifier for a single interface. A packet sent to a Unicast address is delivered to the interface identified by that address.
- ☛ **Anycast:** An identifier for a set of interfaces (typically belonging to different nodes). A packet sent to an Anycast address is delivered to one of the interfaces identified by that address.
- ☛ **Multicast:** An identifier for a set of interfaces (typically belonging to different nodes). A packet sent to a multicast address is delivered to all interfaces identified by that address.



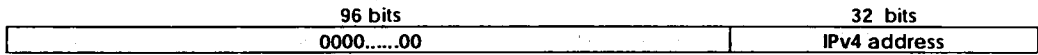
(a) Provider-based global Unicast Address



(b) Site-Local Address



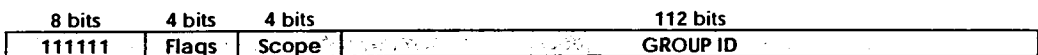
(c) Link-Local Address



(d) IPv4-compatible IPv6 address



(e) Subnet-router Anycast Address



(f) Multicast Address

Figure 54: IPv6 address formats

Unicast Addresses

Unicast addresses may be structured as on of provider based global, link local, site local, IPv4-compatible IPv6, and loopback.

- ☛ **Provider-based Unicast addresses** are assigned by an **Internet Service Provider (ISP)** to an organization, offering globally unique Internet addresses to all of the organization's members for easy integration within the worldwide Internet community. Devised as part of CIDR, the basic mechanism for assigning these addresses through ISPs is already in place.

- **Site-local-use addresses** can be assigned to the network devices within an isolated intranet. Later, should the organization decide to join the Internet community, all of its sitewide local addresses automatically become globally unique provider-based addresses with one administrative operation. This is much easier than the corresponding IPv4 procedure, which usually involves tediously changing every address on every computer within the Intranet.
- **Link-local-use addresses** are designed for use by individuals on a single communications link – such as mobile laptop computer users connected through phone lines (voice or ISDN) or radio links.
- **IPv4 Compatible IPv6 addresses** are designed to accommodate the lengthy period of transition during which IPv4 and IPv6 will coexist. This employs a technique known as Automatic Tunneling to carry IPv6 traffic on an IPv4 backbone.
- **Loopback Address** is used by a node to send an IPv6 packet to itself. The specific address used for this is 0:0:0:0:0:0:1

Anycast Addresses

An Anycast address enables a source to specify that it wants to contact any node from a group of nodes via a single address. A packet with such an address will be routed to the nearest interface in the group, according to the router's routing metrics. The Anycast address could refer to the group of routers associated with a particular provider or a particular subnets, thus dictating that the packet be routed through that provider or internet in the most efficient manner.

Anycast addresses are allocated from the same address space as Unicast addresses. Thus, members of an Anycast group must be configured to recognize that address, and routers must be configured to be able to map an Anycast address to a group of Unicast interface addresses. One particular form of Anycast address. The subnet-router Anycast address is illustrated in figure 68(e). Thus, the Anycast address is identical to a Unicast address for an interface on this subnetwork, with the interface ID portion set to zero. Any packet sent to this address will be delivered to one router on the subnetwork; all that is required is to insert the correct interface ID into the Anycast address to form a Unicast address.

Multicast Addresses

IPv6 includes the capability to address a predefined group of interfaces with a single multicast address. A packet with a multicast address is to be delivered to all members of the group. A multicast address format is shown in figure 68(f).

The flag is indicative of permanent or well-known multicast address assignments. The flag field consists of 4 bits with 3 leading zeros followed by a "T" bit.

- T=0 indicates a permanently assigned or well-known multicast address, assigned by the global internet numbering authority
- T=1 indicates a nonpermanently assigned, or transient address

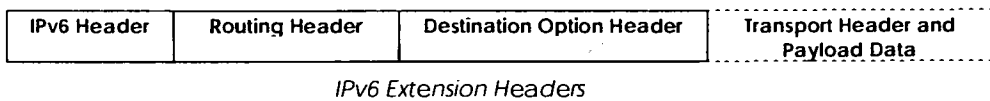
The scope value is used to limit the scope of the multicast group and the assigned values are: 0:reserved, 1:node-local, 2:link-local, 5:site-local, 8:organization-local, 14:global, 15: reserved and the rest are unassigned.

The group ID identifies a multicast group, either permanently or transiently, within the given scope. In case of a permanent multicast address, the address itself is independent of the scope field, but that field limits the scope of addressing for a particular packet. Non-permanently assigned multicast addresses are only meaningful within a given scope, enabling the same group ID to be reused, with different interpretations, at different sites. Multicasting is a useful capability in a

number of contexts. For example, it allows hosts and routers to send neighbor discovery messages only to machines that are registered to receive them, removing the necessity for all other machines to examine and discard irrelevant packets. Also, a multicast address can be assigned that has a scope of link-local with a group ID configured on all nodes on a LAN to be a subnet broadcast address.

4.3.5. Exceptional Extension Headers

To allow IPv4 packet headers the flexibility to carry optional information relevant to the routing process or host applications, IPv4 headers included an options field. This little-used field is carried by all IPv4 packets and is meant to convey information about security, source routing, and other optional parameters. The IPv4 options field has been replaced in IPv6 by flexible extension headers that travel after the primary IPv6 header and before the transport header and application payload. IPv6 extension headers are optional and provide a powerful means to support security, fragmentation, source routing, network management, and many other functions. An IPv6 packet can carry virtually any number of extension headers between the initial header and the higher layer payload. Figure shows encryption and fragmentation headers travelling after the primary IPv6 header and before the Transmission Control Protocol (TCP) header.



The IPv6 extension header architecture replaces the IPv4 options field and also impacts the protocol type field, which is currently used to indicate the type of protocol within the datagram's payload, e.g., TCP or User Datagram Protocol (UDP). IPv6 replaces the protocol type field with a next header field that indicates the protocol carried in the next extension or payload header (e.g., a TCP/UDP header or an IPv6 optional header).

The IPv6 standards groups have already defined a number of extension headers and have also created a suggested (but not mandatory) guideline for the order of header insertion.

The suggested order for extension headers is as follows:

- (Primary IPv6 header)
- Hop-by-Hop options header
- Destination options header-1
- Source Routing header
- Fragmentation header
- Authentication header
- IPv6 Encryption header
- Destination options header-2
- (Upper-layer headers)
- (Payload)

Each extension header typically occurs only once within a given packet, except for the destination header, as explained on the following page.

I. Hop-by-Hop Options Header

When present, this header carries options that are examined by intermediate nodes along the forwarding path. It must be the first extension header after the initial IPv6 header. Since all routers along the path read this header, it is useful for transmitting management information or debugging commands to routers. One currently defined application of the hop-by-hop extension header is the Router Alert option, which informs routers that the packet should be processed

completely by a router before it is forwarded to the next hop. An example of such a packet is a **Resource reSerVation Protocol's (RSVP)** resource reservation message.

II. Destination Options Headers

There are two variations of this header, each with a different position in the packet. The first incidence of this field is for carrying information to the first destination listed in the IPv6 address field. This header can also be read by a subsequent destination listed in the source routing header address fields. The second incidence of this header is used for optional information that is only to be read by the final destination. For efficiency, the first variation is typically located towards the front of the header chain, directly after the hop-by-hop header (if any). The second variation is relegated to a position at the end of the extension header chain, which is typically the last IPv6 optional header before transport and payload.

III. Source Routing Header

The IPv6 routing extension header is an incarnation of the source routing function supported currently by IPv4. This optional header allows a source node to specify a list of IP addresses that dictate what path a packet will traverse. IETF RFC 1883 defines a version of this routing header called "Type 0," which gives a sending node a great deal of control over each packet's route. Type 0 routing headers contain a 24-bit field that indicates how intermediate nodes may forward a packet to the next address in the routing header. Each bit in the 24-bit field indicates whether the next corresponding destination address must be a neighbor of the preceding address (1 = strict, must be a neighbor; 0 = loose, need not be a neighbor).

When routing headers are used for "strict" forwarding, a packet visits only routers listed in the routing header. In "loose" forwarding, unlisted routers can be visited by a packet. So if routers B and C are listed as strict but are not adjacent to each other (i.e., in order to get from B to C, a packet must pass some other router), packets will be dropped at B. This is a valuable feature when security and traffic control require that packets take a rigidly defined path. The strict/loose feature works in conjunction with another routing header field that contains a value equal to the total number of segments remaining in the source route. Each time a hop is made, this "segments left" field is decremented.

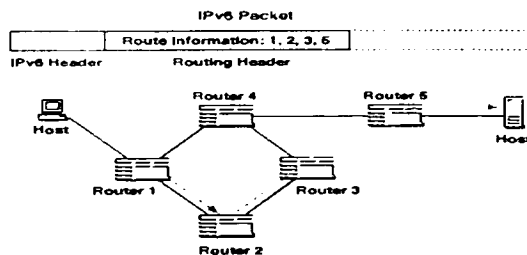


Figure 55: Source Routing Extension Header

When Type 0 routing headers are used, the initial IPv6 header contains the destination addresses of the first router in the source route, not the final destination address. At each hop, the intermediate node replaces this destination address with the address of the next routing node, and the "segments left" field is decremented.

IV. Fragmentation Header

IPv4 has the ability to fragment packets at any point in the path, depending on the transmission capabilities of the links involved. This feature has been dropped in IPv6 in favor of end-to-end fragmentation/reassembly, which is executed only by IPv6 source and destination nodes. Packet fragmentation is not permitted in intermediate IPv6 nodes. The elimination of the fragmentation field allows a more streamlined packet and better router performance for the majority of cases where fragmentation is not required. Today's networks generally support frame sizes that are large enough to carry typical IP packets without fragmentation. In the event that

fragmentation is required, IPv6 provides an optional extension header that is used by source nodes to divide packets into an arbitrary number of smaller units.

The IPv6 fragmentation header contains fields that identify a group of fragments as a packet and assigns them sequence numbers. Because IPv6 routers do not fragment packets between end nodes, the responsibility for sending the correct size packet is with the source node, which needs to determine the **Maximum Transmission Unit (MTU)** of the links in the end-to-end path. For instance, if two FDDI networks with 4500-byte MTUs are connected by an Ethernet with an MTU of 1500, then the source station must send packets that are no larger than 1500.

If higher level applications are using larger payloads, the source node can make use of the IPv6 fragmentation extension header to divide large packets into 1500-byte units for network transmission. The IPv6 destination node will reassemble these fragments in a manner that is transparent to upper layer protocols and applications. End nodes performing fragmentation can determine the smallest MTU of a path with the MTU path discovery process (e.g., RFC1191)

Typically, with this technique, the source node sends out a packet with an MTU as large as the local interface can handle. If this MTU is too large for some link along the path, an ICMP "Datagram too big" message will be sent back to the source. This message will contain a packet-too-big indicator and the MTU of the affected link. The source can then adjust the packet size downward (fragment) and retransmit another packet. This process is repeated until a packet gets all the way to the destination node. The discovered MTU is then used for fragmentation purposes. Although source-based fragmentation is fully supported in IPv6, it is recommended that network applications adjust packet size to accommodate the smallest MTU of the path. This will avoid the overhead associated with fragmentation/reassembly on source and destination nodes.

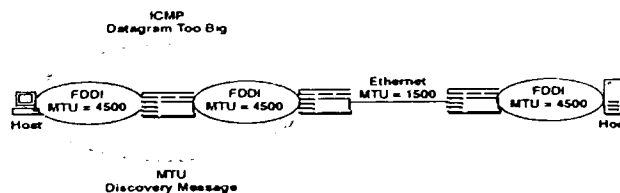


Figure 56: MTU Discovery Process

V. Authentication Header

The current lack of a standardized network-layer security scheme is one of the most glaring deficiencies of the IPv4 Internet. Regular press reports of hackers spoofing servers and snooping data streams have become a constant reminder of the damage that can be done to IP-based corporate networks. The IPv6 standard addresses this situation with two important extension headers, one that enables the authentication of IP traffic for security purposes, and another that fully or partially encrypts IP packets. Implementation of security at the IP level can benefit "security aware" applications, as well as "security ignorant" applications that don't take explicit advantage of security features.

The IPv6 authentication extension header gives network applications a guarantee that the packet did in fact come from an authentic source. This combats the increasingly common occurrence of hackers configuring an IP host to impersonate another, to gain access to secure resources. Such spoofing can be used to obtain valuable financial and corporate data and can give persons outside the enterprise control of servers for malicious purposes. With IPv6 authentication headers, hosts establish a standards-based security association that is based on the exchange of algorithm-independent secret keys (e.g., MD5).

In a client/server session, for instance, both the client and the server need to have knowledge of the key. Before each packet is sent, IPv6 authentication creates a checksum based on the key combined with the entire contents of the packet. This checksum is then re-run on the receiving side

and compared. This approach provides authentication of the sender and guarantees that an intervening party has not modified data within the packet. Authentication can take place between clients and servers or client and clients on the corporate backbone. It can also be deployed between remote stations and corporate dial-in servers to ensure that the perimeter of the corporate security is not breached.

VI. IPv6 Encryption Header

Authentication headers eliminate a number of host spoofing and packet modification hacks, but they do not prevent the nondisruptive reading (sniffing, snooping) of the content of packets as they traverse the Internet and corporate backbone networks. This is the area addressed by the **Encapsulating Security Payload (ESP)** service of IPv6 -- another optional extension header. Packets protected by the ESP encryption techniques can have very high levels of privacy and integrity - something that is not widely available with the current Internet, except with certain secure applications (e.g., private electronic mail and secure HTTP Web servers). ESP provides encryption at the network layer, making it available to all applications in a highly standardized fashion.

IPv6 ESP is used to encrypt the transport-layer header and payload (e.g., TCP, UDP), or the entire IP datagram. Both these methods are accomplished with an ESP extension header that carries encryption parameters and keys end-to-end. When just the transport payload is to be encrypted, the ESP header is inserted in the packet directly before the TCP or other transport header. In this case, the headers before the ESP header are not encrypted and the headers and payload after the ESP header are encrypted. This is referred to as "transport-mode" encryption. If it is desirable to encrypt the entire IP datagram, a new IPv6 and an ESP header are wrapped around all the fields (including the initial address fields) of the packet. Full datagram encryption is sometimes called "tunnel-mode" encryption because the contents of the datagram are only visible at the endpoints of the security tunnel.

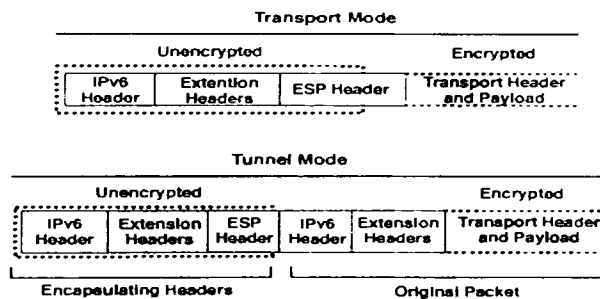


Figure 57: Tunnel Mode and Transport Mode of IPv6 Encryption

Fully encrypted datagrams are somewhat more secure than transport mode encryption because the headers of the fully encrypted packet are not available for traffic analysis.

For instance, full tunnel-mode encryption allows the addresses contained in IPv6 source routing headers to be hidden from packet sniffing devices for the public portion of a path. There is a considerable performance penalty for full encryption, due to the overhead and processing cost of adding an additional IPv6 header to each datagram. In spite of its cost, full ESP encryption is particularly valuable to create a security tunnel (steel pipe) between the firewalls of two remote sites. The full datagram encryption in the tunnel ensures that the various headers and address fields of encrypted packets will not be visible as traffic traverses the public Internet. Within the tunnel, only the temporary encapsulating address header is visible. Once through the tunnel and safely within a firewall, the leading ESP headers are stripped off and the packet is again visible, including any source routing headers required finishing the path.

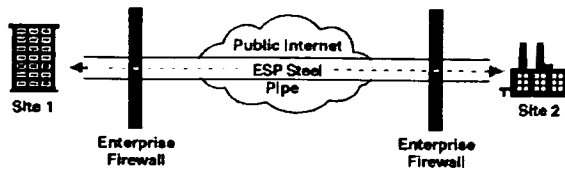


Figure 58: Firewalls and Steel Pipe

The encryption and authentication services of IPv6 work hand-in-hand to create a flexible and powerful security solution. In some cases an authentication header will be carried inside a fully encrypted or partially encrypted datagram, providing an additional layer of data integrity and verification of the sender's identification. In other cases, the authentication header may be placed in front of the encrypted transport-mode portion of the packet. This approach is desirable when the authentication takes place before decryption on the receiving end, which is the logical order in many cases. Taken together, the authentication and encryption services of IPv6 provide a robust, standards-based security mechanism that will play a critical role in the continuing expansion of commerce and corporate operations onto IP-based network fabrics.

4.3.6 The IPv6 Address Architecture ¹¹⁰

Much of the discussion of IPv4 versus IPv6 focuses on the relative size of the address fields of the two protocols (32 bits versus 128 bits). But an equally important difference is the relative abilities of IPv6 and IPv4 to provide an advanced hierarchical address space that facilitates efficient routing architectures. IPv4 was initially designed with a class-based address scheme, which divided address bits between network and host but did not create a hierarchy that would allow a single high-level address to represent many lower-level addresses. Hierarchical addressing systems work in much the same way as telephony country codes or area codes, which allow long-haul phone switches to efficiently route calls to the correct country or region using only a portion of the full phone number.

Class A	0	Network ID (7 bits)	Host ID (24 bits)
Class B	1	0	Network ID (14 bits) Host ID (16 bits)
Class C	1	1	0 Network ID (21 bits) Host ID (8bits)

Figure 59: IPv4 Address Classes

As the Internet grows, the non-hierarchical nature of the original IPv4 address space is proving to be increasingly inadequate. The limitations of IPv4 addressing are currently hampering both the local and global levels of internetworking. To combat IPv4 deficiencies at the local area network level, the subnetting technique has been developed to create a more granular division of large networks. With subnet addressing, a single network address can stand for a number of physical networks, which conserves address space considerably (e.g., a single Class C address can be used to access several physical networks).

At the level of large internet backbones and global routing, IPv4 addresses can be more efficiently aggregated with supernetting, a form of hierarchical addressing. With supernetting, backbone routers store a single address that represents the path to a number of lower level networks. This can considerably reduce the size of routing tables in backbone routers, which increases backbone performance and lowers the amount of memory and number of processing routers required. Subnetting and supernetting have been particularly useful in extending the viability of the IPv4 Class C addresses. Both of these techniques are made possible by pairing addresses stored in routers with bit masks that indicate which bits in an address are valid at the various levels of the hierarchy.

¹¹⁰ David Lee , Daniel L.Lough, The Internet Protocol version 6, IEEE Potentials, May 1998

The process of creating an IPv4 routing hierarchy was formalized in **Classless Interdomain Routing (CIDR)** which uses bit masks to allocate a variable portion of the 32-bit IPv4 address to network, subnet, or host. For instance, CIDR allows a number of (plentiful) Class C addresses to be summarized by a single prefix address, allowing Class C addresses to function in a similar way to hard-to-get Class A and Class B addresses. CIDR has extended the life of IPv4 and helped the Internet scale to its current size, but it has not been implemented in a consistent way across the Internet and enterprise networks. Consequently, the routing table efficiencies and address space conservation advantages of CIDR are not today fully realized, nor will they ever be fully realized, due to the legacy nature of IPv4 networks and the difficulty of restructuring them. IPv4 will continue to waste its already inadequate address space as it continues to burden routers with inefficient routes and excessively large routing tables.

Yet another downside of IPv4 is found at the departmental and workgroup level of internetworking, in the high administrative workload associated with maintaining subnet bit masks and host addresses within the subnet structure, particularly where there are large, dynamic populations of end users. When an end user is moved in the subnetting environment, careful attention must be paid to ensure that the host renumbering process does not disrupt connectivity at any level of the stack. The complexities and pitfalls of current subnetting methods can eventually make IPv4 less than viable in large organizations that experience ongoing growth of internetwork user populations.

4.3.6.1 The IPv6 Address Hierarchy

In a direct response to the experience gained from IPv4, IPv6 has been designed from the ground up to provide a highly scalable address space that can be partitioned into a flexible and efficient global routing hierarchy. At the top of this hierarchy, several international registries assign blocks of addresses to **Top Level Aggregators (TLA)**. These TLAs are essentially the public transit points (exchanges) where long-haul providers and telcos establish peer connections -- for example, MAE on the East Coast of the U.S.A., and Telehouse in London, England. TLAs allocate blocks of addresses to **Next Level Aggregators (NLA)**, which represent large providers and global corporate networks. When an NLA is a provider, it further allocates its addresses to its subscribers. Routing is efficient because NLAs that are under the same TLA will have addresses with a common TLA prefix. Subscribers with the same provider have IP addresses with an NLA common prefix.

Although a number of allocation schemes are possible within IPv6's huge address space, IETF designers favor an aggregation-based hierarchy because it combines the advantages of provider and geographic allocation approaches. Provider allocation divides the hierarchy along lines of large service providers, regardless of their location. Geographic allocation divides the hierarchy strictly on the basis of the location of providers/subscribers (as does the telephony system of country and area codes). But both of these approaches have their drawbacks because large backbone networks often don't conform strictly to geographic or provider boundaries. Some large networks, for instance, may connect to several ISPs. And many large networks span numerous countries and geographical regions.

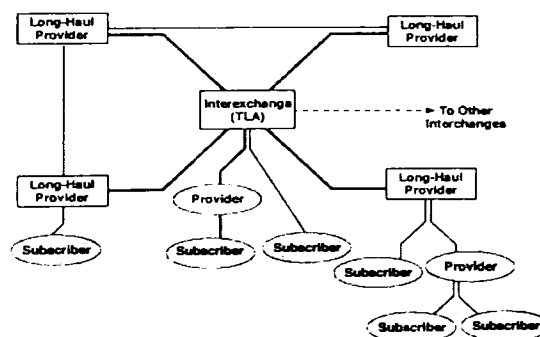


Figure 60: Aggregation-based Allocation Structures

Aggregation-based allocation is based on the existence today of a limited number of high-level exchange points, where large long-haul service providers and telco networks interconnect. The use of these exchange points to divide the IPv6 address hierarchy has a geographical component because exchanges are distributed around the globe. It also has a provider orientation because all large providers are represented at one or more exchange points.

As shown in figure, the first 3 address bits indicate what type of address follows (unicast, multicast, etc.). The next 13 bits are allocated to the various TLAs around the world. The following 32 bits are allocated to the next lower level of providers and subscribers.

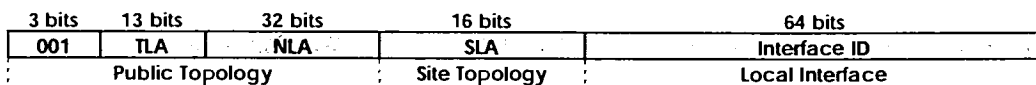


Figure 61: Aggregation-based IPv6 Addresses

Next level aggregators can divide the NLA address field so as to create their own hierarchy, one that maps well to the current ISP industry, in which smaller ISPs subscribe to higher level ISPs, and so on. This is accomplished by the ongoing subdivision of the 32-bit NLA field.

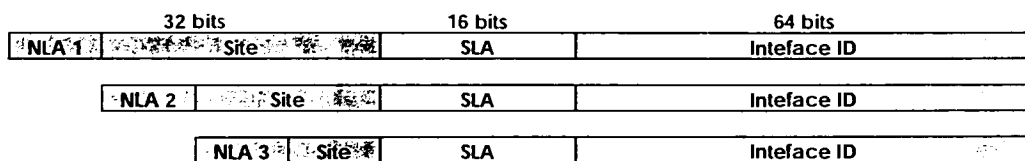


Figure 62: Subdividing the NLA Address Space

Following the NLA ID are fields for subscriber site networking information: **Site Level Aggregator (SLA)** and Interface ID. Typically, service providers supply subscribers with blocks of contiguous addresses, which are then used by individual organizations to create their own local addressing hierarchy and identify subnets and hosts. The 16-bit SLA field supports up to 65,535 individual subnets. The 64-bit Interface ID, which is used to identify an IPv6 interface on a network link, will typically be derived from the installed IEEE LAN adapter address.

Today's Internet backbone routers must maintain up to 40,000 or more routes. As the Internet continues to scale, IPv6's uniform application of hierarchical routing will likely be the only viable method for keeping the size of backbone router tables under control. With an aggregator-based address hierarchy, all of a subscriber's internal network segments can be reached through one or more high-level aggregation points. This allows backbone routers around the globe to efficiently summarize the routes to a customer's networks with high-level TLA address prefixes. Forwarding routes in highest level backbones can be quickly calculated by looking only at the TLA portion of the address. IPv6's large hierarchical address space also allows a more decentralized approach to IP address allocation. Service providers can allocate addresses independently from central authorities, encouraging global network growth and eliminating bureaucratic bottlenecks in the growth process.

Aggregation-based addresses are just part of the total address space that has been defined for IPv6. Other address ranges have been assigned to multicasting and to nodes that only require unique addressing within a limited area (site-local and link-local addresses).

Site- and link-local addresses are available for private, internal use by all enterprises, and are not allocated by public registry authorities. Site-local addresses are a flexible way for networks to start off with non-unique local addresses that are later made globally unique by adding a prefix. This has an advantage: if an ISP changes, site local addressing can remain the same because it is not directly interfaced to the outside world. Link local addresses can be used for applications that are limited in scope to a single link, and also for temporary "bootstrapping" of stations before they receive a globally unique address (more on this in the section below).

4.3.6.2 Host Address Configuration ¹¹¹

IPv6 clearly has large enough address architecture to accommodate Internet expansion for decades to come. But the usefulness of IPv6 addresses will be severely limited if they are not matched with equally advanced configuration and management services. Fortunately, there is a great deal of work underway to ensure that IPv6 hosts can have their addresses automatically configured and reconfigured in a cost-effective and manageable way. Automatic address configuration is a very necessary component of hierarchical routing fabrics because it supports cost-effective numbering and renumbering of large populations of IP hosts.

Autoconfiguration capabilities are important whether provider-based or geographic address allocation is in effect. Occasionally, it may be necessary to renumber every host within an organization, as would be the case with a company that relocated its operations (with geographic addressing) or changed to another service provider (with provider-based addressing). Configuration of IP addresses is a constant fact of life at the workgroup and department levels of large networked organizations. IP addresses need to be configured for new hosts, for hosts that change location, and for hosts connected to physical networks that receive address modification (e.g., a new prefix). In addition to these traditional requirements for configuration, new requirements are emerging as large numbers of hosts become highly mobile.

The process of autoconfiguration under IPv6 starts with the **Neighbor Discovery (ND)** protocol. ND combines and refines the services provided in the IPv4 environment by **Address Resolution Protocol (ARP)** and **Internet Control Message Protocol (ICMP)**. Although it has a new name, ND is actually just a set of complementary ICMP messages that allow IPv6 nodes on the same link to discover link layer addresses and to obtain and advertise various network parameters and reachability information. In a typical scenario, a host starts the process of autoconfiguration by self-configuring a link-local address to use temporarily. This address can be formed by adding a generic local address prefix to a unique token (typically the host's IEEE LAN interface address). Once this address is formed, the host sends out a ND message to the address, to ensure that it is unique. If no ICMP message comes back, the address is unique. If a message comes back indicating that the link-local address is already in use, then a different token is used (e.g., an administrative token or a randomly generated token).

Using the new link local address as a source address, the host then sends out a ND router solicitation request. The solicitation is sent out using the IPv6 multicast service. Unlike the broadcasted ARPs of IPv4, IPv6 ND multicast solicitations are not necessarily processed by all nodes on the link, which can conserve processing resources in hosts. (IPv6 currently defines several permanent multicast groups for finding resources on the local node or link, including all-routers group, an all-hosts group, and a **Dynamic Host Configuration protocol (DHCP)** server group). Routers respond to the solicitation messages from hosts with a unicast router advertisement that contains, among other things, prefix information that indicates a valid range of addresses for the subnet. Routers also send these advertisements out periodically to local multicast groups, whether or not they receive solicitations. ND message exchange is shown in figure.

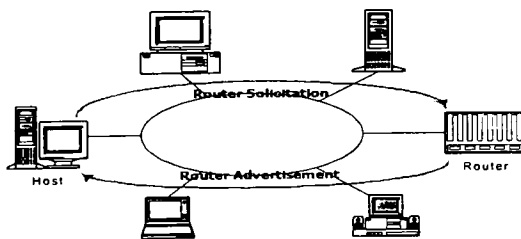


Figure 63: ND Message Exchange

¹¹¹ RFC 1884 - IPv6 Addressing Architecture

Using the router advertisement message, the router can control whether hosts use stateless or stateful autoconfiguration methods. In the case of stateful autoconfiguration, the host will contact a DHCP or similar address server, which will assign an address from a manually administered list. DHCP is increasingly popular for autoconfiguration in IPv4 networks and the standard is being extended to the IPv6 environment. With the stateless approach, a host can automatically configure its own IPv6 address without the help of a stateful address server or any human intervention. The host uses the globally valid address prefix information in the router advertisement message to create its own IPv6 address. This process involves the concatenation of a valid prefix with the host's link layer address or a similar unique token. As long as the token is unique and the prefix received from the router is correct, the newly configured IP address should provide reachability for the host that extends to the entire enterprise and the Internet at large.

The advantages of stateless autoconfiguration are many. For instance, if an enterprise changes service providers, the prefix information from the new provider can be propagated to routers throughout the enterprise, and hence to all stateless autoconfiguring hosts. Hypothetically, if all hosts in the enterprise use IPv6 stateless autoconfiguration, the entire enterprise could be renumbered without the manual configuration of a single host. At a more modest level, workgroups with substantial move/change activity also benefit from stateless autoconfiguration because hosts can receive a freshly configured and valid IP number each time they connect and reconnect to the network.

To support the growing universe of mobile computing devices, IETF workers have formulated a draft plan to allow IPv6 hosts to maintain connectivity with their "home" IP address while on the road. Before leaving on a trip, users will be able to request that their local router forward all traffic destined for their home IP address to a temporary "foreign" address. The foreign address is typically autoconfigured by concatenating the mobile host's token (e.g., a LAN adapter address) with the prefix of the foreign network. At each stop on the trip, a new prefix can be used. This approach reduces the complication involved when name servers try to resolve names to addresses of mobile computers that are often not at their home network. With the IP forwarding features, **Domain Name Service (DNS)** entries can remain essentially untouched, even if a host moves to the other side of the world and all points in between.

To further facilitate host renumbering in highly dynamic situations, IPv6 has a built-in mechanism to create a graceful transition from old to new addresses. Fundamental to this mechanism is the ability of IPv6 nodes to support multiple addresses per interface. IPv6 addresses assigned to an interface can be identified as valid, deprecated, or invalid. In the renumbering process, an interface's address would become deprecated when a new address was automatically assigned (e.g., in the case of network renumbering). For a period of time after the new (valid) address is configured, the deprecated address continues to send and receive traffic. This allows sessions and communications based on the older address to be finished gracefully. Eventually the deprecated address becomes invalid and the valid address is used exclusively. Multiple IP addresses allow renumbering to occur in a highly dynamic, nondisruptive manner that is virtually transparent to end users and applications.

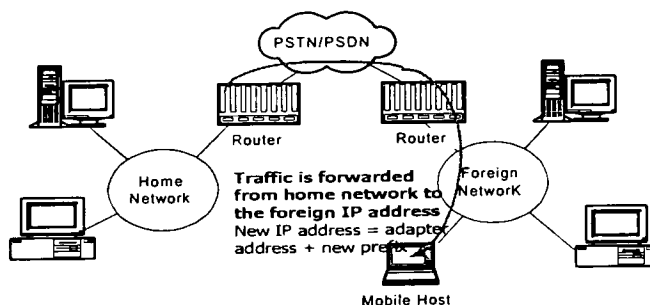


Figure 64: Forwarded IP Traffic

The above described stateless autoconfiguration process is particularly suited to conventional IP/LAN environments with 48-bit addressing and native multicast services. Other network environments with different link characteristics may require modified or alternative configuration techniques. For instance, current **Asynchronous Transfer Mode (ATM)** networks do not inherently support multicast services or 48-bit IEEE addressing, due to the use of virtual circuits and telephony-style calling numbers. Multicasting solutions for ATM are seen in the emerging **Multicast Address Resolution Server (MARS)** that is being developed for IPv4 multicast over ATM. Plans are being devised to use MARS-style functionality to extend the IPv6 Neighbor Discovery protocol across ATM networks. This would allow network renumbering and stateless autoconfiguration to take place seamlessly in hybrid ATM/IPv6 fabrics.

4.3.7 Other Protocols and Services

The preceding discussion focuses on some of the more innovative and radical changes that IPv6 brings to internetworking. In many other areas, protocols and services will operate much the same as they do in the current IPv4 regime. As the industry moves to IPv6, DHCP and DNS servers are being modified to accommodate 128-bit addresses, but in terms of basic functionality, there will be little change. This is also generally true for interior and exterior routing protocols.

Open Shortest Path First (OSPF) protocol, the cornerstone of high-performance, standards-based internetworking, is the IETF recommended **Interior Gateway Protocol (IGP)** for IPv6. OSPF is being updated with full support for IPv6, allowing routers to be addressed with 128-bit addresses. 128-bit records will replace the 32-bit link-state records of current OSPF. In general, the OSPF IPv6 link-state database of backbone routers will run in parallel with the database for IPv4 topologies. In this sense, the two versions of OSPF will operate as "ships in the night," just as the routing engines for NetWare, DECnet, AppleTalk, and other protocols coexist in the same router without major interaction. Given the limited nature of the OSPF IPv6 upgrade, those engineers and administrators who are proficient in OSPF for IPv4 should have no problems adapting to the new version. An updated version of RIP is also available, referred to as **Routing Information Protocol Next Generation (RIPng)**.

As with the interior gateway protocols, work is underway to create IPv6-compatible versions of the exterior gateway protocols that are used by routers to establish reachability across the Internet backbone between large enterprises, providers, and other autonomous systems. Today's backbone routers use the **Border Gateway Protocol (BGP)** to distribute CIDR-based routing information throughout the Internet. BGP is known by providers and enterprises and has a large installed base. Consequently, BGP has the inside track for IPv6. Currently, work is underway to define BGP extensions that will allow it to be used to exchange reachability information based on the new IPv6 hierarchical address space.

4.3.8 Transition Mechanisms from IPv4 to IPv6 ¹¹²

The problem is how to convert the Internet into IPv6, without disrupting the operation of the existing IPv4 network. The transition will proceed in two phases. At the end of phase 1 there will be both IPv4 and IPv6 hosts and routers. At the end of phase 2 there will only be IPv6 hosts and routers. That means, that the **SIT (Simple IPv6 Transition)** should at least support the following points:

- IPv6 and IPv4 hosts can interoperate
- IPv6 routers and hosts can be deployed in the Internet in a highly diffuse and incremental fashion, with few interdependencies
- the transition should be as easy as possible for end-users, system administrators, and network operators to understand and carry out.

¹¹² Stephen A. Thomas, *IPng and the TCP/IP Protocols*, John Wiley & Sons, Inc. 1998

The *SIT* provides a number of features, including:

- Incremental upgrade.
Existing installed IPv4 hosts and routers may be upgraded to IPv6 at any time without being dependent on any other hosts or routers being upgraded.
- Incremental deployment.
New IPv6 hosts and routers can be installed at any time without any prerequisites.
- Easy Addressing.
When existing installed IPv4 hosts or routers are upgraded to IPv6, they may continue to use their existing address. They do not need to be assigned new addresses.
- Minimal upgrade dependencies.
The only prerequisite to upgrading hosts to IPv6 is that the DNS server must first be upgraded to handle IPv6 address records. There are no prerequisites to upgrading routers.
- Low start-up costs.
Little or no preparation work is needed in order to upgrade existing IPv4 systems to IPv6, or to deploy new IPv6 systems.

The following mechanisms are employed in *SIT* to realize the above features:

- Use of the dual IP layer (IPv4 and IPv6) technique in hosts and routers for direct interoperability with nodes implementing both protocols.
- Two IPv6 addressing structures that embed an IPv4 addresses within IPv6 addresses.
- A mechanism for tunneling IPv6 packets over IPv4 routing infrastructures. This technique uses the embedded IPv4 address structure, which eliminates the need for tunnel configuration in most cases.
- An optional mechanism for translating headers of IPv4 packets into IPv6, and the headers of IPv6 packets into IPv4. This technique allows nodes that implement only IPv6 to interoperate with nodes that implement only IPv4.

Types of Nodes

IPv4-only node: A host or router that implements only IPv4. An IPv4-only node does not understand IPv6. The installed base of IPv4 hosts and routers existing before the transition begins are IPv4-only nodes.

IPv6/IPv4 node: A host or router that implements both IPv4 and IPv6.

IPv6-only node: A host or router that implements IPv6, and does not implement IPv4. The operation of IPv6-only nodes is not addressed here.

IPv6 node: Any host or router that implements IPv6. IPv6/IPv4 and IPv6-only nodes are both IPv6 nodes.

IPv4 node: Any host or router that implements IPv4. IPv6/IPv4 and IPv4-only nodes are both IPv4 nodes.

4.3.8.1 Types of Hosts and Routers

To understand the Transition Model, it is necessary to know the various kinds of hosts and routers. In the model there exists 4 types:

- IPv4-only-nodes
These are host and routers that only understand IPv4.
- IPv6/IPv4-nodes
The routers and hosts of this category have both the IPv4 and the IPv6 protocol stacks. In addition to that they have mechanisms such as IPv6-over-IPv4 tunneling. These nodes can directly interoperate with both IPv4 and IPv6 nodes, but for communication with IPv4-only-nodes they have to be configured with an IPv4-compatible IPv6 address.
- IPv6-only-nodes
That are hosts and routers that only understand IPv6.
- IPv6/IPv4-header-translating-router
These routers translate IPv6 packets into IPv4 packets and vice-versa.

4.3.8.2 Types of IPv6 Addresses & Transition Mechanisms

I. Types of Addresses

Basically there are two kind of addressing that needs to be addressed here. They are,

IPv4-compatible IPv6 address: An IPv6 address, assigned to an IPv6/IPv4 node, which bears the high-order 96-bit prefix 0:0:0:0:0:0, and an IPv4 address in the low-order 32-bits. The automatic tunneling mechanism uses IPv4-compatible addresses.

IPv6-only address: The remainder of the IPv6 address space. An IPv6 address that bears a prefix other than 0:0:0:0:0:0.

II. Techniques Used in the Transition

The mechanisms used to integrate the IPv4 and IPv6 protocols stacks can be broadly categorized as follows,

IPv6-over-IPv4 tunneling: The technique of encapsulating IPv6 packets within IPv4 so that they can be carried across IPv4 routing infrastructures.

IPv6-in-IPv4 encapsulation: IPv6-over-IPv4 tunneling.

Configured tunneling: IPv6-over-IPv4 tunneling where the IPv4 tunnel endpoint address is determined by configuration information on the encapsulating node.

Automatic tunneling: IPv6-over-IPv4 tunneling where the IPv4 tunnel endpoint address is determined from the IPv4 address embedded in the IPv4-compatible destination address of the IPv6 packet.

4.3.8.3 IPv6-over-IPv4 Tunneling

Tunneling is used to carry IPv6 packets across IPv4 routed network areas. One of the requirements for tunneling is that the begin and endpoints of the tunnel are IPv6/IPv4-nodes with IPv4-compatible IPv6 addresses. Tunneling means that the whole IPv6 packet is mapped into a body of an IPv4 packet and sent across the IPv4 network area. The endpoint of the tunnel has to be either an IPv6/IPv4-header-translating-router or an IPv6/IPv4-node to de-encapsulate the packet. The destination address of the new IPv4 packet is the address of the node representing the tunnel endpoint. There are two types of tunneling: automatic tunneling and configured tunneling.

I. Automatic Tunneling

Automatic tunneling is used between two IPv6/IPv4-hosts. It is "end-to-end". It can also be used if a router is going to send an IPv6 packet to an IPv6/IPv4-host that is connected to the same IPv4 network area. It is important that the endpoint of the tunnel is the destination host.

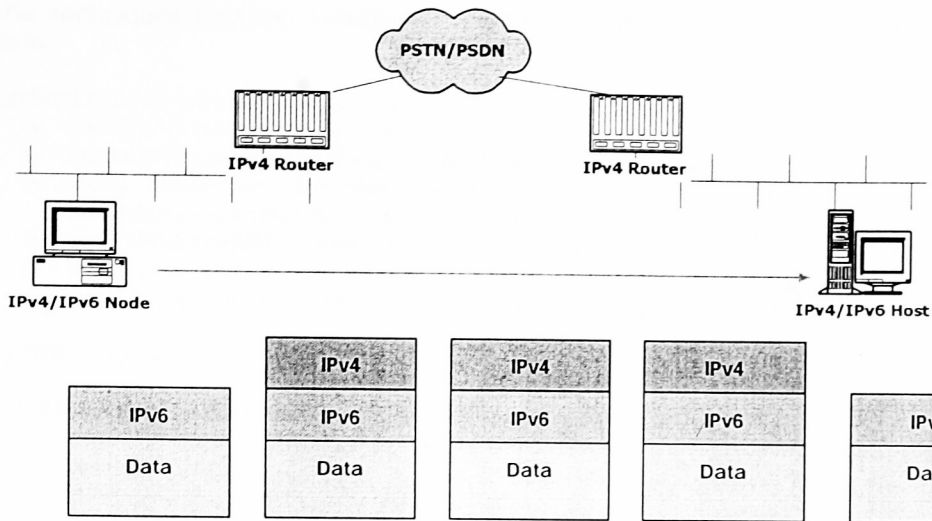


Figure 65: Net-Structure & Packet-Structure

II. Configured Tunneling

Configured tunneling is used if the destination host is different from the endpoint of the tunnel. In this case, the destination address for the IPv4 header, i.e. the address of the endpoint of the tunnel, could not be simply mapped from the IPv6 destination address. The endpoint of the tunnel has to be configured in the IPv6/IPv4-node.

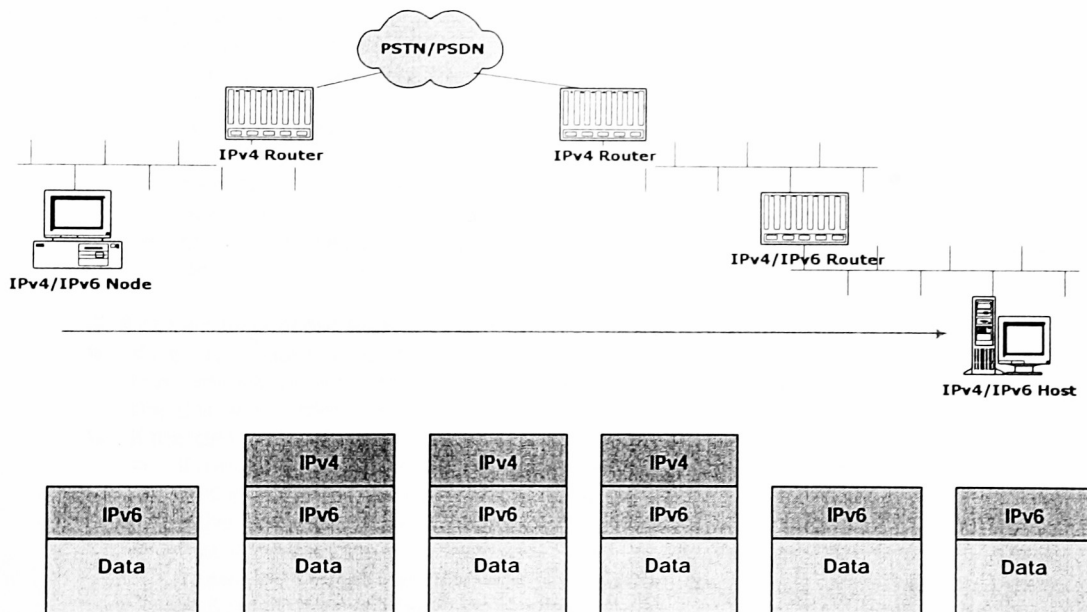


Figure 66: Net-Structure Packet-Structure

4.3.8.4 Default Algorithm used for transition

This section elucidates as to how the default sending algorithms works. It presents a combined IPv4 and IPv6 sending algorithm that IPv6/IPv4 nodes can use. The algorithm can be used to determine when to send IPv4 packets, when to send IPv6 packets, and when to perform automatic and configured tunneling. It illustrates how the techniques of dual IP layer, configured tunneling, and automatic tunneling can be used together. Note that is just an example to show

how the techniques can be combined; IPv6/IPv6 implementations may provide different algorithms

This algorithm has the following properties:

- ☛ Sends IPv4 packets to all IPv4 destinations.
- ☛ Sends IPv6 packets to all IPv6 destinations on the same link.
- ☛ Using automatic tunneling, sends IPv6 packets encapsulated in IPv4 to IPv6 destinations with IPv4-compatible addresses that are located off-link.
- ☛ Sends IPv6 packets to IPv6 destinations located off-link when IPv6 routers are present.
- ☛ Using the default IPv6 tunnel, sends IPv6 packets encapsulated in IPv4 to IPv6 destinations with IPv6-only addresses when no IPv6 routers are present.

The algorithm is as follows:

I. If the address of the end node is an IPv4 address then:

- ☛ If the destination is located on an attached link, then send an IPv4 packet addressed to the end node.
- ☛ If the destination is located off-link, then:
 - ☞ If there is an IPv4 router on link, then send an IPv4 format packet. The IPv4 destination address is the IPv4 address of the end node. The datalink address is the datalink address of the IPv4 router.
 - ☞ Else, the destination is treated as "unreachable" because it is located off link and there are no on-link routers.

II. If the address of the end node is an IPv4-compatible IPv6 address (i.e. bears the prefix 0:0:0:0:0:0), then:

- ☛ If the destination is located on an attached link, then send an IPv6 format packet (not encapsulated). The IPv6 destination address is the IPv6 address of the end node. The datalink address is the datalink address of the end node.
- ☛ If the destination is located off-link, then:
 - ☞ If there is an IPv4 router on an attached link, then send an IPv6 packet encapsulated in IPv4. The IPv6 destination address is the address of the end node. The IPv4 destination address is the low-order 32-bits of the end node's address. The datalink address is the datalink address of the IPv4 router.
 - ☞ Else, if there is an IPv6 router on an attached link, then send an IPv6 format packet. The IPv6 destination address is the IPv6 address of the end node. The datalink address is the datalink address of the IPv6 router.
 - ☞ Else, the destination is treated as "unreachable" because it is located off-link and there are no on-link routers.

III. If the address of the end node is an IPv6-only address, then:

- ☛ If the destination is located on an attached link, then send an IPv6 format packet. The IPv6 destination address is the IPv6 address of the end node. The datalink address is the datalink address of the end node.
- ☛ If the destination is located off-link, then:
 - ☞ If there is an IPv6 router on an attached link, then send an IPv6 format packet. The IPv6 destination address is the IPv6 address of the end node. The datalink address is the datalink address of the IPv6 router.
 - ☞ Else, if the destination is reachable via a configured tunnel, and there is an IPv4 router on an attached link, then send an IPv6 packet encapsulated in IPv4. The IPv6 destination address is the address of the end node. The IPv4 destination address is the configured IPv4 address of the tunnel endpoint. The datalink address is the datalink address of the IPv4 router.
 - ☞ Else, the destination is treated as "unreachable" because it is located off-link and there are no on-link IPv6 routers.

4.3.9 Transition Scenarios ¹¹³

The major transition mechanisms that are integral to the IPv6 design effort and these techniques include dual-stack IPv4 /IPv6 hosts and routers, tunneling of IPv6 via IPv4, and a number of IPv6 services, including IPv6 DNS, DHCP, MIBs, and so on. The flexibility and usefulness of the IPv6 transition mechanisms are best gauged through scenarios that address real-world networking requirements.

Scenario 1: No Need to NAT Take, for instance, the case of two large, network-dependent organizations that must interface operations due to a *merger and acquisition (M&A)*, or a new business partnership. Both of the enterprises in this scenario have large IPv4-based networks that have grown from small beginnings. Both of the original enterprises have a substantial number of private IPv4 addresses that are not necessarily unique within the current global IPv4 address space. Combining these two non-unique address spaces could require costly renumbering and restructuring of routers, host addresses, domains, areas, exterior routing protocols, and so on. This scenario is quite common in the current business climate, not only for M&A projects, but also for large outsourcing and customer/supplier networking relationships, where many hosts from the parent, outsourcer, supplier, or partner must be integrated into an existing enterprise address structure. Regardless of the scenario, IPv6 is an excellent approach to this challenge.

The task of logically merging two enterprise networks into a single autonomous domain is an expensive and potentially disruptive project. To avoid the cost and disruption of comprehensive renumbering, enterprises may be tempted to opt for the stopgap solution of a network address translator (NAT). In the case of the M&A scenario, a NAT could allow the two enterprises to maintain their private addresses in a more or less status quo fashion. To accomplish this, a NAT must conduct address translation in real time for all packets that move between the two organizations. Unfortunately, this solution introduces all the problems associated with NATs that were discussed in Part I, including performance bottlenecks, lack of scalability, lack of standards, and lack of universal connectivity among all the nodes in the new enterprise and the Internet.

In contrast with NAT, IPv6 provides a robust "future-oriented" solution to the logical integration of two physical networks. For the sake of the discussion, the two originally independent enterprises will be known as Enterprise A and Enterprise B. The first step is to determine which hosts need access to both sides of the new organization. These hosts are outfitted with dual IPv4/IPv6 stacks, which allow them to maintain connectivity to their original IPv4 network while also participating in a new IPv6 logical network that will be created "on top" of the existing IPv4 physical infrastructure.

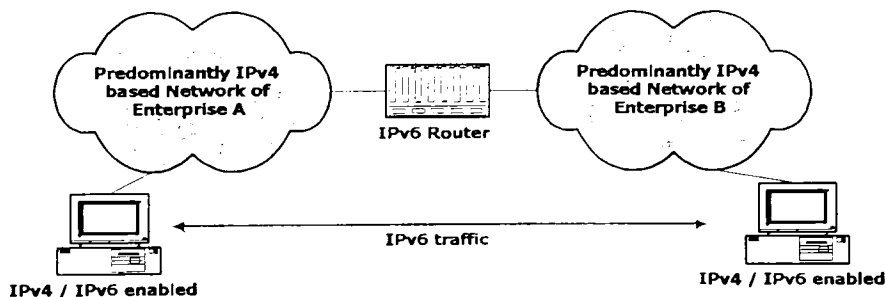


Figure 67: IPv6 Unites Private Address Spaces

It's likely that the accounting department of the integrated enterprises will have financial applications on servers that will need to be accessed by accounting employees in both Enterprise A and Enterprise B. Both servers and clients will be given IPv6, but they will also retain their IPv4 stack components. The IPv6 sessions of the accounting department will travel over the existing local and remote links as "just another protocol," requiring no changes to the physical network. The only requirement for IPv6 connectivity is that routers that are adjacent to accounting department

¹¹³ RFC 1933 - Transition Mechanisms for IPv6 hosts & routers

users must be upgraded to IPv6 capabilities. Where end-to-end IPv6 connectivity can't be achieved, one of the IPv4/IPv6 tunneling techniques can be employed.

As integration continues, other departments in the newly merged enterprises will also be given IPv4/IPv6 hosts. As new departments and workgroups are added, they may be given dual-stack hosts, or in some cases, IPv6-only hosts. Hosts that require communications to the outside world via the Internet will likely receive dual stacks to maintain compatibility with IPv4 nodes exterior to the enterprise. But in some cases, hosts that only require access to internal servers and specific outside partners may be able to achieve connectivity with IPv6-only hosts. A migration to IPv6 presents the opportunity for a fresh start in terms of address allocation and routing protocol structure. IPv6 hosts and routers can immediately take advantage of IPv6 features such as stateless autoconfiguration, encryption, authentication, and so on.

Scenario 2: IPv6 from the Edges to the Core For a great many corporate users, connectivity requirements focus primarily on access to local e-mail, database, and applications servers. In this case, it may be best to initially upgrade only isolated workgroups and departments to IPv6, with backbone router upgrades implemented at a slower rate. IPv6 protocol development is more complete for "edge" routing than for high-level backbone routing, so this is an excellent way for enterprises to gracefully transition into IPv6. As shown in figure, independent workgroups can upgrade their clients and servers to dual-stack IPv4/IPv6 hosts or IPv6-only hosts. This creates "islands" of IPv6 functionality.

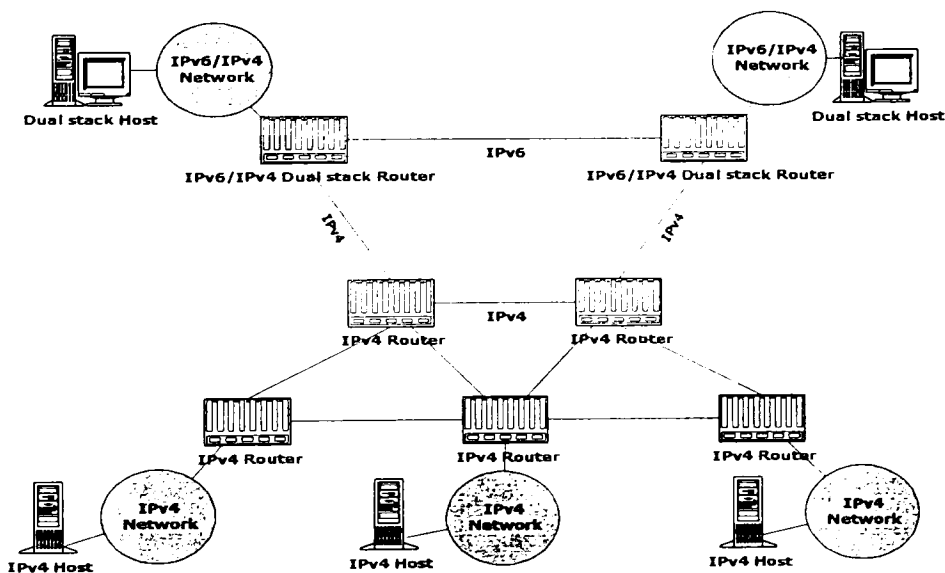


Figure 68: Islands of IPv6

As enterprise-scale routing protocols such as OSPF and BGP for IPv6 mature, the core backbone IPv6 connections can be deployed. After the first few IPv6 routers are in place, it may be desirable to connect IPv6 islands together with router-to-router tunnels. In this case, one or more routers in each island would be configured as tunnel endpoints. As described in Part I, when hosts use full IPv6 128-bit addressing, tunnels are manually configured so that the routers participating in tunnels know the address of the endpoints of the tunnel. With IPv4-compatible IPv6 addresses, automatic, non-configured tunneling is possible.

From a routing protocol standpoint, tunnels appear as a single IPv6 hop, even if the tunnel is comprised of many IPv4 hops across a number of different media. IPv6 routers running OSPF can propagate link-state reachability advertisements through tunnels, just as they would across conventional point-to-point links. In the IPv6 environment, OSPF will have the advantage of flexible

metrics for tunnel routes, to ensure that each tunnel is given its proper weight within the topology. In general, routers make packet-forwarding decisions in the tunneling environment in the same way that they make decisions in the IPv6-only network. The underlying IPv4 connections are essentially transparent to IPv6 routing protocols. The inevitable migration from IPv4 to IPv6 needs to be also addressed in looking at possible integration options for IP and ATM. There are fields in IPv6 such as priority and flow labels that can be used effectively in a quality centric, flow based multi-service networks. This will impact the multi-service models that are implemented before the transition from IPv4 to IPv6 takes place. As such we need to be aware of issues associated with it in being able to scale and use features provided by evolving technologies and architectures such as IPv6 in facilitating tighter integration between IP and ATM networks.

4.4. RSVP and integrated services on IP networks

The current Internet consists of a multitude of networks built from various link-layer technologies and relies on the **Internet Protocol (IP)** to internetwork between them. IP makes no assumptions about the underlying protocol stacks and offers an unreliable, connectionless network-layer service that is subject to packet loss, reordering, and packet duplication, all of which, together with queuing delay in router buffers, will increase with network load. Because of the lack of any firm guarantees, the traditional IP delivery model is often referred to as "best-effort" with an additional higher-layer end-to-end protocol such as the **Transmission Control Protocol (TCP)** required to provide end-to-end reliability.

TCP does this through the use of such mechanisms as packet retransmission, which further adds to the overall information transfer delay. For traditional non-real-time Internet traffic such as **File Transfer Protocol (FTP)** data, the best-effort delivery model of IP has not been a problem. However, as we move further into the age of multimedia communications, many real-time applications are being developed that are delay-sensitive to the point where the best-effort delivery model of IP can be inadequate even under modest network loads. Although the problem has been alleviated somewhat through making certain applications adaptive to network load where possible, there is still a firm need to provide many applications with additional service classes offering enhanced quality of service (QoS) with regard to bandwidth, packet queuing delay, and loss. These additional enhanced QoS delivery classes would supplement the best-effort delivery service in what could be described as an integrated service Internet.

TCP/IP networks to date have relied on the best-effort delivery model. There are no implicit or explicit guarantees of delivery in a timely fashion. If something happens in the network and the packets are lost or delayed, then it is up to the end station and the underlying mechanisms to resolve the situation. This is satisfactory for the earlier breed of applications such as email and ftp where there was no time criticality. With the advent of multimedia applications new demands have been placed on the network. The network, in fact must ensure sufficient resources in terms of bandwidth, buffers, and so on to service the needs of the applications. This introduces basic requirements, such as:

- Resource reservation – Applications must have a way of telling the network how much bandwidth and delay they need to operate. The network in turn should be able to check its available resources in either accepting or denying the request.
- Traffic control – The network should identify the application data flows and then deliver the requested levels of network resources as applicable.

To address these requirements IETF is developing the **Integrated Services Architecture (ISA)**. ISA is an enhancement to the current best-effort model that will enable TCP/IP networks to deliver QoS service guarantees to those applications that request them. There are two separate but related efforts, namely Integrated services and RSVP.

4.4.1 Integrated Services

In response to the growing demand for an integrated services Internet, the Internet Engineering Task Force (IETF) set up an Integrated Services Working Group [which has since defined several service classes that, if supported by the routers traversed by a data flow, can provide the data flow with certain QoS commitments. In contrast, best-effort traffic entering a router will receive no such service commitment and will have to make do with whatever resources are available. The level of QoS provided by these enhanced QoS classes is programmable on a per-flow basis according to requests from the end applications. These requests can be passed to the routers by network management procedures or, more commonly, using a reservation protocol such as RSVP, which is described herein. The requests dictate the level of resources (e.g., bandwidth, buffer space) that must be reserved along with the transmission scheduling behavior that must be installed in the routers to provide the desired end-to-end QoS commitment for the data flow.

In determining the resource allocations necessary to satisfy a request, the router needs to take account of the QoS support provided by the link layer in the data forwarding path. Furthermore, in the case of a QoS-active link layer such as asynchronous transfer mode (ATM) or certain types of local area network (LAN), the router is responsible for negotiations with the link layer to ensure that the link layer installs appropriate QoS support should the request be accepted. This mapping to link-layer QoS is medium-dependent, and the mechanisms for doing so are currently being defined by the **Integrated Services over Specific Lower Layers (ISSLL)** Working Group of the IETF. In the case of a QoS-passive link layer such as a leased line, the mapping to the link-layer QoS is trivial since transmission capacity is handled entirely by the router's packet scheduler. Each router must apply admission control to requests to ensure that they are only accepted if sufficient local resources are available. In making this check, admission control must consider information supplied by end applications regarding the traffic envelope their data flow will fall within. One of the parameters in the traffic envelope that must be supplied is the maximum datagram size of the data flow, and should this be greater than the **maximum transmission unit (MTU)** of the link, admission control will reject the request since the integrated services models rely on the assumption that datagrams receiving an enhanced QoS class are never fragmented. Once an appropriate reservation has been installed in each router along the path, the data flow can expect to receive an end-to-end QoS commitment provided no path changes or router failures occur during the lifetime of the flow, and provided the data flow conforms to the traffic envelope supplied in the request. Service-specific policing and traffic reshaping actions, as described in the next two subsections, will be employed within the network to ensure that nonconforming data flows do not affect the QoS commitments for behaving data flows.

4.4.2 Resource Reservation Protocol

The **Resource reSerVation Protocol (RSVP)**¹¹⁴ was designed to enable the senders, receivers, and routers of communication sessions (either multicast or unicast) to communicate with each other in order to set up the necessary router state to support the services described previously. It is worth noting that RSVP is not the only IP reservation protocol that has been designed for this purpose. RSVP identifies a communication session by the combination of destination address, transport-layer protocol type, and destination port number. It is important to note that each RSVP operation only applies to packets of a particular session; therefore, every RSVP message must include details of the session to which it applies. RSVP is not a routing protocol; it is merely used to reserve resources along the existing route set up by whichever underlying routing protocol is in place.

¹¹⁴ RFC 2379 - RSVP over ATM Implementation Guidelines

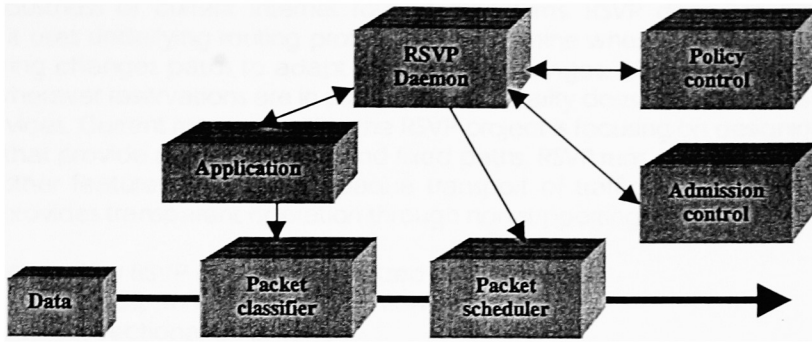


Figure 69: RSVP Operational Model

A host uses RSVP to request a specific **Quality of Service (QoS)** from the network, on behalf of an application data stream. RSVP carries the request through the network, visiting each node the network uses to carry the stream. At each node, RSVP attempts to make a resource reservation for the stream. To make a resource reservation at a node, the RSVP daemon communicates with two local decision modules, *admission control* and *policy control*. Admission control determines whether the node has sufficient available resources to supply the requested QoS. Policy control determines whether the user has administrative permission to make the reservation.

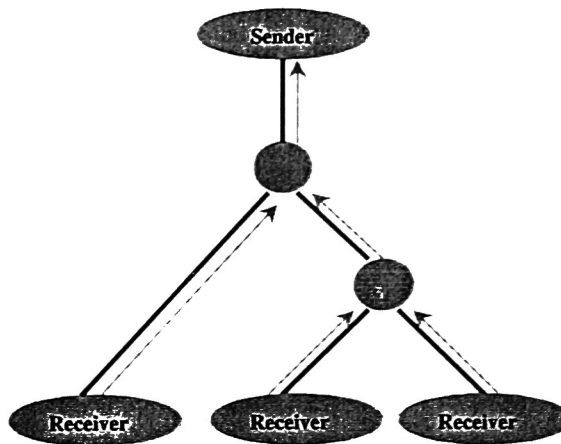


Figure 70: Receiver oriented reservation

If either check fails, the RSVP program returns an error notification to the application process that originated the request. If both checks succeed, the RSVP daemon sets parameters in a *packet classifier* and *packet scheduler* to obtain the desired QoS. The packet classifier determines the QoS class for each packet and the scheduler orders packet transmission to achieve the promised QoS for each stream.

A primary feature of RSVP is its scalability. RSVP scales to very large multicast groups because it uses receiver-oriented reservation requests that merge as they progress up the multicast tree. The reservation for a single receiver does not need to travel to the source of a multicast tree; rather it

travels only until it reaches a reserved branch of the tree. While the RSVP protocol is designed specifically for multicast applications, it may also make unicast reservations. RSVP is also designed to utilize the robustness of current Internet routing algorithms. RSVP does not perform its own routing; instead it uses underlying routing protocols to determine where it should carry reservation requests. As routing changes paths to adapt to topology changes, RSVP adapts its reservation to the new paths wherever reservations are in place. This modularity does not rule out RSVP from using other routing services. Current research within the RSVP project is focusing on designing RSVP to use routing services that provide alternate paths and fixed paths. RSVP runs over IP, both IPv4 and IPv6. Among RSVP's other features, it provides opaque transport of traffic control and policy control messages, and provides transparent operation through non-supporting regions.

The features supported by RSVP can be summarized as,

- Support for both unicast and multicast data flows
- Reservation for unidirectional data flows
- Receiver of data flow issues reservation request
- Receiver heterogeneity – different receivers can request and receive different QoS levels from the network for the same IP multicast flow
- Maintains soft state in routers – if reservations time out or route changes packets will continue to flow over the best effort path to destination
- Supports several reservation styles that enable in some cases multiple reservations to be merged into a single reservation, thus better utilizing network resources.
- Supports both Ipv4 and Ipv6

On the other hand limitations of RSVP are that,

- ↳ Adds complexity to the router
- ↳ Per flow state - does not scale well
- ↳ Suitable for small networks
- ↳ Separation of reservation mechanism from routing operation
- ↳ It's not an exhaustive approach to resource allocation (does not explore all available paths)
- ↳ Soft state reservation -needs to be periodically refreshed
- ↳ As the network scales and no. Of flows increases there is a need for control and admission policies - more refinement is needed to current draft(RSVP admission policy WG)
- ↳ Need for receiver and sender controls (currently receiver based)
- ↳ Soft state approach - changes per given duration and stability
- ↳ Throughput and delay guarantees require involvement of lower layers
- ↳ RSVP doesn't map easily to ATM QoS specifications

4.5 Summary

There are two basic types of networking protocol: trusted and best-effort. Connection oriented networks are trusted for use with applications requiring sequencing of information message units. Connectionless networks use a best effort approach and are employed by applications not concerned with sequential delivery of data. The connections made by these networks are predicted on a network addressing scheme and the ability to route the information between the source and destination. Internet Protocol (IP) has become the defacto standard for routing between source and destination. Its popularity has grown such that its current incarnation will not provide the expected address requirements for the near future, fostering the need for new IP standard. The Ipv6 provides for an address space almost 4.5 times that of the current IP, thus enhancing the service given to various applications by applying a QoS. The QoS is enhanced even more by the use of RSVP. But is only a signaling protocol and cannot ensure the allocation of necessary resources, but can make a request for them. Therefore, using RSVP connections between IP stations can guarantee minimal service requirements based on resource requirements and resource availability.

5. Research Findings – Part III

IP over ATM or IP/ATM integration – A new paradigm shift?

The network layer provides logical addressing of machines on a large network. This allows a machine to be recognized across various networks independent of what the hardware interface may be and allows the dynamic assignment or manual modification of a machine's identity on the network. Rather than having to live with the hardware interface address of the network card of each computer, network administrators can assign a unique IP address so the machine can be recognized anywhere in the intranet or even across the global Internet. This means companies can also use different network transmission technology within their networks and still have them exchange traffic.

In our quest for high speed, reliable, scalable networks, we have now reached the question of layer 2 switching or a combination of layer 2/Layer 3 switching. This will unfold the answers as to the best approach in internetworking IP and ATM. ATM is a switched circuit-based networking system. This means that each computer must establish a direct line to the computer it wishes to communicate with through however many mechanisms that are in place along the path. Unlike packet-based networks like the Internet, switched circuit networks, like the current analog or POTS telephone line, must be able to establish that clear line across the network between two systems. The benefit of such a system is stricter control allowing greater security, reliability, and administration. The way ATM was designed also resulted in faster and larger networks.

ATM cells of data are transferred through hardware-based switching systems that can work much faster because of the small, fixed-sized cells. IP packets, on the other hand, very often have unpredictable packet lengths and come in the form of large packets. This results in the packet becoming fragmented across small bandwidth connections often requiring repetitious reassembly at different points on the network. Fragmentation is a well-known phenomenon of IP traffic and is supported in all IP network stacks. Furthermore, each packet has to be checked to determine its destination and any other optional parameters needed for transmission that takes up valuable CPU cycles in processing. Very often the processing that occurs in routers delivering the packets across networks is mostly unnecessary.

In an ATM switched circuit network, all the parameters defining where and how information is to be delivered is arranged ahead of time by the source and destination computers and by all the switches in between. This is akin to the process of dialing a number on your telephone to get to the destination. This set-up procedure takes a short time (in computer cycles that are) and once established, all the switches need to do is forward the packets along the path marked out on the network. They do not have to reprocess each cell as it comes through, and the destination point is already known.

Essentially the amount of time to deliver a cell in a switched network is magnitudinally smaller than that in a routed network. That's not to say that routed networks are worthless; in fact the same properties that make routed networks slower also make them more intelligent. Having come to no certain conclusion on the role of layer 2 and layer 3 devices in integrating IP and ATM, we take an in depth view of the existing methodologies in internetworking IP and ATM in order to identify the best possible technique(s) which would be most beneficial in terms of speed, quality of service, scale, cost, performance and flexibility.

5.1 The challenges for interworking IP and ATM

The success of *Asynchronous Transfer Mode (ATM)* lies largely in its ability to transport legacy data traffic, mostly IP, over its network infrastructure. The complexity of interoperating IP with ATM originates from following two major differences between them.

Connection-oriented vs. Connectionless

ATM is connection-oriented¹¹⁴, that is, a connection need to established between two parties before they can send data to each other. Once the connection is set up, all data between them is sent along the connection path. On the contrary, IP is connectionless so that no connection is needed and each IP packet is forwarded by routers independently on a hop-by-hop basis. When we need to transport IP traffic over an ATM network. We have two options. Either a new connection is established on demand between two parties or the data is forwarded through preconfigured connection or connections. With the first approach, when the amount of data to be transferred is small, the expensive cost of setting up and tearing down a connection is not justified. On the other hand, with the second approach the preconfigured path(s) may not be an optimal path and may become overwhelmed by the amount of data being transferred.

QoS-aware(trusted) vs. Best Effort

Quality of Service is an important concept in ATM networks. It includes the parameters like the bandwidth and delay requirements of a connection. Such requirements are included in the signaling messages used to establish a connection. Current IP (IPv4) has no such concepts and each packet is forwarded on a best effort basis by the routers. To take advantage of the QoS guarantees of the ATM networks, the IP protocol need to be modified to include that information. The new draft of IPv6 has built into it quality of service features which can be exploited by higher level signaling protocols such as **Real Time Protocol (RTP)** and RSVP¹¹⁵ to deliver specific services based on quality on a limited scale.

Further, there are a number of issues that need to be addressed in any IP over ATM solution. These include Virtual Path/Virtual Circuit (VP/VC) usage and circuit setup & teardown policy, bandwidth efficiency and encapsulation, address support, routing/address resolution, multicast, QoS support, signaling, and IPv6 support. The overall determinant of success will be scalability.

Other challenges

All of the native modes of ATM as a sublayer of the B-ISDN have been developed with the aim of achieving an unprecedented degree of scalability. The very large ATM address space, whether NSAP end-point identifier or E.164, was defined partly for this reason, as was the PNNI with its undeniable complexity and its multiple hierarchical layers. However, this scalability is dependent upon reuse of the VP/VC space at each network interface by employing **Switched Virtual Path/Virtual Circuits (SVP/SVC)**, and hence on the speed with which signaling can set up and tears down circuits on demand.

The VP/VC usage problem in IP over ATM arises from the fundamental difference between classical IP and ATM, the connectionless vs. Connection oriented approach. In a network where all traffic flows across connections, and the lifetime of a connection is considerably greater than the time required to set it up and tear it down by end-to-end signaling, SVP/SVCs are reasonable and their use is indeed scalable. However, a connectionless network has very different requirements. Each packet (or frame, or Protocol Data Unit) is a relatively small and self-contained package of information, which is expected to make its own way across the network. As it is, at least in principle, unrelated to the packet, which precedes it or follows it, its characteristic timescale is very short. It is untenable to imagine setting up an end-to-end connection, with all the signaling and state information, which must be delivered to switches along the way, to route, a single packet across even a relatively small network, and such a process is manifestly unscalable in a global Internet. But the connectionless paradigm is the very soul of the Internet. In the broader context, we would be justified in mapping "flows" over IP to Virtual Circuits. However, traffic analysis shows that by reasonable definitions, flows average only around 100 packets. This sets fairly stringent limits on how much time can usefully be spent recognizing flows, and in setting up Virtual Circuits or other stateful paths across the Internet to carry them.

¹¹⁴ Ulyess Black, *Internetworking with ATM*, Prentice Hall, 1995

¹¹⁵ Stephen A. Thomas, *IPng and the TCP/IP Protocols*, John Wiley & Sons, Inc. 1998

There is no simple solution to this problem. In current ATM implementations, cell switching can be done very quickly but connection establishment is much slower,¹¹⁶ usually by several orders of magnitude. For instance, the StrataCom BPX Service Node (ATM switch), which supports 384,000 active connections per node with a throughput of 20Gbps, processes only 4000 "calls" per second. This results in a difficult decision between using persistent VCs, either Permanent Virtual Circuits or "Soft PVCs", and setting up new VCs on demand. There are pure "connectionless" approaches, where VCs are only used to connect neighboring routers, and pure connection-oriented approaches, where new VCs are always created. Both extremes have disadvantages; either they do not make full use of ATM capability or they are too expensive, or both. It is commonly agreed that there is a spectrum of possibilities in between. The ability to scale from local network to Internet proportions is a major issue.

On the other hand, connections are frequently necessary even in a connectionless network. TCP is the most common means of achieving connections in the Internet, but only the end-stations know about such connections in current implementations. This doesn't help with any type of service guarantee, bandwidth reservation, etc., which requires setting up state in intermediate routers or other switch points. As we discussed in the Introduction, there is a significant and growing demand for service guarantees for the delivery of Integrated Services, and the most straightforward method of enabling a service guarantee is by offering connections. A relatively non-intrusive method of establishing end-to-end connections even in a purely IP environment is through flow identification¹¹⁷.

This is typically done by recognizing some combination of source IP address and port number, destination IP address and port number, and protocol type. Most vendors have some implementations that do just this kind of flow recognition, coupled with simplified next hop lookup. In such implementations, the flow state in the router is setup on the first packet of a potential flow, which can lead to an actual increase in the router's burden with no concomitant performance improvement under some traffic patterns. Statistical approaches have been proposed in which the border router will set up a circuit only after a number of packets greater than some threshold have been sent in a recognizable "flow". This raises questions of the speed and efficacy of flow identification. As we will see, the same issue arises in **Multi Protocol Label Switching (MPLS)**¹¹⁸, the current generic term for coupled IP routers and ATM switches. Algorithms or heuristics also need to be devised for closing connections and tearing down circuits. The problem is not completely solved. If the number of packets that must be seen before setting up a new circuit is too high and if the tear down condition is too tight, there will still be inefficient use of network resources. Better solutions to this problem require QoS information to be used when making the decision. This requires QoS routing¹¹⁹, which by itself is very much an open issue.

Another class of fundamental questions revolve around the extent to which it is beneficial to map different flows above IP into separate connections below IP, or conversely to merge different flows with the same destination into a single connection below IP. This will be discussed further along with MPLS. Encapsulation deals with the way in which IP packets are sent over ATM VCs. RFC1483, the basic specification on multiprotocol encapsulation over ATM AAL5, defines 2 techniques: LLC/SNAP encapsulation, which allows different protocols to be multiplexed onto a single VC; and VC based multiplexing, which assigns each protocol a separate VC. The choice of which to use clearly depends upon a trade off between VC conservation and encapsulation efficiency.

Routing/address resolution is the central theme of much ongoing research in the IP over ATM domain. Different models for IP/ATM integration provide very different answers. We will go over these solutions in considerable detail, especially the scalability implications for enterprise network deployment. **Integrated PNNI (I-PNNI)** is the most thorough ATM centric approach; the others are generally ways of making an ATM network look like a classical broadcast network to applications. The **Multicast Address Resolution Server (MARS)** model is the most widely adopted method of

¹¹⁶ Raif O Onvural, Asynchronous Transfer Mode Networks - Performance Issues, Artech House, 1995

¹¹⁷ Jim Metzler, Lynn DeNoia, Layer 3 Switching, Prentice Hall, 1999

¹¹⁸ Jim Metzler, Lynn DeNoia, Layer 3 Switching, Prentice Hall, 1999

¹¹⁹ Uyless Balck, ATM: Foundation for Broadband Networks, Prentice Hall, 1995

supporting multicast over ATM with **Protocol Independent Multicast - Sparse Mode (PIM-SM)**¹²⁰ over ATM.

RSVP is a resource reservation protocol used to support CoS/QoS guarantees for data streams. With RSVP, many other service types besides "best-effort delivery" might be supported on the Internet: for example, real-time delivery service. ATM has been designed with native support for Quality of Service guarantees, and might seem a perfect substrate for implementing Integrated Services for IP. However, the fit between IP RSVP and ATM QoS may not be as easy as it first appears.

Signaling for call setup and teardown and other administrative functions is quite different under ATM than in more common network media. Older ATM switches generally supported only out of band signaling, but these are obsolete and have largely been replaced. Signaling in true ATM networks is done using OAM cells, sometimes over well known PVP/PVCs, sometimes within application level Virtual Circuits. Older proprietary signaling schemes are still in use (primarily Fore Systems' SPANS) but are gradually being displaced by industry standards for UNI and NNI signaling. The final major issue we will discuss is IPv6 support. IPv6 is the "Next Generation" IP protocol designed to salvage the Internet from an address space explosion. Many new features have also been added to IPv6, such as support for flow labeling, which is a good thing for real-time traffic, and neighbor discovery, which is difficult for ATM to handle, since it assumes the underlying network inherently supports multicast. With these considerations fresh in our mind we look at an appropriate framework for IP/ATM internetworking.

5.2 Framework for IP and ATM Internetworking¹²¹

In some discussion of IP/ATM internetworking, distinctions have been made between local area networks (LANs), and wide area networks (WANs) that do not necessarily hold. The distinction between a LAN, MAN and WAN is a matter of geographic dispersion. Geographic dispersion affects performance due to increased propagation delay. LANs are used for network interconnections at the major Internet traffic interconnect sites. Such LANs have multiple administrative authorities, currently exclusively support routers providing transit to multihomed internets, currently rely on PVCs and static address resolution, and rely heavily on IP routing. Such a configuration differs from the typical LANs used to interconnect computers in corporate or campus environments, and emphasizes the point that prior characterization of LANs do not necessarily hold. Similarly, WANs such as those under consideration by numerous large IP providers, do not conform to prior characterizations of ATM WANs in that they have a single administrative authority and a small number of nodes aggregating large flows of traffic onto single PVCs and rely on IP routers to avoid forming congestion bottlenecks within ATM.

The following characteristics of the IP and ATM internetwork may be independent of geographic dispersion (LAN, MAN, or WAN).

- The size of the IP / ATM internetwork (number of nodes).
- The size of ATM IP subnets (LIS) in the ATM Internetwork.
- Single IP subnet vs. multiple IP subnet ATM internetworks.
- Single or multiple administrative authority.
- Presence of routers providing transit to multihomed internets.
- The presence or absence of dynamic address resolution.
- The presence or absence of an IP routing protocol.

IP over/and ATM should therefore be characterized by:

- Encapsulations below the IP level.
- Degree to which a connection oriented lower level is available and utilized.
- Type of address resolution at the IP subnet level (static or dynamic).
- Degree to which address resolution is extended beyond the IP subnet boundary.
- The type of routing (if any) supported above the IP level.

¹²⁰ RFC 2337 - Intra-LIS IP multicast among routers over ATM using Sparse Mode PIM

¹²¹ RFC 1932 - IP over ATM: A Framework Document

ATM-specific attributes of particular importance include:

- The different types of services provided by the ATM Adaptation Layers (AAL). These specify the Quality-of-Service, the connection-mode, etc. The models discussed mostly assume an underlying connection-oriented service.
- The type of virtual circuits used, i.e., PVCs versus SVCs. The PVC environment requires the use of either static tables for ATM-to-IP address mapping or the use of inverse ARP, while the SVC environment requires ARP functionality to be provided.
- The type of support for multicast services. If point-to-point services only are available, then a server for IP multicast is required. If point-to-multipoint services are available, then IP multicast can be supported via meshes of point-to-multipoint connections (although use of a server may be necessary due to limits on the number of multipoint VCs able to be supported or to maintain the leaf initiated join semantics).
- The presence of logical link identifiers (VPI/VCI) and the various information element (IE) encodings within the ATM SVC signaling specification, i.e., the ATM Forum UNI versions.

This allows a VC originator to specify a range of "layer" entities as the destination "AAL User". The AAL specifications do not prohibit any particular "layer X" from attaching directly to a local AAL service. Taken together these points imply a range of methods for encapsulation of upper layer protocols over ATM. For example, while LLC/SNAP encapsulation is one approach (the default), it is also possible to bind virtual circuits to higher level entities in the TCP/IP protocol stack.

Some examples of the latter are single VC per protocol binding, **TCP & UDP over Lightweight IP (TULIP)**, and **TCP & UDP over Nonexistent IP connection (TUNIC)**, discussed further herein. The number and type of ATM administrative domains/networks and type of addressing used within an administrative domain/network. In particular, in the single domain/network case, all attached systems may be safely assumed to be using a single common addressing format, while in the multiple domain case, attached stations may not all be using the same common format, with corresponding implications on address resolution.

Also security/authentication is much more of a concern in the multiple domain case. IP over ATM proposals do not universally accept that IP routing over an ATM network is required. Certain proposals rely on the following assumptions:

- The widespread deployment of ATM within premises-based networks, private wide-area networks and public networks, and
- The definition of interfaces, signaling and routing protocols among private ATM networks. The above assumptions amount to ubiquitous deployment of a seamless ATM fabric, which serves as the hub of a star topology around which all other media is attached.

There has been a great deal of discussion over when, if ever, this will be a realistic assumption for very large internetworks, such as the Internet. Advocates of such approach point out that even if these are not relevant to very large internetworks such as the Internet, there may be a place for such models in smaller internetworks, such as corporate networks. The NHRP protocol, not necessarily specific to ATM, would be particularly appropriate for the case of ubiquitous ATM deployment. **Next Hop Resolution Protocol (NHRP)** supports the establishment of direct connections across IP subnets in the ATM domain. The use of NHRP does not require ubiquitous ATM deployment, but currently imposes topology constraints to avoid routing loops. The NHRP is looked in greater detail later. The Peer Model assumes that internetwork layer addresses can be mapped onto ATM addresses and vice versa, and that reachability information between ATM routing and internetwork layer routing can be exchanged. This approach has limited applicability unless ubiquitous deployment of ATM holds. The Integrated Model proposes a routing solution supporting an exchange of routing information between ATM routing/switching and higher levels routing/switching. This provides timely external routing/switching information within the ATM routing /switching and provides transit to external routing information through the ATM routing/switching between external routing domains. Such proposals may better support a possibly lengthy transition during which assumptions of ubiquitous ATM access do not hold. The **Multiprotocol over ATM (MPOA)**¹²² effort is one which deal with the issues of such integration, which is looking into the

¹²² .J. Duffy Hines. **ATM - The key to High-speed Broadband Networking**. M&T Books. 1996

specifics of the integration issues associated with them. We will take an in-depth look at existing models and techniques for integrating IP and ATM with a view of identifying the winning combination of solutions that will meet the requirements of future networks.

5.3 Models for IP and ATM internetworking architecture

5.3.1 The Classical IP Model /Overlay Model¹²³

The overlay model views ATM as a data link layer protocol on top of which IP runs. In overlay model ATM networks will have its own addressing scheme and routing protocols. The ATM address space is not logically coupled with the IP addressing space and there will be no arithmetic mapping between the two. Each end system will typically has an ATM address and an unrelated IP address as well. Since there is no mapping between the two addresses, the only way to figure out one from the other is through some address resolution protocol.

With the overlay model, there are essentially two ways to run IP over ATM. One treats ATM as a LAN and partitions an ATM network into several logical subnets consisting of end systems with the same IP prefix. This is known as Classical IP over ATM. In Classical IP over ATM, end systems in the same logical subnet communicate with each other through end-to-end ATM connections, and like in LAN, ARP servers are used in logical subnets to resolve the IP addresses into ATM addresses. However, traffic between end systems in different logical subnets has to go through a router even though they are attached to the same ATM network. This is not desirable since routers introduce a high latency and become the bandwidth bottleneck. **Next Hop Resolution Protocol (NHRP)** steps in to solve this problem. Working in an ATM network partitioned into logical subnets, it allows an end system in one subnet to resolve the ATM address (from the IP address) of an end system in another logical subnet and establish an end-to-end ATM connection, called a short-cut, between them.

The other approach uses an ATM network to simulate popular LAN protocols like Ethernet or token ring. IP runs on top of it in the same way it runs on top of Ethernet or token ring. This is known as **LAN Emulation (LANE)**. LANE allows current IP applications run over an ATM network without modification. This will help accelerate the deployment of ATM networks. However, like in Classical IP over ATM, traffic between different **emulated LANs (ELAN)** still needs to travel through a router. As a combination of LANE and NHRP, Multiprotocol Over ATM (MPOA) solves the problem by creating shortcuts that bypasses routers between ELANs.¹²⁴

The Classical IP Model retains the classical IP subnet architecture. This model simply consists of cascading instances of IP subnets with **IP Level (L3)** routers at IP subnet borders. Forwarding IP packets over this Classical IP model is straightforward using already well-established routing techniques and protocols.

SVC-based ATM IP subnets are simplified in that they:

- Limit the number of hosts, which must be directly connected at any given time to those that may actually exchange traffic.
- The ATM network is capable of setting up connections between any pair of hosts. Consistent with the standard IP routing algorithm connectivity to the "outside" world is achieved only through a router, which may provide firewall functionality if so desired.

The IP subnet supports an efficient mechanism for address resolution. Issues addressed by the IP over ATM Working Group, and some of the resolutions, for this model are:

¹²³ Koichi Asatani et al. *Introduction to ATM networks and B-ISDN*, John Wiley & Sons, 1997

¹²⁴ Ulyess Balck. *ATM: Foundation for Broadband Networks*, Prentice Hall, 1995

- Methods of encapsulation and multiplexing. This issue is addressed in RFC1483, in which two methods of encapsulation are defined, an LLC/SNAP and a per-VC multiplexing option.
- The definition of an address resolution server (RFC1577).
- Defining the default MTU size. This issue is addressed in RFC1626, which proposes the use of the MTU discovery protocol (RFC1191).
- Support for IP multicasting. The proposal for IP multicasting is currently defined by a set of IP over ATM WG Works in Progress referred to collectively as the IPMC documents. In order to support IP multicasting the ATM subnet must either support point to multipoint SVCs, or multicast servers, or both.
- Defining interim SVC parameters, such as QoS parameters and time-out values.

Signaling and negotiations of parameters such as MTU size and method of encapsulation. RFC1755 describes an implementation agreement for routers signaling the ATM network to establish SVCs initially based upon the ATM Forum's UNI version 3.0 specification, and eventually to be based upon the ATM Forum's UNI version 3.1 and later specifications. Topics addressed in RFC1755 include (but are not limited to) VC management procedures, e.g., when to time-out SVCs, QoS parameters, service classes, explicit setup message formats for various encapsulation methods, node (host or router) to node negotiations, etc. RFC1577 is also applicable to PVC-based subnets. Full mesh PVC connectivity is required. For more information see RFC1577.

The ROLC NHRP Model The **Next Hop Resolution Protocol (NHRP)**¹²⁵, currently a work in progress defined by the **Routing Over Large Clouds (ROLC)** Working Group performs address resolution to accomplish direct connections across IP subnet boundaries. NHRP can supplement RFC1577 ARP. There has been recent discussion of replacing RFC1577 ARP with NHRP. NHRP can also perform a proxy address resolution to provide the address of the border router serving a destination off of the NBMA, which is only served, by a single router on the NBMA. NHRP as currently defined cannot be used in this way to support addresses learned from routers for which the same destinations may be heard at other routers, without the risk of creating persistent routing loops.

5.3.2 Conventional Model¹²⁶

The "Conventional Model" assumes that a router can relay IP packets cell by cell, with the VPI/VCI identifying a flow between adjacent routers rather than a flow between a pair of nodes. A latency advantage can be provided if cell interleaving from multiple IP packets is allowed. Interleaving frames within the same VCI requires an ATM AAL such as AAL3/4 rather than AAL5. Cell forwarding is accomplished through a higher level mapping, above the ATM VCI layer. The conventional model is not under consideration by the IP/ATM working groups.

5.3.3 The Peer Model¹²⁷

The Peer Model places IP routers/gateways on an addressing peer basis with corresponding entities in an ATM cloud (where the ATM cloud may consist of a set of ATM networks, inter-connected via UNI or PNNI interfaces). ATM network entities and the attached IP hosts or routers exchange call routing information on a peer basis by algorithmically mapping IP addressing into the NSAP space. Within the ATM cloud, ATM network level addressing (NSAP-style), call routing and packet formats are used. In the Peer Model no provision is made for selection of primary path and use of alternate paths in the event of primary path failure in reaching multihomed non-ATM destinations. This will limit the topologies for which the peer model alone is applicable to only those topologies in which non ATM networks are singly homed, or where loss of backup connectivity is not an issue. The Peer Model may be used to avoid the need for an address resolution protocol and in a proxy ARP mode for stub networks, in conjunction with other mechanisms suitable to handle multihomed destinations. During the discussions of the IP over ATM working group, it was felt that the problems

¹²⁵ RFC 2336 - Classical IP and ARP over ATM to NHRP Transition

¹²⁶ Koichi Asatani et al. *Introduction to ATM networks and B-ISDN*. John Wiley & Sons. 1997

¹²⁷ George C. Sackett, Christopher Y. Metz. *ATM and Multiprotocol Networking*. McGraw-Hill, 1996

with the end-to-end peer model were much harder than any other model, and had more unresolved technical issues. While encouraging interested individuals/companies to research this area, it was not an initial priority of the working group to address these issues. The ATM Forum Network Layer Multiprotocol Working Group has reached a similar conclusion.

To run IP on top of ATM networks, we first need to figure out how to relate ATM protocol layers to TCP/IP protocol layers. Two models, one called peer model and the other overlay model, are proposed [Models]. Peer model considers the ATM layer a peer networking layer as IP and propose the use of the same addressing scheme as IP for ATM-attached end systems. ATM signaling requests will contain IP addresses and the intermediate switches will route the requests using existing routing protocols like OSPF. This scheme was rejected because although it simplifies the addressing scheme for end systems, it complicates the design of ATM switches by requiring them to have all the functions of an IP router. Moreover, if the ATM network will also support other networking layer protocols like IPX or Appletalk, the switch has to understand all their routing protocols.

5.3.4 The PNNI and the Integrated Models¹²⁸

The Integrated model (proposed and under study within the Multiprotocol group of ATM Forum) considers a single routing protocol to be used for both IP and for ATM. A single routing information exchange is used to distribute topological information. The routing computation used to calculate routes for IP will take into account the topology, including link and node characteristics, of both the IP and ATM networks and calculates an optimal route for IP packets over the combined topology. The PNNI is a hierarchical link state routing protocol with multiple link metrics providing various available QoS parameters given current loading. Call route selection takes into account QoS requirements. Hysteresis is built into link metric readvertisements in order to avoid computational overload and topological hierarchy serves to subdivide and summarize complex topologies, helping to bind computational requirements. Integrated Routing is a proposal to use PNNI routing as an IP routing protocol. There are several sets of technical issues that need to be addressed, including the interaction of multiple routing protocols, adaptation of PNNI to broadcast media, support for NHRP, and others. PNNI has provisions for carrying uninterpreted information.

PAR and I-PNNI: With above approaches, ATM and IP each run a separate routing protocol. For ATM it is PNNI and for IP it is OSPF. With IP, the routers have no idea about the internal topology of the ATM network, and with ATM, the switches do not distinguish between an ATM-attached router and an ATM end system. Sometimes it is desirable for the routers to understand the routing protocols of ATM to figure out how to establish end-to-end ATM connections with other routers. This resulted in **PNNI Augmented Routing (PAR)**, in which ATM attached routers behave like an ATM switch and exchange topology and reachability information with switches and other routers. Another approach, called **Integrated PNNI (I-PNNI)**, propose the use of PNNI as the single protocol to be used in a network of switches and routers.

These reflect various ideas the research groups at IETF and ATM Forum had with regard to possible internetworking model for IP and ATM. Some of them have gone through many cycles of refinement and reengineering before it did go into the level of any serious implementations. We discuss some important techniques/implementations at this level of maturity in trying to predict the next wave in it.

5.4 Techniques for internetworking IP and ATM: Intra-subnet

TCP/IP is the defacto standard for internetworking heterogeneous network host and applications. ATM has emerged with the ability to integrate not only TCP/IP data traffic but also other forms of information including voice and video. We review some of the existing models, which support

¹²⁸ Thomas M. Chen, Stephen S. Liu, *ATM Switching Systems*, Artech House, 1995

unicast and multicast IP traffic running over ATM in a single subnet environment. Classical IP and ARP over ATM (RFC 1577), is a technique for supporting IP over ATM in a single **Logical IP Subnet (LIS)**. An LIS is a group of IP hosts that share a common IP network number and subnet mask, and who communicate with each other directly using ATM connections. IP multicast over ATM enables IP multicast transmissions over an ATM network through the use of **Multicast Address Resolution Servers (MARS)** and ATM point-to-point connections. A MARS resolves IP group addresses with the ATM addresses of the group members. **LAN Emulation (LANE)** is an alternative model put forth by ATM Forum. All these techniques limit their scope within a subnet and are limited in their scope in inter-subnet implementations. We will see why it is so?

5.4.1 IP and ARP over ATM¹²⁹ – A classical approach

Classical IP and ARP over ATM is at most the natural approach for IP over ATM. Here the ATM is considered as a replacement for the LAN segment or the physical connectivity with the use of IP routers to interconnect two or more subnets. This purely limits the ATM's capability to the intra subnet connectivity.

To run IP's connectionless datagram protocol over the connection-based ATM network, a scheme for mapping IP addresses into ATM addresses or virtual connection IDs is needed, as well as rules for determining when to build a new call when SVCs are used. RFC 1577¹³⁰ defines the packet encapsulation used, the mechanism to map IP addresses into ATM addresses, and the conventions for when to set up and tear down virtual connections between systems. RFC 1577 defines only operation within each **logical IP subnet (LIS)**; connections between stations on different subnets go through a router, even when physical ATM connectivity exists between them.

Perhaps the most significant virtue of Classical IP over ATM is its simplicity. In a simple PVC network, IP addresses are mapped to virtual circuits, or virtual connections, manually. The user configures each station with a local address table that specifies which virtual connection corresponds to each IP address on the ATM network. This approach, with its requirement for manual maintenance of address tables, is difficult to manage for all but the smallest networks. It may, however, be the only approach available if the ATM network supports only PVCs, as is the case with most ATM WANs today. IP packets are carried in **ATM adaptation layer (AAL) 5 PDUs**¹³¹ (protocol data units). Each PDU contains an **LLC/SNAP (Logical Link Control/subnetwork access protocol)** header to identify the protocol, followed by the data, which is followed by the normal AAL 5 trailer with pad, length, and **CRC (cyclic redundancy check)** for validity checking. The default Classical IP over ATM packet length is 9,180 bytes, which is large enough to hold Ethernet, token ring, **Fiber Distributed Data Interface (FDDI)**, and **Switched Multimegabit Data Service (SMDS)** packets without fragmentation. However, since longer packet sizes usually give better file transfer performance, packet sizes up to the AAL 5 maximum of 64 kbytes are allowed if all stations in the ATM IP subnet defined in RFC 1577 as a LIS are configured for the larger value.

To use SVCs, end-stations must have a way of mapping IP addresses into ATM addresses and virtual connections automatically on demand. One additional protocol element is needed: an **ATM address resolution protocol (ATMARP)** server. The ATMARP server is a resource on each LIS that end-stations can query to find what ATM address to use to get a packet to a given destination IP address. It does the same job on ATM that the distributed broadcast-based ARP does on legacy LANs. The ATMARP server on each LIS automatically maintains a database for mapping IP addresses to ATM addresses. The ATMARP server may be a software module running on a file server or workstation, or it may be built into a router or ATM switch on the network.

In an IP/ATM network using SVCs, each station in a LIS initially sets up an ATM connection to the ATMARP server to register itself. The Classical IP over ATM specification does not cover how the

¹²⁹ RFC 2225 - Classical IP and ARP over ATM

¹³⁰ RFC 1577 - Classical IP and ARP over ATM

¹³¹ RFC 1483 - Multiprotocol Encapsulation over ATM Adaptation Layer 5

station finds out the ATMARP server address; this step typically is handled manually. The ATMARP server, when it accepts the call, sends an inverse ATMARP request to the calling end-station, requesting its IP address. The server then maintains the received address information in a local address table for answering requests from other stations. To keep the table current with any changes in the network and to minimize the size of the table required, the server ages out unused address entries after 20 minutes, if they are not reverified. The end-station may keep the ATM connections open to the server, or periodically reconnect to the server to refresh its entry in the address tables.

Classical IP over ATM requires no changes to a conventional router-based internetwork. Classical IP can be routed in the same way as conventional IP packets are forwarded from the originator to a router, and from router to router until reaching the final destination. Along the way, the IP header and upper-layer protocols and data remain essentially unchanged (except for certain control fields and the possible fragmenting of the packet into smaller IP datagrams), while the lower-level MAC layer encapsulation may be completely replaced with a different type of header at each router hop. Because IP sees ATM as simply another subnet type (along with Ethernet, token ring, FDDI, frame relay, and WAN circuits), internetworks in which all these media types are mixed can easily incorporate ATM into a campus backbone or into a high-performance workgroup, which communicates with the rest of the net via a router.

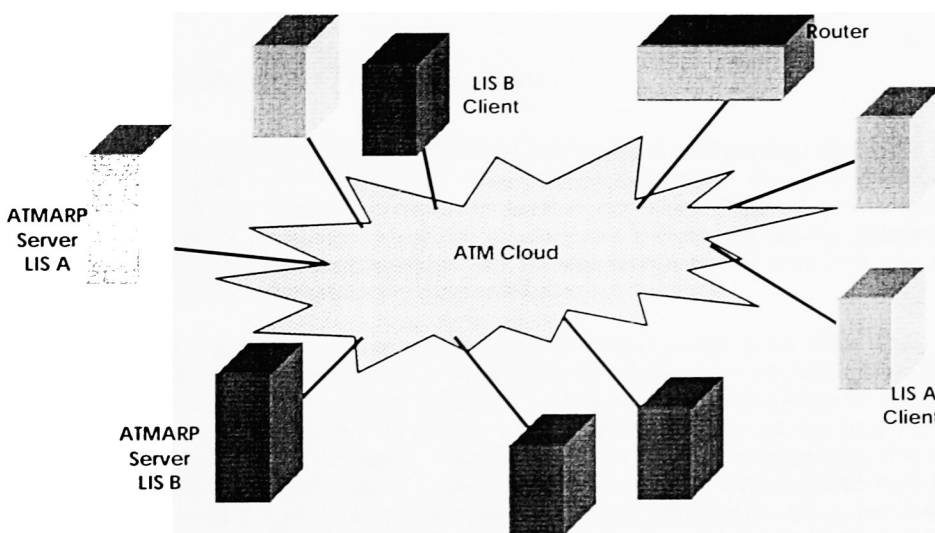


Figure 72: Classical Model

Within each LIS, systems communicate via point-to-point ATM virtual connections. IP packets are encapsulated into AAL 5 PDUs at the edge of the ATM network. The ATM cells in the PDU are then sent switch by switch through the ATM network, to be reassembled into an IP packet at the destination or ATM network's far edge. At the IP level, the ATM network appears as a single hop, regardless of the number of ATM switches involved, just as a telecommunications circuit is considered a single hop by routers, regardless of the number of circuit switches and multiplexers the circuit traverses.

Although it works adequately now, Classical IP's single-subnet operation limits the potential of ATM in IP networks because there is no general way to switch traffic between IP subnets over an ATM internetwork. This is considered a problem because of the greater potential ATM offers for quality of service management, reduced latency, and higher throughput.

At least two implementation groups are working to develop more advanced IP over ATM solutions. One is the IETF group that is working on how to **Routing Over Large Clouds (ROLC)**, including ATM, frame relay, and SMDS. The other is the ATM Forum's **Multiprotocol Over ATM (MPOA)** group, which is focusing on the general problem of carrying multiple protocols (such as IP, IPX/SPX, Appletalk, and others) over ATM, again by bypassing routers.

One likely outcome of the work by one or both of these groups is the development of a cut-through routing scheme, in which routing databases maintained by routers will be accessible for virtual circuit call setup. This would provide optimal routing over an ATM network. A current proposal now being considered is the IETF's **Next Hop Routing Protocol (NHRP)**, in which an ATM address resolution request, much like RFC 1577's ATMARP, is passed from ATM router to ATM router until the ATM address for the destination IP address (or that of the router at the exit to the ATM network) is learned. Another area with work in progress is IP multicast over ATM. There are some applications and protocols today that use IP multicast.

In the *Classical IP over ATM (RFC 1577) the protocols share single virtual channel and based on the Multiprotocol Encapsulation over ATM (RFC 1483) virtual channels are used for each protocol which* defines two ways to transport upper-layer protocols using AAL type 5 which aren't the only way to transport legacy traffic over an ATM connection. ATM Forum's LAN emulation specification is one other model for transporting IP over ATM.

5.4.2 IP Multicast over ATM¹³² – An extension

Multicast is a field under very active development within the IP community. While it is sometimes invoked as a critical test of ATM's suitability as a medium to carry IP, its use in the Internet as a whole is still largely experimental. No universal solution yet exists. There are two very different regimes in which different protocol sets are being developed: **Dense Mode** and **Sparse Mode**. The Sparse Mode architecture and protocols as well as the Dense Mode protocols are defined in articles and Internet Drafts, not yet as Standards (RFCs).

Multicast as it is evolving in the Internet is something of a hard problem for ATM. While all current ATM switches support some form of Point-to-Multipoint forwarding, like all ATM connections this is intrinsically unidirectional. Moreover, with standard ATM switches it is impossible to directly support Multipoint-to-Point functionality when using PDUs of greater than single cell size (e.g. AAL5) due to the so called **VC Merge** problem. That is, when PDUs are larger than a single cell, the destination host depends upon receiving all cells of a given PDU in order and unmixed with cells from other PDUs on a given Virtual Circuit. This is because AAL5 lacks information on a per cell basis for demultiplexing different PDUs within a single VC. Thus in order to support Multipoint-to-Point, or more generally Multipoint-to-Multipoint, ATM generally needs to utilize multiple virtual circuits. This leads to scaling problems and the possibility of VC exhaustion. Another related incompatibility has been that IP multicast is receiver-initiated while, until UNI 4.0 ATM signaling, supported only sender controlled group membership. Hence alternative approaches have been developed.

There are at present two ways to implement multicast service over ATM¹³³. ATM **Multicast Servers (MCS)** and **ATM VC meshes**. In the former, multicast packets are first sent to the server, and then are redistributed to all the receivers. In the latter, each sender sets up a Point-to-Multipoint VC to all receivers. The **Multicast Address Resolution Server (MARS)** protocol details the Address Resolution aspect, proposed for use in a Classical IP over ATM environment. It can support either the VC mesh model or the MCS model. There is at least one MARS server in each Cluster (usually the same as a LIS) which maintains a list of all the local receivers in each of the groups. When nodes join or leave a certain multicast group, MARS_JOIN or MARS_LEAVE messages are sent to the MARS server and are further forwarded to all members of the group.

¹³² RFC 2022 - Support for Multicast over UNI 3.0/3.1 based ATM Networks.

¹³³ RFC 2226 - IP Broadcast over ATM Networks

The VC mesh mechanism is suitable for small groups, and is consistent with the Cluster or LIS, but will not scale to large cloud or Internet proportions. In the VC mesh model, when a host wants to send to a group, it sets up its own **Point-to-Multipoint (P2MP)** VC for each group it is sending to (conventional IP multicast routing protocols could be used to forward multicast traffic between Clusters).

The MultiCast Server (MCS) model extends the MARS model to use Servers rather than VC meshes. The MCS establishes a P2MP tree with itself as the source, to all registered multicast group members in the LIS. There may be more than one MCS within a LIS for fault tolerance, but only one is active at any given time. This alleviates the requirement for full mesh connectivity between all members of the multicast group, which may be an inefficient use of ATM resources. This arrangement is illustrated in Figure.

VC mesh vs. Multicast servers

A cluster defines a set of ATM hosts that participate in an ATM level multicast and we will look at the two models that exist for such purposes. The first model called the VC mesh involves ATM hosts establishing a point-to-multipoint VC with other ATM hosts that are members of that specific group. This would mean each host needs to have a point-to-multipoint connection to all other hosts in the group. The second model called the Multicast Server (MCS), uses a direct VC to a multicast server which in turn transmits over a point-to-multipoint VCs to members of the multicast group.

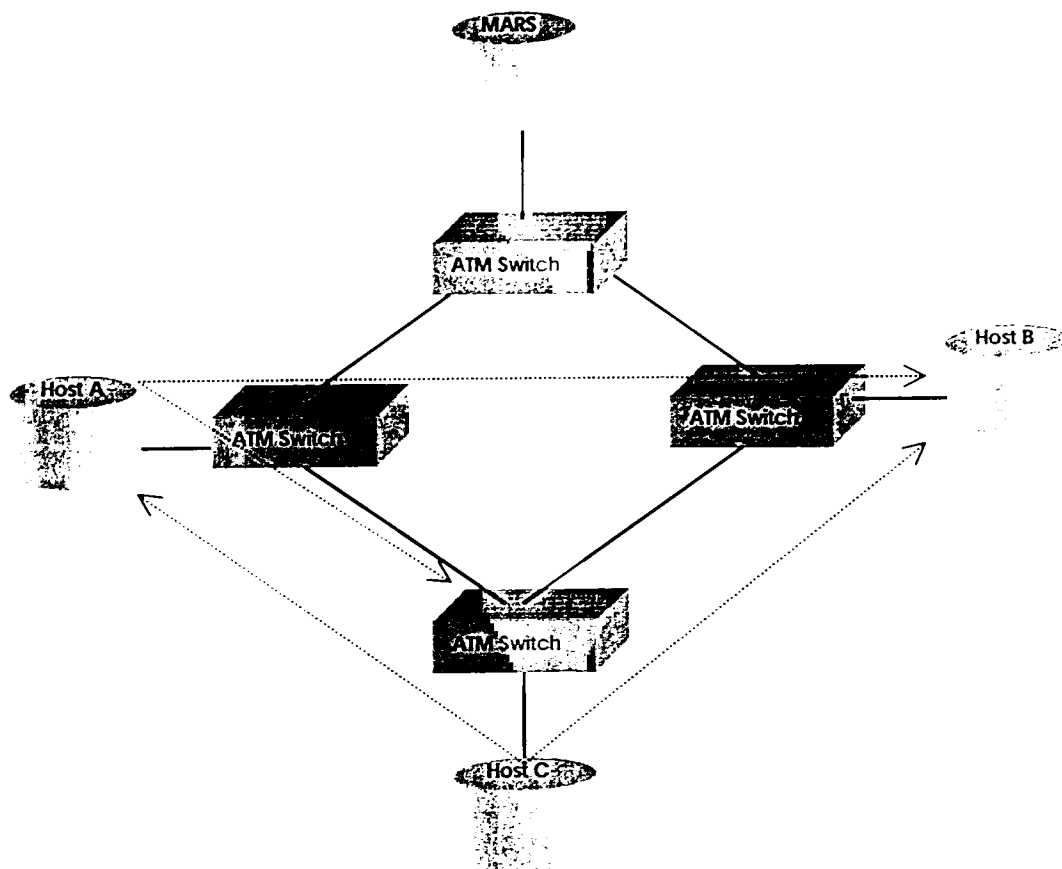


Figure 73: VC Model for IP over ATM multicast

Both models will achieve the same result, but there are trade-offs. It can be visually confirmed that a VC mesh will require many more VCs and that this will extract a toll in the network resource consumption. More, buffers, switch control blocks, and UNI signaling overhead will be required to support multicast traffic. But, VC mesh does offer optimal performance and leverages the switching fabric so that the multicasting is performed for multicast applications.

On the other hand, an MCS is easy to manage and conceptualize. There are fewer VCs required to support a multicast group. Any changes in group membership would only impact the point-to-multipoint VC from the MCS. Since VCs are expensive (considering the bandwidth), an MCS might be a better approach. But this done sacrificing certain degree of performance in that the traffic over a VC to MCS must be reassembled into an AAL frame before being transmitted back to the multicast group. MCS does offer better control and management of the multicast group in a centralized manner.

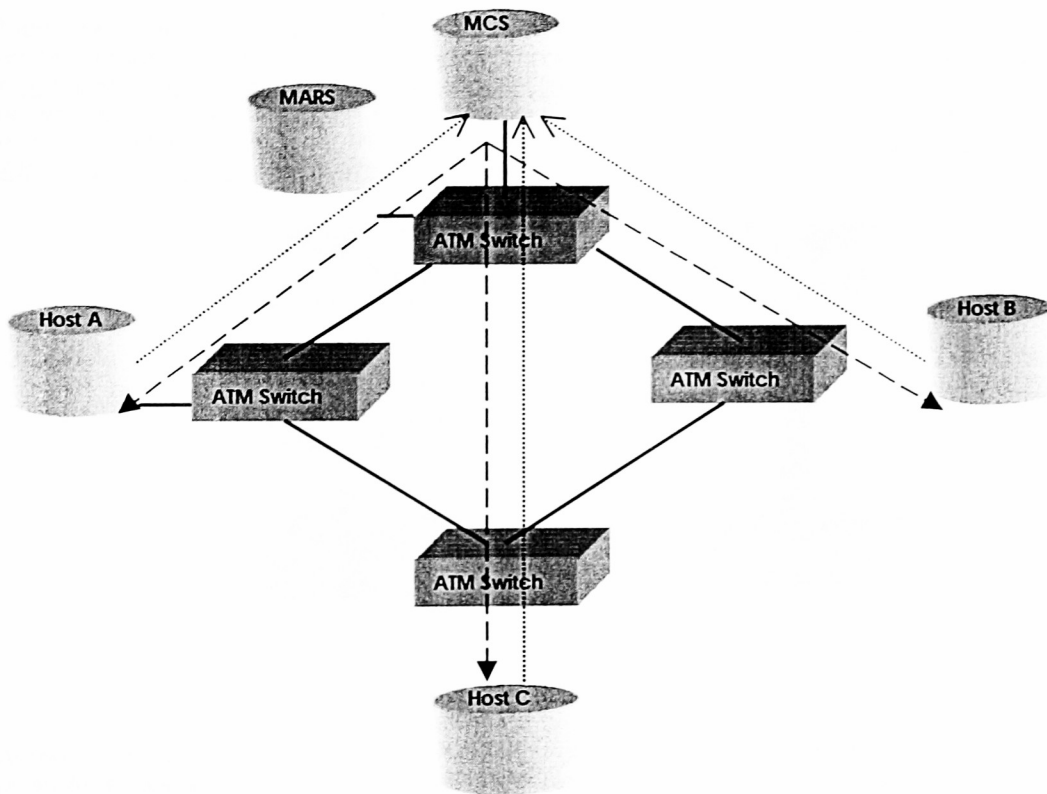


Figure 74: Multicast server model for IP over ATM multicast

5.4.3 LAN Emulation (LANE)¹³⁴ – An Overlay approach

Given the vast installed base of LANs and WANs today and the network and link layer protocols operating on these networks, a key to ATM success will be the ability to allow for interoperability between these technologies and ATM. Few users will tolerate the presence of islands of ATM without connectivity to the remainder of the enterprise network. The key to such connectivity is the

¹³⁴ ATM Forum - "LAN Emulation Over ATM Specification -- Version 1." ATM Forum Specification, February 1995.

use of the same network layer protocols, such as IP and IPX, on both existing networks and on ATM, since it is the function of the network layer to provide a uniform network view to higher level protocols and applications.

Despite superficial similarities, the goals of LAN emulation and Classical IP are completely different. LAN emulation works with all protocols and completely hides ATM from the upper layers. Classical IP over ATM supports one protocol and doesn't attempt to emulate the existing MAC layer. It is much simpler and generates much less overhead than LAN emulation. The price to pay is that any device with a legacy LAN adapter must go through a router or bridge to reach a workstation with the extra overhead associated with routers. However there are some advantages. Because there is no necessity to maintain compatibility with the existing MAC layer, designers decided to have a maximum frame size of more than 9000 bytes. This large frame size reduces packet overhead particularly when performing data transfer. Also the Classical IP protocol stack knows it is running over ATM and so it allows users to invoke ATM's quality of service features, such as specifying the maximum cell delay or acceptable cell loss for a connection.

There are, however, two fundamentally different ways of running network layer protocols across an (overlay mode) ATM network. In one method, known as native mode operation, address resolution mechanisms are used to map network layer addresses directly into ATM addresses, and the network layer packets are then carried across the ATM network. Native mode protocols will be examined in the next section. The alternate method of carrying network layer packets across an ATM network is known as LAN emulation (LANE).

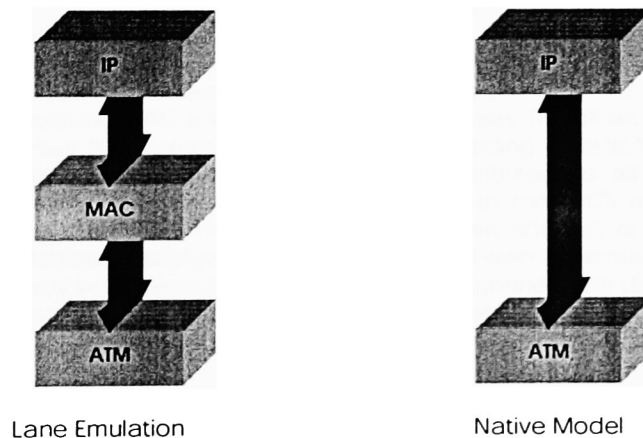


Figure 75: Overlay Models

As the name suggests, the function of the LANE protocol is to emulate a local area network on top of an ATM network. Specifically, the LANE protocol defines mechanisms for emulating either an IEEE 802.3 Ethernet or an 802.5 Token Ring LAN. What LAN emulation means is that the LANE protocol defines a service interface for higher layer (that is, network layer) protocols, which is identical to that of existing LANs, and that data sent across the ATM network are encapsulated in the appropriate LAN MAC packet format. It does not mean that any attempt is made to emulate the actual media access control protocol of the specific LAN concerned (that is, CSMA/CD for Ethernet or token passing for 802.5). In other words, the LANE protocols make an ATM network look and behave like an Ethernet or Token Ring LAN, albeit one operating much faster than a real network.

Note that the LANE protocol does not directly impact ATM switches. LANE, as with most of the other ATM internetworking protocols we will discuss later, builds upon the overlay model. As such, the LANE protocols operate transparently over and through ATM switches, using only standard ATM signaling procedures. ATM switches may well be used as convenient platforms upon which to implement some of the LANE server components, which we discuss below, but this is independent

of the cell relay operation of the ATM switches themselves. This logical decoupling is one of the great advantages of the overlay model, since they allow ATM switch designs to proceed independently of the operation of overlying internetworking protocols, and vice versa.

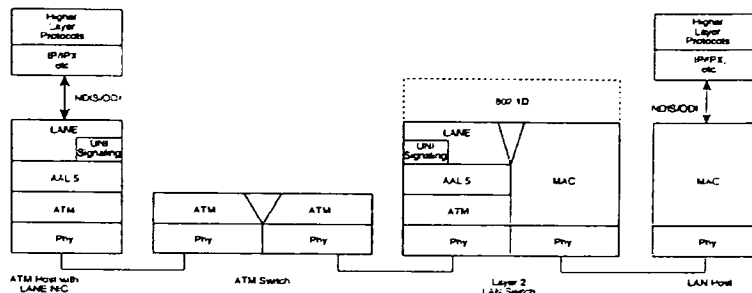


Figure 76: LANE protocol model

LANE divides an ATM network into multiple emulated LANs. These LANs operate independently and communication between emulated LANs is only possible through routers or bridges. Figure indicates the relationship between entities in LANE. Hosts in an emulated LAN are called **LAN Emulation Clients (LEC)**. Each emulated LAN has a **LAN Emulation Server (LES)**, a **LANE Configuration Server (LECS)**, and a **Broadcast and Unknown Server (BUS)**.

The Configuration Server assigns hosts in an ATM network to different LANs, the Broadcast and Unknown server handles all the broadcast/multicast traffic, while the LES is responsible for the **LAN Emulation Address Resolution Protocol (LE_ARP)**. LE_ARP allows the LES to fulfill the basic responsibility of LANE, resolving MAC addresses into ATM addresses. This allows LECs to set up direct SVC connections between themselves for unicast data forwarding. Broadcast/multicast traffic is sent first to the Broadcast/Unknown server and then redistributed to all the receivers. ATM addressing schemes are flexible. Connection from the LES to the LECS may occur across a Permanent Virtual Circuit (PVC), be initiated from a well-known address, or by using a protocol defined in the **Integrated Local Management Interface (ILMI)**. A web of permanent switched virtual circuits, both bi-directional and unidirectional, are used to interconnect the LECs, the LECS, the LES and BUS for signaling and control. Figure below illustrates the relationships between LEC, LES, LECS and BUS.

The LAN Emulation protocol defines mechanisms for emulating either an Ethernet (B02.3) or Token Ring (B02.5) LAN to attached host LECs. Supporting IP over LAN Emulation is the same as supporting IP over either of these IEEE B02 LANs, with no modification to higher layer protocols such as the common NDIS driver interface for IP and similar protocol stacks. It should be noted, however, that LANE provides no means of directly connecting between Ethernet and Token Ring emulations. A gateway is still required to bridge between them. Forwarding packets between different emulated LANs must be accomplished via routers, either ATM-attached conventional routers or a form of ATM router implementing LANE at two or more interfaces to different emulated networks.

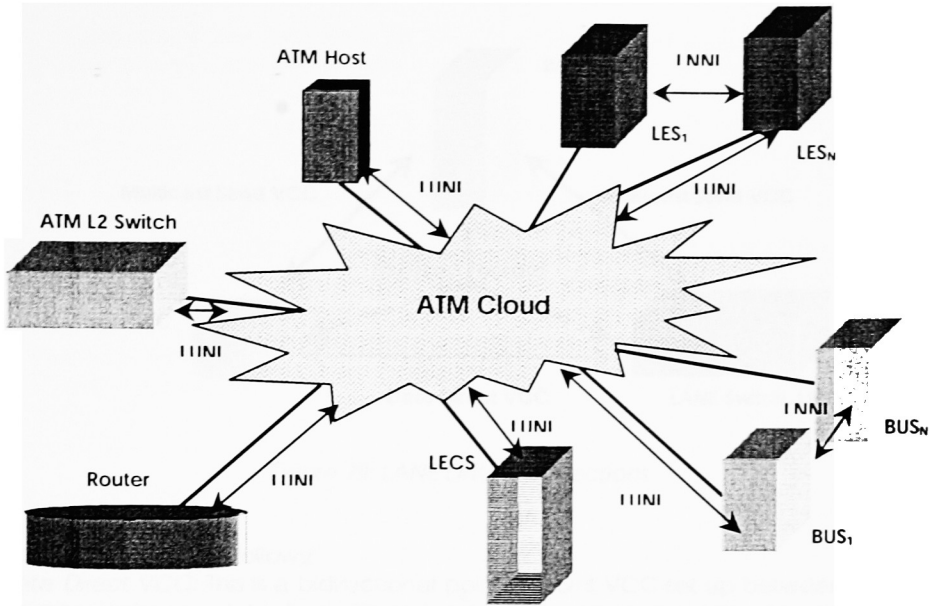


Figure 77: LAN Emulation Components and Protocol Interfaces

The Phase 1 LANE entities communicate with each other using a series of ATM connections. LECs maintain separate connections for data transmission and control traffic.

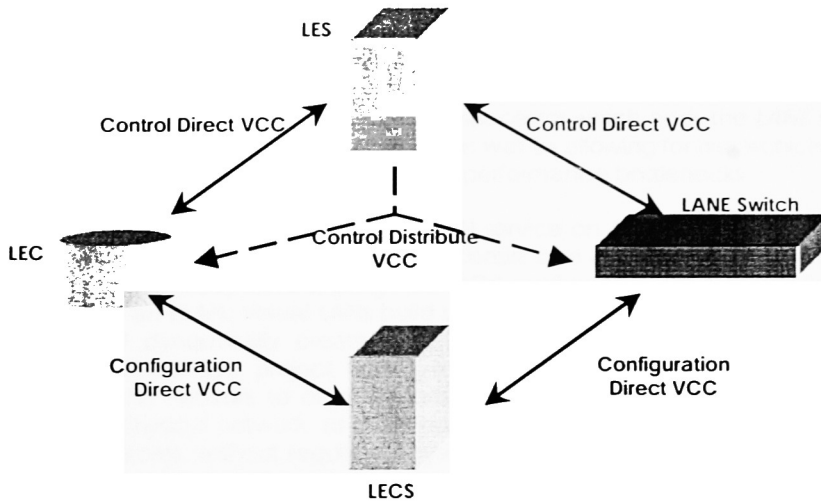


Figure 78: LANE Control Connections

The control connections are as follows:

- ⚡ *Configuration Direct VCC*: This is a bidirectional point-to-point VCC set up by the LEC to the LECS.
- ⚡ *Control Direct VCC*: This is a bidirectional VCC set up by the LEC to the LES.
- ⚡ *Control Distribute VCC*: This is a unidirectional VCC set up from the LES back to the LEC; this is typically a point-to-multipoint connection

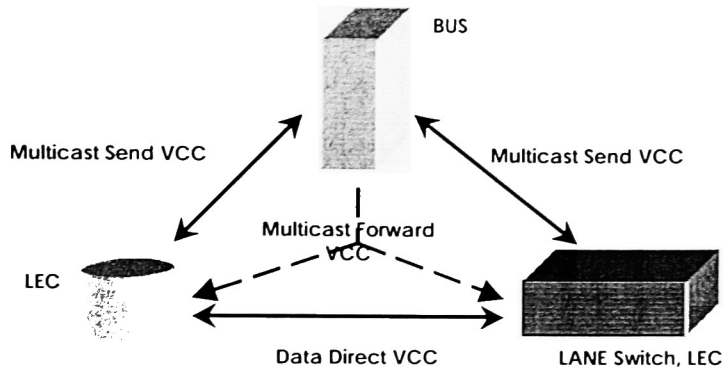


Figure 79: LANE Data Connections

The data connections are as follows:

- ⌚ **Data Direct VCC:** This is a bidirectional point-to-point VCC set up between two LECs that want to exchange data. Two LECs will typically use the same data direct VCC to carry all packets between them, rather than opening a new VCC for each MAC address pair between them, so as to conserve connection resources and connection set-up latency. Since LANE emulates existing LANs, including their lack of QoS support, data direct connections will typically be UBR or ABR connections, and will not offer any type of QoS guarantees.
- ⌚ **Multicast Send VCC:** This is a bidirectional point-to-point VCC set up by the LEC to the BUS.
- ⌚ **Multicast Forward VCC:** This is a unidirectional VCC set up to the LEC from the BUS, this is typically a point-to-multipoint connection, with each LEC as a leaf

LANE Phase 1 is currently deployed, while Phase 2 is the successor which adds the **LANE NNI (LNNI)** protocol to permit redundant LESs and replicated BUSs, as well as allowing for hierarchical BUSs. This is intended to avoid both single point failure modes and performance bottlenecks.

Further, LANE is used by vendors to provide a virtual LAN service on ATM backbones. Such virtual LANs are implemented on *switched internetworks* that consist of a combination of (bridging) LAN switches, ATM end systems (typically servers, using ATM NICs), and routers with ATM interfaces ("ATM routers") all connected to an ELAN. Virtual LANs build upon LANE and give network administrators the ability to easily and dynamically create and reconfigure virtual networks, tracking the formation and change of ad hoc project teams. In other words, virtual LANs allow network administrators to adapt the network to organizational work flows, rather than constraining the organization around the physical network, as they must currently do. Allowing centralized logical reconfiguration of end systems, without requiring physical network reconfiguration, can also help reduce the costs of moves, add and changes," which constitute a significant proportion of network support costs, given the increasing dynamism of work groups. For instance, a node could be physically moved, but still retain membership of the same VLAN it used to belong to before, without ending up on the "wrong" side of a network firewall. Conversely, a node could be made a member of a new virtual LAN through a change in its ELAN membership, without requiring any physical network changes. In the latter case, depending upon the protocol, the node may need to change its network layer (e.g. IP) address, though other protocols, such as DHCP, can also help automate this process.

These powerful benefits of virtual LANs will likely spur the widespread deployment of LANE. However, the limitations of LANE must also be understood. As noted earlier, LANE is essentially a LAN bridging standard. As such, much as with physically bridged LANs, ELANs are susceptible to such phenomena as broadcast storms. These factors tend to limit the applicability of ELANs to small

workgroups, where virtual LANs also offer the most powerful advantages. This means that a large enterprise network is likely to support a large number of virtual LANs (VLANs).

5.5 Techniques for Internetworking IP and ATM: Inter-subnet

The IP model consists of networks interconnected by routers. Packets travelling from one network to another must pass through a router. We did see that the Classical model is no exception. In that ATM connectivity was limited to a single US. All inter US traffic still had to go through a router, even if the USs were attached to a single ATM network. In a decade, where we see phenomenal growth in both IP and ATM networks each evolving in its own way to support features offered by one another. Here we address techniques used to overcome certain limitations posed by earlier models in facilitating host in different USs to communicate over the ATM network without having to pass through a router.

In Classical IP over ATM, end systems in the same logical subnet communicate with each other through end-to-end ATM connections, and like in LAN, ARP servers are used in logical subnets to resolve the IP addresses into ATM addresses. However, traffic between end systems in different logical subnets has to go through a router even though they are attached to the same ATM network. This is not desirable since routers introduce a high latency and become the bandwidth bottleneck. **Next Hop Resolution Protocol (NHRP)** steps in to solve this problem. Working in an ATM network partitioned into logical subnets, it allows an end system in one subnet to resolve the ATM address (from the IP address) of an end system in another logical subnet and establish an end-to-end ATM connection, called a short-cut, between them.

Next Hop Resolution Protocol (NHRP) is one such protocol, which facilitates direct ATM connectivity between source and destination on different subnets without the routers in the data path. But the NHRP model uses layered routing in which the internetwork layer routing and ATM routing operate independently. When a packet is to be sent across an ATM cloud, it is first routed to a router of the ATM network based on IP routing, since it's initially carried as IP payload. Since the IP routing protocol has no knowledge of the topology of the ATM network, it may make bad decisions when choosing the ingress router. QoS routing is also a very difficult issue when the dynamics of the ATM network are unavailable. **Integrated PNNI (I-PNNI)** model has been proposed just to solve these problems. We will also look at I-PNNI model which would also lead us to the **Multi Protocol over ATM (MPOA)** model for IP/ATM internetworking which builds on the concepts of LANE and NHRP. Despite its name, Multi Protocol Over ATM is an integration effort with the same underlying intention as I-PNNI, which is to provide a clean internetworking of ATM networks with legacy subnetworks. Further, we have seen another spurt of development in the L2/L3 flow based switching essentially taking a similar path as MPOA which is the **Multi Protocol Label Switching (MPLS)**. With all these developments more and more the industry is driven towards an integrated approach.

5.5.1 Next Hop Address Resolution Protocol (NHRP)¹³⁵ – A step in the right direction

The extra hop problem of the Classical model is one of the questions that NHRP is engineered to address. The **IP Over Non Broadcast Multi Access (NBMA) Network (ION)** working group has developed this new protocol, NBMA **Next Hop Routing Protocol (NHRP)**, which can support cut-through routing in order to eliminate these extra hops. NHRP is intended for use over both connectionless NBMA subnetworks (eg: SMDS) and connection oriented NBMA subnetworks (eg: ATM), so does not include mechanisms for connection establishment for the latter case. These must be provided by other protocols and as such MPOA, Frame Relay and X.25 networks are likely candidates for NHRP implementation.

¹³⁵ Katz, D. and Piscitello, D. - NBMA Next Hop Resolution Protocol (NHRP). Internet Draft, May 1995.

NHRP¹³⁶ uses **Local Address Groups (LAGs)** to model the NBMA networks. The main difference between the LIS model and the LAG model lies in how the local/remote forwarding decision is made. In the LIS model the decision is purely based on address information. Only nodes with the same IP network/subnet address can directly talk to each other. In the LAG model, any two nodes on the same NBMA network can establish a direct communication regardless of their IP addresses, while the local vs. remote forwarding decision is based upon QoS or traffic considerations. In heterogeneous networks, destinations will often lie outside the boundary of the NBMA network; NHRP has the ability to provide address resolution information for the destination router when the destination is not directly attached to the NBMA network.

A physical NBMA network may be partitioned into several disjoint NBMA Logical subnetworks. A NBMA Logical subnetwork is a collection of hosts and routers which share unfiltered connectivity. There are **Next Hop Servers (NHS)** in the Logical NBMA subnetwork, providing NHRP service within an NBMA cloud. Each NHS serves a set of destination hosts, which may or may not be on the NBMA network. Each station on the NBMA network must have a NHS in the same Logical NBMA subnetwork, which can provide authoritative address resolution information on its behalf. This NHS is the serving NHS of the station. Each entity which uses the NHRP service is a **Next Hop Client (NHC)**. While NHRP can be deployed transparently in a LIS which includes ARP services and hosts which do not understand NHRP, it does require all routers on the path between the NHC and the serving NHS of the destination to be NHRP-capable.

NHCs cache the results of LIS protocol address to NBMA address resolution requests as these are learned; the information may come from NHRP Resolution Reply packets, manual configuration, etc. NHRP Resolution Requests may be triggered by several different events; for instance, a host has a data packet to send, or a routing protocol update packet. When the trigger is a data packet, that packet must somehow be handled while awaiting the outcome of the Resolution Request. It may optionally be dropped, be buffered pending the Resolution Reply, or forwarded via the existing (non-shortcut) routed path toward the destination. The latter choice is recommended by the Draft, but may lead to disordering of packets once the shortcut is established. This should not be a problem for IP but may adversely affect other protocols.

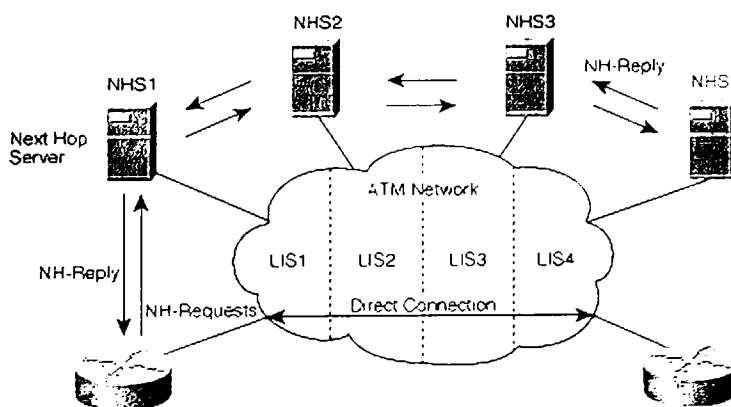


Figure 80: NHRP Model¹³⁷

In the Classical model, all data packets are forwarded hop-by-hop via intermediate routers. Routing decisions are made at each router every time a packet arrives. In the NHRP model, the same routing/forwarding mechanism is used, but not to forward the actual data packets. Instead, it is used to forward the NHRP Resolution Request packets from the source, which must be an NHC,

¹³⁶ RFC 2336 - Classical IP and ARP over ATM to NHRP Transition

¹³⁷ ATM Networks, <http://cne.gmu.edu/~sreddiva/ATMnet.html>

to the serving NHS of the destination, which can provide the address resolution information. Only the NHRP request packet encounters the extra hop processing overhead. The address resolution information is used by the supported overlay protocol to establish a direct NBMA connection. Subsequent data packets are sent directly from the source to the destination via the newly established connection. NHRP requires a contiguous deployment of NHRP capable routers. The NHRP request packets may be dropped if not recognized by an intermediate router. In this case, packets have to be forwarded hop-by-hop just as in the Classical model.

Stable routing loops are a possibility under NHRP. This is because it violates a fundamental tenet of IP routing that routing updates be sent across all paths through which data also flows. NHRP shortcuts are used only for data forwarding, and do not establish router adjacency. Even so, stable loops only form under relatively unlikely, even pathological network conditions. Loops are a possibility only when a back-door path exists between routers which is outside of, and unknown to, the NBMA network. Stable loops are likely to occur only at the boundaries between administrative domains, where inter domain routing protocols lose route metrics. However, the possibility is sufficiently serious that several options are under development. Route Record extensions for NHRP packets have been defined to aid in detecting loops, while administrative measures, particularly using routers at any boundary between administrative domains, are highly recommended.

Although NHRP solves the extra hop problem, it introduces some of its own. The first problem is the requirement of contiguous deployment of NHSs, which may not happen for a long time across the existing ATM Internet backbones. Second, the current NHRP focuses only on unicast routing. It may be possible to use NHRP to support multicast, where shortcut point-to-multipoint VCs can be used to avoid extra hops. Unfortunately, this approach is unlikely to be scalable. The sender will be overwhelmed if it attempts to set up short-cuts for a very large number of receivers. This problem is even worse in the IPv6 environment where multicast is an indispensable protocol element.

5.5.2 Integrated Private Network Network Interface (I-PNNI)¹³⁸ – A new dimension

PNNI¹³⁹ is an hierarchical link state routing protocol and a signaling protocol, used together to establish Switched Virtual Circuits (SVCs) in a private ATM network; in this context, a private network is one which uses NSAP format ATM addresses. The ATM Forum's main goals in developing PNNI are:

- ⚡ Quality of Service support
- ⚡ Universal scalability

PNNI's signaling is an extension of UNI signaling protocols, making use of well known VPI/VCI's to carry signaling messages. In its routing and addressing architecture, however, it draws heavily on the philosophy and world view of the Nimrod project. Nimrod is an IETF sponsored attempt to produce a Next Generation routing architecture, to accompany IPng. Given the slow deployment of IPv6, though, Nimrod development seems to be languishing.

Like the Nimrod work, PNNI is a map based routing protocol; that is, one, which distributes descriptive information about the network or portions of the network, as opposed to distributing routing tables. Link State routing protocols such as OSPF and IS-IS is essentially map based. The alternative, Distance Vector schemes (for instance, BGP and RIP) can be thought of as causing routers to distribute their view of the entire network rather than a map of their vicinity. Like Nimrod, PNNI mappings abstract sections of the network which lie at differing levels of hierarchy; these hierarchical maps allow sources to select their own routes across the network. Herein lies the biggest departure from current Internet practice; paths are explicitly chosen by sources rather than fully distributed, hop-by-hop paths in which each switch or router selects its own next hop.

¹³⁸ I-PNNI Accepted As Work Effort, <http://www.baynetworks.com/News/Press/9604232.html>

¹³⁹ ATM Forum -ATM Forum 94-0471R7- P-PNNI Draft Specification, March 1995

Integrated-PNNI (I-PNNI)¹⁴⁰

There are certainly a few large ATM clouds in existence. If and as large ATM networks become more commonplace, it will increasingly be required that routers talk to one another across these clouds. So long as IP routers have no knowledge of the internal topology of the ATM cloud, routing is at best inefficient and potentially unmanageable. At least 2 approaches based on PNNI have been developed to resolve the dilemma. **PNNI Augmented Routing (PAR)** is the less ambitious of the two. PAR requires that ATM connected routers run an instance of PNNI along with their normal IP routing protocol (OSPF, RIP etc.). ATM switches in the cloud run PNNI Phase 1. PNNI running on the edge routers allows them to see the topology of the ATM cloud. Switches in the cloud are also aware of the edge routers, and can set up SVCs, which originate and terminate at a router. Such routers are **designated restricted transit nodes**, which implies that they can never be an intermediate node in a SVC. PNNI has been designed with the ability to carry reachability information, which it doesn't understand between specific nodes, such as these edge routers; this is done using **TLV encoding (type/length/value)** for the IP specific information. PAR allows routers to learn about each other across an ATM cloud without either manual configuration of PVCs or the use of an IP over ATM protocol such as LANE, Classical IP, or NHRP, using existing IP routing protocols.

Integrated PNNI (I-PNNI) is the more ambitious alternative to PAR. I-PNNI is an extension of the PNNI to carry internetwork layer routing information, thus allowing routing information to be exchanged between the ATM control plane and Layer 3 protocols such as IP. This results in a nearly complete integration of ATM and non-ATM networks. In this approach, ATM switches and IP routers all appear as nodes in the overall topology map, as they do in PAR. Reachability and metric information can be calculated based on this combined topology and thus can be used to find the best routes. Using I-PNNI in IP routing might make I-PNNI the first fully QoS-aware routing protocol on the Internet. It allows the router to select special paths for QoS sensitive packets.

However, deployment of I-PNNI across any substantial portion of the Internet is unlikely in the foreseeable future. It requires major changes in organization. Both routers and switches must adopt the PNNI Peer Group hierarchy, node identifiers and peer group identifiers. Both switches and routers announce local topology via PTSPs, though I-PNNI introduces a new **PNNI Topology State Element (PTSE)** which uses TLV encoding to carry IP addressing information separately from ATM addressing information. Most profoundly, standard IP routing protocols are replaced by I-PNNI routing; although I-PNNI routing does not change the way in which IP routers announce IP address reachability from the way in which it is now done under OSPF and IS-IS, it *isn't* OSPF or IS-IS, and there will be great resistance to its deployment within the Internet community. This is only likely to change if and as IPv6 becomes accepted, since as we recall IPv6's **Nimrod** routing and addressing is very close in spirit to PNNI.

I-PNNI can also be used in a routing domain in a private network which contains both ATM and legacy networks. **Border Gateway Protocol (BGP)** can be used to exchange routing information with other routing domains. This is the most likely scenario for its early deployment.

5.5.3 Multi Protocol over ATM (MPOA)¹⁴¹ – A futuristic approach

In essence, MPOA expands on schemes like **LAN Emulation (LANE)** from the ATM Forum, as well as **Classical IP over ATM**, **Next-Hop Routing Protocol (NHRP)**, and **Multicast Address Resolution Server (MARS)** from the IETF (Internet Engineering Task Force). Each of these approaches solves a piece of the ATM internetworking problem; what makes MPOA different is its ability to integrate these solutions into a unified whole. What's more, it adds a new concept: virtual routers.

In a nutshell, MPOA does three things. First, it defines a high-performance, low-latency way to route IP and other protocols across an ATM switching fabric. Second, it lets net managers build virtual

¹⁴⁰ Callon, R. - *ATM Forum 94-0789: Integrated P-NNI for Multiprotocol Routing*, September 1994.

¹⁴¹ RFC 1754 - *Recommendations for the ATM Forum's Multiprotocol BOF*

subnets that span routed boundaries, so users can be grouped together as part of a virtual network regardless of where they are physically located in the network—even if they are not directly connected to ATM. Finally, MPOA permits applications to use ATM's quality-of-service capabilities. As noted, MPOA is based partly on earlier solutions to these problems, like LAN emulation. Like conventional bridged networks, LAN emulation nets can support all higher-layer protocols, such as TCP/IP and IPX/SPX. LAN emulation also simplifies the development of ATM adapters and bridges, freeing vendors from the necessity of implementing network-layer intelligence.

But LAN emulation networks are susceptible to the same inherent scaling and performance limitations that plague traditional bridged networks. Connecting virtual LANs still requires traditional routers, which can introduce performance bottlenecks. Even the fastest routers, capable of handling up to 500,000 packets a second, offer only a fraction of the capacity of today's low-end ATM switches. As the number of ATM-attached hosts within each emulated LAN increases, and the number of ATM switches in each emulated LAN grows apace, conventional routers will not be able to keep up.

These routers also introduce latency: Even if both the source and destination are ATM-attached, the source has to establish an ATM virtual circuit (VC) to the router, which then sets up another circuit to the destination. Scalability also can be a problem. The current version of the LAN emulation spec doesn't allow traffic from multiple virtual LANs to share the same virtual circuit. This limits scalability because as the number of virtual LANs grows, the number of virtual circuits grows too. Each virtual circuit requires a certain amount of setup overhead. Further, switches can't support an infinite number of virtual circuits. The ATM Forum is addressing this issue in version 2.0 of the spec.

The MPOA work has several goals:

- ⌘ Provide end-to-end Layer 3 connectivity across an ATM network, for hosts either directly attached to the ATM network or indirectly through routers on non-ATM IP subnets.
- ⌘ Allow formation of heterogeneous IP subnets (or subnets based on other network-layer protocols) across both ATM and non-ATM networks.
- ⌘ Provide direct connectivity between ATM-attached devices below Layer 3.
- ⌘ Ensure interoperable, distributed routing across all network segments, using both routing and bridging information to locate edge devices nearest an addressed end system.

The design of MPOA has largely been the creation of a framework under which existing ATM elements and legacy internetworking elements can be brought together. It is a new model only in this regard. The building blocks of MPOA have been discussed above:

- ⌘ LAN Emulation (LANE)
- ⌘ Next-Hop Resolution Protocol (NHRP)
- ⌘ Multicast Address Resolution Server & Connection Server (MARS/MCS)
- ⌘ UNI 3.1 signaling (optionally, UNI 4.0) and RFC1483 encapsulation
- ⌘ IEEE 802.1d spanning tree protocol for VLAN support

In this model, the behavior of the system is modeled using Logical Components. There are two kinds of Logical Components: MPOA Servers and MPOA Clients. A collection of functions provided by a single Logical Component is called a Functional Group (FG). Forwarding and routing functions are now modeled using different Functional Groups and can be provided by different physical boxes. This allows the definition of Virtual Routers, where the route calculation is performed in a distributed fashion by a collection of route servers, which together present the behavior of a traditional bridge/router. A key benefit of MPOA is intended to be the integration of intelligent **VLANs (Virtual LANs)**.

Hiding Protocols

LAN emulation effectively hides the network- and higher-layer protocols from the ATM fabric, which introduces a new set of problems. One is additional protocol overhead. For example, IP hosts have to go through extra steps to get the destination's ATM address. These steps result in high amounts of broadcast traffic. Since each host must listen to every broadcast, this wastes CPU cycles on all machines. Moreover, the process consumes network bandwidth. Two problems result from the MAC-to-ATM mapping. First, applications running over a LAN emulation network can't take advantage of ATM's multimedia and quality-of-service attributes. This is true even if these apps use network-layer quality-of-service protocols like the IETF's **Resource reSerVation Protocol (RSVP)** because there's no mechanism for passing the RSVP quality-of-service request down to the ATM fabric.

Second, there's a performance impact that comes from LAN emulation's maximum frame size limitations: The spec requires all devices in an emulated LAN to use the same maximum frame size, also called the **maximum transmission unit (MTU)**. Ethernet MTUs are on the order of 1,500 bytes, but ATM attached devices can field much larger MTUs, and larger MTUs can mean better throughput. But in an emulated LAN with Ethernet and ATM hosts, the ATM devices must use the smaller Ethernet MTU. The IETF has defined specifications to provide native IP support over ATM, thus solving some of these problems. This means that IP nets (though not those running other protocols, like IPX) can run over ATM without requiring LAN emulation software. IETF Requests for Comment (RFCs) 1577 (Classical IP over ATM) and 1483 (multiprotocol encapsulation over AAL) define ways to deploy IP networks that map directly to the ATM fabric.

One-Stop ARP

One of the key contributions in RFC 1577 is the definition of an enhanced IP ARP (address resolution protocol) mechanism. An ARP server works like a LAN emulation server but responds to queries for network-layer addresses instead of MAC addresses. Mapping IP directly to ATM eliminates some limitations of the LAN emulation model. For example, it reduces the protocol overhead resulting from address translation. With IP over ATM, ARP requests are forwarded directly to the ARP server, which replies with an ATM address. This gives the originating station everything it needs to signal an ATM connection to the target destination in a single step instead of a complex back and forth process. By cutting down on the number of steps needed to establish connections, this approach minimizes broadcast traffic and improves latency. Another advantage of mapping IP directly over ATM is the ability to use large MTUs, because network-layer devices understand how to handle IP fragmentation.

Even so, fundamental restrictions remain. A significant drawback of the current RFC 1577 spec is that like LAN emulation it needs a conventional router to interconnect different subnets. As a result, the throughput and latency problems described earlier still apply. The IETF **Routing Over Large Clouds(ROLC)** Working Group is developing the NHRP, which will help solve this problem. Another limitation is the fact that these RFCs don't define how to handle broadcasts or multicasts. The IETF IP Over ATM Working Group is currently developing the Mars scheme to address this difficulty.

IP over ATM also requires expensive full-blown routers for linking Ethernet/token ring segments to ATM, as opposed to some of the newer low-cost LAN switching products being developed today. Finally, the scheme defines only how to handle IP over ATM. There is still a need for a way to handle routing for other network-layer protocols.

Making Good

That's where the ATM Forum's Multiprotocol Over ATM Working Group comes in. MPOA ties together many of the issues addressed by earlier schemes. In particular, MPOA supports multiple network protocols over ATM in an efficient, scalable manner. It also defines network-layer virtual subnets that can span geographic boundaries. Further, MPOA allows traffic to be forwarded directly to its destination over a one-hop virtual circuit. By mapping network-layer protocols directly to ATM,

MPOA reduces overhead and broadcast traffic over the network, supports variable-size MPUs, and paves the way for applications to take advantage of ATM's underlying quality-of-service capabilities. Finally, MPOA defines a standard for virtual routing and virtual routing provides for a cost-effective and scalable way to handle routing over ATM. The MPOA architecture comprises the following components:

Edge devices. Sometimes referred to as multilayer switches, edge devices are intelligent switches that use either the destination's network-layer address or its MAC-layer address to forward packets between legacy LAN segments and ATM interfaces.

ATM-attached hosts. These are ATM adapter cards that implement the MPOA protocol as part of their drivers. They let ATM-attached hosts communicate efficiently with one another or with legacy LANs connected by an edge device.

Route servers. Not physical devices as such, route servers are a collection of functions that make it possible for network-layer subnets to be mapped onto ATM. Route servers can be implemented as standalone products, or they can consist of software added to existing routers or switches.

Serving the Route

In essence, route servers subsume the functions of the LAN emulation broadcast/unknown server and ARP servers. A route server maintains network-layer, MAC-layer, and ATM address information, which is used to establish direct virtual circuits between any two end-points (edge devices or ATM-attached hosts) that must communicate with each other. To communicate routing information with conventional routers, the route server runs protocols like RIP (routing information protocol), OSPF (open shortest path first), and IPNNI (integrated private network-to-network interface), ensuring interoperability with routed internetworks. In place of emulated LANs, MPOA defines Internet address summarization groups or virtual subnets. These groups denote both a Layer 3 protocol and an address range. In other words, they define a linked set of hosts and a particular protocol linking them. For example, an Internet address summarization group might consist of an IP subnet, meaning the associated IP devices plus the IP protocol.

The MPOA Working Group has agreed to furnish hooks that allow address summarization group traffic to be backwardly compatible with LAN emulation. This means that adapters and edge devices incorporating LAN emulation software can communicate with MPOA devices if they are within the same Internet address summarization group for example, if all devices lie within the same IP subnet. Allowing hosts to communicate with devices in other Internet address summarization groups requires more than this, however. In particular, the cards and switches must support MPOA rather than just LAN emulation, and a route server must be present. LAN emulation-only edge devices can communicate with other MPOA Internet address summarization groups solely through a router or gateway device that supports both LAN emulation and MPOA.

Virtual Routers

A key aspect of the MPOA model is its call for virtual routers, a set of MPOA devices operating over an ATM fabric that collectively provide the functionality of a multiprotocol router (see Figure 1). Since the edge devices accept data from an attached subnet, they are analogous to router interface cards. The ATM switching fabric can be seen as the backplane of the router, linking edge devices. The route server is analogous to the control processor.

The virtual routing approach makes it possible to deliver routing functions more efficiently and cost-effectively than today's routers can: Edge devices don't have to be as intelligent as a full-blown router. It also makes for more efficient scaling because adding forwarding capacity simply means adding switches, and adding additional routing capabilities means adding software to the route server. Management is easier too: The whole virtual router architecture comprising multiple switches and route servers can be managed as a single router. Finally, virtual routing allows for the creation of virtual subnets, since an Internet address summarization group can contain hosts that physically lie anywhere in the network.

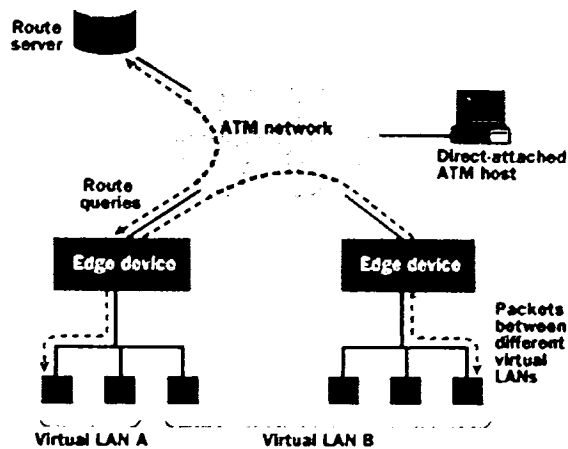


Figure 81: MPOA in operation

A significant portion of the MPOA work involves defining the protocols between the route server and the edge devices; the MPOA model basically splits the routing functionality between these two components, and standardizing on this protocol means that edge devices from one vendor will be able to work with route servers from another. It's also possible to implement both route server and edge device functions in one box (a physical as well as virtual router). In this case, having a standardized protocol isn't as important, since everything happens within the router.

That said, the whole point of the MPOA model is to divide routing up so that vendors can choose to implement different kinds of functions in different devices. Here's how it works. Edge devices examine the destination address of packets received on legacy LAN segments and decide how to forward those packets. If the packet doesn't need to go outside the Internet address summarization group, the work of the edge device is done: It merely bridges the packet, using LAN emulation to resolve the ATM address and establish a virtual circuit to the destination. If the packet must be routed, the edge device examines it to determine the destination network-layer address of the edge device and looks up the ATM address corresponding to that network-layer address. The edge device then establishes a direct virtual circuit to the appropriate destination.

The edge device gets the ATM address from either the route server or its own memory cache. The route server knows or can use various routing protocols to discover the ATM address of any device in the network. However, the design goal here is to minimize the number of times the edge device must visit the route server to retrieve this information. To that end, the edge device maintains its own address cache. Much of the MPOA effort is devoted to devising effective cache-management techniques, including ensuring cache coherency between edge devices and route servers. At no point in this process must the packet be forwarded to a standard router. Instead, the edge device handles packet switching, while the route server performs address and routing resolution. It is hoped that this architecture will eliminate the scaling and performance bottlenecks described earlier.

If the local route server doesn't know the appropriate ATM address, it can propagate the query to other route servers. The destination ATM address that the route server provides is either the address of the recipient host (if the host is ATM-attached) or the address of the edge device to which the non-ATM host is attached.

Migration Issues

One of the key issues addressed by the MPOA Working Group was interoperability with routed architectures. The entire MPOA virtual router including edge devices, route servers, and the ATM infrastructure is designed to work with existing routers using today's router protocols. Thus, it should

be possible for net managers to use MPOA in conjunction with, rather than as a replacement for, their current routed networks. Moreover, as the routing function becomes more virtual, the products implementing the virtual router can become more specialized. Edge devices and ATM switches become the network's brawn, focusing on high-performance bit-hauling and switching, and route servers become the network's brains.

That's not to say that traditional routers simply disappear. Instead, they migrate toward the provision of specialty services, such as packet conversion between ATM and other WAN services or firewall filtering. MPOA also must coexist with emulated LANs, and routers can come in handy here. Depending on how the adapters and edge devices in a LAN emulation network are implemented, the network may require routers to integrate with MPOA.

5.5.4 IP Switching¹⁴² – An alternative approach

The basic assumptions at the core of internetworking were to a large extent quite contradictory with the classical telco's ideas. Indeed, the Internet focused on a packet-based, connectionless paradigm, which mainly catered to the needs of data networks. This translated in a fundamental move of intelligence from the internals of the network to the edges of the network, as a lot of state information traditionally maintained within the connection-oriented network elements simply disappeared, taking with it the associated service intelligence. A second important difference was that the Internet Protocol was designed as a network layer protocol purposefully build to run on top of any link layer technology available now or in the future. Therefore, what we have today is an ubiquitous internetwork, providing universal connectivity, universal addressing and a universal service development layer.

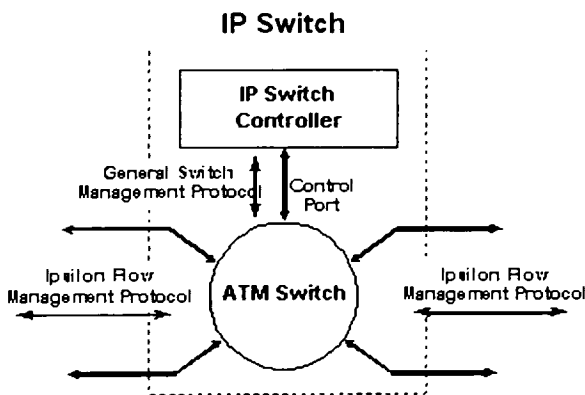


Figure 82: Ipsilon's IP switch implementation

Today, we just do not have precise figures about the number and the duration of (end to end) flows generated by applications in a mixed data and multimedia environment. Facing with this situation, it is preferred that the network adopts a protective strategy where routes are computed and maintained in advance independently of actual flow requests. This is exactly the strategy adopted by IP routing. However, we should remark that, with the IP technology, the network does not have to maintain a route towards each individual destination but have only to maintain routes towards a group of destination or a group of network. This aggregation property makes the IP systems very scaleable in terms of network states that we have to maintain in backbone switches/routers.

¹⁴² IP Switching, <http://www.telecoms-mag.com/marketing/articles/sep97/cattell.html>

The need for an Intermediate CO layer

Now, the question is do we still need in IP networks an intermediate stratum between IP and the transport infrastructure based on a **Connection Oriented (CO)** -mode of operation (CO L2)? Is ATM a good candidate for this CO L2 ?

One could debate the following arguments in favor of a connection-mode of switching beneath the IP layer:

- ↳ **Faster forwarding:** In pure IP networks (e.g. Packet over Sonet), all individual network layer packets have to be processed at layer 3 before forwarding. The process has, then, to be repeated at each of the intermediate nodes to a destination. "Shortcut routing" refers to the ability to forward network layer packets (L3) directly at the data link layer (L2). Some years ago, when shortcut routing was proposed, it was felt that it would result in a gain in forwarding speed. We could observe that this is no longer true because, today, we can achieve L2 switching and L3 forwarding at the same speed with dedicated hardware.
- ↳ **Traffic engineering:** in connectionless-mode network like IP, the routes towards the destination are usually the shortest paths. However, as many shortest paths might run along the same links, congestion could occur on some links, while other links remain unused leading to sub-optimal use of the available resources. Traffic engineering tries to avoid this by relaying traffic from the congested links to less utilized links. We can perform traffic engineering by manipulating routing metrics (e.g. administrative weight), e.g. by increasing the cost of using a congested link, some of the traffic going over that link will choose other paths towards their destination. However, this method is insufficient and complex: metric adjustments of one (congested) link will cause a change in many shortest paths, which might introduce new congested links. The ultimate way to achieve traffic engineering is to establish CO L2 paths: this method results in isolating and deviating the traffic between some ingress router and egress router from the (congested) shortest path by forwarding this traffic along a CO L2 path.
- ↳ **Virtual Private Networks services:** a very popular service for the Internet is VPN. Here, the issue is to provide security and QoS. This can be achieved by maintaining CO L2 paths between the corporate networks preventing third parties from using these labeled paths (security) while providing QoS.

We can conclude that there is a need for a intermediate CO layer between the transport infrastructure and the IP layer because a CO layer will provide traffic isolation which is a very powerful tool to implement advanced and required features like VPN and traffic engineering. ATM is currently the only CO technology switching at the high speed required by the Internet. An additional advantage of ATM is that it inherently offers QoS, which is, despite the effort of the IETF, not yet the case for IP. The combination of IP and ATM will allow to provide some level of QoS in the Internet.

DiffServ as an interim measure for QoS provisioning

Enterprises banking on the promise of combined voice, video, and data networks with guaranteed quality of service (QoS) may be in for a surprise when they try to run delay-sensitive applications across both LANs and WANs. Vendors and observers acknowledge that despite a major push for multiservice networks, mechanisms for prioritizing traffic across all of the technologies that go into a typical enterprise network - Ethernet, frame relay, and ATM - are still in their infancy in meeting the needs of multi-service applications. The users enticed by emerging QoS technologies for packet networks, such as Resource Reservation Protocol, IP Type of Service, and IEEE 802.1p tags, should not assume that the priorities established by those tools can be carried across WAN gear or a service provider's network.

The **Differentiated Services (DiffServ)** working group of the Internet Engineering Task Force (IETF), also called DiffServ, is in the early stages of developing a priority specification for IP traffic on WANs. This specification will place a DiffServ bit, chosen on the basis of IP priority mechanisms and preset profiles, on each IP packet. The standard would establish a common understanding of this bit. Because ATM has a finer quality of service than DiffServ and DiffServ's relatively crude settings will not be enough to provide meaningful QoS across a WAN. But, on particular schemes for DiffServ classes of service, mainly within the community of Internet Service Providers (ISPs) offering

backbone Internet connectivity, DiffServ is considered as an interim solutions to provide the QoS, though it is still far from wide deployment. ISP's enthusiasm is backed by the fact that DiffServ approach scales well to their needs and investments (there is no need to keep a per flow state in transient routers) and is relatively easy to implement. On the other hand, it is not the best approach to ensure QoS guarantees. Further, it's yet unclear as to whether common agreement could be made in a short time on particular number of queues and priority levels (drop preferences) within each queue supported by DiffServ. Opinions vary from 8 levels with 3 priorities in each level down to 2 levels in total. There is no particular progress in mapping DiffServ to ATM, especially with regard to the ATM classes of service. A recent proposal on a guaranteed frame rate for DiffServ over ATM could be mentioned as one of a few examples towards meeting this objective.

Most promising scenario for DiffServ over ATM is to use a DiffServ enabled **Virtual Private Network (VPN)** overlaying existing Internet connectivity. In this case VPN flows with QoS assurance will be tunneled among DiffServ routers in several domains. If these advances will happen and be deployed as predicted now, and then linked with the emerging MPLS technology, then it is very much likely that DiffServ will be used in next generation networks. It is very questionable whether DiffServ will find flexible and elegant mapping to ATM service classes to exploit full capabilities of this link layer technology. On the other hand, RSVP could become in this future Internet a general purpose IP signaling used to control and manage DiffServ/MPLS aggregated flows over the backbone. In any case, when these advances will happen, it is most likely that existing equipment will already be obsolete.

The latest version of the IPv6 specification includes a traffic class field of 8 bits, which within the Differentiated Services working group is being termed the "DS field". Six bits of the DS field are used as a codepoint (DSCP) to select the per-hop behavior packet experiences at each node. A two-bit field is currently unspecified and ignored by differentiated services-compliant nodes when determining the per-hop behavior to apply to a received packet. While work on the mapping between Differentiated Services and ATM is in its infancy (and in particular the mapping to specific ATM classes) future work will exploit the efficient processing of the IPv6 header and the associated DS field structure to provide better delay characteristics.

MPLS as the model for IP over ATM

We have shown in the previous section the benefits of having IP combined with CO L2 switching technologies like ATM. ("IP over ATM"). Hence, a model to support IP over ATM networks needs to be defined. The industry and research institutes have proposed several architectures and discussed in standardization for, the most promising models are MPOA and MPLS. All those schemes for IP over ATM can be classified according to the routing model they use, i.e. the layered routing or integrated routing. Integrated routing means that a single routing stratum is used by IP forwarding and by ATM switching function whereas, with layered routing, the ATM is using its own routing layer (e.g. PNNI) which is disconnected from the IP routing. MPOA is a way to carry IP over ATM using a layered routing scheme whereas MPLS falls in the category of integrated routing

Integrated routing is recommended because it simplifies address resolution, problem which is inherently present in any layered routing environment, and it provides an optimum strategy for resource management which can not be achieved at all with a layered routing approach:

- ⤵ Layer integration does not merely simplify address resolution, it removes the need for address resolution altogether as there is only one addressing mechanisms established in the integrated layer.
- ⤵ IP multicast: with the integrated routing approach, the direct use of IP multicast protocols on top of ATM entirely eliminates the need for complex protocols trying to emulate a broadcast on top of a non-broadcast network such as ATM.
- ⤵ Traffic engineering: with the layered routing approach, efficient traffic engineering is almost impossible because IP routing will not view the resources of the ATM network. On the other hand, with the integrated routing approach, IP routers and ATM switched share the same view and, hence, optimum traffic engineering can be achieved.

MPLS used in conjunction with ATM, because it corresponds to add on overlay CO L2 network over the transport infrastructure, will provide today the means to deploy extra essential features like VPN and traffic engineering which can not be optimally implemented at the IP level. In addition, MPLS being based on the integrated routing approach will provide the tools to deploy traffic engineering, VPN, and IP multicast in a scaleable way. In addition, the association of MPLS and ATM enhanced with RSVP provides the necessary elements to establish QoS flows over the Internet. We can conclude that IP/ATM with MPLS combines the best of both world: it offers the scalability properties of IP and IP routing protocols while optimizing the use of ATM for its first customer, i.e. IP.

5.5.5 Multi-protocol Label Switching (MPLS) – An evolutionary approach

Multiprotocol label switching (MPLS)¹⁴³ is not a silver bullet to cure existing or forthcoming problems, but rather an enabling technology, which addresses some of these scaling issues. It does this by replacing the standard destination-based hop-by-hop forwarding paradigm with a label swapping forwarding¹ paradigm. This has the benefit of simplifying the packet-forwarding engine, enabling easy scaling to terabit rates. Furthermore, it decouples forwarding from routing, enabling one to apply new specialized or customized routing services without requiring changes in the forwarding path.

MPLS has largely evolved from the need to use the high speeds of existing label-swapping technologies, such as asynchronous transfer mode (ATM), with existing router networks, by seamlessly integrating the datagram nature of IP to the label-swapping hardware capability of switches. This approach eliminates several of the problems associated with overlaying an IP network on top of a connection-oriented network.

All of the IP-over-ATM approaches have several scaling issues. The approaches described above use a complete or partial mesh of VCs to connect host members in a logical subnet. This implies that at maximum $n(n - 1)$ connections are required to connect n members in a logical subnet. This can consume a lot of connections for large memberships, increasing the overhead to set up, maintain, and tear down these connections. Besides the resource factor, each member in a logical subnet has $n - 1$ routing peers. This drastically increases the computational overhead for the routing protocols because the computational overhead is directly proportional to the number of links (physical or virtual) in a network.

In both CIP and LANE, the data must be forwarded between logical subnets, which does not fully exploit the switching infrastructure. Multicast is also based on logical subnets, and must be IP forwarded between subnets. Although NHRP and MPOA provide mechanisms to shortcut the VC across multiple subnets for unicast data, the mechanism is traffic-driven and thus has associated setup, maintenance, and teardown overheads besides the caching latency issues. The salient selling point of switches, besides their low prices and fast speeds, was QoS. However, none of these approaches exploit that potential. The connections used in these schemes are still equivalent to best effort. The protocol and signaling mechanisms only increase the complexity of configuration, maintenance, and operations of an overlay network. Thus, the benefits of switching were available at an unreasonable cost.

Label switch routers (LSR) in MPLS use link-level forwarding to provide a simple and fast packet-forwarding capability. Normal network layer forwarding requires parsing a relatively large header and performing a longest match algorithm in order to forward packets. However, label-swapping packet forwarding is based on a simple short-label exact match. This results in a simpler forwarding paradigm.

Network-layer routing maintains information from common routing protocols (such as OSPF and BGP) to determine how packets ought to be routed. This routing information partitions the entire forwarding space into **Forwarding Equivalency Classes (FECs)**. A set of packets following the same

¹⁴³ Jim Metzler, Lynn DeNoia, *Layer 3 Switching*, Prentice Hall, 1999

path, belonging to the same FEC, is also referred to as a "stream" and forwarded in a similar manner. In turn, each FEC is assigned a short, fixed-length, locally significant identifier known as a "label." A packet is "labeled" by either encoding a label in an available location in the data link layer or network-layer header, or encapsulating the packet with a header specifically for this purpose. As a packet enters an MPLS network, a conventional layer-three lookup is performed; however, in addition to the conventional next hop, the associated FEC with the assigned label is found. The packet is forwarded to its next hop with the assigned label. At subsequent nodes, the label is used as an index into a table, which specifies the new outgoing label and next hop. The old label is replaced with the new, and the packet is forwarded to the next hop. This eliminates the need for network-layer lookups at all but the first node in the path. Information from the routing protocols is used to assign and distribute labels to MPLS peers. In general, an MPLS node receives an "outgoing" label mapping from the peer that is the next hop for a stream, and allocates and distributes "incoming" labels to upstream peers for a given stream. The labels are extended into a switched path through the network as each MPLS node "splices" the incoming to outgoing labels. This series of one or more concatenated labels is termed a **Label Switched Path (LSP)**. The label distribution for the unicast is done via the **Label Distribution Protocol (LDP)**. MPLS neighbors create an LDP peering session for the exchange of LDP's distribution and withdrawal procedures. LDP supports two styles of label distribution: independent or ordered. In the case of independent label distribution, each node may at any time distribute labels for each stream it recognizes. In the case of ordered control, label distribution for the node for that stream initiates a stream. An LSR is considered an egress LSR for a stream if its next hop for that stream is not an LSR or the node is located at a routing boundary. Any given node may only distribute the incoming labels for each stream it recognizes if it is the egress, or if it has an outgoing label for the given stream. This method ensures that label-to-stream mappings are consistent throughout a network with regard to granularity, and provides a higher probability that unlabeled packets are not forwarded to downstream nodes.

Downstream peers perform label allocation within MPLS, where downstream is defined with respect to routing. There are two forms of label allocation: downstream and downstream-on-demand. In downstream allocation the label assignments are made by the downstream node and distributed to the neighboring LSRs. In downstream-on-demand allocation an upstream LSR specifically requests a label assignment for a stream from a downstream node. Downstream-on-demand allocation is useful in ATM networks where merging of LSPs is not possible. In both independent and ordered distribution, labels can be distributed in liberal or conservative mode. In the liberal mode, an LSR R assigns labels for all neighbors for a FEC. The neighbors maintain the label from Rd even when Rd is not the downstream next-hop for that FEC. This way an LSR can immediately start using a pre-distributed label when a next hop changes. In the conservative mode, labels are maintained only by those neighbors of Rd that are using Rd as the downstream next hop for that FEC. This scheme is useful in scenarios when the LSR has limited label space. The liberal and conservative allocation mode are interoperable, and the mode of operation is exchanged by the LDP peers when initializing an LDP peer session

MPLS has emerged as a promising technology that will improve the scalability of hop-by-hop routing and forwarding, and provide traffic engineering capabilities for better network provisioning. It decouples forwarding from routing and allows multiprotocol support without requiring changes to the basic forwarding paradigm. The MPLS standardization effort is still in a relatively early stage, and there are a number of technical issues, which need to be resolved before the standard is complete. Still further work is needed before multivendor interoperability becomes a reality. However, MPLS has the potential to offer a useful technique for improving several aspects of network operation.

5.5.6 IP Integrate service over ATM¹⁴⁴ – An integrated approach

The concept of IP Integrated Services¹⁴⁵ has largely been dealt through Resource Reservation, and the **Resource reSerVation Protocol (RSVP)**. RSVP fits reasonably well with the design philosophies of

¹⁴⁴ RFC 1821 - Integration of Real-time Services in an IP-ATM Network Architecture

the Internet; it maintains only soft, periodically refreshed state information in the intermediate nodes, while most of the control states and related complexities are maintained in end systems. This allows RSVP to adapt automatically to route and membership changes. However, it is an unfinished specification, has not been widely implemented, and it is not clear whether and how it will function in a global Internet, for administrative as well as technical reasons. As previously mentioned, there is a substantial sentiment within the IETF that "best effort" is good enough so long as "infinite bandwidth" is available. But of course there is no agreement on this approach, and RSVP is the alternative.

It took the RSVP working group more than 2 years to finalize the protocol. Now the final version (version 13) of the RSVP draft standard is available and is believed to be on track for RFC standardization soon. Alpha versions of RSVP implementations are also available. RSVP is not a routing protocol. It may operate together with any routing protocols available on the internet. Its aim is to accept a path (or distribution tree), however obtained, and set up and maintain resources over that path for the use of the entity which requested that reservation. It also introduces the concept of *filtered reservations*. It is possible to make reservations, which may only be used by packets from certain specified sources, and the list of allowable sources may either be fixed or dynamically changeable with time. Such flexibility is particularly important for multicast groups. A fixed filter allows switches or routers to merge individual reservation requests, knowing that the characteristics of the reservation will not change. Conversely, a dynamic filter reservation gives the receiver the ability to change sources from time to time, or "change channels".

RSVP features receiver initiated reservation which means that the receiver is responsible for joining the distribution tree and setting up reservations on all the intermediate nodes. This decision is consistent with the receiver-initiated establishment of multicast distribution trees. The decision also enables RSVP to accommodate heterogeneous receivers on the same distribution tree, which leads to an extremely flexible and scalable design.

At first glance, implementing IP Integrated Services over ATM seems a natural match, since ATM's built-in support for QoS, and native connection-orientation, were designed with service guarantees in the forefront. On deeper consideration, though, this seems to present intractable problems, given the different logistics of the two technologies. Both the underlying network (ATM) and the overlay network (IP) have mechanisms for implementing service guarantees. Since these mechanisms differ, there is no agreed upon "division of labor" between the layer mechanisms or means of communicating QoS parameters, and most importantly, no agreement that Layer 3 entities *should* be able to communicate QoS parameters to one special type of data link (Layer 2) network, there are clearly many hurdles to overcome. Still, a good deal of the work has already been done. Many issues related to running RSVP over ATM are still unsettled. In an IP-over-ATM environment, ATM SVCs are usually used to support QoS guarantees and RSVP is used as the internet level signaling protocol to convey the QoS requirements. Several issues have to be addressed in such an environment. The first issue is when and where to use SVCs.

In the LIS model, this is not a problem since full-mesh VC connection within the LIS is assumed, and hosts never talk to anyone outside the LIS except through IP routers. RSVP over ATM¹⁴⁶ will operate in just the same way as over legacy networks. In the NHRP model, SVCs are used to establish direct short-cut links between sender and receiver when the data volume is high or the QoS requirement is tight. However only unicast short-cut VCs may be used since RSVP needs bi-directional connections to transmit RESV and PATH messages, while ATM point-to-multipoint VCs are unidirectional. Since NHRP for multicast is still not a reality, this is probably a good thing. Heterogeneity is another problem in RSVP over ATM. RSVP allows different receivers of the same session to have different QoS requirements, or even have different QoS classes. The ATM UNI 3.x and 4.0 only support homogeneous QoS for all receivers. To solve this problem, a VC, which has the maximum QoS requirement of all the receivers, has to be established. Of course, some resources are wasted, and some intended "best effort" receivers might get real-time links, resulting in an actual degradation of the desired QoS. So in the presence of best effort receivers in a session, it may be desirable to set up "best effort" VCs to each of them.

¹⁴⁵ RFC 2382 - A Framework for Integrated Services and RSVP over ATM

¹⁴⁶ RFC 2379 - RSVP over ATM Implementation Guidelines

RSVP also addresses QoS renegotiations and dynamic membership, currently not supported by ATM networks. QoS renegotiations and membership changes have to be implemented by setting up a new connection and tearing down the old one. These issues and questions are being dealt with in ongoing work in the IETF **Integrated Services over Specific Lower Layers (ISLL)**. This work is directed toward mapping RSVP to ATM UNI services. Borden et al. outline the issues raised by RSVP-over-ATM; Berger's draft gives guidelines for implementing RSVP over ATM. Berson and Berger and Williams et al. provide methods for using ATM VCs with QoS under RSVP. Borden and Garrett provide suggestions for service mappings between IP Integrated Services and ATM Quality of Service. Crawley narrows the focus to IP Integrated Services over LANE.

5.5.7 IPv6 and ATM¹⁴⁷ – Extending the boundaries

Internet Protocol Version 6 is the next generation Internet Protocol and was discussed in detail in the later part of the previous section. IPv6 has been designed largely to solve the address space crisis on the Internet. It is a more streamlined protocol which cleans up many "relics" in IPv4 and also provides many new features such as address auto configuration, security, hierarchical routing (Nimrod), flow labeling, etc. The Flow Label Field is not intended to be used in routing, though it is certainly considered as a forwarding label in flow based switching. It together with the source and destination addresses identifies the flow to which a packet belongs. The label is necessary since the packet might be encrypted so that the port and protocol information is no longer available to routers.

Deployment of IPv6 in NBMA networks (ATM networks in particular) presents a number of new challenges. Among the foremost is the use of address auto configuration. Address auto configuration allows a host to configure one or more addresses per interface automatically and without explicit system administration. However, IPv6 fundamentally changed the Address Resolution Protocol. In IPv4, a different ARP protocol is defined for each medium. In IPv6, defines a common **Neighbor Discovery (ND)** protocol for all media types. This ND protocol assumes that if the data link address of a certain node is not available, it still can be reached by sending a multicast message. This requires that the data link protocol inherently supports multicast, which means there is a straightforward mapping between the IPv6 multicast address and the data link multicast address. Unfortunately, as we know, ATM's native multicast support is weak. It was originally proposed that the MARS model be used; routers would perform block MARS_JOINS for an appropriate range of IPv6 multicast addresses. However, since the development of NHRP, ION has amended this strategy; while MARS is used for ND, for destinations not considered as neighbors hosts send packets to their default router. The router in turn issues an NHRP query to determine the target's ATM address, and on learning it issues a Redirect to the IPv6 source, identifying the flow destination as a Transient Neighbor.

This is one of the stickier issues for bringing IPv6 to ATM networks. Routing models are much closer between **IPv6 (Nimrod)** and **ATM (I-PNNI)**. However, IPv6 has not become as immediate an issue as we envisioned at the outset of this project/proposal. Deployment is in trial systems only, running over IPv4 tunnels. Alternative solutions to the "Address Explosion" problem which is IPv6's greatest one has reduced the urgency with which IPv6's development was then regarded. Foremost among these is **Classless Inter Domain Routing (CIDR)**. Other limitations of IPv4 still drive the development and eventual deployment of IPv6, but not to any great degree within the horizons of this report.

¹⁴⁷ RFC 1680 - IPng Support for ATM Services

5.6 Summary

The ability to support legacy IP traffic is vital to the success of ATM networks. Treating ATM as yet another LAN technology, Classical IP over ATM is the simplest to implement. However, the drawback of it is that inter-LIS traffic has to travel through a router even though both parties are directly connected to the ATM network. NHRP fixes this problem by augmenting it with an address resolution protocol so that shortcut connections can be established between end systems that belong to different LISs. To accelerate the deployment of ATM technology, LANE emulates Ethernet and Token Ring LANs on an ATM network so that existing IP software running on such LANs can run on ELANs without modification. However, ELAN suffers the same drawback as Classical IP over ATM, that is, inter-ELAN traffic has to travel through a router. MPOA combines LANE and NHRP technology to support both IP routing and LAN bridging over an ATM network. In addition to these data models, two routing schemes, PAR and I-PNNI are proposed to be used in the environment of interconnected ATM network and routers. PAR allows the routers automatically discover each other and build ATM connections to exchange routing information. I-PNNI allows PNNI protocol to be used in a hybrid mesh consisting of ATM switch and routers. Actually these routing enhancements can be used in align with any data model. But the intra-subnet models do not address QoS and as such we need to look at the Inter-subnet models more closely in asserting the best combination of models to address the integration perspectives of IP and ATM towards QoS centric, multi-service network.

On the other hand IP switching has taken an evolutionary approach to utilize ATM structure under IP to achieve terabit wire speeds and has found a great degree of success which has spawn off an initiative by IETF in terms MPLS. But, IETF initiatives are mostly based on RSVP, which is a signaling scheme rather than a routing protocol. If we need to assure QoS, we need a network that could base their routing/forwarding decisions on applications flows and desired quality of service. To date, we have only PNNI/I-PNNI which are capable of meeting this objective in a complex, dynamic environment that would prevail in a multi-service network. The question remains with the advent of this multiplicity how do we set about integrating the desired technologies and associated models to provide a multi-service platform which is scalable, flexible, reliable, time-sensitive and service independent in the most cost effective manner with the maximum reach. The networks of the future will be multi-service centric and they have to be ubiquitous in service delivery rather than mere reachability. Technologies will evolve.... applications will evolve we need to adapt them in way which would give us the desired impetus to take us to the next evolution in a graceful manner in trying to meet our service objectives in the most optimal manner.

6. Discussion – Is IP/ATM Integration the answer?

We did look into the intricacies of IP and ATM internetworking in light of our objective to find the best mean(s) of internetworking the two to realize the potential of a combined IP/ATM architecture has to offer future high speed networks in terms of QoS, performance, scalability, transparency, auto-configuration, operation & management, and many other features that it has to offer. We have reached a point in our research where we need to look at the network implementations of the past, present and make reasonable conclusions in making our recommendations as to the best way to approach the issues at hand in integrating IP and ATM. We use the word integrate in a deliberate sense, since we have come to a certain conclusion that IP and ATM have to be integrated to a greater extent to exploit the features of one another, rather than continue to run them as two separate networks or basing them on the traditional overlay model.

Asynchronous Transfer Mode (ATM) has been conceived as a multi-service technology since its inception. The introduction of new service categories within the ATM layer makes it more suitable for virtually unlimited range of applications ranging from video to data. By using these capabilities as service building blocks, users have the flexible access to the network resources and can achieve a satisfactory compromise between performance and cost. Additionally, network operators are able to flexibly share network resources among different customers and fulfill their needs in a cost-effective way.

On the other hand, the Internet was conceived on the principle of connectionless Internet Protocol (IP) datagram delivery whereby no per-session state need be set up in the network prior to sending a user's datagrams. Instead, each IP datagram is routed hop by hop toward its destination according to the destination's globally unique IP address, which is contained in the IP header. Each router that is traversed will examine this IP destination address and look for a match in its routing table in order to determine the correct outgoing interface for the next hop of the journey toward the destination. This connectionless model makes no assumptions about the underlying networks. Furthermore, it does not implement call admission control or per-flow resource reservation and consequently is unable to offer any QoS guarantees. The delivery service is termed *best-effort*, which means that each IP data flow is subject to an indeterminate level of packet loss, reordering, and delay, all of which increase with network load. On top of this core service, end-to-end reliability can be achieved through appropriate transport-layer protocols such as Transmission Control Protocol (TCP), which uses such techniques as positive acknowledgment and retransmissions. The pooling of resources inherent in the traditional Internet philosophy is a key strength that ensures high utilization of resources, while overload results in graceful degradation of service rather than total collapse. In addition, a connectionless model is very robust and handles an extremely wide range of failure scenarios without imposing a heavy signaling burden. But at what cost? The answer to this question lies in our objective to seek the best approach in balancing off the many factors that come into play in making such a choice.

Where is this driving us? The classical model of IP has proven incredibly successful, and the number of hosts on the Internet continues to double approximately every year. Within this overall growth rate there is also an increasing demand for multimedia applications that ask a lot more from the network than the more traditional types of Internet traffic such as File Transfer Protocol (FTP). In fact many of these multimedia applications have quite stringent **Quality of Service (QoS)** delivery needs in terms of packet delay, loss rate, and minimum bandwidth. Furthermore, as the World Wide Web is increasingly used for businesses and Electronic Commerce, there are a growing number of users for whom delay-bounded access of information is especially important. It is clear that if the Internet is to keep pace with such demands it must offer QoS support as well as increase the bandwidth available to end users. QoS support for IP flows at the IP layer is feasible by reserving resources on a per-flow basis which can be initiated by the user on demand using a reservation protocol such as RSVP. Also, the advents of fiber optic cables and associated **Wave Division Multiplexing (WDM)** technologies have resulted in a transmission medium with massive potential bandwidth capacity.

However, the bottleneck arises at the communications nodes used to interconnect such media. The concept of IP routing was geared more toward flexibility than speed; class of service than quality of service; cost than performance, etc. Consequently, for a given cost, switches based on

asynchronous transfer mode (ATM), which were designed from the outset with high switching speeds in mind, are able to operate at higher bit rates than conventional IP routers. ATM achieves high bit rates through hardware switching (based on ASICs) of fixed-length cells with a small fixed-length header based on a one-to-one match with switching table entries. In contrast, forwarding decisions in an IP router were traditionally carried out in software (based on Microprocessors) on the longest match of the address prefix with entries in a routing table. In addition to ATM's ability to switch at high speed, current implementations also support QoS on demand, so it may appear as though ATM solves all the problems the Internet is currently experiencing. In reality such an assumption is untrue, although a detailed discussion is beyond the scope of our research.

It will suffice to say that both ATM and IP have their own relative strengths and weaknesses, which explains why as they both evolve they do so toward a common point somewhere between the two technologies as they were initially conceived. For example, the introduction of resource reservations and per-flow state within IP networks **Class of Service (CoS)** mimics the ATM philosophy, while the introduction of the **Available Bit Rate (ABR)** service into ATM provides a similar service to that of TCP over IP.

Now the question is what is the best approach to deliver QoS (rather than CoS) in multi-service networks which would scale from the needs of an Enterprise network to that of the Internet in a seamless manner as to be transparent to the end nodes immaterial of where they are in the global Internet cloud. To recapture couple of key issues from our research findings we dwell into an in depth discussion on the key issues facing both IP and ATM and their related flow-based models namely MPLS and MPOA which could complement each other in its own way in the whole integration architecture and related perspectives.

6.1 IP and ATM – What are the Open Issues?

IP is ubiquitous. There is no doubt that IP is one of the greatest success stories in the history of information technology. Anyone who has closely followed the growth of the original BITnet to its current form would agree on this. But this is the past. Today IP and the structures built upon it, such as the World Wide Web, are changing our lives, our interactions, and our businesses. The growth rates for IP traffic and users are unique and immense - networks that once appeared inexhaustible have been brought to their knees. We have to agree the IP network model was never built to cater to all these growing demands imposed on it by the current business model, but on the contrary has evolved and sustained the immense load placed on it. In essence we need to reengineer the network and related architectures to make it optimal to cater to the future needs of an imminent, multi-service network. The current work on IPv6 is only one of the desired reengineering processes, which is afoot. What are the other desired reengineering processes?

Today IP is moving from the realms of academia and entertainment into mainstream business. E-commerce, Web based business, corporate intranets, and extranets are clearly the future. At the same time the world's public network infrastructures have been moving forward on a plan that stretches over decades: A plan that was based on the emergence of broadband services and applications; A plan that extended the global public telephone and data services of today into an all-encompassing, global, networking infra-structure for tomorrow. Moreover, this plan is evolving around a very specific technology - and that technology is ATM.

ATM is no marketing whim; ATM is the well-defined and logical extension of all the **Broadband Integrated Services Data Networking (B-ISDN)** work that has been progressed for the last 10 years. So today, we are running towards a world with two de-facto realities, IP services and an ATM infrastructure. We need to look at how this world will be structured, to identify what other technologies are needed to bring it to fruition, and to ascertain what benefits will flow from the merger of these two technologies.

6.2 What are our Objectives?

It is our objective to achieve service ubiquity in terms of guaranteed QoS delivery and not just network ubiquity. IP and ATM have the potential to reach the far corners of the Internet and have the potential to be pervasive as the need may be. But is our ultimate objective – just network reach?

Since the IP evolution, the processor speeds have increased 30 folds and the number of host on the Internet has increased exponentially by the year. Further the applications have moved more and more from being just data centric to multi-media centric with stringent QoS requirements. IP was engineered to cater to the needs of research community and has evolved to carry the load that has been placed on it over the years. But it's time we re-engineered our networks to cater to the growing requirements of the multi-service traffic.

IP is connectionless, unreliable, best-effort service that is flexible. On the other hand ATM is connection oriented, reliable, trusted service that is time sensitive. But we need networks that are time sensitive, reliable, flexible, scalable and service independent in order to accommodate multi-service traffic. Our objective is service Ubiquity rather than network ubiquity. To meet these requirements IP networks need to be,

- time-sensitive
- reliable in their delivery mechanism
- sensitive to multiservice traffic
- Able to differentiate based on traffic flow and QoS requirements
- Efficient in forwarding/routing traffic flows
- Stable, scalable and service independent
- Manageable and auto-configurable

We need an underlay network/technology or an integrated service network, which can address these issues and give these much desired features in order for IP networks to be ubiquitous in a service perspective. ATM meets this bill and has the potential to resolve many of these and provide answers to more. IP and ATM are two protocol of much potential. Though they have been viewed as competitors, their complementary strengths from an alliance that combines the best aspect of both technologies. In light of the impending multi-service delivery requirements, these convergence of two worlds, is inevitable, coupled with the dramatic growth we are experiencing confronts us with a number of challenges.

We need to ensure that the networks we are building:

- ↳ *Can deliver to our customers the services they require and demand.*
- ↳ *Can provide semantic and time transparencies required by the applications.*
- ↳ *Be flexible, scalable and service independent*
- ↳ *Make best use of the resources (bandwidth, people, and capital) that we employ in their development.*
- ↳ *Are capable of meeting the demand for growth in subscribers and bandwidth.*
- ↳ *Are able to support current and future applications, from telephony to multimedia.*
- ↳ *Enable us to deliver the breadth of services at affordable costs.*
- ↳ *Have the reach to meet the needs of a global community into the new millennium*

But these benefits do not come without challenges and they are a couple of them are recaptured herein for clarity.

- ↳ Connection-oriented Vs. Connectionless
- ↳ QoS-aware(trusted) Vs. Best Effort
- ↳ Other challenges
 - Cell switching & connection establishment and flow lengths he ability to scale
 - flow identification & service guarantee
 - QoS based routing
 - Multi protocol encapsulation and multiplexing
 - routing and address resolution

- signaling & route distribution

6.3 What must we Achieve?

Let us look at this in a little more detail. We will take each of the points from the above, and consider the implications on the enterprise network as a whole: Today competition in the networking industry, and efficiency within our corporate networks, demand that we deliver services tailored to meet the needs of each individual user. Years ago you could build networks that offered a single type and quality of service (X.25, POTS) and expect our users to adapt to its demands. Now we must build a network that delivers the services our customers demand, when they demand them, and at a cost which meets their objectives. Achieving this is about delivering Quality of Service rather than Class of service.

Quality of Service (QoS) means reliably providing a service to a subscriber that matches their expectations. It is not about delivering a perfect service, nor is it about simple priority. The network operator must take the bandwidth, circuits, and switches that comprise the network, and harness them to meet the subscriber's goals. Quality of Service is an end-to-end issue. Quality of Service is measured by the end users of the service from their perspective, without regard to the rest of the network.

Quality of Service may mean many things. It may cover availability, or throughput, or "good-put" or end-to-end delay, or delay variation or reliability, or data loss, or a combination of these parameters. Each subscriber will have an implicit or explicit contract with the network to deliver what he or she requires. This will probably be defined in terms of Quality of Service (QoS) as defined in a **Service Level Agreement (SLA)**.

But we have seen there is a definite confusion in the usage of terms **Class of service (CoS)** in IP related terminology and the Quality of Service as used in ATM related terminology. So what are we proposing to achieve?

6.4 Class of Service or Quality of Service?

In providing Cos/Qos we need to look at the elements of QoS support in order to ensure delivery of same. The figure below depicts the elements and the interactions between them.

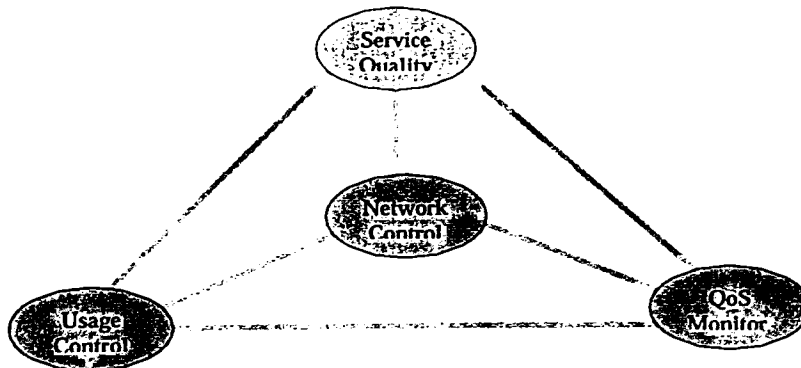


Figure 83: Elements of QoS support

Class of Service is a relative measure. A network that implements class of service can ensure that over any link in the network traffic belonging to one class of service can take priority over traffic of another class. This does not mean that it can make any absolute statement about the service quality that the end user of the network will see. Nor does it mean that any call placed across the network will see a constant quality of service; it will vary over time. Nor does it mean that we can predict with any accuracy the service that will be delivered to the subscriber.

Quality of Service is an absolute concept. A network that implements Quality of Service can predict and guarantee the service that will be provided to any subscriber. This may be a very high performance service or it may be "best-efforts". We are operating in a competitive world, this means we must make the most of the resources we have, and we must be efficient, effective, and reliable in every area of our business lives.

This is equally true of a network. Unfortunately, this is less easy than it seems. If we run a network at a small fraction of its capacity, service quality is rarely an issue. There is enough resource to meet any demand that the customers place upon it. Traffic will move swiftly and reliably across the network, and customers will receive a service that is stable and reliable. This what is being attempted in most of the high speed networks at their initial phase of implementation and it holds true to networks such as **Very high-speed Bandwidth Network Services (vBNS- IP over ATM)¹⁴⁸, Abilene (Packet over Sonet)¹⁴⁹** as well as **Internet 2 (IP over ATM)¹⁵⁰**. Over provisioning capacity, bandwidth and circuits does not mean achieving the desired quality of service. It's being liberal with the available capacity since the resources are under-utilized. QoS becomes a critical issue in the event of congestion and/or faced with limited resources or whence dealing with applications (such as in Healthcare, E-commerce, etc.) requiring stringent service guarantees. This state of affairs will exist without any intervention by the networking technology. However, the downside of designing a network in this way is that much of the capacity of the network remains unused and we are not planning for contingencies in the event of congestion or QoS issues. Ultimately QoS will be the service differentiator in marketing network services. We are wasting resources and money and are not being far-reaching in our designs/goals to deliver multi-service capabilities into the next millennium. If this approach is adopted we will be faced with another fiasco like the Y2K - millennium bug.

However, if we add traffic to the network to reduce this wasted capacity, we immediately have to face a new set of issues. These are issues relating to queuing within the network. These issues translate into variable network performance and unpredictable Quality of Service.

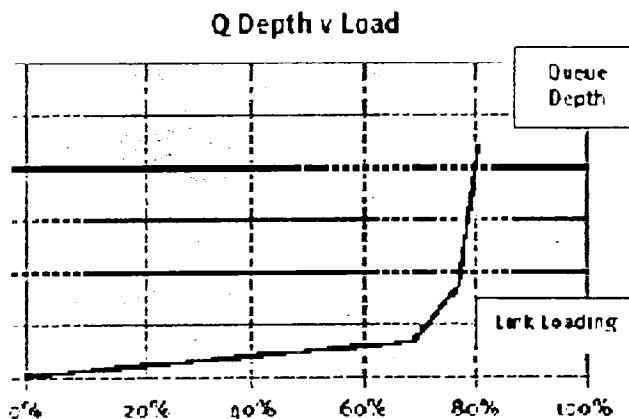


Figure 84: Q depth v Loading graph¹⁵¹

The relationship between **queue length** and **link loading** is both very fixed and very non-linear. As the load on the link increases the probable queue depth in the adjacent switch starts to rise. Once you pass 60% loading this increase is very rapid. Longer queues mean unpredictable delays, as your traffic may have to wait behind many other messages. Moreover, if the message lengths are variable, you cannot predict how long you will need to wait, even if you know the length of the

¹⁴⁸ VBNS initiative, <http://www.vbns.net>

¹⁴⁹ Abilene Project, <http://www.ucuid.edu/abilene/home.html>

¹⁵⁰ Internet 2 initiative, <http://www.internet2.edu>

¹⁵¹ VBNS initiative, <http://www.vbns.net>

queue. In IP networks, where each queued packet can be up to 1500 bytes long, this uncertainty of delay becomes very problematic.

This means that we need a technology that can control network delay if we are going to be able to deliver Quality of Service while running our networks at more than 50% of their possible capacity. The rate of growth of the Internet network is immense. In recent years, compound annual growth rates in excess of 100% have seemed commonplace. Doubling of deployed bandwidth every three to four months has been typical. At the same time the number of subscribers and the number of IP based intranets has been growing at a similar rate. A networking strategy that will operate for 10 users in a department will struggle when faced with 1,000 users across an enterprise. Similarly, a public network that can support a few hundred subscribers will not typically scale to support millions. This is a key issue which has to be addressed at the phase of design. Scalability touches many aspects of network design. It includes addressing space, routing information, privacy and closed user groups and less obvious issues such as Convergence Time.

Convergence Time is the time taken for a network to reconfigure itself to take account of a change in its structure or topology. This is bounded by the time it takes for all elements in the network (who are affected by the change) to become aware of the change and to react to the change. It is not uncommon for this to take several minutes in routed networks using protocols such as OSPF. In a network where Quality of Service is important, this becomes harder still. If an element in the network fails, much of the traffic passing through the element must be re-routed through other paths. To continue to deliver QoS new paths must be chosen that have sufficient capacity to accept these re-routed calls.

More over, calls that were not directly impacted by the failure must not be disadvantaged by the sudden imposition of re-routed traffic on to the routes they are using. This complexity clearly grows with network size. Many of these issues are exponential in their impact, so large networks cannot employ the same strategies as smaller networks. These problems of scale can only be addressed by technologies that we redesigned for the purpose - such as ATM. Today networks must support many services to meet the needs of their subscribers. We must be able to ensure that any call in the network receives the correct level of service for its content. The table shows how the various aspects of service need to be adjusted to suit each type of traffic.

Traffic demand on Networks	Voice	File Transfer	Transaction processing	Video Conferencing	Video Broadcasting
Average bandwidth demand	Very low	High	Low	Low /High	Very high
Peak Bandwidth	Low	High	High	High	Ver high
Delay	Very low	High	Low	Low	Very High
Delay variation	Very low	High	High	Very low	Low

Table 7: Aspects of service and traffic types

In an IP environment the browser model makes this tougher still. A user working with a browser may select sites with differing types of content in a single session. Therefore, the data-transfer needs of that individual user will change during the life of his connection. The network must be able to monitor these changing requirements and select the correct Quality of Service as needed. These continuously changing traffic patterns have come to be known as "flows". Each flow contains one set of logically connected data, flowing to and from the same source and requiring the same type of Quality of Service.

Flows can be viewed as the connectionless network's version of the calls we see in a connection-oriented network. Flow management is an important part of managing a high-performance IP network. This diversity of customer, and service demands must be supported in a competitive environment. Our networks must be able to maximize the use of deployed bandwidth to deliver

the maximum service at the minimum cost. This again requires that we manage Quality of Service while operating networks at, or near, their capacity limit.

At the same time, we must ensure that we develop networks that are manageable from a human perspective. Skilled management, design, and operations staff are an expensive and bounded resource. The **Operation, Administration and Maintenance (OA&M)** support systems that we deploy must deliver maximum performance and reliability without the need to increase operational costs. Again, ATM with its predictable performance, and its built OA&M functionality, was engineered to meet this challenge. ATM's traffic planning, call admission control, policing, and shaping capabilities also help to enable the automation of network operation, and to control manpower costs. We are no longer operating in local or national environments; communications, customers, and suppliers are global. Any network we design must be able to inter-operate with other national and international networks to allow the construction of a global network infrastructure to support our business goals. In a broader sense we need to cater for,

- ↳ Services with different QoS requirements - service definitions
- ↳ Means for users to communicate what they need - signaling or admission control
- ↳ Means for providers to ensure usage - policing/shaping
- ↳ Means for providers to find routes - QoS based routing
- ↳ QoS based forwarding - buffer allocation, drop policy, queuing discipline, service policy and traffic management

This demands that standards are defined and compliance is achieved, to ensure reliable operation. ATM evolved from the work of the standards bodies. It is a global standard, designed to work with the networks of today and to develop the networks of tomorrow, based on QoS guarantees. The options in reengineering IP should consider these factors in maintaining service ubiquity. Having come to certain conclusions the necessity for IP and ATM integration, we revisit Part III of our research findings in order to analyze the models and architectures we reviewed more objectively.

6. 5 Options for Multi-service delivery in to the next Millennium

Given this set of requirements and the inevitability of both ATM and IP in our world, what types of network architecture are needed? There are in reality three options open:

- ↳ A pure IP network,
- ↳ A pure ATM network
- ↳ **An IP network with an ATM core – IP/ATM integration**

In reality, the first two options discount themselves as they deny the inherently multi-service nature of our environment. We see that most groups including IETF, ATM Forum and other researchers have a general consensus on this and have already recognized this reality, and have developed adaptation and/or internetworking mechanisms to allow ATM vendors to provide non-ATM access with a goal of multi-service provisioning. To build an ATM-only network without these adaptation mechanisms would be a failure to recognize reality.

Neither will pure IP networks work. IP cannot deliver guaranteed Quality of Service, only "best-efforts" class of service. This would not meet the needs of many categories of mission-critical application or traffic such as toll-quality voice. Therefore, in reality, only one option exists. That option is to harness the sheer power of an ATM core for its performance and Quality of Service. Then surround this ATM core by multiple adaptation layers - each crafted to meet the needs of its service. One of those services will be IP. Therefore the next step is to look at the models which we analyzed in detail in the earlier chapters with view of identifying possible models, architectures and techniques which could deliver these requirements.

The table below recaptures the features of IP and ATM.

	IP	ATM
Key Driver	connectionless	Time sensitive
Key constraint	Time insensitive	Connection oriented
Service Delivery	Best effort	Trusted
OA&M	Manual	Built-in features
Processing	Packet by packet	Cell by cell based on per flow basis
Networking Technology	Assumes heterogeneity	Assumes Homogeneity

Table 8: IP Vs ATM features

We see from the table by integrating IP and ATM we have the flexibility to offer flexible, best-effort delivery on one end of the continuum and time sensitive, trusted delivery on the other end of it. With the multitude of option that we looked at in researching possible methods & techniques for integration and in keeping with our objective of providing a QoS capable network we can map our finding in terms of QoS to a range which is acceptable for multi-service delivery. This is shown in the figure below.

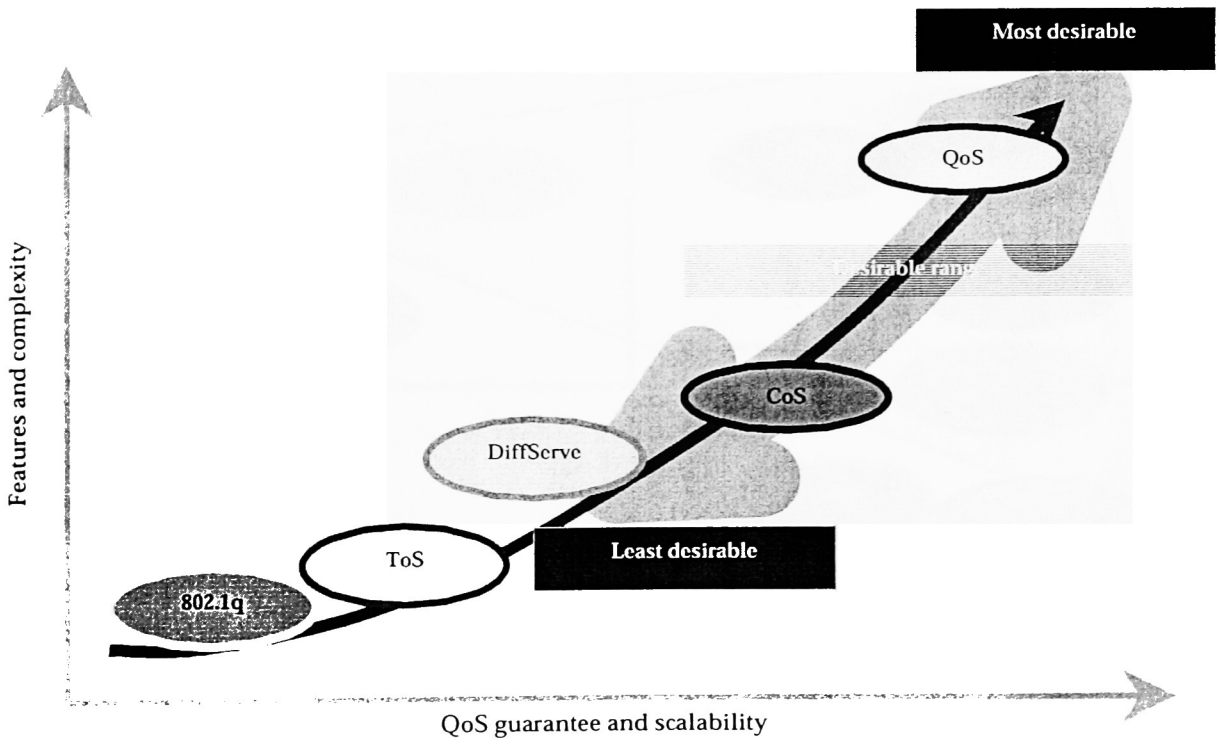


Figure 85: Desirable spectrum of Quality for multi-service delivery

This illustrates our stand on quality in meeting the necessary service objective of multi-service delivery. But, in realizing this objective we also need to seamlessly integrate IP and ATM to be able to map function and features of one to the another without compromising quality. In order to preserve the openness of integrating multitude of technology options we need to look at the generality at higher and lower layers and this is analyzed in the table below.

	Generality at higher layers	Generality at lower layers
LANE	Yes, 802 services	No, over ATM only
CLIP	No, IP only	No, over ATM only
MARS	Yes, focus on IP	Yes, focus on ATM
NHRP	Yes, focus on IP	Yes, NBMA networks
MPOA	Yes, internetwork layer	No, Focus on ATM (LANE)
MPLS	Yes, focus on IP	Yes

Table 9: Models and generality

This shows us that in order to preserve the network and service ubiquity we need focus ourselves on the inter-subnet models and namely MPOA and MPLS with the associated models necessary for their proper functioning. Based on our analysis and study findings we could classify the models and techniques which are used for various levels of integration between IP and ATM as connection - oriented, connectionless, layer 2 centric and layer 3 centric. Based this categorization they are depicted on the diagram below with the two axes showing interoperability and manageability.

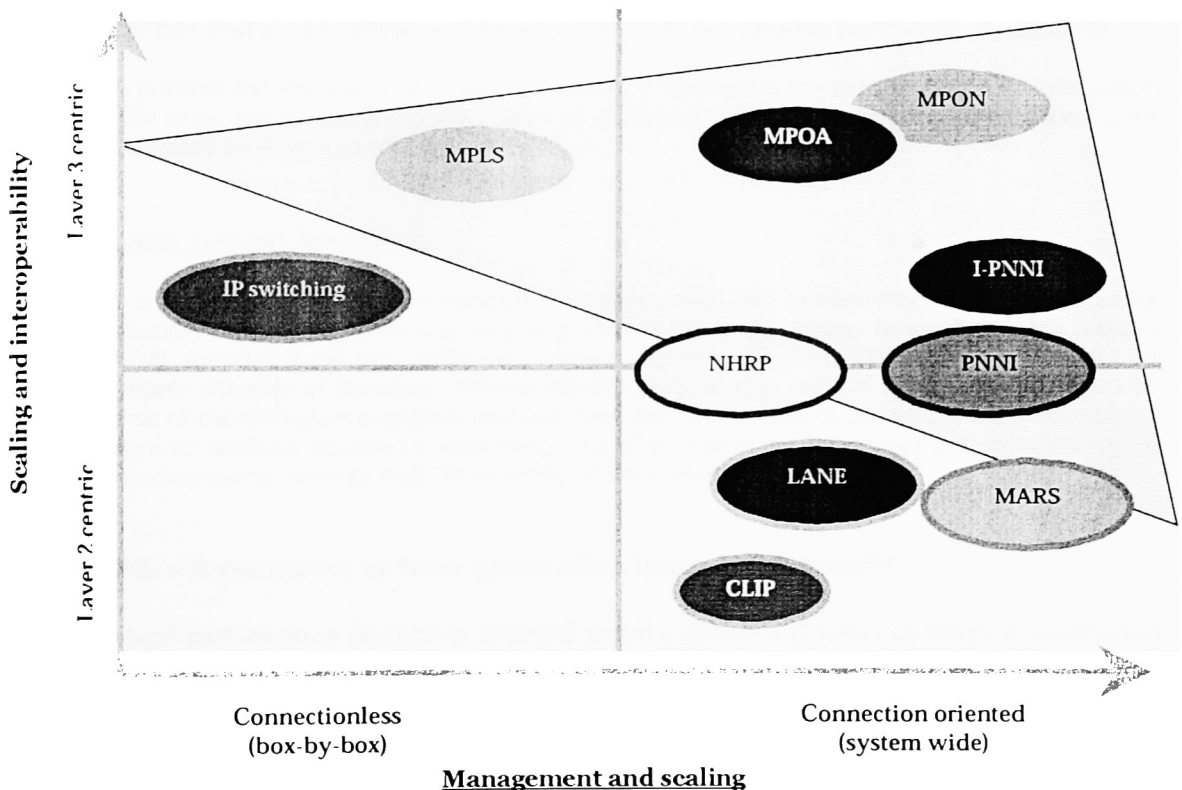


Figure 86: Delta of best fit for IP/ATM integration

From the above figure the delta of best fit highlights the models/architectures that can be used to achieve our said objectives. Namely, MPOA and MPLS. These are two models that would

revolutionize the next wave of multi-service networks. The focus should be on these models to incorporate the necessary parameters to facilitate seamless integration between the two. In retrospect we see that MPOA and MPLS are more similar than different in their approaches.

	MPOA	MPLS
Standard based:	ATM Forum (completed)	IETF (still in progress)
IPv6/ATM:	Yes	Yes
QoS/NetOps:	Few	Many
Scalability:	Very Large	large
Auto-config:	Yes	??
QoS/Multimedia:	Full QoS Based routing	Priority queues.??
Scope:	LAN/WAN	LAN/WAN

Table 10: MPOA vs MPLS

We see MPOA and MPLS are both standard based approaches, but MPOA is more mature in terms of QoS delivery and features and in it's capability to auto- configure itself. This is an important feature in maintaining stability as well as to scale in large networks. MPLS will help the integration since it can be streamlined to meet the service requirements and with that of the ATM. The need is to map RSVP signaling to that of PNNI and IP CoS to that of the ATM. With the Ipv6 Adopting a priority scheme and flow labels in it's headers and routing protocols being engineered to be QoS sensitive, the integration is one of inevitable and a necessary step towards bringing a true multi-service platform that could deliver reliable, scalable and manageable bandwidth on demand.

So having justified the imminent necessity to adopt an approach towards IP and ATM integration with an ATM core, we review some of today's multi-service networks in order to identify the trends in delivering multi-services into the next millennium.

6.6 A Case based Reviews

There are a number of high speed network initiatives which are currently in their early stages of implementation. Some of them being **very high-performance Backbone Network Service (vBNS – IP over ATM)**, **Internet 2 (IP over ATM)** and other associated projects like **Abilene (Packet over Sonet)**, **Qbone (Quality of Service)**, **Mbone (Multicast services)**, etc. It is our intention here to review some of these implementations and also look into the directions of carriers in general with a slight detour to address couple of interesting trends in multi-service network implementations in campus environments, namely that of University of Rochester.

6.6.1 vBNS – A precursor to Next generation Internet initiative¹⁵²

The **very-high-performance Backbone Network Service (vBNS)** is a National Science Foundation (NSF) sponsored high-performance network service implemented by MCI. The vBNS is an IP over ATM backbone network with IP backbone transport as its basic service. The service is provided in two distinct logical views. The first one is via a mesh-like network, constructed using point-to-point PVCs. All customers use this point-to-point mesh for IP transport over the vBNS backbone. Within the mesh OSPF is used for the internal routing. Between a customer network and the vBNS, BGP is used for the peering. The second view for IP connectivity over the vBNS is an ATM Logical IP Subnetwork (LIS). This LIS connectivity is provided to all vBNS infrastructure routers and hosts, and also to those customers who have direct ATM connections. Since no routing protocol is run on the LIS, transit

¹⁵² vBNS initiative, <http://www.vbns.net>

traffic does not traverse this US normally. Both the mesh and the US provide best effort service. As a design objective, this structure for the best effort service access is to be preserved when other services are added.

IP over ATM

Switching IP traffic over the vBNS' dedicated PVPs is accomplished by using two meshes of PVCs. One mesh of PVCs support "point to point" links between IP routers over which IP routing protocols are run. And a second mesh supports a **Logical IP Subnet (LIS)** as well as non-IP ATM traffic. The "point to point" PVCs provide a full mesh of interconnections between the vBNS' ATM connected routers (this includes almost all the routers on the network). Several routers are connected with a second "backup" PVC. The routers utilize this mesh of PVCs as if it were a mesh of point-to-point circuits. It is a flat architecture and each router is only one hop away from any other router. OSPF is used as an **Interior Gateway Protocol (IGP)** which enables the routers to share information about link costs and outages. Unless there is an outage, the lowest cost PVC joining any two routers is used. In the case of an outage a "backup" PVC is used if available; otherwise the information provided by OSPF is used to choose an alternate route. At the IP layer this alternate route will appear to be two, or more, hops long.

The LIS PVCs also provide a full mesh of interconnections between the vBNS' ATM connected routers. The mesh's end points also include additional ATM ports at the SCCs. Devices attached to these ATM ports can use IP to communicate with other members of the LIS or they can use the PVCs to exchange non IP traffic (e.g. video over ATM) with other ATM connected devices. Scheduled equipment upgrades will make it possible to replace the LIS PVC mesh with Switched Virtual Circuits (SVCs) in early 1997. Using SVCs instead of a mesh of PVCs will reduce network complexity and increase network robustness.

Network architecture

The vBNS' architecture is best described in terms of a layered approach. The IP and ATM layers of the network are built upon over 25 000 km of Sonet backbone. The links to connected R&E institutions and other networks, along with a routing policy to control how routes are distributed, complete the big picture.

Each vBNS POP is equipped with a standard suite of equipment, which provides access to the network through user-network interface (UNI) ports on a Fore ASX-1000 ATM switch. These ports are available to serve direct ATM connections via leased lines or ATM connections via a commercial ATM service provider. Frame-based connections to the Cisco 7507 router are also available as are ports which support Packet-over-Sonet. In addition, the supercomputer center POPs also have Ascend GRF 400 routers, which offer supercomputers high-speed network accesses by way of a **high-performance parallel interface (HIPPI)**.

The ATM Layer

The vBNS network is implemented on top of MCI's ATM network, which offers ATM service on a commercial basis. The vBNS was the first "customer" on MCI's ATM network, is the only customer with an OC-12 access rate, and will continue to be the first recipient of advanced capabilities.

MCI's ATM network transports vBNS traffic over a set of **Permanent Virtual Paths (PVP)**, each of which carries a bundle of permanent virtual circuits. The PVPs traverse the shared backbone, but do not merge at the commercial backbone switches. They constitute a full mesh between vBNS ATM switches-that is, they connect every node directly to every other node. The topology of permanent virtual paths (PVPs) in the vBNS backbone is a full mesh - that is, every node is connected directly to every other node. A mesh of point-to-point **Permanent Virtual Circuits (PVC)** which connect IP routers runs over the PVP mesh. ATM traffic is VP-switched through the ATM network and vBNS switches and VC-switched at the connection between the vBNS switches and the IP routers.

The vBNS' PVPs are configured to deliver **Unspecified Bit Rate (UBR)** service. This service offers no quality-of-service guarantees, but it turns out to be extremely well suited to the bursty traffic carried over the vBNS-thanks to the very high capacity of the vBNS network and constant monitoring of traffic loads and network performance. UBR can almost always provide the desired bursting capacity when it is needed without any need to reserve it.

ATM SVC-based logical IP subnet

On the vBNS, ATM-attached sites have the opportunity to participate in a switched virtual circuit-based logical IP subnet. The configuration allows for higher speed connections in the case of OC12-connected hosts, because an IP host on the logical IP subnet (LIS) can signal for an ATM connection to another IP host on the US, bypassing completely the vBNS backbone IPv4 routers. ATM end stations on this LIS are numbered from an IP network that is a subset of the block of addresses assigned to the vBNS backbone. Since this network address block is announced to vBNS sites, US attached hosts may communicate with off-LIS hosts via the vBNS backbone routers, all of which participate in the US.

The vBNS ATM switch functions as the ATM **Address Resolution Protocol (ARP)** server for the LIS, resolving IP addresses into ATM addresses. SVC call setup messages are routed across the vBNS ATM backbone using Fore's proprietary Forethought-PNNI routing protocol, a precursor to the ATM Forum's PNNI 1.0 protocol (where PNNI stands for private network-to-network interface).

The setup messages are moved from the vBNS backbone to a campus border switch by **User-Network Interfaces (UNIs)** at the boundary of the vBNS-connected institution together with static routes in both directions. In the future, the vBNS ATM switches will use ATM Forum PNNI 1.0 as their ATM routing protocol, initially with static routes and UNI-UNI interfaces at the backbone-connected institution boundary. The goal is to connect **Network-to-Network Interfaces (NNIs)**, with routing information being exchanged via PNNI across the vBNS-connected institution boundary.

The ATM SVC LIS service is offered to all ATM-attached sites that use an ATM switch (some sites use an ATM service to connect to the vBNS, but terminate that service in an IP router). In addition, all of the ATM-attached vBNS backbone equipment (Cisco and Ascend routers, Fore switches, and Digital and Sun hosts) participate in the US. The ATM addresses currently in use on the vBNS US are the international code designator ICD format prefixes that are assigned by the switch vendor. A data country code DCC format address block has been assigned to the vBNS; an ATM address plan is under development to use this contiguous block. Such a plan would reduce the number of addresses that need to be exchanged across the vBNS-connected institution boundary from more than 30 to just one.

IP layer

The vBNS network IP layer infrastructure is implemented using a set of IP routers running three protocols-the **Border Gateway Protocol (BGP)**, the **internal BGP (iBGP)**, and the **Open Shortest Path First (OSPF)** protocol-over a full mesh of ATM permanent virtual circuits. BGP is used as an external routing protocol. The vBNS currently has BGP peering sessions up and running between the vBNS routers and their counterpart routers/route servers.

The core vBNS services

The vBNS currently provides a set of core services to the community of directly attached R&E institutions and SCCs. One is a high-speed best-effort IPv4 datagram delivery service. The others are an IPv4 multicast service, an ATM switched virtual circuit logical IP subnet service, and ATM permanent virtual circuits across the vBNS backbone as needed. Among services under development, are a reserved-bandwidth service and a high-speed IPv6 datagram delivery service.

Best-effort IPv4 service: The most fundamental service of the vBNS is the high-speed best-effort delivery of IPv4 datagrams across an uncongested backbone of continental extent. Because of

the logical design of the vBNS backbone, packets can be delivered between vBNS-attached sites with only two backbone router hops, regardless of how far they are apart. End-to-end packet paths between hosts would include at least two more router hops: a campus border router at each communicating site. Packet paths between vBNS-attached sites are notable for having relatively few routers and rather fast links (at least 44.736 Mb/s for access, and 622.0B Mb/s for the backbone).

In the event of a backbone trunk failure, however, the model of two backbone router hops between vBNS-attached sites is violated. In such cases, alternative IP paths are dynamically recomputed. These alternative paths will include at least one additional backbone router hop. Using as few backbone router hops as possible helps to minimize the number of times an IP datagram has to undergo the processes of segmentation and reassembly as it enters and exits the ATM portion of the network. The result is a reduction in end-to-end switching latency across the vBNS backbone.

IP multicast service : Multicast service at the IP layer enables a source to use a group address to send an IP datagram to a set of receiving end-systems. The service, provided by the network, delivers the packet to all members of the group. Unlike this point-to-multipoint delivery service, the traditional unicast service model, has a single end-system send a packet to exactly one receiving end-system.

The vBNS maintains a native IP multicast service, meaning the network is capable of routing and delivering IP multicast packets without tunneling-that is, without any need for dedicated multicast routers and their attendant inefficiencies. The vBNS uses a **Protocol-Independent Multicast (PIM)** dense-mode configuration among all vBNS Cisco 7507 routers. **Distance Vector Multicast Routing Protocol (DVMRP)** unicast routing is used, allowing the vBNS to support delivery of Mbone traffic. The Mbone (multicast backbone) is an experimental IPv6 network which is built, using tunnels, over the commodity Internet.

Most directly connected institutions take advantage of the vBNS' multicast service. The vBNS also exchanges multicast routes and traffic with other R&E networks. These connections are in the form of exchanges over local media-for example, supercomputer center FDDI rings, tunnels to PIM routers or routed hosts, and PIM connections to border routers over point-to-point PVCs. Current multicast applications include communication between Web caches; videoconferencing; Mbone sessions; and updates from the PC-based traffic-measuring devices used to monitor network performance.

Although the vBNS backbone supports multicast with a PIM dense-mode configuration that works reliably, the backbone will move to a sparse-mode configuration later this year. A sparse-mode configuration is more efficient in an Internet backbone environment where group membership is not densely populated.

Now being tested is experimental code from Cisco Systems Inc., with which BGP routes may be tagged as multicast. The Mbone suffers from well-known problems, chief among them being the unscalability of a single routing domain. A multicast **Exterior Gateway Protocol (EGP)** is needed to grow Internet multicast in a way that mirrors the scalability characteristics of unicast Internet routing. The vBNS intends to deploy the multicast border gateway protocol on its production routers when it is feasible and safe to do so.

Reserved bandwidth service: The growing traffic load and the desire to effectively support current high-performance research applications are what motivated the development of the reserved bandwidth service, MCI's **Quality-of-Service (QoS)** enhancement for the vBNS. With over 100 education institutions added, the vBNS may start to experience traffic congestion. The number of bandwidth-sensitive applications, particularly between supercomputers, is not large, but each flow may have a high data volume. Such traffic characteristics-a few highly bursty and bandwidth-

sensitive application flows-is a strong argument for something like the reserved bandwidth service that can dynamically and efficiently allocate bandwidth on a per-session basis.

The reserved bandwidth service is based on allocation of bandwidth to application flows and is dynamically triggered by **resource-reservation protocol (RSVP)** signal messages. Since the service can be reliably provided only within the vBNS backbone, end-to-end QoS is not promised. It may, however, be achievable-depending on the service guarantee, if any, of the connection to the vBNS. Using per-flow mechanisms for the implementation with bandwidth committed, this service is somewhat stronger than the controlled-load service. Delay variance is bounded but not offered as a service parameter, and the service does not attempt to give the same effects as the guaranteed service.

Reserved-bandwidth service is a practical match between the needs of the users and the capabilities of the technology. Working jointly with some vendors, MCI has identified the requirements and feasibility of the major implementation mechanisms. These include token-bucket-based packet policing, netflow-based packet classifier, the mapping of RSVP to ATM signaling, **Weighted Fair Queuing (WFQ)** for packet queuing and **Weighted Random Early Discard (WRED)** for packet discard. These mechanisms are at different stages of development and some are under intensive testing by the vBNS engineers.

As a leading-edge backbone network connecting R&E institutions, the vBNS is also considered as a valuable platform for technology experimentation in a production environment. At present, guaranteeing quality of service on a backbone network is an important research focus. While providing production-level service, the vBNS must adopt an evolutionary strategy to QoS research and development.

Following the reserved bandwidth service, the plan is to offer differentiated services as a way to address the issue of full scalability facing today's Internet. An SVC is set up through the VBR VPs for each reserved bandwidth flow on a demand basis. Requests for reserved bandwidth service that cannot be supported within vBNS backbone resources will be denied. In such a case, applications can use the best-effort service, if appropriate. There are two advantages to this VP/VC strategy for reserved bandwidth traffic. One is that the ATM network configuration is simple and static. Signaling is not required of the ATM network switches and provisioning changes should be rare. Also having a single reserved bandwidth VP per physical path allows optimal use of available bandwidth resources. This allows full bandwidth to be allocated between any end-points that need it. A static VP configuration with multiple VPs per physical trunk would required pre-determined bandwidth allocation to the VPs, dividing the bandwidth in a way that would seldom correspond to application needs.

Router QoS Mechanisms

Flow Setup : Use of the reserved bandwidth service by user applications requires a per-flow path set up once an **RSVP RESV** message is received. vBNS routers are required to initiate signaling for an appropriate SVC across the vBNS to carry the flow's traffic. To do this, the **Rspec** parameters from the RSVP RESV message must be mapped to corresponding UNI signaling parameters. When the signaling is complete, the resulting VPI/VCI must be added to the state information associated with the flow, so that the data packets belonging to the flow can be routed onto the correct ATM virtual connection to cross the vBNS. The mapping between RSVP Rspec parameters and ATM signaling parameters follows current work in the IETF.

RSVP PATH messages first use the existing best effort PVC until a **VBR SVC** is set up for the traffic flow requesting the service. From then on, this SVC will be used to carry the RSVP PATH message for the same flow. **RESV** messages for a flow also uses an existing best effort PVC. If a reverse VBR SVC is set up for the flow, the RESV messages will also use this SVC. If no reverse path SVC is set up, the RESV messages will keep using the best effort PVC.

Packet Classification and Policing : The token bucket based scheme is used to implement both packet classifier and packet policing. Each reserved bandwidth service flow will be associated with a token bucket. The state for a token bucket keeps the packet classification criteria and rate information passed by RSVP messages. It also keeps track of actual packet data rate for packet policing. Rate information includes average rate, peak rate and allowed burst size. Since traffic is not shaped from the input side of a router to the output side, the buffer space required at the token bucket is not large. Classification criteria include source and destination address and other header information similar to those that are used for access control list. This allows extension to support class-based packet classification. Non-conforming packets can be either discarded or mapped to a lower precedence class at the network operator's choice. There are active and passive modes in which the token bucket mechanism can actively set the precedence bits or just passively use the precedence bits set by customers for classification.

Packet Queuing and Discard: The current design is to use WFQ on the output, with each reserved bandwidth flow in a queue receiving a weight equal to its reserved rate. To meet the loss goals of the reserved traffic, a drop policy must be used. The preferred algorithm is Weighted RED for selecting packets to discard, with each class having a separate MIN, MAX and probability weight applied to it as a drop mechanism. Using this *WRED* mechanism, we can treat non-conforming packets that belong to reserved bandwidth flows with a discard probability either higher than the best effort traffic or in between the reserved bandwidth and best effort.

Once the ATM policing function has been configured to match the router QoS thresholds, the router must ensure that packets transmitted on a SVC do not then get dropped by the ATM policing function. This may happen as a result of processing jitter in the router causing transmitted cells not to conform to their contract. This will require shaping, preferably in hardware, of the outgoing traffic. The shaping capability is in our implementation plan but may not be offered at the beginning depending on vendor's delivery.

Coordinated Admission Control for IP over ATM: Since the Reserved Bandwidth is an IP service implemented over ATM, there are two levels of admission control to coordinate. At the IP layer, RSVP can be used to configure thresholds for bandwidth reservation both on an interface basis, and on an individual flow basis. The admission control algorithm checks that an incoming flow request does not result in either threshold being violated. For a particular data flow request, it is the outbound interface that should be checked for sufficient capacity. Subinterfaces are subject to the aggregate threshold for the physical interface configuration as well as their individual thresholds.

Admission control at the ATM layer is based on a **Call Admission Control (CAC)** computation when the request for a SVC is initiated. The SVC is subject to ATM admission control. In order to avoid flows passing RSVP admission control only to be rejected at the ATM layer, coordinated configuration of the RSVP bandwidth threshold with the ATM CAC algorithm is needed.

Class of Service

End to end signaling and per flow resource allocation and scheduling, the core mechanisms to support vBNS "reserved bandwidth" service, demand high processing power and large memory space in proportion to the number of active traffic flows that require such service. It is expected, that simultaneous vBNS traffic flows requiring "reserved bandwidth" will initially be a small number compared against the total active end to end flows at any time. This number is "reasonably small" so that the mechanisms discussed in this paper can support this service with adequate performance. Although vendors have not committed to any specific size parameters, their in-depth disclosure on design and implementation gave us a degree of confidence to provide the "reserved bandwidth" on a limited basis given our current user groups.

Extending our QoS offering to include class based services, or differentiated services, is a logical step to address the scalability issue and to satisfy the diversified QoS needs by our customers. For example, the Internet2 community is looking at differentiated services as a model for their first QoS

requirement. As its initial backbone infrastructure, we are actively looking at the different implementation choices to support the premium and/or the assured class along with the reserved bandwidth service.

For these class based services, we desire to use the same implementation mechanisms used for the "reserved bandwidth". The key difference is no per flow states or queues. Instead, queues and states will be on a per class basis. Signaling to trigger the service on a dynamic basis is likely to be unnecessary. Packets of application flows with similar network resource demands will be associated with a same class identifier and given the same treatment at points of packet queuing or dropping. We are evaluating the potential problems and required extensions to these mechanisms to support both per flow and per class services in the same environment. We are also interacting with the Internet2 community on issues like bandwidth broker, policy control, etc.

IPv6 service

IP version 6 (IPv6) is the next generation Internet Protocol. It is currently undergoing standardization within the Internet Engineering Task Force (IETF) as a successor to the current version (IPv4) to address the latter's shortcomings. IPv6 promises a significantly larger address space, improved auto-configuration capabilities, expanded routing and addressing mechanisms, a simpler header format, better support for options, and improved security and privacy support.

Support for IPv6 in commercial products (routers and hosts) is gradually becoming available. Commercial IPv6 implementations are now mature enough to support an experimental IPv6 service offering on the vBNS. In its initial offering, the vBNS intends to deploy native, rather than tunneled IPv6 on the vBNS backbone. MCI is installing three IPv6-capable routers spread geographically across the backbone in the vBNS POPs in Peryman, Md., Chicago, and San Francisco. We are installing dedicated IPv6 routers, as opposed to using the existing IPv4 routers for both protocols. The IPv6 routing code is not yet production quality and might jeopardize the stability of the vBNS IPv4 service if this were not done.

The IPv6 routers will be ATM-attached at the OC-3 rate to the vBNS backbone ATM switches and fully meshed with each other using permanent virtual circuits. The routers will communicate with each other over those circuits using native IPv6 over ATM. Any vBNS ATM-connected site wishing to do likewise may configure a PVC from its campus IPv6 router to the nearest one on the vBNS.

Each vBNS IPv6 router will also serve as the end-point for multiple IPv6-in-IPv4 tunnels from vBNS-connected sites that are not capable of native IPv6 and will also terminate tunnels that traverse network access points to sites on the global 6bone. The 6bone is an experimental IPv6 network which is built, using tunnels, over the commodity Internet. The vBNS has been assigned IPv6 address space for use on the experimental 6bone and will delegate addresses from this space to vBNS-connected sites wishing to take advantage of the IPv6 service.

6.6.2 Internet 2 – the Next generation internet initiative¹⁵³

Internet 2 seeks to enable emerging set of advanced network-based applications within and across many colleges and universities. It will do so by working with the information technology industry to develop common standards and support services for new classes of applications and to ensure the availability of the advanced communications services required. Many of the communications service technologies required are still in the pre-competitive stages of development. They should benefit greatly from a broadly based test ed encompassing the research and education community.

Fundamental to the Internet 2 infrastructure design is maintenance of a "common bearer service" for communication among network applications. The "bearer service" is the basic information

¹⁵³ *Internet 2 initiative*. <http://www.internet2.edu>

transport interface for wide area communications, analogous to layer 3 in the ISO network model. One of the greatest strengths of the existing Internet is the ability of any node to communicate with any other node in a compatible transport format. We must preserve this strength in Internet 2. To the extent possible, the I2 bearer service must be backward compatible with the existing Internet. That existing infrastructure will continue to be the access path to all non-participants in Internet 2, as well as to university members served by local **Internet Service Providers (ISPs)**.

The common bearer service today is the **Internet Protocol (IP)** version 4. I2 will deploy IP version 6 (IPv6) as early as possible. All implementations must be backward compatible with IPv4. In addition to IPv6, I2 must enable applications to specify a network **"quality of service" (QoS)** along dimensions including transmission speed, bounded delay and delay variance, throughput, and schedule. Meeting these requirements as soon as possible is the design challenge we have undertaken. Fortunately technologies intended to provide these capabilities have been under development for several years, and initial production versions are almost ready for serious testing in the field. Beyond the technologies themselves, Internet 2 participants will require both the most cost-effective services and services with predictable costs. Advanced services delivery and support systems will not be inexpensive. As these services migrate into the commodity market, we want to ensure that there is a sound economic model for their use. The engineering design for the I2 Project infrastructure must enable end-user management of costs as well as services.

Engineering Overview

A number of technical and practical considerations underlie the overall architecture for Internet 2 infrastructure. One of these is the need to minimize overall costs to the participating campuses by providing for access to both the commodity Internet and advanced services through the same high-capacity local connection circuit. In addition, other campus programs and projects can be accommodated by means of a flexible regional interconnection architecture. For example, a metropolitan area network service might offer high capacity Internet service to student and faculty residences, and the campus will want a high capacity interconnection with that service.

For wide-area advanced services, a single interconnect service among gigapops (probably the NSF-sponsored vBNS) will suffice at first. A number of service providers will be able to offer attractive advanced services as technologies migrate into the private sector. The design of Internet 2 must optimize the campus's ability to acquire services from the widest variety of service providers. The overall architecture of Internet 2 is shown in Figure 1. The key new element in this architecture is the gigapop (for "gigabit capacity point of presence") a high-capacity, state-of-the-art interconnection point where I2 participants may exchange advanced services traffic with other I2 participants. Campuses in a geographic region will join together to acquire a variety of Internet services at a regional "gigapop".

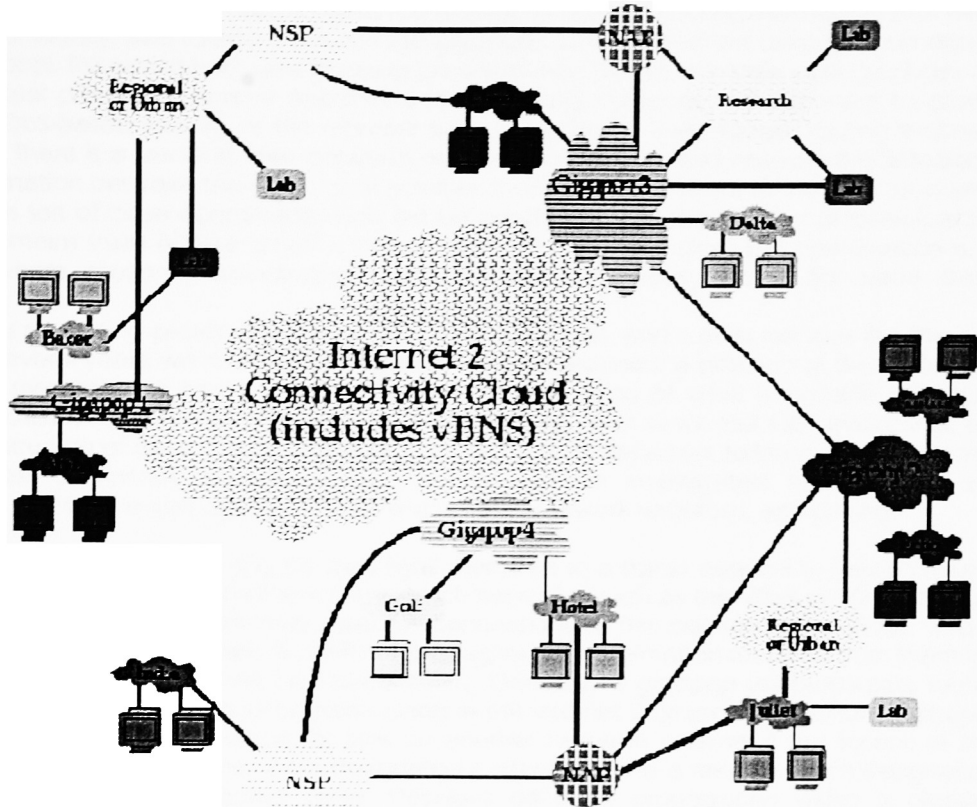


Figure 87: Internet 2 network layout

The research questions related to the network itself (as opposed to specific application areas) include:

Network service requirements. In particular, what network QoS levels are really needed for advanced real-time multimedia applications?

Protocols for delivering different QoS levels. In particular, how much state information must be maintained in routers and/or switches to deliver high-quality differentiated service? Is it possible to achieve the levels of QoS support we want without using link-level circuit switching?

Management. What are the administrative implications of a multi-QoS network, especially from network-management and cost-allocation perspectives?

Cost Recovery. How can authorization and attribution for QoS requests be handled efficiently in a "stateless" communications service?

Routing and Quality-of-Service Protocols

In Internet 2, internet layer routing will be managed for both IPv4 and IPv6. Consortia of universities will construct the Internet 2 gigapops, and the consortia will have their own infrastructure to interconnect their gigapop(s) and their members. In many cases consortium gigapops will provide consortium-specific gigapop services to members of their consortia before they are connected to other gigapops. In particular the consortia may have already established routing policies for traffic within and between themselves, as well as between themselves and other network services, before they connect to each other. Unlike other networks where a backbone and all backbone switches are owned and managed together, Internet 2 will be built by linking units under coordinated but separate administrations.

Past experience has shown that it is all too easy for one entity providing specialized services to its members to accidentally leak inappropriate routes to other entities. Routing information thus needs to be filtered, ideally, and routing between the gigapops to be performed using an inter-domain routing protocol. This would both give gigapop consortia more freedom in their routing policies and provide mutual protection against accidental route leaking. However, we also want to provide support for QoS-based routing. At the moment support for QoS in inter-domain routing is close to nonexistent. There is a tradeoff here between network functionality and network predictability. If close coordination between the gigapops is possible, then we can attempt to use an intra-domain protocol. This sort of close coordination only will be possible at all if the number of participants in Internet 2 remains small (where "small" is measured in terms of how close the coordination is, for example, where routing administrators exchange e-mail regularly on a first-name basis).

Since there is no routing protocol which satisfies both our needs, and it does not look like there will be one for several years, we need to find ways to engineer around the problem at the outset, and to promote research into routing for the longer term. Discussion of what is possible for various protocol families follows. First we discuss the basics of routing, with or without QoS awareness, and then add discussion of the enhanced possibilities, which will be important to Internet 2. In all cases, the usefulness of various aspects of QoS routing can be investigated hand-in-hand with explorations of resource management, assigning value to network resources, and pricing.

Routing for IPv4 : Internet 2 will only be used by I2 members as a transit network to reach (1) other Internet 2 participants, or (2) other special research networks (such as the vBNS or ESnet) through designated paths. A consortium may establish connections to the commercial Internet, and to other services, for its own purposes, but will not propagate any information received from them into Internet 2. Routing information will be filtered strictly. Generally a gigapop will propagate routing information only for sites known to be participants in the Internet 2 project. In addition a gigapop may propagate routing information for sites on another research network if the source of that routing information within Internet 2 is a designated path to that site a network. Such designations will be decided by the Collective Entity. Decisions on route propagation within a gigapop consortium are purely that consortium's business. We recommend that a consortium propagate to its members information about reachability to other Internet 2 participants via Internet 2, but only if the members can be relied on not to leak this information outside of the consortium. That is, a consortium, including all of its member universities, must not inadvertently tell sources outside of Internet 2 to reach sites elsewhere on Internet 2 through it, unless it is a designated path for interconnecting with those sources. I2 will not provide transit routing among other backbone networks.

QoS-capable routing protocols for IPv4 are still scarce, if they exist at all. There is no support for QoS in either *BGP* or *IDRP*. *QoS-capable OSPF* is still being worked out. *Integrated PNNI* is a possibility. The intent of I-PNNI is to use the routing protocol developed for PNNI for both ATM and IP. PNNI has drawn upon the knowledge gained in using its predecessors and has advantages as a routing protocol design. I-PNNI is intended to offer QoS-based routing to IP as well as ATM. It is not an inter-domain protocol (yet - the possibility is being looked into), but has abstraction and aggregation of network elements. QoS-capable routing for IPv4 will be part of Internet 2's development agenda. This does not mean that the Internet 2 community will necessarily do the work, but rather that the Internet 2 community will give priority to promoting the development of QoS-capable routing through various means

Routing for IPv6: *IPv6 routing (nimrod)* is still under development. I-PNNI is intended to have support for IPv6. IDRP has support for IPv6 in theory, but IDRP implementations are not considered strategic and will need work. IDRP has limited QoS support. At this time it looks like IDRP will be superseded by a new project, *BGP4++*, *OSPF* and *RIP for IPv6* have been tentatively specified, but QoS-capable OSPF is still being developed. Those sites wishing to experiment with IPv6 may use either RIPv6 or static routes until appropriate routing protocols have been implemented. This is feasible because we can expect few sites to be working with IPv6 in the near future, and tight coordination between them will be possible. The static routes will need to be implemented without regard for any

hierarchy of relationships in the Internet 2 project. QoS routing for IPv6 will be part of Internet 2's development agenda. IPv6 addresses can be allocated by the Collective Entity

Route information at the ATM layer : ATM route information will be necessary because many of the QoS-related network functions with which we wish to experiment involve dynamic allocation of resources at the ATM layer. ATM can be expected to use permanent virtual connections for some functions (for example, carrying IP packets which do not require special virtual connections) and switched virtual connections for others. Where possible, switched virtual connections are always preferable to permanent virtual connections, to minimize complexity of configuration and to support rerouting in case of network problems.

Intra-domain routing has been developed for ATM (PNNI). At this time there are no policy filters available in any commercial ATM product. However, ATM routing does have effective QoS support. Until more sophisticated routing is available, ATM routing will have no filtering. This is feasible because we can expect few sites to be working with ATM in the near future, and tight coordination between them will be possible. It is also feasible with less coordination than IP routing because virtual connection setup can be monitored and managed. ATM addresses can be allocated by the Collective Entity.

Routing and Internet 2 Hierarchy : The above sections only deal with the topmost levels of Internet 2, and treat the lower layers as opaque, although it makes recommendations for them. The general rules laid out above for interactions between gigapops, as well as between gigapops and external networks, apply at every level within Internet 2.

Quality of Service Dimensions :Based on discussions thus far, and likely to change as concrete applications take form, we expect I2 to permit requests for at least five dimensions of Quality of Service (QoS):

- ↳ Transmission speed. The minimum effective data rate to be provided, plus perhaps a target average and a tolerable maximum limit. Thus, for example, a user might request a connection whose data rate never falls below 50Mbps, and agrees not to expect transmission faster than 100Mbps.
- ↳ Bounded delay and delay variance. Especially for video and other signals that carry real-time information, the maximum effective interruption allowed. A user might specify that there be no gap between packets long enough to interrupt or freeze live video.
- ↳ Throughput. The amount of data to be transmitted in a specified time period. A user might specify that a terabyte of data be moved within ten minutes.
- ↳ Schedule. The starting and ending times for the requested service. A user might specify that the requested connectivity be available at some exact time in the future for some specified period (which of course would arise from the other QoS specifications).
- ↳ Loss rate. The maximum packet loss rate to be expected within a specified time interval.

The more extreme the QoS requested, the more demanding it is of network resources, and the more disruptive a request is to other users. These costs of providing service must be clear enough to users that they are encouraged not to request any higher level of service than they need. Whether complete information and communal spirit are sufficient remains to be seen. We expect that universities will prefer predictable costs at an institutional level, but may offer different allocation schemes to users on campus. Indeed, part of the research agenda for I2 is to identify the economic and public policy issues that reflect both marketplace and social forces. It is likely that within campuses, several allocation schemes may be employed, including those that foster rational consumption and some that address other goals. We expect that I2 traffic will involve IP routing over ATM switching over **Synchronous Optical Network (SONET)** transmission, but as outlined under "connectivity" it may be too early to resolve this. We expect that RSVP and related protocols will communicate QoS requests, and that management of links up and down the network hierarchy will serve those requests.

The more extreme the QoS requested, the more demanding it is of network resources, and the more disruptive a request is to other users. These costs of providing service must be clear enough to

users that they are encouraged not to request any higher level of service than they need. Whether complete information and communal spirit are sufficient remains to be seen. We expect that universities will prefer predictable costs at an institutional level, but may offer different allocation schemes to users on campus. Indeed, part of the research agenda for I2 is to identify the economic and public policy issues that reflect both marketplace and social forces. It is likely that within campuses, several allocation schemes may be employed, including those that foster rational consumption and some that address other goals.

We expect that I2 traffic will involve IP routing over ATM switching over SONET transmission, but as outlined under "connectivity" it may be too early to resolve this. We expect that RSVP and related protocols will communicate QoS requests, and that management of links up and down the network hierarchy will serve those requests.

6.6.3 Sprint's ION¹⁵⁴ – Future directions of multiservice providers

Sprint has been privately testing the revolutionary **Integrated On-Demand Network (ION)** capability with both businesses and consumers for the past year. An initial roll out to large businesses will begin later this year. The service will be generally available to businesses in mid-1999, with consumer availability late in 1999.

Sprint's Integrated On-Demand Network also creates a new cost standard for the telecommunications industry. By utilizing cell-based network technology, the network cost to deliver a typical voice call will drop by more than 70 percent. For example, Sprint's costs to provide a full-motion video call or conference between family, friends or business associates will be less than to provide a typical domestic long distance phone call today.

"We have moved beyond the outdated cost structure of the last 100 years, we will be offering every Sprint customer their own multi-billion dollar, unlimited bandwidth network in the same monthly price range that many customers spend today for communications services." William T. Esrey, Sprint's chairman and chief executive officer.

Sprint's investment in ION provides the fabric for truly redefining local phone services. They have been able to create a network of the future that can serve as the basis for their competitive local phone strategy as well. Sprint's long distance network is already built and covers the entire United States. Its reach will be extended through **Metropolitan Broadband Networks (BMAN)** available in 36 major markets nationwide in 1998 and in a total of 60 major markets in 1999. These BMAN networks will allow Sprint ION to pass within proximity of 70 percent of large businesses without having to utilize **Digital Subscriber Line (DSL)**. For smaller business locations, telecommuters, small/home office users and consumers who may not have access to BMANs, ION supports a myriad of the emerging broadband access services, such as DSL.

"We are opening new vistas for the ways in which people communicate. If you are a Sprint customer, you will be online, all the time. You will not have to access this network of breathtaking power and speed; you will be part of it," said Esrey.

Unmatched capabilities

Sprint is able to deliver this revolutionary new capability because its network supports a seamless, integrated service to the desktop over an **Asynchronous Transfer Mode (ATM)** backbone network. This network fabric provides the speed, flexible bandwidth, scalability, service consistency, security and telephone voice quality that neither the Internet nor non-ATM-based networks can deliver.

¹⁵⁴ Sprint's ION service. <http://www.sprint.com/Stamp/press>

"Sprint's Integrated On-Demand Network gives us capabilities our competitors don't possess, because of the limitations of their network architectures, the traditional telecommunications providers are mostly consolidating and bundling different services, not integrating them. They are building separate data networks that are not integrated with their legacy voice networks. As a result, many competitors will be forced to rationalize disparate networks or risk being disadvantaged in cost and capability." Ronald T. LeMay, Sprint's President and Chief operating officer

In addition, Sprint's ION leap-frogs the bandwidth-only capabilities of DSL and cable modems. ION provides customers with robust voice, video and data services, along with the capability to customize multiple services, all combined with access to unlimited bandwidth, available on demand, all the time, whether they are across town or across the country.

Sprint's ION will integrate existing customers' networks and greatly simplify network management for Sprint's customers, which the emerging carriers' IP networks will not be able to accomplish. Unlike Sprint's ION, the emerging carriers networks cannot allow customers to "grab" bandwidth as needed. While the emerging carriers claim to be deploying networks to selected U.S. cities, Sprint's high-speed integrated network is deployed across the country and within most major cities through metropolitan broadband network rings. Several major corporations have already committed to utilizing Sprint's Integrated On-Demand network services in the months to come. Coastal States Management, Ernst & Young LLP, Hallmark, Silicon Graphics, Sysco Foods and Tandy will be initial customers.

For these businesses, and others like them, ION offers a significantly more productive and efficient communications solution than today's model. High speed, integrated communications will be available to corporate locations, branch offices, small businesses and the small office/home office worker. The result is an enhanced virtual private network that enables applications such as collaborative product development, supply-chain management, distance learning and telecommuting.

Businesses will be able to both provide communications internally and develop a totally new way to think about integrating suppliers, partners and clients into a truly integrated multimedia network. For small businesses, Sprint's next-generation network is the great equalizer - delivering the same communications power that is available to large businesses.

More than a decade ago, Sprint ushered in the era of pin-drop quality and reliability when it introduced the first nationwide, all-digital fiber optic network. Sprint was first to market with a variety of products and services, including the first public data network, the first national public frame relay service and the first nationwide ATM service offering. Additionally, Sprint deployed the first coast-to-coast SONET ring route and was the first carrier committed to deploying Dense Wave Division Multiplexing on nearly 100 percent of its fiber miles. SONET allows voice, video and data services of any bandwidth size to be transmitted to its destination with guaranteed delivery. Metropolitan Broadband Networks extend that powerful service and delivery guarantee in major markets. This same innovation and execution prowess is the foundation for ION.

Another technological advancement developed over the past four years as part of ION is the ability to carry pin-drop quality voice traffic over an ATM network and to seamlessly connect to any public switched network. This capability will be transparent to customers using the Sprint network. Network capacity is not an issue. Through deployment of Dense Wave Division Multiplexing and other fiber-optic technologies, Sprint can efficiently and quickly scale network capacity, as the marketplace demands, while simultaneously improving unit economics. In 1998, a single Sprint fiber pair will be able to simultaneously carry over 2 million calls --- the equivalent of the combined peak time voice traffic of Sprint, AT&T and MCI. Next year, that same fiber pair will be able to simultaneously carry four times the combined voice traffic of Sprint, AT&T and MCI.

"In the Year 2000, one pair of Sprint fiber will have the capacity to handle 34 million simultaneous calls, or 17 times today's combined volumes of Sprint, AT&T and MCI, without having to physically construct any new fiber. Sprint led the way over a decade ago with

crystal-clear quality and the construction of our nationwide fiber network. Since that time we have led the industry in numerous 'first to market capabilities' across the emerging and high-growth data market. Today, with its innovative ION applications, Sprint establishes its place as the pre-eminent provider of total services to our customers," Esrey

6.6.4 University of Rochester¹⁵⁵

University of Rochester is undertaking a complete integrated approach for delivering voice and data services over an extensive ATM backbone for the University and medical facilities. By leveraging today's high-speed technologies such as ATM, UofR will implement a state-of-the-art backbone to deliver transparent voice features and functionality, as well as centralized voice mail, and **Open Application Interface (OAI)** applications. Their objective of choosing an ATM technology in backbone is pre-empted by their desire to collapse their existing and future data services on to the same ATM backbone in order to optimize on the performance/cost factors providing for a scalable, future proof infrastructure to facilitate their growing multi-service requirements.

The new backbone is based on the Catalyst B540 Multiservice ATM Switch enabling non-blocking switching at speeds up to OC-4B. Building upon the Cisco **Internetwork Operating System (IOS)** software as well as supporting the latest ATM Forum specifications, Tag Switching (Multiprotocol Label Switching), and Frame Relay-to-ATM interworking, the Catalyst B540 have been engineered to deliver the most advanced services and functions for UofR.

Voice Backbone : The figure depicts the new voice backbone for the University of Rochester. The backbone provides a fully meshed dual OC-3 topology to offer high availability of voice services for UofR. Each Cisco Catalyst B540 is equipped with ATM **Circuit Emulation (CES)** modules to carry voice traffic between the PBX systems in a transparent manner, seamlessly integrating all the nodes into a single, distributed PBX.

FCCS Signaling : A separate network is used to carry FCCS signaling traffic and OAI specific information, based on the Cisco Catalyst 5505 switches running LAN emulation. The FCCS signal information will appear on a specific VLAN of its own VLAN. For improved fault tolerance, each Catalyst 5505 is dual homed; one link to the local Catalyst B540 and the second to a selected neighboring Catalyst B540.

Simple Server Redundancy Protocol (SSRP): In order to provide fault tolerant access for OAI applications across the UofR ATM backbone, redundant LANE services have been provisioned. The Simple Server Redundancy Protocol (SSRP) that provides redundancy for all LANE services. Figure below is a diagram depicting where primary and secondary services will be configured for the new network.

The following assumptions have been made for the SSRP services:

- a single **Emulated LAN (ELAN)** for all Catalyst 5505 locations,
- routing to the ELAN will be provided by an existing(s) UofR 7500 series router,
- recommend one or two (using HSRP) ELAN routing point(s) for ELAN services

There will be a Primary LECS, LES and BUS for the ELAN and a redundant Secondary LECS, LES and BUS services. All Catalyst 5505s will have one (1) LEC provisioned.

¹⁵⁵ University of Rochester, <http://www.umd.rochester.edu>

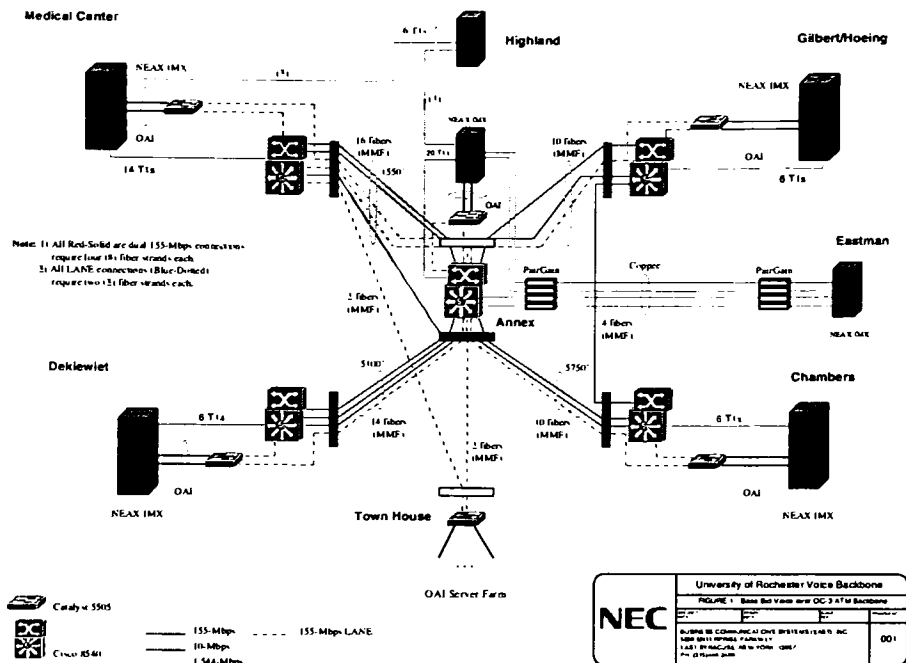


Figure 88: University of Rochester ATM backbone

ATM Circuit Emulation : The Catalyst 8540's is the core component which provide for the Soft PVC's for Circuit Emulation Services. This will provide added robustness for circuit rerouting in the event of a link failure between Catalyst 8540's. All CES services will be provisioned using an *unstructured* T1 configuration. Unstructured CES services in a Catalyst 8540 emulate point-to-point connections over T1 leased lines.

Robustness of CES services will be obtained through Soft PVC configuration. Each T1 will be defined by an active side and *passive* side interface.

VLANs and IP addressing: All the devices are mapped to a single LIS and the breakdown of the IP addressing plan is shown below. The tables represent a range of addresses that can be used for the ATM and Ethernet switching equipment. Since the ATM backbone will consist of one ELAN, one IP subnet will suffice the requirements and the subnet 192 is divided into the following ranges:

- Catalyst 8540 ATM switch range: 172.31.192.1 – 172.31.192.20
- Catalyst 5505 Ethernet switch range: 172.31.192.21 – 172.31.192.40
- Cisco 2511 Router Ethernet 0 port: 172.31.192.41
- IP address pool for Dial-in Users on Cisco 2511: 172.31.193.1 – 172.31.193.16
- NEC IMX PBX address range: 172.31.192.61 – 172.31.192.80
- AimWorx address range: 172.31.192.81 – 172.31.192.100
- CiscoSecure platform address: 172.31.192.17

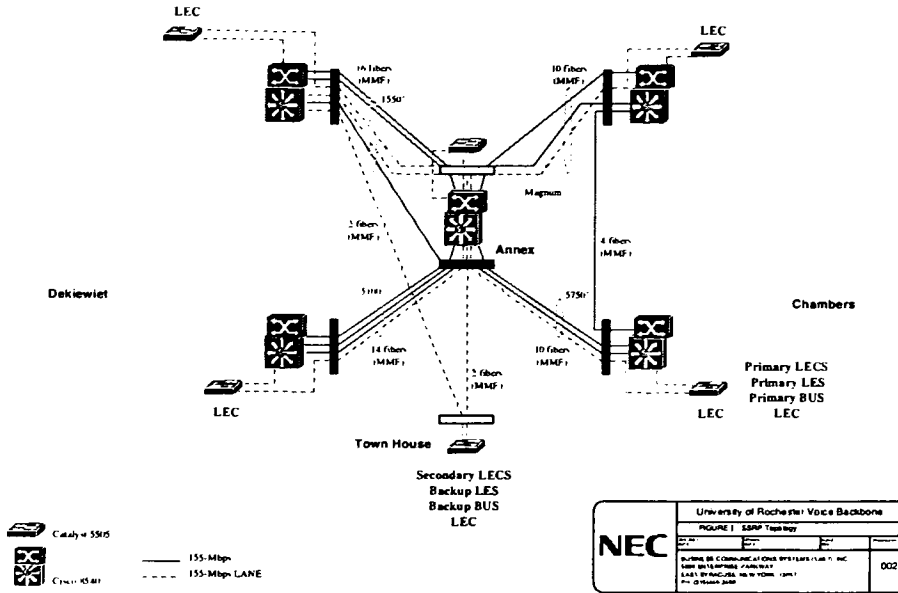


Figure 89: LANE topology and SSRP topology

Network Fault Tolerance and System Downtime: The new voice backbone has been designed to provide maximum reliability and availability of voice services for the University of Rochester. Fault tolerance and downtime for the Catalyst 5505 and 8540 are discussed below in detail below.

Catalyst 5505 Components: Each Catalyst 5505 has been provisioned with dual Supervisor Engines, one active and one standby, and dual, load-sharing power supplies for maximum availability of FCCS services. The active supervisor engine module sends information to the standby supervisor engine module to keep the NVRAM configuration on the standby supervisor engine current. If the software images on the active and standby supervisor engine modules are different, the active supervisor engine module downloads its image to the standby supervisor engine module.

Supervisor Engine Failure: If the background diagnostics on the active supervisor engine detect a major problem or an exception occurs, the active supervisor engine resets. The standby supervisor engine detects that the active supervisor engine is no longer running and becomes active. The standby supervisor engine can detect if the active supervisor engine is not functioning and can force a reset, if necessary. Once the reset supervisor engine comes up, it behaves as if a hot swap occurred, and then enters the standby mode.

When the transition from standby to active Supervisor engine in the Catalyst 5505 occurs, all modules that are on-line are reset and go through normal system diagnostic procedures. This allows the new active Supervisor engine to test the operation and function of each module. During the reset phase, the Spanning Tree algorithm will be recalculated for both Ethernet and LANE modules, in addition to the LANE module reestablishing all **Switched Virtual Circuits (SVCs)** for ELAN connectivity. During this time, the IMX will be unable to process voice calls due to absence of FCCS signaling, and the end user will hear a fast busy signal when attempting to place a call. Based on testing in the Syracuse lab, downtime is approximately 3 minutes.

LAN Emulation Link failure: The Catalyst 5505 is equipped with a dual PHY LAN Emulation module for uplinking to the Catalyst 8540 ATM backbone. The dual PHY LANE module is a **User-to-Network Interface (UNI)** module and provides an active and standby links for redundancy purposes. The

active link is provisioned with a MAC address from the Supervisor engine during the boot process. Through ILM registration with the local Catalyst B540 ATM switch, the **LANE** module is configured with a 20 byte **NSAP ATM address**. Because the Catalyst 5505 is dual homed, the NSAP address will change depending on what PHY is active.

If a link status change occurs on the LANE module, then LANE components such as the LES, BUS and LEC must change to reflect the new NSAP address change.. The LEC must go through the initialization and configuration, joining and registration processes for the LEC to become a participating member of an ELAN. During this phase, the IMX will be unable to process voice calls due to absence of FCCS signaling, and the end user will hear a fast busy signal when attempting to place a call. Based on testing in the Syracuse lab, estimated downtime is approximately 2 to 3 minutes.

Catalyst 8540 Components: Each Catalyst 8540 has been provisioned with dual **Route Processors (RP)** and **Multiservice Switch Processors (MSP)**, one active and one standby, and dual, load-sharing power supplies for maximum availability of FCCS services. The high system availability of the Catalyst B540 is achieved through the **HSA (High System Availability)** functionality of the Cisco IOS.

Route Processor Failure:In the event of a Multiservice Switch Processor (MSP) malfunction, failure auto detection schemes will identify the event and procedures for switchover to the standby module would be initiated. All the PVCs through the MSP would be preserved and would not suffer any cell loss during the switchover. While any Route Processor failure would not preserve SVCs during the initial IOS release, future releases of the IOS would deliver SVC preservation. During this phase, the IMX will be unable to process voice calls due to absence of end-to-end FCCS signaling through the ATM backbone. The end user will hear a fast busy signal when attempting to place a call. Based on testing in the Syracuse lab, estimated downtime is approximately 2 to 3 minutes.

Multiservice Switch Processor Failure:In the event of a Route Processor (RP) malfunction, auto detection schemes would initiate a switchover to the standby RP and the new RP would become available to receive control messages in less than 30 seconds. All the PVCs through the MSP would be preserved and would not suffer any cell loss during the switchover. While any Route Processor failure would not preserve SVCs during the initial IOS release, future releases of the IOS would deliver SVC preservation. During this phase, the IMX will be unable to process voice calls due to absence of end-to-end FCCS signaling through the ATM backbone. The end user will hear a fast busy signal when attempting to place a call. Based on testing in the Syracuse lab, estimated downtime is approximately 2 to 3 minutes.

Switch Virtual Circuit Reroute:The Catalyst B540 runs Private Network-Network Interface (PNNI) 1.0 code. PNNI is a dynamic routing protocol that provides quality of service (QoS) routes to signaling based on the QoS requirements specified in the call setup request. Dynamic routing protocols adapt to changing network conditions by advertising reachability and topology status information changes.

Voice calls will be routed between NEC IMXs over the backbone using Soft PVCs. By using Soft PVCs, automatic rerouting of voice calls would occur if the primary route becomes unavailable. Existing voice calls will be dropped and no new voice calls will be processed until a new Soft PVC is recalculated from route convergence. Based on testing in the Syracuse lab, estimated downtime is approximately 1 to 2 minutes.

This in fact highlights the use of an ATM core to build a voice network which could scale to integrate other services in a multi-service environment. The proposed solution is expected to integrate UofR's data services with the imminent connection to the Internet2 which is also ATM based in its core. As the services scale to multi-user level integrating them into a network infrastructure which can meet the growing needs in terms of QoS, performance, scalability as well as cost are of immense importance. This shows how IP and ATM can be integrated today with a view of protecting the investment in terms of adapting to future multi-service platforms.

6.7 Is efficiency of an issue in the overall picture?

In the OSI layers model the ATM is layer is considered to be a physical layer, with its **Adaptation Layers (AALs)** providing the equivalent of the link layer. Every link layer protocol has a limit to the maximum data size it can support, or its maximum **Protocol Data Unit (PDU)**. For example: Ethernet can support 1500 bytes IP packets, IEEE 802.3 with Connectionless mode LLC support 1492 bytes IP packets, etc. This size is called **Maximum Data Unit (MTU)**, and it is important for the IP network layer to know about, so it can divide large packet to suitable size for more efficient transmission. The default MTU for IP over ATM is defined in RFC-1626 to be 9180 bytes. It is the size that all IP over ATM nodes must support.

The AAL used for IP are AAL3/4 or AAL5. AAL5 is more economic and more commonly used. AAL5 can support AAL_SDU (Service Data Unit) up to 64 kilobytes long. It is quite large size, especially when considering the examples of link layers given above (1500 bytes) or even the "default MTU for IP over ATM" (9180 bytes).

How the AAL5's SDU does encapsulate IP packet? RFC-1483 specifies how to use AAL5 for multi-protocol encapsulation. This "multi-protocol" includes IP protocol. AAL SDU contain the IP packet + 8-byte header. The header is actually used for LLC/SNAP information needed for multi-protocol use in the same link (the same as defined in ISO 802.3). The PDU of AAL5 contain this SDU + 8-byte Trailer information + 0.47 bytes padding (to ensure the PDU length is a multiple of cell payload). This is displayed in figure.

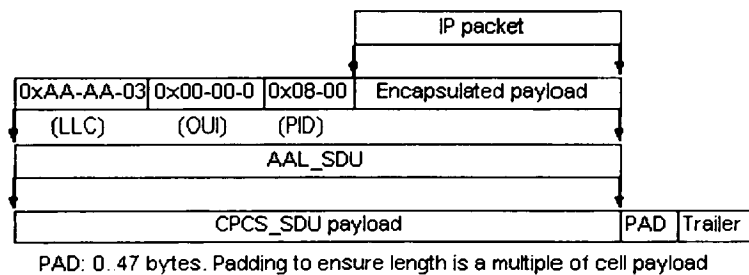


Figure 90: RFC-1483 encapsulation of IP packet

The PDU of AAL5 contains all the control information needed for the link layer and the IP packet itself. Its segmentation results in 53-bytes cells burst. The cell number depends on the size of the IP packet, which is the most important factor influences the efficiency. Generally, the larger the IP packet size, the more efficient it is, though even for small IP packet the efficiency is very close to its maximum value.

The most straightforward definition of efficiency is the average fraction of the cell's 53 bytes used to carry "useful data". The segmentation to cells in fixed size can result in some inefficiency, because the last cell may contain the padding, which is a waste of space. The smaller the IP packet is, the greater the inefficiency. Figure below plots the efficiency of the bit usage on AAL5 versus IP packet size. The "useful data" is the IP packet, excluding the LLC/SNAP header. Maximum theoretical efficiency is $100 \times (48/53) = 91\%$. Practically this value will never be reached, because of the overhead of the 8-bytes header and the 8-bytes trailer, but in large enough packets these overheads can be ignored.

The efficiency as a percentage of the "useful data" from the overall data **per one cell** does not take into account some other traffic paid for. For example, there are some cells used as control data of the network layer. For a **realistic traffic of IP packets**, it is required to find some metric that estimate the average number of ATM cells per IP packet. The first step is to define **Equivalent ATM Cells (EAC)**, which is how many ATM cells transmitted for every 100 typical TCP/IP or UDP/IP

packets. The "typical" 100 cells are empirical value, taken from the statistics of the IP traffic. The second step is to calculate the *Average_Packet_Size* for the given traffic. It's also calculated from the same statistics. The last step is to define **Effective Cell Utilization (ECU)** to be the average percentage of the 53 bytes cells actually used for the end-to-end IP traffic.

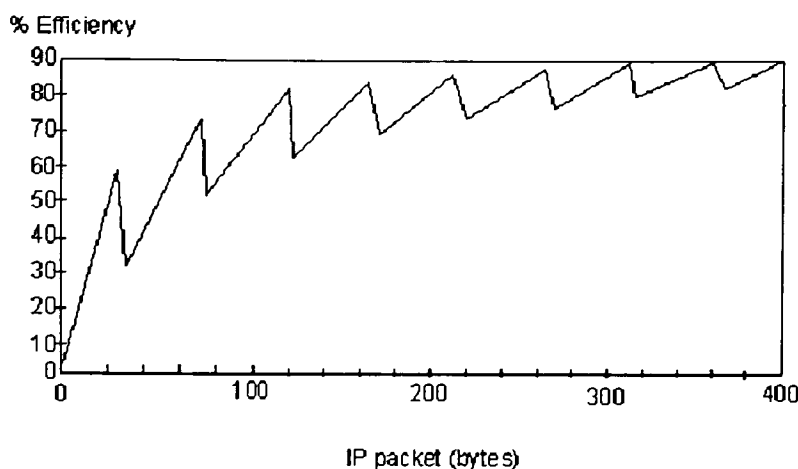


Figure 91: Efficiency of IP packets on AAL5 versus the packet size.

The ECU is given in:

$$ECU = 100 * (\text{Average_Packet_Size}) / (53 * EAC)$$

The EAC and ECU already reflect any overhead of using ATM for IP traffic (both the LLC/SNAP header and the trailer are considered to be overhead). The worst case ECU is 37.7%, when sending empty TCP/UDP payloads. Though the transport layer has empty payload, the IP will include header data of 40 bytes long, which will use 2 cells (EAC = 200). So $ECU = 100 * 40 / (53 * 200) = 37.7\%$. The best theoretical ECU is 90.5%, for 64k bytes IP packets, when $ECU = 100 * 64000 / (53 * 133500)$, though this value is approached in much smaller IP packets.

There are several documented research¹⁵⁶ to show that in most of the TCP over ATM implementations, the parameters that mainly influence the maximum TCP throughput over ATM, are the following:

- ⤵ IP MTU size of the ATM interface
- ⤵ TCP maximum segment size (MSS)
- ⤵ Processing power of the machine
- ⤵ Send/receive socket buffer size

The maximum throughput values that the end-system can achieve are close to the loop back measurements, but are much lower than the theoretically expected values. Possible reasons for the observed behavior have been listed as,

- ⤵ The current version of the PCI buses not being optimized for high speed network adapters such as ATM adapters, although the transfer rate that it provides is more than 1Gbit/s.
- ⤵ Most operating systems are not optimized for ATM
- ⤵ Relatively low processing power compared to the available transmission speeds
- ⤵ System memory or buffer sizes being insufficient due to the mis-match between system parameters and the TCP's retransmit policy.

¹⁵⁶ TCP Performance over ATM, <http://www.cc.surrey.ac.uk/Personnel/I.Andrikopoulos/Papers>

Besides, in many experiments the most powerful workstations that had been used did not reach the theoretical maximum, which indicates that the current adapter card driver, and protocol implementations reached their upper limits. So we are faced with the dilemma of re-engineering the functional transport protocol which we have depended for decades. This is more so as we move into gigabit speeds. Undoubtedly, as we agree there is an associated protocol overhead in running IP over ATM there are other issues associated with the higher layers, which needs to be addressed at the same time. We will discuss some of those issues here.

6.8 How efficient are the transport protocols associated with IP at higher wire speeds?

With the advent of high speed ASICs and the improvements in transmission technology has made today's network very fast and in fact is getting faster at a higher rate than the processing power of our desktops. This has in deed created an unusual caviar in high speed networks. The caviar being, no longer is the bandwidth the limiting factor but it is the processing power of the desktop. So we are faced with a situation of rethinking what we need to optimize in order to improve the overall throughput and efficiency of the network end-to-end. So when we analyze the impact of using various technologies on efficiency we need to see the greater picture. Raw bandwidth calculations as what was done before will not suffice on the contrary if will only serve to defeat the purpose if used in the wrong context. What we need to look at is the bandwidth-delay product, which is the product of the bandwidth and the round trip delay instead of the raw bandwidth which is the capacity of the pipe from sender to receiver and back in a full duplex connection.

Protocols for high speed networks¹⁵⁷

Lets look at TCP, based on which most of the multiservices are built in IP based networks. We have to deal with a number of issues as we implement faster networks. One such issue is that many protocols such as TCP use 16 to 32 bit sequence numbers which would give you at most 2^{32} , approximately 4 billion sequences. At gigabit speeds, assuming a data rate of 1 Gbps, it only takes 32 seconds to exhaust the range of sequence numbers. But we know that each packet on the internet has a TTL as high as 120 seconds which prohibits us from wrapping around the sequence number back to zero immediately, given the fact that all the bytes were received correctly by the recipient. The early assumption by many protocol designers, that the time to use up the entire sequence space would greatly exceed the maximum packet lifetime is beginning to be critical issues as we attempt to deliver multiservice applications at gigabit wire speeds.

The second issue is that communication speeds have improved much faster than computing speeds. So in order for the sluggishness in the speed of processing the protocol must be made even simpler so it will consume fewer processing cycles, which would facilitate a closer match with the desired wire speeds

The third issue is the flow-control algorithm implemented by most protocol such as TCP performs very badly on lines with high bandwidth-delay product. For example, a 4000 km line operating at 1 Gbps. The round trip transmission time is 40 msec, in which time a sender can transmit 5 megabytes. If an error occurs, it will be 40 msec before the sender is informed of it. If a flow control algorithm such as Go-back-N is used, the sender has to transmit not only the bad packet but also 5 megabytes worth of data that came after that. This is clearly an inefficient way of managing resources at high speeds.

The forth issue is that gigabit lines are fundamentally different from megabit lines in that long ones are delay limited rather than bandwidth limited. This is clearly shown in the figure. It shows the time taken to transfer 1 Megabit file at 1Gbps across 4000 km line. This proves a point, which needs to be addressed in high speed networks.

¹⁵⁷ Andrew S. Tanenbaum, Computer Networks, Prentice Hall, 1996

The fifth issue is one worthy of mentioning, not so much a technological or protocol specific one, but is a result of new multi-media centric applications. Many such applications are sensitive to variances in packet arrival times as well as the mean delay itself. In such cases slow but uniform delivery rate is most preferable to a fast-but-jumpy one.

Therefore, the **key here is to Design for speed and application requirements and not for bandwidth optimization.** So in this context which way do we go? Is Packet over Sonet the best alternative, given our objectives?

6.9 Is Packet over Sonet a viable alternative?

There are various differences in operating IP-over-SONET compared to running IP-over-ATM. Some of the important issues are summarized below.

Protocol Overheads

By far the biggest reason that if anyone is considering deploying **Packet-over-SONET (PoS)** as opposed to IP-over-ATM is the overhead imposed by ATM cell headers (5-bytes out of every 53-bytes), sometimes referred to as the cell tax. Additional overhead is added by AAL5 (padding, 8-byte trailer) and LLC/SNAP encapsulation (8-bytes).

The following table indicates the overheads introduced by each layer in the protocol stack when running IP-over-ATM over a SONET STS-3c link with an IP packet size of 576 bytes:

Protocol Layer	Available Bandwidth (Mbps)	Percent of Line Rate	Percent Overhead Added by Each Layer
SONET	155.520	100	3.7
ATM	149.460	96.6	9.43
AAL	135.362	87.5	6.41
LLC/SNAP	126.937	80.7	1.37
IP	125.918	79.6	0

Table 11: IP over ATM over SONET overhead

A similar comparison for IP-over-PPP over a SONET STS-3c link with an IP packet size of 576 bytes gives the following approximate results:

Protocol Layer	Available Bandwidth (Mbps)	Percent of Line Rate	Percent Overhead Added by Each Layer
SONET	155.520	100	3.7
PPP	149.460	96.6	1.54
IP	147.15	95.4	0

Table 12: IP over PPP over SONET overhead

These tables show that IP achieves only about 80 percent of the available line rate when operating over ATM whereas it achieves 95 percent of the line rate when running over SONET. The added capacity when running IP-over-SONET is very compelling when expensive wide-area or otherwise bandwidth-constrained links are used for interconnecting backbone routers. For environments where bandwidth is plentiful, such as local area networks, bandwidth efficiency is not as much of an issue.

Bandwidth Management

ATM provides a full suite of capabilities for managing the bandwidth allocation to the various information streams (VCCs) flowing over a link. It assigns flexible bandwidth to these VCCs based on the required quality of service. Because of its cell-switched nature, ATM allows multiple

information streams to share the same link at the same time, while guaranteeing a certain amount of bandwidth for each stream.

Point-to-point (PPP), on the other hand, does not have any provision for bandwidth management. It provides a simple point-to-point link, and the IP layer has to schedule its packet transmissions to ensure that each information flow receives its fair share of link bandwidth. There can be problems over slow links, in which the transmission of a large packet belonging to a low priority flow can block the transmission of other high priority packets. For example, a large packet in a low-priority file transfer flow can delay a much smaller but more time-sensitive voice packet. This variability in delay can negate the benefits of the bandwidth efficiency provided by IP-over-SONET, for delay sensitive real-time applications, over bandwidth constrained links.

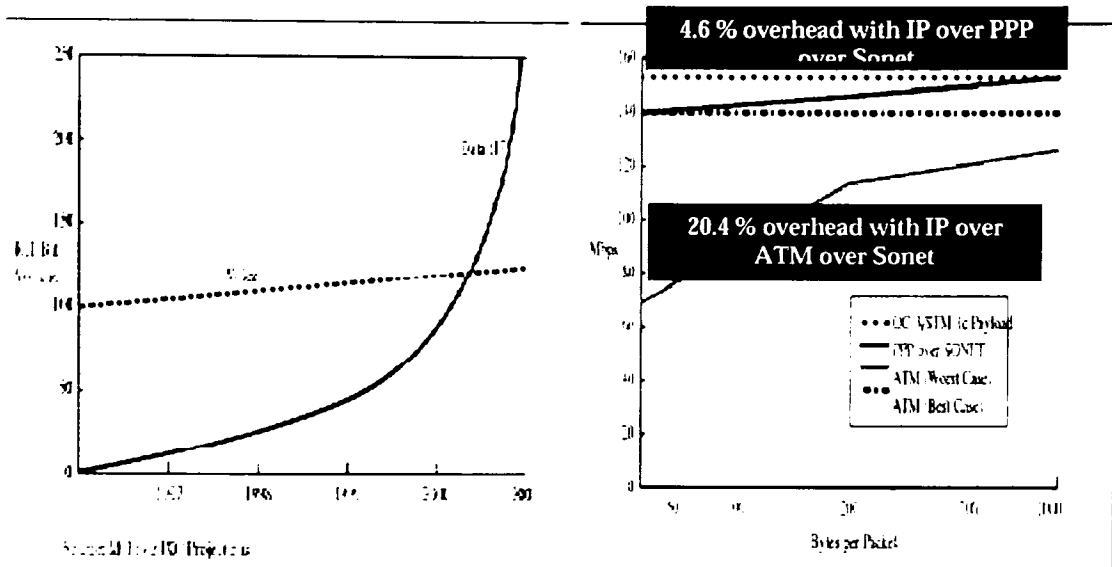


Figure 92: IP over ATM and IP over Sonet performance

Quality of Service

Quality of service (QoS) relates to parameters such as end-to-end packet delay, jitter, loss and throughput. ATM provides a rich set of QoS parameters that can be negotiated for each VCC. Intelligent queuing and scheduling mechanisms in the switches ensure that the negotiated QoS is provided. ATM provides various service classes that can fit different application requirements. For example, applications with very specific QoS requirements can use a **Constant Bit Rate (CBR)** or **Variable Bit Rate (VBR)** service. On the other hand, applications with elastic requirements can use **Available Bit Rate (ABR)** or **Unspecified Bit Rate (UBR)** service. These native ATM capabilities make it very easy to provide QoS at the IP level, at which each information flow with a specific QoS requirement can map to its own VCC with a specific QoS. For example, a voice flow can map to a real-time CBR or VBR connection while a file transfer can map to an ABR connection.

PPP operates over a single point-to-point link and does not provide any QoS capabilities. As mentioned earlier, the IP layer has to manage its packet transmissions intelligently to ensure proper QoS for the information flows. Although ATM provides a rich set of QoS parameters, the QoS-based services are restricted to the ATM path connecting two routers. To provide end-to-end QoS to IP packets, the routers still have to provide intelligent queuing and scheduling mechanisms. In that sense, when an IP network is overlaid on top of an ATM network, the routers see ATM connections as point-to-point links, similar to PPP, even though the actual communication may occur over a network of ATM switches.

Addressing & Routing

ATM is specified as a full network layer with extensive capabilities for addressing end systems and routing connections. ATM networks can span vast geographical areas, providing a universal interconnection mechanism between routers regardless of their location. In contrast, PPP operates over direct point-to-point links only and has no addressing or routing capabilities. In order to create a backbone network, point-to-point links have to be provisioned between the backbone routers. Multiple links have to be provisioned to allow for link failures. In some cases, a full mesh may need to be configured to minimize the number of hops needed to cross the backbone. A full mesh is not only very expensive, but may also be infeasible because access to pure SONET links in the wide area is limited.

When used with SVCs, ATM enables any-to-any connectivity among routers without the need to provision a full mesh. Even if some links in the ATM network fail, dynamic SVC routing can find alternate routes and always ensure a connection between any two routers. The most useful capability is that connections to other routers may be established over a single ATM interface, which can easily be obtained from the carriers.

In the case of backbone router networks, most routers will need to communicate with each other, which means a full mesh connectivity may eventually be needed regardless of whether point-to-point links are provisioned or SVCs are used. However, ATM still enables more flexible network engineering because of its ability to route the SVCs over different links and to connect one router to multiple destinations over the same access link.

Flow Control

ATM uses functions, such as **Call Admission Control (CAC)**, traffic shaping and **User Parameter Control (UPC)** or policing, and to ensure that information flows stay within the boundaries of the negotiated traffic contract. Excess traffic is tagged and may be discarded under network overload conditions. Thus end users get implicit information about congestion in the network based on tagged or lost packets. ATM's cell-level discarding can interact poorly with TCP's packet-level flow control. To alleviate this problem somewhat, hooks, such as **Partial Packet Discard (PPD)** or **Early Packet Discard (EPD)**, have been designed for ATM to recognize packet (AAL frame) boundaries and to discard entire frames under overload conditions.

Recently, the ATM Forum defined the ABR service. It provides explicit feedback for flow control by indicating the allowed rate at which the ATM endpoint can send traffic into the network. This rate may change as the network load changes, allowing the user to access the available bandwidth without overloading the network. Ideally, ABR will remove cell loss in the network and push congestion conditions towards the boundary of the ATM network. This will require routers to buffer more packets.

PPP provides no flow control mechanisms, so TCP's flow control operates directly over PPP links. As noted earlier, routers, whether they are connected over ATM or directly over SONET, see a pipe (of a certain bandwidth) between each other and have to employ suitable buffering mechanisms to ensure reasonable throughput.

Multiprotocol Encapsulation

ATM provides two mechanisms for multiple protocols to share the same link. The first mechanism, known as VCC Multiplexing, assigns each protocol to a separate VCC. The ATM layer multiplexes and demultiplexes VCCs so users do not need to add any other encapsulation headers to

distinguish the various protocols. The second mechanism, known as LLC Multiplexing, allows multiple protocols to share the same VCC. It adds an 8-byte encapsulation header to each packet that identifies the protocol to which it belongs. This form of multiplexing may be used when the number of VCCs available is limited (due to cost or capacity), and there is a need to share the VCCs among the various protocols.

PPP provides a form of multiprotocol encapsulation similar to LLC Multiplexing in ATM. It uses a 1- or 2-byte protocol identifier field as an encapsulation header. For the most part, the multiprotocol encapsulation capabilities of PPP and ATM are equivalent.

Fault Tolerance

ATM provides recovery from failed links and switches by routing connections around them using a dynamic routing protocol, called the **Private Network Node Interface (PNNI)** protocol. PNNI only had re-routing capability during the initial connection establishment, but the current version PNNI 2 has provisions for automatic re-routing of an established connection that is released due to network failure.

PPP does not have any fault tolerance capability because it operates over a single link. However, the underlying SONET layer has built-in protection switching to switch to the alternate ring when the working ring breaks down. This capability is also available to ATM when it operates over SONET.

6.10 Adaptation Layers for IP in an ATM World

The integration of ATM and IP into a coherent network is non-trivial. The two technologies have some very different approaches and perspectives on how networking should occur. It is vital that this integration moves beyond the simple building of overlay networks and embraces complete interworking. Unless this is achieved, our objectives are "obscured by clouds within clouds", which prevent the integrated management of **Quality of Service** and efficient resource management. Before trying to understand how these technologies can be brought together, and how these new adaptation layers will work, we first need to understand how they differ.

Two Visions of Networking

The first point that separates ATM from IP is the issue of connection-oriented and connectionless networking. IP is connectionless. If one user wishes to talk to another, they just dispatch a packet with the address of the recipient attached. No procedure is initiated to establish a connection between the two users, and no message is sent to conclude the conversation either. By contrast ATM is connection-oriented. When one user wishes to talk to another, a call set-up must occur and a defined path between the two parties must be established. Once this path is in place, any conversation between the two parties may occur. When they are finished, the path is removed.

These two strategies have their respective strengths. The connectionless approach is simple to implement and has proven to be a robust solution. However, it does not support full management of QoS. Connection-oriented networking can be more complex, as call establishment and clearing procedures are needed. However, these same processes allow paths with guaranteed performance to be established for each call there by providing complete QoS support. In order to run IP over ATM, we must therefore have a mechanism for integrating these connectionless and connection-oriented worlds.

A number of approaches have been discussed in the earlier sections as to how IP and ATM can be integrated. Each one attempts to build a set of ATM connections that can be used to carry the connectionless IP traffic to its destination. By doing this the traffic that is placed on an ATM connection is delivered to its destination without further analysis of the IP header. (It is switched at level 2, not routed at level 3). This is quicker and more efficient. It also confers the benefits of ATM QoS on the IP traffic.

Both the ATM Forum and the IETF have recognized the need for an effective mechanism to link ATM and IP. A number of standards have been developed to map IP over ATM networks. These include:

- RFC1577 - Classical IP over ATM, IETF
- LAN Emulation (LANE), ATM FORUM
- NARP and NHRP, IETF
- Multi-Protocol over ATM (MPOA), ATM FORUM
- Multi-Protocol Label Switching MPLS, IETF

These efforts have seen a progression from relatively simplistic approaches - suited to the needs of small communities in emulated LAN environments - to the latest developments that are designed to allow the core of the Internet to operate over ATM.

✂ Traffic or Topology

These various strategies can be grouped into two primary categories: those that are driven by network topology and those that are driven by network traffic. Topology-driven systems analyze the topology of the network and attempt to establish a set of connections that link the various end-points of the network. Their objective is to build a mesh of ATM virtual circuits that form shortcuts between end-points. By contrast, traffic-driven systems attempt to analyze the IP traffic flowing in the network and then build virtual circuits to carry these flows.

Clearly, traffic-driven systems offer a significant benefit. In the process of analyzing the traffic the network has the opportunity to learn not only where the traffic is flowing, but also to discover the type of traffic and the types of quality of service it needs. Once this is done, the network can pass the traffic flow to a virtual circuit with the appropriate type of service. Today there are only really two approaches to providing efficient adaptation of IP traffic to ATM core networks: **MPOA (Layered approach)** and **MPLS (integrated approach)**.

✂ MPOA Goes With the FLOW

The **Multi-Protocol Over ATM (MPOA)** specification of the ATM Forum is a good example of a traffic-based strategy. MPOA is designed as client/server architecture. MPOA Clients and their MPOA Server(s) are connected via LANE. MPOA Clients detect flows of packets that are being forwarded to a router that contains an MPOA Server. When the Client recognizes a flow that could benefit from a shortcut, bypassing the normal routed path, it requests the information to establish a shortcut to the destination. If a shortcut is possible, the MPOA Client caches the information, sets up a shortcut VCC, and forwards frames over the shortcut.

In this way MPOA edge systems monitor traffic entering the network and search for flows. Traffic that is not part of an identifiable flow is carried across a default (routed) channel. Traffic that is identified as belonging to a flow will be switched to its destination in an ATM VC. The network operator typically has the opportunity to tune this process of creation and clearing of virtual circuits to match the characteristics of their particular network and their users' traffic profiles.

The edge systems are able to utilize a wide variety of strategies to identify a flow. The specification requires that they monitor packet rates between source and destination pairs. Thresholds can be set to define packet rates and traffic volumes that will trigger the establishment of a circuit. Vendors have the option of providing additional mechanisms to control the establishment of circuits. For example, a switch can use the source port, the IP socket address or other information from the user's profile to both identify flows, and to classify traffic. In effect, these networks are driven by a set of rules defined by the switch vendor or network operator. They are "policy-based" networks.

✂ QoS with Today's Applications

One of the very significant benefits of this approach is the ability of the network to support existing multimedia applications without changing them. The flow detection intelligence of an MPOA network chooses the QoS required for each application by watching the application at work.

Traditional IP networks can only overcome their inherently variable service quality by the addition of new concepts such as RSVP and IPv6.

Both these technologies would require the re-development of many applications to take advantage of the promised "multi-media IP". This is too costly to contemplate. MPOA provides the logical way to up-rate our IP infrastructures to support these vital applications without wholesale application redevelopment.

⌘ **Dynamic Networks with MPOA**

Another advantage of these flow and policy-based strategies is that they adapt and adjust to the changing traffic patterns in a network. As new flows are found, and old flows cease, virtual circuits are established or cleared as necessary. This process results in a continual "fine tuning" of the networks structure to respond to the immediate requirements of the subscribers.

This is a revolution in network management. Historically we have either built static PVC based networks, or relied on the users to place SVC calls, as they needed them. PVC networks required manual intervention to respond to changing network requirements. SVC networks could result in chaos; each user makes decisions based only on their own requirements, without any awareness of the state of the network, or of other users' needs. MPOA specifications are complete. Today many vendors offer solutions using MPOA. It is proving a reliable and solid mechanism for carrying IP traffic.

⌘ **MPLS Work in Progress**

By contrast, **Multi-Protocol Label Switching (MPLS)** is at an earlier stage in the standardization process, and we have yet to see any working examples end-to-end. The MPLS standards are expected to be complete by the middle of 1999. MPLS has some similar goals to MPOA. It is based on the idea that routing is in fact two processes: The first process partitions the entire set of possible packets into a set of forwarding equivalence classes (FECs). The second process maps each FEC to a next hop. In MPLS, the assignment of a particular packet to a particular FEC is done just once, as the packet enters the network. The FEC to which the packet is assigned is encoded with a short, fixed length, value known as a "label". When a packet is forwarded to its next hop, the label is sent along with it; that is, the packets are labeled.

At subsequent hops, there is no further analysis of the network layer header of the packet. Rather, the label is used as an index into a table that specifies the next hop, and a new label. The old label is replaced with the new label (switched), and the packet is forwarded to its next hop. However, the mechanisms that are used to partition the incoming packets into FEC groups are not yet specified. We should expect some vendors to use just the IP destination address and others to use other information from the packet header.

⌘ **Scale is Important**

One of the significant issues about any IP-over-ATM scheme is its **scalability**. When we move from the small campus network to consider national or international networks, we must be sure that the mechanisms we choose for the network design are able to support the scale of the network. In reality, the only way this can be proven is by experience. Only when large networks are deployed can we be sure of the scalability of any technology. This is one reason for the acceptance of MPOA.

⌘ **MPOA for Today and for Tomorrow**

Today you can see MPOA networks in operation around the world. There is on going efforts to extend the functionality of MPOA through a set of MPOA extensions such as **MPOA-VPN** support, **MPOA for Frame Relay**, and **MPOA-QoS** extensions. These developments will enable MPOA to support the largest IP networks. Alongside this work is the development of **Real-time Multimedia**

Over ATM (RMOA) protocols and **Frame-Based ATM**. RMOA will exist as a parallel mechanism to MPOA and will provide efficient, optimized support for real-time traffic of any type. This work is allied to the development of version 3 of the H.323 protocols from the ITU, and will therefore provide an integrated and standardized approach to the carriage of real-time traffic in an ATM or IP environment.

With the emergence of gigabit transmission rates, there is also considerable work being done on mechanisms to allow the ATM control planes to be used with large, variable-length, frames. This work is known as Frame - Based ATM. It is not necessary to segment traffic into short cells at very high speeds, as network link latencies are reduced to such a degree. By allowing the ATM control plane protocols such as PNNI, MPOA, and RMOA to be extended to a frame-based environment, the Forum is providing a path for the complete integration of IP and ATM.

Frame-based ATM avoids the problems of VC-merge and the necessity for SAR processing and points the way forward for high-performance, very high-speed, ATM- based solutions for IP, real-time and non-real-time traffic.

6.11 Summary

We have come to a point where we have sufficient cause to recognize both IP and ATM for their specific strengths and niches and more so why they should be integrated in multi-service environments of tomorrow. The issue is how the Internet Protocol traffic will best work with Asynchronous Transfer Mode, or ATM, switching, the technology of the future for the enterprise network.

The ATM Forum, an industry consortium of vendors and carriers, has been working on a scheme it calls **Multiprotocol-Over-ATM (MPOA)**. At the same time, ATM switch vendors, have latched onto something called IP switching based on technology from Ipsilon Networks Inc. Both methods aim to address the fundamental issues of integrating IP and ATM. The plan is to combine the best features of the two protocols to increase the throughput of the enterprise network.

There are some basic design differences between IP and ATM, however. A key difference is the fact that the former is connectionless whereas the latter is connection-oriented. IP, because it is connectionless, requires all routers in the network to look at every bit of information that crosses their path to decide where to send that information. ATM, however, automatically knows the destination of a group of data, or a datagram, through special addressing and circuit provisioning. Each job gets its own circuit in ATM, so there's no need to look at each packet at every network node. But that special addressing required with ATM adds bits to each transmission, so it may be just as efficient to send small amounts of data through the router network as it would be to add overhead to them and send them via an ATM connection.

The question is how to enable the network to specify what traffic goes through the IP router network and what goes over ATM Switched Virtual Circuit, or SVC, connections. The MPOA and IP switching methods both base their routing decisions on "persistent flow," but there are differences in how each camp defines persistent flow. MPOA says if this is addressed to a specific destination, its persistent flow. It doesn't do anything else. IP switching intends to look at a lot of information in the headers to use TCP/IP's own coding to do a better job of guessing what flows are persistent and what flows are only one or two datagrams long. If something is only one or two datagrams long, there's no sense in me setting up an SVC to expedite it. On the other hand, if the flow is going to be a thousand datagrams long or a million datagrams long. IP switching tries to make a better judgment on which flows are persistent and which flows are not persistent.

Typically, persistent flow traffic represents larger amounts of data, such as file transfers. Nonpersistent flow traffic might be a single message sent by a network element in response to a query from a network management system. IP switching at current seems a better approach where QoS is not a prerequisite, since it's never worse than MPOA and it's sometimes better, but the choice is tough because there are certain conditions under which the two approaches are almost

100 percent equivalent. But MPOA and IP switching will not offer the same type of performance. MPOA, unlike IP switching, will support a variety of protocols over ATM in addition to IP and in delivering QoS instead of CoS, which is the only means by which one can guarantee service deliveries in large, scalable, enterprise networks. Perhaps more important, it will do it in a way that will allow end-to-end connections to be made based on quality of service needs. MPLS adopts many of MPOA's underlying concepts. So they are more similar than different and have a greater chance of being integrated and can facilitate tight integration too, depending on how IETF and ATM forum move in facilitating this. The table below shows the common, underlying principles in MPOA and MPLS. It shows in trying to adopt QoS capabilities MPLS is moving more closer to ATM than ever.

	ATM	MPLS
Switching Field	VP/VC	Label
Routable object	Virtual circuits	Label switched paths (LSP)
Source routing	Designated Transit Lists(DTL)	Explicit route list
Path setup	PNNI	Modified RSVP QOSPF Nimrod

Table 13: ATM and MPLS principles

	ATM	MPLS
Queuing	Per VC queuing	Per LSP queuing
Traffic scheduling	Weighted per VC scheduling	Weighted per LSP scheduling
QoS Routing	PNNI routing	To be determined

Table 14: QoS delivery in ATM and MPLS

These attributes of the ATM/MPLS can be used in a manner to make connections regardless of whether there's a specific need for quality of service. The Internet Engineering Task Force's resource reservation protocol, known as RSVP, will let some IP applications make requests for connections based on quality of service or other requirements. Those requirements could be bandwidth, time sensitivity or class of service in terms of error rate or reliability depending on how they are setup end-to-end. Also factors such as efficiency come in to play when higher layer protocol as tunneled or carried in the payload of an underlying network. ATM is blamed for imposing an undue overhead as compared to Packet over Sonet. In this context, ATM Forum has come up with a frame-based solution to overcome this. But, is this the right approach? Is efficiency is what we are seeking to accomplish? Each technology has its place in the broader map, and as such technologies such as PoS, xDSL will most likely be used in the edges over point-to-point high-speed links carrying either IP or ATM depending on the QoS and integrated service requirements. But we still need to reengineer the transport layer protocols such as TCP to perform better on such lines with high bandwidth-delay products.

By replacing end-to-end ATM signaling with proprietary hop-by-hop signaling, IP switching trades off scalability, performance, and low latency for "fewer lines of code." By forgoing PNNI-the routing protocol designed specifically to leverage the QoS capabilities inherent in ATM, IP switching sacrifices the ability to perform QoS-intelligent routing. MPOA, in contrast, leverages all of the benefits of ATM while ensuring that it interoperates easily with other technologies and protocols. If IP switching is going to give up all the benefits of ATM, why bother converting those frames into cells? MPLS is proposing address these issues which would guarantee QoS by means of mapping **CoS (in IP) to QoS (in ATM)** and by implementing a QoS aware routing protocol such as PNNI in the core. But ATM is able deliver QoS guarantees end-to-end by means of MPOA and PNNI today. Besides ATM has been chosen as the alternative for many of the next generation Internet initiatives such as vBNS and Internet 2, but how IP will be integrated will change depending on the impending integrated service and QoS requirements. But they do have definite plans to implement

QoS based routing in the core. We also see Sprint promoting ATM in a big way with couple of universities like University of Rochester adopting ATM as their choice of technology/platform for the next millennium in delivering services to their customers/users.

This trend is also reflected among major manufacturers of switching/routing equipment. Though most vendors are implementing some form of flow switching based on IP, they do have definite plans to support MPOA/PNNI in the future. It is the same with ATM providers in that today they aren't offering various qualities of service levels that they could, but with the on going effort to fine tune the MPOA/PNNI specification to meet the demands of the millennium, they do have definite plans for it in the future. The industry stride is towards integrating IP and ATM is definite. It will be a combination of MPOA, MPLS or both with a QoS aware routing protocols such as PNNI/1-PNNI in the core.

7. Final thoughts and Recommendations

"Cisco Systems, Inc., the worldwide leader in networking for the Internet, today announced the first-of-its-kind integrated IP-ATM Class of Service (CoS) capability available with Cisco's latest Cisco IOS® software releases. Cisco's IP-ATM CoS capability is part of Cisco's ongoing investment in solutions that bring the richness of Cisco IOS software to customers with investments in IP and ATM networks." SAN JOSE, Calif. November 17, 1998 -- Cisco Systems, Inc.

Toward a New IP and ATM Integration Paradigm

Integrated Services

The growing importance of multimedia applications based on IP provides evidence of the limitation introduced by the availability of a unique service class that is best effort. Although many applications, such as e-mail and file transfer protocol (ftp), take advantage of increased speed but do not require a specific QoS, other applications (very sensitive are those requiring a high level of interactivity) do require guaranteed quality.

These considerations have brought about an evolution of the present enterprise network toward an integrated services architecture, introducing QoS support. This evolution entails dramatic changes in the models and implementations. Key elements of the integrated services are:

- The definition of new service classes
- The definition of a control protocol to reserve resources in the network, such as RSVP (Resource Reservation Protocol), DiffServe to assure CoS
- The identification of specific flows that require QoS guarantees – MPOA/MPLS
- The introduction of classification and scheduling mechanisms in the routers to deal with different IP flows
- The definition of an appropriate routing protocol which can integrate the L2/L3 devices in enable user applications to take advantage of QoS – PNNI/ I-PNNI

The support of new service classes relaxes traditional connectionless behavior toward a more connection-oriented paradigm. The identification of IP "flows" allows associating with each of them a specific path, although maintained as a soft state. The IP service classes currently being defined are guaranteed QoS, for the delivery of IP packets within a fixed delay and with no loss, and controlled load, which allows emulation of a best-effort service over uncongested networks. RSVP can be seen as "IP signaling," enabling reservation of resources within IP networks to be used by specific unidirectional IP flows. RSVP provides a way for every sender to establish paths for identified IP flows. The resource reservation is committed on a flow basis, along the established path; it is up to the receiver to start a reservation procedure.

ATM technology offers a powerful mechanism to identify different flows by means of separate VCs and to associate a proper QoS with each VC. This allows IP applications to make much more efficient use of the transport resources and to exploit the intrinsic QoS support of ATM. An effective support of RSVP in an ATM environment requires a strong integration between functions and mechanisms that are present both at the IP and ATM layers, such as,

- ⌘ Mapping IP service classes(Class of service) onto corresponding ATM service classes (Quality of service)
- ⌘ Mapping the IP reservation protocol(RSVP) to ATM signaling(PNNI and I-PNNI)
- ⌘ Managing ATM VCs to carry specific IP flows

The integration of RSVP and ATM requires bringing together the PNNI based integrated services model, MPOA and MPLS. Such model should cope with several problems, such as:

- z Definition of a flow identification mechanism (label), able to support different aggregation levels; labels have to be used to discriminate both IP packets and ATM cells belonging to the same flow.
- z Preservation the overall scalability of the model, supporting high levels of aggregation; a merge functionality at the ATM layer is required in order to reduce the number of used VCs.
- z Definition of control protocols for routing and label information distribution, able to support new services based on QoS and multicast.

The different approaches reviewed earlier have strengths and weaknesses with respect to the different problems, and none of them appears to solve all the networking requirements. A realistic solution to integrate IP routing and ATM switching will probably be a combination of the above approaches.

Although layered routing is the simplest approach to routing IP over ATM networks, its simplicity creates some mismatches, which make the concept of layered routing far inferior to integrated routing. This is particularly the case in multi-service, enterprise networks. The partial topology maps generated by two independent routing protocols, both lacking global topology information about the network, result in fragmented end-to-end routes. These routes eventually become concatenations with L3 topology information of partial routes established with L2 topology information. Apart from considerations of overall routing optimality, the partial topology view also makes layered routing more vulnerable to routing loops. Without special precautions, transient effects present a higher risk of creating routing loops.

The use of a routing protocol with complete L2/L3 topology information eliminates the above problems. An integrated routing scheme that optimizes the use of different node types (IP-only, ATM-only, combined IP/ATM node) and different forwarding modes is more complex and introduces some problems that have yet to be resolved. For example, rapidly varying network conditions may cause transmission and processing overload due to excessive information flooding and route recalculations. Consequently, constraints need to be imposed that may well result in less optimum route assignments. However, these deviations are more manageable (and thus predictable) than the random results of layered routing paths.

Dual Mode Routing and Forwarding: While there are clear benefits to determining a routing path from a single protocol with complete L2/L3 topology information, there is no reason routing paths and forwarding modes should be assigned in the same way for different traffic types. As mentioned earlier, two alternatives for integrated routing are currently being developed. Both are primarily suited to particular traffic types.

Best-Effort Traffic: Best-effort traffic is typically well served with the MPLS-style enhanced forwarding scheme. The MPLS-based model is primarily aimed at boosting the forwarding throughput of best-effort traffic along L3 routing paths, which are determined by traditional hop-by-hop routing algorithms. Its major merit is that it effectively combines L2/L3 forwarding technologies with minimum control overhead.

In practice, partial L2 shortcuts (i.e., concatenated VP/VC segments) are established for aggregate L3 traffic flows. The aggregation allows economical usage of costly resources (VP/VC space, state synchronization overhead, L3 processing, etc.), while the associated lightweight control protocol, which is basically a VP/VC handshake between neighboring nodes, ensures rapid adaptation to varying network conditions. The lightweight nature of an MPLS-style control protocol makes it possible to adapt quickly to changing conditions, for example, by setting up, removing, or rearranging L2 shortcuts depending on actual traffic load or changing topologies. As the L3 topology map is not influenced by the actual forwarding mode, it is possible to optimize forwarding without a topology update or route recalculation penalty. What happens in practice is no more than a local swap between forwarding at L2 and L3. One could call it "complete L3 caching" as an advanced extension of "route caching" which is common in current routers. This

creates a very flexible and adaptive L2/L3 forwarding scheme, which makes optimum use of the available resources, while offering a genuinely best effort service to its customer traffic.

Guaranteed QoS Traffic: Where non default requirements come into the picture (real-time or multimedia traffic, guaranteed delivery, etc.), better results are obtained by using more sophisticated signaling and routing protocols which are designed to meet these requirements efficiently. Routes can vary depending on specific QoS requirements. Not only network topology, but also complex traffic parameters (e.g., link load, latency, available bandwidth) are taken into consideration in determining the route. The upcoming I-PNNI is a good example. Based on PNNI, it has a high degree of scalability thanks to its distributed and hierarchical nature, but also provides the tools needed to handle sophisticated QoS parameters and take them into account during routing.

Dual-Mode Integrated Routing: While it is essential to determine a route based on complete knowledge of the state of the dynamic components of a network (i.e., L2 and L3 topology and, if necessary, traffic conditions), the integration of L2/L3 routing does not prevent different types of traffic from being routed and forwarded in a distinct fashion. More specifically, a differentiation between best-effort and guaranteed QoS traffic enables the routing and forwarding characteristics to be matched more closely to the nature of each traffic type. Such a dual-mode integrated routing paradigm capitalizes on the merits of dedicated routing/forwarding schemes: a best-effort mode, applying a uniform simple routing metric with an associated lightweight signaling protocol to flexibly combine L2/L3 forwarding, and a guaranteed mode, applying appropriate complex routing metrics to meet specific QoS guarantees, with associated fully featured signaling). Note that applying different routing algorithms does not prevent a single routing protocol from being used to collect the necessary L2/L3 topology information.

The dual-mode approach also prevents dedicated best-effort/QoS routing schemes from being extended and applied where they are inappropriate. The MPLS-based model for best-effort traffic holds as long as the signaling is lightweight enough to justify a single cost metric (e.g., hop count) for different types of forwarding. The lightweight nature will, however, quickly erode when QoS guarantees need to be included. The I-PNNI model, on the other hand, offers excellent opportunities for routing and forwarding with hard QoS guarantees, but runs into complexity problems which require solutions that are far over engineered for best-effort traffic.

What needs to be further elaborated is how MPLS can rely on the scalable routing of the PNNI family while skipping its complex control part. How to flexibly share bandwidth between MPLS and (I-)PNNI VP/VC sets also needs further in-depth study. As a first step, a static bandwidth boundary can be installed, treating the best-effort segment as a large constant bit rate pipe and leaving the rest to the resource management of ATM protocols. In a later stage, an unspecified or available bit rate pipe seems more appropriate to ensure maximum sharing of bandwidth resources. Finally, as well as sharing bandwidth, the VPI/VC space could be dynamically shared between both protocols.

Advantages of integration in retrospect

- ⌘ Link Speed
 - ⌘ Aggregate slow link speeds and IP based flows into high speed core links
- ⌘ Fast automatic rerouting
 - ⌘ SPVC can be routed around failure (without affecting router connectivity) via PNNI
- ⌘ Traffic Engineering
 - ⌘ SPVC assigned to preferred paths(DTL)
 - ⌘ PNNI handles reroute on failure and unanticipated load(crankback)
 - ⌘ Fully utilizes all ATM resources (OA&M)
- ⌘ Traffic Management
 - ⌘ Traffic policing and shaping (GCAC)
 - ⌘ Per VC scheduling and queuing (policying & scheduling)
 - ⌘ Packet level discard (EPD, PPD)

ATM - Enabled IP Networks

ATM and IP are facts of life. Each has its benefits and strengths. Given the work that has been undertaken by the ATM Forum we now have the technology to build integrated IP/ATM networks. By deploying ATM core networks and IP access layers, we are able to get the best of both worlds. These "ATM-enabled" IP networks deliver the best of IP with the reliability, guaranteed performance and multi-service capability of ATM.

IP flexibility and open architecture is preserved. ATM-enabled IP networks can interwork with normal IP routed networks to provide the reach and openness we require. ATM's guaranteed quality of service shines through. ATM-enabled IP networks can control network latency and jitter to carry voice, data, video, and anything else you need without having to change the application. Other traffic can be supported directly by ATM - enabled IP networks. Frame Relay, voice, and video traffic can be efficiently adapted to run alongside the IP traffic.

Bridging the Gap between IP and ATM

The ATM Forum **Voice Trunking over ATM (VTOA)** work is showing the way for efficient, reliable, and clean transport of bulk voice services over a statistical network. Its real-time VBR adaptation mechanisms are a perfect match for the variable, but time-critical, characteristics of toll-quality voice traffic. Voice over IP clearly will have its place, ideally suited to the ad-hoc, low volume, corporate voice traffic. The **Real Time Multimedia Over ATM (RMOA)** through H.323 and PSTN/ISDN/IP/ATM integration provides a clear model for the future service of voice/video content. In order to bridge the gap that exist between IP and ATM architectures the following steps are necessary to meet the said objectives.

- ⌘ Scaling with a mesh of router adjacencies
 - ⌘ Separation of IP data forwarding from adjacencies on routers
 - ⌘ use of I-PNNI, PAR, QOSPF
- ⌘ Direct support for Differentiated services
 - ⌘ ATM UBRw
- ⌘ Enhanced efficiency with high speed interfaces
 - ⌘ Frame based ATM
 - ⌘ PoS with MPLS
- ⌘ Measurement based Multiprotocol Switch Router
 - ⌘ supports frames and cells per port basis
 - ⌘ Multiple routing and signaling protocol - PNNI, RSVP, MPLS, etc
 - ⌘ Multiple instances of routing/switching protocols
 - ⌘ Configured bindings between an instance and set of ports
 - ⌘ Configured interworking between instances
 - ⌘ Constrained based routing
- ⌘ IP routers and routing protocol must become smarter about constraints such as traffic load
 - ⌘ Traffic engineering of coarse scale paths between routers
 - ⌘ Fine-grained forwarding based on path congestion
 - ⌘ Applicable to overlay models and MPLS
 - ⌘ Work in progress - QOSPF, IS-IS, nimrod

IP's MPLS and ATM's MPOA are much more alike than different, but MPOA can deliver today what MPLS promises to deliver tomorrow. With the advent of technologies such as MPOA, MPLS, RSVP, PNNI, etc under the auspices of ATM Forum and IETF Integrated multi-services are showing that best concepts from both IP and ATM inevitably merging to form the landscape for the future multi-service oriented network and this is in deed a New Paradigm in multi-service networks. The networks of tomorrow will be based on ATM cores and IP shells. The cores will be engineered on high speed technologies such as SONET, B-ISDN and PDH whilst the access networks are engineered using available technologies such as Frame Relay, ISDN, xDSL, FDDI, Gigabit Ethernet, Fast Ethernet, etc depending on the service requirements and levels that need to prevail. The figure below shows a snap shot of what the future networks will be.

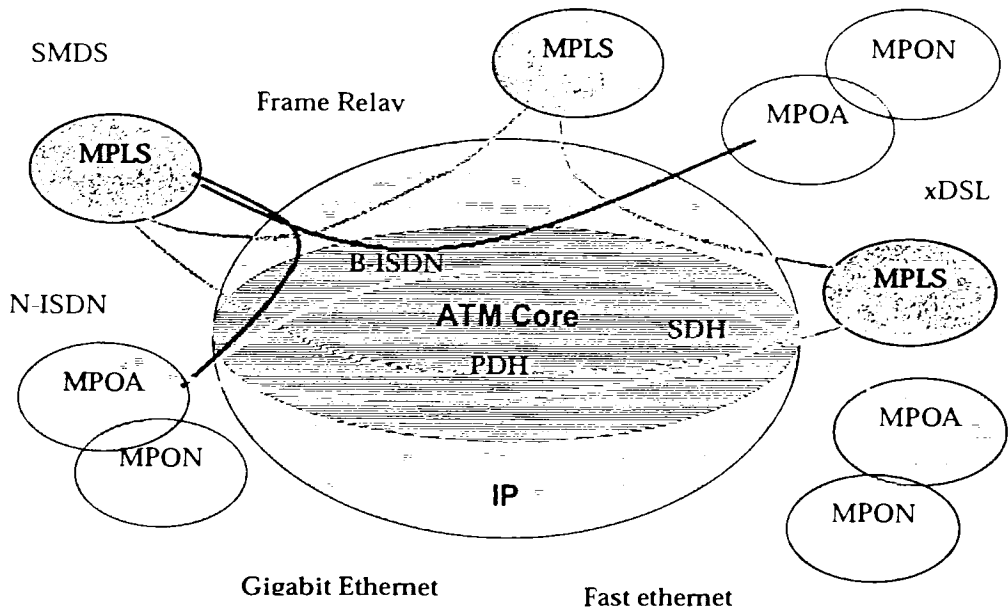


Figure 93: Multi-service networks of the future

In summary, MPOA/MPLS together with PNNI/I-PNNI as their routing counterparts with their future extensions show the way forward for the development of high-performance, multi-service, ATM-enabled IP networks.

"ATM enabled, IP networks – Is the new paradigm in multi-service internetworking"

Appendices

Appendix 1 – OSPF definitions

Here are some definitions, which are necessary to understand the sequence of operations described later in this section:

Area : A set of networks within a single autonomous system that have been grouped together. The topology of an area is hidden from the rest of the autonomous system, and each area has a separate topological database (see below). Routing within the autonomous system takes place on two levels, depending on whether the source and destination of a packet reside in the same area (intra-area routing) or different areas (inter-area routing).

- Intra-area routing is determined only by the area's own topology. That is, the packet is routed solely on information obtained within the area; no routing information obtained outside the area can be used.
- Inter-area routing is always done via the backbone.

The division of an autonomous system into areas enables a significant reduction in the volume of routing traffic required to manage the routing database for a large autonomous system.

Backbone : The backbone consists of those networks not contained in any area, their attached routers, and those routers that belong to multiple areas. The backbone must be logically contiguous. If it is not physically contiguous, the separate components must be connected using virtual links (see below). The backbone is responsible for the distribution of routing information between areas. The backbone itself has all the properties of an area; its topology is separate from that of the areas.

Area Border Router : A router connected to multiple areas. An area border router has a copy of the topological database for each area that it is connected to. An area border router is always part of the backbone. Area border routers are responsible for the propagation of inter-area routing information into the areas to which they are connected.

Internal Router : A router which is not an area border router.

AS Border Router (ASBR) : A router which exchanges routing information with routers belonging to other autonomous systems. All routers in the AS know the path to all AS boundary routers. An ASBR may be an area border router or an internal router. It need not be part of the backbone.

The nomenclature for this type of router is somewhat varied. RFC 1583, which describes OSPF uses the term AS Boundary Router. RFC 1267 and 1268 which describe BGP use the terms Border Router and Border Gateway. RFC 1340 which describes the interaction between OSPF and BGP uses the term AS Border Router. We shall use the last term consistently when describing both OSPF and BGP.

Virtual Link : A virtual link is part of the backbone. Its endpoints are two area border routers which share a common non-backbone area. The link is treated like a point-to-point link with metrics cost equal to the intra-area metrics between the endpoints of the links. The routing through the virtual link is done using normal intra-area routing.

Transit Area : An area through which a virtual route is physically connected.

Stub Area : An area configured to use default routing for inter-AS routing. A stub area can be configured where there is only a single exit from the area, or where any exit may be used without preference for routing to destinations outside the autonomous system. By default inter-AS routes are copied to all areas, so the use of stub areas can reduce the storage requirements of routers within those areas for autonomous systems where a lot of inter-AS routes are defined.

Multi-access Network : A physical network that supports the attachment of multiple routers. Each pair of routers on such a network is assumed to be able to communicate directly.

Hello Protocol : The part of the OSPF protocol used to establish and maintain neighbor relationships. This is not the Hello protocol described in the Hello Packet.

Neighboring routers : Two routers that have interfaces to a common network. On multi-access networks, neighbors are dynamically discovered by the Hello protocol.

Each neighbor is described by a state machine, which describes the conversation between this router and its neighbor. A brief outline of the meaning of the states follows.

Down : Initial state of a neighbor conversation. It indicates that there has been no recent information received from the neighbor.

Attempt : A neighbor on a non-broadcast network appears down and an attempt should be made to contact it by sending regular Hello packets.

Init : A Hello packet has recently been received from the neighbor. However, bi-directional communication has not yet been established with the neighbor (that is, the router itself did not appear in the neighbor's Hello packet).

2-way : In this state, communication between the two routers is bi-directional. Adjacencies can be established, and neighbors in this state or higher are eligible to be elected as (backup) designated routers.

ExStart : The two neighbors are about to create an adjacency.

Exchange : The two neighbors are telling each other what they have in their topological databases.

Loading : The two neighbors are synchronizing their topological databases.

Full : The two neighbors are now fully adjacent; their databases are synchronized.

Various events cause a change of state. For example, if a router receives a Hello packet from a neighbor that is down, the neighbor's state changes to init, and an inactivity timer is started. If the timer fires (that is, no further OSPF packets are received before it expires) the neighbor will return to the down state. Refer to RFC 1583 for a complete description of the states and information on the events, which cause state changes.

Adjacency : A relationship formed between selected neighboring routers for the purpose of exchanging routing information. Not every pair of neighboring routers becomes adjacent. In particular, not every pair of routers will stay synchronized. If all neighbors were to be synchronized, the number of synchronized pairs on a multi-access network such as a LAN would be $n(n-1)/2$ where n is the number of routers on the LAN. In large networks, the synchronization traffic would swamp the network, rendering it unusable. The concept of adjacencies is used to limit the number of synchronized pairs to $2n-1$, ensuring that the amount of synchronization traffic is manageable.

Link State Advertisement : Refers to the local state of a router or network. This includes the state of the router's interfaces and adjacencies. Each link state advertisement is flooded throughout the routing domain. The collected link state advertisements of all routers and networks form the area's topological database.

Flooding : The process of ensuring that each link state advertisement is passed between adjacent routers to reach every router in the area. The flooding procedure is reliable.

Designated Router : Each multi-access network that has at least two attached routers, has a Designated Router. The Designated Router generates a link state advertisement for the multi-access network. It is elected by the Hello protocol. It becomes adjacent to all other routers on the network. Since the topological databases of all routers are synchronized through adjacencies, the Designated Router plays a central part in the synchronization process.

Backup Designated Router : In order to make the transition to a new Designated Router smoother, there is a Backup Designated Router for each multi-access network. The Backup Designated Router is also adjacent to all routers on the network, and becomes Designated Router when the previous Designated Router fails. Because adjacencies already exist between the Backup Designated Router and all other routers attached to the network, new adjacencies do not have to be formed when the Backup Designated Router takes over from the Designated Router, shortening the time required for the takeover considerably. The Backup designated router is elected using the Hello protocol.

Interface : The connection between a router and one of its attached networks. Each interface has state information associated with it, which is obtained from the underlying lower-level protocols and the OSPF protocol itself. A brief description of each state is given here. Please refer to RFC 1583 for more details, and for information on the events that will cause an interface to change its state.

Down : The interface is unavailable. This is the initial state of an interface.

Loopback : The interface is looped back to the router. It cannot be used for regular data traffic.

Waiting : The router is trying to determine the identity of the Designated Router or its backup.

Point-to-Point : The interface is to a point-to-point network or is a virtual link. The router forms an adjacency with the router at the other end.

Note: The interfaces do not need IP addresses. Since the remainder of the internet has no practical need to see the routers' interfaces to the point-to-point link, just the interfaces to other networks, any IP addresses for the link would be needed only for communication between the two routers. To conserve the IP address space, the routers can dispense with IP addresses on the link. This has the effect of making the two routers appear to be one to IP but this has no ill effects. Such a link is called an unnumbered link.

DR Other : The interface is on a multi-access network but this router is neither the Designated Router nor its backup. The router forms adjacencies with the Designated Router and its backup.

Backup : The router is the Backup Designated Router. It will be promoted to Designated Router if the present Designated Router fails. The router forms adjacencies with every other router on the network.

DR : The router itself is the Designated Router. The router forms adjacencies with every other router on the network. The router must also originate a network links advertisement for the network node.

Type of Service (TOS) metrics : In each type of link state advertisement, different metrics can be advertised for each IP Type of Service. A metric for TOS 0 (used for OSPF routing protocol packets) must always be specified. Metrics for other TOS values can be specified; if they are not, these metrics are assumed equal to the metric specified for TOS 0.

Link State Database : Also called the directed graph or the topological database. It is created from the link state advertisements generated by the routers in the area.

RFC 1583 uses the term Link State Database in preference to topological database. The former term has the advantage that it describes the contents of the database, the latter is more descriptive of the purpose of the database -- to describe the topology of the area. We have previously used the term topological database for this reason, but for the remainder of this section where we discuss the operation of OSPF in more detail, we will refer to it as the Link State Database.

Shortest-Path Tree : Each router runs the Shortest Path First (SPF) algorithm on the Link State Database to obtain its shortest-path tree. The tree gives the route to any destination network or host as far as the area boundary. It is used to build the routing table.

Because each router occupies a different place in the area's topology, application of the SPF algorithm gives a different tree for each router, even though the database is identical. Area border routers run multiple copies of the algorithm but build a single routing table.

Routing table : The routing table contains entries for each destination: network, subnet or host. For each destination, there is information for one or more types of service (TOS). For each combination of destination and type of service, there are entries for one or more optimum paths to be used.

Area ID : A 32-bit number identifying a particular area. The backbone has an Area ID of zero.

Router ID : A 32-bit number identifying a particular router. Each router within the AS has a single router ID. One possible implementation is to use the lowest numbered IP address belonging to a router as its router ID.

Router Priority : An 8-bit unsigned integer, configurable on a per-interface basis indicating this router's priority in the selection of the (backup) Designated Router. A Router Priority of zero indicates that this router is ineligible to be the Designated Router.

Appendix 2 – BGP definitions

Here are some definitions, which are necessary to understand the sequence of operations described later in this section:

BGP speaker : A system running BGP.

BGP neighbors : A pair of BGP speakers exchanging inter-AS routing information. BGP neighbors may be of two types:

Internal : A pair of BGP speakers in the same autonomous system. Internal BGP neighbors must present a consistent image of the AS to their External BGP neighbors. This is explained in more detail below.

External : A pair of BGP neighbors in different autonomous systems. External BGP neighbors must be connected by a BGP connection as defined below. This restriction means that in most cases where an AS has multiple BGP inter-AS connections, it will also require multiple BGP speakers.

BGP session : A TCP session between BGP neighbors, which are exchanging routing information using BGP. The neighbors monitor the state of the session by sending a keepalive message regularly (the recommended interval is 30 seconds).

AS Border Router (ASBR) : A router which has a connection to multiple autonomous systems.

The nomenclature for this type of router is somewhat varied. RFC 1583, which describes OSPF, uses the term AS Boundary Router. RFC 1267 and 1268, which describe BGP-3, use the terms Border Router and Border Gateway. RFC 1340, which describes the interaction between OSPF and BGP-3, uses the term AS Border Router. We shall use the last term consistently when describing both OSPF and BGP. BGP-3 defines two types of AS Border Router, depending on its topological relationship to the BGP speaker which refers to it.

Internal : A next hop router in the same AS as the BGP speaker.

external : A next hop router in a different AS from the BGP speaker.

The IP address of a border router is specified as a next hop destination when BGP-3 advertises an AS path (see below) to one of its external neighbors. Next hop border routers must share a physical connection (see below) with both the sending and receiving BGP speakers. If a BGP speaker advertises an external border router as a next hop, that router must have been learned of from one of that BGP speaker's peers.

AS connection : BGP-3 defines two types of inter-AS connection.

physical connection : An AS shares a physical network with another AS, and this network is connected to at least one border router from each AS. Since these two routers share a network, they can forward packets to each other without requiring any inter-AS or intra-AS routing protocols (that is, they require neither an IGP nor an EGP to communicate).

BGP connection : A BGP connection means that there is a BGP session between a pair of BGP speakers, one in each AS, and this session is used to communicate the routes through the physically connected border routers that can be used for specific networks. BGP-3 requires that the BGP speakers must be on the same network as the physically connected border routers so that the BGP session is also independent of all inter-AS or intra-AS routing protocols. The BGP speakers do not need to be border routers, and vice versa. In fact, BGP speakers do not need to be routers: it is quite feasible for a host to provide the BGP function and to pass exterior routing information to one or more border routers with another protocol.

The term BGP connection can be used to refer to a session between two BGP speakers in the same AS.

Traffic type : BGP-3 categorizes traffic in an AS as one of two types:

local : Local traffic is traffic, which either originates in or terminates in that AS. That is, either the source or the destination IP address is in the AS.

transit : Transit traffic is all non-local traffic. One of the goals of BGP is to minimize the amount of transit traffic.

AS Type : An AS is categorized as one of three types:

stub : A stub AS has a single inter-AS connection to one other AS. A stub AS only carries local traffic.

multihomed : A multihomed AS has connections to more than one other AS but refuses to carry transit traffic.

transit : A transit AS has connections to more than one other AS and will carry transit traffic. The AS may impose policy restrictions on what transit traffic will be carried.

AS number : A 16-bit number uniquely identifying an AS. This is the same AS number used by BGP and EGP.

AS path : A list of all of the AS numbers traversed by a route when exchanging routing information. Rather than exchanging simple metric counts, BGP-3 communicates entire paths to its neighbors.

References and Bibliography

Books, Publications and Papers

- 📖 David E. McDysan and Darren L. Spohn, **Hands-On ATM**, McGraw-Hill, 1998
- 📖 Koichi Asatani et al, **Introduction to ATM networks and B-ISDN**, John Wiley & Sons, 1997
- 📖 Uyles Black, **Internetworking with ATM**, Prentice Hall, 1995
- 📖 Andrew S. Tanenbaum, **Computer Networks**, Prentice Hall, 1996
- 📖 Uyles Balck, **ATM: Foundation for Broadband Networks**, Prentice Hall, 1995
- 📖 William Stallings, **ISDN and Broadband ISDN with Frame Relay and ATM**, Prentice Hall, 1995
- 📖 Byeong Gi Lee, Minho Kang, Jonghee Lee, **Broadband Telecommunications Technology**, Artech House, 1996
- 📖 Stephen Saunders, **Glgabit Ethernet Handbook**, McGraw-Hill, 1998
- 📖 John Atkins, Mark Norris, **Total Area Networking**, John Wiley & Sons, 1996
- 📖 I.J. Duffy Hines, **ATM - The key to High-speed Broadband Networking**, M&T Books, 1996
- 📖 Udo W. Pooch, Denis Machuel, John McCahn, **Telecommunications and Networking**, CRC Press, 1991
- 📖 Daniel Minoli, **Enterprise Networking**, Artech House, 1993
- 📖 Thomas M. Chen, Stephen S. Liu, **ATM Switching Systems**, Artech House, 1995
- 📖 Raif. O Onvural, **Asynchronous Transfer Mode Networks - Performance Issues**, Artech House, 1995
- 📖 George C. Sackett, Christopher Y. Metz, **ATM and Multiprotocol Networking**, McGraw-Hill, 1996
- 📖 Daniel Minoli, Emma Minoli, **Delivering Voice over Frame Relay and ATM**, John Wiley & Sons, 1998
- 📖 Jim Metzler, Lynn DeNoia, **Layer 3 Switching**, Prentice Hall, 1999
- 📖 Mohammad A Rahman, **ATM Systems and Technology**, Artech House, 1998
- 📖 James Martin, **Asynchronous Transfer Mode – Architecture and Implementation**, Prentice Hall, 1997
- 📖 William Stalling, **ISDN and B-ISDN with Frame Relay and ATM**, Prentice Hall, 1995
- 📖 Stephen A. Thomas, **IPng and the TCP/IP Protocols**, John Wiley & Sons, Inc., 1998
- 📖 John T. Moy, **OSPF - Anatomy of an Internet Routing Protocol**, Addison-Wesley, 1998
- 📖 Merilee Ford, **Internetworking Technologies Handbook**, Cisco Press, 1997
- 📖 David M.Piscitello, A.Lyman Chapin, **Open Systems Networking - TCP/IP and OSI**, Addison-Wesley, 1993
- 📖 James Martin, **TCP/IP Networking**, Prentice Hall, 1994
- 📖 Todd Lammle, **TCP/IP for NT Server 4**, Sybex Press, 1997
- 📖 K.Washburn, J.T.Evans, **TCP/IP - Running a successful Network**, Addison-Wesley, 1993
- 📖 Floyd Wilder, **A guide to the TCP/IP Protocol suite**, Artech House, 1993
- 📖 Dr.Sidnie Feit, **TCP/IP - Architecture, Protocols & Implementation**, McGraw-Hill, 1997
- 📖 Timothy Parker et al, **TCP/IP Unleashed**, SAMS Publishing, 1996
- 📖 Douglas E. Comer, **Computer Networks and Internets**, Prentice Hall, 1997
- 📖 Andrew S. Tanenbaum, **Computer Networks**, Prentice Hall, 1996
- 📖 Fred HalSall, **Data Communications, Computer Networks and Open Systems**, Addison Wesley, 1993
- 📖 Masaka Ohta, **IETF and Internet Standards, IEEE Communications**, September 1998
- 📖 Richard R Parry, **Mobility and the Internet**, IEEE Potentials, May 1998
- 📖 David Lee , Daniel L.Lough, **The Internet Protocol version 6**, IEEE Potentials, May 1998
- 📖 David Lee, Danial L Lough, Scott Midkoff, **The next Generation of the Internet: Aspects of the IPv6**, IEEE Networks, February 1998
- 📖 William Stalling, **IPv6:The New Internet Protocol**, IEEE Communications, July 1996
- 📖 George Xylomenos, George Polyzos, **IP Multicast for Mobile Hosts**, IEEE Communications, January 1997






Universal Resource Locators

- 🌐 Assorted experiments with TCP/IP over ATM, <http://www.data.slu.se/robert/fasl.rapport.html>
- 🌐 ATM Boundary switching, <http://www.tepikom.ru/atm/ipoa.html#topp>
- 🌐 ATM Network Group Conference Publications, <http://www.ee.ust.hk/~ustatm/conference.html>
- 🌐 IP over ATM (ipatm) Working Group, <http://www.com21.com/pages/ietf.html>
- 🌐 Detailed Behavior of TCP over ATM, <http://www.eecs.harvard.edu/~rtm/tcp-atm/tcp-atm-1.html>
- 🌐 TCP/IP Throughput Performance Evaluation for ATM Local Area Networks, <http://www.ee.surrey.ac.uk/Personal/I.Andrikopoulos/Papers/ifip96.html>
- 🌐 VBNS initiative, <http://www.vbns.net>
- 🌐 Internet 2 initiative, <http://www.internet2.edu>
- 🌐 Abilene Project, <http://www.ucaid.edu/abilene/home.html>
- 🌐 Sprint's ION service, <http://www.sprint.com/Stemp/press>
- 🌐 University of Rochester, <http://www.utd.rochester.edu>
- 🌐 ATM Tutorial, <http://juggler.lanl.gov/lanp/atm.tutorial.html>
- 🌐 Cisco Connection Online, Internetworking Guide
- 🌐 ATM Glossary, http://www.data.com/Tutorials/ATM_Glossary
- 🌐 ATM Primer, http://www.npac.syr.edu/users/dpk/ATM_Knowledgebase/getting_started.html
- 🌐 ATM Basics, <http://cne.gmu.edu/~sreddiva/bconcepts.html>
- 🌐 Datacommunication tutorials, <http://www.data.com/Tutorials>
- 🌐 ATM standards, <http://cne.gmu.edu/~sreddiva/ATMstand.html>
- 🌐 ATM Tutorial, http://www.npac.syr.edu/users/mahesh/homepage/atm_tutorial
- 🌐 ATM Switching, <http://cne.gmu.edu/~sreddiva/ATMswitch.html>
- 🌐 ATM Standard & Interoperability, http://www.npac.syr.edu/users/dpk/ATM_Knowledgebase
- 🌐 UNI & protocols, <http://cne.gmu.edu/~sreddiva/Unip.html>
- 🌐 ATM Networks, <http://cne.gmu.edu/~sreddiva/ATMnet.html>
- 🌐 ATM Forum, <http://www.atmforum.com>
- 🌐 What is ATM, <http://www.whatis.com/atm.htm>
- 🌐 ATM cell switching technologies, <http://www.sims.berkeley.edu/resources/infoecon/FAQs>
- 🌐 Gigabit Networking, <http://www.arl.wustl.edu/~jst/gigatech/kits.html>
- 🌐 ATM interface specifications, <http://cell-relay.indiana.edu/~jst/atmforum/ps.html>
- 🌐 ATM Multiservices, <http://www.telecoms-mag.com/marketing/articles/nov97/rehana.html>
- 🌐 ATM Quality of Service, <http://www.telecoms-mag.com/marketing/articles/feb97/bennet.html>
- 🌐 IP Switching, <http://www.telecoms-mag.com/marketing/articles/sep97/cattell.html>
- 🌐 ATM promises, <http://www.telecoms-mag.com/marketing/articles/jan97/cooper.html>
- 🌐 ATM networks, <http://www.telecoms-mag.com/marketing/articles/may97/holtz.html>
- 🌐 ATM in service markets, <http://www.telecoms-mag.com/marketing/articles/mar98/nelsen.html>
- 🌐 TCP/IP Networking, <http://www.lmu.edu/admin/IS/training/protected/tcpip.html>
- 🌐 Intranets and Virtual Private Networks, <http://www.riscpa.org/seminar/intvnpn.htm>
- 🌐 Mbone over ATM, <http://alpha.ps.iit.nrc.ca/english/mbone-atm.html>
- 🌐 I-PNNI Accepted As Work Effort , <http://www.baynetworks.com/News/Press/9604232.html>
- 🌐 PNNI White paper, <http://www.cabletron.com/white-papers/pnni-1/>

Request for Comments

- 📄 RFC 1577 - Classical IP and ARP over ATM
- 📄 RFC 1483 - Multiprotocol Encapsulation over ATM Adaptation Layer 5
- 📄 RFC 1680 - IPng Support for ATM Services
- 📄 RFC 1754 - Recommendations for the ATM Forum's Multiprotocol BOF
- 📄 RFC 1755 - ATM Signaling Support for IP over ATM
- 📄 RFC 1821 - Integration of Real-time Services in an IP-ATM Network Architecture
- 📄 RFC 1926 - An Experimental Encapsulation of IP Datagrams on Top of ATM
- 📄 RFC 1932 - IP over ATM: A Framework Document
- 📄 RFC 2022 - Support for Multicast over UNI 3.0/3.1 based ATM Networks.
- 📄 RFC 2225 - Classical IP and ARP over ATM

- ☞ RFC 2226 - IP Broadcast over ATM Networks
- ☞ RFC 2269 - Using the MARS Model in non-ATM NBMA Networks
- ☞ RFC 2331 - ATM Signalling Support for IP over ATM - UNI Signalling 4.0 Update
- ☞ RFC 2336 - Classical IP and ARP over ATM to NHRP Transition
- ☞ RFC 2337 - Intra-LIS IP multicast among routers over ATM using Sparse Mode PIM
- ☞ RFC 2364 - PPP Over AAL5
- ☞ RFC 2379 - RSVP over ATM Implementation Guidelines
- ☞ RFC 2380 - RSVP over ATM Implementation Requirements
- ☞ RFC 2381 - Interoperation of Controlled-Load Service and Guaranteed Service with ATM
- ☞ RFC 2382 - A Framework for Integrated Services and RSVP over ATM
- ☞ RFC 2383 - ST2+ over ATM Protocol Specification - UNI 3.1 Version
- ☞ RFC 2386 - A Framework for QoS-based Routing in the Internet
- ☞ RFC 0791 - Internet Protocol
- ☞ RFC 0950 - IP Subnet extensions
- ☞ RFC 0919 - IP Broadcast Datagrams
- ☞ RFC 0922 - IP Broadcast Datagrams with subnets
- ☞ RFC 1042 - Standard for the Transmission of IP Datagrams over IEEE 802 Networks
- ☞ RFC 0792 - Internet Control Message Protocol
- ☞ RFC 1256 - ICMP Router Discovery Messages
- ☞ RFC 1112 - Internet Group Multicast Protocol
- ☞ RFC 0768 - User Datagram Protocol
- ☞ RFC 0793 - Transmission Control Protocol
- ☞ RFC 0896 - Congestion Control in IP/TCP networks
- ☞ RFC 0826 - Address Resolution Protocol
- ☞ RFC 0903 - Reverse Address resolution Protocol
- ☞ RFC 1032 - Domain Administrator's Guide
- ☞ RFC 1033 - Domain Administrator Operations Guide
- ☞ RFC 1034 - Domain Names - Concepts and Facilities
- ☞ RFC 1035 - Domain Names - Implementation and Specification
- ☞ RFC 1180 - TCP/IP Tutorial
- ☞ RFC 1467 - Status of CIDR Deployment in the Internet
- ☞ RFC 1517 - Applicability Statement for Classless Inter-Domain Routing (CIDR)
- ☞ RFC 1518 - An Architecture for IP Address Allocation with CIDR
- ☞ RFC 1519 - Classless Inter-Domain Routing (CIDR): Address Assignment/Aggregation Strategy
- ☞ RFC 1520 - Exchanging Routing Information Across Provider Boundaries in the CIDR Environment
- ☞ RFC 0904 - Exterior Gateway Protocol - Specification
- ☞ RFC 1074 - The NSFNET Backbone SPF Based Interior Gateway Protocol.
- ☞ RFC 1092 - EGP and Policy Based Routing in the New NSFNET Backbone.
- ☞ RFC 1093 - The NSFNET Routing Architecture.
- ☞ RFC 1104 - Models of Policy Based Routing.
- ☞ RFC 1133 - Routing between the NSFNET and the DDN.
- ☞ RFC 1222 - Advancing the NSFNET Routing Architecture.
- ☞ RFC 0891 - DCN Local-Network Protocols
- ☞ RFC 1058 - Routing Information Protocol
- ☞ RFC 1076 - Distance Vector Multicast Routing Protocol
- ☞ RFC 1721 - RIP Version 2 Protocol Analysis
- ☞ RFC 1722 - RIP Version 2 Protocol Applicability Statement
- ☞ RFC 1723 - RIP Version 2 - Carrying Additional Information
- ☞ RFC 1724 - RIP Version 2 MIB Extension
- ☞ RFC 1583 - OSPF Version 2
- ☞ RFC 1245 - OSPF Protocol Analysis
- ☞ RFC 1246 - Experience with the OSPF Protocol
- ☞ RFC 1247 - Open Shortest Path First Version 2
- ☞ RFC 1253 - OSPF Version 2: Management Information Base
- ☞ RFC 1370 - Applicability Statement for OSPF
- ☞ RFC 1583 - OSPF Version 2
- ☞ RFC 1195 - Integrated IS-IS
- ☞ RFC 1265 - BGP Protocol Analysis

-  RFC 1266 - Experience with the BGP protocol
-  RFC 1267 - A Border Gateway Protocol 3 (BGP-3)
-  RFC 1268 - Application of the Border Gateway Protocol in the Internet
-  RFC 1654 - A Border Gateway Protocol 4 (BGP-4)
-  RFC 1655 - Application of the Border Gateway Protocol in the Intern

