HF-DSR

An Implementation of Dynamic Source Routing Designed for HF Ad-Hoc Networks

RIT Computer Science
Masters Project Proposal

Prepared by Michael Stringer
May 24, 2004

Approvals:

Chair _____

Reader _____

Observer_____

Summary

The purpose of this project shall be to implement a level 3 data communications protocol called HF-DSR which shall incorporate major elements of the Directed Source Routing (DSR) ad-hoc routing algorithm [Johnson+, 2001]. This protocol will be designed to operate efficiently over a High Frequency (HF) radio network with transient node participation. The deliverables of this project shall include a detailed protocol specification, PC-based protocol implementation, and test data drawn from a "bench top" test setup incorporating a small number of network nodes operating both in ideal conditions and with controlled radio interference. Additional "real world" testing is beyond the scope of this project due to a lack of testing resources.

DSR features that shall be designed and implemented by HF-DSR include route discovery, route maintenance, and automatic route shortening. The implementation shall be targeted toward a desktop or laptop Microsoft Windows XP Professional platform and shall use a synchronous RS-232 connection to an HF modem.

An HF radio oriented level 2 protocol shall be used by HF-DSR in order to transfer data between single hops of a network. The data link protocol used will be the Harris RF Communications implementation of NATO Standard Agreement (STANAG) 5066, an efficient node-to-node data delivery protocol optimized for HF conditions.

Overview

Ad-hoc networks differentiate themselves from traditional networks by incorporating specialized routing algorithms that allow network nodes to transiently participate as major contributors to network operation. In contrast with traditional wired networks, ad-hoc networks require additional control information in order to pool bandwidth and connectivity among members of the network. The associated advantage of this additional control information is that network administrators are absolved of responsibility to manually maintain ad-hoc networks in the presence of major network topology changes. [Johnson+, 2001]

Protocols designed with wired networks in mind, such as TCP/IP, are quite conducive to operation over relatively static networking infrastructures. Additionally, address discovery protocols such as Dynamic Host Configuration Protocol (DHCP) [DHCP] for Ethernet and Internet Protocol Control Protocol (IPCP) [IPCP] for PPP allow TCP/IP network adapters to acquire addresses in the subnetwork they are connected to at a particular time. These address discovery protocols allow some degree of network mobility within TCP/IP networks. Routing protocols such as Routing Information Protocol (RIP) [RIP] allow routing tables to change among existing routers. However, TCP/IP does not allow address discovery and dynamic routing protocols to be combined into a mechanism where networks may be dynamically augmented by fully functional routers and continue to operate efficiently. In contrast, one common property of ad-hoc network protocols is that some mechanism exists to propagate addressing and routing information throughout level 3 networks.

Ad-hoc networks also do not assign any static routing role to any node in the network. In contrast to TCP/IP where distinct nodes are tasked with being routers and gateways, ad-hoc networks require that all nodes be minimally capable of performing some routing function on demand. Typically the minimal roles of an ad-hoc network node are to forward data packets intended for some other node and to pass addressing information throughout the network.

There exists considerable discussion concerning ad-hoc networking over wireless media, but the main focus of these concern relatively high-performance "line of sight" networks. These types of networks are fast, typically able to support network speeds of 16 kilobits per second and higher. “Line of sight” networks normally have low latency, which allows them to implement channel access using channel partitioning methods such as TDMA (Time Division Multiple Access) [TDMA] or CDMA (Code Division Multiple Access) [CDMA]. These have the advantage of providing constant bit-rate channels to each network participant. However, these mechanisms depend on the available channel bandwidth to be sufficiently large to divide it among all network participants. Relatively low-bandwidth, low-speed, high-latency channels such as 3 kilohertz HF radio channels are not given the same discussion priority as high-bandwidth, high-speed, low-latency networks.

HF radio channels possess certain properties that are conducive to ad-hoc network operation. These radio waves propagate extremely well in two conditions. HF radio deployments at sea exhibit “ground wave” propagation in which the HF radio waves follow the surface of the water over the horizon while maintaining consistently high signal quality. In this manner, ships within a fleet can communicate effectively without any infrastructure. Additionally, HF radio waves exhibit “sky wave” propagation in which the waves bounce off the Earth's ionosphere and surface at obtuse angles. “Sky wave” propagation allows land or sea based deployments to transmit over thousands of miles. However, “sky wave” channels are of much lower quality than “ground wave” channels and are subject to greater interference due to changing atmospheric conditions and solar activity.

Data communication across an HF radio channel is accomplished by means of an HF modem. HF modems can implement several distinct waveforms. The more advanced waveforms, such as MIL-STD-188-110B, allow bit rates up to 12800 bits per second for certain environments. Additionally, the waveforms can provide data interleaving, which assists in error correction but adds up to 9.6 seconds to every transmission. Due to the speed limitations and latency issues, protocols that are optimized for HF traffic exhibit relatively few long transmissions that contain a low overhead-to-data ratio. Similarly, media access control must avoid extraneous transmissions.

The Dynamic Source Routing (DSR) algorithm described by Johnson and Maltz [Johnson+, 2001] provides some features of ad-hoc network routing that scale well to operation over HF radio networks. DSR primarily sends network control information on demand. Nodes do not initiate route discovery to build up routes that may not be used. This is appropriate in HF networks since bandwidth is extremely limited. In addition,

DSR accommodates unidirectional connections between nodes. "Sky wave" propagation in HF radio networks often delivers much higher quality signals in one direction than in another, creating unidirectional links.

The route discovery process in DSR involves the source node broadcasting a route discovery request, which gets selectively rebroadcast by multiple nodes until the desired destination node receives the route request. The destination node then performs an analogous route request process to discover a route to the source node. Although mechanisms exist to limit the scope of total packet transmissions to a maximum of twice the number of nodes in the network, it is likely that several nodes will transmit at the same time, thus causing interference. A channel access scheme such as CSMA/CA should be used to avoid collisions in situations when contention is expected.

Functional Specification

Channel Access

Resolving contention between multiple nodes on an HF channel can be particularly problematic due to the high latency associated with the medium. If a node on an HF network transmits data at an interleaver setting that causes an interleaver delay of 2n seconds, then carrier detect will not be asserted from the HF modem to the PC on any receiver until n seconds have elapsed. As a result, there will be contention on an HF channel if multiple nodes begin transmitting within half of an interleaver delay of one another. These characteristics must be accommodated by ensuring that multiple stations with simultaneous outgoing data defer transmission until their turn to send. This can be accomplished with slot-based or token passing schemes.

TDMA is able to overcome this problem at the expense of both effective channel throughput and additional control information. Each node is given a specific time slice that allows it to begin a transmission uninterrupted. However, time slices remain allocated for all nodes regardless of whether they have data to transmit. As a result, channel throughput decreases proportionally to the number of nodes in the network. Additionally, time slices must be dynamically assigned to nodes entering the network and unassigned from nodes leaving the network. Additional control information is required to perform this negotiation, which further decreases throughput and introduces a point of failure.

Token passing schemes can resolve channel access problems, but they introduce the well-known "lost token" problem. This problem is exacerbated by the transient nature of the network that consistently gains and loses nodes. Token loss is rare on wired networks due to their underlying reliability. However, the inherent variable quality of HF channels (especially "sky wave" networks) would cause token loss to occur much more often.

CSMA/CA operates by waiting until an ongoing transmission ends, and then randomly choosing a time slice to begin sending its data. One advantage of CSMA/CA is that minimal control information is transmitted between nodes, which is a critical consideration in low bandwidth networks. Additionally, the presence of higher level

protocol traffic (such as HF-DSR) causes nodes to switch into a monitoring state. Monitoring nodes defer their outgoing data until such time as monitored data ceases.

One disadvantage of this scheme is that some nodes will transmit during the same time slice. This occurs because individual nodes do not coordinate ownership of particular time slices. These collisions must be resolved by higher level protocols. [Holcomb+, 2003]

HF-DSR shall use a form of CSMA/CA because its lack of coordinated control schemes is ideal for an ad-hoc network. This mechanism will be augmented by an encoded End of Transfer (EOT) timer that allows monitoring nodes to determine approximately how long a data transfer has before completion. In this manner, monitoring nodes need only receive one value of the EOT timer to determine when they may attempt to acquire the HF channel. [STANAG 5066]

Data Link Layer

HF-DSR requires its underlying level 2 protocol to implement specific features. To ensure basic data consistency over variable HF channel conditions, level 3 data transfers will be divided into one or more data segments. Each data segment gets transmitted as a level 2 protocol packet. A verification method such as Cyclic Redundancy Check (CRC) must validate received packets. Additionally, an Automatic Repeat Request (ARQ) method must exist for correcting corrupted or missing over-the-air data. The size and relative location of segments within the level 3 data transfer must be indicated within level 2 packets. Finally, the level 2 protocol must uniquely identify the participants in a transfer by node or adapter identifier. [STANAG 5066]

Level 2 ARQ functionality must be handled with care when considered for a level 3 protocol specifically designed to support asymmetric wireless transfer media. ARQ control packets are typically much smaller than higher level data payloads. As HF modems often have a minimum transmission time (interleaver delay), the data rate of a modem sending an ACK packet may be lowered so that the entire minimum transfer time is used to send the ACK packet. The effect of lowering the data rate of a modem is to improve tolerance of poor channel conditions. As a result, ACK control packets sent using this mechanism will display superior error tolerance than data being transferred in the forward direction.

STANAG 5066

NATO STANAG 5066 is a level 2 protocol that implements the abovementioned features. Additionally, Harris RF Communications' implementation of STANAG 5066 uses CSMA/CA for channel access augmented by an EOT timer. Finally, the Harris implementation of STANAG 5066 allows both ARQ and non-ARQ functionality at the data link layer.

Harris RF Communications has allowed this project to use its STANAG 5066 protocol implementation. This protocol implementation is a feature included in their RF-6710W

Wireless Messaging Terminal.

STANAG 5066 is a data link protocol architecture that contains three primary subsystems. The DTS (Data Transfer Sublayer) contains a low level ARQ engine and is primarily responsible for assembling complete C_PDUs (Channel Access Sublayer Protocol Data Units) from segmented packets or D_PDUs (Data Transfer Sublayer Protocol Data Units). Conversely, outgoing C_PDUs are segmented into one or more D_PDUs. The CAS (Channel Access Sublayer) implements channel access and is responsible for establishing, breaking, and monitoring node-to-node connections throughout the network. The SIS (Subnetwork Interface Sublayer) provides an interface between the CAS and an external TCP/IP service that can exchange S_PRIMITIVES with a STANAG 5066 client. [STANAG 5066]

HF-DSR shall be implemented as a STANAG 5066 client that exchanges S_PRIMITIVES with the SIS service exposed on the Harris RF Communications RF-6710W software package.

Route Discovery

The remainder of HF-DSR's algorithms shall reside within a level 3 protocol. The general form of a level 3 transfer shall be defined as one or more level 2 transfers beginning at node N_s, ending at node N_d, and proceeding through zero or more intermediate nodes N_i ($s < i < d$). In order to minimize channel access contention, the complete data of a level 3 transfer T shall be transferred from N_n to N_{n+1} before any data from T gets transferred to N_{n+2}. Once routes are established for T, only nodes N_n and N_{n+1} are emitting radio transmissions concerning T at any one time (n will vary during the course of the level 3 transfer). This technique also assists route maintenance as described below.

In order to transfer data from N_s to N_d, a path must first be discovered between the two nodes. HF-DSR inherits a path discovery algorithm from DSR [Johnson+, 2001]. The operation of the algorithm is summarized as follows:

1. Node N_s broadcasts a specific message type denoted as “Path Request”. The “Path Request” consists of a pseudo-unique numerical identifier and a network address list of nodes that have previously (re)transmitted this “Path Request”. Initially, the list contains only the network address of N_s.
2. Node N_{s+i} ($0 < i < d$) receives the “Path Request”. If N_{s+i} has already received a “Path Request” starting with the network address of N_s and the same pseudo-unique identifier, no further action is taken. Otherwise, the “Path Request” message is rebroadcast with the same identifier and node N_{s+i} appended to the network address list.
3. Node N_d receives the “Path Request” message. At this point, the network address list contains a path from N_s to N_d (the “Forward Path”). N_d broadcasts a “Reverse Path Request” message. The “Reverse Path Request” message consists of a pseudo-unique numerical identifier (not necessarily the one from the “Path Request”), a network address list of nodes that have previously (re)transmitted this “Reverse Path Request”,

and the "Forward Path".

4. Node N_{s+i} $(0 < i < d)$ receives the "Reverse Path Request". If N_{s+i} has already received a "Reverse Path Request" starting with the network address of N_s and the same pseudo-unique identifier, no further action is taken. Otherwise, the "Reverse Path Request" message is rebroadcast with the same identifier and node N_{s+i} appended to the network address list.
5. Node N_s receives the "Path Request" message. At this point, the network address list contains a path from N_d to N_s (the "Reverse Path"). Additionally, N_s has a "Forward Path". N_s is now capable of sending data to N_d along the "Forward Path".

Data Transfer

Once the path discovery algorithm is complete, data transfers may begin between N_s and N_d. Aside from the data payload, additional information shall be sent from N_s to N_d during a level 3 transfer. The "Forward Path" shall be sent. This is used by each node N_{s+i} $(0 \leq i < d)$ between N_s and N_d to determine N_{i+1} for each N_i. Also, the "Reverse Path" shall be sent. This is used by N_d to format a final acknowledgment of success ("Final ACK") and direct this message toward N_s. Finally, a "Transfer Id" shall be sent. This will be used to associate the data transfer with its "Final ACK".

The "Final ACK" shall be a small message that N_d sends to N_s to confirm receipt of a data transfer. A "Final ACK" shall contain the "Reverse Path", which will be used by each node N_{s+i} $(0 < i \leq d)$ between N_s and N_d to determine N_{i-1} for each N_i. The "Final ACK" shall also contain the "Forward Path" so that N_s can observe if it differs from the initial "Forward Path" sent with the data transfer. The "Forward Path" received in the "Final ACK" can differ due to applications of the route maintenance algorithm described below. Finally, the "Final ACK" shall contain the "Transfer Id" in order that node N_s can associate the "Final ACK" with its outgoing message.

If the underlying level 2 protocol is using an ARQ mechanism between adjacent nodes in the "Forward Path" and "Reverse Path", then no additional mechanism is necessary to suggest successful transfer between these nodes. If a level 2 ARQ system is unavailable or undesirable, then other algorithms may be used to verify transfer to a neighboring node:

- Pure non-ARQ transmission. Each node sends its data to its neighbor and declares its transfer complete. Broken routes can only be detected if the source node N_s uses timer expiration to treat a transfer as unsuccessful.
- Passive acknowledgment [Jubin+ 1987]. Node N_i confirms transfer with N_{i+1} by overhearing transfer from N_{i+1} to N_{i+2}. This method depends on symmetrical communication between N_i and N_{i+1}.
- Explicit level 3 acknowledgment [Johnson+, 2001]. Node N_{i+1} sends an acknowledgment to N_i using the same path discovery algorithm as N_s and N_d use to determine the "Forward Path" and the "Reverse Path".

These final three methods can be characterized by an increasing direct correspondence between network bandwidth usage and transmission reliability. HF-DSR shall implement

the first two methods as bandwidth is at a premium on HF networks.

Route Maintenance

Route lifetime shall directly correspond to the length of time that routes are useful. A "Forward Path" route shall be declared broken on a when node N_i cannot forward data to node N_{i+1} ($s \leq i \leq d$). In this scenario, N_i shall initiate the route discovery algorithm to generate a route between N_i and N_d. The "Reverse Path" shall stay unmodified. Additionally, the "Forward Path" shall be modified to incorporate the nodes between N_i and N_d.

While sending the "Final ACK", the "Reverse Path" can become broken in the same manner as the "Forward Path". When node N_i cannot forward data to node N_{i-1} ($s \leq i \leq d$), the "Reverse Path" shall be declared broken. In this scenario, N_i shall initiate the route discovery algorithm to generate a route between N_i and N_s.

This scheme is possible only due to the requirement that all data in a transfer is delivered from N_n to N_{n+1} before any data gets transferred to N_{n+2}. If this were not the case, then the path could not be modified without notifying node N_s. DSR presents a route maintenance scheme with significantly more transfer overhead.

Automatic Route Shortening

Step 2 of the above path discovery process shall be augmented to facilitate automatic route shortening. If node N_i receives a "Path Request" message containing node addresses from N_s to N_{i-1}, it has the option of replacing any subset N_j through N_k (inclusive) in the message's network address list ($s < j < k < i$). The replacement shall occur when there is a shorter subset of nodes linking N_{j-1} and N_{k+1} within one of the other routes in which N_i is a participant.

To avoid replacing N_j through N_k with a set of nodes that are no longer connected, node N_i shall apply criteria to its existing routes before performing a replacement. These criteria may include the timestamp of last successful transfer using the route, the length of time the route has been in existence, or other criteria.

Functional Specification

High-Level Architecture

HF-DSR shall be implemented as a STANAG 5066 client. It shall exchange information with implementations of STANAG 5066 by making a TCP/IP connection over the well-known TCP/IP port 5066 [IANA]. Communication between HF-DSR and STANAG 5066 shall occur using instances of S_PRIMITIVE [STANAG 5066].

HF-DSR performance data will be collected by establishing a controlled collection of nodes in a laboratory environment. Each node will consist of a laptop or desktop

computer running Microsoft Windows XP Professional, a Harris RF-5710 or RF-5710A HF modem, and a Harris PRC-138, Harris RF-5022, or RF-5022E radio. Each node shall run the Harris RF-6710W Wireless Messaging Terminal and the HF-DSR implementation.

Equipment connectivity shall be as follows:

- Every computer shall run the following software:
  - One instance of HF-DSR
  - One instance of Harris RF-6710W Wireless messaging terminal..
- Every instance of HF-DSR shall connect to an instance of RF-6710W by using a TCP/IP connection to "localhost".
- Every RF-6710W shall be connected by 25-pin RS-232 synchronous connection to an HF modem.
- Every HF modem shall be connected to the audio input of a radio.
- Every radio shall transfer data using low power transmissions of 1 watt or less.

Based on availability, a subset of the nodes may use HF channel simulators placed inline between the HF modems and the radios. HF channel simulators are audio signal processors that simulate atmospheric effects on modem audio output. In this manner, the collection of nodes on a laboratory bench can simulate "ground wave" and "sky wave" networks.

The precise quantity of computers, HF modems, and HF channel simulators available for data collection is uncertain. A minimum of 5 computers, 5 HF modems and two HF channel simulators will be available.

Detailed Architecture

HF-DSR shall be implemented using C# within Microsoft Visual Studio 2003 .NET.

Separate threads shall be used to handle socket transmission, socket reception, user interface, and core HF-DSR functionality. In this matter, communication latency will be kept to a minimum.

User interface functionality will be implemented by a dialog box. Local and remote address information and message content may be set from this dialog. Detailed protocol status such as current timer values, stored routes, and route participation will be shown.

HF-DSR shall also produce a log file that indicates all significant protocol events. Events logged will include discovery of successful and unsuccessful message failures, route establishment, route deletion, route modification, and connectivity status involving STANAG 5066.

HF-DSR shall use a C# delegate-based state machine to associate states with actions. Certain user inputs and network events will cause state transitions. Example states may include "IDLE", "MESSAGE_IN_PROGRESS", and "NO_CONNECTION_TO_5066".

Data Collection and Analysis

Initially, a performance baseline will be taken with no channel simulation applied. Total end-to-end HF-DSR message throughput will be measured with various amounts of network load and differently-sized networks. Once these measurements are complete, simulated interference can be gradually applied. Additional throughput measurements will be taken at different interference levels. Throughput vs. interference with differently sized networks and different network saturation will be plotted and analyzed.

Analysis will consist of identifying trends that cause significant protocol behavior. Such trends include absolute throughput variance and radio of HF-DSR control data transmitted to total data transmitted. Once trends are isolated, modifications will be suggested to HF-DSR to minimize negative trends and maximize positive trends.

Expectations

High performance of the HF-DSR network will be directly proportional to the stability of network routes throughout message transfer. Route discovery will require a significant amount of control information to be exchanged. Fewer changes in network topology will require fewer invocations of route discovery, whether in the form of initial route discovery or route maintenance.

One invocation of route discovery can cause a maximum of 2n-2 transmissions in an n-node network. The effect will be to cause immediate contention on the underlying channel that must by resolved by lower level channel access. This network saturation spike will cause queued messages to be delayed. Additionally, many of the transmissions during route discovery will be redundant since only one route needs to be discovered.

It may be preferable to delay rebroadcasting “Path Request” and “Reverse Path Request” messages for a time inversely proportional to the length of the currently accumulated path. In this manner, the path discovery process will be converted into a depth first search instead of a breadth first search. While this method is less likely to discover the optimum path, it will allow both a “Forward Path” and a “Reverse Path” to be found much more quickly.

Data transfer will be very efficient under normal circumstances, particularly due to the lack of additional control information once routes are discovered. As a result, relatively stable network topologies will produce excellent network performance.

Deliverables

Report

Deliverables shall include a written report documenting the basis and the progress of the HF-DSR effort. A preliminary Table of Contents for the report follows:

I. Introduction
II. System Overview
III.HF-DSR Protocol Specification
IV.Data Gathering
V. Data Analysis
VI.Conclusion

Source Code

Deliverables shall include source code for the HF-DSR executable developed for the purposes of data collection. The source code shall be written in C#. Project files shall be included to reproduce an executable on Microsoft Visual Studio 2003 .NET.

Schedule

May 16th, 2004 through May 29th, 2004 (two weeks): Detailed protocol traffic design documentation will be written. This documentation shall include packet formatting details and high-level protocol state machine drawings.

May 30th, 2004 through June 26th, 2004 (four weeks): First functional implementation of HF-DSR will be implemented. Minimally, route discovery will be implemented. At this point, laboratory testing will begin.

June 27th, 2004 through August 7th, 2004 (six weeks): HF-DSR will be made fully functional during this period. Full functionality is specified in this document. Additional requirements discovered during laboratory testing may be incorporated into HF-DSR design.

August 8th, 2004 through September 25th,2004 (six weeks): Data gathering and analysis will occur during this time. Experimental design changes may be introduced to improve performance.

September 26th, 2004 through October 23th, 2004 (four weeks): A final written report will be composed during this period. The written report shall be formatted as shown in the "Deliverables" section.

October 24th, 2004 through November 6th, 2004 (two weeks): The project defense shall occur during this period.

Current Status

Project prototyping has begun. The prototype can currently bind and unbind from a local STANAG 5066 implementation. The prototype also suggests an architecture for the full HF-DSR implementation.

References

[CDMA] CDG: Technology: Welcome to the World of CDMA. (1999). Retrieved May 24, 2004 from http://www.cdg.org/technology/cdma%5Ftechnology/a%5Fross/cdmarevolution.asp.

[DHCP] Resources for DHCP. (2003, November 22). Retrived May 24, 2004, from http://www.dhcp.org

[Holcomb+, 2003] Holcomb, M.T. and Weston, J.H. Enhancing Channel Utilization and Performance of STANAG 5066. In Ninth International Conference on HF Radio Systems and Techniques. Institute of Electrical Engineers, London: 2003.

[IANA] Internet Assigned Numbers Authority. (2003, September 22). Retrieved February 2, 2004 , from http://www.iana.org/assignments/port-numbers

[IPCP] McGregor, G. RFC 1332 – The PPP Internet Protocol Control Protocol (IPCP). May 1992. Retrieved May 24, 2004 from http://www.faqs.org/rfcs/rfc1332.html

[Johnson+, 2001] David B. Johnson and David A. Maltz. DSR: The Dynamic Source Routing Protocol For Multihop Wireless Ad Hoc Networks. In Perkins, Charles. Ad Hoc Networking. Addison-Wesley, New York, 2001.

[Jubin+, 1987] J. Jubin and J.D. Tornow. The DARPA Packet Radio Network Protocols. Proceedings of the IEEE 75(1):21-32, January 1987.

[RIP] Malkin, G. RFC-1388 – RIP Version 2. January 1993. Retrieved May 24, 2004 from http://www.rfc-archive.org/getrfc.php?rfc=1388.

[STANAG 5066] NATO Standardization Agreement, "Profile for High Frequency (HF) Radio Data Communications STANAG 5066" Version 1.2

[TDMA] IEC: Time Division Multiple Access (TDMA). (2003). Retrieved May 24, 2004 from http://www.iec.org/online/tutorials/tdma/.