2005

QoS multi meshed tree routing in tethered MANET

Vishal Gogula

QoS Multi Meshed Tree Routing in Tethered MANET

Masters Project Proposal

Vishal Gogula
Department of Computer Science
Rochester Institute of Technology
vrg4975@cs.rit.edu

Project Chair
Dr. Hans-Peter Bischof
Department of Computer Science
Rochester Institute of Technology

Project Reader
Dr. Nirmala Shenoy
Department of Information Technology
Rochester Institute of Technology

Table of Contents

Abstract

The “QoS Multi Meshed Tree Routing in Tethered MANET” is designed to provide end-to-end quality of service (QoS) in terms of bandwidth in Tethered Mobile Ad hoc Networks (tMANET). This project is a part of the research project titled "Framework for Seamless Roaming, Handoff and QOS Mapping in Next Generation Networks” going on at the Laboratory for Applied Computing headed by faculty members Dr. Nirmala Shenoy and Prof Bruce Hartpence.

MANET is a wireless network comprising of a collection of mobile nodes that change their locations dynamically and organize themselves arbitrarily as a result the network’s topology keeps changing rapidly. In order to facilitate communication within the network a routing protocol is used to discover routes between nodes. Every node in an Ad hoc network can act as a router and help in routing process and forwarding of data in reaching the destination. The Multi-Meshed Tree (MMT) routing protocol has been developed to serve this purpose. It uses tree like-structure for forming routes between nodes and the branches (links) in the trees can mesh with each other. Every node in the network is assigned a unique ID for advertising, routing and forwarding purpose. Each node can have multiple IDs i.e. one of the node id is a primary node id used by the data flow in reaching the destination, while other node ids are secondary used in case the primary node id fails.

The increasing use of MANETs for transferring of multimedia applications such as video, voice and data leads to the need of QoS support. MANETs should provide quality of service (QoS) in terms of bandwidth for these kinds of transfers to ensure quality delivery. Apart from providing QoS this routing protocol should also take dynamic topology and shared wireless medium into consideration.

As MANETs are highly mobile networks, the nodes move in and out of the network on the fly. This project aims at designing a routing protocol QoS MMT (Multi Meshed Tree) which satisfies mobility and end-to-end QoS requirement in terms of Bandwidth. Our protocol handles the mobility aspect efficiently. This project involves performance study of QoS MMT with simulations using network simulation tool OPNET. The simulations are also used for studying the ‘Node-Joining Time’ and to show mobile nodes can join the network quickly.

The results from the simulations are extensively used to compare with another QoS supporting protocol AQOR in following performance metrics:

a. Throughput
b. End-to-End delay
c. Packet loss

In this routing protocol, the route discovery, bandwidth reservation and forwarding is done at the layer 2 data-link layer. This routing protocol includes all the features like route discovery, link failure, bandwidth calculation, resource reservation and resource release using control messages like advertisements, connect request, connect accept etc. The main aspect of this routing protocol is formation of routes based on the bandwidth information at each mobile node.

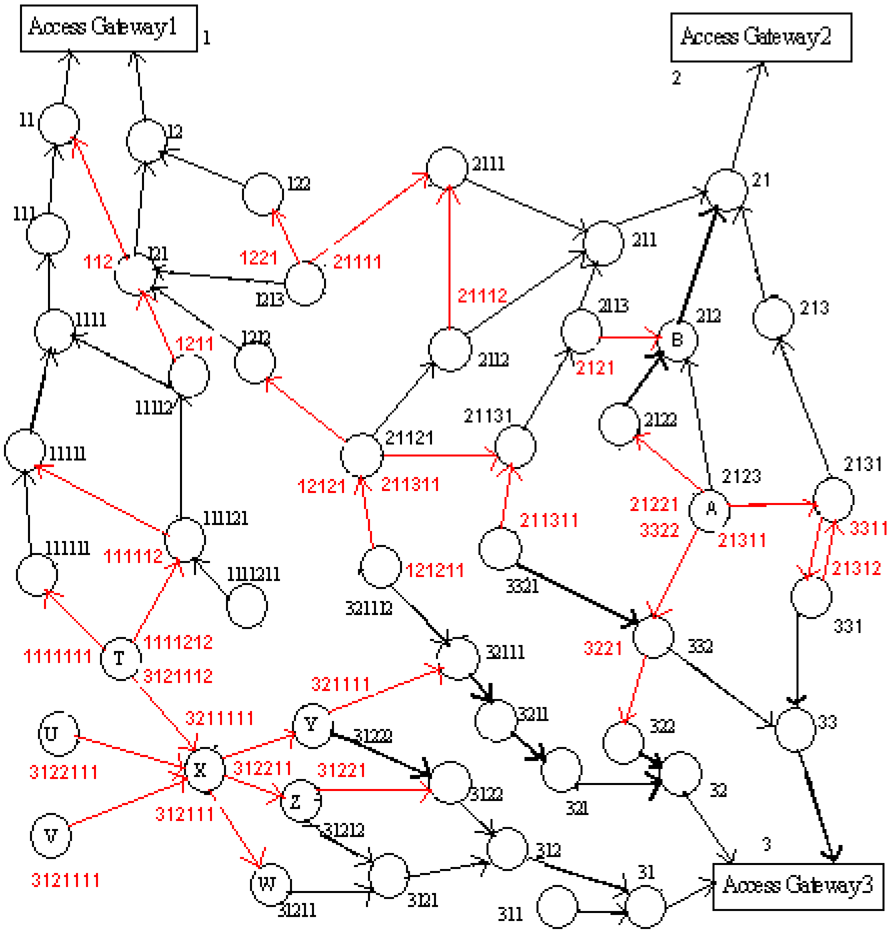
Overview

A tethered MANET can have multiple access gateways to provide access to the wired backbone or the outside world. A gateway acts both as a peer wireless node and as a bridge to the wired network. The below figure shows a tethered MANET comprising of two fixed access gateways. The route discovery or wireless network establishment starts from the Access Gateway using limited flooding. The gateway broadcasts control packets which includes the available Bandwidth $B_{available}$ called advertisements to its neighboring nodes one-hop away. Upon receiving the advertisement, a decision is made by the node whether to request for a connection if the requested Bandwidth $B_{requested}$ of the node is within the range of the broadcasted bandwidth $B_{available.}$ A connection is established between the Gateway and neighboring nodes after exchange of series of control messages and unique node IDs are assigned to each of the nodes. Now the nodes at the tier-1 which are connected to the Gateway starts broadcasting advertisements to its neighboring nodes advertising the $B_{available}$ and this processes continues until all the nodes in the vicinity requiring connection are assigned node IDs.

Multiple node IDs can be assigned to node shows the meshing of the branches of the trees indicating different route connections from a source node to the Gateway. One of the node ids is used as primary node id in forwarding data towards the destination. The requested bandwidth is reserved along this route for the data flow. The node can still keep listening to new 'advertisements' for new node ids but these ids are kept as secondary which are in case the primary node id fails. The secondary node ids doesn't request for bandwidth reservation when it sends out 'connect request' control message. If in case any of the primary node ids fails a secondary node id is picked by the node to be used as primary and a new 'connect request' message is sent out with a bandwidth reservation request.

Due to dynamic nature of the network the topology keeps changing rapidly. The protocol also supports link failures discovery and also features rapid recovery mechanisms.

In the below figure nodes A, X, T, Z have been assigned multiple node ids which allows a particular node to reach the destination via different routes in case of the primary route failure due mobile node movement. The node ids in red are the secondary node ids and the links in red are the secondary links. For instance let us suppose the node A has initially been assigned node id 2123 when it sends out 'connect request' to node B (212). Node A uses 2123 as primary node id to forward data via route 2123→212→21→2 towards the destination. But node A still keeps accepting 'advertisements' from other mobile nodes. After listening to couple of 'advertisements' and sending out 'connect request' messages without specifying Bandwidth reservation request node A is assigned node ids 21221, 3322, 21311 which are now secondary node ids. If in case node B (212) moves away as a result connection is lost and bandwidth recovery mechanism is triggered. Node A now chooses one of the secondary node ids as a primary node id and sends out a new 'connect request' and starts forwarding data using new route. Hence this protocol clearly supports quite QoS recovery in case of QoS route failure. A node can still join the topology though it has no data to forward. For example node X is assigned three node secondary ids 3211111, 312211, 312111 which can be used as primary node ids if the node X has data to forward later on.


Access Gateway1
1
Access Gateway2
2
Access Gateway3
3
11
12
111
122
121
112
1221
1213
21111
2111
21
211
21112
213
2113
212
1111
1211
1212
2112
B
2121
11112
2122
21131
11111
21121
2123
2131
12121
211311
A
21221
3311
3322
21311
111121
211311
111111
111112
1111211
121211
3321
21312
321112
331
1111212
T
1111111
3221
332
32111
3121112
321111
3211111
U
Y
3211
31222
322
33
3122111
X
312211
31221
Z
3122
32
V
312111
31212
321
312
3121111
W
31211
3121
311
31

Technical Specification

In the following sub-sections, we will discuss the basics of the working of the protocol. We will see how different tasks like initialization, route advertisement, bandwidth reservation, resource releases and route repairs etc. are performed.

Initialization/Route Exploration:

The access gateways are assigned a unique single digit node id initially. The route exploration starts from the gateways with flooding the neighbors with advertisements. The advertisements are layer 2 frames which includes the Gateway's MAC address, node ID and the available Bandwidth $B_{available.}$ All the nodes in the proximity of the Gateway i.e. one-hop away on receiving the advertisement checks if its requested Bandwidth $B_{requested}$ is lesser than the available Bandwidth $B_{available}$. If the Gateway has enough Bandwidth the node sends back a connection request message 'connect request' to the Gateway. On receiving the connection request the Gateway again checks if it can provide the requested Bandwidth. If the request is accepted the Gateway allocates the requesting mobile node, a node id, which is their node id appended by another single digit number via a 'connect allocate' message.

On receiving the 'connect allocate' message if the mobile node accepts this, it will pass the IPv6 advertisement to its IP layer. The mobile will thus acquire a care-of address using the network prefix in the advertisement. At layer 2, the mobile will acquire a node id. The mobile will then respond back with a 'connect accept' message. On receiving the 'connect accept' message, Gateway will register the mobile's node id and its care-of address. The gateway then updates its Bandwidth and starts sending out new 'advertisement' with new updated bandwidth. It will then forward the IPv6 binding update message to the home network.

A node can still listen to 'advertisements' and request for node ids though it doesn't have any data to forward or it already has primary node id assigned. It can send out 'connect request' control packet without specifying bandwidth reservation request field. Bandwidth is not reserved for this request but a node id is assigned. The node ids assigned are secondary ids which are used in case of a primary node id failure or if the node has data to forward later on. If the node requires to use the secondary node id as primary it sends out a new 'connect request' specifying bandwidth reservation request. Bandwidth is reserved for this flow and it can start using the new primary node id for forwarding data.

<table>
<tr><td>sender_mac
(16 bits)</td><td colspan="2">destination_mac
(16bits)</td></tr>
<tr><td colspan="3">sender_node_id
(32bits)</td></tr>
<tr><td>sequence_number
(16 bits)</td><td>type
(4bits)</td><td colspan="2">Ipv6_address
(32 bits)</td></tr>
<tr><td>bandwidth
(16 bits)</td></tr>
</table>

Fig 1: Advertisement Packet

The first tier of mobile nodes now starts broadcasting ‘advertisement’ control messages which include mobiles own node id, mobile’s Bandwidth $B_{available,}$ MAC Address and the list of all the links Bandwidth’s connected up to the Gateway from this node. The second tier nodes on hearing the advertisements sends out a ‘connect request’ message if the requested Bandwidth $B_{requested}$ is less than the Bandwidth $B_{available.}$ It can selectively send ‘connect request’ messages depending upon the bandwidth available and distance from the Gateway. The first tier mobiles on receiving the ‘connect request’ message addressed to them checks if the requested bandwidth is available up to the Gateway.

sender_mac (16 bits)	destination_mac (16bits)	type (4bits)
Ipv6_address (32 bits)		
bandwidth (16 bits)		

Fig 2: Connect Request

If the request is satisfied it will allocate a node id which will be their node id appended with a single digit number and responds back with a ‘connect allocate’ message.

sender_mac (16 bits)	destination_mac (16bits)		
sender_node_id (32bits)			destination_node_id (32bits)
sequence_number (16 bits)	type (4bits)	Ipv6_address (32 bits)	

Fig 3: Connect Allocate

The second tier mobiles will respond with a “connect accept” message that carries the Bandwidth allocated and may also carry an IPv6 binding update to the home network. The first tier mobiles will reframe the above frame with their MAC address and forwards to the gateway with reserving the updating the Bandwidth on the all links along the way. The second tier node will not start sending data until it receives an ‘ack’ message from the Gateway.

sender_mac (16 bits)	destination_mac (16bits)	alloc_node_id (32 bits)
alloc_node_id1 (32 bits)		alloc_node_id2 (32 bits)

alloc_node_id5 (32 bits)	forwarders_node_id (32 bits)	
sequence_number (16bits)	type (4bits)	ipv6_address (32bits)
timestamp (32bits)	bandwidth (16bits)	

Fig 4: Connect Accept

If the first tier node after receiving 'connect request' message finds outs that there is no enough Bandwidth along its existing link up to the Gateway or if it receives a more feasible 'advertisement' with more bandwidth and small distance it sends out a new 'connect request' message for an entirely new route exploration to the Gateway. When a new route is found the first tier node is assigned a new node id which in turn sends out a 'connect allocate' message to the second tier node assigning a new node id. After the mobile is assigned a node id and received an 'ack' from the Gateway it starts sending out data according to a procedure discussed below.

Data forwarding mechanism:

In this project we consider the flow of data from the mobile nodes to the outside world i.e. external traffic. All the traffic originating from the nodes will be forwarded to the Gateway which acts as a bridge to the outside network. Our routing protocol makes routing relatively simple and easy without even maintaining any internal cache of neighborhood information or routing entries at each node. After successfully establishing the route satisfying QoS requirements from source node to the gateway, the source node starts sending out data packets. The routing protocol uses sequence numbers for each of the packets to avoid duplicate data packets. Initially each data packet is assigned a sequence number 0 and all the nodes receiving it maintain this sequence number. If a node receives a data packet with lower sequence number it rejects it.

The data packet includes originating mobile MAC address, its own node id as the source and Gateways node id as the destination. The first digit of the mobiles node id gives the Gateways node id. These upstream packets will be carried forward by the mobiles whose node ids are a part of the mobile node id. A mobile node need not maintain routing entries for finding out the address of the parent where it has to forward the packet. The node id of the parent node is one digit lesser than the node's own node id.

sender_mac (16 bits)	sender_node_id (32bits)
destination_node_id (32bits)	type (4bits)
sequence_number (16 bits)	Ipv6_address (32 bits)

forwarders_node_id (16bits)	timestamp (32bits)	
originating_node_id (32bits)		bandwidth (16 bits)

Fig 5: Data Packet

Each mobile node forwarding the data packet holds on to the packet until it receives an 'ack' mobile receiving the packet. This helps to quickly identify a route failure due to mobility of the nodes and helps to initiate instant route recovery mechanisms. The procedure for detecting route failures is discussed later.

Route advertisement:

The traffic consumed by 'advertisement' control messages can be considerably reduced by each node if the IPv6 data packets forwarded by that node can be used to serve as an advertisement. Hence, each node along the route of the flow reframes the data packet to include their MAC address and node id. The mobile nodes within the transmission range receiving the data packet to whom the data is not intended for can use it as an advertisement and can send a 'connect request' message.

If the mobile has no data packets to send upstream, then it starts sending normal advertisement messages to maintain the topology. Once a mobile stops receiving data packets it assumes that the mobile no longer needs the required bandwidth and performs resource release process. The process of resource release due to route failure or mobile movement is discussed in the next section.

Procedure for QoS Route failure detection and recovery:

Due to mobile nature of the nodes in MANET the network topology usually keeps changing which results in packet losses and QoS violation. This type of communication is highly not desirable for multimedia applications like video, audio etc. Instant QoS violation detection becomes crucial for these kinds of transfers. A mobile node can detect a Neighbor loss if it doesn't receive 'Hello' or 'I am alive' message in time. In our QoS routing we also use Bandwidth reservation timeout for resource release. If a destination node in a flow doesn't receive a data packet in a certain time it triggers QoS recovery and releases the Bandwidth reserved for the flow.

Uplink mobile failure or movement

The source node acquires a node id through route detection when it requires sending data. After the route has been set up the source node starts transferring data packets to its parent node along the reserved route.

An intermediate node holds onto the data packet until it receives an 'ack' for the data packet send. The node retries to the send the data and after missing 'n' acks from the destination node it

senses that the uplink node has moved and triggers the QoS recovery mechanism. The intermediate node sends out 'route failure' message back to the source node along the downlink. The source node then chooses one of its secondary node id and sends out a new 'connect request' with bandwidth reservation request to its new parent. The parent node then checks if the requested bandwidth $B_{requested}$ is less than the $B_{available}$ and then sends out a 'connect allocate' message back with the same secondary node id it previously assigned. The source node then sends out 'connect allocate' message which travels all the way to the Gateway which updates its node id list and starts using the new route to send data to the source node with new primary node id.

Downlink mobile failure or movement

An intermediate mobile along the reserved route detects the failure of the downlink node if it fails to receive data packets before a bandwidth reservation timeout. When a reserved route is being formed all the nodes along the route maintains a timeout for each of the flow going through it. When a downlink node fails the intermediate node sends out the 'route failure' message to the Gateway node along the uplink. The bandwidth recovery is performed and the reserved resources are released at each of the node along the way.

Mobile movement

When a mobile moves away it fails receiving message from some of its child mobiles or not able to contribute to the flow. In such a case it releases its reserved bandwidth and starts listening to other mobile messages. It takes up a new node id reserves required bandwidth along the route up to the Gateway, and proceed along the steps defined under the initialization process.

Thus instant route failure detection and recovery becomes really crucial for a multimedia specific application. Our routing protocol handles this very carefully and successfully.

Project Deliverables

1. Specification of QoS MMT protocol.

2. Simulation model of QoS MMT protocol using OPNET.

    a. Building control packets
    b. Coding mobile node and gateway process model to handle bandwidth requests
    c. Implementing Decision algorithm in mobile node id/bandwidth allocation process
        i. A node requesting for a new node id would become a primary node id if that requesting node is already not assigned a node id
        ii. A node requesting a new node id would become a primary node id if that requesting node already has a primary node id from the same Gateway and the new node id has less no. of hops
        iii. A node requesting a new node id would not become a primary node id but becomes a backup node id if the already existing primary node id is from a different Gateway though the new node id has less no. of hops

3. Studying the performance of QoS MMT protocol.

    The tests are conducted with mobile nodes moving at a rate ranging from 5mph – 35mph and the number of nodes joining the network from 30-60 with each node requesting varying bandwidth from 64kbps-400kbps.

    The following statistics are collected from the simulations for static network:
        i. Node-Joining Time
        ii. End-End delays
        iii. Throughput

    The following statistics are collected from the simulations for mobile network:
        i. Node-Joining Time
        ii. End-End delays
        iii. Throughput
        iv. Route failure notification
        v. Route availability
        vi. Utilization (iv+v)

4. Documentation of the results.

5. Final Report.

Schedule

1.	Complete Design of the QoS protocol	1 weeks
2.	Reading OPNET	1 weeks
3.	Simulating the QoS protocol	3 weeks
4.	Performance Measurement	2 weeks
5.	Comparative Study with existing QoS protocols	1 weeks
6.	Finalizing Report	2 weeks

References

- Chenxi Zhuand M.Scott Corson. "*QoS Routing for mobile ad hoc networks*"
- S. Chen and K. Nahrstedt. "*Distributed Quality-of-Service in Ad-Hoc Networks*"
- Elizabeth M. Royer and C-K Toh. "*A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks*"
- Andrew S. Tanenbaum, Vrije University, Amsterdam, The Netherlands. "*Computer Networks*"
- www.OPNET.com