TRUST BASED QOS-AWARE PACKET FORWARDING MODEL FOR AD HOC NETWORK INDEPENDENT OF ROUTING PROTOCOL

A Thesis by

Vijaya Laxmi

Bachelor of Engineering, Nagpur University, India, 2008

Submitted to the Department of Electrical Engineering and Computer Science
and the faculty of the Graduate School of
Wichita State University
in partial fulfillment of
the requirements for the degree of
Master of Science

December 2010

TRUST BASED QOS-AWARE PACKET FORWARDING MODEL FOR AD HOC NETWORK INDEPENDENT OF ROUTING PROTOCOL

The following faculty members have read and examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Masters of Science with a Major in Electrical Engineering.

____

Ravi Pendse, Committee Chair

____

Krishna Krishnan, Committee Member

____

John Watkins, Committee Member

DEDICATION

To my beloved and supporting parents, sisters, brothers-in-law and friends

ACKNOWLEDGEMENTS

This thesis would not have been possible without the support and assistance of many people. I would like to thank my advisor, Dr. Ravi Pendse, for his continuous support, guidance and encouragement throughout the course of my graduate studies at Wichita State University. I would like to thank my committee members for their suggestions in reviewing my report. I would also like to thank doctoral student, Nagaraja Thanthry for his help, ideas and suggestions throughout the course of my thesis work. It is a great pleasure to thank my parents, sisters and brothers-in-law for their patience, guidance and support that I have received over the years.

ABSTRACT

The need for users to be able to setup wireless networks as, and when they require, has led to a boom in MANET. The constantly changing status of wireless links, mobility and resource scarcity, pose serious problems when a node in an ad hoc network is required to not only be able to communicate with other neighbors (multiple hops away), but also have demand QOS of intermediate nodes to its delay sensitive packets. As this technology has matured, resource starving of best effort traffic in the presence of priority traffic is not acceptable. Moreover, a true seamless, wireless network would be one in which intermediate nodes do not always need to support the same type of routing protocol in their TCP/IP stack to allow communication between the source and destination node. This research proposes to solve the QOS issues in a wireless ad hoc network by enriching the nodes in a network with trust databases, and a pool table to keep records of its previous interactions with all malicious and trustworthy nodes. A node can assign trust points to well behaving nodes and deduct points away from the database for a bad node. Thus, a node can always have a look at an intermediate node's trust points and its previous performance to decide if this node can be trusted to properly forward its multimedia traffic by satisfying the QOS request. Also, QOS favors are returned promptly to provide incentives for nodes to become trustworthy. This author has proposed to solve the QOS issues in a MANET in a unique way, and has also tried to capture the dynamism of wireless channels by using a Best Effort (BE) timer to gain the best utilization of a costly channel and to provide fairness. A Universal Packet Format is used in this research to ensure communication between two nodes which may be separated by nodes that do not support the same routing protocol in their TCP/IP stack.

Hence, an attempt toward a comprehensive solution for achieving the goals of a seamless QOS aware ad hoc network is made in this research work.

TABLE OF CONTENTS

Chapter Page

TABLE OF CONTENTS (continued)

Chapter Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

LIST of ABBREVIATIONS/NOMENCLATURE

AMRIS	Ad hoc Multicast Routing protocol utilizing Increasing id-numberS
AODV	Ad-Hoc On demand Distance Vector Routing Protocol
BRPHT	Bandwidth Reservation for Priority of Heterogeneous Traffic
BE	Best Effort
CW	Contention Window
DSDV	Destination-Sequenced Distance Vector Routing Protocol
DSR	Dynamic Source Routing Protocol
FE node	Foreign Egress node
HMRSVP	Hierarchical Mobile RSVP
IETF	Internet Engineering Task Force
IntServ	Integrated Services
ITU-T	International Telecommunication Union
LE node	Local Egress node
MAC	Medium Access Control

LIST of ABBREVIATIONS/NOMENCLATURE (continued)

MANET	Mobile Ad hoc NETwork
MBN	Mobile Backbone Node
MH	Mobile Host
MSpec	Mobile Specifications
MS	Mobile Station
MAODV	Multicast AODV
ODMRP	On Demand Multicast Routing Protocol
OLSR	Optimized Link State Routing Protocol
OSPF	Open Shortest Path First
OSI	Open Systems Interconnect
PDR	Packet Delivery Ratio
PQAMP	Priority based QOS-Aware MAC protocol
QOS	Quality of Service
RSVP	Resource Reservation Protocol
RERR	Route ERRor

LIST of ABBREVIATIONS/NOMENCLATURE (continued)

RREP	Route REPly
RREQ	Route REQuest
RP	Routing Protocol
RT	Routing Table
TDMA	Time Division Multiple Access
TTL	Time To Live
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network

CHAPTER 1

INTRODUCTION

Section 1.1 provides an introductory session for readers to become familiar with the basic and important properties of an ad hoc network. In section 1.2 intrinsic issues faced by a MANET are discussed, in order to build some background work for the forthcoming proposal to solve some of them. It is rightly said that unless one knows the problem statement, they cannot find a solution for it. Section 1.3 raises a question, in the mind of the reader regarding the need for trust in a MANET, and then tries to provide an answer to it. Section 1.4 explains the usage and importance of interoperability in an ad hoc network.

Keywords: QOS, trust, mobile ad-hoc networks, interoperability, fairness

1.1 MANET Overview

A set of random nodes which form a network on the fly have been defined as an ad hoc network. A MANET is a mobile version of the ad hoc network setup. These networks are characterized by frequent topology changes due to link breaks, resource scarcity, and insecurity. Even though there are a lot of issues with the ad hoc networks, they have gained a lot of popularity since they are inherently infrastructure-less and cheap. Such computer to computer networks are very popular and can consist of anything between two nodes to several hundred nodes.

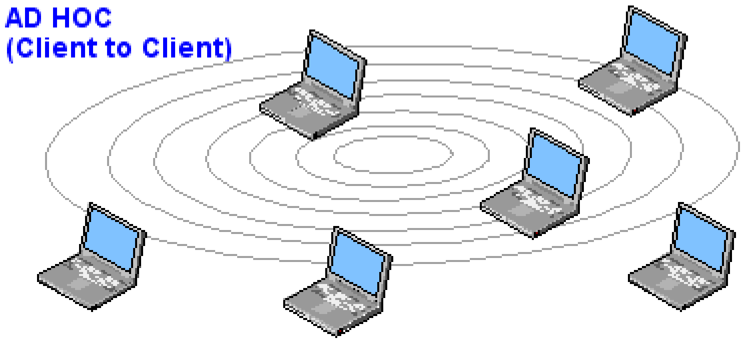


Figure 1.1 Typical setup of ad hoc network

Nodes act as independent routers and support the network by forwarding packets for each other. When there is a specific need; such as, file transfer, playing multiple-player computer games and sharing documents in a conference, computer to computer networks can be used and easily setup on machines using a Windows operating system. Another interesting usage for such networks is seen when hosts need internet connectivity. In such scenarios, to avoid the hassle of setting up internet connectivity for every host, people can share internet connectivity and use one of the hosts as the gateway to the internet. It should be noted here that an ad hoc network requires a wireless adapter to be installed on the device which is planning to join the network. Most of the devices on the market today have a wireless antenna which allows them the capability of joining a MANET.

1.2 Inherent Issues With QOS in Ad Hoc Networks

One of the significant issues in a wireless network is the dynamism of wireless channels. It may be the case that a wireless link between two nodes can be overloaded at times, even for best effort traffic; however, after awhile the link may be rendered free for traffic with specific

QOS demands. Deployment of QOS in such fickle networks is the greatest challenge. In the self-forming ad hoc networking world a lot of steps are taken to mitigate the impact of such dynamism. While all the layers must learn to survive and cope with the after-effects of time-varying channels at the layer 3 of the OSI model; there have been detailed efforts focused on finding and maintaining routes. One approach is to utilize path diversity by rapidly switching between paths so that packets are temporarily sent along good paths. More so, when packets are sent by an application to a host which cannot afford to sustain greater delays, and whose successful delivery is contingent upon the availability of certain level bandwidth resources, there is no use in discovering paths which cannot support the incumbent traffic. Multimedia traffic, such as audio, streaming video, internet gaming, etc., cannot afford more than the ITU-T suggested 150 ms of end-to-end delay. There has been a lot of research done in this field to ensure that such delay sensitive traffic is treated with high priority by all nodes in a network. Due to the boom in applications like video/teleconferencing, video streaming, IP telephony, etc., and the need for people to be always connected, QOS requirements have become a necessity in ad hoc networks. Though true prioritization of the delay sensitive traffic may not be totally possible, a better than best effort service can be achieved. General problems with wireless networks, such as the limited transmission range of wireless nodes, power attenuation, channel noise, and signal strength attenuation with distance, can be easily attributed to be present in ad hoc networks too.

1.3 Trust in an Ad Hoc Network

The best feature of a wireless ad hoc network is that a node can start communication with a destination, which is not present within its radio communication range, with the help of other intermediate nodes. These intermediate nodes may choose to be selfish for several reasons such as, saving power (by entering into sleep mode), saving CPU cycles, and acting as a black hole in the network. The consequences of such node activities can clearly be seen to be disrupting to the overall communication between nodes, by splitting the ad hoc network into islands of nodes unable to effectively reach each other. Due to the limited transmission ranges of a node, it has to depend on neighboring nodes to reach a destination which is multiple hops away. This is where trustworthy routing comes into the picture. As the network is formed on the fly, there is always some level of trust among nodes to forward each other's traffic. [1] If combined with the prospect of getting instant rewards for being a trustworthy and responsible node in the network, nodes would no longer want to lose rewards or in essence be selfish. Regardless of how strong the routing protocol is, it may fail in the presence of misbehaving or selfish nodes. Routing and packet forwarding in a MANET always assumes some level of cooperation among independent routers (nodes).

As elucidated by Richard Dawkins in "The Selfish Gene" [2], reciprocal selflessness is beneficial for any natural system where favors are granted simultaneously by everyone. Thus there is a fundamental motivation for cooperation among entities because of instant rewards. The advantage of being well behaved and a responsible entity is not so obvious when there is a delay in doing a favor and getting a repayment for it. As explained above, this would happen if a mobile node agrees to forward traffic for its neighbors. Dawkins [2] has used an ecological case to explain the survival chances (hence, gene selection) of birds cleaning parasites from each other's head and body, a task which they otherwise cannot do themselves. Birds have been categorized by Dawkins into two types: 'suckers', those that are always ready to help others, and 'cheats' who get other birds to do their grooming for them, but they never return the favor. In such a system, cheats have a clear advantage over suckers; but as time passes both species are driven toward extinction. A third, new species is then introduced by Dawkins, called 'grudgers'.

Grudgers are helpful to everyone in the beginning, but they hold a grudge against those species who fail to return favors; and consequently, no longer help them. As per Dawkins' work, simulations have been done to prove the same with a large population of cheats and a tiny population of grudgers and suckers, where the grudgers win slowly over time. Dawkins has defined winning as obtaining the greatest amount of benefit, assuming overall cost, for grooming the head of another bird, and gain for having grooming being done for oneself. Profit is to get the species multiplied and loss will lead toward extinction of a particular species. The logic is that the suckers do a lot more favors than they receive, due to the presence of a large number of cheats, eventually leading to a decrease in the number of suckers. On the other hand, the number of cheats multiplies. The grudgers suffer a few losses too, but quite a few less than suckers. Once the suckers become extinct, the grudgers multiply quickly at the expense of the cheats, since they do not groom the head of a cheat twice, and the cheats would not help other cheats anyway. Over time cheats slowly decrease in number because the probability of a first (time) help by any grudger increases with the higher population of grudgers. As a result, the population of all grudgers is bound to increase; whereas, other species are bound to become extinct. By defining an apt cost, an instant reward, and maintaining a database of past interactions with untrustworthy nodes, it will help achieve similar results as the grudger species, i.e. pushing the cheats (malicious nodes) toward extinction.

The need at hand is to convert a simple ad hoc network to an intelligent group of nodes which are enriched with a database of trust relationships for other nodes, and determine trust points based on their past and present interactions. In this paper, the author has done a novel research to modify existing ad hoc networks to induce intelligence in them; thereby alleviating problems of QOS and routing without a significant change in overhead or end-to-end delay. As this resource hungry technology has matured, it has become imperative to utilize the wireless channels to the best possible extent not only for QOS demanding traffic, but also for BE traffic. Fairness and support for QOS have always been self-conflicting requirements of wireless network nodes. A scheme is devised by the author of this paper to balance the two most conflicting criteria of a MANET by taking into consideration the time varying nature of link load. This is very important in the ad hoc world, as bogging down of BE traffic in the presence of voice/video traffic is something which is not really expected of a mature technology. So far, not much work has been done to achieve getting the two conflicting goals together.

1.4 Need of Interoperability in MANET

In ad hoc networks, where nodes need to be able to communicate to their neighbors as and when they want to, it must be assured that even though nodes support different routing protocols (RPs) in their stack, they can still communicate. Otherwise, the ad hoc network is not actually providing seamless connectivity. Rather, it appears to be a group of island nodes who can only talk to each other within the island. It should be noted here that nodes can normally communicate with each other only if they talk the same language (RP). In order to support such a network which is expected to support advanced applications such as; communications in emergency disaster management, video conferencing in a workshop or seminar, and communications on a battlefield, a platform needs to be devised to allow nodes which originally support different RPs to communicate intelligibly. Routing protocols for ad hoc networks can be divided into two general types, viz reactive and proactive. Reactive protocols find routes to a destination only when there are impending packets to be sent at the source node. On the other hand, proactive protocols store information about the best possible routes beforehand. Proactive protocols such as DSDV, OLSR have low latency in route discovery but are not scalable for large networks due to high overhead, storage requirements and power consumption at the node. In the case of reactive protocols such as AODV, DSR nodes no longer need to store a routing table (RT) and therefore, they do a complete route lookup to find a path to a destination due to the protocols on-demand characteristic. Even though this type of RP is scalable to large networks, the latency in route discovery can be high for huge networks.

Every RP has its own advantages and disadvantages; hence, their usage needs to be carefully exploited to achieve the goals of greatest throughput and scalability. From the above discussion, it is evident that no single RP would be suitable for ad hoc networks. Therefore, the author has tried to implement interoperability of RP in an ad hoc network to get a fully seamless, wireless network along with QOS and fairness for end users.

The rest of this thesis is organized as follows. First, a summary of all related work done by researchers in the field is covered. Next, Chapter 3 describes the proposed methodology in detail with the help of entity flow diagrams for both DSR and AODV routing protocols. Chapter 4 and 5 looks into the mathematical modeling aspect of various parameters used in the proposed trust model, along with an analytical comparison of the trust model with RSVP. Chapter 6 discusses the present implications of the thesis work done, along with future work. It also provides for comprehensive analysis and results.

CHAPTER 2

LITERATURE SURVEY

This chapter of thesis work will review the various research papers related to this work. For the sake of the completeness of this research paper, Section 2.1 discusses in detail the DSR routing protocol. Section 2.2 reviews the AODV routing protocol. Section 2.3 explains the traditional methods used to obtain QOS in an ad hoc network along with cross layer approaches for the same goal. In Section 2.4 techniques being used for providing fairness and utilizing resources in a MANET are explained. RSVP for network is summarized in Section 2.5. Section 2.6 elucidates the research done so far in the field of interoperability of routing protocols in an ad hoc network.

2.1 Dynamic Source Routing Protocol

DSR is an on demand multipath routing protocol developed for wireless networks. [3] [4]. As DSR is a source routing protocol, each intermediate node (between source and destination) append their IP address at the time of route discovery to the route request. This enables the destination to reverse the sequence of addresses in order to send a RREP to the source. Mobile hosts maintain a route cache where they cache learnt source routes. The two basic operations of the protocol are route discovery and route maintenance. Whenever a sender node needs to send packets to a receiver node, it checks its route cache for a possible path. If the node is unable to retrieve any path information for the current destination, then the route discovery process is triggered. It should be noted here that each entry in the route cache has an expiry timer associated with it, after which the route automatically gets deleted from the cache.

2.1.1 Route Discovery

Ad hoc nodes are assumed to be selfish, meaning that they may not be ready to forward packets for their neighbors and they may not be ready to take part in all protocols of the network. Route discovery enables ad hoc nodes to find paths to destination nodes on-demand. The sender node generates a RREQ and broadcasts the packet. All neighboring nodes within the wireless transmission range of the sender listen to the broadcast. The packet consists of a target address (destination), initiator address (source), and a unique request ID (generated at sender node). The pair, consisting of the initiator address and request ID, is maintained by each node in the ad hoc domain for detecting duplicate RREQs. The neighboring nodes reply to the packet based on the following steps:

a) If the initiator address and request ID for the current RREQ is found in the list of the node's recently seen RREQs, then silently drop the routing control packet.

b) Or, if this node's address is already listed in the ROUTE RECORD in RREQ, then silently drop the packet.

c) Or, if the target address matches the node's own IP address then send a RREP to the initiator node by following the path already recorded (route record).

d) Otherwise, append the node's IP address and rebroadcast the RREQ.

The target host finally sees the RREQ and generates a RREP to reply to the sender node. Discarding duplicate RREQ packets and recently seen RREQs prevents routing loops and maintains updated route information.

2.1.2 Route Maintenance

Due to the inherent unreliability and resource constraints of a wireless medium, periodic routing updates is not a feasible option. Usually these routing updates are designed for route maintenance in conventional routing protocols. In an ad hoc domain, the Mobile Host (MH) does not send any kind of periodic updates. Routes are monitored while they are being used by nodes i.e., they are active. A hop-by-hop/end to end acknowledgement, or any kind of explicit ACK, is used in such a medium to gain information about route breaks or the maximum number of retransmissions as early as possible. The RERR packet contains the address of the host which detected an error in link and of the node which it was trying to reach. As soon as a RERR packet is received, the hop in error is deleted from the route cache of the node and all other paths where this node appears are truncated as well.

2.1.3 Optimizations Over DSR

There are several optimizations done over basic DSR. A MH in a wireless medium can configure its Network Interface Card (NIC) into a promiscuous receive mode, so that it can over-hear packets intended for others and add the path to its cache.

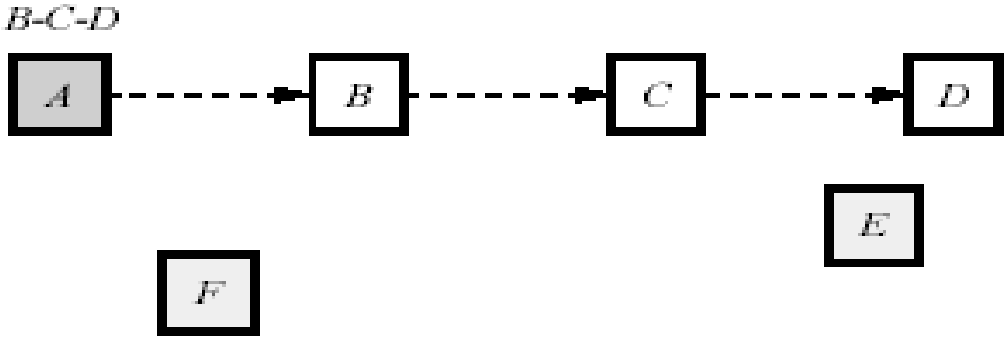


Figure 2.1 Route optimization in DSR [4]

In the sample scenario above, even though node B is just relaying packets from source node A to destination node D, B can add route to D in its cache in order to fully utilize its routing table. Now if node A learns about a shorter path to D then it can add that route to its cache anytime. Occasionally, if a node is not the target node, it may still send a RREP to initiator nodes. Let us assume that node F sends a RREQ broadcast for node D, and node B is in the transmission range of F. B already has a route to D in its cache (B-C-D), so it can send a RREP to F (F-B-C-D).

2.2 Ad-Hoc On Demand Distance Vector Routing Protocol

[5]AODV is a reactive protocol similar to DSR which also supports multicast routing. [6] It is a single path routing protocol where it maintains only one path per destination. The MH remains silent until it needs to send packets and has no routing information for the destination. Routing information for the destination is collected by a RREQ/RREP query cycle. A RREQ is sent as a one hop broadcast. Nodes receiving this packet set up backward pointers (Figure 2.2.1) to the sender node. [7] A typical RREQ consists of the following fields:

- Source address - sender node's IP address

- Source sequence # - a unique number used to maintain freshness of reverse route to sender node.
- Broadcast id - it is generated each time a new RREQ is generated by sender node.
- Destination address - receiver node's IP address.
- Destination sequence # - determines how fresh a route should be before it is accepted by sender node.
- Hop count - hops in an ad hoc domain

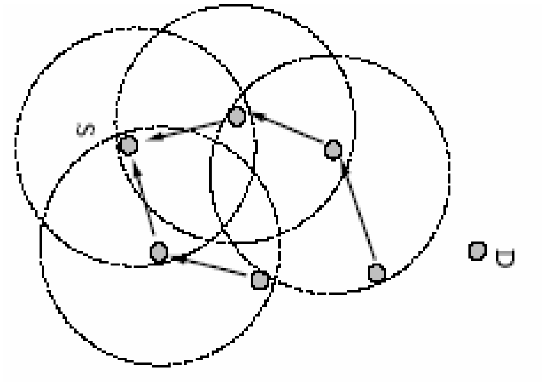


Figure 2.2.1 Backward path setup for AODV protocol

A RREP is sent by a neighboring relay node, if it is either the receiver itself or if it knows of a path to the destination with a sequence number greater than or equal to what is present in RREQ. A RREP is now sent by this node to the sender, or a RREQ is rebroadcasted again. In order to avoid stale RREQs from oscillating in the network, each node keeps track of <source address; broadcast ID> to mark uniqueness of RREQs. The redundant RREQ is supposed to be silently dropped by the MH. Forward pointers (Figure 2.2.2) are set toward the sender node once a RREP is propagated to it by other nodes in the path. When the sender node receives a RREP with a greater sequence number or lesser hop count it updates its routing table. Just like DSR, every route has a timer associated with it so that routes which are no longer used are deleted from the node's routing table. Each node's routing table consists of an entry of:

a. Receiver

b. Next Hop

c. Number of hops (metric)

d. Sequence number for the receiver

e. Active neighbors for this route

f. Expiration time for this entry

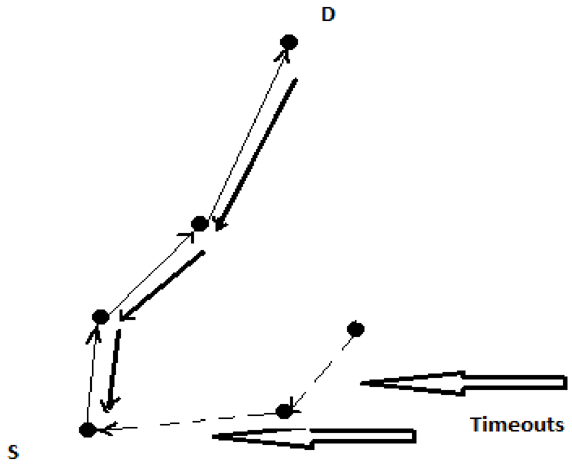


Figure 2.2.2 Forward path setup for AODV protocol

If a node becomes dead or in the event of a link failure, an upstream node sends a RERR to the sender node and the path gets deleted from the routing table. If the sender still needs the route, a fresh route discovery is triggered.

2.3 Traditional QOS Support and Cross Layer Method

Researchers in [8] have proposed to prioritize channel access at MAC layer, with the help of a retry limit and up to five varying priority levels (HIGHEST, HIGH, MEDIUM, LOW, and NORMAL). These levels are based on the type of traffic (originated at the node or relayed traffic) and role of the node (sender, receiver or forwarder). Significant delay and throughput improvement has been achieved with the Priority based QOS-Aware Mac Protocol (PQAMP). Even though this approach is easy to implement, it has a drawback i.e. the unfairness of MAC level access in such protocol. In some cases, best effort traffic is not starved but is kept waiting longer for channel access in order to reduce the back off window sizes of RT traffic. In [9] studies have been made that provide for QOS in ad hoc networks for multimedia traffic. MAC protocols designed by IETF such as IEEE 802.11e [10] are designed for providing quality of service to delay sensitive traffic. IEEE 802.11e has eight levels of prioritizing traffic for serving the users. However, that does not scale well for ad hoc networks and is difficult to implement. Moreover, the protocol suffers from throughput degradation at heavy traffic load. [11] Work done in this paper tries to combine TDMA scheduling along with route discovery for optimization of end-to-end delay. According to researchers, QOS requirements can be satisfied with the help of a TDMA based system depending on the free available bandwidth at each node. Time slot assignments are done dynamically based on the respective priority of the traffic. Dynamic time slots are assigned based on the current status of congestion in the network. Route discovery is needed to find candidate route vectors for the scheduling algorithm to decide on any new connection. The scheme here requires a lot of time synchronization and proper scheduling which could be difficult to control in an ad hoc network. Moreover, time slots at the sender and receiver should be free at the same time; or else, there is a waste of bandwidth. Authors of [12] [13] [14] [15] have realized that to provide better than best effort QOS service, by utilizing traditional layered design in multi-hop wireless networks, is very difficult. Hence, their research points toward utilizing congestion control (of transport layer), rate control (of transport layer), and admission control (of network layer) to improvise the channel access at MAC layer for resource hungry multimedia traffic. The choice of MAC layer protocol affects the route discovery which in turn affects the end to end delay of packets. One of the advantages of this cross layer method is that, the role of each inter-dependent layer in packet transmission is carefully exploited to get the best of QOS service in a dynamic ad hoc network. Given the transient nature of ad hoc networks, a joint design method of multiple layers of protocol provides better control over QOS parameters. Even though these cross layer approaches tend to get the best of all layers they are, at present, very complicated for implementation. Actual gains in service levels achieved by a best of cross layered scheme are still open to future research and simulations.

2.4 Fairness and Resource Utilization Schemes

Studies in [16] indicate that fairness problems related to the hidden terminal issues with IEEE 802.11 can be solved by improving the back-off algorithm. The back-off algorithm of IEEE 802.11 is very partial with the nodes which have a small Contention Window (CW), as they tend to gain channel access better than nodes having a larger CW. Researchers have tried to enhance the back-off algorithm to include several parameters viz number of neighboring nodes, along with their congestion states, to assuage the exponential rise of CW and provide fairness. But this kind of scheme would kill delay sensitive traffic, as it would have to contend for channel access with its neighboring nodes. An ad hoc network desires fairness, along with QOS support, for a balanced performance. In fact, it is a difficult task to incorporate both QOS and fairness in such networks due to the unique characteristics of each kind of traffic. In [17] researchers have proposed a scheme to include awareness of the congestion states of neighbors and have tried to maintain fairness of traffic flow. Flow weight assigned to traffic flow will vary according to the service requirement (minimum bandwidth) of each traffic flow. Resources are allocated first to the highest weighted flow, and then to the lower ones, in that order. This ensures that priority traffic gets the required bandwidth. Among BE traffic the leftover bandwidth in the network is equally distributed. Starvation of BE traffic is therefore seen when a heavy load of priority traffic is present. To minimize this effect, a distributed algorithm is proposed wherein varying flow weights are assigned to each traffic, and all information (service tag, flow weights, etc.) regarding a flow is updated within a two hop neighborhood. Each outbound packet contains an arrival time and service tag of the next packet; consequently, allowing the receiver to check its own table of flows and send back the number of flows having lesser service tags than the flow. The sender adjusts its back-off timer accordingly.

2.5 RSVP Techniques for Ad Hoc Networks

Authors in [18] propose residual bandwidth of nodes to be inserted into the normal routing protocol updates; and consequently, using it for admission control of multimedia traffic. There is a level of control exercised when the residual bandwidth of each node is considered before admitting a call. Again, for mobility induced congestion, further congestion control mechanisms are needed. The congestion control mechanism is now triggered to sense flow rate in advance and control it. The Mobile Backbone Node (MBN) is elected dynamically to form a backbone network to support the co-existence of BE and priority traffic which involves additional overhead costs. This approach tends to be unfair with BE traffic and it is not easy to estimate the congestion state of nodes in a multi-hop wireless network. A Bandwidth Reservation for Priority of Heterogeneous Traffic (BRPHT) model proposed in [19] provides three classes of services, viz RT-CBR (RT traffic generated at constant rate), RT-VBR (RT traffic with a bursty rate) and NRT (Non RT traffic). Nodes which need to send RT traffic send reservation requests to reserve slots with the help of RF (reservation frame) in TDMA super frame for transmission. A receiver gets to choose the amount of bandwidth (slots) that can be allotted to the requesting nodes. A scheduling algorithm is followed at the receiver in order to multiplex requesting nodes on their reserved slots, so that each node gets their minimum requested bandwidth. Unlike traditional IEEE 802.11e, where one slot per requesting node is reserved, BRPHT takes into consideration the varying QOS needs of users/applications and tries to avoid wasting the bandwidth by multiplexing packets of several nodes in one slot. One drawback of this method is that perfect synchronization of nodes is assumed on a subframe basis, so that slots are properly requested/allotted for/to maintain QOS requirements of RT traffic. At high traffic periods the proposed protocol is shown to perform better than IEEE 802.11e as far as the BE packet delay is concerned. This can be explained because the BE traffic of a node needs to contend only with the BE traffic of its neighbors, unlike IEEE 802.11e, where BE and RT traffic need to contend together for accessing the same channel.

RSVP is a standard protocol for providing QOS for integrated services on the internet. [20] RSVP protocol is used by a host to request specific qualities of service (maximum tolerable delay, minimum required bandwidth, etc.) from nodes in the network for particular application data streams or flows. RSVP is also used by routers to deliver quality of service requests to all nodes along the path(s) of the flows and to establish and maintain states to provide the requested service. RSVP cannot be directly applied to mobile environments due to the dynamic nature of nodes. Researchers in [21] have proposed a mobile version of RSVP. In this pointer forwarding Hierarchical Mobile RSVP (HMRSVP) scheme, advance resource reservations are made for approximately a forward one-step path from a mobile host along a forwarding chain (ideally Markov chain infinite states). For modeling of scheme, a hierarchical wireless environment of a tree topology is assumed. Methodology is shown to work great for nodes with high locality movement. Authors of [22] [23] believe that advance reservations of bandwidth will help maintain QOS requirements of MH in a mobile environment. In this scheme, MH will present an MSpec with a list of all possible locations that a MH could go to in the near future. Active reservations are the ones which are currently used for traffic flow. Passive bandwidth reservations are made in advance to accommodate nodes' mobility to provide a fast and smooth handoff. However, due to the transient nature of nodes in a high mobility environment, there is a limit to the number of passive reservations which can be made.

2.6 Approach for Interoperability Issue

Every ad hoc routing protocol has unique characteristics, which makes it difficult to envision a single routing protocol to suit the whole world. Moreover, in a disaster prone area where self-organizing ad hoc nodes form a network, it is possible that the nodes may support different routing protocols. Unless every node in a domain understands each other's language (routing protocol), no communication can take place among them. To avoid such a situation, interoperability of routing protocols in an ad hoc network is needed in order to realize a truly seamless MANET. Another advantage of such a scheme is the choice of the best suited routing protocol for a particular scenario/node/application, which can be chosen without regard to RP running on neighboring nodes. An observation can be made here that network resources are optimized due to the increase in node density, irrespective of their supported routing protocol.

Interestingly, in the past the area of study involving an ad hoc network had not been extensively researched. It is only recently that this topic is being [24] discussed in the networking world. Researchers in [25] have worked to design a scheme for facilitating interoperability in a multicasting environment. A composite packet for this requirement has been formed with the following fields:

Fprotog: type of the protocol e.g. MAODV, ODMRP, AMRIS

Fmtypeg: the type of message e.g. request, reply, data

fsrcAddrg: address of node initiating the message

fdstAddrg: address of destination node or multicast group

FnextAddrg: next hop towards destination

fseq numberg: sequence number of messages exchanged (required to detect duplicate messages).

When a node receives a routing packet, it will check for the protocol field. If it matches the RP supported by a node, the packet will be handled by a default process. Otherwise, mtype is checked and routing control packets are silently dropped. If the packet consists of data, then two approaches are analyzed in this paper. In a flood based approach, the node rebroadcasts the data to its one hop neighbors. Nodes also cache packet headers to prevent looping and unnecessary retransmissions. This approach is not scalable to large and unstable ad hoc networks for obvious reasons; such as, excessive control overhead and limited resources in ad hoc scenarios. In a facilitator approach, facilitator nodes are randomly selected between two neighboring domains as long as they have full routing knowledge of multicast groups within its domain. Fac-request and fac-reply are sent or received periodically by all facilitators for inter-domain link creation and maintenance. In the work done by Madhusudan [26] the ad hoc nodes' support for various routing protocols compatible with each other is proposed. Nodes in each ad hoc domain may support different routing protocols in their network stack. The concept of egress node (Figure 2.6.1) is introduced and is analogous to an Area Border Router in OSPF. They lie in an overlapping domain supporting two or more routing protocols. Due to their strategic location, they can overhear routing control packets from both ad hoc domains. Egress nodes could be Local (LE) or Foreign (FE), depending upon the relative position of node. Each node in an ad hoc network will maintain an egress table, along with the routing table, which will contain information about reaching the Local Egress (LE) nodes, domains that are connected through them and the last discovered inter domain destination. LE nodes will contain two routing tables, a viz routing table (intra domain) and a proxy route cache (inter domain).

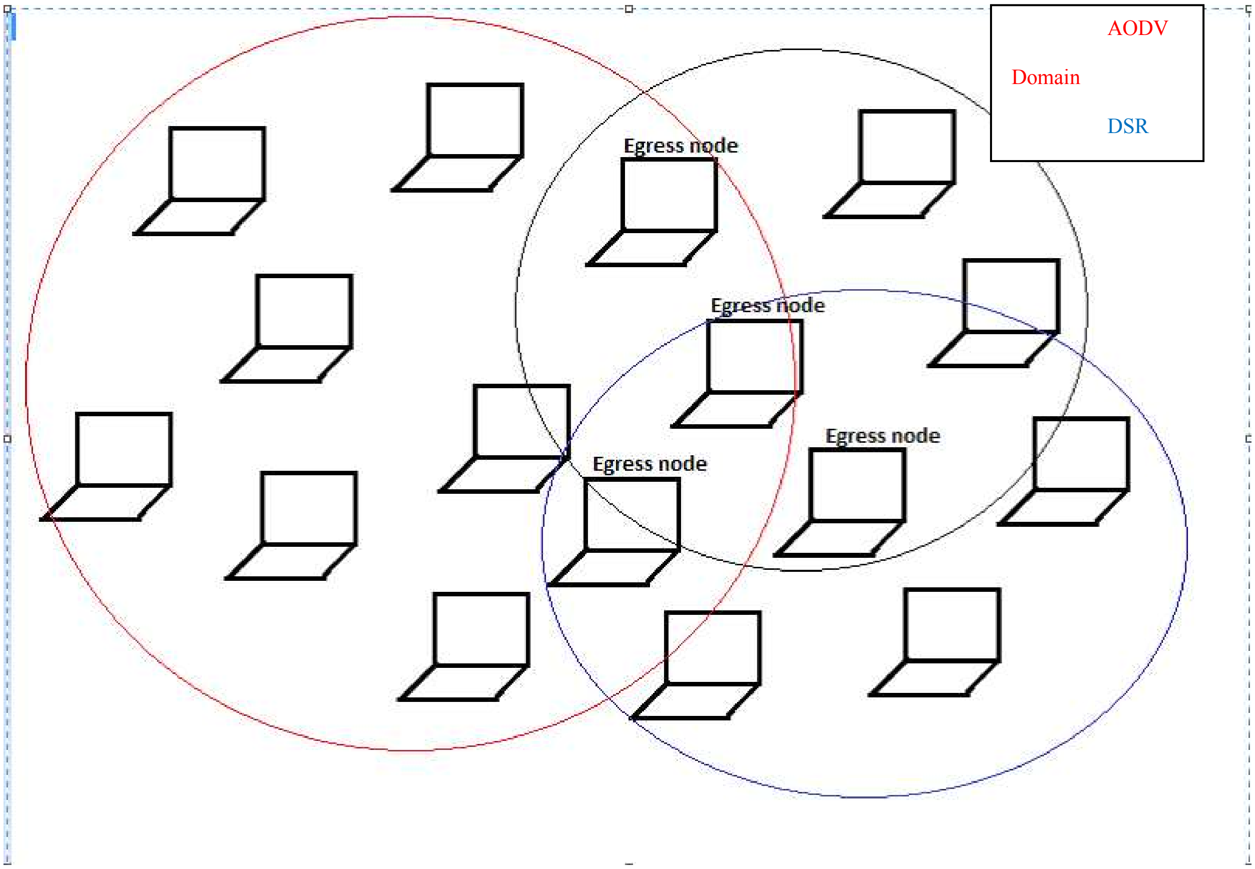


Figure 2.6.1 Typical network setup for interoperability

Three phases in which interoperability functions are described are as follows below:

<u>Bootstrap phase:</u> At boot up, nodes build an egress table with the help of a regular exchange of universal hello packets. Nodes put their node ID in the egress node ID field so that this information is updated to all the intra domain nodes. Eventually all egress nodes are discovered. All intra domain nodes build a synchronized egress table due to a continuous exchange of the hello packet.

<u>Route discovery phase:</u> In this phase, nodes work to discover the inter domain destinations along with a route to the egress nodes. All nodes maintain a routing table (intra domain routes)

and proxy route cache (inter domain routes). Whenever a node needs to send any multimedia traffic, with user requested minimum delay and maximum bandwidth, a RREQ is generated by a source; such that, all nodes within a domain hear the packet. A RREQ is modified to contain PPS (Packets Per Second) and delay requirements in the packet itself, depending on the application needs. Hereafter, the source node goes into an active state and waits for a reply from all nodes within the domain. If the egress node knows about the destination (from its RT or proxy route cache), it sends a RREP to the source nodes, or else it sends a RREQ as a universal packet (with TTL set to 63) to its neighboring Foreign Egress (FE) nodes. This request is propagated into a foreign domain until the TTL expires or the destination is found. A route reply is sent to the source node with metric equal to the number of hops required to reach it from the foreign egress node. In case the destination is not found by the foreign egress node, an ICMP destination unreachable is sent to the source node which ultimately traverses to a passive state.

Data exchange phase: As the name suggests, data packets are exchanged after the route to a destination has been determined. The source node decides which path to take depending upon the metrics advertised with the help of the egress (LE/FE) nodes.

Universal packet format

<table>
<tr><td>Route Type</td><td>Code Bits</td><td>Header Length</td><td>Request Id</td><td>TTL</td><td>Sequence Number</td></tr>
<tr><td colspan="6">Egress Node ID</td></tr>
<tr><td colspan="6">Source Address</td></tr>
<tr><td colspan="6">Destination Address</td></tr>
<tr><td colspan="6">Header Checksum</td></tr>
</table>

Figure 2.6.2 Universal packet format header

Route type – Routing Protocol used by a node when exchanging Hello packets.

Code bits – Helps to identify the type of packets. Packet types could be any of the following:

Hello packet

Route discovery packet

Route reply packet

Payload packet

Opcode Value	Functionality
001	Exchange of Hello
010	Route Request
011	Route reply
100	Data exchange
111	Destination unreachable

Figure 2.6.3 Opcode values for hello packet

Egress node ID – Egress node for the particular routing domain.

Header length – Length of header used to determine length of data.

Source address – The address of a source node that is originating a RREQ within the domain.

Destination address - The address of the node for which the source node has initiated a RREQ within the domain.

Request ID – Unique identity of a route request generated by an egress node.

Time To Live - The maximum time a packet can live. This is used to avoid looping of packets in a domain and consumption of scarce bandwidth in an ad hoc domain.

Header checksum – Checksum of header computed every time the universal header is processed.

Sequence number – Used for maintaining updated route information and to avoid loops in the network.

In many recent works, importance of interoperability at all levels of an ad hoc setup is being focused in order to realize the best feature of being truly seamless. The authors of paper [27] have envisaged a uniform wireless architecture for boundless integration of heterogeneous networks, and also presented an interesting food-for-thought for future networking. Even though the task at hand is complex, work is being done on providing a seamless network access to users. For example, at any MS, mobile phone users can switch over to APs, BS anytime/anywhere or WPAN devices, depending on resource availability and ease of access. Issues at various layers of an OSI model are studied, with focus on the network layer which is supposed to act as a transparent interface between various access technologies (WLAN, Cellular-based networks, WPAN, MANET, etc.) and end-to-end upper layers.

CHAPTER 3

METHODOLOGY

Hereby, a schema is proposed to forward multimedia packets in a highly dynamic ad hoc environment, independent of routing protocol based on the trust levels of nodes in the network. In an ad hoc domain, there has to be a certain level of trust between nodes to forward each other's traffic. A human tendency to trust someone who is known to always fulfill a promise, over somebody who often fails, is an idea used here by the author to get reliable delivery of priority packets in an ad hoc domain. Moreover, the prospect of getting rewards for being a trustworthy node in the network prevents more nodes from being selfish. The study not only facilitates priority traffic handling, but also provides fairness for all priorities of traffic types prevailing in an ad hoc network. In a disaster stricken area or on battlefields, where ad hoc networks are commonly found, it is never guaranteed that all nodes in the domain support the same routing protocols. In this proposal, routing domains are no longer a barrier to forward packets, and the usage of universal routing control packets leads to a seamless communication for all kinds of nodes.

The goal of Section 2 is to solve the interoperability of routing protocols in a MANET**.** In Section 3, an overview of the proposal is presented. In Section 3.1 and 3.2, elucidation of the proposal in DSR (Dynamic Source Routing) and AODV (Ad Hoc On-demand Distance Vector) routing protocols are given, respectively.

3 Universal Packet Approach

As described in Section 2.6 of Chapter 2, the universal packet format allows for having the facility to route seamlessly between different routing protocols.

3.1 Overview of Proposal

In this model, the presence of malicious mobile nodes which would tend to be selfish or power saving is assumed. The wireless medium is considered to have a fixed attenuation throughout without any ambiance noise. Whenever the source node needs to send any traffic, it makes a check for available paths in the routing table (RT) and also looks up the previous trust states (in case a path is found). A trust value of zero is initially assumed for a new node in the network. If a route is not found, a RREQ is generated so that every other node in the domain hears it. In this RREQ, Max delay and Min bandwidth (based on application in use) is included and it is used to calculate the priority of traffic according to use of the priority table (refer Table 2). The nodes develop a trust relationship in such a network where they promise to deliver packets for each other. This trust level, which nodes assign to each other, is very useful in handling priority traffic. Just like two people talking different languages cannot understand each other, nodes supporting different routing protocols in their network stack cannot communicate intelligibly. To overcome that limitation, universal packet approach, as explained in the previous section, is used. In this research, care has been taken to best utilize available network resources. Sometimes nodes may get overburdened and not be able to handle the load any more, even if they had promised to support priority traffic flow. In such cases, participating nodes will send a QOS_LOST message (explained in detail in the following section) back to the source node to alert the source of a congested node in the path. This helps source nodes to keep track of congested nodes in their network paths and change their trust points accordingly. It is also intuitive as to why an ad hoc node would carry traffic for others, if it is simply overloaded because of that traffic. It is easier to understand this behavior since a node, if overloaded by traffic, may not carry traffic for other nodes.

3.2 DSR as a Foreign Domain

Wait_timeout for DSR (as per RFC definition) = RING0_REQUEST_TIMEOUT (30 seconds)

NODE_TRAVERSAL_TIME (assumed to be 40 milliseconds) is a conservative estimate of the average one hop traversal time for packets and should include queuing delays, interrupt processing times and transfer times.

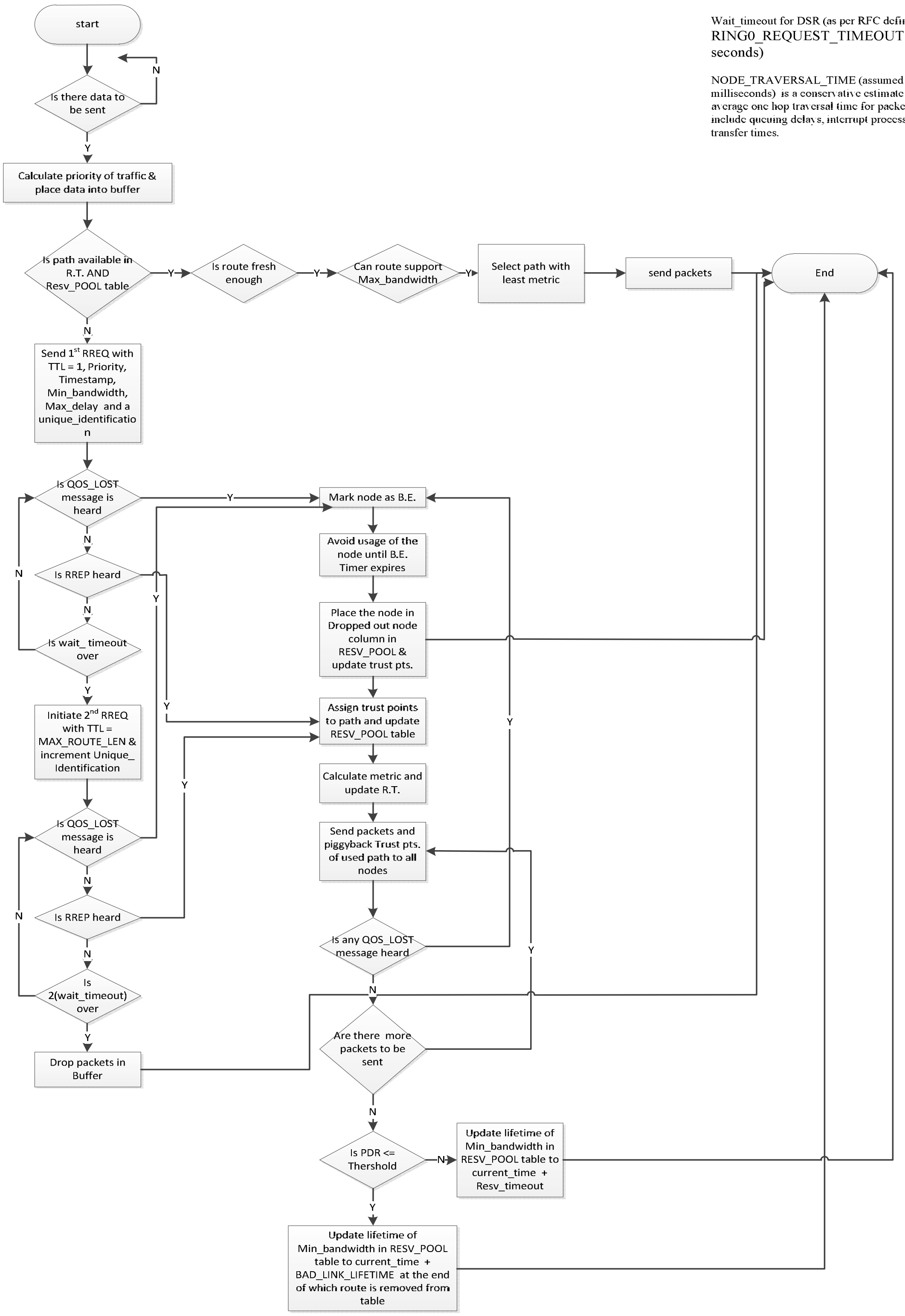

start
Is there data to be sent
N
Y
Calculate priority of traffic & place data into buffer
Is path available in R.T. AND Resv_POOL table
Y
Is route fresh enough
Y
Can route support Max_bandwidth
Y
Select path with least metric
send packets
End
N
Send 1st RREQ with TTL = 1, Priority, Timestamp, Min_bandwidth, Max_delay and a unique_identification
Is QOS_LOST message is heard
Y
Mark node as B.E.
Avoid usage of the node until B.E. Timer expires
Place the node in Dropped out node column in RESV_POOL & update trust pts.
N
Is RREP heard
Y
N
Is wait_ timeout over
N
Y
Initiate 2nd RREQ with TTL = MAX_ROUTE_LEN & increment Unique_ Identification
Assign trust points to path and update RESV_POOL table
Calculate metric and update R.T.
Y
Is QOS_LOST message is heard
Send packets and piggyback Trust pts. of used path to all nodes
N
Is RREP heard
Y
Is any QOS_LOST message heard
Y
N
N
Is 2(wait_timeout) over
N
Y
Drop packets in Buffer
Are there more packets to be sent
Y
N
Is PDR <= Thershold
N
Update lifetime of Min_bandwidth in RESV_POOL table to current_time + Resv_timeout
Y
Update lifetime of Min_bandwidth in RESV_POOL table to current_time + BAD_LINK_LIFETIME at the end of which route is removed from table

Figure 3.2.1 Flowchart for processing done by a source node in DSR domain

3.2.1 Introduction

Whenever a node in an ad hoc domain needs to send packets to any destination which is not in its domain, a RREQ is generated, as explained in Section 2. However, it should be noted that if the source node has a path in the routing table which is fresh enough, and is also in the RESV_POOL table (this would confirm support for requested bandwidth), then there is no need to initialize the routing protocol. When the LE (Local Egress) node fails to find a path to a destination, it generates a universal route request. The user is requested to provide the maximum delay and minimum bandwidth in the RREQ packet, depending on the application in use. Additionally, the RREQ will also contain the timestamp and priority (for calculation of priority refer Section 4) of traffic. FE (Foreign Egress) nodes now check their routing table and the RESV_POOL table to see if they have a path available for the destination in the RREQ packet.

Each node on the ad hoc network will assume different roles depending on whether they need to send packets or just relay packets. Along with those roles, the role of a source node will be interchangeably used with a FE node and an original source node. The reason for such an overlapping definition of the source node role is that for intra domain communication, the source node will be seen as the original source itself; but for inter domain communication, the source is the FE node. Processing done by nodes which are the destination itself will be covered under the relay nodes. Proper flow diagrams for the source node in a DSR domain, is as shown in Figure 3.2.1.

3.2.2 Processing at Source Node

Consider that the source node needs to send some packets to a destination. In such a scenario, the node first decides if the routing process has to be initialized by checking for available routes in the RT and Table 1. The source node is required to calculate the priority of traffic (refer Section 4 and Table 4 for calculation). In the event of a route not being present, data is placed into buffers and a RREQ with a unique identification ID is generated with TTL = 1, priority, timestamp, maximum delay and minimum bandwidth requirements. A wait timer is started to timeout the RREQ in case no RREP is heard for it. The timer is taken from the RFC specifications of DSR, and it is RING0_REQUEST_TIMEOUT (30 seconds). Whenever a QOS_LOST message is heard, the node is marked as a BE node and a BE timer is initialized. This node is not used for sending traffic for same or higher priority levels until the BE timer expires and an entry is updated for it in Table 1. The BE timer is a unique way of accommodating the dynamism of ad hoc networks, where links can get overloaded for a while but become free later on.

Here is a packet format for a QOS_LOST message:

Bandwidth_requested	Available_bandwidth ready to share	Source (node ip address)	Destination (original source ip address)

Figure 3.2.2 QOS_LOST message

Whenever a RREP message is heard back from the destination, the source route is assigned trust points (refer Table 3 and Section 4 for calculations). The route discovered is put in the routing table after a path metric calculation is done, and also in the RESV_POOL Table; and consequently, used for packet transmission. On the first packet sent toward a destination using this path, the source node sends out trust points to that destination so that all the relay nodes update their Table 1 with current trust points of the complete path. If the node does not hear any RREP, another RREQ is sent out with another unique identification ID and TTL = MAX_ROUTE_LEN (15 nodes as defined in RFC). As previously discussed, the QOS_LOST message is processed in a similar fashion. At this point, even with double the wait timeout value, if there is no RREP heard, then packets are dropped from the buffer.

A method has been proposed for relay nodes to have some accountability to the source node with respect to promises of QOS support. A relay node may not be able to support traffic after establishing a trust relationship with the source node for the traffic. In situations like this, some flexibility has been given to relay nodes to relinquish any previously made promises. Relay nodes can drop out of the promise made by sending a QOS_LOST message back to the original source, should they become overburdened by the traffic. The source node will update an entry into the RESV_POOL table (refer Table 1) for this node and change its trust points accordingly. At this point, if the source route has any alternative path to the destination which can support the traffic, then that can be used, if not, the route discovery procedure needs to be initialized. Once the source node has sent all packets to the destination, it calculates the PDR (Packet Delivery Ratio) and matches it against threshold values (refer Table 2) for tweaking trust points (refer Table 3) depending on their performance. A path which meets the threshold requirements for PDR will have their lifetime (refer Table 1) increased by a RESV timeout. On the other hand, if a path fails to fulfill the threshold requirements, it is timed out from Table 1 after a period of (current time)/2.

3.2.3 Processing at Relay Nodes

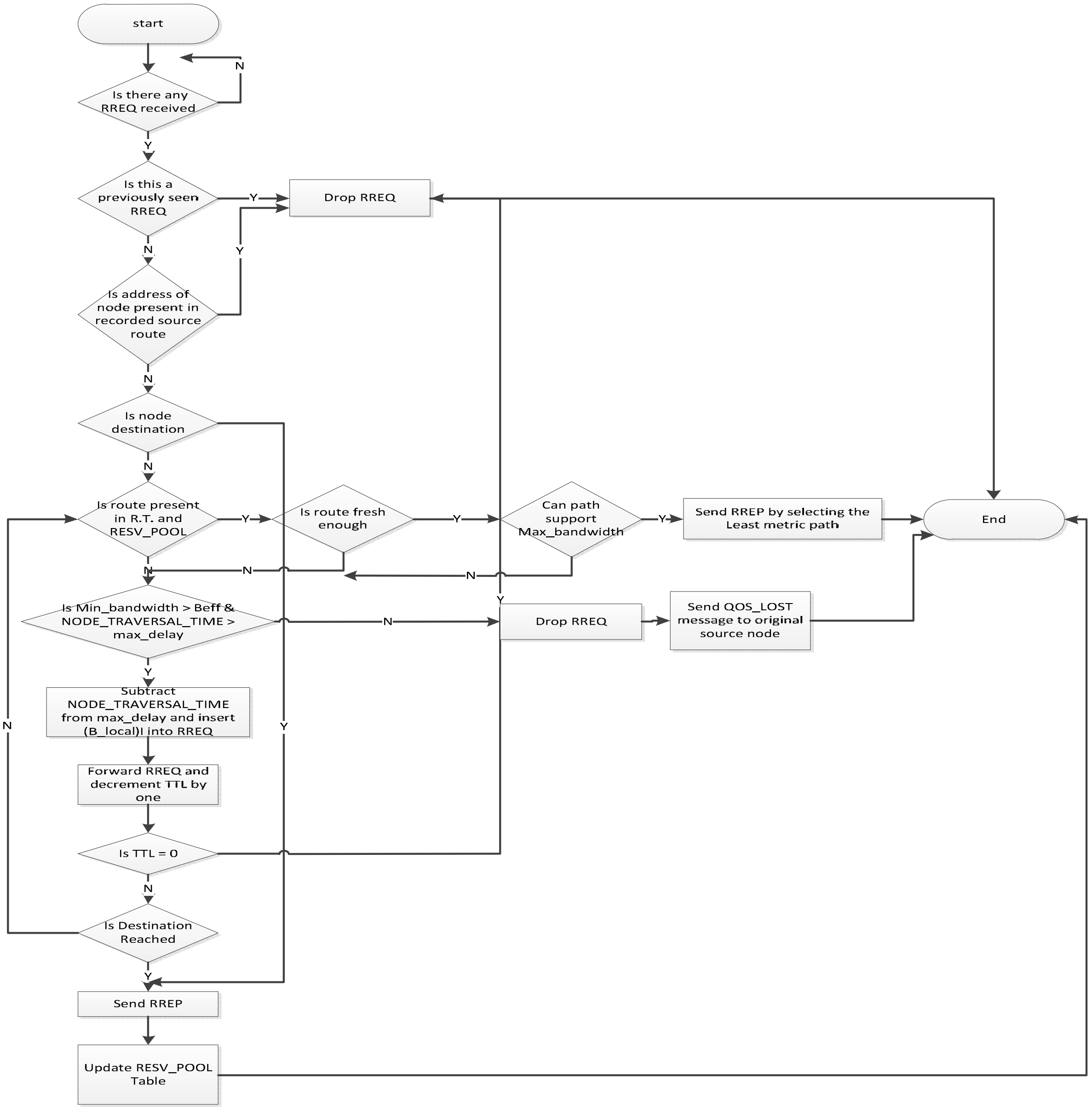

start
Is there any RREQ received
N
Y
Is this a previously seen RREQ
Y
Drop RREQ
Y
N
Is address of node present in recorded source route
N
Is node destination
N
Is route present in R.T. and RESV_POOL
Y
Is route fresh enough
Y
Can path support Max_bandwidth
Y
Send RREP by selecting the Least metric path
End
N
N
N
Y
Is Min_bandwidth > Beff & NODE_TRAVERSAL_TIME > max_delay
N
Drop RREQ
Send QOS_LOST message to original source node
Y
Subtract NODE_TRAVERSAL_TIME from max_delay and insert (B_local)I into RREQ
N
Y
Forward RREQ and decrement TTL by one
Is TTL = 0
N
Is Destination Reached
Y
Send RREP
Update RESV_POOL Table

Figure 3.2.3 Flowchart for processing done by a relay node in DSR domain

Consider that a node comes across an RREQ for a target address. First, the node needs to check and see if it has already seen the RREQ previously, if so, the packet is dropped. The uniqueness of each RREQ packet is defined by the combination of an identification ID and source address. To avoid routing loops, nodes check to see if their home IP address is already present in the source route or not. A node then looks into its route cache and RESV_POOL table, searching for the route with required resources to forward traffic of specified priority. At this point, if a valid route is found, the node sends an RREP after making sure that the route information is fresh enough to support the incumbent traffic. Thus, maintaining a trust relationship with neighboring nodes in a MANET would help relay nodes to reduce computations and the processing of packets. Suppose a route is not present with a relay node, it will then compare the requested minimum bandwidth to its' available link capacity (reservable bandwidth) and decide if the request can be supported. During this process the nodes also verify if delay requirement can be met. Delay requirement is met only if NODE_TRAVERSAL_TIME is less than the requested maximum delay. NODE_TRAVERSAL_TIME (assumed to be 40 milliseconds) is a conservative estimate of the average one hop traversal time for packets and should include queuing delays, interrupt processing times and transfer times. Relay nodes subtract the NODE_TRAVERSAL_TIME from the requested delay and forward the packet further. After making sure both the delay and bandwidth requirements can be met by the node, the RREQ is further forwarded. Any node which is incapable of providing the requested resources will send a QOS_LOST message to the original source. The TTL field in the IP header is used to curb the ever propagating RREQ packets in an ad hoc network. When the destination node is reached it generates a RREP. The RESV_POOL table is fed with the current path information and once the source node sends out the trust points, those are updated as well.

TABLE 1

RESV_POOL Table for DSR

Min Bandwidth	Original source	Priority	Current lifetime (min)	B.E. node IP address	Single trust pt.	Group trust pt.
		Check			Check	Check
		Priority			Trust	Trust
		Table			Ratio	Ratio
					Table	Table

TABLE 2

Priority Table

Priority	Resv_timeout (min)	Retry limit	Bandwidth Threshold (%)	B.E. Timer (min)
4	40	3	90	20
3	30	4	85	15
2	20	5	80	10
1	10	6	75	5

TABLE 3

Trust Ratio Table

Priority	Retry Limit	Trust Ratio (+)	Trust Ratio (-)
4	0	245	X
	1	240	X
	2	235	X
	Timeout	X	245
3	0	245	X
	1	235	X
	2	230	X
	3	225	X
	Timeout	X	240
2	0	235	X
	1	230	X
	2	225	X
	3	220	X
	4	210	X
	Timeout	X	235
1	0	230	X
	1	225	X
	2	220	X
	3	215	X
	4	210	X
	5	205	X
	Timeout	Timeout	230

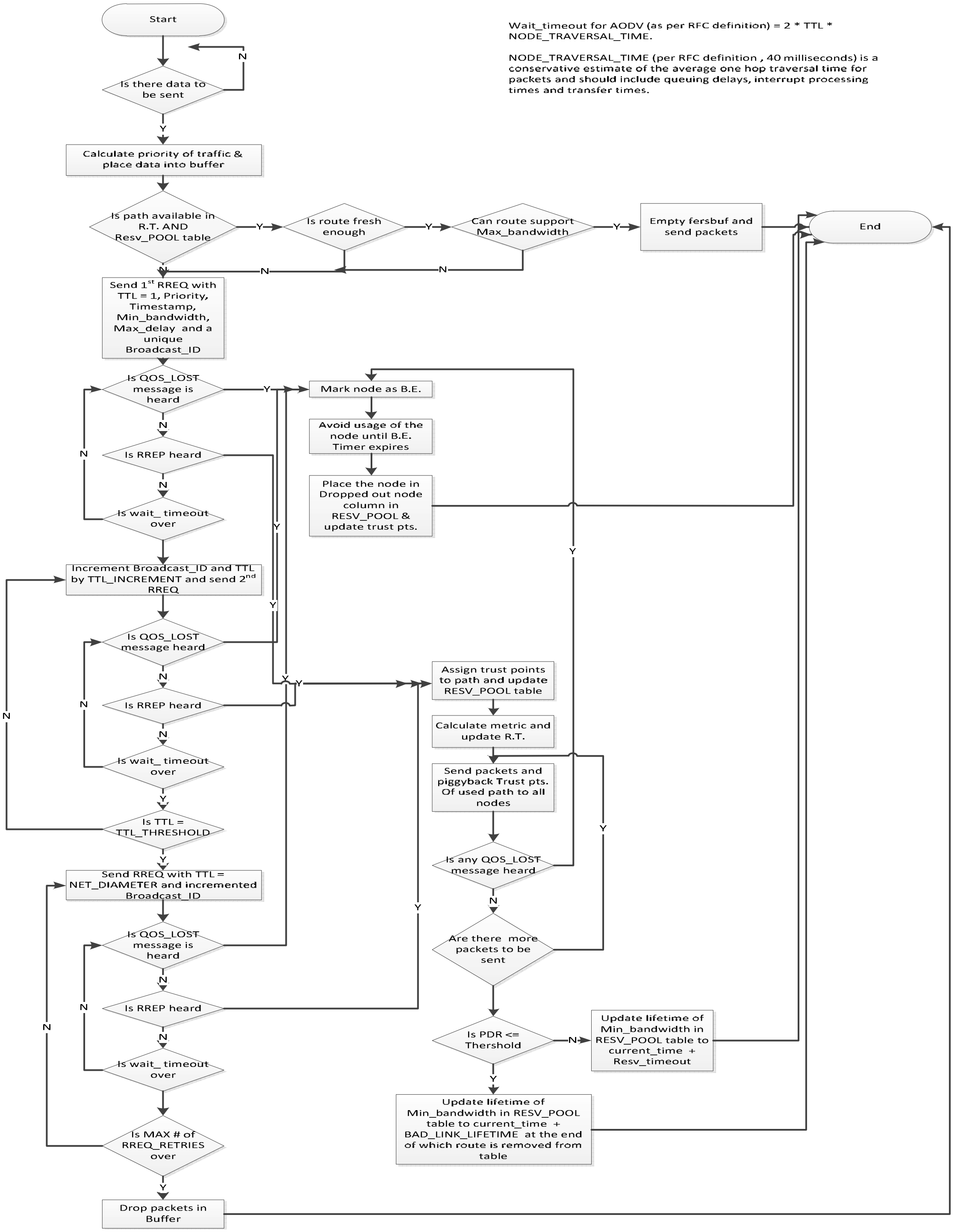
TABLE 4

Priority Calculation Table

Priority		Delay (ms)			Bandwidth (Mbps)		
Calculated	Max	User Req	Max	Weight	User Req	Max	Weight
3.04	4	43	300	0.75	47	100	0.25

Due to the presence of the RESV_POOL table and routing table, nodes will need to use less horsepower to find routes to a destination with specific QOS requirements. Although there would be some delay in initial route discovery, the prospect of getting QOS satisfaction for applications belittles the delay. This author has proposed the idea of making the nodes in an ad hoc network more intelligent by developing trust relationships among them. Moreover, once a node already knows of a path which can support incumbent traffic, it does not have to initialize the route discovery.

3.3 AODV as a Foreign Domain


Start
Is there data to be sent
N
Y
Calculate priority of traffic & place data into buffer
Is path available in R.T. AND Resv_POOL table
Y
Is route fresh enough
Y
Can route support Max_bandwidth
Y
Empty fersbuf and send packets
End
N
N
N
Send 1st RREQ with TTL = 1, Priority, Timestamp, Min_bandwidth, Max_delay and a unique Broadcast_ID
Is QOS_LOST message is heard
Y
Mark node as B.E.
Avoid usage of the node until B.E. Timer expires
Place the node in Dropped out node column in RESV_POOL & update trust pts.
N
Is RREP heard
N
N
Is wait_ timeout over
Y
Y
Increment Broadcast_ID and TTL by TTL_INCREMENT and send 2nd RREQ
Y
Is QOS_LOST message heard
N
Y
Y
Assign trust points to path and update RESV_POOL table
Is RREP heard
N
N
N
Is wait_ timeout over
Calculate metric and update R.T.
Y
Send packets and piggyback Trust pts. Of used path to all nodes
Y
Is TTL = TTL_THRESHOLD
Y
Send RREQ with TTL = NET_DIAMETER and incremented Broadcast_ID
Is any QOS_LOST message heard
N
Y
Is QOS_LOST message is heard
N
Are there more packets to be sent
N
Is RREP heard
N
N
Is wait_ timeout over
Is PDR <= Thershold
N
Update lifetime of Min_bandwidth in RESV_POOL table to current_time + Resv_timeout
Y
Update lifetime of Min_bandwidth in RESV_POOL table to current_time + BAD_LINK_LIFETIME at the end of which route is removed from table
Is MAX # of RREQ_RETRIES over
Y
Drop packets in Buffer

Wait_timeout for AODV (as per RFC definition) = 2 * TTL * NODE_TRAVERSAL_TIME.

NODE_TRAVERSAL_TIME (per RFC definition , 40 milliseconds) is a conservative estimate of the average one hop traversal time for packets and should include queuing delays, interrupt processing times and transfer times.

Figure 3.3.1 Flowchart for processing done by a source node in AODV domain

Whenever a node in an ad hoc domain needs to send packets to any destination which is not in its domain, a RREQ is generated as explained in Section 2. Although, it should be noted that if a source node has a fresh path in the routing table and also in the RESV_POOL table (this would confirm support for requested bandwidth), there is no need to initialize the routing protocol. When the LE (Local Egress) node fails to find a path to a destination, it generates a universal route request. The user is requested to provide the maximum delay and minimum bandwidth in a RREQ packet, depending on the application in use. Additionally, a RREQ also contains timestamp and priority (for calculation of priority, refer Section 4) of traffic. The FE (Foreign Egress) nodes now check their routing table and RESV_POOL table to see if they have a path available for the destination in the RREQ packet.

Each node on the ad hoc network will assume different roles, depending on whether they need to send packets or just relay packets. Along with those roles, the role of a source node will be interchangeably used for a FE node and an original source node. The reason for such an overlapping definition of a source node role is that for intra domain communications, a source node will be seen as an original source itself; however, for inter domain communications, the source will be the FE node. Processing done by nodes which are the destination itself will be covered under the relay nodes. AODV is not a source routing protocol like DSR; and therefore, does not accumulate routes in packet, thus reducing the load on nodes. However, due to our strict criteria for meeting the QOS requirements, and most importantly, maintaining a trust relationship among nodes, the protocol is tweaked to accumulate source IP addresses of relay nodes on the path. Another point of difference between DSR and AODV, is that AODV is a single path protocol and maintains only one path per destination in its RT. Proper flow diagrams for the source node in an AODV domain are as shown in Figure 3.3.1.

3.3.1 Processing at Source Node

Consider that the source node needs to send some packets to a destination. In that scenario, the node must first decide if the routing process has to be initialized by checking for available routes in the RT and Table 5. The source node is required to calculate the priority of traffic (refer Section 4 and Table 4 for calculation). In the event that a route is not present, data is placed into buffers and a RREQ with a unique broadcast ID is generated with TTL = 1 (TTL_START, as defined in RFC), priority, timestamp, maximum delay and minimum bandwidth requirements. A wait timer is started to timeout the RREQ in case no RREP is heard for it. The timer is taken from the RFC specifications of an AODV and it is 2 * TTL * NODE_TRAVERSAL_TIME. Whenever a QOS_LOST message is heard, the node is marked as a BE node and a BE timer is initialized. This node is not used for sending traffic for same or higher priority levels until the BE timer expires and an entry is updated for it in Table 5. The BE timer is a unique way of accommodating the dynamism of ad hoc networks where links can get overloaded for awhile but become free later on.

Here is a packet format for a QOS_LOST message:

Bandwidth_requested	Available_bandwidth ready to share	Source (its own ip address)	Destination (original source ip address)

Figure 3.3.2 QOS_LOST message

Whenever a RREP message is heard back from the destination, the source route is assigned trust points (refer Table 3 and Section 4 for calculations). The route that is discovered will be put in the routing table, after the path metric calculation is done, and RESV_POOL table, then used for packet transmission. On the first packet sent toward the destination using this path, the source node sends out trust points to the destination so that all the relay nodes update their Table 5 with current trust points of the complete path. If the node does not hear any RREP, another RREQ is sent out with a unique broadcast ID and TTL = TTL + TTL_INCREMENT (increment value is 2 as defined in RFC). The AODV does an expanding ring search to optimize performance of the routing protocol. As previously discussed, the QOS_LOST message and RREP is processed in a similar fashion. Once the value of TTL reaches TTL_THRESHOLD (value is 7 as per RFC definition of the protocol), the subsequent RREQs are sent with TTL = NET_DIMATER (value is 35 as per RFC definition) until the max number of RREQ_RETRIES (value is 2 as per RFC definition) is over. At this point, even with the max number of RREQ_RETRIES exhausted, if there is no RREP heard, the packets are dropped from the buffer.

A method has been proposed for relay nodes to have some accountability to the source node with respect to promises of QOS support. A relay node may not be able to support traffic after establishing a trust relationship with the source node for the traffic. A situation like this has been accommodated in the schema by the author, and some flexibility has been given to the relay nodes to relinquish any previously made promises. Relay nodes can drop out of the promise, accomplished by sending a QOS_LOST message back to the original source, if they become overburdened by the traffic. The source node will update an entry into the RESV_POOL table (refer Table 5) for this node and change its trust points accordingly. At this point, since an AODV is a single path protocol and a route discovery procedure needs to be initialized. Once the source node has sent all packets to the destination, it will calculate the PDR (Packet Delivery Ratio) and match it against threshold values (refer Table 2) for tweaking trust points (refer Table 3), depending on their performance. Basically, a path which meets the threshold requirements for PDR, will have their lifetime (refer Table 5) increased by RESV_timeout. On the other hand, if a path fails to fulfill the threshold requirements, it will be timed out from Table 1 after a period of BAD_LINK_LIFETIME (timer as per RFC definition).

Due to the presence of the RESV_POOL table and routing table, nodes will need to use less horsepower to find routes to a destination with specific QOS requirements. Although there would be some delay in the initial route discovery, the prospect of getting QOS satisfaction for the application belittles the delay. This author has proposed the idea of making the nodes in an ad hoc network more intelligent by developing trust relationships among them. Moreover, once a node already knows of a path which can support incumbent traffic, it does not have to initialize the route discovery.

3.3.2 Processing at Relay Nodes

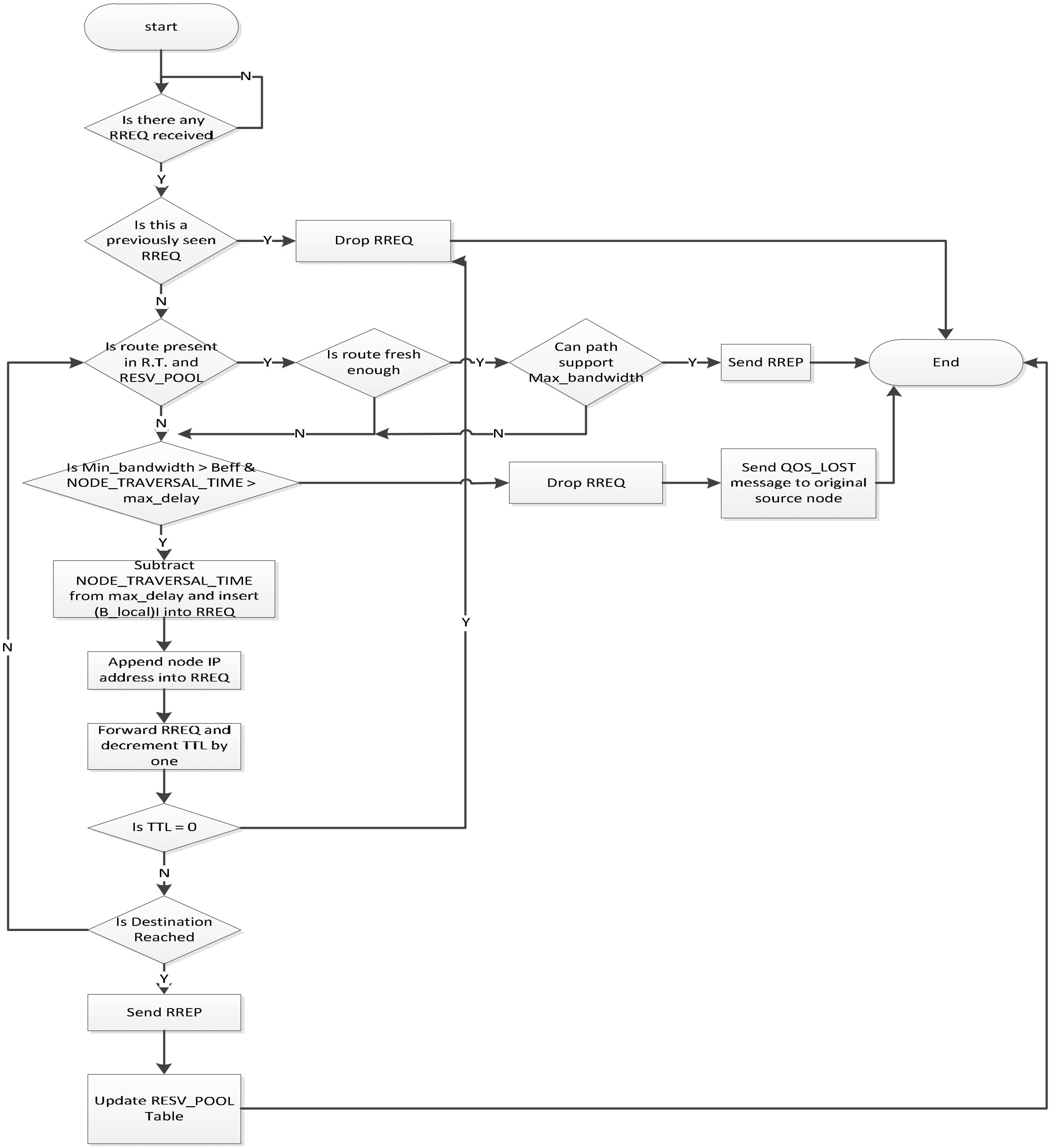

start
N
Is there any RREQ received
Y
Is this a previously seen RREQ
Y
Drop RREQ
N
Is route present in R.T. and RESV_POOL
Y
Is route fresh enough
Y
Can path support Max_bandwidth
Y
Send RREP
End
N
N
N
Is Min_bandwidth > Beff & NODE_TRAVERSAL_TIME > max_delay
Drop RREQ
Send QOS_LOST message to original source node
Y
Subtract NODE_TRAVERSAL_TIME from max_delay and insert (B_local)l into RREQ
Y
N
Append node IP address into RREQ
Forward RREQ and decrement TTL by one
Is TTL = 0
N
Is Destination Reached
Y
Send RREP
Update RESV_POOL Table

Figure 3.3.3 Flowchart for processing done by a relay node in AODV domain

Consider that a relay node has seen a RREQ for a target address. First, a node needs to check and see if it has already seen the RREQ previously, if so, the packet is dropped. The uniqueness of each RREQ packet is defined by the combination of a broadcast ID and source address. To avoid routing loops, nodes check to see if their home IP address is already present in the source route or not. A node then looks into its route cache and RESV_POOL table, searching for the route with required resources to forward traffic of specified priority. At this point, if a valid route is found, the node sends a RREP after making sure that the route information is fresh enough to support the incumbent traffic. Thus, maintaining a trust relationship with neighboring nodes in a MANET would help relay nodes reduce computations and the processing of packets. Suppose a route is not present with a relay node, it will then compare the requested minimum bandwidth to its' available link capacity (reservable bandwidth) and decide if the request can be supported. Another verification which the node does is to establish whether delay requirements can be met. NODE_TRAVERSAL_TIME (value is 40 milliseconds, as per RFC definition) is a conservative estimate of the average one hop traversal time for packets and should include queuing delays, interrupt processing times, and transfer times. Relay nodes subtract the NODE_TRAVERSAL_TIME from the requested delay and forward the packet on further. The delay requirement is met only if NODE_TRAVERSAL_TIME is less than the requested maximum delay. After making sure both the delay and bandwidth requirements can be met by the node, the RREQ is further forwarded. Any node which is incapable of providing the requested resources will send a QOS_LOST message to the original source. A TTL field in an IP header is used to curb the ever propagating RREQ packets in an ad hoc network. When a destination node is reached it generates a RREP. The RESV_POOL table is fed with the current path information and once the source node sends out the trust points, those are updated as well.

TABLE 5

RESV_POOL Table for AODV

Min Bandwidth	List of participating node IP address	Original source	Priority	Current lifetime (min)	B.E. node IP address	Single trust pt.	Group trust pt.
			Check			Check	Check
			Priority			Trust	Trust
			Table			Ratio	Ratio
						Table	Table

Due to the presence of the RESV_POOL table and routing table, nodes will need to use less horsepower to find routes to a destination with specific QOS requirements. Although there would be some delay in the initial route discovery, the prospect of getting QOS satisfaction for an application belittles the delay. This author has proposed the idea of making the nodes in an ad hoc network more intelligent by developing trust relationships among them. Moreover, once a node already knows of a path which can support incumbent traffic, it does not have to initialize the RP.

CHAPTER 4

MATHEMATICAL MODEL AND ANALYSIS

This chapter is organized as follows: Section 4.1 gives a mathematical formula to facilitate the calculation of priority of incumbent traffic to/from ad hoc nodes. In Section 4.2, a mathematical formula is proposed to find the available link capacity (bandwidth) with a MANET node. In Section 4.3, the formula for delay calculation for the trust based model of Ad Hoc Routing Protocol is shown. Section 4.4 will explain the procedure for the final composite path metric calculation for the schema proposed by the author. In Section 4.5, fuzzy dynamic programming is used to develop a relational database for trust points of nodes, so that they can use it to assign trust points to other nodes. Section 4.6 performs an analytical comparison of RSVP and the trust model with the help of pseudo codes. In Section 4.7, metrics for a final evaluation of the work are presented.

4.1 Priority Calculation

In this research, QOS provisions have been made more granular and user, or more precisely, application friendly. When demand for QOS is generated by users, the point of concern is the maximum delay and minimum bandwidth that can be afforded in order to deliver an acceptable performance of the application in use. The design proposed here lets users select their required delay and bandwidth; and consequently, the node calculates the priority of the flow.

The priority of the flow is calculated as shown in the equation (4.1):

$$P = Max\ \#\ of\ priorities\ [(1 - \frac{user\ delay}{\max delay}) * 0.75\ + (\frac{user\ bandwidth}{\max bandwidth}) * 0.25] \qquad (4.1)$$

The weight which is assigned to the delay is designed to be more than the bandwidth, since multimedia traffic is delay sensitive. Again, these weights are user configurable so users can pick a value depending on the type of constraints placed by the applications they wish to run.

4.2 Available Bandwidth Calculation

IEEE 802.11 has 4 states; namely, busy state, carrier sensing channel busy, virtual carrier sensing busy, and idle state**.** [18]

$$R = \frac{channel\ busy\ period}{Time} \qquad (4.2.1)$$

Where R is the channel utilization ratio, a channel busy period is the sum of the first three states among the total number of states of IEEE 802.11. The time for which a node listens to the medium is the total time T, namely, a channel busy period. A constant α is considered for smoothing out the effective bandwidth estimated. Let us assume that at time t, the channel utilization ratio is R_t and at a previous interval (t-1) it was $R_{t-1,}$ then total R_t is given as:

$$R_t = \alpha(R_{t-1}) + (1 - \alpha)R \qquad (4.2.2)$$

T ($0<R_t<1$) and $\alpha =$ smoothing constant $\epsilon[0,1]$

$$Beff = W(1 - R_t) \tag{4.2.3}$$

Where B is the available bandwidth of the node at time t, and W is raw channel bandwidth

$$Beff < R_i * L_i \tag{4.2.4}$$

Where $R_i$transmission is rate and $L_i$is the minimum frame size.

4.3 Delay Calculations for Trust Based Model of Ad Hoc Routing Protocols

a) Delay calculation for DSR protocol:

$$\mathrm{D(p) = DSR\ Route\ discovery\ delay + RTT\ (ICMP)\ \ of\ path + Negotiation\ Delay} \tag{4.3.1}$$

As source nodes have inserted a timestamp field into the RREQ packet, on receiving the RREP, a node can know exactly the time taken for route discovery by subtracting current time from the value of timestamp in a RREP packet. A negotiation delay would be a function of the processing power of the nodes in the network, time taken for ping packets to traverse back to the source node, retry limit and retry interval. The processing delay of each node can be safely neglected considering the CPU speed of modern processors. For example, if a 32-bit CPU tries to process a packet size of around 160 bytes, with a speed of 1 Gigahertz, it only takes a few nanoseconds. Negotiation delay can be given as:

$$\mathrm{D}n = Max(R * RTTsi + (R - 1) * Tretry) \tag{4.4}$$

Where RTT is the round trip time taken by the ICMP ping packets to travel all the way from ith node (relay) back to the source, R is the retry limit per priority, and $T_{retry,}$ the retry interval, which is estimated as 2RTT.

b) Delay calculation for AODV protocol

$$\mathrm{D(p) = AODV\ Route\ discovery\ delay + RTT\ (ICMP) of\ path + Negotiation\ Delay} \qquad (4.3.2)$$

4.4 Final Metric Calculation

Bandwidth of the path is a concave function of the nodes' effective bandwidth [9], i.e. the node with minimum bandwidth is considered as the path bandwidth. Similarly, trust of path is decided by the least trustworthy node's assigned value. Delay has already been calculated in the above sub-section.

The final metric calculation for final path selection is given by equation (4.5):

$$f(p) = \frac{\big(B(p) * T(p)\big)}{D(p)}$$

$$f(p) = (\min[B_i(p), B_j(p), B_k(p) \ldots\ldots, B_n(p) \qquad (4.5)$$

Where f(p) is the final metric of path, B(p) is the available bandwidth of the path, T (p) is the trust value of the path and D(p) is the delay of the path. i ,j,k…. n being the nodes along the path.

4.5 Fuzzy Dynamic Modeling of Trust Points

In Ad hoc networks, trust is represented by the relationships built as nodes interact with each other. This can be abstracted as the association between a trusting node and a trusted node. Trust relationships are determined by the rules to evaluate the evidence with a quantitative way, generated by the previous behavior of a node (as maintained in RESV_POOL_TABLE). Accordingly, fuzzy logic is a methodology to build a mapping from a particular input to a logical output, which provides the basis for several decisions made. Because of the movement of nodes and the dynamic nature of the links in wireless networks, the value of trust in a MANET always has a natural uncertainty and incompleteness; and therefore, evaluation models of trust focus on the collection and the quantization of the dynamic information. The trust association between two nodes can be a direct result of any favor done by an ad hoc node to its (maybe) neighboring node or an indirect result of favors done which may include priority packet forwarding. Whatever the case may be, trust is a naturally fuzzy concept.

Now we will let Tij represent the trust value assigned from node i to node j. Tij can be desired from the previous record per the RESV_POOL_TABLE at the two nodes. According to [28], [29], a formula can be developed as per equation (4.6):

$$T_{ij} = \rho\left(\frac{1+\sum_{k=1}^{I} S_k}{2+\sum_{k=1}^{I} N_k}\right) + (1-\rho)\frac{\alpha E_p + \beta C_q + \gamma M_t}{\alpha+\beta+\gamma} \qquad (4.6)$$

Sk denotes during the most recent I times interactions between the nodes, the real total trust count at the kth time between node i and node j. Nk presents the expected trust point of node i at the kth time. Node i often repeats the calculation of trust at different time instances. Let

Sk denote the time when node i assigns trust to node j. At time k, node i observes that node j performs the action times upon the request of performing the action times. Hence, we have Nk≥Sk. The trust entries at RESV_POOL_TABLE are made for a period of time; and thus, they carry less importance than the assigned trust values. Ep, Cq, Mt represent the node information at the current time. Ep is the energy consumption information which represents the power resources as the mobile embedded system. Cq is the bandwidth utilization percentage, and Mt is the memory utilization percentage which represents the storage resources. α, β and γ are all positive integers which represent the weight values of the three aspects, ρ ∈ [0,1], is the variable coefficient. The proportion of the history records and the node condition can be tuned with it to let the formula be more practical and scalable.

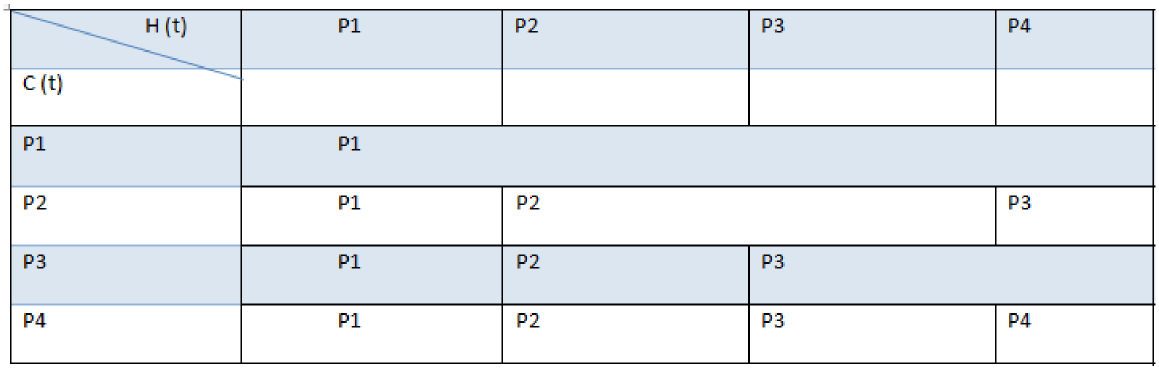
Trust Evaluation with the help of Fuzzy Logic

When the node *i* makes a request to node *j* for transmission of data packets of certain priority, node *i* has trouble deciding whether node *j* can provide the service at that time. In other words, node *i* is uncertain if node *j* can be trusted to forward priority traffic with the correct QOS values. Such a situation can be handled by node *i* from the interaction records of node *j* in the form of RESV_POOL_TABLE. Let *C(t)* represent the capability of the requested node (node *j*) to provide packet transfer services at time *t*, which includes the remnant bandwidth at that point. Let *H(t)* represent at time *t*, the history behavior of node *j* to offer data transfer services between the past time intervals, such as packet-delivery ratio (PDR). Let *TL(t+1)* refer to the node's trust level at time *t+1*. Assume the fuzzy member function of *C(t)* consists of four fuzzy sets: *Priority_1 (P1)*, *Priority_2 (P2)*, *Priority_3 (P3)*, and *Priority_4 (P4)*. The fuzzy membership function of *H(t)* and *TL(t+1)* consists of four different levels of fuzzy sets: *Priority_1 (P1)*, *Priority_2*

(P2), *Priority_3 (P3)*, and *Priority_4 (P4)*. According to the social control theory [35], the fuzzy inference rules are given in Table 6.

TABLE 6

Fuzzy Based Trust Table



H (t) / C (t)	P1	P2	P3	P4
P1	P1			
P2	P1	P2		P3
P3	P1	P2	P3	
P4	P1	P2	P3	P4

The rules in Table 6 give a direct mapping from $H \times C$ to TL. It is based on the careful analysis of the node's current and previous conditions. When an over utilized node does not have enough CPU cycles, buffer memory, or available bandwidth at link level to relay data; it may also be untrustworthy in the next time interval, irrespective of its historic trust values, which may be high. The formation of the table makes sure that the assigning of trust values of the nodes are more stable and more importance is given to the current state of the node, instead of previous ones. This is a basic rule followed in the above table and the actual relationship can be formulated with R_l:

(4.7)

And h H, c C and u TL

$$R_1(h,c,u) = H\ (h) \wedge C\ (c) \wedge TL\ (u) \tag{4.8}$$

For all n rules, the (fuzzy) inference relation can be given as in equation (4.9):

$$R\ (h,c,u) = \ \vee_{l=1}^{n} R_l\ \ (h,c,u) \tag{4.9}$$

For every pair of given input *H** and *C**, with the help of general relationship *R*, the output must be formulated as per equation (4.10):

$$TL^{*}\ = (H^{*}\ \times C^{*})\ \circ\ R \tag{4.10}$$

Hence, with the maximal membership degree approach, the trust value u*∈[0,1], can be calculated with the fuzzy methods. This is the basic model with two aspects *C(t)* and *H(t).*

4.6 Analytical Comparison With the Help of Pseudo Code

Comparison of the trust model for routing protocols and RSVP is a worthwhile exercise, since both of them share a common goal of achieving QOS aware path discovery for nodes in an ad hoc network.

The scenario, where a route is already present in both the RT and RESV_POOL_TABLE, requires much less processing; and thus, a comparison with RSVP and the trust based model is done for the case when the route is unknown. Assume that a source node has some packets to be sent to a particular destination, but it does not have a route in the RT or RESV_POOL_TABLE.

Here is how a pseudo code for the QOS aware route discovery, consisting of trustworthy nodes, is done.

```
main (Route_Trust-model_discovery)
{
     Do {if (Route !present in RESV_POOL_TABLE && R.T.)
                    if (RREQ !present in RREQ_Seen_Table)

                    if (Beff > Minimum_Bandwidth && NODE_TRAVERSAL_TIME <
        Maximum_Delay)

                   { TTL.Current = TTL – 1 ;

                   Maximum_Delay.Current = Maximum_Delay –
        NODE_TRAVERSAL_TIME;

                    if (Beff.node < Bandwidth.Current)

                     { Bandwidth.Current = Beff.node;}

                    } Rebroadcast RREQ;

                     else

                     drop RREQ;

                     Send QOS_LOST message to source node;

          } while (Relay_node.IP_Addresss != Destination_node.IP_Address)

        send RREP;
}
```

RSVP is a layer 4 protocol which operates along with a routing protocol to provide QOS fulfillment of traffic flows in a network. In RSVP protocol, resource reservation is requested by the receiver host application and passed onto RSVP local process. The destination node initiates a RESV message to be sent along the path as discovered by the routing protocol (in backwards direction), back to the source. Every node in the path has two modules, viz admission control and policy control, to make decisions for the resource reservation in the network. The two level controls for RSVP is to determine if enough resources (admission control) are present to support the flow and if nodes have administrative rights (policy control) to accept reservations. For comparison purposes with the proposed trust based model, it can be safely assumed that all nodes in the network have administrative rights to reserve resources. If the admission control part succeeds, RSVP daemon sets parameters of the packet classifier and packet scheduler to get requested QOS. If admission control fails, the RSVP daemon returns an error notification to the host application. An RSVP reservation request is made up of a flowspec and a filterspec, which determines the amount of QOS needed, and identifies data packets which need requested QOS, respectively.

This is a pseudo code for RSVP protocol functioning above routing protocols.

```
main (Route_RSVP_discovery)
{
         do{ if (Route !present in R.T.)
                if (RREQ !present in RREQ_Seen_Table)
                     Rebroadcast RREQ;
              else
                   drop RREQ;
            } while (Relay_node.IP_Adresss != Destination_node.IP_Address)

            send RREP;
            Reverse data.path;
            Send RESV.message to source_node.IP_Address;

        do { Relay_node (Accept_RESV.mesage)
                        { if (Result.admission_control == True)
                          Set parameters in node's packet scheduler;
                          Set filter_spec in RSVP_daemon for traffic;
                          service.class_flow is set into RSVP_daemon;
                          Add Tspec for defining specific data_flow;
                          Reserve Rspec_value for the data_flow at link layer;
                          else (Reject_RESV.message)
                          send Resv_Error packet to destination_node;
                        }
          } while (Relay_node.IP_Address != Source_node.IP_Address)
}
```

4.7 Metrics

4.7.1 Comparison of Delay Constraints of a Trust Model and RSVP

When the trust model is used, to get a QOS aware path of trustworthy nodes, the important factors such as, the presence of route in the routing table and RESV_POOL_TABLE, become functions of probability. In order to correctly estimate the delay incurred when the trust model is used, the uncertainty of the presence of routes should be taken into consideration.

Now, let us define P (i) as the probability of route being present in the routing table of the source node and P (r) being the probability of route being present in the RESV_POOL_TABLE. Hence, 1- P (i) and 1 – P (r) becomes the probability of route not being present in the route table and RESV_POOL_TABLE, respectively. Whenever a route is not present in the RT and RESV_POOL_TABLE, the source node has to perform a complete route discovery procedure for routing protocol (as explained in Chapter 3). A case where route is present in the pool table but not in the RT is unrealistic, since the route first enters the RT and thereafter, a QOS check is performed to insert route in the pool table. A case where route is present in RT, but not in the pool table is possible, since the reservation for that route may have timed out. In that case, a complete route discovery needs to be done because if we try to use the route just by looking at the RT, we may end up passing QOS sensitive data on a route which may not have enough resources to support audio/video traffic.

As per equation (4.3.1) and (4.3.2), it is seen that the delay of the path based on the trust model in DSR, and AODV routing protocol is as follows:

$$\mathrm{D(p)} = \text{DSR Route discovery delay} + \text{RTT (ICMP) of path} + \text{Negotiation Delay}$$

$$\mathrm{D(p)} = \text{AODV Route discovery delay} + \text{RTT (ICMP) of path} + \text{Negotiation Delay}$$

Consequently, the general formulae for total delay for the path discovery in the proposed model can be given as:

$$\text{Delay (path)} = \{(1\text{- P (i)}) + (1\text{-P (r)})\}\ \text{D (p)} \tag{4.11}$$

$$\text{Delay (RSVP)} = \{\text{P (i)}\}\ \text{RSVP delay} + \{1\text{- P (i)}\}\ \{\text{Route discovery delay} + \text{RSVP delay}\}$$

$$\text{Delay (RSVP)} = \text{RSVP delay} + \{1\text{- P (i)}\}\ \text{Route discovery delay} \tag{4.12}$$

Hence, from the above formulae, it is clear that RSVP is not as suitable for delay sensitive traffic as the trust model proposed. Other than normal route discovery delays, RSVP has added delays for the route.

4.7.2 Comparison of Overhead Constraints of Trust Model and RSVP

As per the definition of overhead of any protocol:

$$Protocol\ Overhead\ = \frac{Total\ number\ of\ control\ packet\ sent}{Total\ number\ of\ sent\ packets} \tag{4.13}$$

In a wireless network, the cost of internal processing at a node is negligible in terms of energy consumption and CPU cycles compared to the cost of transmission (as described in previous sections). This author has looked at the overhead caused by extra messages. Hence, total overhead O in a network of n nodes is defined as follows. Each transmission tx of all control messages is considered along with the origination or reception.

$$O\ (path) = \frac{\sum_{i=1}^{n} QOS\_LOST}{\sum_{i=1}^{n} RREQ + \sum_{i=1}^{n} RREP + \sum_{i=1}^{n} RRER} \tag{4.14}$$

$$O\ (RSVP) = \frac{\sum_{i=1}^{n} RESV + \sum_{i=1}^{n} PATH + \sum_{i=1}^{n} RESV\_ERR}{\sum_{i=1}^{n} RREQ + \sum_{i=1}^{n} RREP + \sum_{i=1}^{n} RERR} \tag{4.15}$$

This author has used this ratio to determine the extra overhead caused by the trust model, relative to any regular routing protocol’s overhead (AODV or DSR). Clearly, the transmission of a QOS_LOST message is the only extra packet transmitted in the proposed model which adds to the extra overhead. The messages RREQ, RREP and RERR in the case of DSR or AODV, are those control messages which are needed for the proper functioning of any routing protocol in general.

Thus, it can be deduced from the above formulae that RSVP has more overhead than the proposed mechanism, due to more control packets being used for its functioning.

CHAPTER 5

FLUID MODEL

This chapter is organized as follows: Section 5.1 gives an explanation of all the parameters of the system used to develop a working model for analytical purposes. The next Section 5.2, provides information about the algorithm developed for allocating flow in an ad hoc network based on the constraints discussed. Section 5.3 reviews the ways that costs get credited/debited from nodes in order to allow trust based multihop communications.

5.1 System Description

Due to exponentially increasing real time networks along different dimensions, performance evaluation of such wireless networks with the help of outmoded packet-level simulations is getting increasingly difficult. To accommodate the higher levels of abstraction, fluid-level simulation is a great methodology for estimating large network models.

The network is modeled as a set N of ad hoc nodes that are enriched with directional, wireless antennas, with N = |N| being the total number of nodes. A user is an entity which needs to send packets to a particular destination; whereas, a node is a term which encompasses users and in addition has topological meanings such as position, velocity, routing and bandwidth constraints. In the set comprising of N nodes, there is a set S of all source nodes which need to send packets to destinations D. To do this, a set of routes between each pair of source and destination nodes has been discovered, where a route r ∈ N belongs to a subset of ad hoc nodes. Any ad hoc routing protocol, like DSR and AODV, can be used to discover these routes. RS(s) can be defined as the subset of routes that start at source s and RD(d) can be defined as the subset of routes that end at destination d.

The traffic flow along a particular route is given by yr where yr ≥ 0. In this model a node is restricted to only one transceiver. Moreover, the node has limited bandwidth capacities left with it locally. The MAC protocol also plays a role in deciding the amount of traffic a node can send or receive in an ad hoc network.

Flow constraint is modeled by mathematically calculating the total bandwidth usage, where the number of packets that is relayed by a node must be both received and sent.

$$C_j \; = \; \sum_{\substack{r:j\in r,\ and \\ r\in R^S(j)\cup R^D\ (j)}} y_r + \sum_{\substack{r:j\in r,\ and \\ r\nexists R^S(j)\cup R^D\ (j)}} 2y_r \tag{5.1}$$

And, constraining the bandwidth usage is as follows:

$$c_j \;\; \leq \;\; C_{j,} \, \forall \, j \;\in N$$

It is noticeable here that the above defined constraint is not fully capturing the interference issue that may occur in any ad hoc network. It is instead a generalization that tries to make sure that the node cannot send/receive traffic to/from two of its neighbors at the same time.

A key problem in an ad hoc network is the introduction of excessive delay in packets while transiting through forwarding nodes across the network to reach the destination node. Delay is induced in the flow due to various factors such as ambient noise, lousy channels and poor signal strength. Total delay, introduced by a source node when trying to send packets to a relay node j on path r, is be given by the forward delay from source s to node j in the path as $d_1\,(j,r)$.

Intuitively enough, the destination node would only introduce half the round trip time delay which is defined below as T_r.

Hence, to model the delay in an ad hoc network, it is assumed that:

$d_1\,(j,r)$ is the forward delay from the source of route r to the node j in path.

$d_2\,(j,r)$ is the reverse delay from the node j back to source of route r.

T_r would thus give the total round trip delay on the route r.

$$d_1\,(j,r) + \; d_2(j,r) = T_r \;\forall\, r \qquad (5.2)$$

The following formulation would be used to consider delay introduced in the packets in an ad hoc network, due to any node:

$$D_j = \; \sum_{r \,\in\, R^S\,(j)} y_r d_1\;(j,r) + \sum_{r\,\in R^D(j)} y_r\;(\tfrac{T_r}{2}) + \sum_{\substack{r:j\in r\; and \\ r\,\nexists\, R^S(j) \cup\, R^D\;(j)}} y_r\;(d_1(j,r) + d_2\,(j,r)) \qquad (5.3)$$

Delay is constrained at a node by the maximum possible delay in reaching two extremities of the network topology:

$$D_j \;\leq\; \rho_j \;\; \forall\, j \;\in N$$

Where $\rho_j$is dependent on propagation delays in the wireless networks which is basically restricted by NET_DIAMETER*NODE_TRAVERSAL_TIME in any ad hoc domain.

5.2 Algorithm for Flow Allocation

Mechanism is shown here to help nodes make decentralized and informed decisions about the choice of the traffic flow on routes. Decisions are made depending on congestion costs announced by relay nodes. Accordingly, the nodes with a given willingness-to-pay for congestions costs, as announced by the relay nodes, can change their resource usage. This author has tried to accommodate the bandwidth and delay prices to present basic constraints faced in ad hoc networks.

Now, we will define the system in more detail. Every user in the domain has a parameter known as w_s(t), the willingness-to-pay field. The users adjust their self-generated rates of traffic flow on each route r ($where,\ r\ \in\ R^S\ (s)$) with time, so that the following expression governs it:

$$x_s(t) = \sum_{r \in R^S\ (s)} y_r\ (t) = \frac{w_s\ (t)}{min_{r \in R^S\ (s)} \sum_{j \in r} \mu_{jr}\ (t)} \tag{5.4}$$

Parameter $y_r\ (t)$ remains positive only on those routes r that attain a minimum in the denominator of the above fraction. Even in a real ad hoc network only the lowest cost route r would be chosen to forward packets. The cost along every route in the domain would be expressed as follows:

$$\mu_{jr}\ (t) \quad = \begin{cases} d_1\,(j,r)\mu_j^T\ (t) \quad + \quad \mu_j^B(t),\ \ j\ is\ the\ source\ node \\ \{d_1\,(j,r) +\ d_2\,(j,r)\}\mu_j^T(t) \quad + \quad \mu_j^T(t),\ \ j\ is\ the\ relay\ node \\ \frac{T_r}{2}\mu_j^T\ (t) \quad + \quad \mu_j^B(t), \quad j\ is\ the\ destination\ node \end{cases}$$

(5.5)

Schema is further modified to include the priority parameter to help relay nodes earn more money i.e., $M_{jr}(t)$ if they are helping to forward priority traffic from one node to another in the network, shown as follows:

$$M_{jr}(t) = \ \mu_{jr}(t) * P_i \quad where\ \ P_i \in\ \{1,2\ ,3\ ,4 \ldots n\} \tag{5.6}$$

The congestion costs for delay and bandwidth is dynamically adapted as shown below.

We know that the product's rule of derivative of function is as follows, and also that derivative of a constant is zero:

$$\frac{dy}{dx}[f(x) * g(x)] = f(x)g(x)' + f(x)'g(x)$$

$\frac{d}{dx}[L] = 0$ if L is a real number

$$\frac{d}{dt}\mu_j^B(t) * P_i = \ P_i \frac{k\ \mu_j^B(t)}{C_j}\left(c_j\,(t) - \ C_j\right) \tag{5.7}$$

And,

$$\frac{d}{dt}\mu_j^T(t) * P_i \;=\; P_i \frac{k\,\mu_j^T(t)}{\rho_j}\left(D_j\ (t) - \rho_j\right) \tag{5.8}$$

The dependence of the RHS of the above equations on both current cost and the respective capacity is done to scale the dynamics of prices in the ad hoc domain with highly varying overall capacities and costs. Hence, under stable operating conditions of the algorithm, as designed above, it would select routes so that traffic sent by user s would deal with congestion costs along the route r, at a decided rate of $w_s(t)$ per unit time.

5.3 Distributing Congestion Costs

A node in the ad hoc domain would not want to be selfish if the benefits of being selfless are provided as quickly as possible. Hence, incentives have been designed for relay nodes so that they can reclaim the favor done for others as soon as there is a need to send packets of any priority. So far, the allocation of packets to any route is decided by $w_s(t)$ parameter. Consider that user j would be receiving an estimated credit for all of the congestion costs it had to sustain from each source whenever a route crossed node j. Accumulated credit can be reused as compensation to other nodes which would act as relay nodes. This would provide a noticeable drive for nodes to act as relay nodes whenever there would be a huge requirement for voice/data traffic, which helps them earn a large amount of money in their capacity as relay nodes.

Each ad hoc node maintains a credit balance of, $b_s(t)$, with an initial value of one when the node first boots up and joins the ad hoc domain. The accumulated balance is distributed as congestion costs to other relay nodes on the path r. It can be seen that any node may not be able to send more traffic, or have more willingness-to-send, than the amount of total money it has to spend in order to send packets downstream toward the destination. This is expressed with the help of the following rule:

$$w_s(t) = \alpha_s b_s(t) for\ some\ parameter\ \alpha_s > 0$$

Consequently, a node's rate of sending is tied to its leftover credit balance. Credit balance is discounted as a function of time as shown below:

$$\frac{db_s}{dt} = -\beta\ (b_s\,(t) - 1)\ -\ w_s(t) + \textstyle\sum_{r:s\,\in r} y_r \mu_{\ sr}(t) \qquad (5.9)$$

For small positive constant β,the above formulae tends to stabilize the summation of credit balances in the system, ∀ sources s ∈ N, approximately close to the total population size of the domain.

When a user enters the system, the total credit of the system automatically adjusts to the new level of population. When the node leaves the system, the total credit moves toward the leftover population.

CHAPTER 6

DISCUSSION

It is truly said,

"Being a consumer of research is a lot easier than being a producer of research."

This chapter is organized as follows: Section 6.1 explains the implications of the proposed protocol for ad hoc networks. The following section gives the concluding remarks for the work done in this thesis work. In the end, future work is given for the schema.

6.1 Implications

The work done has many important implications in the networking world. Multimedia traffic, which is usually high priority, can be sent through more trustworthy nodes in the network, avoiding a lot of malicious nodes with the help of a trust database maintained at each node. A scan through the RESV_POOL_TABLE will quickly tell the source node if a path in the routing table has enough capacity to handle the incoming packets. A path, which does not have enough resources needed to be used by delay sensitive audio/video traffic, can be avoided. The quintessence of the proposed schema lies in decreasing the delay of obtaining resourceful routes for sending QOS traffic, by adding very little extra overhead (QOS_LOST message) as compared to RSVP (RESV, PATH, RSVP_ERROR). The research done has also tried to capture the dynamism of wireless links by using the BE timer. This also facilitates the best utilization of costly wireless channels and to some extent, gives fairness to the traffic flows. When an ad hoc network needs to be setup for disaster relief, emergency scenarios, or conferences, nodes can easily communicate without supporting two routing protocols in their TCP/IP stack, due to the usage of a universal routing control packet.

This research can be used as a comprehensive solution for the three most important issues arising in an ad hoc network: viz QOS, interoperability and fairness.

6.2 Conclusion

The research done so far on the trust based QOS model can be safely concluded that, as far as QOS requirements are concerned, a higher level of satisfaction is guaranteed than when using RSVP. The amount of time spent by the proposed trust model on the successful transmission of packets will be small due to lesser chances of retransmissions, since every node sticks to the promise it has made or loses trust points. Hence, this model promises to perform better than DSR and AODV to forward multimedia QOS hungry traffic. In the event of the presence of best effort traffic, fairness is considered as well. In the design proposed so far, care has been taken so that best effort traffic does not suffer from resource starvation. With the help of strict metrics, such as delay and overhead, which were used to compare the trust model and RSVP, it was clearly shown by equations (4.11), (4.12), (4.14) and (4.15) that the trust model outperforms RSVP. Pseudo codes written for RSVP and the trust model showed that a lot more processing needs to be done by an ad hoc node when it uses RSVP. This is true because the trust model integrates QOS along with route discovery.

6.3 Future Work

Further improvisation on the protocol can be done by fine tuning the trust relationship of the nodes. Performance can also be maximized by applying a dynamic threshold value for the trust on which nodes below that threshold value might be given only Best Effort traffic, unless they improve in their negotiation skills and go above the threshold value. Security can be added as a test parameter to assign trust values to nodes in the ad hoc network. The next step in future work will be to implement the proposed trust model, along with the mathematically modeled trust points, in a real life ad hoc environment.

REFERENCES

LIST OF REFERENCES

[1] Sonja Buchegger, JeanYves Le Boudec "Performance Analysis of the CONFIDANT Protocol(Cooperation Of Nodes: Fairness In Dynamic Adhoc NeTworks)",IBM Zurich Research Laboratory, EPFLICLCA Ecublens, sob@zurich.ibm.com,jeanyves.leboudec@epfl.ch

[2] Richard Dawkins. The Selfish Gene. Oxford University Press, 1989 edition, 1976

[3] RFC 4728 for DSR protocol, http://www.ietf.org/rfc/rfc4728.txt [May, 2009]

[4] David B. Johnson, David A. Maltz,"Dynamic Source Routing in Ad Hoc wireless Networks", Computer Science Department, Carnegie Mellon University.

[5] C. E. Perkins, E. M. Royer, and S. Das, "Ad-hoc on-demand distance vector (AODV) routing," IETF RFC3561, July 2003

[6] Dr. Aditya Goel and Anjali Sharma ,"Performance analysis of mobile Ad Hoc network using AODV protocol", Department of Electronics & Communication Engineering, NIT, Bhopal, India.

[7] RFC 3561 for AODV protocol, http://www.ietf.org/rfc/rfc3561.txt [May, 2009]

[8] Nityananda Sarma and Sukumar Nandi, "A Priority based QoS-Aware MAC Protocol (PQAMP) in Mobile Ad hoc Networks", Proceedings of the 4th ACM symposium on QoS and security for wireless and mobile networks (2008).

[9] Qiang Ni Ireland National University, Ireland, "Performance Analysis and Enhancements for IEEE 802.11e Wireless Networks", Network IEEE 2005.

[10] Document for ETS within a single domain http://tools.ietf.org/html/draft-ietf-ieprep-domain-frame-08

[11] Dimitrios J. Vergados, Maria Koutsogiannaki and Dimitrios D. Vergados, "Optimizing End-to-End TDMA Scheduling in Ad-hoc Networks on Random Topologies" ,The Sixth Annual Mediterranean Ad Hoc Networking Workshop, Corfu, Greece, June 12-15, 2007 .

[12] Xi Zhang,Jia Tang, "Cross-Layer-Model Based Adaptive Resource Allocation for Statistical QoS Guarantees in Mobile Wireless Networks", IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, VOL. 7, NO. 6, JUNE 2008.

LIST OF REFERENCES (continued)

[13] Ryam Baradaran , Mohammad Hossein Yaghmaee," A Constraint Based Routing Algorithm For Multimedia Networking"- IAENG International Journal of Computer Science, 33:2, IJCS_33_2_2

[14] Q. Qu, Y. Pei, J. W. Modestino, X. Tian, and B. Wang, BCross-layer QoS control for video Communications over wireless ad hoc Networks,[EURASIP J. Wireless Commun. Network. vol. 5, pp. 743–756, 2005.

[15] Setton, T.oo, X Zhu, A.Goldsmith. Girod, B Cross-layer design of ad hoc networks for real-time video streaming, [IEEE Wireless Commun Mag., vol. 122. No.4, 1. Pp. 59-65, Aug. 2005.

[16] Amine Berqia, Noufissa Mikou, Youssef Dehbi, Blaise Angoma, "Fairness and QoS in Ad-Hoc Networks", Proceedings of the 67th IEE Vehicular Technology Conference, VTC Spring 2008, May 2008.

[17] Muhammad Mahbub Alam, Md. Abdul Hamid, Choong Seon Hong, "QoS-Aware Fair Scheduling in Multihop Wireless Ad Hoc Networks", Proceedings of 8th IEEE ICACT 2006.

[18] Kaixin Xu, Ken Tang, Rajive Bagrodia,Mario Gerla, Michael Bereschinsky"Adaptive Bandwidth Management and QoS Provisioning in Large Scale Ad Hoc Networks" Military Communication conference, Oct 2003 .

[19] Ghalem Boudour, Cédric Teyssié, Zoubir mammeri, "Bandwidth Reservation for Heterogeneous Traffics in Mobile Wireless Networks",Fifth Advanced International Conference 2009.

[20] RFC 2205 for RSVP, http://www.faqs.org/rfcs/rfc2205.html [May, 2009]

[21] Gwo-Chuan Lee, Tsan-Pin Wang, and Chien-Chao Tseng, "Resource Reservation with Pointer Forwarding Schemes for the Mobile RSVP" IEEE COMMUNICATIONS LETTERS, VOL. 5, NO. 7, JULY 2001.

[22] Srinivas Guntupalli, Amar Agrawal, Vivekananda Tadala, Mobile RSVP, 2005.

LIST OF REFERENCES (continued)

[23] Bongkyo Moon and Hamid Aghvami, ,"Reliable RSVP path reservation for multimedia communications under an IP micromobility scenario", King's college London , IEEE Wireless Communications , October 2002.

[24] Usman Javaid, Djmal-Eddine Meddour, Tinku M. Rasheed and Toufik Ahmed, France Telecom R&D Division, France LaBRI, University of Bordeaux I, France, "Cooperative wireless access networks convergence using ad-hoc connectivity: Opportunities an issues", Wireless World Research Forum, 2008

[25] Kumar Viswanath and Katia Obraczka, "Interoperability of Multicast Routing Protocols in Wireless Ad-Hoc Networks", University of California, Santa Cruz, 2004; Wireless Communications & Mobile Computing archive: Volume 6 , Issue 2 (March 2006) table of contents, Special Issue on Ad Hoc Wireless Networks, Pages: 225 – 234 (2006).

[26] Madhusudan Raghuveer, "Interoperability of Ad hoc routing protocols," B.E. Anna University, India 2005.

[27] Dave Cavalcanti and Dharma Agrawal, and Carlos Cordeiro, Philips Research USA Bin Xie and Anup Kumar "Issues in integrating Cellular networks, WLANS, AND MANETS: A futuristic heterogenous wireless network", University of Cincinnati, University of Louisville. Wireless Communications, IEEE, June 2005: Volume: 12, Issue: 3; on page(s): 30- 41.

[28] S. Ganeriwal and M. B. Srivastava, "Reputation-based framework for high integrity sensor networks," in Proc.ACM Security for Ad-Hoc and Sensor Netw., 2004, pp. 66-67.

[29] Manickam, J. Martin Leo, and Shanmugavel. S, "Fuzzy based trusted ad hoc on-demand distance vector routing protocol for MANET", Advanced Computing and Communications 2007, Dec.2007, pp.414-421.

[30] Yu Ping and Wang Ying, "A Revised AODV Protocol with QOS for Mobile Ad hoc Network", Department of Electronic and Communication Engineering, North China Electric Power University; 2009 2nd IEEE International Conference on Computer Science and Information Technology.