Delay Sensitive Routing For Real Time Traffic Over Ad-hoc Networks

2008

Dipika Darshana
University of Central Florida

Delay Sensitive Routing for Real Time Traffic over Ad-hoc Networks

by

DIPIKA DARSHANA
B.S. Dhirubhai Ambani Institute of Information and Communication Technology, 2006

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the School of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Fall Term
2008

Major Professor: Mainak Chatterjee

ABSTRACT

Wireless ad hoc network consists of inexpensive nodes that form a mobile communication network. Due to limitations of the transmission range, the nodes rely on each other to forward packets such that messages can be delivered across the network. The selection of the path along which a packet is forwarded from the source node to the destination node is done by the routing algorithm. Most commonly used routing algorithms, though effective for non-real time applications, cannot handle real-time applications that require strict delay bounds on packet delivery.

In this thesis, we propose a routing protocol that ensures timely delivery of real time data packets. The idea is to route packets in such a way that irrespective of factors like traffic load and node density, the average delay remains within acceptable bounds. This is done by carefully accessing the resources available to a route before a session is admitted along that route. Each link in the route is checked for sufficient bandwidth not only for the new session to be admitted but also for the sessions that are already using that link. The new session is admitted only if the admission does not violate the delay bounds of any on-going sessions. This method of route selection coupled with per-hop link reservations allows us to provide bounds on the delay performance. Extensive simulation experiments have been conducted

that demonstrate the performance of the proposed routing protocol in terms of throughput, session blocking probability, packet drop probability, average path length, and delay.

To my Teachers, Family and Friends!!

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to all those who helped me in successful completion of my thesis. First of all I would like to thank my advisor, Dr. Mainak Chatterjee for his valuable feedback and guidance at all stages without which the completion of this work would not have been possible. I am thankful to Dr. Ratan K. Guha and Dr. Cliff C. Zou for serving on my final examination committee and for their valuable suggestions to make my work better.

I would also thank Air Force Organization of Scientific Research (AFOSR) for supporting this research.

I would extend my thanks to the EECS department at UCF for helping me as and when needed. My sincere thanks to the International Service Center and the UCF Thesis and Dissertation Office for their valuable suggestions towards the final presentation of the thesis.

I would like to acknowledge my labmates, Mukundan Venkataraman and Mohammad Zubair Ahmad for their help. I would also thank Shamik Sengupta, Wenzing Wang, Mustafa Ilhan Akbas and others for maintaining the friendly and lively environment in the Lab, that makes it an ideal place to work.

A special note of thanks to my friends Bhaskar Kumar and Sindhu G. Shankar for their lively company and support throughout my grad school. Last but not the least I would like to thank my parents Mrs. Bibha Verma and Mr. Ajit Kumar for their constant moral support and encouragement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

Wireless ad hoc networks result from random deployment of inexpensive nodes over an area of interest forming a *self-organizing* and *self-configuring* multi-hop network. The success of such networks relies on co-operative behavior of the nodes. Each node acts as a router for forwarding packets for other nodes while generating packets of its own. These networks do not have any backbone infrastructure support and in most cases are short lived. Military applications and emergency rescue services are some of the examples of ad hoc networks.

The nodes that comprise such networks are devices that can communicate over the wireless channel. Two devices can communicate with each other as long as they are within transmission range of each other. These devices could be heterogeneous in nature in terms of computational capability, memory, energy consumption, transmission and reception ranges. Typical examples of such devices are laptops, PDAs, phones and any other device with a radio interface. Since the users carrying these devices are mobile and move about in the area of internet, the topology of the underlying network is dynamic and is constantly changing.

1.1 Applications of Ad hoc Networks

The ease of deployment and the technology advancements leading to inexpensive manufacturing of devices used for ad hoc network has created commercial and government interests for *distributed mission critical applications*. Examples include situations where network infrastructure is either not available, not trusted, or should not be relied on during times of emergency. A few examples include:

- Military equipments for command, control, communications, computing, intelligence, surveillance, reconnaissance and targeting systems.
- Emergency and rescue missions after natural calamity like an earthquake or flood.
- Temporary offices such as campaign headquarters.

Besides these, many civilian applications have gained enormous popularity. Some of them include:

- Environmental applications like forest fire detection, biocomplexity mapping of the environment [24] and flood detection [22].
- Health applications such as diagnostics, drug administration in hospitals, monitoring the movements and internal processes of insects or other small animals, telemonitoring of human physiological data, and tracking and monitoring doctors and patients inside a hospital [23][25][26][27].

- Home automation that is seen in many daily use gadgets including remote operations for televisions, micro wave and VCRs [28].
- University applications like distance lectures.
- Audio/video conferences or voice chat applications.

With the variety of applications being supported by these networks, the traffic carried exhibits diverse characteristics. Some of the applications demand timely dissemination of information while others are intolerable to lost packets. Commercial applications target to provide inexpensive services and the mission critical deployment also aims at low cost and high efficiency results. The limited radio bands (frequency spectrum) and high quality requirements, of particularly real time traffic, make the task difficult. This has led to investigations into various areas that could further enhance performance and efficiency. While quality assurance is necessary for real time traffic, it is also necessary to utilize the limited resources efficiently. Hence an effective balance between service quality and efficiency achieved is needed.

1.2 Elastic and Real Time Traffic

The traffic generated from the applications discussed in Section 1.1 can broadly be categorized as either elastic or non-elastic domain. The elastic traffic refers to those applications that do not have stringent requirements regarding timely delivery of data packets. Accuracy is however critical here. Some of the popular examples for this class of traffic are email, web-browsing and file transfers (FTP).

Unlike elastic traffic, non-elastic or real time traffic has critical delay requirements and demands that the data packets be delivered within certain delay bounds. It essentially requires certain quality of service assurance in terms of bandwidth, delay, and jitter. Reliability in terms of packet delivery is not the prime concern and packet drops to a certain extent are tolerable. Examples for this class of traffic are: real-time audio (IP telephony, streaming music), streaming video, or any traffic demanding small delay and delay variance (jitter), for example interactive distance lectures or video conference.

1.3 Quality of Service and Resource Management

Quality of service is a self evident term. It means assurance of good quality services. It is coupled with resource management as efficient management of resources has a key role to play for assuring QoS. The needs and requirements to meet QoS is application dependent. In adhoc networks every packet requiring some level of QoS assurance follows the same path that has enough resources reserved to support the connection needs.

As discussed above, QoS for elastic traffic is met if data reaches the destination error free; the number of retransmissions and the time taken does not degrade the quality. Hence a best effort approach works fine for elastic traffic. Similarly QoS for real time traffic is met only if the packets are delivered in a timely manner. This demands for resource reservation such that the packet flow is smooth.

Non-real time is more flexible and can adjust to resource fluctuations. In order to support various classes of traffic, it becomes essential to use the network's limited resources effectively. These differences make it necessary to have a specific set of rules to route real time traffic in a way that ensures timely delivery while utilizing the resources efficiently for a high throughput.

1.4 Challenges for Quality of Service over Ad hoc Networks

Achieving quality of service in ad hoc networks is very challenging. Due to the inherent properties of the wireless networks, mobility of nodes and the resource constraint devices, the protocols designed for wired domain cannot be directly applied. The following are some of the differences between the traditional wired networks and mobile ad hoc networks:

1. Infrastructureless architecture: Generally the nodes are randomly deployed. They follow a co-operative approach by forwarding each other's packets along the established route from source to destination. There may or may not be a centralized entity to monitor the nodes. Hence there is a need for a mechanism to enable nodes to find routes and forward successfully along them.

2. Wireless medium: The wireless medium provides the channel for transmission. Nodes need to contend for the medium to transmit. Though this is a medium access control

(MAC) layer problem, it makes successful transmission difficult as nodes may transmit without being aware of each others transmissions' and hence result in collision.

3. Limited resources: The shared wireless channel has limited bandwidth and is contended for by various nodes simultaneously. Interference from neighboring nodes, fading and collisions further restrict the bandwidth usage.

4. Energy constrain: The nodes have limited battery life. The transmission power and collisions requiring multiple retransmissions exhaust the energy. Hence, all protocols should be effective and try to minimize retransmissions and save energy.

5. Node mobility: The nodes are scattered and may move randomly, changing their locations. This accounts for a dynamic topology and results in new neighbors disrupting previously established routes.

1.5 Contributions of this Work

In this research, we propose a network layer routing protocol for ad hoc networks for carrying real time traffic. The proposed routing protocol is based on the availability of resources (bandwidth) at every hop of a route. Alongside, resource reservation and residual bandwidth distribution schemes are also proposed. In particular, the contributions are:

- We devise a route discovery scheme that is based on delay estimation along any route. The delay of a route is calculated by summing the delays at each hop. Of the multiple possible paths between the source and destination, the path that offers the minimum end-to-end delay is selected.

- We use a M/M/1 model to calculate the delay. We consider the requirement of a new session in terms of the packet generation rate and evaluate if the residual bandwidth of the individual links can handle the traffic generated.

- We also propose a resource reservation mechanism that ensures that the quality of the on-going sessions never degrades. This is achieved with the help of hard and soft reservation. The minimum amount of bandwidth needed to support the traffic is reserved while the residual bandwidth is distributed over the sessions to process packets faster.

- We conduct C based simulation experiments to validate the proposed schemes. We vary parameters like session generation rate, number of sessions requested, and network size. Performance of the proposed schemes is shown with respect to metrics like throughput, session blocking probability, packet dropping probability, average path length for a route, and average delay.

1.6 Organization of the Thesis

The rest of the thesis is organized as follows. The background and related work are described in Chapter 2. The proposed routing protocol is presented in Chapter 3. Illustrative examples are provided for easy comprehension. Chapter 4 presents the simulation model and the results are compared with the naive approach. Conclusions are drawn in the last chapter.

CHAPTER 2
BACKGROUND AND RELATED WORK

2.1 Routing Protocols

A route is defined as the path taken by a packet to reach the destination from the source. There are different ways to choose a path and the rules for deciding upon any path are based on the network capabilities and traffic requirements. As discussed in [8], routing protocols can be broadly classified into the following two types:

- Table-driven: These protocols require each node to maintain routing tables. The tables have routing information entry corresponding to every other node in the network. Any change in the network topology results in updates being propagated throughout to maintain consistent information. Examples include, DSDV(Distance Sequence Destination Vector) [4][5], and WRP (Wireless Routing Protocol) [29].

- Source-initiated (demand driven): As the name suggests, these are requested by the source nodes as and when required. This does not require maintaining up-to-date routing tables. The routes are discovered when required and maintained till any path failure occurs or is no longer required. Examples are AODV (Ad Hoc On-Demand Distance Vector Routing) [1], DSR (Dynamic Source Routing) [2], and TORA (Temporally Ordered Routing Algorithm) [3].

Most of the earlier routing protocols proposed for mobile ad hoc networks like AODV, DSR, TORA, DSDV and the lightweight mobile routing protocol [6] emphasize on finding the shortest-path route with a high degree of availability since the topology is assumed to change very frequently. Though these have performed well for the elastic traffic, these do not take into consideration the real time traffic requirements for path discovery and hence do not perform well at providing the desired level of quality of service.

2.2 Real Time Traffic in Wired Domain

The traditional Internet follows the best effort approach. It tries to deliver the packets without assuring any hard guarantees with respect to delay or packet loss. This approach meets the requirements for most of the elastic traffic but the increasing popularity for real time applications with its associated QoS demands led to the development of two well known schemes for providing quality of service over the wired networks. They are Intserv [7] and Diffserv [9][10].

2.2.1 IntServ

IntServ known as Integrated Services [11][12][7] provides quantitative quality of service on *per-flow* basis. Its main features are highlighted below:

- It requires every application to make individual reservation.
- It uses flow specs to describe what the reservation is made for. The flow spec includes traffic specifications which describes the nature of traffic. The other specification describes the service guarantees.
- RSVP is the underlying mechanism that propagates the attributes of the data flow.
- Each of the routers need to store the specifications of the traffic flow and also police the traffic.

IntServ is an approach that requires the routers to store too many state information. Hence there are scalability limitations. It works well for small traffic load. However for heavy traffic load where too many flows are concurrent, it is a big overhead to maintain specifications for each of the flows.

2.2.2 DiffServ

DiffServ known as Differentiated Services [13][9][10] is the current approach towards attaining quality of service over the traditional Internet. It's main features are:

- It works on the principle of classification of traffic into various classes.

- Each packet is classified into a particular class at the boundary of the network and is then forwarded on a per hop basis based on the set of rules associated with that class of traffic to which the packet belongs. Hence, each router is equipped to differentiate traffic based on its class.

DiffServ is simpler as compared to IntServ because the routers simply participate in forwarding the packets and are relieved from maintaining various state information and policing. The challenge in this scheme is to design rules that would help routers to differentiate between different classes of traffic and also require the end hosts to abide by the data flow agreements.

2.3 Real time Traffic in Wireless Domain

IntServ and DiffServ cannot be directly used for wireless networks. Though DiffServ performs well for wired networks, ad hoc networks cannot implement it because of the lack of a backbone support. Extensive research that takes into account the limited capabilities of the nodes and the scarce spectrum has led to development of proposals some of which derive from these two schemes. Let us briefly discuss some of the popular schemes.

2.3.1 Flexible Quality of Service Model for Mobile Ad Hoc Networks

Flexible quality of service model for mobile ad hoc networks (FQMM) has been proposed for supporting quality of service for mobile ad hoc networks [14]. It extends both IntServ and Diffserv approaches to the mobile ad hoc networks.

It defines three types of nodes' behavior. A node may have one or more of the following roles to play at the same time.

1. An **ingress** node is the source node that needs to send data.
2. An **interior** node is a node that is involved in packet forwarding.
3. An **egress** node is the destination node.

FQMM proposes a hybrid per-flow and per-class provisioning scheme. It provides per-flow provisioning to highest priority traffic while the remaining flows receive per-class provisioning. It works well for small to medium size network with less than 50 nodes. The proposal claims to have a better performance in terms of throughput and service differentiation than the best effort model. However it still has the following problems:

1. It fails to provide per-flow provisioning for all.
2. Due to broad classification as per the Diffserv approach, the service levels are not accurate.

3. FQMM assumes major portion of traffic belongs to the less priority class which might not be true.

4. FQMM needs policing at the source node (ingress) for traffic shaping once a route is discovered.

2.3.2 INSIGNIA

INSIGNIA is a circuit based model that requires explicit connection establishment prior to data transmission [15][16]. It is an in-band signaling protocol which means that the control information is included in the IP packet header. Service mode, payload type, bandwidth indicator and bandwidth request fields are the typical variables that are included in the header. The source nodes specify appropriate fields in the IP header in order to request for reservations in control packets called *reservation packets*. This specification includes setting of the RES bit for the service mode, and also specifying the maximum and minimum bandwidth requirements by the source. As the reservation packets traverse the intermediate nodes, each node performs admission control modules, allocates resources and establishes flow-state. This means that if the request is accepted, the resources are committed and subsequent packets are scheduled accordingly. The reservation policy is soft-state which means that it is committed for a given amount of time. After the timer expires the reservations are canceled unless any updates are sent. However if the node is not capable of making the resource reservations then the packets receive best effort services. Unless the source is acknowledged of a path it keeps generating the reservation request. Once the destination receives reservation confirmation it verifies the reservation status. If all the intermediate nodes have been able to provide the required reservations the application is assured of the desired quality of service else there is partial reservation and the service offered is that of best effort. Flow state needs to be stored in every node in the route in soft state manner.

However, there are some problems with INSIGNIA. They are:

1. A RES packet may be degraded to best effort service in the case where re-routing or insufficient resources exist along the new/existing route. Hence quality of service level degrades.

2. It requires the intermediate nodes to store the flow information and hence suffers form the scalability problem as seen in Intserv.

3. The source needs to specify the minimum and maximum bandwidth requirements. This is another drawback and a fine-grained approach can help in achieving better quality of service needs and resource utilization.

4. The bandwidth is wasted due to any partial reservations made. The reservations made for traffic that receives best effort treatment accounts for this problem.

2.3.3 Packet Marking Based QoS for Real Time Traffic

The packet marking based QoS for real time traffic [17] tries to balance between the needs of real time and best effort traffic. It tries to minimize the queuing delay of time sensitive flows while trying to provide fair allocation to both kinds of traffic so that none monopolizes the network. This scheme is based on the proposal made by Gibbens and Kelly as in [18] for admission control of real time traffic and adjusts the transmission of elastic traffic with a congestion control mechanism. It assumes that real time flows generate packets at a constant rate. The call admission control mechanism is based on a trial period. Any new request made for real time flows has to go through a trial period. During this period its effect on ongoing sessions is estimated and accordingly it is allowed or rejected. The trial period helps in determining if the network has sufficient resources to admit a new session or not. This estimate is done through packet marking based on configuration of virtual queues. Every node maintains separate virtual queues for real time and elastic traffic, besides maintaining two physical queues for the same. Whenever a packet arrives, it is placed in the appropriate physical queue and also an imaginary counterpart is enqueued in the virtual queue. If this new arrival overflows the virtual queue then the packet in the physical queue is marked. If more than a certain number of packets are marked the call is blocked as per the scheme.

Though this scheme ensures quality of service to the admitted calls it has the following problems:

1. The trial period is an overhead as it needs some reservation to establish the flows.

2. The ongoing sessions may suffer temporary degradation due to the trial period.

3. The onset for marking packets is another issue. If the virtual queues start marking packets when the link utilization exceeds 75%, a session with only 5% bandwidth requirement may be denied services. Thus the scheme limits the bandwidth utilization that the real-time traffic could have utilized and unnecessary blocking takes place.

2.3.4 Adaptive QoS for MANETS

The adaptive QoS for MANETS [19] is a routing protocol that tries to provide different class of service to different applications. To achieve this, it classifies applications into the following classes in decreasing order of priority:

- Urgent Messages (UMs): These are the applications that are connectionless and demand minimum end-to-end delays.

- Urgent Flows (UFs): These are urgent calls and hence require fast call set up and also fast rerouting in case of failures. However, these occur infrequently and have low to medium bandwidth requirements of upto 100 Kbps.

- Regular Flows (RFs): These are normal voice and video traffic with high bandwidth requirements.

- Best Effort (BE): These are connectionless applications that have no bandwidth requirement.

The scheme depends on clustering, channel allocation, and route computation. The network is divided into clusters and nodes have knowledge of network topology beyond the local neighbors. Within each cluster each node is supposed to participate in link state protocol to exchange information about the topology. Besides these, the gateway nodes need to exchange information at cluster level. This exchange of information helps in finding routes for the different classes of traffic described earlier.

The routing for these different class of traffic confers to the following set of rules:

- Urgent Messages (UMs): The path selection criterion is based on minimum end-to-end delay. The computation for delay is simply $\frac{1}{C}$, where C is the link speed. This however does not include the queuing delay.

- Urgent Flows (UFs): The shortest path selection criterion is based on $\frac{1}{C} + \lambda I$, where C is link speed, λ is a tuning parameter, and I is the bandwidth used by the interfering links.

- Regular Flows (RFs): These are expected to take the routes that are least loaded.

- Best Effort (BE): The minimum hop path is the route selection criterion here.

Though this scheme tries to provide differentiated services it still does not solve the classification problem of the original differentiated services. Its problems are listed as follows:

1. The classification problem is similar to that of Diffserv approach. The real time traffic whether urgent or regular has same QoS needs.

2. The nodes need to store huge topology information which is an overhead.

3. This scheme requires the formation of clusters and involves hierarchical routing which is not necessary for small size deployments.

CHAPTER 3
PROPOSED ROUTING PROTOCOL

In this research, we design a routing protocol that is particularly targeted for real-time applications. Since the traffic is real time we want to ensure timely delivery of packets and hence average delay per packet is considered as the most important path selection criterion. The naive method would be to pick any random path that offers the required bandwidth needed to support the traffic. However problem arises when new flows are admitted as they may interfere with the ongoing sessions adversely. At the same time it is not a good idea to block flows that could have been supported if paths were chosen wisely. The proposed route selection algorithm is coupled with the resource allocation mechanism that accommodates large number of concurrent real time flows with quality assurance.

3.1 Target Application Domain

QoS aware routing does not make sense if the routes get disrupted frequently and the promised QoS is violated. Ideally, one can assume that the environment is less dynamic and topology remains in the same state during a session. An example could be a scenario where devices equipped with audio-video capturing capabilities be deployed at targeted locations. These nodes would monitor the area constantly and any event that triggers the sensors would lead to a request of path to send the information to a centrally placed node via the typical multihop communication.

3.2 Consideration for Algorithm Design

We consider a two dimensional geographical area with a certain number of nodes confined within the region. Ensuring quality of service would be difficult if the nodes have high mobility. Hence we assume nodes do not move to the extent that their neighbors change while transmitting packets. The proposed scheme is *distributed* in nature and does not involve any centralized path computation which is usually very expensive. Moreover, the proposed scheme is a *source-initiated* on-demand route acquisition system that does not require routing table exchanges or maintenance.

Our notion of quality of service is guaranteeing delay bounds. Average end-to-end delay is the metric for deciding the best path if multiple paths exist between a given source-destination pair. For every event that triggers a node, it initiates a new path discovery phase. Though multiple paths may exist between source and destination nodes, any intermediate node participating in route discovery is aware of just the *minimum delay* path through it.

3.3 M/M/1 Systems

We assume that the packets arrive at a node following a Poisson distribution and their processing times are exponentially distributed. Thus, the queue at any node can be modeled as a M/M/1 system [30]. The assumption of a M/M/1 model is justified as short messages have shown to follow a Markovian arrival process. Also, the service times of each packet is independent of each other. Moreover, it is easy to analyze M/M/1 systems because of their mathematical tractability

M/M/1 theory is applicable to any queuing station with a single server. The *Kendall's* notation describes any M/M/1 queue to have the following characteristics. The first "M" defines the memoryless attribute of the arriving traffic while the second one defines the exponential nature of the service time. The last numerical figure is a count of the number of servers. The arrival and service process as defined by memoryless and exponential terms respectively are discussed next.

Arrival process: The memoryless feature of incoming traffic relates to a Poisson process. Rate λ is the packet generation rate for any traffic request. It is the long time average and is a fixed value for a transmission.

Service Process: This defines the nature of the probability distribution of the service times. If the service rate is μ, then the probability distribution of the service time is exponential with mean $\frac{1}{\mu}$. In this work μ is the channel bandwidth allocated for a transmission.

3.4 Delay Calculation

We use the M/M/1 model to calculate the delay. Each request for transmission has its individual constant average traffic generation rate and packet size. If packet arrival rate is λ and service rate is μ, then the average delay is $\frac{1}{\mu-\lambda}$. This delay is inclusive of service time and queuing delay. The queuing at any node in the network is represented in Fig. 3.1.

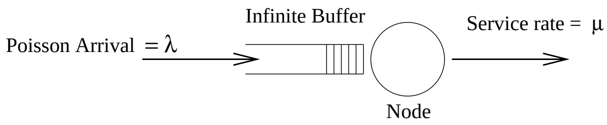

Figure 3.1: M/M/1 Model

The packets arrive randomly in the queue as per the Poisson distribution. The buffered packets are then served as per the FIFO (first in first out) policy at a rate equivalent to the channel capacity of the outgoing link.

The delay incurred by any packet at a node i is estimated as

$$\mathcal{D}_i = \frac{1}{\mu-\lambda} \text{ seconds}$$

The delay between source and destination can be estimated by adding up delays incurred at each hop along the path. From a source **S** to destination **D**, total delay along path is the summation of individual delays at each hop, i.e.,

$$\mathcal{D}_{total} = \sum_{i \in P} \mathcal{D}_i$$

where P is the set of nodes comprising the path from source to destination.

3.5 Resource Reservation Techniques

The delay calculation above takes into account the service rate which is essentially an estimation of available bandwidth or resource. The available bandwidth needs to be more than the data rate being injected into the system by the source. This assurance can be achieved by reserving resources/bandwidth. There are typically two types of reservation mechanisms.

1. Soft Reservation: This is the case when a reservation is made for a given period of time. As soon as resources are allocated to a flow, an associated timer is started at the end of which the resources are released. However if updates are sent before expiration then the timer is set accordingly and the reservation period may be extended.

2. Hard Reservation: Though the reservations are made in a similar way, hard reservation needs explicit release message in order to release the reserved resources. There is no lifetime or timer associated with hard reservation.

The proposed scheme uses a combination of both soft and hard reservation. The minimum bandwidth required to support the session is allocated permanently until the session is over while the residual bandwidth allocation varies as sessions begin or end between a given pair of nodes.

3.6 Algorithm Description

Let us describe the algorithm for selecting minimum delay path on a per hop basis. The delay is estimated at each node along the path with sufficient resources and the path with minimum end-to-end delay is selected. The formal algorithm is presented in Fig. 3.2 with the detailed steps discussed below.

1. A source node creates a request by specifying the traffic descriptors like the packet arrival rate, λ packets/second and the packet size, c bits/packet.

2. The source node then checks its outgoing links for available bandwidth which is the service rate μ. All neighbors receive the request, but only the ones with sufficient resources along the link will further send the request to their neighbors.

3. While doing so, the source node calculates the delay along each outgoing link and then forwards the request to its neighbors. In this way the delay incurred up to that neighbor in the path is known. In case the link capacity is not enough to support the session the delay is set to infinity and such requests are not forwarded any further.

4. A node that receives the request for the first time performs the same set of operations as the source node. It also calculates the delay along each outgoing link with sufficient resources, and adds this delay to the delay incurred in reaching this node from the source and forwards the request further.

```
For every node $N_i$ in the network, setup neighbor list $L(N_i)$;
For every neighbor $N_j$ of node $N_i$ set channel capacity $C_{ij}$;
A request for transmission with packet arrival rate= $\lambda_k$ is initiated by node $N_k$;
for all outgoing links $OL_k$ through $N_k$ do
    if $bandwidth_{available}(OL_k) > \lambda_k$ then
        Calculate $delay = \frac{1}{capacity_{available}(OL_k) - \lambda_k}$;
        Add its own id in the request packet;
        Cache the request packet;
    else if $bandwidth_{available}(OL_k) < \lambda_k$ then
        $delay = \infty$
    end if
    Forward the request packet to the neighbor;
end for
Check the cache of every node $N_r$ that receives the request packet;
if This request is received for the first time then
    for all outgoing links $OL_r$ through $N_r$ do
        if $bandwidth_{available}(OL_r) > \lambda_k$ then
            Calculate $delay = delay + \frac{1}{bandwidth_{available}(OL_r) - \lambda_k}$;
            Add its own id in the request packet;
            Cache the request packet;
        else if $bandwidth_{available}(OL_k) < \lambda_k$ then
            $delay = \infty$
        end if
        Forward the request packet to the neighbor;
    end for
else if This request is present in the cache then
    if the delay in the new request is less than the delay in cache then
        for all outgoing links $OL_r$ through $N_r$ do
            if $bandwidth_{available}(OL_r) > \lambda_k$ then
                Calculate $delay = delay + \frac{1}{bandwidth_{available}(OL_r) - \lambda_k}$;
                Add its own id in the request packet;
                Replace the cache request packet;
            else if $bandwidth_{available}(OL_k) < \lambda_k$ then
                $delay = \infty$
            end if
            Forward the request packet to the neighbor;
        end for
    end if
end if
```

Figure 3.2: Minimum delay route algorithm

5. If the node has already forwarded this request earlier it checks to see if the new path along which it receives the request has a lower delay to offer. If so, then it repeats the above step and updates its cache with the new request packet.

6. If the receiving node is the destination node then it does not forward the request further.

The above algorithm for route selection is explained in Fig. 3.2

In the proposed algorithm, the service rate μ needs to be estimated in order to calculate the delay. This μ is the exponential service rate that would be offered if the session is admitted. The next section describes the steps involved for service rate estimation.

3.7 Resource Allocation

Though packet arrival process is bursty in nature, we can always find the long term average arrival rate. As long as the bandwidth is enough to support this arrival rate, the quality of service is acceptable. Hence any new flow being requested to be admitted into the system should be ensured a minimum guarantee on the bandwidth. Hence bandwidth equivalent to the packet arrival rate as specified in the request can be allocated for the session till it ends. This hard reservation of resources are dedicated to a flow until the flow ends.

Larger bandwidth results in faster processing of the packets and hence delay is less. Thus a higher service rate is always preferred. At any instance, there could be some channel

Calculate channel capacity C_{AB} between nodes N_A and N_B;
Set residual bandwidth $RB_{AB} = C_{AB}$;
if A new session S_i is allowed through nodes $N_A \rightarrow N_B$ with packet arrival rate λ_i **then**
    $RB_{AB} = RB_{AB} - \lambda_i$;
    $\mu_i = \lambda_i + \frac{\lambda_i}{\sum_{k=1}^{n} \lambda_k} * RB_{AB}$; where n is total number of sessions through this pair of nodes
    for all ongoing sessions through this pair of nodes **do**
        $\mu_l = \lambda_l + \frac{\lambda_l}{\sum_{k=1}^{n} \lambda_k} * RB_{AB}$;
    end for
end if
if An ongoing session S_j terminated between nodes $N_A \rightarrow N_B$ with packet arrival rate λ_j **then**
    $RB_{AB} = RB_{AB} + \lambda_j$;
    for all ongoing sessions through this pair of nodes **do**
        $\mu_l = \lambda_l + \frac{\lambda_l}{\sum_{k=1}^{n} \lambda_k} * RB_{AB}$;
    end for
end if

Figure 3.3: Resource allocation scheme

bandwidth that has not been assigned to any ongoing session; this is known as the residual bandwidth. We distribute this residual bandwidth proportionally amongst all ongoing sessions. This assignment is like soft reservation. However, allocation of a part of the residual bandwidth is not for any pre-determined time period. Hence, the extra allocated bandwidth can be withdrawn in order to accommodate any new flow that would share the residual bandwidth.

In a nutshell, all flows are admitted if and only if the delay requirements can be met. This is achieved through hard reservation. If residual bandwidth is available, then it is shared in a proportional manner. The algorithm for resource allocation is explained in Fig. 3.3

3.8 Illustrative Examples

Let us demonstrate the route selection algorithm and resource allocation scheme through some illustrative examples.

3.8.1 Example for Path selection

Suppose the source S in Fig. 3.4 generates a request with $\lambda = 20$ packets/second. The available bandwidth along the link from source S is mentioned in parenthesis. For example, $N8(30)$ means node with id 8 has a bandwidth of 30 packets/second along the link from S to $N8$. The outgoing links for nodes $N1$, $N2$, $N3$, $N4$, $N7$ and $N8$ as marked with

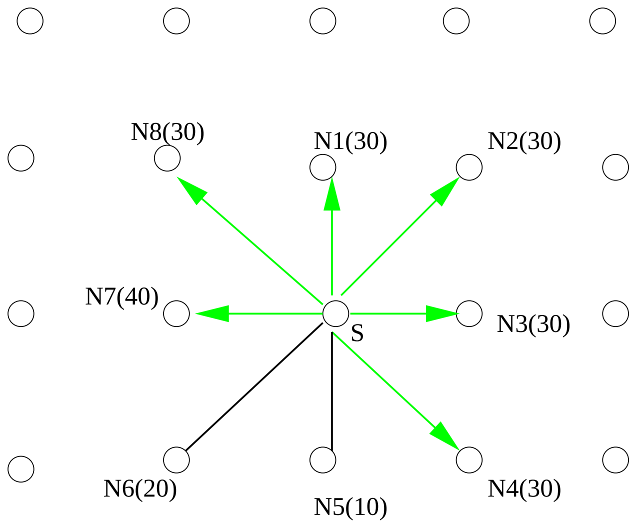

Figure 3.4: Session request being made by node S *for* $\lambda = 20$ *packets/second*

light arrows accept the request as they offer bandwidths of 30 and 40 packets/sec which is sufficient for supporting the transmission. Nodes $N5$ and $N6$ have residual bandwidth of 10 and 20 respectively; hence they cannot support the new request from node S.

While sending its request to these nodes, the source node calculates the delay as discussed earlier. The delay along paths to nodes $N1$, $N2$, $N3$, $N4$ and $N8$ is estimated as

$$\frac{1}{\mu-\lambda} = \frac{1}{30-20} = 0.1 \text{ seconds/packet}$$

Similarly the delay along $N7$ is

$$\frac{1}{\mu-\lambda} = \frac{1}{40-20} = 0.05 \text{ seconds/packet}$$

Any node receiving the request takes one of the following two actions

- Action1: If the request is new, it stores it in its cache and forwards the same request to all qualifying neighbors but for the one from which it had received the request. It also calculates the delay along the path it forwards and adds it to the delay it received to compute the total delay along the path. Nodes that receive the request further repeat the same process.

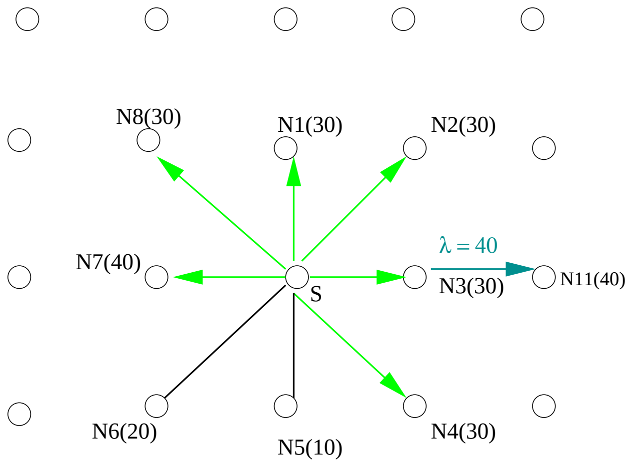

Figure 3.5: Action taken when a new request is made

As in Fig. 3.5 $N3$ receives request for the first time from source S. It stores the request in its cache and forwards it to all its qualifying neighbors. The delay it received from S as calculated earlier was 0.1 seconds/packet. $N11$ has capacity of 40 packets/second. The delay along $N3$ to $N11$ is

$$\frac{1}{\mu-\lambda} = \frac{1}{40-20} = 0.05 \text{ seconds/packet.}$$

Hence $N3$ sends the request to $N11$ with the total delay of

$$0.1 + 0.05 = 0.15 \text{ seconds/packet}$$

- Action2: If a request has already been received by a node, then the receiving node compares the delay of the new request with the delay of existing request.

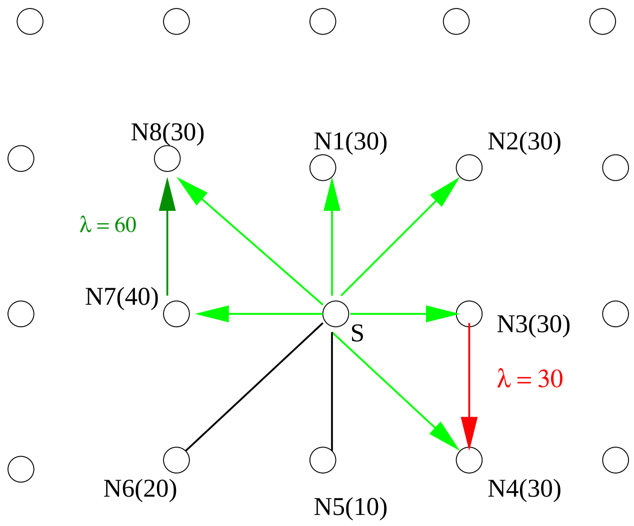

Figure 3.6: Action taken when repetitive requests are made for a session

As shown in Fig. 3.6, $N3$ calculates the delay along $N4$ to be

$$0.1 + \frac{1}{30-20} = 0.1 + 0.1 = 0.2 \text{ seconds/packet}$$

However, $N4$ already received a request with delay = 0.1 second/packet. Hence it ignores the packet. $N7$ calculates delay along $N8$ as

$$0.05 + \frac{1}{60-20} = 0.05 + 0.025 = 0.075 \text{ seconds/packet}$$

$N8$ also has an earlier request with delay of 0.1seonds/packet from S. Since the new delay along the longer path is less, it updates its cache and further forwards to its neighbors.

In this way, any node is aware of just one path which happens to be the minimum delay path through it.

3.8.2 Example for Resource Estimation

Let us now consider an example that illustrates how resource allocation is done for allowing new flows into the system. Assume there are two nodes $N1$ and $N2$ with total channel capacity of 100 packets/second. For simplicity assume that packets have same size of n bits each.

At time t_1 there is a request of $\lambda_1 = 20$ packets/second through this path. Say the path is

$$P_1 = S1 \rightarrow N1 \rightarrow N2 \rightarrow N10 \rightarrow Destination$$

assignment as per our scheme is

$$\mu_1 = \lambda_1 + x_1 \times \text{residual bandwidth}$$

$$\mu_1 = 20 + x_1 \times (100 - 20) = 100 \text{ packets/second}$$

$x \leq 1.0$ is a multiplicative factor that decides the ratio in which the residual bandwidth is distributed. Its calculation is discussed next.

Now at time t_2 there is another request of $\lambda_2 = 30$ packets/second. Say the path for this transmission is

$$P_2 = S2 \rightarrow N5 \rightarrow N1 \rightarrow N2 \rightarrow N11 \rightarrow N15 \rightarrow Destination$$

This uses the link $N1 \rightarrow N2$ which is also serving the session requested earlier by S1. Thus there is a change in the residual bandwidth allocation between $N1$ and $N2$ for the previous session S1 as well. This change is as follows

$$\mu_1' = \lambda_1 + x_1' \times \text{residual bandwidth}$$

where

$$x_1' = \frac{\lambda_1}{\lambda_1 + \lambda_2}$$

and

$$\mu_1' = 20 + \frac{20}{20 + 30} * (100 - 20 - 30)$$

$$\mu_1' = 40 \text{ packets/second}$$

The assignment for second session is

$$\mu_2 = \lambda_2 + x_2 \times \text{residual bandwidth}$$

where

$$x_2 = \frac{\lambda_2}{\lambda_1 + \lambda_2}$$

and

$$\mu_2 = 30 + \frac{30}{20 + 30} * (100 - 20 - 30)$$

$$\mu_2 = 60 \text{ packets/second}$$

Hence the resource allocation scheme can be summarized as follows:

$$\mu_i = \lambda_i + \frac{\lambda_i}{\sum_{k=1}^{n} \lambda_k} \times \text{residual bandwidth}$$

where n is the total number of ongoing sessions between the given pair of nodes.

CHAPTER 4
SIMULATION MODEL AND RESULTS

To validate the proposed protocol, we conducted extensive simulation experiments. The simulation was implemented using C++.

4.1 Simulation Model

We considered a set of nodes that were randomly distributed over a square region. For simplicity, the nodes were assumed to be identical with respect to their transmission and receiving power. The nodes remain in their positions so that the path is not disturbed during a session. The channel capacity between any pair of nodes is assumed to be constant. We consider that the nodes generate flows that are real time data packets only. The packet arrival process is assumed to be Poisson and the service time is exponentially distributed. Packets are lost due to channel errors. Since we do not deal with the medium access control (MAC) layer, we assume a MAC that accounts for collision free environment.

4.2 Simulation Parameters

The simulation is carried for three different network sizes with 60, 80 and 100 nodes. The nodes are randomly placed over a 200 $\times$ 200 area and the packets they generate are destined for a centrally located sink. The transmission range is 40 meters for each node. The rate at which requests for packet transmission are generated is varied from as low as 0.002 requests/second to 1 request/second. The request is made for either audio or video traffic. Data for an audio session has packet generation rate of 20 packets/second with each data packet of size 100 bytes. Video traffic has packet generation rate between 25 and 500 packets/seecond with packet size of 500 bytes. All packets are generated as per the Poisson model and packet service times are exponentially distributed. Hence, all nodes are modeled as M/M/1 systems.

Table 4.1 summarizes the simulation parameters used in the experimental setup.

Table 4.1: Simulation parameters

Number of nodes	60,80,100
Total area	200x200
Transmission range	40 (meters/sec)
Audio Packet generation rate	20 packets/sec
Audio Data packet size	100 bytes
Video Packet generation rate	[25-500] packets/sec
Video Data packet size	500 bytes

4.3 Metrics of Interest

The simulation is carried to study the behavior of the proposed routing scheme with respect to various metrics. Throughput, session blocking probability, average number of hops, fraction of paths through minimum hop, and average delay incurred are some of the metrics that are used to evaluate the routing protocol. Let us formally define these metrics.

- Throughput: Throughput is defined as the average number of packets received successfully at the destination per unit time. This is obtained by counting the total number of packets successfully reaching the receiver and dividing by the simulation time.

- Session Blocking Probability: It is the ratio of denied session requests to total number of requests made. The sessions are denied due to insufficient resources in the network.

- Packet Dropping Probability: Packets in a wireless network can be dropped either due to lossy channel or due to buffer overflow. Here we measure the packet dropping probability as the ratio of packets dropped due to buffer overflow to total number of packets generated.

- Average Path Count: It is the average number of hops taken from source to destination. All requests are routed through the minimum end-to-end delay path. This path is decided based on bandwidth availability and may not be the minimum hop path. This way, we can find the average number of hops taken to reach the destination.

- Fraction of Paths Through Minimum Hop: As mentioned earlier for the average path count, the packets may be routed through a path that has more hops. This metric helps in estimating what percentage of total paths take the minimum hop path.

- Average Delay: This is the average delay incurred by a packet when it travels from a source to the destination. This is obtained by dividing the total delay by total number of successful packets.

4.4 Simulation Results

4.4.1 Throughput

Throughput of a network is a measure of its packet delivery efficiency. The plot presented in Fig. 4.1 shows the throughput for varying session generation rates. When requests for session are made at a higher rate, i.e. the time between two successive requests for route discovery is less, the throughput is higher.

Each simulation point corresponds to throughput obtained for a given session generation rate. The rate is varied from 0.002 sessions/second to 0.05 sessions per second. 500 requests were made one after the other for a given session generation rate and the corresponding throughput is obtained for that particular session generation rate. The throughput increases from approximately 0.2 Mbps to about 2.5 Mbps. The plot displays almost a linear growth pattern with slight fluctuations. Hence we conclude that for higher request generation rate

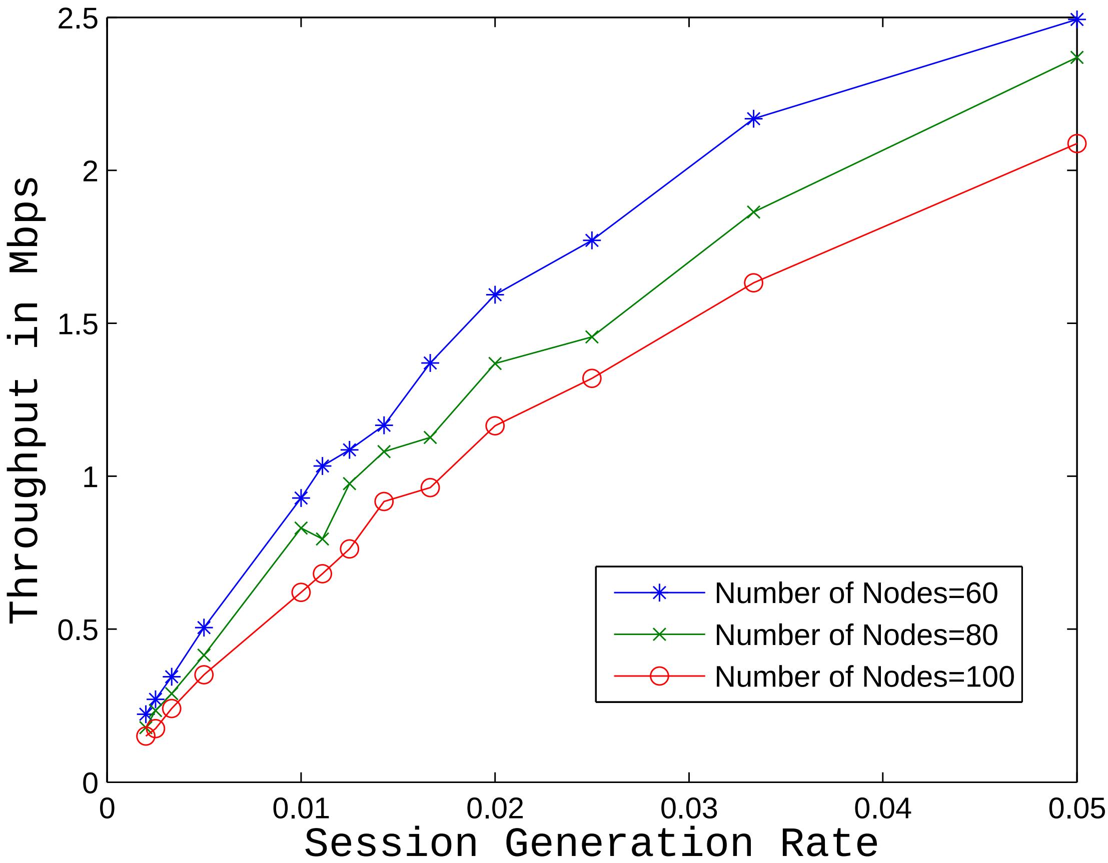

Figure 4.1: Throughput with increasing session generation rate

the packet delivery rate scales well. Hence the system is expected to have good performance with respect to request rate.

Fig. 4.2 shows the effect of increasing number of requests at a given session generation rate. The rate of requests is fixed at 0.02 sessions/second and number of requests made is varied form 10 request to 500 requests. The throughput increases initially till 100 sessions have been requested. At around 100 requests the system reaches its saturation.

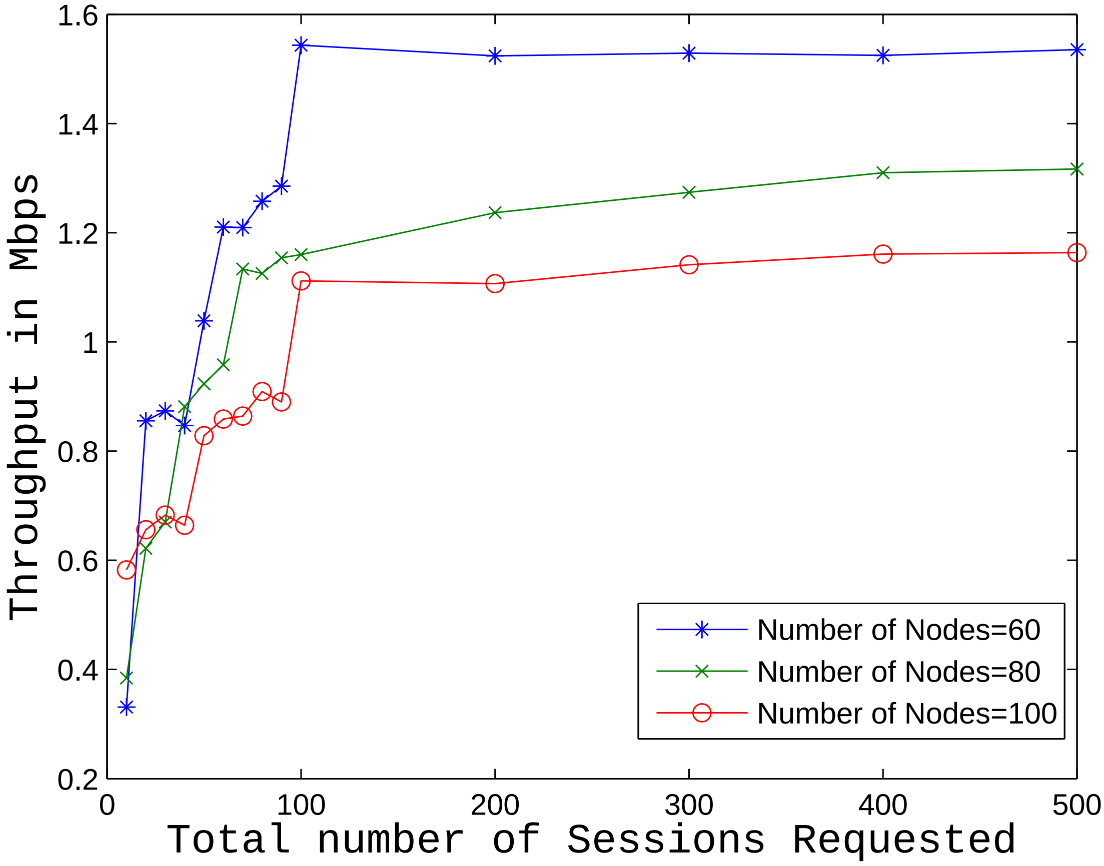

Figure 4.2: Throughput with increasing number of requests for a fixed session generation rate of 0.02 sessions/second

In the same plot, we observe the throughput for different network sizes. The throughput is slightly higher for smaller network size. A smaller network has higher saturation level. It is due to the smaller routes taken in a smaller network. The smaller routes result in less number of hops taken along the path which results in less number of packets loss due to channel related issues.

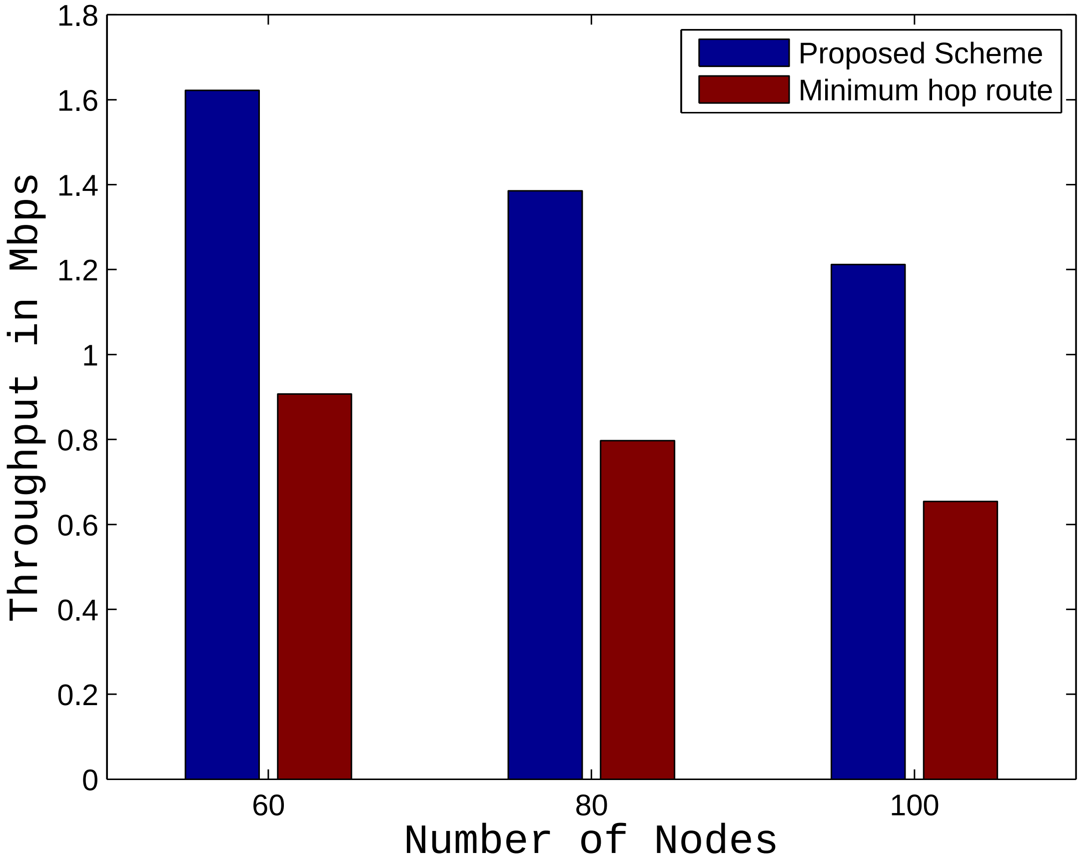

Figure 4.3: Throughput compared for our scheme and minimum hop route algorithm

Fig. 4.3 shows a comparison between the minimum hop based route selection and the proposed scheme. The throughput for the proposed scheme for any network size is higher than the minimum hop path selection algorithm. Thus we see that a systematic route selection results in higher throughput for the network.

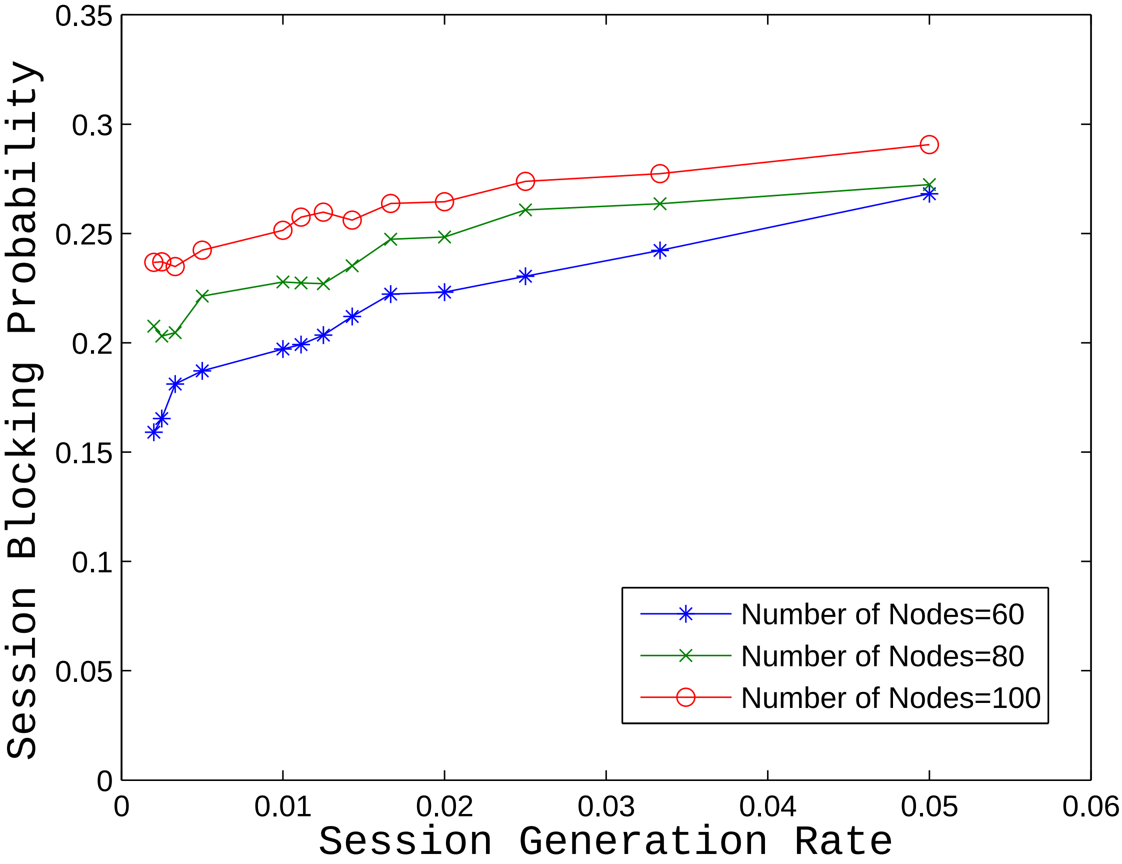

Figure 4.4: Session blocking probability with increasing session generation rate

4.4.2 Session Blocking Probability

We study the session blocking probability of the proposed scheme. As more and more requests are generated in a given time, the resource limitations would deny admission to some of the requests. The blocking probability increases for higher request rate as seen in Fig. 4.4. However even for high rates equivalent to one request every 20 seconds i.e., at a rate of 0.05 sessions/second the blocking probability is 26 to 29% for the different network sizes. For most of the applications the request is not so frequent and the blocking probability varies between 0.15 and 0.25 for different network sizes.

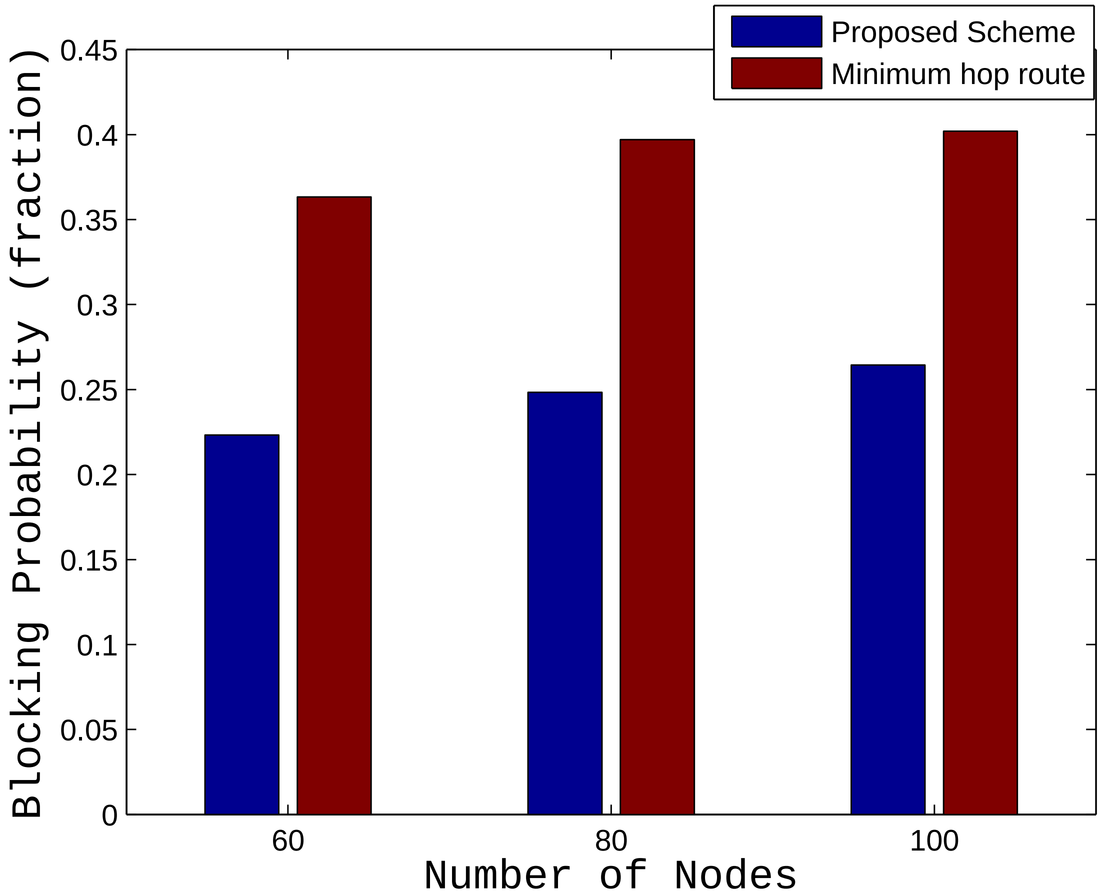

Figure 4.5: Blocking probability compared for our scheme and minimum hop route algorithm

Fig. 4.5 shows the effect on blocking probability for the proposed scheme and the minimum hop path selection strategy. The blocking probability is lower for the proposed scheme for the different network sizes. Hence more and more requests are denied and the network utilization suffers if the routes are not chosen strategically.

4.4.3 Packet Dropping Probability

The M/M/1 model has infinite buffer size. For practical purposes it may not be possible to implement infinite buffer size. We study the packets loss incurred due to buffer overflow in this experiment. We vary the buffer size from 0.15625 megabytes to 5 megabytes. Fig. 4.6 shows that the packet drop probability for a buffer size of 0.625 megabytes is 4% and drops to nearly 0% for a buffer size of 1 megabyte or higher.

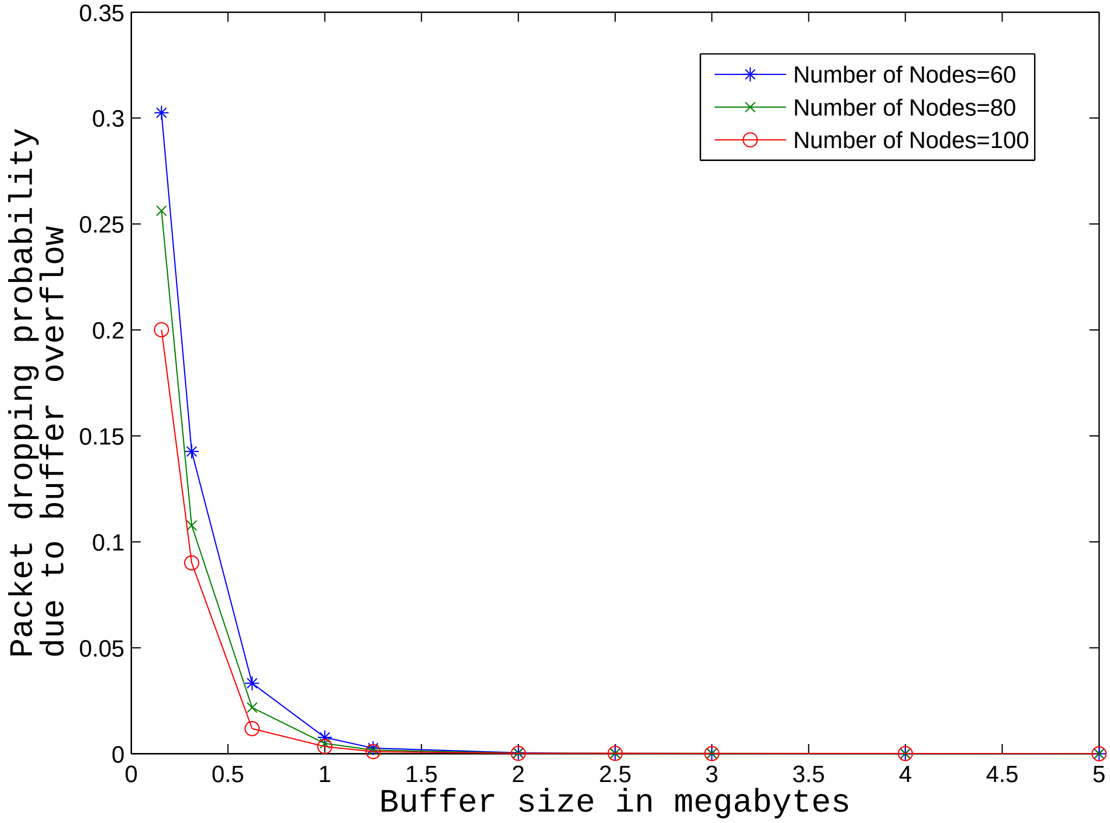

Figure 4.6: Packet dropping probability due to buffer overflow

4.4.4 Average Path Count

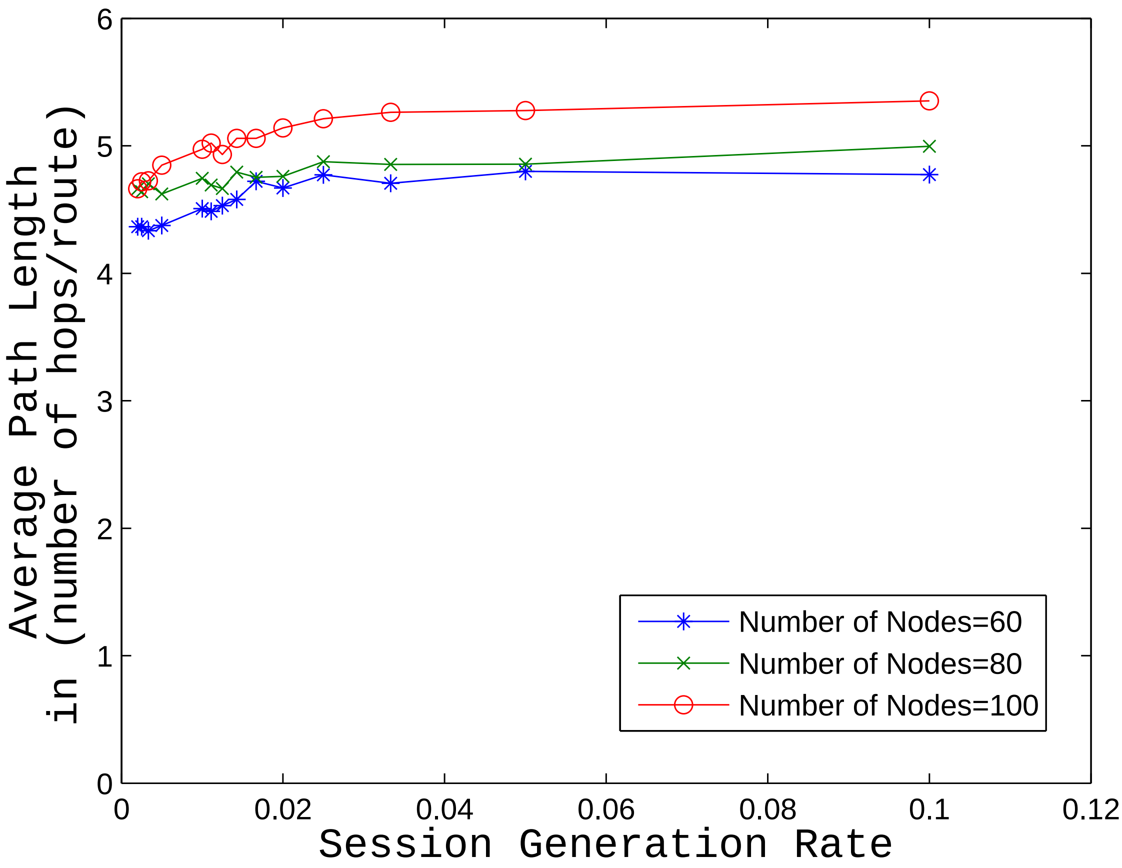

Figure 4.7: Average Number of hops taken along a path with increasing session generation rate

Fig. 4.7 shows the average number of hops taken per request. This value does not change with the request rate. There is a very small range in which it fluctuates as can be seen from the plots. Also the fluctuation is lesser for higher network size. The plots show that for a network size of 60 nodes the path length varies between 4.4 hops/route to 4.7 hops per/route. Similarly, for 80 and 100 nodes the range is between 4.6 to 4.9 and between 4.6 to 5.2 hops/route respectively.

The hops taken along a path is slightly higher for higher request rate. The shorter paths with sufficient resources to support the requests are exhausted soon and for further requests being made it becomes necessary to take longer paths to ensure quality.

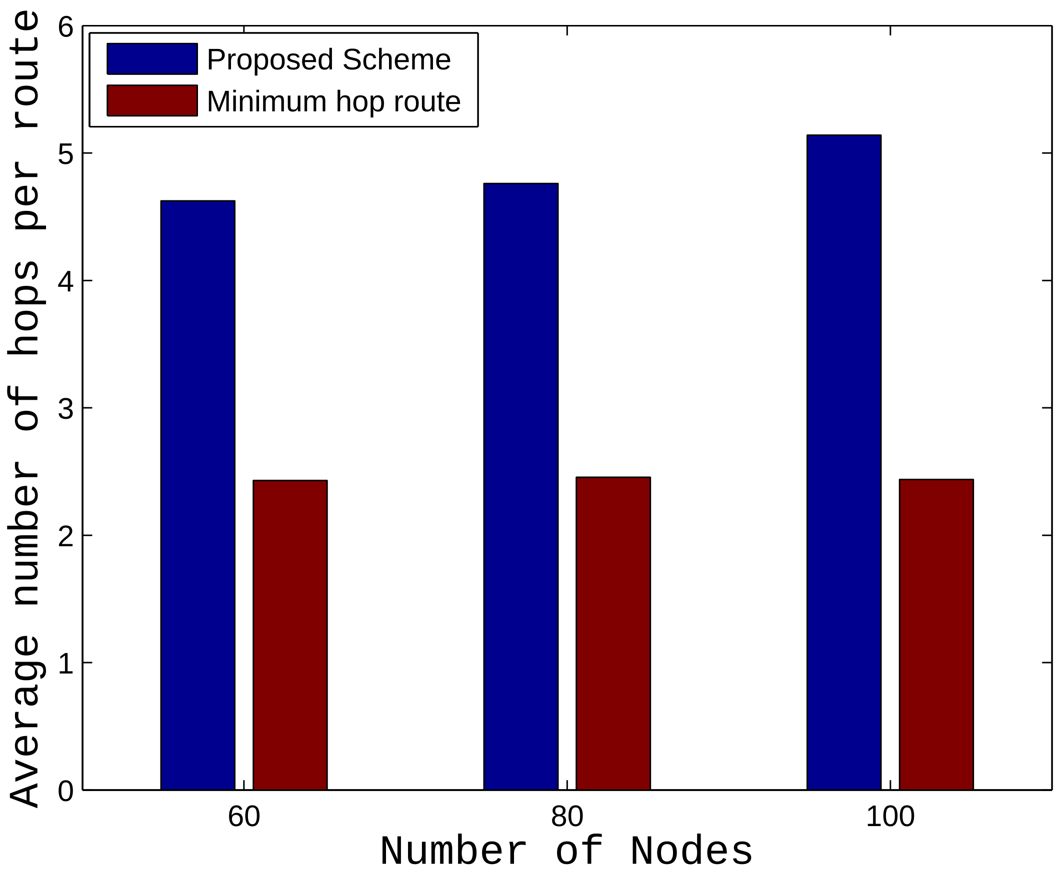

Figure 4.8: Average Number of hops taken along a path for the proposed and minimum hop scheme

Fig. 4.8 shows that the hops taken due to our proposed scheme is higher as compared to the minimum hop path scheme. Our routing protocol selects routes that offer the least delay irrespective of the number of hops.

4.4.5 Percentage of paths through minimum hop

As seen earlier in Fig. 4.8 the path count is higher for our scheme. The result in Fig. 4.9 shows that about one-third of the paths happen to be through the minimum hop path irrespective of the network size. This value further drops for higher request rates.

4.4.6 Average Packet Delay

We consider delay guarantees for assuring quality of service in our scheme. It is thus very essential to see delay performance of the scheme. Fig. 4.10 shows the delay performance for our scheme. The experiment is conducted for 500 sessions with varying session generation rates for different network size of 60, 80 and 100 nodes. There is a small variation in the delay for different network size for lower session generation rate. However gradually with increasing session generation rate the delay performance is same irrespective of the network size as well. The delay fluctuates in a very small range around 4.6 seconds, and is almost a straight line for increasing session generation rate at the value of 4.5 seconds.

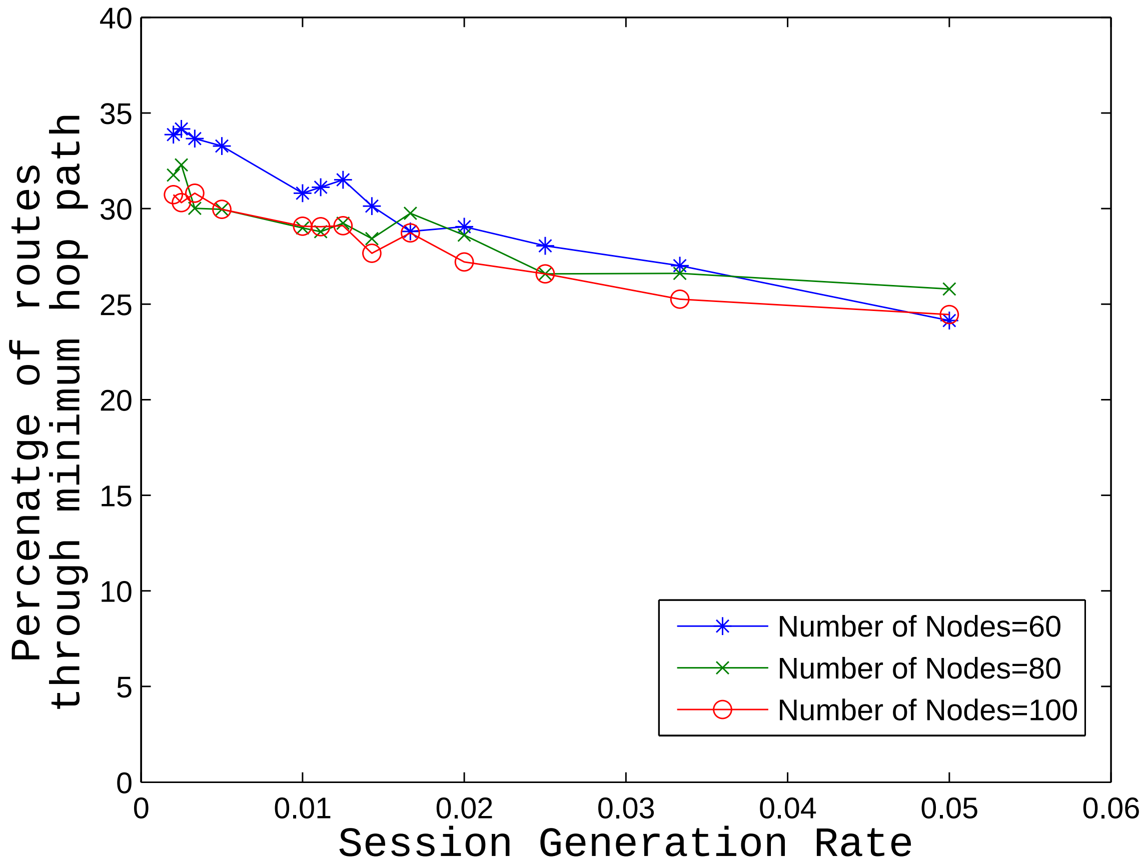

Figure 4.9: Percentage of paths through minimum hop

Thus we see that varying parameters like session generation rate and network size do not affect the delay performance adversely. The delay performance is consistent and hence we successfully achieve delay guarantee.

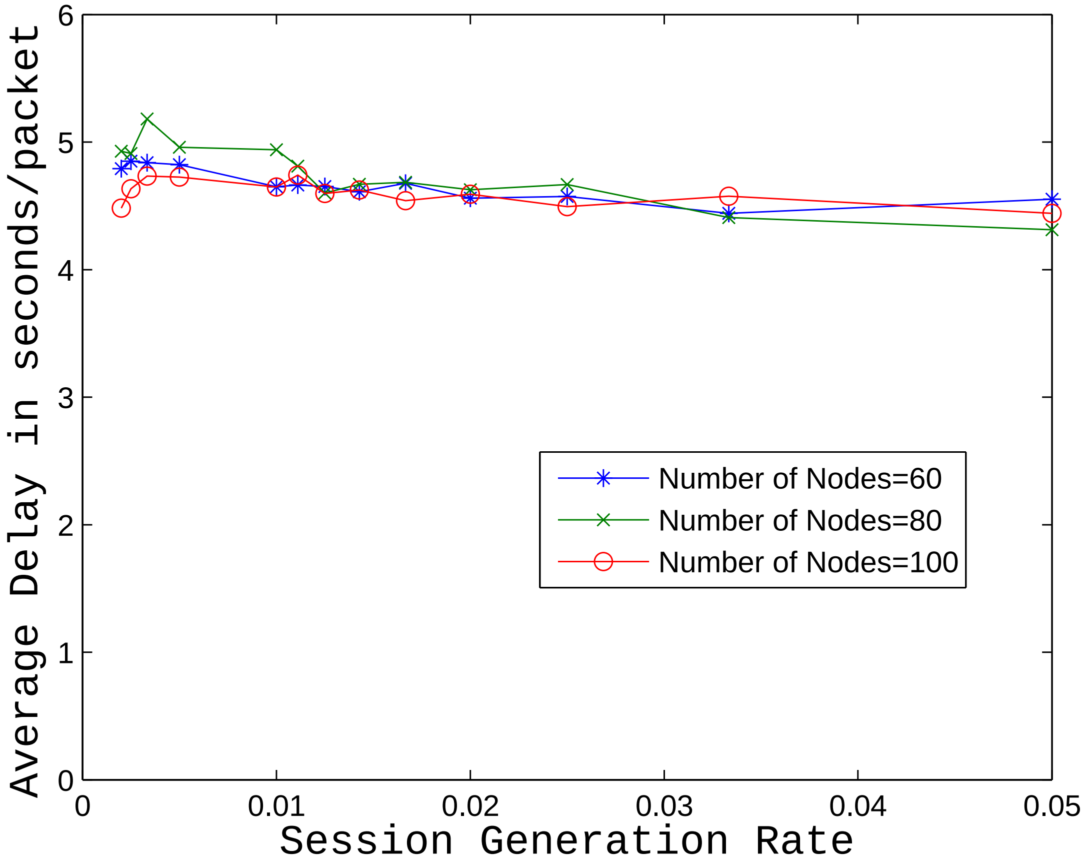

Figure 4.10: Average packet delay with increasing session generation rate

CHAPTER 5
CONCLUSIONS

In this thesis we proposed a delay based routing protocol for ad hoc networks that supports real time applications. The proposed routing protocol estimates the delay of a new session based on resource availability. If resources are available, i.e., there is enough bandwidth, then a new session is admitted and a route that yields the minimum delay from the source to the destination is found. Delay at each hop is determined using the M/M/1 model; the end-to-end delay is calculated by summing the delays at each hop. For better delay performance, the routing protocol is complimented with adaptive resource management. Resources are reserved for sessions that have already been admitted and the residual bandwidth is distributed proportionally to aid faster processing of packets. In this way, the best possible quality of service is made available to the data traffic at any time. All session requests were dealt individually as per the traffic descriptors like packet generation rate and packet size. Simulation results reveal that the proposed routing scheme adheres to the delay bounds. The results also show that the delay variation is very small irrespective of varying parameters like number of nodes and traffic generation rate. Hence, the target delay is achieved using our approach.

LIST OF REFERENCES

[1] Charles E. Perkins and Elizabeth M. Belding-Royer, "Ad-hoc On-Demand Distance Vector Routing", *WMCSA*, 1999, pp. 99–100.

[2] David B Johnson and David A Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks", *Mobile Computing*, Kluwer Academic Publishers, 1996, Vol. 353.

[3] Vincent D. Park and MobileM. Scott Corson, "A Highly Adaptive Distributed Routing Algorithm for Wireless Networks", *INFOCOM*,1997, pp. 1405–1413.

[4] Charles E. Perkins and Pravin Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for mobile computers", *ACM SIGCOMM*, 1994, pp. 234–244.

[5] Ramanathan, R. and Redi, J., "A brief overview of ad hoc networks: challenges and directions", *Communications Magazine, IEEE*, May 2002, Vol. 40, Issue 5, pp. 20–22.

[6] M. Scott Corson and Anthony Ephremides, "A distributed routing algorithm for mobile wireless networks", *Wireless Networks*, 1995, Vol. 1, Issue 1, pp. 61–81.

[7] S. Shenker and C. Partridge and R. Guerin, "Specification of Guaranteed Quality of Service", *RFC* 2212, 1997.

[8] Elizabeth M. Royer and Chai-Keong Toh, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks", *IEEE Personal Communications*, Apr. 1999, Vol. 6.

[9] D. Grossman, "New Terminology and Clarifications for Diffserv", Apr. 2002.

[10] K. Nichols and B. Carpenter, "Definition of Differentiated Services Per Domain Behaviors and Rules for their Specification", RFC Editor, United States, 2001.

[11] R. Braden and D. Clark and S. Shenker, "Integrated Services in the Internet Architecture: an Overview", 1994.

[12] Alberto López and Jukka Manner and Andrej Mihailovic and Hector Velayos and Eleanor Hepworth and Youssef Khouaja, "A Study on QoS Provision for IP-Based Radio Access Networks", *IWDC '01: Proceedings of the Thyrrhenian International Workshop on Digital Communications*, Springer-Verlag, 2001, ISBN 3-540-42592-6, pp. 139–157.

[13] Zeinalipour-Yazti Demetrios Csyiaztics, “A Glance at Quality of Services in Mobile Ad-Hoc Networks”.

[14] H. Xiao and W. Seah and A. Lo and K. Chua, “A Flexible Quality of Service Model for Mobile Ad-Hoc Networks”, IEEE VTC.

[15] Seoung-Bum Lee and Gahng-Seop Ahn and Xiaowei Zhang and Andrew T. Campbell, “Evaluation of the INSIGNIA Signaling System”, *NETWORKING ’00: Proceedings of the IFIP-TC6 / European Commission International Conference on Broadband Communications, High Performance Networking, and Performance of Communication Networks*, Publisher Springer-Verlag, London, UK, 2000, ISBN 3-540-67506-X, pp. 311–324.

[16] Seoung-Bum Lee and Gahng-Seop Ahn and Xiaowei Zhang and Andrew T. Campbell, “INSIGNIA: An IP-Based Quality of Service Framework for Mobile ad Hoc Networks”, *Journal of Parallel and Distributed Computing*, 2000, Vol. 60, Issue 4, pp. 374–406.

[17] Teresa Tung and Jean Walrand, “Providing QoS for Real-time Applications”, *2nd IASTED Communications, Internet, and Information Technology (CIIT)*, Scottsdale, AZ, Nov. 2003.

[18] R. Gibbens and F. Kelly,“Distributed connection acceptance control for a connectionless network”, *in Proc. of the 16th International Teletraffic Congress, Edinburgh, Jun. 1999.*

[19] Antonis Dimakis and Linhai He and John Musacchio and Hoi-Sheung Wilson So and Teresa Tung and Jean Walrand, “Adaptive Quality of Service for a Mobile Ad Hoc Network”.

[20] Yi Wang, “An Overview of Wireless Ad hoc Networks:Challenges and Future”, Dec. 2003.

[21] Ian F. Akyildiz, Tommaso Melodia and Kaushik R. Chowdhury, “A survey on wireless multimedia sensor networks”, *Computer Networks*, 2007, Vol. 51 Issue. 4, pp. 921–960.

[22] P. Bonnet and J. Gehrke and P. Seshadri, “Querying the Physical World”, *IEEE Personal Communications*, Oct. 2000, Vol. 7, pp. 10-15.

[23] Nirupama Bulusu and Deborah Estrin and Lewis Girod and John Heidemann, “Scalable Coordination for Wireless Sensor Networks: Self-Configuring Localization Systems”,*Proceedings of the 6th IEEE International Symposium on Communication Theory and Application*, IEEE,St. Martin’s College, Ambleside, Cumbria, UK, Jul. 2001.

[24] Alberto Cerpa and Jeremy Elson and Michael Hamilton and Jerry Zhao and Deborah Estrin and Lewis Girod, “Habitat monitoring: application driver for wireless communications technology”, *SIGCOMM LA ’01: Workshop on Data communication in Latin*

America and the Caribbean, ACM, ISBN 1-58113-354-5, San Jose, Costa Rica, 2001, pp. 20–41.

[25] J. M. Kahn and R. H. Katz and K. S. J. Pister, “Next Century Challenges: Mobile Networking for "Smart Dust"”, *International Conference on Mobile Computing and Networking MOBICOM*, 1999, pp. 271–278.

[26] Noury, N. and Herve, T. and Rialle, V. and Virone, G. and Mercier, E. and Morey, G. and Moro, A. and Porcheron, T., “Monitoring behavior in home using a smart fall sensor and position sensors”, *Microtechnologies in Medicine and Biology, 1st Annual International, Conference*, 2000, pp. 607-610

[27] Rabaey, J.M. and Ammer, M.J. and da Silva, J.L., Jr. and Patel, D. and Roundy, S. “PicoRadio supports ad hoc ultra-low power wireless networking”,Computer, Jul. 2000, Vol. 33, Issue. 7, pp. 42-48.

[28] Petriu, E.M. and Georganas, N.D. and Petriu, D.C. and Makrakis, D. and Groza, V.Z., “Sensor-based information appliances”, *Instrumentation and Measurement Magazine, IEEE*, ISBN 1094-6969, Dec. 2000, Vol. 3 Issue. 4, pp. 31-35.

[29] Shree Murthy and J. J. Garcia-Luna-Aceves, “ An Efficient Routing Protocol for Wireless Networks”, *MONET*, 1996, Vol. 1 Issue 2, pp. 183-197.

[30] Dimitri Bertsekas and Robert Gallager, “Data networks (2nd ed.)”, Prentice-Hall, Inc., 1992, ISBN 0-13-200916-1.