TOWARDS ACHIEVING QOS GUARANTEES IN MOBILE AD HOC NETWORKS

A Thesis

Submitted to the Faculty

of

Drexel University

by

Harpreet S. Arora

in partial ful£llment of the

requirements for the degree

of

Master of Science

October 2003

Acknowledgements

Several people supported my efforts during my graduate years and I would like to thank them. First of all, I must point out the invaluable help of my advisor, Dr. Lloyd Greenwald. This dissertation would not have been possible without his encouragement and valuable guidance. Special thanks to him for giving so much time and attention to my dissertation and helping me with the documentation and presentation.

I would like to extend my thanks to Dr. Harish Sethu, who gave me an opportunity to work with him and explore the £eld of computer networks. I started my work in the £eld of Ad Hoc Networks under his guidance. Thanks for all the encouragement, guidance, help and support. I would also like to thank him for serving on my committee.

I want to thank Dr. Vassilis Prevelakis for serving on my committee.

I would like to thank John for helping me with the administration of my computer. I want to thank my friends Sachin, Sundar and Arvind for their help, advice, inspiration and company during my extended hours in the lab.

Thanks to Salil for helping me with the formatting of the thesis. I am thankful to Mallika for helping me with the editing of the thesis.

Last but not least, I would like to thank my family for their constant support and guidance.

Table of Contents

List of Tables

List of Figures

Abstract

TOWARDS ACHIEVING QOS GUARANTEES IN
MOBILE AD HOC NETWORKS
Harpreet S. Arora
Lloyd Greenwald, Ph.D.

Mobile Ad Hoc Networks represent future generation wireless networks, with a high degree of versatility and robustness, capable of being deployed quickly and economically at places lacking any infrastructure. The development of these networks has been hindered by the characteristics of these networks such as bandwidth scarcity and ¤uctuations, node mobility, hostile working conditions and battery power constraints. Projected applications of such networks include defense based applications (war scenarios), disaster relief operations (earthquake, rural areas) and commercial applications (home networking, extending Internet connectivity). Many of these applications require a distinction in the quality of connections being supported in terms of bandwidth availability, end-to-end delay and jitter. For such applications, the need of a QoS model integrated within the nodes is impervious. Such a model must have the ability to distinguish ¤ows based on their QoS needs and have mechanisms that work to meet those requirements. Further, since all nodes are peers, the QoS model must operate in a fully distributed manner without relying on a central coordinator. Such requirements combined with the hostile working conditions of an ad hoc network make the task of designing such a model very challenging. We study these challenges in detail, survey the available QoS models, their abilities and drawbacks in providing QoS guarantees to ¤ows in a MANET. We then propose a novel QoS model that differentiates the ¤ows into classes and attempts to provide bandwidth and delay guarantees to ¤ows of highest priority class.

In the £rst part of the dissertation, we study the challenges that lie in the way of designing a QoS model for MANETs. In particular, we identify synchronization effects associated

with poor control timing of routing protocols. Such effects can lead to tremendous waste of network resources and can cripple the network during critical times. As a part of our novel work, we propose schemes to pacify such effects.

In the second part of the thesis, we present a thorough survey of the proposed QoS schemes for the wired Internet and MANETs. We realize that the QoS schemes for the wired Internet cannot be directly applied to a MANET environment. The schemes are differentiated into three categories and each one is studied in detail. We analyze and compare the performance of two of these models in detail. We also present the shortcomings of each model. The understanding of their mechanisms helps us form the basis of our model.

In the third part, we present the design of our novel QoS model that uses the basic components of the Diffserv architecture and de£nes additional functionality to realize our goals as stated in the problem statement. We implemented our QoS model as a part of the Network Simulator (NS-2). We present simulation results to validate the working of the model. The results clearly show that the performance improvement of the high priority ¤ow over that of SWAN and Diffserv.

1. Introduction

From the advent of Arpanet to the current day Internet and wireless cellular networks, the technology of networking world has come a long way and has in¤uenced the daily life of man heavily. Networking solutions have become a part and parcel of the modern life. Despite the rapid advances in technology and development of networking infrastructure, there are voids where the conventional networks do not £t in suitably, either because of economical reasons or tactical. Such places as hill-stations and battle £elds and times as natural catastrophes demand the deployment of networks rapidly and conveniently which can form the basis of communications and data transfers. Mobile Ad Hoc networks are projected as a solution under such circumstances. Formally, a mobile ad hoc network consists of a collection of wireless nodes, all of which may be mobile and co-operate to form a wireless network dynamically, without the need for any infrastructure or administrative support. In common terms, a MANET is a network formed by a collection of mobile nodes that can detect the presence of each other and con£gure themselves to form a network. This concept is different from a set of walkie-talkies where the nodes can only communicate directly if they are within radio range of each other. In a MANET, a node desiring to transmit to a distant node may establish communication via intermediate nodes that are ready to relay packets for the source node. Thus nodes within a MANET are capable of acting as hosts as well as routers.

Such networks offer unique bene£ts and versatility for many applications and scenarios such as disaster relief and battle £elds. Projected applications of such networks include but are not limited to:

1. Rescue and search operations

2. Battle £eld, communication between a group of soldiers

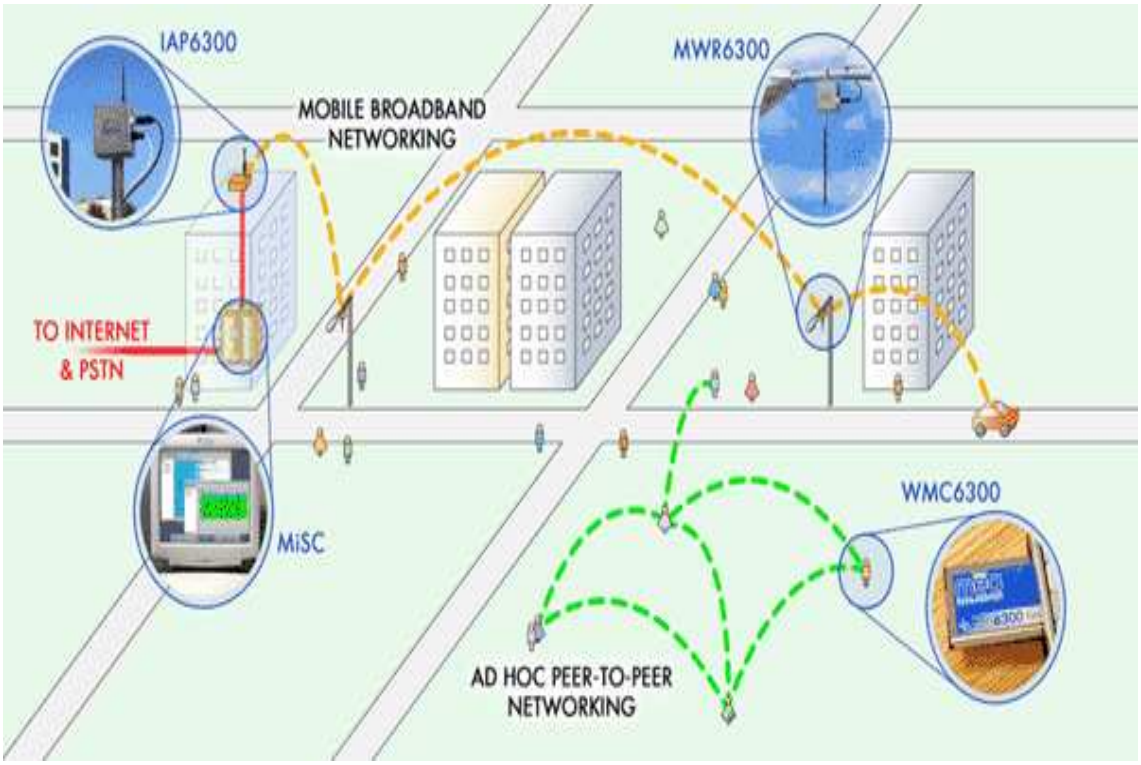

Figure 1.1: Mobile Ad Hoc Network in Action.

3. A group of islands and ships communicating with the help of ¤oating balloons and passing airplanes

4. A group of people with portable computers forming an ad hoc network to share data in a conference room

5. A group of people on a trekking trip with handsets that form an ad hoc network

A majority of these applications involve voice communications while some may require video transmissions (Command and Control in a war or disaster relief operation). Commercial deployment of these networks will certainly make use of multimedia based applications such as streaming audio and video. These applications demand uninterrupted and clear connections for their lifetime. Further, in war and disaster relief scenarios, some connections may be more important than others. A message of extreme urgency would want to reach the destination without any network delay or disruption. These requirements cannot be satis£ed by the current TCP/IP model of the Internet which can only support best-effort service. Although TCP assures that the packets reach their destination safely, it does not guarantee any bound on the delay, nor can it assure a minimum bandwidth for the connection. Thus there is a need for a QoS architecture that can provide the required service differentiation as well as deterministic service quality to the demanding connections. An ideal QoS model should be able to differentiate ¤ows on the basis of priorities and provide deterministic service guarantees to the admitted ¤ows while making a good utilization of the network resources. Many QoS models have been proposed for the conventional Internet, which are discussed in detail later. These models can be broadly classi£ed into two categories : Integrated Services (Intserv) based and Differentiated Services (Diffserv) based. However these models are not directly applicable to the paradigm of ad hoc networks due to the characteristics of these networks such as mobile nature of nodes resulting in unpredictable topology, scarce wireless bandwidth which varies at the mercy of environmental conditions, limited power of the nodes and peer-to-peer nature of nodes

requiring co-operation for relaying other's packets among many. These characteristics not only make ad hoc networks very different from the conventional Internet, but also make the task of providing QoS assurances in these networks extremely challenging. We discuss the challenges posed by these networks toward the development of a QoS model in detail in Chapter 3. In our quest for a perfect QoS model, we investigate in detail the effect of routing protocols on QoS in MANETs. The study is performed using Network Simulator (NS-2), one of the most popular simulators used by the MANET research community. In doing so, we discover the occurrence of a synchronization effect due to periodic properties of routing protocols. The effect poses a threat to providing hard QoS guarantees in these networks. We study the causes of this effect in detail and some simple schemes to pacify the effect.

Several QoS schemes have been proposed for MANETs which are either a modi£cation of the conventional Intserv and Diffserv based models or schemes that add QoS awareness to the MAC layer or network layer. Most of these schemes provide soft or relative QoS assurances. With such schemes, the requested resources may not be available for the entire duration of the connection. Many of these schemes are evaluated under simplistic scenarios and do not provide adequate differentiation. Some of the schemes also require a modi£cation of the network layer and MAC layer.

We aim to have a QoS model that can provide adequate service differentiation to the traf£c ¤ows. We desire that the ¤ows of highest priority class achieve tight QoS guarantees, while ¤ows of other classes (except the best-effort traf£c) get soft guarantees. We further aim to make the model independent of the existing routing and link layers.

In this thesis, we design a QoS model that ful£lls the above requirements. The model differentiates traf£c into classes and attempts to maintain delay and bandwidth bounds for traf£c of highest priority class. To £nd an appropriate starting point, we perform an extensive literature survey of the existing schemes. In particular, we delve into the details of two classless QoS schemes, Diffserv and SWAN, that appeared as potential candidates

after our initial survey. A detailed understanding of their mechanisms and a comparative analysis helps us to choose mechanisms for our model to achieve our goal. Diffserv forms the basis of the new model, which is bolstered with more mechanisms to support tight QoS guarantees. In our initial work, we focus on providing tight guarantees to ¤ows of highest priority class. The model makes very simplistic and realistic assumptions about the network and traf£c characteristics. The details of this model and the associated mechanisms are presented in Chapter 6.

To evaluate the model, we perform extensive simulations using NS-2. Simulations are performed under varied mobility and traf£c conditions. The results of the simulations are presented in Chapter 7. The performance of the new model is compared with that of existing QoS models such as Diffserv and SWAN. A high improvement in the packet delivery ratio and end-to-end latency of the highest priority ¤ow with the new model can be clearly seen. We £nally conclude with discussion of our future work and directions we will take to explore the full potential of the model.

2. Problem Statement

In this chapter, we present our problem statement which acts as our guideline towards the goal. We £rst de£ne the characteristics of the network that we consider for our experiments and then the problem statement. In doing so, we try to ensure that the network and traf£c characterictics cover a wide range of scenarios while still being speci£c enough to provide us with a de£nite guideline at all times.

2.1 Network Characteristics

Consider a set of wireless nodes placed within a restricted region to form an ad hoc network. The transmission and reception range of each node is £xed at r and the interference range of each node is i where, $i \geq r$. The density of the nodes is such that the network is fully connected (there are no partitions within the network). This assumption is made, since a partition within the network will make the QoS framework dependent on the routing protocol, especially during times of link breaks across the partitions, thus making the evaluation of the model complicated. We also assume that all nodes within the network are cooperative. In our work, we do not consider security as an issue. We therefore assume that none of the nodes are hostile. Some of the nodes may choose not to relay traf£c belonging to other ¤ows. Such nodes can be considered as dead from the perspective of the routing protocol. This does not affect the functioning of our protocol.

For our simulations, we consider a network with a maximum of 50 nodes within an area of 1500 m X 300 m. Each node is capable of sending, receiving, forwarding and differentiating the following types of traf£c :

1. UDP traf£c (voice, video and data)

2. TCP traf£c (data)

Each node that is a source of traf£c must classify the traf£c into one of the classes as described below.

2.2 Traf£c Characteristics

1. **Highest priority class :** The traf£c belonging to this class is voice traf£c sent at a rate of 20kbps in the form of 80 byte packets sent at a rate of 32 packets/second. The total amount of traf£c in the network belonging to this class must not exceed 5% of the total network capacity.

2. **Medium Priority Class :** This class supports other UDP real time traf£c. Real time traf£c is modeled as constant-bit rate (CBR) traf£c. The sources of such ¤ows generate packets at a constant rate depending upon the applications being supported. For our simulations, we support voice traf£c (same characteristics as for highest priority class) and video traf£c (generated at a rate of 64kbps, 80 byte packets). The traf£c sources start at random instants of time and can be in the form of small bursts or long lasting ¤ows.

3. **Best Effort Traf£c Class :** All TCP traf£c and remaining UDP traf£c.

2.3 Problem Statement

Since mobility of nodes adds another dimension to the problem of providing QoS guarantees, we consider a static network separate from a dynamic network and de£ne a problem statement for each. The two **problem statements** are presented below.

2.3.1 Static Networks

In such a network, all nodes are placed randomly within the speci£ed region, however their positions are £xed for the entire duration of the simulation. All of the above assumptions apply. We aim to provide the following:

a. Ensure that the ¤ows belonging to the *highest priority class* get preemptive priority over traf£c of other classes. After the initiation of any new ¤ow belonging to this class, the network must detect and stop any interfering ¤ows belonging to the lower classes in less than t_1 seconds. Thus a ¤ow belonging to this class must be guaranteed to have resource availability in less than t_1 seconds after its initiation. Flows belonging to best-effort traf£c should be preempted before preempting any ¤ows of Medium priority class. In doing so, it must be ensured that only those ¤ows that interfere with the highest priority ¤ow to the extent that they lower the resource availability of highest priority ¤ows to less than their requirements are preempted. Thus if a node carrying a highest priority ¤ow (@ 32 packets/sec) is not able to transmit more than p packets per second for s seconds, it considers that the ¤ow is under interference from ¤ows of lower priority classes and makes provisions to preempt the interfering ¤ows until it is able to attain a transmission rate of 32 packets/sec again. The node under interference must preempt only those lower priority ¤ows as are necessary to reattain the rate of 32 packets/sec for its highest priority ¤ows.

b. Attempt to maintain end-to-end delays of less than 200 ms for any ¤ows belonging to *medium priority class*. This should be done at the expense of the rate control of UDP and TCP ¤ows belonging to the best-effort traf£c. However, the guarantees to the ¤ows of this class are soft and may be void in the event of changes in network traf£c conditions ot topology. Some of the ¤ows may be preempted or demoted to best-effort to accommodate ¤ows belonging to highest priority class and other ¤ows belonging to the same class.

2.3.2 Dynamic Networks

These networks differ from the static networks, since the movement of nodes results in time varying topology. Because of this, the network performance is heavily dependent upon the performance of the routing protocol which attempts to keep the network connected despite the changes in topology. Under these conditions, it is not wise to de£ne bounds on the achievable throughput or delay. Hence, for dynamic networks, we aim to provide guarantees which are as tight as possible subject to the performance of the routing protocol.

In such a network, each node is capable of moving randomly within the speci£ed region. The average speed of each node de£nes the mobility of the network. All the assumptions stated at the beginning apply here. Within these networks, we aim to :

a. Ensure that the ¤ows belonging to highest priority class get preemptive priority over traf£c of lower classes. After the initiation of any new ¤ow belonging to this class, the network must detect and stop any interfering ¤ows belonging to the lower classes within t_2 seconds *provided that the path to the destination is available*. To maintain the characteristics of highest priority ¤ow, ¤ows belonging to best-effort traf£c must be preempted before preempting any ¤ows of medium priority class. In doing so, it must be ensured that only those ¤ows that interfere with the highest priority ¤ow to the extent that they lower the resource availability of highest priority ¤ows to less than their requirements are preempted, thus maximizing network resource utilization.

b. Attempt to maintain end-to-end delays of less than 200 ms for any ¤ows belonging to the medium priority class, subject to the availability of the path between the source and destination. This should be done at the expense of rate control of UDP and TCP ¤ows belonging to best- effort traf£c. However, the guarantees to these ¤ows are soft and may be void in the event of changes in network traf£c and mobility conditions. Some of the ¤ows may be preempted or demoted to best-effort to accommodate ¤ows belonging to highest priority class and other ¤ows belonging to the same class.

3. QoS Challenges In MANETs

Ad hoc networks differ radically from the conventional wired or cellular networks. The characteristics of these networks makes the task of providing QoS guarantees in these networks extremely dif£cult. Some of these dif£culties are well known and well pointed out in many papers including [1], [2] and [3]. These challenges are discussed brie¤y below. During our research, we also found some unique problems associated with the most popular routing algorithms for MANETs. These are discussed in Section 3.2. We also present some dif£culties we encountered while designing the model for providing QoS guarantees.

3.1 Well Known Challenges

3.1.1 Unpredictable Network Topology

An ad hoc network consists of nodes that can move arbitrarily in random directions and with different speeds. Further, these nodes are not af£liated to any £xed base station. The network formed is thus dynamic and has an unpredictable, time varying topology. The rate of change of topology depends upon the speed and the movement patterns of the nodes. For example, a collection of ships moving in the same direction may have a stable topology even if they are moving fast. The constant change in topology results in inaccurate state information at the nodes making convergence of routing information dif£cult. In earlier work [4], we showed that the capacity of the network and the utilization of its resources change with the speed of the nodes. This implies that if the parameters of a QoS model within the nodes of the network are tuned to operate at a particular speed, the model might fail to provide assurances if the nodes start to move faster. Then the parameters must either be re-tuned, or the nodes must be made aware of the average speed of movement of the network nodes, so that the parameters can be adjusted dynamically. It was also seen that at

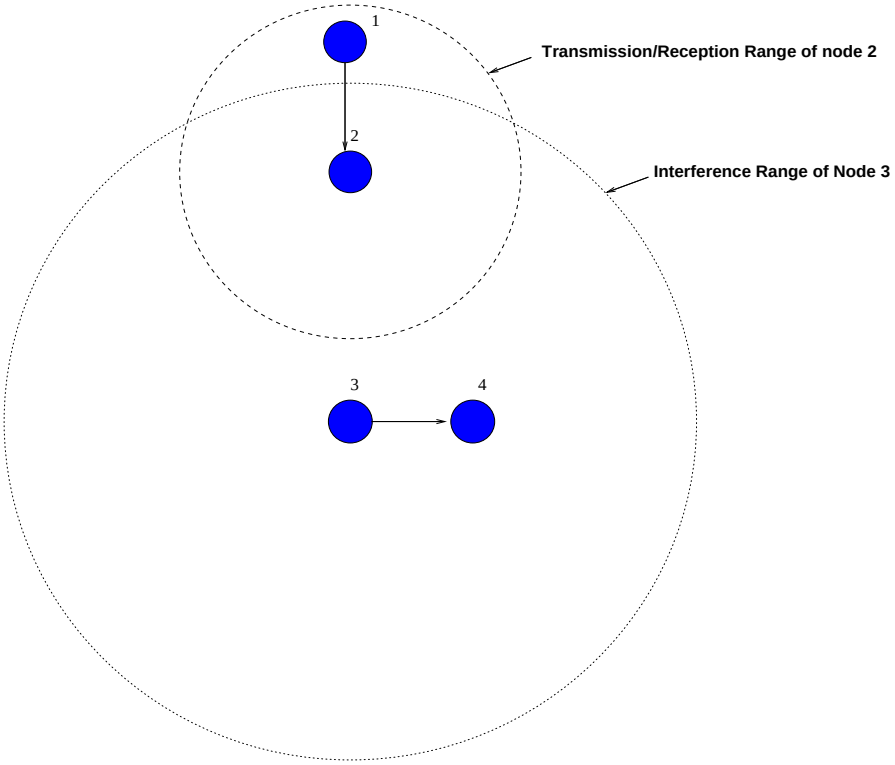

Figure 3.1: Interfering Nodes

very high speeds, the network capacity degrades exponentially due to excessive link breaks. In order to make any assurances, it is important that the rate of change of topology is not ”too” fast, otherwise, ¤ooding may be the only routing solution. In our simulations, we use the random way-point model to generate random scenarios and test them under varying conditions of node movement and mobility.

3.1.2 Scarce Radio Bandwidth

Scarcity of radio spectrum implies that only a limited amount of bandwidth will be available to a MANET node. This bandwidth is further in¤uenced by external conditions

such as weather, physical obstacles, fading, noise and interference from outside sources and from other distant nodes. Other factors that affect the available bandwidth are the well known hidden terminal and exposed terminal problems. Protocols have been designed to reduce the effect of these problems, though none eliminate them completely. The IEEE 802.11 protocol [5], for example, speci£es RTS-CTS-DATA-ACK sequence to overcome the hidden and exposed terminal problems. However, this mechanism is only effective in reducing the effect of these problems but cannot eliminate them completely. A simple scenario where these mechanisms do not work is when two nodes are within interference range of each other but not in the transmission range. Such a scenario is shown in Figure 3.1.

As shown, node 1 has data to transmit to node 2 and node 3 has data to send to node 4. When node 1 wants to send a data packet, it waits for the medium to be idle for a speci£ed period of time. It then transmits a RTS (Request-to-Send) packet. Since node 1 is outside the interference range of node 3, it always senses the medium idle. If node 3 transmits a packet to node 4 while node 1 transmits to 2, a collision takes place at node 2. Then, node 2 does not receive the RTS packet correctly. The collision can also take place for a data packet. If, upon receiving the RTS packet, node 2 senses the medium idle, it replies with a CTS (Clear-to-Send) packet. This packet informs the neighboring nodes about the forthcoming transmission of data packet and hence effectively reserves the medium for the duration of transmission of the packet. However, node 3 is outside the reception range of node 2, hence it does not receive the CTS packet. Node 1 receives this packet and assumes that the medium has been reserved. It then starts its data packet transmission. If node 3 now starts transmitting a packet to node 4 before sensing the medium to be busy, a collision takes place at node 2, resulting in loss of data packet. These collisions reduce the effective bandwidth at node 2. In such a situation, node 2 has no means of reserving the medium for transmission or reception. In that case, it cannot accurately estimate the bandwidth availability for making long term reservations. With the unavailability of accurate information, mechanisms such as admission control and rate control for the maintenance

of admitted ¤ows is very dif£cult. Further, the limitation of available bandwidth requires that the exchange of control messages be limited. This restriction may require that nodes transmit control updates less frequently or restrict them to immediate neighbors. This results in nodes having stale information of the network conditions till the next update and further delays the convergence of control information.

3.1.3 Limited Battery Power

The mobility of nodes limits their size, which in turn limits the energy reserves available to them. Thus energy conservation is a key requirement in the design of ad hoc networks. Transmission of packets is the single largest consumer of power. Transmission power control used for communications impacts the operational lifetime of devices in different ways. For devices, where the transmission power accounts only for a small percentage of the overall power consumed, (e.g., a wireless LAB radio attached to a notebook computer) reducing the transmission power may not signi£cantly impact the device's operational lifetime. In contrast, for small computing/communication devices with build-in or attached radios (e.g. cellular phones, PDAs, sensors, etc.) reducing the transmission power may signi£cantly extend the operational lifetime of a device, thus, enhancing the overall user experience. Hence for such devices, it is necessary that the number of control packets transmitted are minimized. For a QoS framework, it is necessary to exchange control information related to bandwidth availability in the neighborhood, admission control and quality feedback. A reduction in the transmission of such packets leads to the problems related to inaccurate state information and may cause ¤ows to lose their quality. Designing a protocol with contradicting goals of achieving desired levels of QoS while reducing power consumption due to transmissions is a challenging task. In the design of our protocol, we aim to minimize the number of transmissions of control packets thereby making our protocol a power-aware QoS protocol.

3.2 Synchronization Effects

In addition to these well known problems, we discovered problems associated with the most popular routing algorithms for MANETs. We discovered the occurrence of synchronization effects which can result in sudden and sharp drops in the throughput of the network. We study these effects in detail and suggest some means to overcome them. The detailed study is presented below.

3.2.1 Overview

Any discussion of a Quality of Service (QoS) solution for mobile ad hoc networks (MANETs) must address two predominant features of these networks: mobility and wireless communications. One thing mobility and wireless communications have in common is that both cause link breakage. Link breakage causes throughput to drop [4] while link layer and network layer protocols combine to re-connect the network through breakage detection, link repair, and/or re-routing. As has been previously pointed out [2], in order to provide QoS, networking protocols must trade-off rapid response to link breakage with bandwidth consumed by protocol control messages. These trade-offs directly in¤uence achievable throughput and latency.

While comparing the behavior of various routing protocols we often observed sharp drops in throughput from nearly 100 percent to nearly 0 percent with only small increases in node mobility. Our analysis reveal synchronization effects that cause these phenomena. The synchronization effects are caused due to frequent link breakages and the routing protocol's attempt to heal them to restore the link. Such phenomena can result in ine£cient use of resources, jeopardizing QoS guarantees during critical times. We investigate the causes of these phenomena in detail and suggest modi£cations to networking protocols to overcome these problems. Our resulting simulations show signi£cant improvements in removing such drastic throughput variations.

We initially observed synchronization effects in large scale random networks. However, the effects can be obscured or averaged out when summarizing the results of multiple sets of large scale simulations. Additionally, due to random movement and the large number of the nodes, these effects are dif£cult to analyze. To truly understand and explain these effects we study a simple network and mobility pattern where the effects can be carefully manipulated. Solving the problems of this particular network and mobility pattern is not our intention. Our intention is to use this scenario as the basis of a focused study that illuminates the interaction between link breakages and routing protocols. To enhance the effect, we initially demonstrate and analyze it by considering periodic movement of nodes. However, we later show that such a phenomenon is as likely to occur in real world situations where the movement of the nodes is stochastic rather than deterministic. Our simulations prove that even when the movement of the nodes is random, such an effect can cause unexpected drops in the throughput. We consider a real life scenario and show the occurrence of this effect in¤uencing the throughput of ¤ows within the scenario. Our study extends to other forms of ad hoc network such as sensor networks [6] where rapid link breakages may be caused by mechanical vibrations, interfering radio signals, or ¤uctuations in battery power.

3.2.2 Rises and Drops in Throughput

Initially, we focus on a single-path network with one source and one destination. Breaks in this scenario are caused by the periodic movement of one of the nodes. All other nodes remain immobile. We simulate this network and study the behavior of the DSR (Dynamic Source Routing) protocol, one of the more popular and mature routing protocols available to MANETs. We study DSR's effectiveness in healing the broken link and its ineff£ciencies. The results we obtain, however, are applicable to any routing protocol that employs periodic dissemination of routing information. Later, we look at complex scenarios with

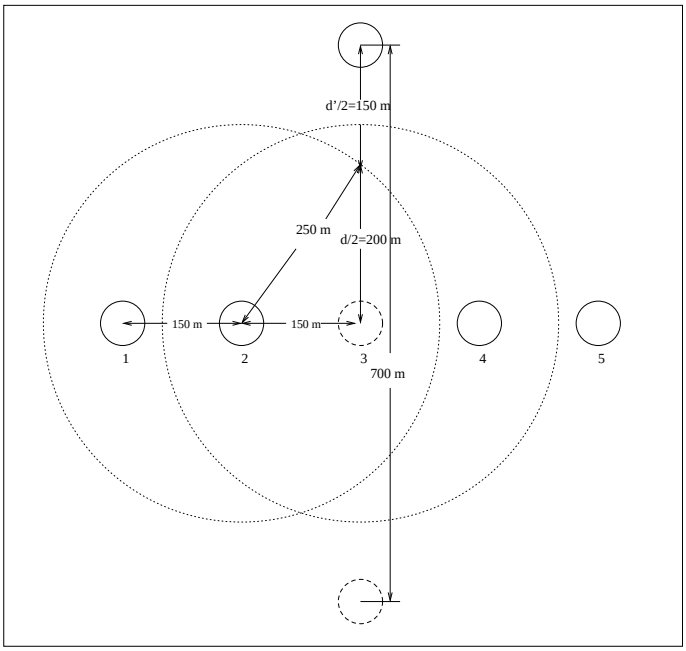

Figure 3.2: Simulation scenario

random formation of nodes where nodes move with random speeds and hence gives rise to stochastic breaks rather than periodic. Our simulations show the presence of synchronization effects in these scenarios affecting the throughput of the ¤ows.

3.2.2.1 Simulation Scenario

Our simulation environment consists of the NS-2 simulator with the CMU Monarch wireless extensions [7]. The target scenario consists of 5 nodes placed as shown in Figure 3.2. The radio range of each node is £xed at 250 meters. The nodes are placed 150 meters apart such that they are only within the range of their immediate neighbors. The middle node oscillates up and down; at the extreme points it is out of range of the other

nodes. This node pauses at the extreme point for some time and then begins moving toward the other extreme. The total distance between the two extremes is 700 meters. The node is within the range of its neighboring nodes for the center 400 meters of this traversal. The £rst node (node 1) acts as a constant bit rate (CBR) source, sending packets at a constant rate to the destination node (node 5) at the other end. Packets of size 256 bytes are sent at a constant rate of 3 packets/second. DSR is used as the routing protocol. Packets can ¤ow from the source to the destination only when the middle moving node is within the range of its immediate neighbors. We vary the speed of the middle node and its pause time at the extreme points.

3.2.2.2 Simulation Results

Figure 3.3 shows the throughput of the constant bit rate ¤ow as measured at the destination, for various speeds of the middle node and for two different pause times at the extreme points. The pause time represents the duration of time the node stops at each extreme point of its traversal before reversing its direction. Figure 3.3 shows unexpected rises and drops in throughput at certain speed changes. In the initial part of the graph, for low speeds, we see abrupt rises in the throughput. At one particular speed increment we see throughput rise sharply to a very high value. At a later speed increment, the throughput drops sharply to nearly 0 percent. This steep drop is a result of a degenerate synchronization effect. We study these effects, their causes and resolutions in subsequent sections.

3.2.3 Throughput and Link Detection Latency

3.2.3.1 Detecting and Healing a Broken Link

To understand the abrupt rises in throughput, we £rst study the functioning of both the network layer and the link layer when a node detects a broken link. IEEE 802.11 is implemented at the MAC layer. For each packet transmission, the IEEE 802.11 protocol at

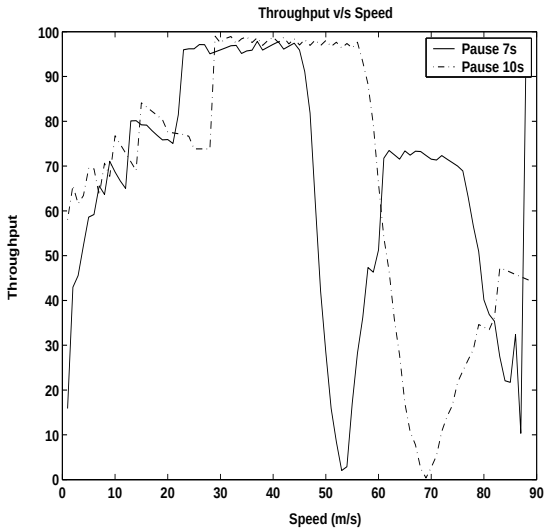

Figure 3.3: Throughput v/s Speed for pause times of 7 seconds and 10 seconds. The plots show unexpected rises and drops at certain speed changes.

each node follows an RTS-CTS-data-ACK sequence. In our scenario, when the MAC layer at node 2 is unable to get a response for seven continuous RTSs, it concludes that the link is broken and reports the same to the DSR agent in its network layer. This interaction between the link layer and network layer is termed *link layer noti£cation* [8]. Such interaction is an effective mechanism for reactive protocols like DSR and AODV to quickly detect broken links. Many proactive protocols like OLSR, do not by default support link layer noti£cation. These proactive protocols rely completely on the network layer to detect and heal a broken link.

On being noti£ed of the broken link, the DSR agent at node 2 sends an RERR (route error) packet back to the source node (node 1) to report about the link. When the source node receives the RERR packet, the DSR agent at its network layer stops relaying the data packets that originate at its application agent to its MAC layer. The DSR agent at the source

node then starts to search for an alternate path. It also starts buffering the packets arriving from the CBR source at its application layer. The router layer buffer in our simulations has a capacity to hold 64 packets. To search for a new path, the DSR agent at the source node broadcasts RREQ (route request) packets using an expanding ring search. The £rst packet is broadcast with a TTL (time to live) of 1 so that it only reaches its immediate neighbors. When no response is received within the timeout period (approximately 50 ms), the next RREQ packet is sent with a TTL equal to 16, which is the maximum number of hops from any source to any destination. The node then waits for a longer period of time before sending the next RREQ packet, again with a TTL of 1. The long timeout period is initially set to 2 seconds, then 8 seconds and subsequently 10 seconds. Henceforth, RREQ packets are sent in sets of 2, £rst with a TTL of 1 and upon its timeout, a second one is sent with a TTL of 16. A long timeout period of 10 seconds is maintained to keep the routing overhead low.

In our simulation scenario there is only one path from source to destination. A link in this path breaks when node 3 moves out of the radio range of its neighbors. Since there is no alternate path to discover, the link is healed and path re-detected when node 3 moves back into radio range of its neighbors and receives an RREQ packet. Since these packets are sent by the source node (node 1) approximately every 10 seconds beyond the £rst 3 attempts, there can be a delay in the detecting and healing of the broken link after node 3 moves back in range.

We de£ne link detection latency, denoted by L, as the amount of time taken by the routing protocol to detect the link after the node is back in range. L is obviously a characteristic of the routing protocol and is an indicator of the ef£ciency of the routing protocol. A good routing protocol should have a low value of L while maintaining low control overhead. We show in Section 3.2.4 that link detection latency is one of the major causes of observed abrupt changes in throughput. Before that, in the following subsection, we develop expressions for link detection latency and throughput for the given scenario.

3.2.3.2 Analytical Expressions

The expressions derived in this section are used in the subsequent sections to explain rises and drops in throughput. We de£ne throughput as the ratio of the number of packets reaching the destination to the total number of packets sent by the application agent. In our simulations, the middle node is moving periodically. Hence we can divide the simulation time into cycles. A cycle consists of the middle node starting from an extreme point, traversing through the reception range of its neighboring nodes, reaching the other extreme and pausing there, until it is ready to go back. With respect to a cycle, throughput can be found by measuring the number of packets arriving at the destination in a cycle and dividing that by the total number of packets that are sent by the source node in that cycle. If we de£ne d as the distance over which the middle node is in range of its neighboring nodes (see Figure 3.2) and s as the speed of the node, then the total cycle time that the middle node spends in the connectivity of its neighboring nodes is

$$T_{connectivity} = \frac{d}{s} - L \tag{3.1}$$

where L is the link detection latency as de£ned in the previous subsection. L is a function of the speed of the node and the time interval between two RREQ packets. To analyze the dependency of throughput on speed, we need to analyze the value of L.

We note that after the source node is informed of the link breakage, it sends RREQ packets initially at intervals of 0 seconds, 2 seconds, and 8 seconds, and subsequently in 10 second intervals. We de£ne I as the maximum interval between two RREQ packets. The link is detected when the moving node receives one of the RREQ packets after the node is back in range. Thus the sum of the time the node is out of range and the time the node is in range before it receives the RREQ packet is an integral multiple of I. We de£ne d' as twice the distance from the point the middle node goes out of range to the extreme point of its trajectory (see Figure 3.2) and p as its pause time at the extreme point. The link detection

latency can then be expressed as,

$$(\frac{d'}{s} + p) + L = Ik$$

or, equivalently,

$$L = Ik - \frac{d'}{s} - p \tag{3.2}$$

where k is a positive integer chosen such that

$$0 \leq L \leq I \tag{3.3}$$

that is,

$$|I(k - 0.5) - \frac{d'}{s} - p| \leq 0.5I \tag{3.4}$$

As soon as a connection is re-established (i.e. path re-detected), the packets that accumulated in the network layer buffer of the source node while the middle node was out of range are transferred to the destination. If there are b packets in the buffer and these packets are transferred at a rate r' (including transmission time, negligible propagation and processing times, and no further queuing delays), then the time available to transfer these packets is,

$$\tau = min(\frac{d}{s} - L, \frac{b}{r'}) \tag{3.5}$$

The packets in the buffer are transferred utilizing the entire bandwidth of the channel. The number of packets in the buffer b when the middle node just enters communication range depends on the time the node stays out of range, which is inversely proportional to the speed of the node. When the speed is such that

$$\frac{d}{s} - L \geq \frac{b}{r'} \tag{3.6}$$

the node stays in connectivity for suf£cient time to empty the contents of the buffer. When there are no more packets in the buffer, a steady connection is established between the source and the destination and packets are transferred at a rate r (this rate now includes

the packet generation rate, CBR). In this case, the time over which a steady connection is established is given by

$$T_{steady} = \frac{d}{s} - L - \tau \tag{3.7}$$

Thus the total number of packets transferred while the middle node is in connectivity is given by

$$n_{received} = (\frac{d}{s} - L - \tau)r + \tau r' \tag{3.8}$$

When the node is moving fast such that,

$$\frac{d}{s} - L < \frac{b}{r'} \tag{3.9}$$

the contents of the buffer cannot be completely emptied. Hence the packets are transferred only for a time τ. Number of packets transferred during this time is,

$$n'_{received} = \tau r' \tag{3.10}$$

In general, the number of packets transferred in a cycle is given by combining (8) and (10) as

$$N_{received} = max(0, \frac{d}{s} - L - \tau)r + \tau r' \tag{3.11}$$

where τ is given by equation (5). The total number of packets sent by the source during a cycle can be divided into the number of packets sent while the middle node was in range and the number of packets sent while the middle node was out of range. The number of packets sent in a complete cycle is thus given by

$$N_{sent} = (\frac{d}{s} + (\frac{d'}{s} + p))r \tag{3.12}$$

The throughput of the network is calculated from equations (11) and (12) as,

$$throughput = \frac{N_{received}}{N_{sent}} = \frac{max(0, \frac{d}{s} - L - \tau)r + \tau r'}{(\frac{(d'+d)}{s} + p)r} \tag{3.13}$$

Thus throughput is a function of speed of the middle node,link detection latency, rate at which packets are sent by the source, rate at which the packets are removed from the buffer,

size of the source buffer, in-range and out-of-range distances, and pause time of the node at the extreme points. In our simulations, we keep the value of r and r' constant. We evaluate the dependence of throughput on L in the subsequent sections. The phenomena of abrupt changes in the throughput as captured by these equations is also explained in the next section.

3.2.4 Explaining Synchronization Effects

3.2.4.1 Abrupt Rises in Throughput

As seen in Figure 3.3, the throughput curve shows abrupt rises at certain node movement speed changes. Referring to equation (13), we see that the only factor in¤uencing throughput as a function of speed is the link detection latency (L). L is itself a function of the speed of the node and its value varies between 0 and I. This can be veri£ed from Figures 3.4 and 3.5. The graphs of link detection latency show a saw-tooth behavior. The value of L increases gradually to nearly 10 seconds, before dropping sharply. To explain this, we refer to equations (2), (3) and (4). As s increases, L also increases gradually till it exceeds I. Then the value of k is decremented by 1, resulting in a sharp drop in L. In other words, as the speed of the node increases, it enters the region of connectivity a little earlier compared to the time it entered the region with a lower speed. Since the RREQ packets are sent at approximately the same time, the moving node is detected a little later by the routing protocol resulting in an increased value of L. Thus at some speed, the speed-pause time combination is such that the routing packets are sent just before the node is in range, resulting in a link detection latency of nearly 10 seconds. Subsequently, for slightly higher speeds, the RREQ packets are sent immediately after the node in the region of connectivity reducing L to nearly zero. Now referring back to equation (13), it can be seen that as L increases with speed, the throughput decreases gradually. When L drops suddenly to zero, the throughput rises sharply. The abrupt changes in the throughput are thus a result of the

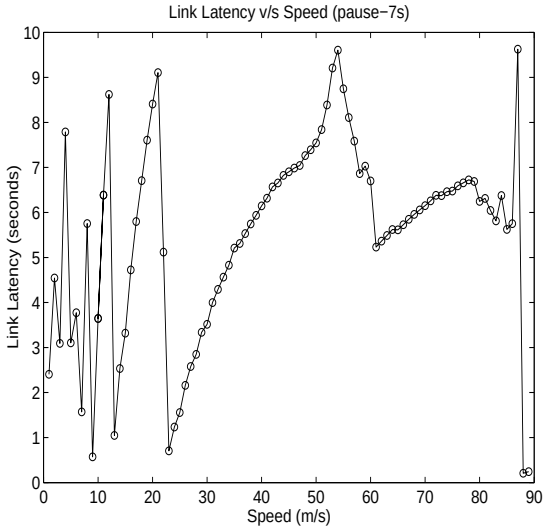

Figure 3.4: Link Detection Latency v/s Speed for pause time of 7 seconds. The graph shows a saw-tooth pattern

abrupt changes in the value of L, caused by a synchronization between the moving node and the RREQ packets. A small change in the speed can thus cause a signi£cant change in the throughput. Since the value of L is in¤uenced by the value of I, the maximum interval between two RREQ packets, it is important to have a right value of I to obtain an optimum throughput under all scenarios. In Section 3.2.6, we change the value of I and observe its effects on throughput and link detection latency.

3.2.4.2 The Region of Constant Throughput

The steady rise in the throughput continues until a speci£c speed of the middle node, beyond which it increases sharply to a very high value. Very few packets are thus dropped once the node starts to move faster than a particular speed in spite of the node staying out of range for some time (which contradicts the intuition, since high speeds should cause more

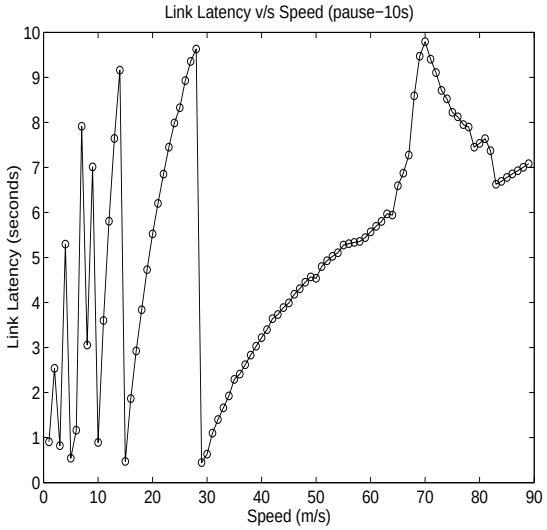

Figure 3.5: Link Detection Latency v/s Speed for pause time of 10 seconds. The graph shows a saw-tooth pattern

link breakages and hence a drop in the throughput). The reason for this can be explained as follows. As soon as the source node detects a breakage in the link, it starts buffering the data packets in its buffer at the network layer. Such a high throughput is attained when the middle node is moving at a suf£ciently high speed that it is able to come back into the range before the buffer starts to over¤ow. Once the node is detected and the link is formed, packets within the buffer are transferred at a much faster rate, utilizing the full bandwidth of the channel (2 Mbps), than they are produced by the application agent (approximately 7200 bps). As soon as the packets within the buffer have been transferred, a steady connection is formed between the source and the destination. If the speed of the node is constrained by equation (6), such that the contents of the buffer are emptied out before the node goes out of range, then the packets are re-£lled in the buffer when the node is out of range. The

amount of time the node spends out of range is given by,

$$\frac{d'}{s} + p + L$$

Since packets are transferred at a rate r during this time, if

$$(\frac{d'}{s} + p + L)r < B \tag{3.14}$$

then the node is back into the range before the buffer over¤ows. Thus the range of speeds constrained by equations (6) and (14) de£ne the region where the throughput is constant and has a very high value. The value of buffer B is crucial in maintaining a high throughput and its value should be carefully selected. We believe that the right value of the buffer size will be in¤uenced by the network topology and the kinds of applications supported by the network.

3.2.4.3 The Sharp Drops in the Throughput

Expression (14) indicates that a high throughput is achievable beyond a certain speed because the moving node comes back into the range before the buffer at the network layer of the source node, that holds packets from the application agent, over¤ows. Theoretically, the throughput should remain steady at near 100 percent beyond this speed, assuming that there is an in£nite bandwidth to transfer the contents of the buffer as soon as the connection is established. In an ideal world, the moving node should get detected as soon as it comes back into the region of connectivity. However, in the real world, the routing protocol has constraints. To keep the overhead low, the DSR agent broadcasts RREQ messages only once every I seconds beyond the £rst three attempts. This I second periodic timeout interval leads to an interesting phenomenon.

In our scenario, the RREQ packet broadcast may just miss the node before the node enters the range for some speed-pause time combination. The node then remains undetected for nearly I seconds before the next RREQ broadcast is made. If the speed of the node is

suf£ciently high that it again moves out of range before the next set of route request packets are broadcast, the node may never be detected at all.

For some speed-pause time combinations, the periodically moving node in our scenario gets synchronized to the route requests such that the node is not detected during the whole course of simulation. The throughput drops to nearly 0 percent, while the control overhead goes very high as seen in Figure 3.9. The source node keeps sending RREQ packets every I seconds for the whole duration of simulation. We call this *a point of degenerate synchronization*. Such points of degradation caused by synchronization and poor timing of control packet dissemination can seriously jeopardize the network performance and render the QoS assurances useless.

When the degenerate synchronization occurs, the node is effectively never in the range. Thus from expression (1) we have $d/s - L = 0$ and from equation (5), τ=0. Substituting these values in expression (13) results in a 0 throughput. Since the value of L is in¤uenced by I as seen earlier, we vary the value of I and observe its in¤uence on degenerate synchronizations in Section 3.2.6.

3.2.5 Synchronizations with Stochastic Speeds

The degenerate synchronization effects do not necessitate periodic movement of nodes. In this section, we show that the synchronization effect can occur in networks where the speeds of the nodes are stochastic and their movement non-periodic. We considered the same scenario with a single moving node. However, the node does not oscillate periodically anymore. Instead, after reaching the end point, it chooses a new speed from a uniform random interval. Such speeds can be seen, for example, in a set of cars moving on the highway where the speeds oscillate about the speed limit. The resulting graphs are shown in Figure 3.6. The three curves correspond to deviations of 5 m/s, 10m/s and 20m/s respectively about the mean which is indicated on the X-Axis. Even with a high deviation of

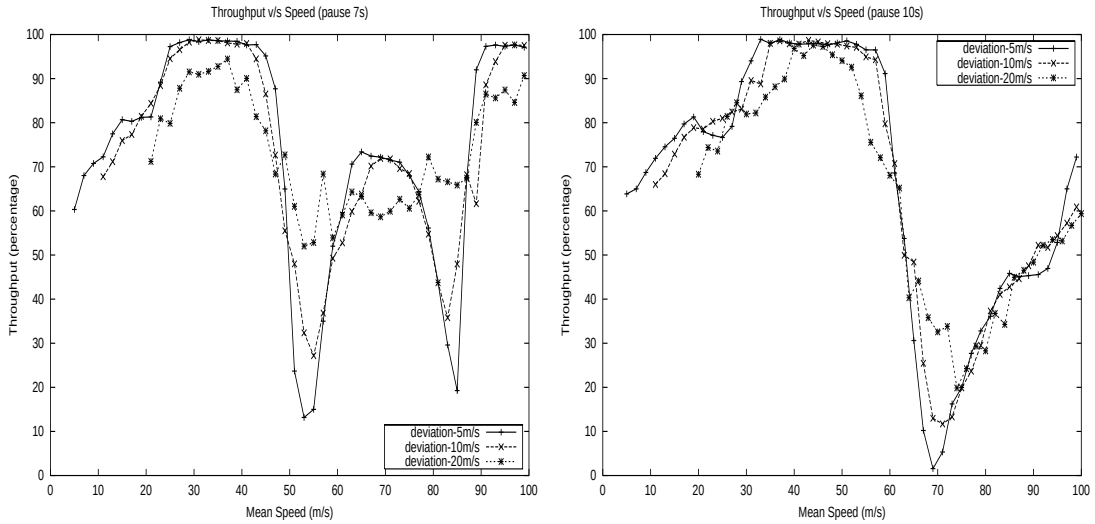

Figure 3.6: Throughput v/s Speed for pauses time of 7 seconds (left) and 10 seconds (right) for deviations of 5 m/s, 10 m/s and 20 m/s around the mean speed. The degenerate synchronizations can be clearly seen

20 m/s, the throughput is not able to recover by a large extent. In real networks such a high variance in the speed is unlikely. Figure 3.7 shows the change in the throughput with an increase in the variance of the node speed. The graph is a result of averaging 10 random runs. As we can see, the improvement in the throughput is linear beyond a certain speed. The gradient of improvement is low and even with large variances in the speed, the improvement in the throughput is marginal. In the next section, we propose some simple techniques to overcome these degenerate synchronizations.

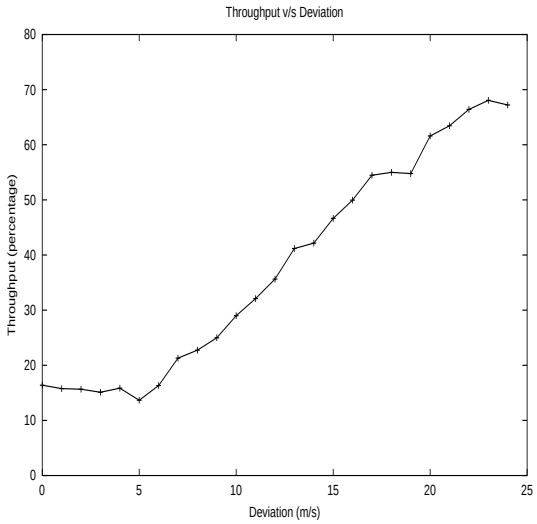

Figure 3.7: Throughput v/s Deviation around the degenerate speed of 70 m/s when the node does not pause at the ends. The improvement in the throughout is linear beyond the speed of 5 m/s.

3.2.6 Overcoming Degenerate Synchronizations

Degenerate synchronizations would not occur in an ideal world where the node would be detected as soon as it comes back into the range. This can happen if the source node sends route request packets at an in£nite rate. Since it is not possible, we identify two practical ways of overcoming the degenerate synchronizations.

3.2.6.1 Reducing the Timeout Interval

From equation (2), we note that the value of I can have a signi£cant effect on the throughput. In our £rst scheme, we reduce the value of I by half, so that the RREQ packets are sent at a maximum interval of 5 seconds instead of 10. Figure 3.8 shows the through-

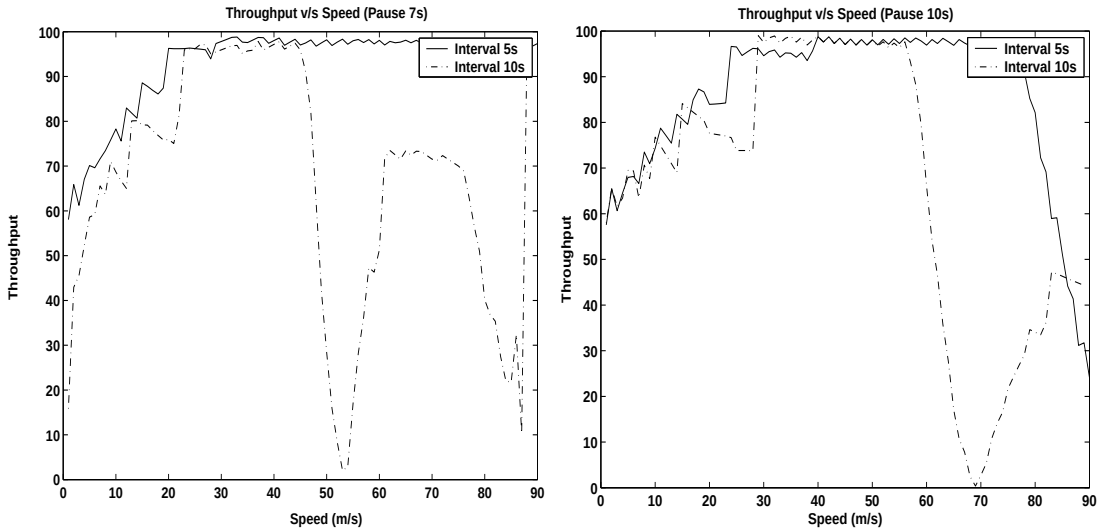

Figure 3.8: Throughput v/s Speed for pause times of 7 seconds (left) and 10 seconds (right), for two control packet intervals of 5 seconds and 10 seconds. The improvement in the throughput with double frequency is clearly seen.

put achieved with the reduced timeout interval. The throughput is maintained close to 100 percent for much higher speeds, suf£cient for all practical purposes. Obviously this improvement comes at a price. Figure 3.9 shows the graphs of control packet overhead with the 2 two timeout intervals. The consistently higher overhead with the reduced interval may be intolerable in a congested, bandwidth constrained network. The right timeout value should be chosen by taking into account factors such as mobility of the nodes, throughput requirements and amount of overhead that can be tolerated. Making the right trade-off could be a dif£cult decision.

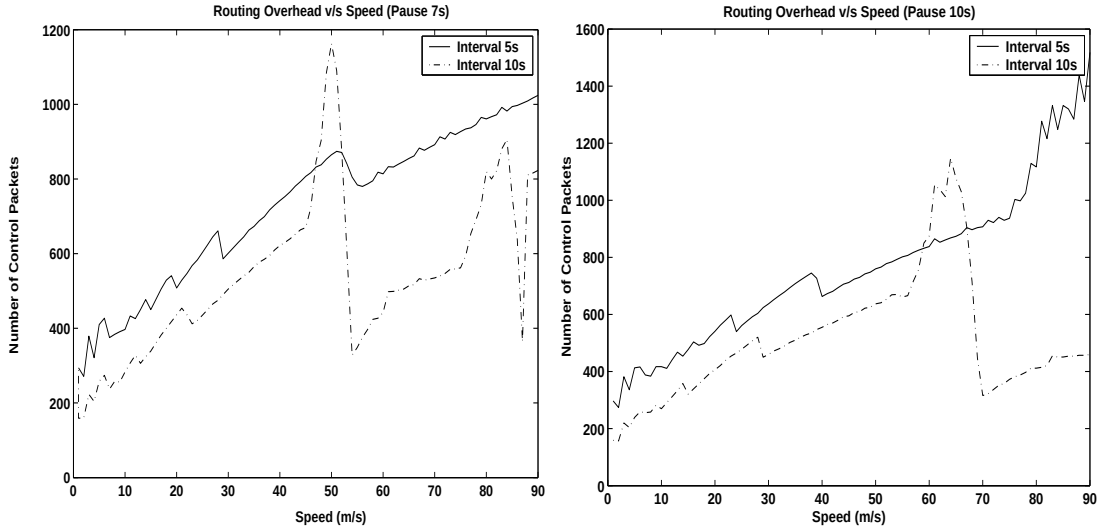

Figure 3.9: Control Packet Overhead v/s Speed for pause times of 7 seconds (left) and 10 seconds (right), for two control packet intervals of 5 seconds and 10 seconds. The consistently high overhead with double frequency may not be acceptable

3.2.6.2 Randomizing the Timeout Interval

We note that synchronizations are caused because of £xed and periodic nature of DSR RREQ packets. Proactive protocols like OLSR [9] refrain from sending periodic broadcasts of hello messages to avoid another kind of synchronization effect which has also been noted in [10]. In OLSR, or similar proactive protocols, periodic broadcasts of hello messages can lead to a synchronization where all nodes try to broadcast hello messages at the same time. This can cause collisions and since most of these messages are sent over UDP, these messages can be lost, leaving stale topology information with the nodes. OLSR overcomes this problem by adding a random jitter to the hello message interval. The periodic broadcasts of DSR packets in our case is causing a different kind of synchronization which can however be dealt with in a similar way. Therefore in our second scheme, we randomize the control

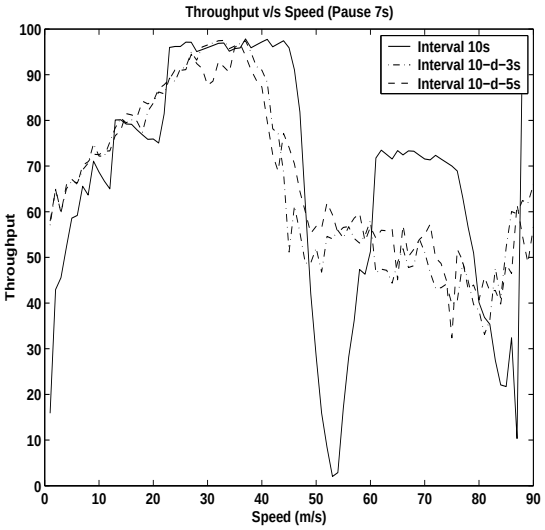

Figure 3.10: Throughput v/s Speed for pause times of 10 seconds, comparing throughputs with different control packet intervals. The uniform random time intervals lead to smoothening of throughput graphs

packet interval so that the time after which the next request packet is sent is chosen from a uniform random distribution. We experimented with 2 different intervals: 10 seconds with a deviation of 3 seconds (uniform distribution of 7 seconds to 13 seconds) and 10 seconds with a deviation of 5 (uniform distribution of 5 seconds to 15 seconds). The results are shown in Figure 3.10. The graphs certainly show a smoothening of the throughput, though now, the peak throughput is not sustained as consistently as for £xed control frequency. The Control Overhead is nearly the same as that with the £xed interval of 10 seconds. This scheme guarantees a minimum throughput under worst case scenarios, and can be used for critical applications where the failure of the network can lead to a catastrophe.

3.2.7 Discussion

In this chapter we have examined the in¤uence of timing and frequency of control packets and buffer size at the network layer on the throughput of the network. Small changes in speed of the nodes can cause large changes in the throughput due to synchronizations between movement of the nodes and control packets. The synchronizations can cause abrupt rises as well as drops in the throughput. The effect has a direct impact on the achievable QoS for mobile or sensor networks. The results produced here are especially important for time-critical applications in crucial scenarios, where overlooking of such effects can result in catastrophic failures.

We identi£ed points of degenerate synchronization where the throughput can unexpectedly drop to 0 percent and studied some ways to overcome these points. This synchronization effect has not been identi£ed previously. We observed that the effect is not limited to simple and deterministic movement patterns, but it is equally likely to occur in a network where the motion of the nodes is stochastic. Its appearance in random and more complex scenarios is either ignored or obscured by running multiple sets of simulations and averaging the results out. Although the effect is explained in the context of DSR protocol, it applies in general to any routing protocol, that exhibits a periodic pattern in disseminating the control information. The problem is studied in context of ad hoc networks but it is equally likely to occur in sensor networks which have the property of exchanging messages periodically. The solutions to the problem are built from known solutions in the wired world, which are associated with a similar, though not the same problem as discussed in [10]. In our ongoing work, we are evaluating some adaptive schemes where the timeout interval is calculated based on the past history and other factors such as the speed of the nodes and the average time to heal a link. Such adaptive schemes, we believe, will result in substantial improvement in the throughput of the network while keeping a low overhead.

In this work, we do not consider the effect of an alternate path when the original path

is lost. The availability of an alternate path may reduce the degenerations to some extent, though they might still affect the network performance. We plan to study this effect in our continuing work.

4. QoS Models

4.1 Introduction

RFC 2386 [11] characterizes QoS as a set of service requirements to be met by the network while transporting a packet stream from the source to the destination. Intrinsic to the notion of QoS is an agreement or a guarantee by the network to provide a set of measurable pre-speci£ed service attributes to the user in terms of end-to-end network delay, delay variance (jitter), available bandwidth, probability of packet loss and so on. RFC 1633 suggests that with the provisioning of QoS, the user must be able to get a service whose quality is suf£ciently predictable that the application can operate in an acceptable way over the duration of time determined by the user.

Before going into the detail of the available models, lets de£ne a set of metrics associated with QoS performance in networks that we use throughout the rest of the thesis.

1. **Bandwidth Bound :** The minimum rate at which packets can be transmitted from the source to the destination throughout the length of duration of the ¤ow.

2. **End-to-end Network Delay Bound:** The maximum amount of time a packet takes to reach from the source to the destination. This bound is essential for real-time applications such as voice conversations.

3. **Delay Variance Bound:** The maximum time difference between the arrival of two consecutive packets at the destination. This metric is crucial in de£ning the lengght of the buffer at the receiver for real-time applications.

4. **Packet Delivery Ratio :** The ratio of the number of packets reaching the destination to the number of packets sent by the source. This metric de£nes the threshold below which the performance of certain applications starts to degrade.

5. **TCP Goodput :** The number of bytes of a TCP connection reaching the destination per second (measured in kbps). This metric is an important indication of the amount of congestion in the network.

Several QoS schemes and models have been developed in general for the wired Internet and also for MANETs. A complete listing of all the schemes is not possible. We £rst brie¤y describe the two well known QoS models for the Internet, the Integrated Services Model and the Differentiated Services model and describe why these models are not directly applicable to MANETs. Then we classify the available QoS schemes and models for MANETs into three broad categories and brie¤y describe each category with examples.

4.2 QoS Architectures for the Internet : Interserv and Diffserv

4.2.1 Integrated Services

The Integrated Services (Intserv) model was the £rst standardized model for the Internet developed by IETF (RFC 1633 [12]). The model offers two kinds of services : Guaranteed QoS, that ensures the requested bandwidth and delay bounds for the duration of the connection and Controlled Load QoS, a better than best effort service for applications that can tolerate some amount of delay but are sensitive to congestion in the network. RFC 2212 provides speci£cation of guaranteed quality of service, while RFC 2211 de£nes speci£cations of the controlled-load network element service.

As a part of guaranteed QoS service, the model provides hard guarantees to the ¤ows by performing admission control and making reservations along the nodes prior to the commencement of the ¤ow. The framework includes four components: packet scheduler, admission control routine, classi£er and reservation setup protocol. The functionality of each of these is described brie¤y below:

1. **Packet Scheduler :** The packet scheduler manages the forwarding of different packet streams using a set of queues and other mechanisms such as timers. The packet

scheduler is implemented at the point where the packets are queued and controls the ¤ow of packets out of the queue. The exact details of the scheduler are implementation speci£c.

2. **Classi£er :** The purpose of the classi£er is to map each incoming packet into some class for the purpose of traf£c control. All packets in the same class get the same treatment from the packet scheduler. A class might correspond to a broad category of ¤ows, e.g., all video ¤ows or all ¤ows attributable to a particular organization. On the other hand, a class might hold only a single ¤ow. The same packet may be classi£ed differently by different routers along the path.

3. **Admission Control :**Admission control implements the decision algorithm that a router or host uses to determine whether a new ¤ow can be granted the requested QoS without impacting earlier guarantees. When a host requests a real-time service along some path through the Internet, the admission control is invoked at each node to make a local accept/reject decision. In addition to ensuring that QoS guarantees are met, admission control is concerned with enforcing administrative policies on resource reservations.

4. **Reservation Setup Protocol:** The reservation setup protocol is necessary to create and maintain ¤ow-speci£c state in the endpoint hosts and in routers along the path of a ¤ow. RSVP has been suggested as a reservation setup protocol for Intserv. RSVP is receiver initiated. A receiver sends a reservation request toward the source that speci£es the amount of resources to be reserved for all, or some subset of, the packets in a particular session. The resource quantity is speci£ed by the ¤owspec, while the packet subset to receive those resources is speci£ed by the £lter spec. As the message ¤ows along the routers, the admission controller at each router checks these speci£cations. If enough resources are available to accept the ¤ow, it makes a reservation block for the ¤ow and sends the message to the next hop. No resources are allo-

cated for the ¤ow yet. If the requested resources are not available, an error message is sent back to the initiator and the reservation request message is discarded. If the message successfully reaches the destination, the destination node responds with a ”RESV” message. This message ¤ows along the reverse path and make reservations for the ¤ow. As the RESV message ¤ows along the routers, resources are reserved and state information is stored for that ¤ow. When the RESV message reaches the initiator, resources have been reserved along the path. The sender node can now start its transmission. The state of the reservation at each router is soft and it need to be periodically refreshed, otherwise resources are deallocated on a timeout.

4.2.1.1 Applicability to MANETs

There are several issues that hinder the direct applicability of the Intserv framework to MANETs.

1. **Admission Control:** The admission control module of Intserv requires that precise information be available with each node about the amount of resources available and in use. However, as we saw in the previous chapter, obtaining precise information is dif£cult in MANETs due to a number of factors. The techniques suggested in literature provide an estimation of the available bandwidth. However, these techniques do not guarantee precise information at all times. With an approximate information, wrong admission control decisions can made. This makes the provisioning of Guaranteed QoS Service in MANETs extremely challenging.

2. **Short Lifetime of Reservations:** In a MANET, movement of nodes and changes in resource availability due to environmental conditions causes paths of the nodes to change over time. Hence, reservations made along one path would become unusable as soon as the ¤ow reroutes. These reservations would block resources till timeout occurs on the soft state. Further, new reservations are required along the new path.

Due to the delay associated with making the reservation, the ¤ow will be required to wait every time a rerouting occurs.

3. **Excessive Control Overhead:** The signaling protocol RSVP makes use of control messages such as “PATH”, “RESV” and “ACK” for each ¤ow. Frequent reservations arising due to rerouting would result in excessive control overhead.

4. **Node Complexity:** The implementation of components such as packet scheduler, admission controller, classi£er and signaling protocol at each node along with the requirement of maintaining per ¤ow state information may result in excessive processing overhead in size and battery power constrained MANET nodes.

Due to these limitations, we shift our focus on the Differentiated Services model that was designed to overcome the limitations of the Intserv model.

4.2.2 Differentiated Services

4.2.2.1 Overview

The Differentiated Services (Diffserv) model [13] was designed to overcome the inherent demerits of the Intserv model. Many RFCs have been developed to standardize various aspects of the model such as de£nition of per hop behavior identi£cation codes (RFC 3140 [14]) and behavior of the nodes to different classes of traf£c (RFC 2597 [15], RFC 3246 [16]). Diffserv is a *fully distributed* and *stateless model*. No state information is required to be maintained at any node. The model aims at pushing the complexity to the edge of the network so that the core can be as simple and fast as possible. Instead of providing QoS at per ¤ow granularity, Diffserv differentiates the traf£c into a £xed number of classes. The network is divided into *edge network* and *core network*. The nodes at the edge of the network are responsible for *classi£cation* of ¤ows, *policing* them to ensure that the traf£c complies the agreement made by the user with the service provider and *marking*

the packets so that they can be differentiated by ¤ows in the core of the network. Nodes at the core of the network provide *Per-Hop-Behavior* (PHB) depending upon the class of the packet, indicated by the *Diffserv code point* DSCP in the header of the packet. Nodes use RIO queue management and schedulers such as weighted round-robin to provide differentiation. The model currently supports two kinds of service: *Premium Service* that offers low jitter, low delay, assured bandwidth end-to-end service and *Assured Service* that provides a better than best-effort service and provides priority to packets in times of congestion.

4.2.2.2 Mechanism

Users sign *Service Level Agreements* (SLAs) with the service providers. The SLAs de£ne the kind of service desired by the user (the amount of bandwidth required, traf£c speci£cations such as average burst size). Flows initiated by the user undergo *admission control* process at the edge of the network. The admission control is performed at the ingress of the network by *bandwidth broker agents* that manage resources within an Autonomous System (AS). Once admitted, the £rst router along the path of the user traf£c acts as an edge router and performs policing, metering and marking as described above. All packets that agree to the SLA are marked with the corresponding DSCP and are transmitted into the core of the network. The non-complying packets are policed and degraded to receive best-effort service (or may even be dropped). Within the core, the nodes look up the DSCP in the header of the packet and provide treatment depending on the mapping of the DSCP with the PHB. The packets are queued into separate queues and scheduled for transmission depending upon the PHB.

4.2.2.3 Applicability to MANETs

Diffserv appears as a potential model for a MANET environment, because of its merits such as low per node complexity, low control overhead due of the absence of an external signalling mechanism and no per ¤ow reservation reservation requirement. However, the model as de£ned for the wired networks, cannot be directly applied to a MANET. There are several issues that need to be resolved, such as distinction between the edge and the core nodes. Intuitively, the source nodes play the role of edge routers and the relaying nodes act as core nodes. Then, each node must have the capability to act as an edge node and a core node, resulting in an increased complexity at each node. Also, the concept of a SLA does not exist in a MANET. Each node itself must be responsible for not overwhelming the network with traf£c. This leads to security concerns.

However, we believe that the merits of the model outweigh the drawbacks associated with it. Hence, in our further work, we explore the applicability of this model to a MANET in more detail.

4.3 Classi£cation of QoS Models for MANETs

The Intserv and Diffserv model are not directly suitable in the context of MANETs. This is because of the characteristics of MANETs such as mobility, broadcast nature of the medium and so on. Many QoS schemes for MANETs can be found in literature, some based on the modi£cation of the conventional models. These schemes can be broadly classi£ed into three categories :

1. QoS aware routing protocols

2. Link layer based schemes

3. Independent schemes

4.3.1 QoS Aware Routing Protocols

Conventional routing algorithms aim at £nding a route to the destination with minimum number of hops without considering factors such as congestion along the links and bandwidth availability along a path. Attempts have been made at devising routing algorithms that take QoS factors into account and £nd a path from the source to the destination that can satisfy the requirements. In [17] authors show that link interferences make the problem of £nding a feasible path in a multi-hop MANET environment an NP-complete problem. This is because, to make a reservation along a path, it is necessary to reserve resources in all nodes that are in the interfering range of any node along the path from the source to destination. Nevertheless, there are many QoS aware routing protocols in literature that claim to provide a complete solution to the problem of £nding a feasible path that satis£es the given requirements. Examples of such schemes are given below.

4.3.1.1 CEDAR (Core Extraction Distributed Ad Hoc Routing Algorithm)

CEDAR [18] aims at £nding a route through the network that satis£es the minimum bandwidth requirements of an application with high probability. A set of nodes is distributively and dynamically elected to form the core of the network by approximating a minimum dominating set of the ad hoc network using only local computation and local state. Each core node maintains the local topology of the nodes in its domain and also performs route computation on behalf of these nodes. QoS routing is achieved by propagating the bandwidth availability information of stable high bandwidth links to core nodes far away in the network, while information about dynamic links or low bandwidth links is kept local. Route computation £rst establishes a core path from the dominator of the source to the dominator of the destination. Using this directional information, CEDAR iteratively tries to £nd a partial route from the source to the domain of the furthest possible node in the core path that satis£es the requested bandwidth using only local information. Effectively,

the computed route is a shortest-widest-furthest path (maximum bandwidth path) using the core path as a guideline.

4.3.1.2 QoS for Ad Hoc On-demand Distance Vector Routing Protocol (AODV)

AODV is one of the most popular and mature routing protocols available to the MANET community, and is used extensively by us for our simulations. Recently, two of the authors of AODV suggested extensions to the model to add support for QoS in [19]. The following is an extract from the introduction section of their document.

Route discovery in AODV is on-demand and follows a route request/route reply query cycle. When a source needs a route to a destination, it broadcasts a *Route Request*(RREQ) control in search of a route. Nodes having a current route to the indicated destination respond by unicasting a *Route Reply* (RREP) to the source node. Extensions can be made to these messages to enables mobile nodes in an ad hoc network to specify, as part of a RREQ, Quality of Service requirements that a route to a destination must satisfy. In particular, the RREQ MAY include a QoS Object extension which speci£es bandwidth and/or delay parameters. Any QoS parameters that have to be measured and accumulated at each hop along the way can be stored along with the associated RREQ message.

Every RREQ QoS extension also carries with it a *session-ID* value which is used to identify the particular QoS ¤ow that will be established according to the parameters of the RREQ. The session-ID has to be stored along with the route table information associated with the particular ¤ow that might be created because of the extended RREQ. Every ¤ow is uniquely identi£ed by the triplet including the source IP address, the destination IP address, and the session-ID.

If, after establishment of such a route, any node along the path detects that the requested Quality of Service parameters can no longer be maintained, that node MUST transmit a ICMP *QOS_LOST* message back to the node which had originally requested the now

unmaintainable level of service. This typically requires additional information to be stored in each per-destination route table entry. The ICMP QOS_LOST message identi£es the broken ¤ow by including the session-ID value associated with the ¤ow.

For QoS parameters that are affected by cumulative measures at each intermediate node of a routing path, an extension is de£ned to enable a running total to be maintained for that measure as the RREQ is propagated. Each intermediate node that elects to forward a RREQ MUST adjust the accumulated value in the extension so that an accurate determination must be made. This approach is better than modifying the values in the QoS object directly, because it enables the original request to be authenticated whenever that is required.

An intermediate node that can satisfy a RREQ with QoS parameters speci£ed typically SHOULD always rebroadcast the RREQ, even if it has a route to the destination. This contrasts with the situation with unconstrained RREQ messages, because without any need for QoS an intermediate is allowed to answer with an RREP message if it has a route to the destination. Unfortunately, the intermediate node is not likely to have current enough information about whether the remaining nodes along the path to the destination can also satisfy the requested QoS. Effectively, according to this speci£cation, every QoS path to the destination will be ultimately approved by the destination itself along with every intermedate node along the path from the source to the destination.

When the destination issues the RREP message, it MUST also copy over the QoS extension into the RREP message. Each intermediate node forwarding the RREP message back to the originator of the RREQ message also MUST copy over the QoS extension.

4.3.2 Link Layer based Schemes

The *Distributed Coordination Function* (DCF) of 802.11a/b link layer protocols in their current form do not have support for QoS guarantees for real-time traf£c. These *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA) based algorithms are not

capable of providing differential treatment to ¤ows. They aim to distribute the available bandwidth equally and fairly among the competing nodes. Several schemes have been proposed which are a variant of the IEEE 802.11 based protocols. Many of these schemes propose modi£cation of the *inter-frame spacing intervals* (DIFS, SIFS) to differentiate service provided to ¤ows. Other schemes propose the use of slotted TDMA based protocols and suggest ways to reserve slots based on bandwidth requirements. Two of these schemes are presented below.

4.3.2.1 IEEE 802.11e

The IEEE 802.11 Wireless LAN standard is being widely used for a variety of applications. The 802.11b version provides data rates up to 11 Mbps at the wireless medium. The new 802.11 a version can achieve data rates of up to 54 Mbps at the wireless medium. However both these versions support only best-effort service. Before going into the detail of IEEE 802.11e, lets £rst look at why the older versions are not capable of providing adequate service differentiation.

The IEEE 802.11a/b speci£cation de£nes two different ways to con£gure a wireless network: *ad-hoc* and *infrastructure mode*. In infrastructure mode, an *Access Point* is needed to connect wireless stations to a distribution system, whereas in an ad-hoc mode, all wireless stations are distributed without access coordinator. Obviously, only the ad hoc mode is applicable to MANETs. Both the modes use DCF as the medium access mechanism. It uses CSMA/CA protocol. In this mode, if the medium is found idle for longer than a DIFS (Distributed Inter Frame Space), then the station can transmit a packet, otherwise a back-off process is initiated. More speci£cally, the station computes a random value called back-off time, in the range of 0 and *Contention Window* (CW) size. The timer is periodically decremented by one for every time slot the medium remains idle after the channel has been detected idle for a period greater than DIFS. As soon as the back-off timer expires,

the station can access the medium. If no acknowledgment is received, the station assumes that collision has occurred, and schedules a retransmission by reentering the back-off process with an increased CW. These mechanisms are de£ned to allow all contending nodes to access the medium in a fair manner. The DCF does not distinguish between packets of different ¤ows or classes.

The IEEE 802.11a/b standards also de£ne a *Point Coordination Function* (PCF) that has a limited support for QoS. The PCF allows stations to have priority access to the wireless medium, coordinated by a station called *Point Coordinator* (PC). The PCF has higher priority than the DCF because it may start transmissions after a shorter duration than DIFS; this time space is called PCF Inter-frame Space (PIFS). Time is divided into repeated periods called *superframes*. With PCF, a *Contention Free Period* (CFP) and a *Contention Period* (CP) alternate over time in which a CFP and the following CP form a superframe. During the CFP, the PCF is used, while the DCF is used during the CP. In the infrastructure mode, the PC is typically located with the AP. During the CFP, the PC polls each station asking for a frame. Thus there is no contention for the medium during this time. A station can only transmit when it is polled by the CP. The requirement of a PC makes the implementation of PCF in ad hoc networks dif£cult.

In keeping up with the growing demand for supporting QoS in wireless networks, the IEEE Working group is currently working on developing a new standard, called *IEEE 802.11e*. It introduces two new coordination functions called *Enhanced DCF* (EDCF) and *Hybrid CF* (HCF). The EDCF operates in CP only while the HCF operates in both CFP and CP. The goal of EDCF is to enhance the DCF access mechanism of IEEE 802.11 and to provide a distributed access approach that can support service differentiation. The proposed scheme provides capability for up to eight types of traf£c classes. It assigns a short CW to classes that should have higher priority to ensure that with a high probability, the high priority classes will be able to transmit before the low-priority ones. The CW_{min} parameter can be set differently for different priority classes, such that the higher priority

classes have a lower value of CW_{min}. For further differentiation, different IFS can be used according to traf£c classes. Instead of DIFS, and *Arbitration IFS* (AIFS) is used. The IFS for a given class should be a DIFS plus some (possibly zero) time slots. Classes with the smallest AIFS will have the highest priority. Per priority differentiation used by EDCF ensures better services to high priority traf£c while offering a minimum service for low priority traf£c.

4.3.2.2 Black Burst Contention Mechanism

The Black Burst mechanism [20] of channel access is a distributed MAC scheme that provides QoS real-time access to ad hoc CSMA wireless networks. The scheme differentiates traf£c into real time traf£c requiring QoS guarantees and data traf£c considered as best effort traf£c, which is insensitive to delay. With this scheme, real time nodes contend for channel after a short inter-frame spacing of t_{med} rather than t_{long} used by data nodes. To resolve which Real time node must transmit next, all contending real time nodes jam the channel with pulses of energy called *Black Bursts* (BBs). The length of the BB transmitted by a real-time node is an increasing function of the contention delay experienced by the node, measured from the instant when an attempt to access the channel has been scheduled until the node starts transmission of its BB. The node with the biggest BB wins contention for that round and transmits its real time packet successfully.

4.3.3 Independent Schemes

Instead of adding QoS to the routing or MAC layer, these schemes act as an independent layer, de£ning separate QoS mechanisms like the Intserv and the Diffserv models. These mechanisms lie between the routing and MAC layers, however, some of them may rely on feedback from either or both of these layers. These schemes are important for our study, because one of the goals of our QoS model is to make it independent of the other layers of

the TCP/IP model.

4.3.3.1 SWAN (Service Differentiation in Wireless Ad hoc Networks)

The SWAN model was developed by the Comet team at Columbia University [21]. The model differentiates traf£c into real time UDP traf£c and best effort TCP traf£c. It is a stateless and fully distributed model that provides *soft QoS assurances* to real-time traf£c. It uses *admission control* for real-time traf£c, rate control of TCP traf£c and *ECN congestion control mechanisms* to ensure that real-time packets meet QoS bounds. Each node comprises an admission controller that maintains information about the status of the outgoing link in terms of the available bandwidth and amount of congestion. It does this by promiscuously listening to all packet transmissions within its range. The admission controller located at the source node sends a *probe message* toward the destination when a new real-time ¤ow requires servicing. The probe message returns carrying the value of the bottleneck bandwidth along the path. If this value is greater than the requirements plus a threshold value, the ¤ow is admitted, otherwise it is rejected and marked as best-effort. All TCP ¤ows are considered as best-effort. The best-effort traf£c passes through a *rate-controller* that shapes the traf£c according to the rate based on the feedback from the MAC layer. The admitted real-time traf£c bypasses the rate controller and has a scheduling priority over best-effort traf£c. The admitted real-time ¤ows only have soft QoS assurances, so that some of the ¤ows may be dropped or downgraded to best effort if network traf£c conditions change due to rerouting of traf£c.

4.3.3.2 INSIGNIA

INSIGNIA [22] is an *in-band signaling* system that supports adaptive reservation-based services in ad hoc networks. Thus all the control information is carried within the header of the data packet itself, without the need of a separate control channel. The INSIGNIA

framework is also developed by the Comet group at Columbia University.

The signaling system supports a number of protocol commands that drive fast-reservation, fast restoration and end-to-end adaptation mechanism. These commands are carried in-band with the data and encoded using the IP option £eld in datagrams. This in-band information is $snooped$ as data packets traverse intermediate nodes/routers and used to maintain *soft-state reservations* in support ¤ows/micro¤ows. To establish reservation-based ¤ows between source-destination pairs, source nodes initiate fast reservations by setting the appropriate £elds in the INSIGNIA IP option £eld before forwarding packets. Reservation packets (i.e. data packet with the appropriate IP option set) traverse intermediate nodes, executing admission control modules, allocating resources and establishing soft-state reservation at all intermediate nodes between source-destination pairs. The reservations need to be periodically refreshed by the packets of the ¤ows. In the event of a change in the path resulting from movement of the nodes, the £rst packet along the new path makes fresh reservations along this path thereby performing a fast restoration. Reservations made along the old path are removed on a timeout. Flows in the network are expected to be adaptive to bandwidth availability. A ¤ow that was allocated a MAX amount of bandwidth initially could be downgraded to MIN amount or even to best-effort in the event of rerouting of a ¤ow or if network conditions change.

The source node continues to send packets with the reservation bit set until the destination node completes the reservation setup phase by informing the source node of the reservation status using a QoS reporting mechanism.

5. Performance Comparison and Analysis of Diffserv and SWAN

The previous chapter presented an overview of the existing QoS Schemes for MANETs. Our concerns about a stateful scheme based on the Integrated Services framework bias us toward a mostly stateless approach. We believe that stateless schemes are more suitable to an ad hoc environment because of their inherent merits like simplicity, scalability and ability to adapt to dynamic conditions. To proceed toward our goal of tight QoS guarantees, we need to understand the intricate details of the working mechanisms of the existing schemes in this category and their points of failure in providing hard guarantees. In this chapter, we compare and analyze the performance of the two stateless QoS frameworks SWAN and DiffServ. The comparison is performed both qualitatively and quantitatively. Quantitative analysis is performed by simulations using Network Simulator (NS-2) [7] under varied mobility conditions and scenarios. The results of simulations and general understanding of the working of the two models are used for their qualitative comparison. We isolate mechanisms of these models that falter in the face of the dynamics of ad hoc networks, leading to a failure in providing QoS guarantees.

SWAN uses admission control for real time traf£c and varies the rate of TCP traf£c based on feedback from MAC layer to maintain delay and bandwidth bounds for real time traf£c. DiffServ, on the other hand, relies on policing, marking and scheduling mechanisms and RED mechanism [23] for congestion control implemented at the queue at MAC layer for providing QoS guarantees to traf£c of different classes. The quantitative comparisons are performed using three routing algorithms: AODV [24] and DSR [25] which are distance vector based reactive algorithms and OLSR [9] which is a link-state based proactive routing algorithm. Throughput and delay graphs of ¤ows under the two schemes suggest that SWAN outperforms DiffServ in terms of throughput and end-to-end packet delays. Detailed analysis of this performance difference is performed in Sections 5.2 and 5.3. We also

observe that neither of the schemes is able to provide absolute guarantee bounds to the traf-£c. SWAN is able to maintain high throughput and low delay for the admitted UDP ¤ows, but treats all real-time traf£c equally. Some UDP ¤ows may be dropped during congestion. DiffServ, on the other hand, has the capability to support multiple classes of UDP traf£c but suffers from the drawback of using static parameters without taking into account the dynamics of the network. The understanding of these mechanisms assist us in the design of our QoS framework presented in subsequent chapters.

5.1 Simulation Environment

We used Network Simulator (NS-2) [7] with its wireless extensions developed at CMU for our analysis. To implement the Diffserv Framework in NS-2, we found that the framework as de£ned for the wired networks is not readily portable to the wireless paradigm. Peer-to-peer nodes in an ad hoc network, for example, cannot be statically classi£ed as edge and core nodes. Each node may assume either functionality depending upon its position in the network. Therefore, in my earlier work, we modi£ed the wired implementation of DiffServ in NS-2 to make it applicable to wireless networks. Each node incorporates the functionality of an edge node and a core node. Marking of the ¤ows is performed at the source node while policing and shaping is done at all nodes along the path. The nodes implement Random Early Detection with priority (RIO) based queues [23]. Packets belonging to different classes are queued separately. The length of the queues and packet dropping probabilities during congestion are dependent on the class they belong to. Packets of higher priority are scheduled for transmission before the packets of lower priority are. The parameters of the queue are con£gured statically. No feedback of any kind is used to change any parameters. SWAN code for NS-2 was downloaded from the SWAN web-page [26] and installed on another instance of NS-2.

The simulation environment is similar to the one described in the problem statement

to be used for the evaluation of the new model. It consists of 51 nodes moving in a 300 meters X 1500 meters area. The movement of the nodes is de£ned by the random-waypoint model. Nodes can move with a maximum speed of 20 m/s. The mobility of the network is changed across simulations by changing the average pause time of the nodes between two consecutive motions. Simulations are performed with average pause times of 300 seconds, 120 seconds, 60 seconds, 30 seconds and 0 seconds. Four nodes were randomly selected to act as TCP sources, generating best-effort traf£c and £ve nodes were selected to source UDP real-time traf£c. TCP sources model FTP and web traf£c and UDP sources model voice and video traf£c. The length of £le transfers generated by the TCP sources and the length of silent period between transfers is a Pareto distribution with a mean of 10 seconds. These scenario and traf£c £les are the same ones as used by SWAN group for their analysis [21].

For DiffServ, the traf£c is divided into two classes: real time UDP traf£c belonging to the higher priority class and TCP traf£c belonging to lower priority class. The queue lengths and parameters are adjusted to yield high throughput of UDP traf£c.

5.2 Comparisons and Analysis

Quality of service can be de£ned as the ability of a network element to provide some level of assurance for consistent network data delivery. In this section, we analyze and compare the level of assurance that SWAN and DiffServ can provide to a ¤ow demanding QoS. The analysis is done by measuring the average throughput available to UDP real-time ¤ows and the average end-to-end delay experienced by the packets of these ¤ows. We also measure the goodput of TCP ¤ows as an indication of the amount of bandwidth share available to the best effort traf£c. Graphs showing average throughput of a voice connection and the goodput available to a web connection are shown in Figure 5.1. For these graphs, DSR is used as a routing protocol. Focusing on the throughput graph,we

can see that SWAN clearly outperforms DiffServ. Similar results are observed for other real-time ¤ows and with other routing protocols (AODV and OLSR) hence not produced here. SWAN uses rate control of TCP and UDP traf£c to maintain manageable levels of congestion in the network. DiffServ, on the other hand, relies on static con£guration of the parameters of the RED queue and scheduler to provide service differentiation. This broadly explains the superior performance of SWAN over DiffServ. More details of this distinction are provided in the next section.

Although SWAN has a better throughput than Diffserv, the throughput is not only much lower than 100%, but also varies with mobility. This can be attributed to the dif£culty in capturing the dynamics of the environment in an ad hoc network. SWAN relies only on feedback from the MAC layer as a measure of congestion in the network. However, other factors such as the average speed of the nodes may be important in determining the effective resource availability [4]. Absence of reservations and source based admission control also causes some ¤ows to be dropped, affecting their throughput.

Figure 5.2 compares the average end-to-end delays experienced by packets of web and video ¤ows in SWAN and DiffServ. SWAN is able to maintain a delay of less than 200 ms which is essential for a voice conversation. Absence of rate control allows DiffServ to build up congestion in the network causing packets to wait in the queues for longer periods of time. SWAN mechanisms ensure that if the delays increase beyond a certain threshold, either the rate of TCP ¤ows is cut down or some of the UDP ¤ows are dropped, thereby maintaining low delays for the remaining ¤ows.

5.3 Explaining the Performance Differences

The Differentiated Services framework is suggested as an end-to-end QoS solution for wired networks. Users sign service level agreement (SLA) with the service provider. Providers engineer the traf£c of the network to meet the SLAs. Traf£c Engineering in such

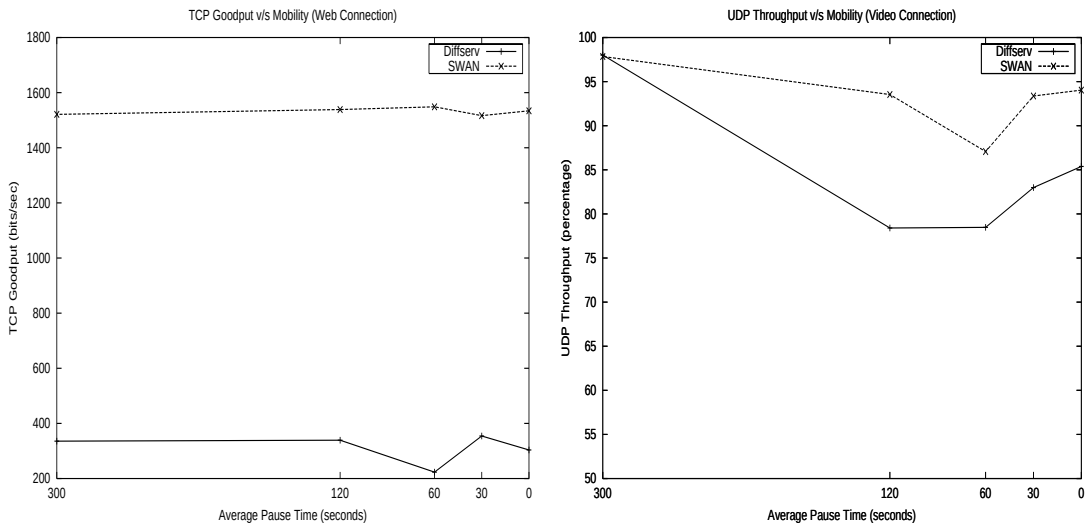

Figure 5.1: (left) Average goodput of a TCP ¤ow measured in kbps v/s mobility. Mobility is increased by decreasing the average pause time of the nodes. (right) Average throughput of a UDP ¤ow v/s mobility

networks is relatively easy due to their static nature. Bandwidth and other resources are unlikely to change by large amounts frequently. Admission control at the edge nodes also ensures that the capacity of the network is not exceeded to violate QoS of the admitted ¤ows. It may be dif£cult to de£ne SLAs in a dynamically formed peer-to-peer ad hoc network. Further, frequently changing bandwidth and traf£c conditions make traf£c engineering a challenging task. It is imperative to use some kind of feedback as a measure of the conditions of the network to dynamically regulate the traf£c of the network. Since the Differentiated Services architecture does not de£ne any scheme for taking corrective action on-the-¤y, a direct application of this model to a wireless network results in a deteriorated performance as seen in the previous section.

SWAN engineers traf£c by observing packet delay at the MAC layer and considering

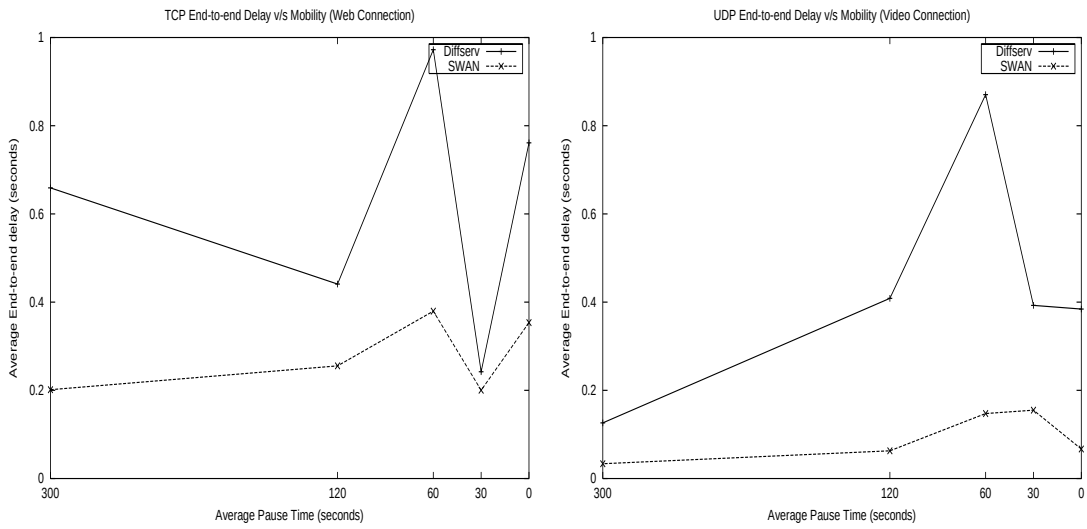

Figure 5.2: Average end-to-end delay experienced by packets of a TCP ¤ow (left) and a UDP ¤ow (right)

it as a measure of the amount of congestion in the network. It uses this to dynamically regulate the rate of both TCP and UDP traf£c. On detecting high levels of congestion, it allows some real-time ¤ows to be dropped or rerouted, thereby maintaining the QoS of the remaining ¤ows. The success of this technique is visible in the results presented in the previous section.

5.4 Traf£c Differentiation

Traf£c differentiation is an important aspect of any QoS scheme. The conventional Internet treats all traf£c as best-effort. However, modern applications such as Voice over IP (VoIP), video conferencing, streaming multimedia demand certain guarantee of resources for the entire duration of connection for their proper functioning. Further, some connections may be more important than others and may require to be treated with higher priority.

Traf£c differentiation is needed for the following two reasons:

a. **Resource requirements:** Traf£c belonging to different classes may have different bandwidth and delay requirements. For example, voice conversations need stringent delay requirements and a low steady bandwidth for the duration of the call. A live video telecast on the other hand, has high bandwidth requirements, though small amount of jitter can be tolerated by prebuffering. Also, in times of congestion, ¤ows belonging to such connections can be more readily dropped or rerouted.

b. **Priority:** Some ¤ows in a network may be more important than others. For example, a voice ¤ow from a Command and Control Center to a group of soldiers may be more important than a voice conversation between two soldiers. Hence in times of congestion, the latter ¤ow should be dropped prior to the former ¤ow.

SWAN differentiates traf£c into 2 classes, best effort and real time traf£c. It treats all UDP traf£c with equal priority and each one of them has an equi-probable chance of getting dropped, or rerouted during congestion. This does not comply with our goal of providing tight guarantees to traf£c of the highest class. DiffServ on the other hand, allows traf£c to be classi£ed according to the above criteria. Packets belonging to lower priority classes may be selectively dropped prior to dropping packets of higher priority classes. Also, a ¤ow consuming large amount of bandwidth could be dropped instead of dropping many ¤ows consuming low bandwidth. This allows ¤exibility in handling network dynamics. In [27], the authors analyze the ability of DiffServ model to provide traf£c differentiation in an ad hoc network. Diffserv enqueues packets belonging to different classes in separate queues and uses a scheduler to schedule the packets out of the queues for transmission according to their priorities.

5.5 Discussion

A summary of the architectural differences between the SWAN and Diffserv frameworks is presented in table 5.1. We can conclude that neither SWAN nor Diffserv presents a complete solution to our problem of tight QoS guarantees. Since one of our important requirements is differentiation of traf£c into multiple classes, Diffserv de£nitely forms the basic block of our new model. However, because of its inability to handle network dynamics, we need to build additional mechanisms which can ensure that the delay and bandwidth bounds of the highest priority traf£c are not violated. SWAN's mechanisms are effective in providing relative differentiation between Real Time traf£c and TCP traf£c. However, they can only provide soft assurances to the admitted traf£c. Further, since SWAN's mechanisms rely of feedback from the MAC layer, the implementation of the framework is dependent upon the link layer protocol. Thus, for our QoS framework presented in the subsequent sections, we do not consider SWAN as a part of our mechanisms.

Table 5.1: Summary of Architectural Differences between Diffserv and SWAN

SWAN	**Diffserv**
Supports two classes of traf£c : real-time and best-effort	Supports up to 16 classes of traf£c (4 bits of DSCP)
Dynamic rate control of TCP based on delay feedback from MAC	Rate control of TCP based on congestion (queue size)
Soft guarantees, admitted ¤ows may be dropped or degraded to best-effort	Flows are not dropped or degraded to best-effort
Scheduler required only for TCP best-effort traf£c	Complex scheduler required to schedule multiple classes of traf£c
Admission Control for real-time traf£c	Currently no support for admission control
ECN employed for rate control of real-time traf£c	Rate control of real-time traf£c done by metering and policing mechanisms

6. The Proposed QoS Model

6.1 Design Considerations

The study of the existing QoS models and comparison of Diffserv and SWAN performed in the previous chapter provide an excellent background for the design of our model. Before we proceed to the design of the model, we need to evaluate our needs and considerations.

The £rst and foremost design consideration is to make the model independent of the network and link layers. The new model must be capable of operating with any combination of these layers. As such, the new QoS framework acts as a new independent layer between the existing link and network layers. Further, we desire that the model be capable of differentiating traf£c into a number of traf£c classes suf£cient for practical purposes. For this, we use a Diffserv based queue consisting of a classi£er, policer, meter, marker, a separate physical queue for each class of traf£c and a scheduler to schedule the packets out of the queues. Since our early evaluations show that SWAN is effective in the rate control of TCP traf£c and provides soft QoS guarantees to real time traf£c, we may optionally use SWAN for traf£c whose priorities lie between that of high priority traf£c and best-effort traf£c. The SWAN mechanisms include an admission controller, a mechanism to estimate the amount of bandwidth at the link layer and a rate controller that controls the rate of TCP traf£c and best-effort UDP traf£c to maintain delay bounds of real time traf£c. The bandwidth estimation mechanism and rate controller of SWAN require a modi£cation of the link layer, which de£es our £rst consideration. Hence, in our initial design, we do not consider the inclusion of SWAN in our model.

Neither Diffserv nor SWAN are able to satisfy one of our most important requirements : the provisioning of tight QoS guarantees to ¤ows of highest priority class. Therefore, we

need additional mechanisms to monitor the activity of the high priority ¤ows and for taking actions to maintain tight bounds on the latency and bandwidth of these ¤ows. Figure 6.1 shows the location and basic components of the new model.

6.2 Tight Guarantees

We initially focus on providing tight guarantees to ¤ows of highest priority class. The ¤ows of this class have clearly de£ned traf£c characteristics such as packet rate and packet size whose values are de£ned in the problem statement. Our protocol must ensure that each of the high priority ¤ows is able to maintain these characteristics on its way from the source to the destination. All ¤ows must therefore be **monitored** at each node through which they pass. Any swaying from the designated packet rate, for example, must trigger a **corrective action** by the node detecting it. We need to de£ne elaborate mechanisms to monitor the activity of each high priority ¤ow. We also require mechanisms that de£ne how a corrective action needs to be taken to tackle any interfering ¤ows affecting the characteristics of a high priority ¤ow. We deal with each of these **monitoring** and **corrective** mechanisms separately. Before delving into the details of the mechanisms, we de£ne the following terms:

1. **Reception Range:** This is the maximum range within which two nodes can communicate with each other directly in one hop. Packets transmitted by one node can be received and processed by the second node. In our simulations, this range is de£ned to be 250 meters.

2. **Interference Range:** This is the range within which transmissions from one node can interfere with transmissions/receptions at another node to cause collisions. This range is usually greater than the reception range. Thus, when two nodes are outside the reception range of each other but within the interference range, packets transmitted by one node 'cannot' be received by the second node, but such packet transmis-

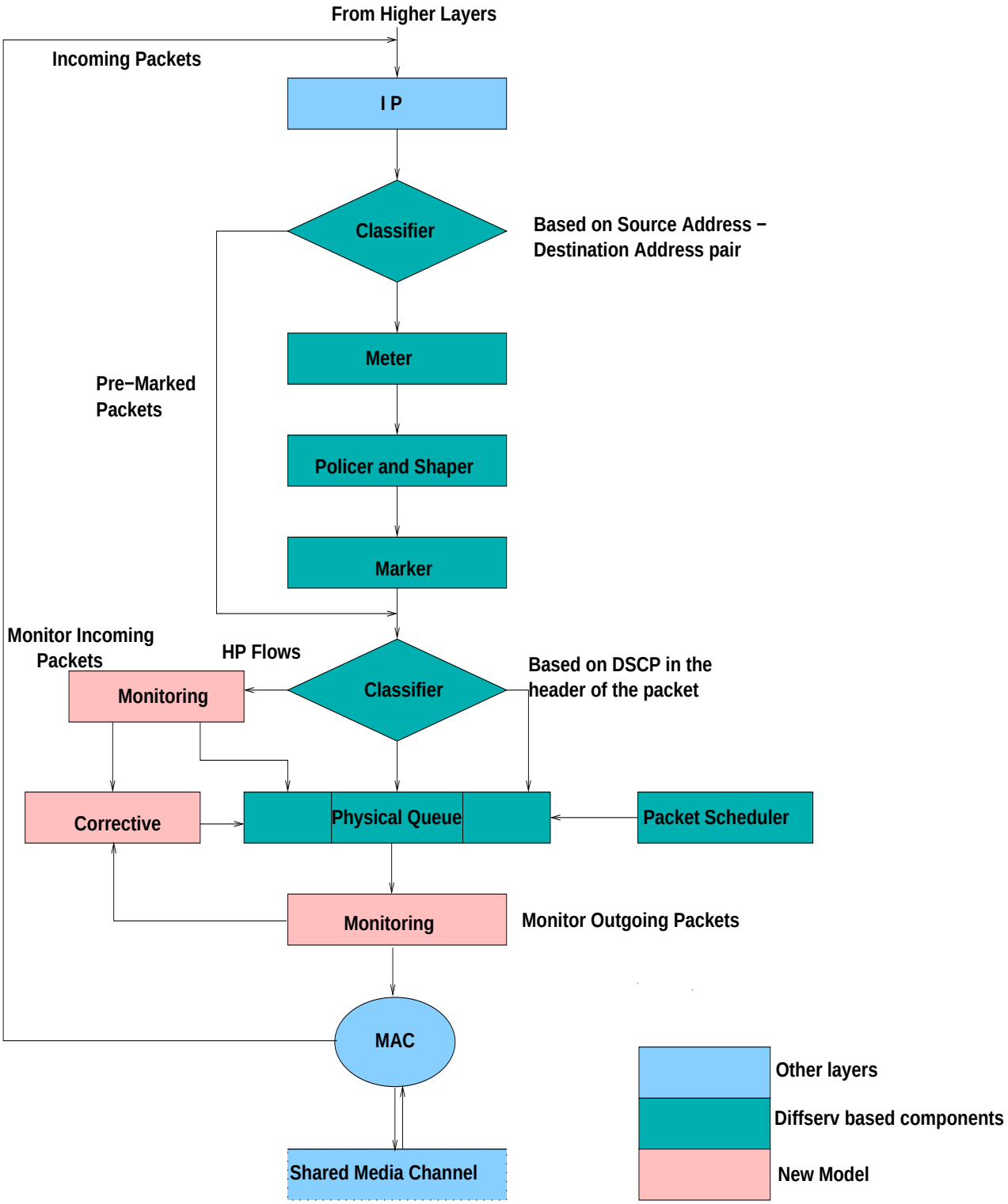

Figure 6.1: Block Diagram of the New Model within a MANET node

sions can interfere and corrupt the transmissions/receptions at the second node. For our simulations, the interference range is de£ned to be 500 meters.

3. **H-Node :** This is the £rst node along the path of highest priority ¤ow under consideration which can receive packets of this ¤ow from the previous hop at the desired rate but cannot transmit at the same rate because of interference from transmissions of other nodes in the network.

4. **H-Flow :** The highest priority ¤ow carried by the H-Node

5. **I-Node :** Node carrying Medium or Low Priority traf£c that potentially interferes with the high priority traf£c carried by H-Node thereby reducing the resource availability to that ¤ow.

We now study the details of the mechanisms required by our protocol as discussed above.

6.3 Monitoring Mechanisms

Since each node is required to monitor the activity of each high priority ¤ow carried by it, we put a bound on the number of high priority ¤ows through each node in order to limit the amount of per ¤ow state information within a node. In doing so, it is ensured that the network has suf£cient resources to support each of these ¤ows at their desired rate. The number of packets of a high priority ¤ow are generated and transmitted by the source at a £xed rate of r packets/sec. Each node carrying a high priority ¤ow monitors the number of packets received and transmitted for each ¤ow within a window of w seconds. For normal operation, the number of packets received at any H-Node within any window must be greater than the threshold r_{th} and the number of packets transmitted within the same window must be greater than t_{th}. Here, r_{th} is the receiving threshold and t_{th} is the transmitting threshold.

Obviously, $r_{th} \leq r \cdot w$, $t_{th} \leq r \cdot w$ and $t_{th} \leq r_{th}$

If the above condition is not satis£ed, it signals the presence of interference affecting either the transmissions by the current node or the receptions by the next node along the path. The conclusion about the next node being affected by interference comes from the observation that although the current node is able to gain hold of the medium for transmission, the next hop is not able to receive those packet because of interference. The presence of interference must trigger immediate corrective action to tackle the ¤ows causing it.

For our simulations, we restrict the number of H-Flows to 2 per node, the packet rate r to 32 packets/second, the monitoring window w is initialized to 2 seconds, the receiving threshold r_{th} is equal to the transmitting threshold t_{th}, both set to 55. At a rate of 32 packets per seconds, the maximum number of packets that can be received and transmitted within any window of size 2 seconds is 64. A threshold of 55 packets, therefore, corresponds approximately to a tolerance rate of 10%. Thus, a node signals the presence of interference, if within a window of 2 seconds, it receives at-least 55 packets but is not able to transmit more than 55 packets. A lower tolerance rate can be set by having the threshold value closer to 64. Although this would lead to a quicker triggering of corrective mechanisms, it might cause false triggering in case of small spikes in the congestion level, resulting in lower network utilization

6.4 Corrective Mechanisms

Once the monitoring mechanism signals the presence of interference obstructing the ¤ow of highest priority traf£c, the Corrective Mechanisms have the task of suppressing the interference to the level that the highest priority traf£c is able to reattain its traf£c ¤ow rate of r packets/sec. In doing so they must encounter three potential sources of interference.

1. **Direct Range Interference Nodes:** These nodes are within the reception range of the node carrying high priority traf£c. Some of these nodes may be carrying medium

priority or low priority traf£c which reduces the bandwidth available to the high priority ¤ow. Being in the direct reception range, these nodes can be informed relatively easily of the interference by broadcasting a message and corrective action can be taken quickly. These nodes are subsequently called as DRI Nodes.

2. **Nodes Outside Direct Transmission Range but within Interference Range:** These nodes are within the interference range of the H-Node, but not in reception range (refer to Figure 3.1). Thus, transmissions from these nodes interfere with the transmissions and receptions of H-Flow. In a random ad hoc network topology with random connections, it is likely that the prevention of interference from the DRI nodes is not suf£cient to restore the rate of the H-Flow. It may be required to stop any interfering ¤ows within these nodes. It is not straightforward to inform these nodes of the interference. Some of these nodes may be multiple hops away from the nodes carrying H-¤ow. Broadcasting a message to nodes two hops away may reach some of these interfering nodes, not necessarily all. However, as the control packet travels in an expanding ring, a 2 hop control packet may cause too many ¤ows to stop, resulting in an exorbitant underutilization of the network resources. In our de£nition of corrective mechanisms, we assume that the network is suf£ciently dense so that stopping interference caused by nodes in direct range is suf£cient to restore the resource availability of the H-¤ow. However, we do not ignore this case completely, but leave this as a subject of our future studies.

3. **Nodes Interfering with the Next Hop :** Some of the nodes may lie within the interference range of the next hop. This implies that the next hop neighbor is not able to receive packets even though the transmitter is able to send them at the desired rate. In this case, the monitoring mechanisms of the next hop will not detect any interference, since the node does not receive at the desired rate. The interfering nodes may be several hops away from the node detecting interference. In this case, even

broadcasting messages 2 hops away may not help. For our mechanisms, we ignore this case and assume that the case is too rare to be considered to add complexity to the protocol. We plan to explore this case in our future work.

As mentioned earlier, our corrective mechanisms only try to suppress interference caused by the nodes within direct range of the H-nodes. As we will see in the next chapter, our simulations demonstrate that these corrective mechanisms are suf£cient to provide the desired delay bounds on the H-¤ows while maintaining a high network utilization.

6.4.1 The Mechanism

When the monitoring mechanism signals the presence of interference, the H-Node broadcasts a *Squelch packet* with a TTL (time to live) of 1. All DRI nodes upon the reception of this packet stop the transmission of packets from any medium or low priority ¤ows that they carry either as sources or relays. All packets that are in the queue of these nodes awaiting transmission, are dropped. In doing so, the transport layer at the source of any TCP ¤ows carried by these nodes would cut down the rate of their ¤ows, suspecting congestion in the network. However, the rate of these ¤ows may build up soon, leading to interference. Also, the CBR sources would continue to generate and transmit packets that get relayed up-to these DRI nodes, where they get dropped. The corresponding source nodes of both the UDP and the TCP ¤ows must therefore be informed of this link break, so that they can stop their transmissions. The DRI nodes explicitly send *Squelch* messages to all sources whose packets they carry for relaying. Note that the DRI nodes keep dropping any medium of or low priority packets that they receive for the next 5 seconds after they receive the $Squelch$ packet, to ensure that by the end of that period, the source nodes have received the $Squelch$ packets and taken the necessary corrective action. On receiving this message, the source nodes stop generating packets for a random time interval in the range of 0 to t_{sstop} seconds. Upon the completion of this interval, the source nodes re-initiate

their ¤ows, presenting their requests to the routing protocol. The sources stop for a duration of t_{sstop} seconds because we assume that a high priority ¤ow lasts for an average of t_{sstop} seconds. For our simulations, we set this value to 100 seconds. However, this value can be changed based on the statistics of the duration of the high priority ¤ow.

This method ensures a fast restoration of resources to the high priority ¤ow. The restoration time is bounded by:

$T_{restore} = T_{detect} + T_{transmit\_control\_packet} + T_{process\_control\_packet}$

where,

T_{detect} : Time it takes for the H-Node to detect interference

$T_{transmit\_control\_packet}$: Time it takes for the H-Node to broadcast the Squelch packet to all the DRI neighbors. This includes the time it takes for the H-Node to create and transmit the packet to the MAC layer, time the packet spends in the queue before being taken by the MAC layer for transmission and time taken by the MAC layer to take control of the medium and broadcast the packet.

$T_{process\_control\_packet}$: Time taken by the DRI node to process the squelch packet and take corrective action.

T_{detect} depends on the time window used by the monitoring mechanism, of the order of a few seconds. $T_{transmit\_control\_packet}$ and $T_{process\_control\_packet}$ are negligible in comparison with T_{detect} and can be ignored.

Note that the $Squelch$ packets are sent over UDP, hence they might get lost in the network before reaching the source nodes. This may cause less than the required number of source nodes to take corrective action.

Although this mechanism results in a quick restoration of the ¤ow rate, it may preempt more ¤ows than necessary for the restoration. This may cause a reduction in the network utilization. For example, if the H-Node has 5 DRI neighbors, 4 of which carry voice traf£c at a rate of 20kbps and 1 carrying video traf£c at 64kbps, it may be suf£cient to stop the 1 ¤ow carrying video traf£c. However, the broadcast of the squelch packet to all 1-hop

neighbors causes all of them to stop. Further, it may be possible to reroute the stopped ¤ows around areas of congestion, without having to stop the ¤ows at all.

The achievement of tight QoS guarantees at the expense of an extremely underutilized network does not comply with our goals. Thus, we de£ne schemes to improve overall network utilization. The parameters used by the schemes can be tuned to achieve a trade-off between quality of connections of high priority ¤ows and network utilization. These schemes are described below.

6.4.2 Optimization Schemes

6.4.2.1 P-Broadcast

This technique aims at improving network utilization by probabilistically selecting the nodes that must take corrective action. A simple way to do this is to have a £x value added to the $Squelch$ packet, called the $p - value$, whose value lies between 0 and 1. The I-nodes, upon receiving the $Squelch$ packet, compute a random number between 0 and 1. If the random number is greater than the p-value, the nodes take corrective action, otherwise, the $Squelch$ packet is discarded. Thus, if the p-value is set to 0.5, then only about 50% of the nodes will take corrective action. A p-value of 0 corresponds to the basic corrective mechanism.

An improvement of this technique would calculate the p-value based of the difference between the number of packets transmitted within the monitoring window and the maximum number of packets that could be transmitted within the same window. A large negative difference between the two values indicates the presence of high interference and thus a low p-value and vice-versa. This results is a larger number of nodes taking corrective action. The p-value is directly proportional to the difference or the Error.

6.4.2.2 Selective Reject

Instead of broadcasting a control packet to all 1-hop neighbors, the H-Node enters promiscuous mode upon receiving a signal from the monitoring mechanism. In this mode, the signal taps all packets that are sent by the 1-hop neighbors and selectively asks a few nodes to stop sending. The victim nodes (nodes that have been asked to stop sending) may either be chosen randomly or based on their packet transmission rate. The restoration time with this method is bounded by:

$T_{restore} = T_{detect} + T_{promiscuous} + T_{transmit\_control\_packet} + T_{process\_control\_packet}$

where T_{detect}, $T_{transmit\_control\_packet}$ and $T_{process\_control\_packet}$ have the same meaning as de-£ned for the basic broadcast mechanism. $T_{promiscuous}$ is the amount of time the H-node stays in promiscuous mode to detect the victim neighbors. Note that we can reduce $T_{promiscuous}$ by having the H-nodes be in the promiscuous mode at all time while they are carrying a H-Flow. However, this results in an increased processing at the node and hence an increase in the battery power consumption.

This scheme increases the network utilization by selectively asking the nodes to stop. However, the improvement in the utilization comes at the cost of increased time the node is required to spend in promiscuous mode to detect the victim neighbors. The accuracy of determining the victim nodes may depend upon the amount of time the node spends listening to the neighbor's transmissions. For example, if the victim nodes are chosen randomly based on the packets overheard from the ongoing transmissions in the past t seconds, the interference may not reduce substantially to bolster the rate of H-Flow to the required level. Or, it may result in cutting down more ¤ows than necessary for the restoration. However, if the victim nodes are chosen based on their packet transmission rates, then the H-Node needs to stay in the promiscuous mode long enough to accurately determine the transmission rates of the neighboring nodes. This increased latency comes in addition to the increased complexity and power consumption of promiscuous mode mechanisms.

6.4.2.3 Rerouting

Instead of stopping the lower priority ¤ows completely, it may be possible to reroute these ¤ows around the area of congestion. Then the overall network utilization will be less affected by the mechanisms that take corrective action. To do that, when the source nodes receive *Squelch* packets from the DRI nodes, they simulate link breaks at the routing layer. The routing layer then tries to look for an alternate path. The reactive protocols broadcast route request packets in the network, which are replied to by either an intermediate node that knows the path to the destination, or the destination node itself. Since the new routes must bypass the areas of congestion, where the corrective action was taken by the H-Node, it is important that the corresponding DRI nodes do not reply to or relay any of these route request packets. Hence, when the DRI nodes receive *Squelch* packets, they compute a random interval in the range of 0 and t_{stop} as computed by the source nodes upon receiving the *Squelch* packets. Within this interval, these nodes do not forward any Route Request packets that are broadcast by the source nodes. This ensures that the source nodes £nd alternate paths if any. Although this scheme seems to be potentially good, there are two concerns in the applicability of this approach :

1. The source nodes may be able to £nd an alternate route to the destinations, however, many of these new routes may start interfering with the H-Flow at different points of its path. Because of this, other H-Nodes may need to take corrective action. The H-¤ow may lose its packet rate more frequently than if using the earlier techniques.

2. In the case of mobile scenarios, the H-Flow may need to reroute. Since the DRI nodes do not forward any control packets within the randomly computed interval, they may drop control packets generated by the nodes for rerouting that ¤ow. If one of the DRI nodes happen to follow on the new route, the H-Flow may not £nd a new route at all thus degrading its performance.

6.5 Implementation

We implemented a part of the model in Network Simulator (NS-2). As mentioned earlier, our initial focus was on providing tight guarantees to ¤ows of highest priority class. We implemented the monitoring mechanisms described in Section 6.3. For the corrective action, we implemented the basic mechanism described in Section 6.4. We also implemented the P-Broadcast technique with the £xed p-values and the Rerouting technique for improving the network utilization. The detailed simulation results and analysis are presented in the next chapter.

7. Simulations, Results and Analysis

In this chapter, we present simulation results to validate our new model. We performed simulations with a large variety of mobility and traf£c scenarios. The simulation results show that the model is able to achieve favorable bandwidth and delay bounds. The results compare the performance of the new model with varying parameters. The performance of the new model is also compared with that of SWAN, Diffserv and MANET stack without any QoS framework.

7.1 Simulation Scenario

7.1.1 Mobility Model

The scenario consists of 50 nodes placed randomly in a 1500 m X 300 rectangular area as shown in Figure 7.1. We consider two kinds of scenarios : static and mobile. The mobile scenarios are based on the random-waypoint model. The nodes move to the end of the simulation area and are re¤ected back with the angle of incidence (similar to re¤ection of light from a mirror). The nodes pause for an average of 10 seconds between movements. Once they pause, they randomly choose a new direction, speed and distance they want to move. Most of the results shown are produced by considering an average of 5 random scenarios.

7.1.2 QoS Model

The model as described in the previous chapter was implemented in NS-2. For improving the overall network utilization, we also implemented the P-Broadcast scheme with £xed p-values and Rerouting Scheme mentioned in Section 6.4. The simulations evaluate the performance of the high priority ¤ow and the overall network utilization with the basic

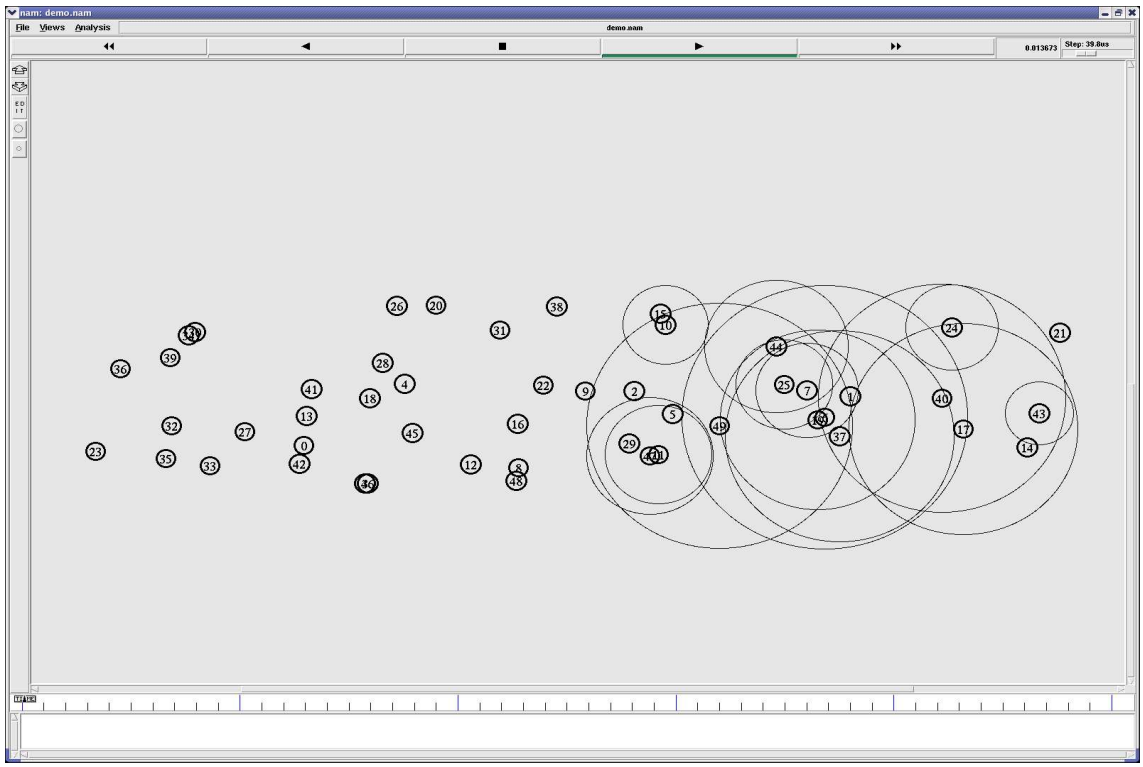

Figure 7.1: Snapshot of a static scenario as seen in Network Animator (NAM)

model and with the optimization techniques. Within the graphs, the suf£x $p0$ corresponds to the basic model, $p0.5$ corresponds to the model with P-Broadcast with £xed p-value of 0.5 and p0.7 corresponds to the model with P-Broadcast and a £xed p-value of 0.7. We also compare the performance of our model with that of SWAN and Diffserv. Although the Rerouting technique is implemented, the results with that technique are not so favorable and hence not shown here. The issues with this technique are still under consideration.

7.1.3 Traf£c Model and Traf£c Differentiation

Although the model is capable of differentiating the traf£c into an arbitrary number of traf£c classes, for our initial evaluation of the new model, we differentiate the traf£c into two classes: High priority class and Low priority class (or best-effort). The high priority class contains one or at-most two ¤ows, whose speci£cations match those described in the problem statement. Essentially, they are constant-bit-rate (CBR) ¤ows generating 80 byte packets at a rate of 32 packets/sec (20kbps). The ¤ows of lower priority class are CBR ¤ows generating 800 byte packets at a rate of 20 packets/sec (128kbps). The number of sources used as low priority traf£c are indicated before the results on an as-needed basis.

7.1.4 Miscellaneous Considerations

We considered AODV as the routing protocol, since it is one of the most mature routing protocols available to the MANET community and it is the £rst protocol to be converted from an Internet Draft to an RFC. MAC 802.11 b is implemented at the MAC layer, which offers a maximum data rate of 11 Mbps.

7.1.5 Parameters Monitored

We measure the running rate of the High Priority ¤ow v/s time and compare it with other models. We also measure the percentage of packets that successfully reach the destination with an end-to-end latency of less than 100 ms and 200 ms. We compare the overall network throughput with different models and with different versions of our new model. Finally, we consider the overall control overhead with the various models and the total overhead of our new model.

7.2 Results and Analysis

7.2.1 Scenario 1

Topology : Static, Random
Number of scenarios considered : 5
Number of High Priority Flows : 1
Type of Low Priority Flows : CBR
Number of Low Priority Flows : Variable from 9 to 13. The minimum number of ¤ows considered is 9, because the saturation in the network starts at around 9 ¤ows.
Rate of Low Priority Flows : 20 packets/sec, 800 byte packets (128 kbps)

Figures 7.2 and 7.3 show the variations of the packet rate of the high priority ¤ow with time in a typical static random scenario. Clearly, the rate of the ¤ow with our new model is much more consistent as compared to that with SWAN and Diffserv. The degree of variations can be easily controlled by adjusting the triggering threshold of the monitoring mechanisms.

Figure 7.4 shows the percentage of packets successfully received at the destination within delay bounds of 100 ms and 200 ms. Our new model outperforms SWAN by more than 15% for low loads and by a much larger margin at higher loads.

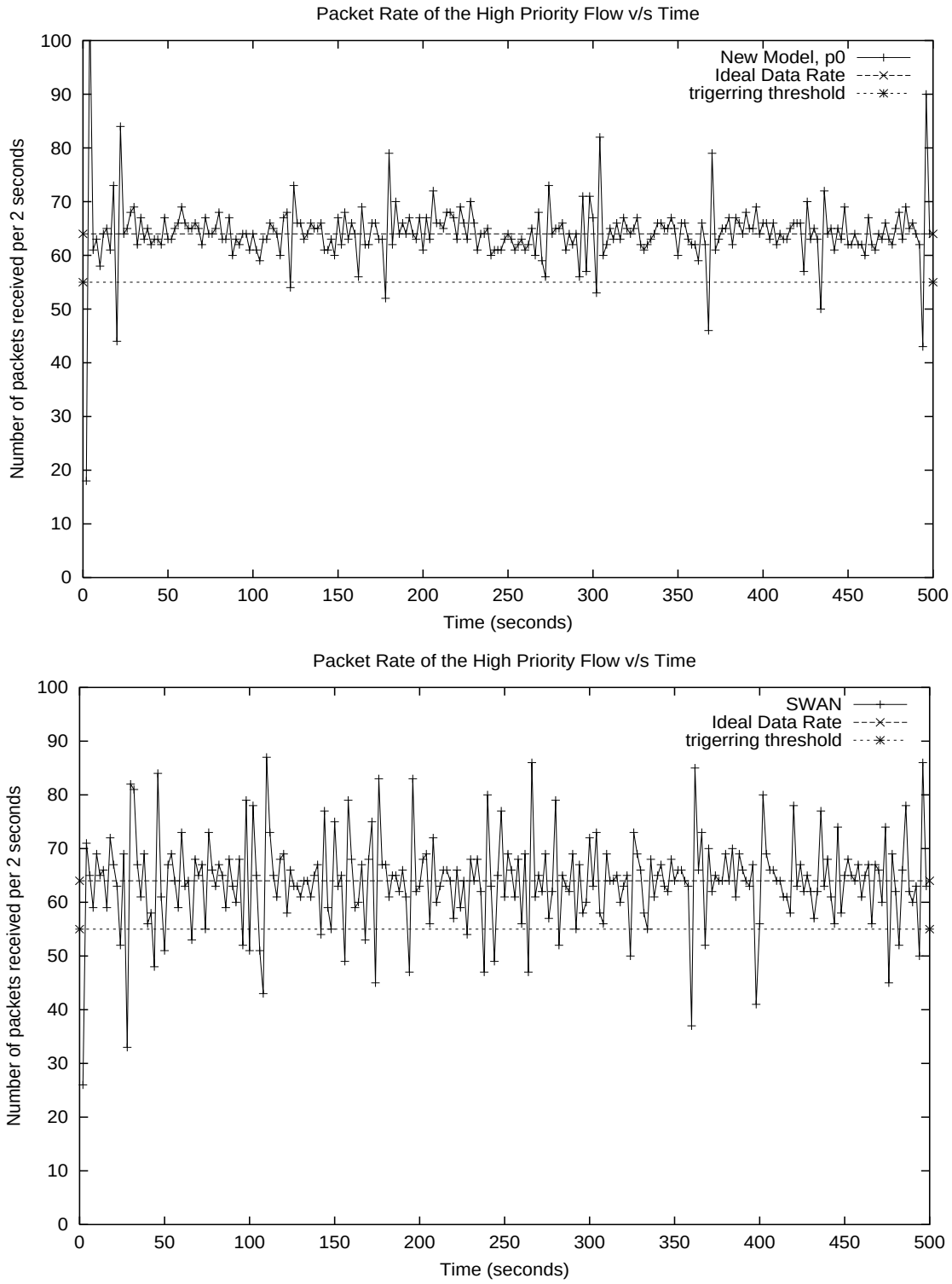

Figure 7.2: Variations in the packet rate of the high priority ¤ow with our new model (top) and with SWAN (bottom)

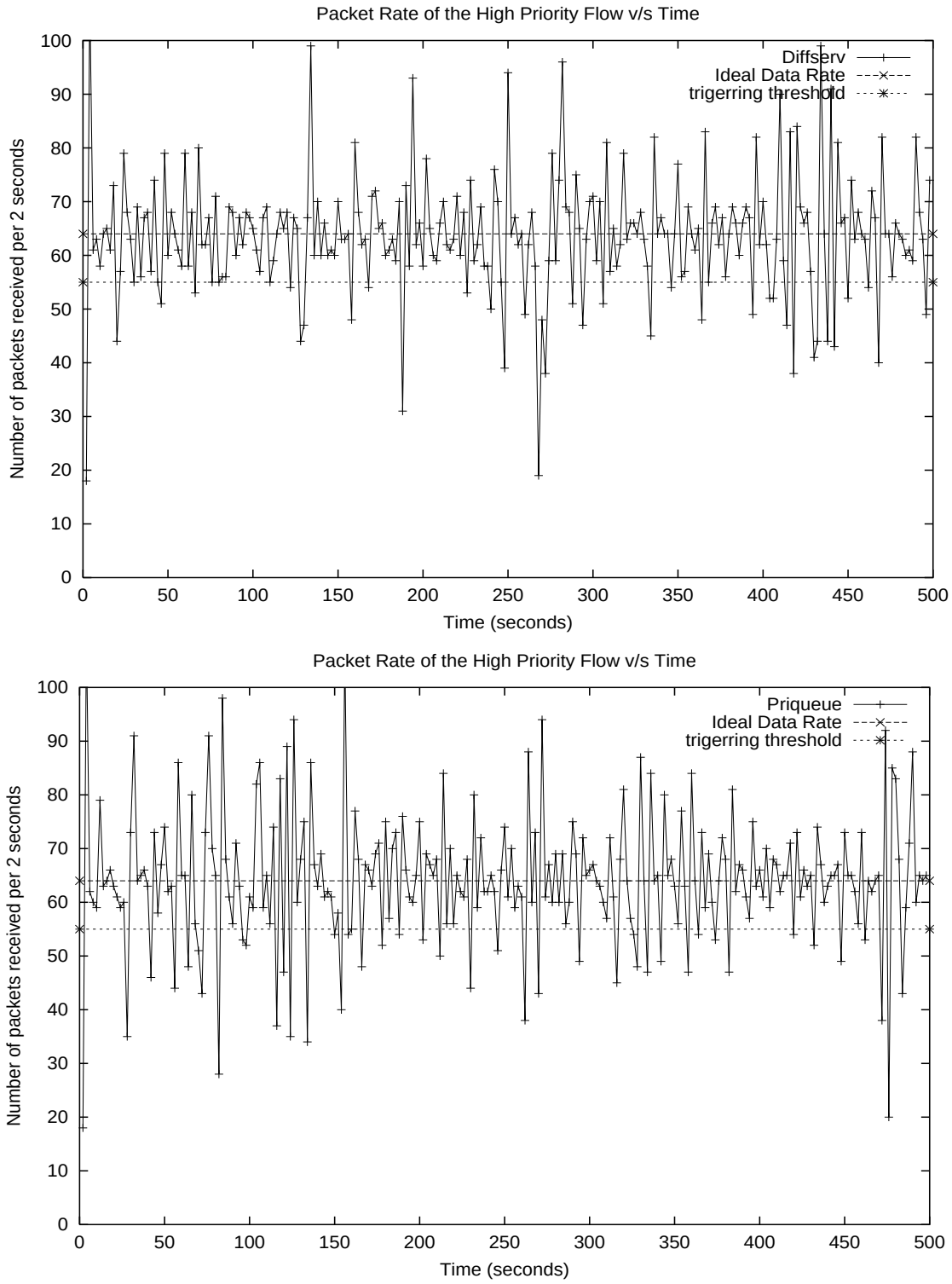

Figure 7.3: Variations in the packet rate of the high priority ¤ow with Diffserv (top) and without any QoS framework (bottom)

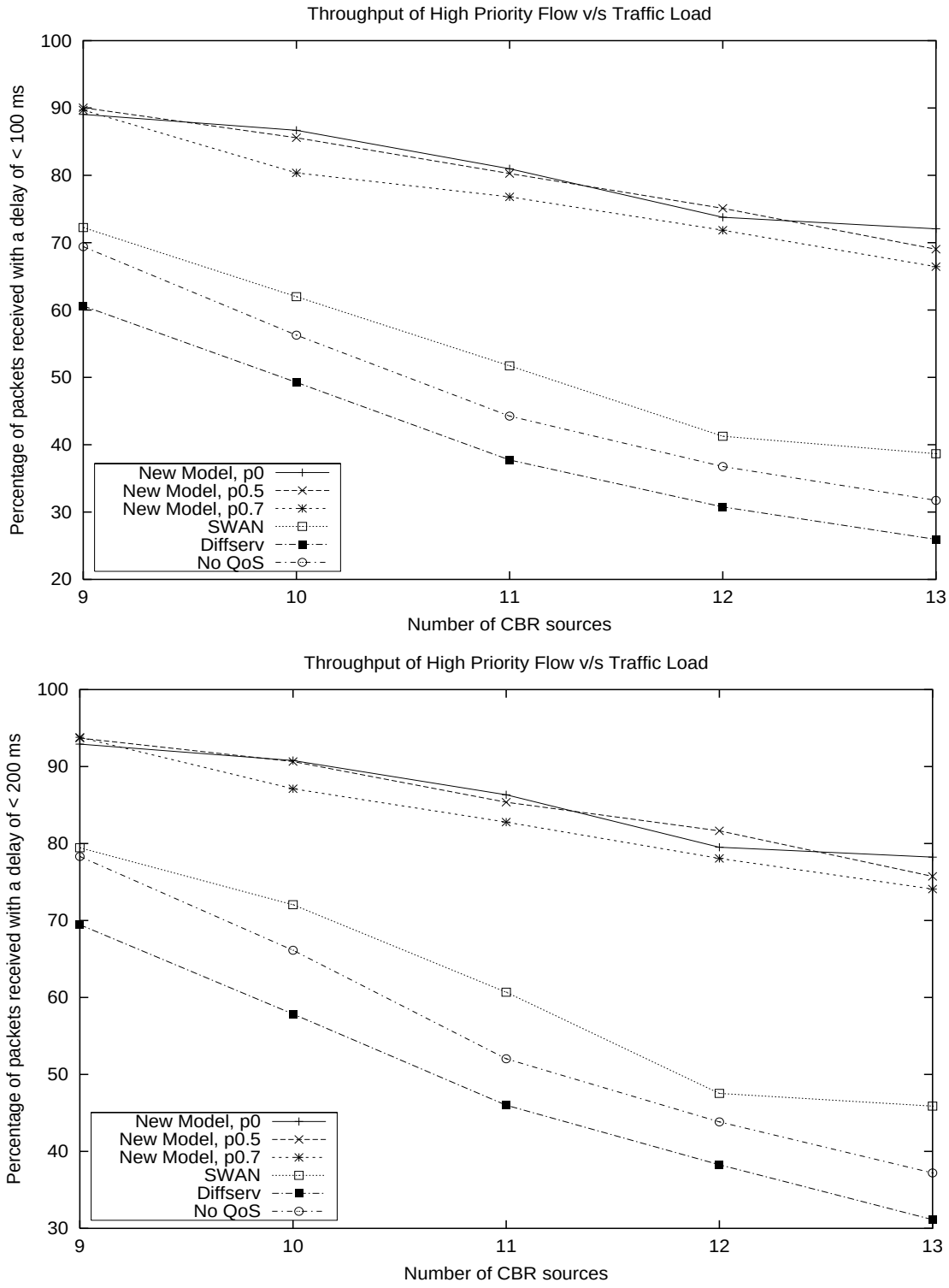

Figure 7.4: Percentage of packets successfully received within delay bounds of 100 ms (top) and 200 ms (bottom)

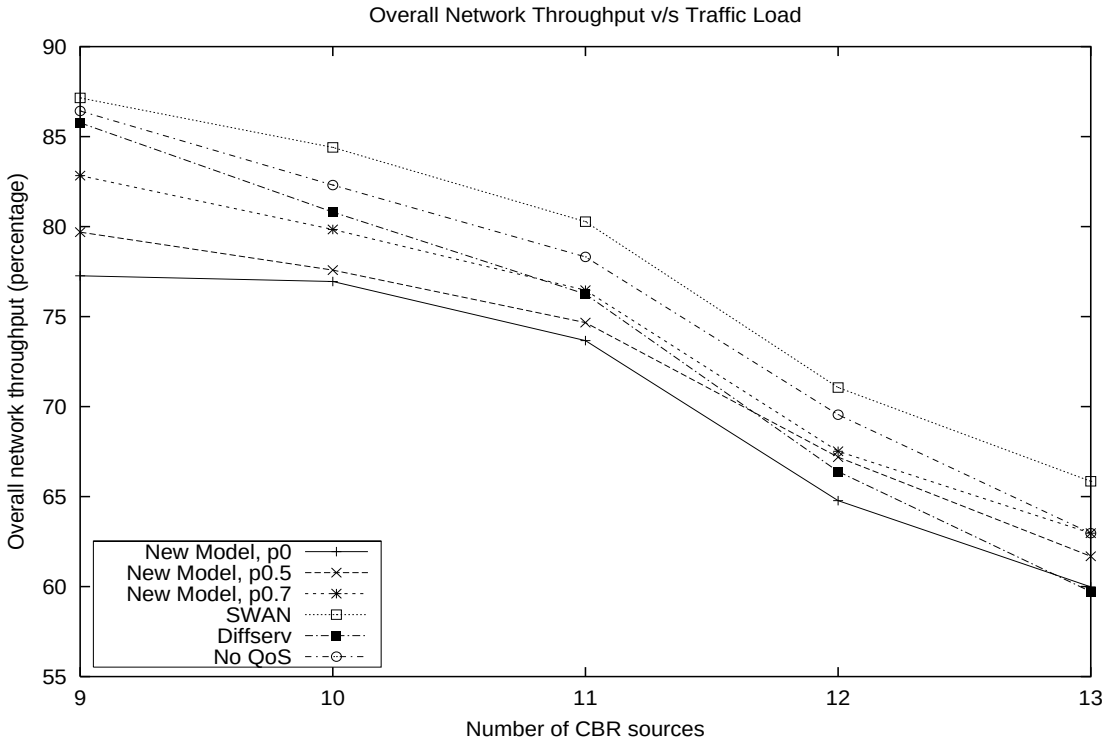

Figure 7.5: Overall throughput of the network

The improvement in the performance of the high priority ¤ow comes at a price. Figure 7.5 shows the overall throughput of the network. This is measured as the ratio of the total number of packets received at all destinations to the total number of packets transmitted by all sources, taken as a percentage. The transmission of $Squelch$ packet results in some ¤ows being stopped for some duration of time. This is implemented by dropping the packets being generated by the application agent at the new model at the source node itself. A decrease in the total throughput indicates that some of the network resources are not used. The throughput can be improved by increasing the p-value in the P-broadcast scheme as seen from the graph. However, an increase in the p-value correspondingly degrades the performance of the high priority ¤ow as evident from Figure 7.4. An appropriate trade-off is an implementation issue and not studied in detail here.

From Figure 7.6 it can be seen that the new model causes a reduction in the total control overhead. This is because, by broadcasting Squelch packets at points of heavy congestion, the model acts as a congestion control agent thereby reducing false link breaks. False link breaks are caused due to the mechanism of link layer feedback. If the link layer is not able to transmit a packet to the next hop even after a speci£c number of attempts, it assumes that the link to the next hop is broken and signals this to the network layer. The routing protocol then deletes the link from its table and initiates a fresh search for the destination that the broken link led to. During heavy congestion, the link layer may signal false link breaks if it is not able to transmit the packets to the next node, even if the link still exists. The new model also incurs a negligible overhead of its own.

7.2.2 Scenario 2

Topology : Mobile, Random
Speed of nodes : Variable. Maximum Speed is shown along the X-axis
Number of Scenarios : 5 scenarios corresponding to each maximum speed.

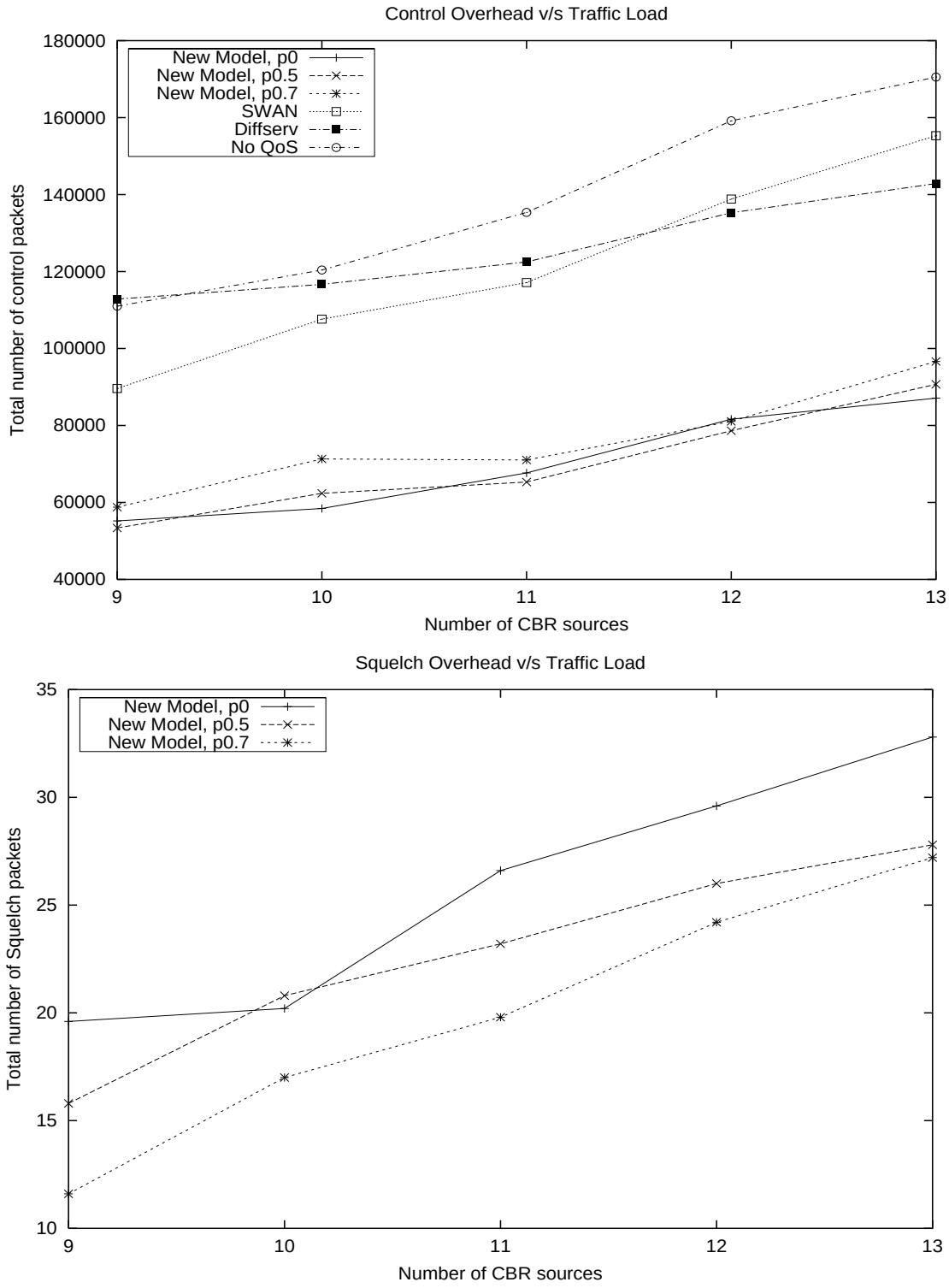

Figure 7.6: Total control overhead (top) and Squelch overhead (bottom)

Number of High Priority Flows : 1

Type of Low Priority Flows : CBR

Number of Low Priority Flows : 10

Rate of Low Priority Flows : 20 packets/sec, 800 byte packets (128 kbps)

In this scenario, the number of traf£c sources in the network and thus the load is kept constant. The performance of the model when the nodes move at different speeds is observed.

Figure 7.7 shows the number of packets successfully received at the destination with a delay of less than 100 ms and 200 ms. Even at high speeds, the performance of the model is pretty consistent, keeping the throughput much higher as compared to Diffserv and SWAN. This indicates that the model performs as effectively with a mobile network as with a static network.

Figure 7.8 shows the overall network throughput at different speeds. Although the overall throughput with the new model is lower than that of SWAN and Diffserv, there is a scope for improvement using the P-Broadcast scheme with a higher p value.

The control overhead with the new model is much lesser than with SWAN and Diffserv, an added asset of our model.

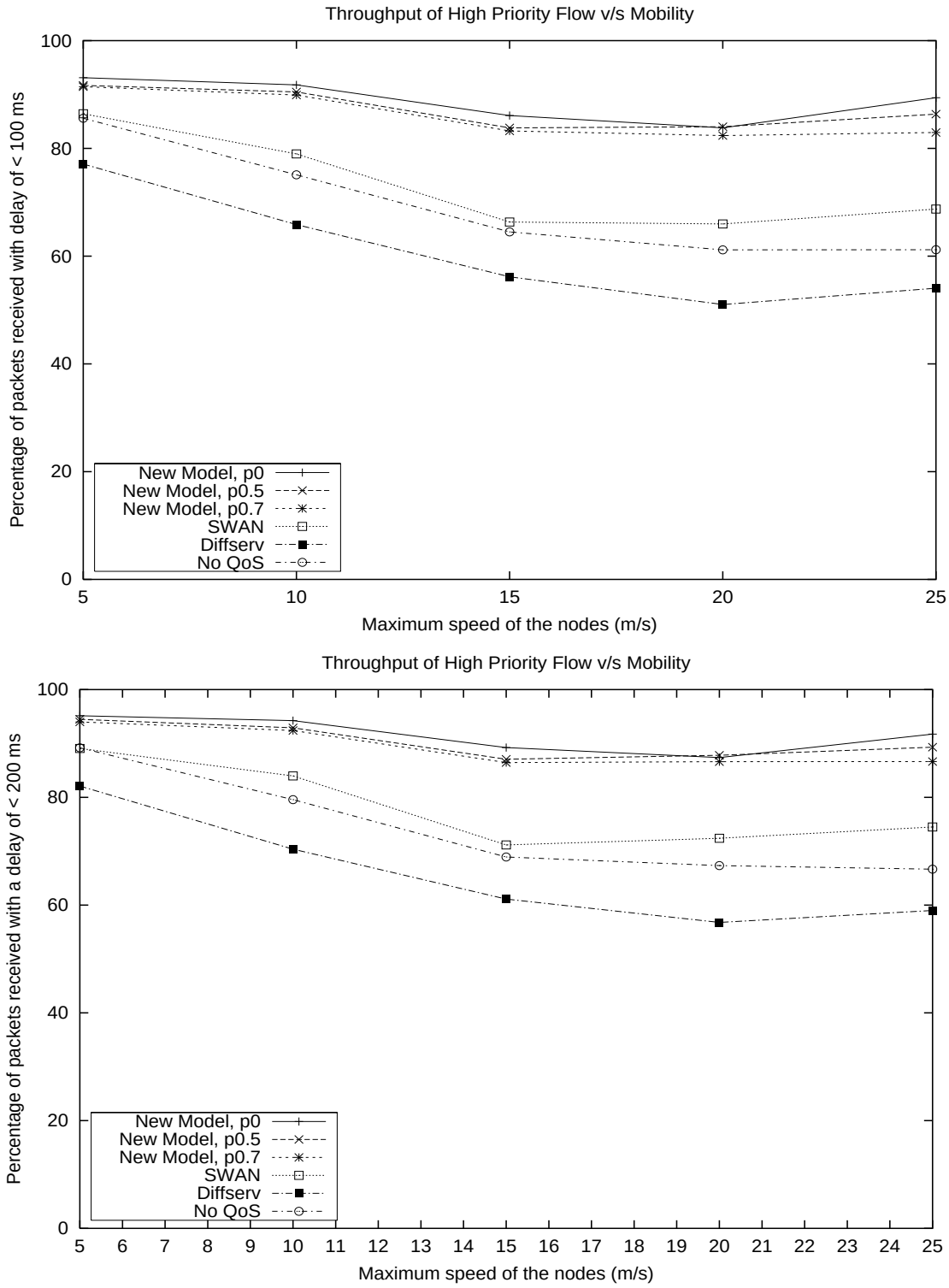

Figure 7.7: Percentage of packets successfully received within delay bounds of 100 ms (top) and 200 ms (bottom)

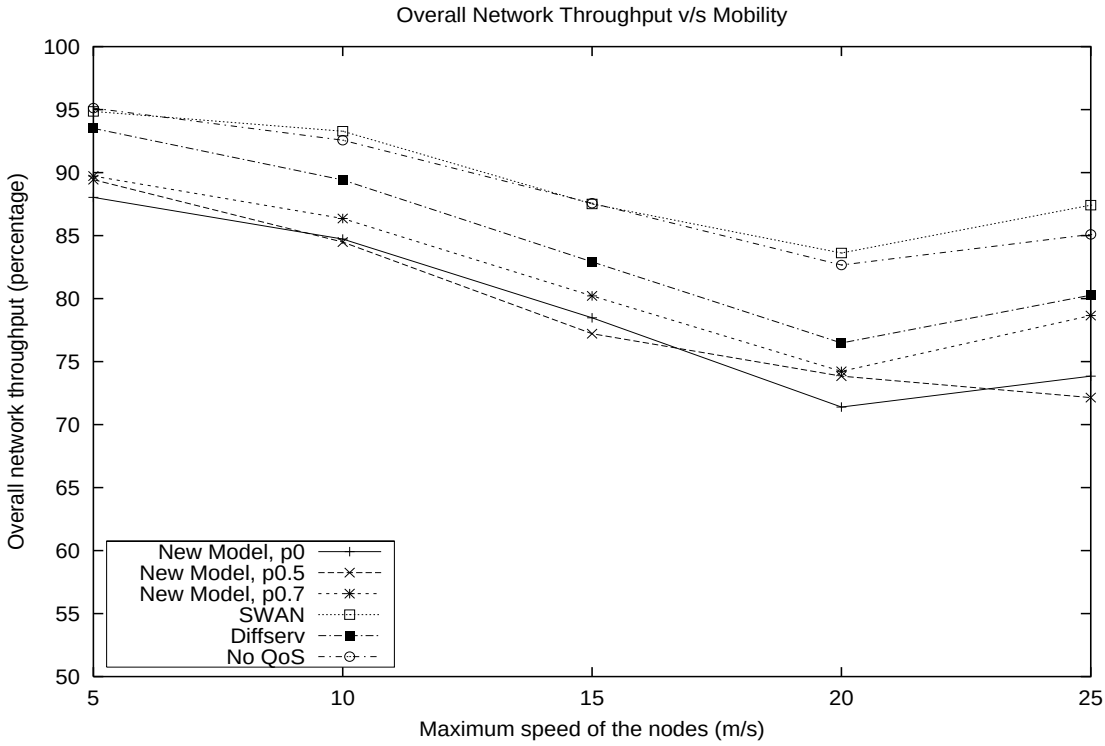

Figure 7.8: Overall throughput of the network

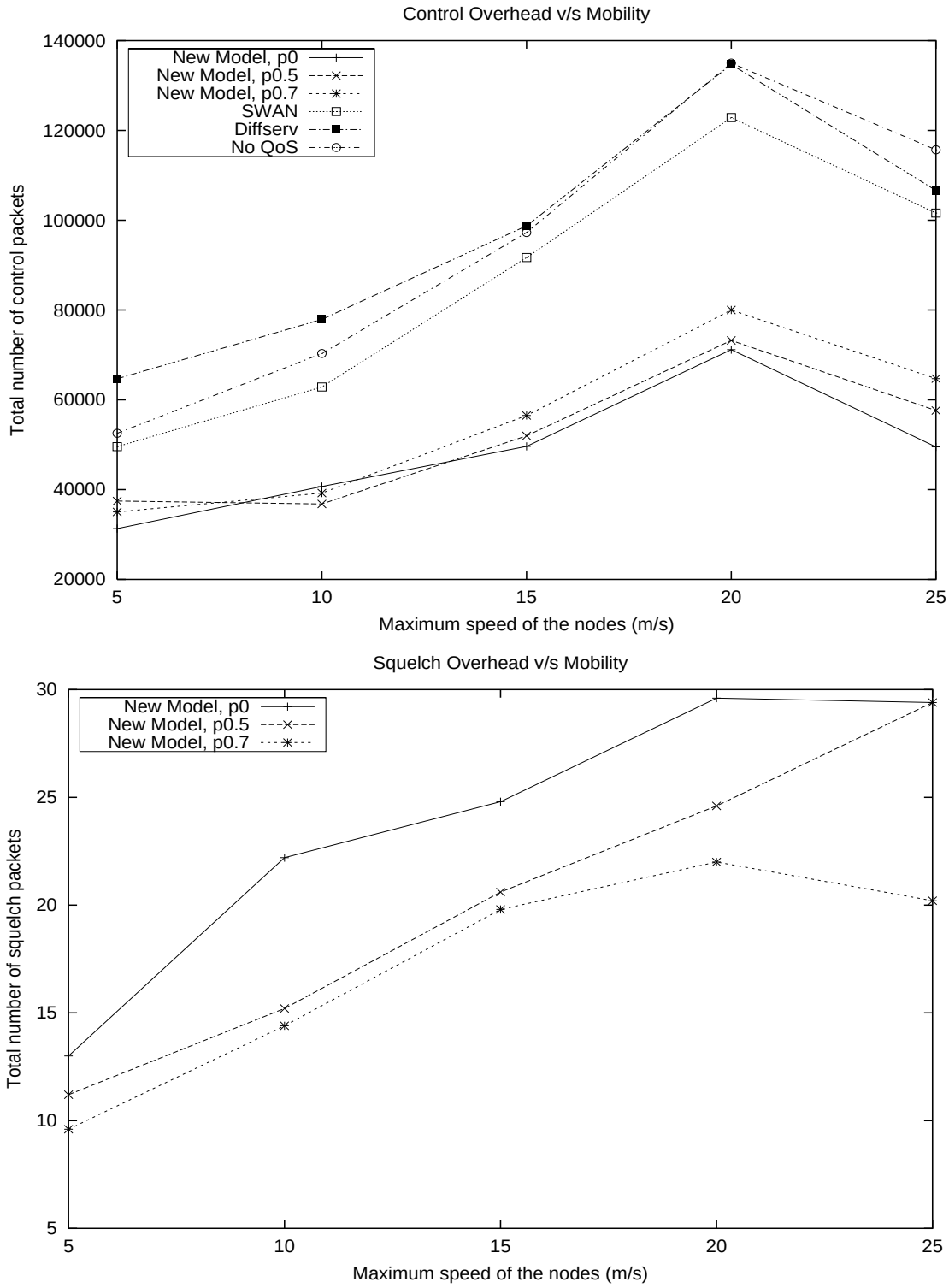

Figure 7.9: Total control overhead (top) and Squelch overhead (bottom)

8. Conclusion

This thesis builds a pathway toward a model for providing tight QoS guarantees in mobile ad hoc networks, an extremely challenging problem. The thesis begins by carefully de£ning a problem statement, which states our assumptions about the network, the traf£c characteristics, the type and number of traf£c classes desired and the kind of QoS assurances expected from the network. This statement lays the foundation of our framework and provides a guideline toward building the pathway. On our way to achieving our goal, we make several novel contributions to the £eld of ad hoc networks that are summarized below.

8.1 Our Contributions

1. We begin by studying the challenges and constraints that graced our way in Chapter 3. In particular, we discover the occurrence of synchronization effects caused due to periodicity of message dissemination as de£ned by the current versions of routing protocols for MANETs. This problem has never been studied before and puts heavy constraints on the development of a QoS framework that bases itself on the routes presented by the overlying routing algorithm. We study the mechanisms of the routing protocols that cause these effects and suggest simple solutions for overcoming them.

2. Further, we study the existing QoS schemes, in general for the Internet, and in particular for MANETs in Chapter 4. We perceive these schemes from the point of view of our problem statement and state each one's ability/inability toward achieving our goal. We isolate two stateless schemes, Diffserv and SWAN as potential building blocks of our framework and delve into the details of their mechanisms, carefully

studying their strengths and weaknesses. We analyze them both qualitatively and quantitatively and compare their performances under varied mobility and traf£c scenarios. We identify that Diffserv's classi£er, policer, queues and scheduler can form the basis of our model. This comparative study also forms a part of our novel contributions to the thesis.

3. Chapter 6 presents the design of our new model. In our initial design, we focus on providing tight QoS guarantees to ¤ows of highest priority class. The model is completely independent of the routing layer and the underlying link layer. Two main mechanisms, $Monitoring$ and $Corrective$ are responsible for maintaining the rate of high priority ¤ows. The detailed working of these mechanisms is presented in the chapter. A comprehensive performance analysis of the scheme, implemented and integrated as a part of NS-2 is presented in chapter 7. Simulation results are presented under a diverse set of mobility and traf£c scenarios. The results indicate the performance improvement of the high priority ¤ow using our new model over SWAN, a well known QoS framework in the MANET community.

8.2 Ongoing Work and Future Directions

Although the design of our new model presents a few steps taken toward our goal of a perfect QoS model, there are several directions yet to be explored to reach the £nal destination. A summary of the possible directions is enumerated below.

1. The results certainly show an improvement in the throughput of high priority ¤ow over that with SWAN and Diffserv. However, it still does not attain the 100% level as desired. One of the reasons for this could be the impact on the performance of the network of the overlying routing layer. A study of this impact is important to understand the true potential and constraints of our model. In our simulations, we only consider AODV as the routing protocol. We also need to study the performance

of the model with other reactive protocols such as DSR and proactive protocols such as OLSR. See our work [28] comparing the performance of AODV, DSR and OLSR with respect to Quality of Service.

2. With a large number of parameters involved in the model, a tuning of the parameters is important to obtain the best results. In our work, several parameters such as the monitoring window (2 seconds) and the triggering threshold (55 packet/2 seconds) are kept constant. A detailed study of the dependence on these parameters of the performance of the model is required for an improved performance.

3. Most importantly, we need to perform a theoretical analysis of the model that can present theoretical bounds on the level of guarantees that can be achieved. Such a study may provide answers to several questions such as :

   - What are the bounds on the end-to-end latency of ¤ows while achieving a 100% throughput in an ad hoc network?
   - Given a network, how many high priority ¤ows can be simultaneously admitted?

   This study will also allow the MANET community to have con£dence in our model.

4. The model currently concentrates on providing tight guarantees to high priority ¤ows. However, the problem statement de£nes the need for soft guarantees to other lower priority ¤ows. Such guarantees are possible with other QoS models such as SWAN. Hence, we need to explore the possibility of integrating these models with our new model.

5. The model is currently designed to optimize the performance of high priority ¤ows. For this optimization, we need to sacri£ce the overall network throughput. The relation between the level of guarantees achieved and the overall network utilization

needs to be clearly de£ned. Based on the importance of the ¤ows, an appropriate trade-off can then be made.

We have mechanisms in place for these studies and form a subject of our future work.

Bibliography

[1] Z. Haas and S. Tabrizi, "On some challenges and design choices in ad-hoc communications," in *Proceedings of Military Communications Conference*, Oct. 1998, vol. 1, pp. 187–912.

[2] S. Chakrabarti and A. Mishra, "Qos issues in ad hoc wireless networks," *IEEE Communications Magazine*, vol. 39, no. 2, pp. 142–148, February 2001.

[3] S. Papavassiliou, S. Tekinay, K. Malick, and K. Walker, "Performance evaluation framework and quality of service issues for mobile ad hoc networks in the mosaic atd," in *Proceedings of* 21^{st} *Century Military Communications Conference*, Oct. 2000, vol. 1, pp. 297–303.

[4] H. Arora and H. Sethu, "Performance analysis of effects of mobility patterns on network performance in mobile ad hoc networks"," in *Proceedings of Applied Telecommunication Symposium*, San Diego, California, April 2002.

[5] IEEE WG, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Speci£cations," *IEEE 802.11 Standard*, 1999.

[6] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, August 2002.

[7] "http://www.isi.edu/nsnam/ns," .

[8] S. Wu, S. Ni, Y. Tseng, and J. Sheu, "Route maintainance in a wireless mobile ad hoc network," in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, 2000, pp. 3015–3024.

[9] T. Clausen and P. Jacquet, "Optimized link state routing protocol," *http://www.ietf.org/internet-drafts/draft-ietf-manet-olsr-11.txt*, July 2003.

[10] S. Floyd and V. Jacobson, "The synchronization of periodic routing messages," *IEEE Transactions on Networking*, vol. 2, no. 2, pp. 122–136, April 1994.

[11] E. Crawley, R. Nair, B. Rajagopalan, and H. Sandick, "A framework for qos-based routing in the internet," *RFC 2386,http://www.ietf.org/rfc/rfc2386.txt?number=2386*, August 1998.

[12] R. Braden, D. Clark, and S. Shenker, "Integrated services in the internet architecture: an overview," *RFC 1633, http://www.ietf.org/rfc/rfc1633.txt?number=1633*, June 1994.

[13] K. Nichols, V. Jacobson, and L. Zhang, "A two-bit differentiated services architecture for the internet," *RFC 2638, ftp://ftp.rfc-editor.org/in-notes/rfc2638.txt*, July 1999.

[14] D. Black, S. Brim, B. Carpenter, and F. Le Faucheur, "Per hop behavior identi£cation codes," *RFC 3140, ftp://ftp.rfc-editor.org/in-notes/rfc3140.txt*, June 2001.

[15] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured forwarding phb group," *RFC 1999, ftp://ftp.rfc-editor.org/in-notes/rfc2597.txt*, June 1999.

[16] B. Davie, A. Charny, and J.C.R. Bennett et. al., "An expedited forwarding per-hop behavior," *RFC 2598, ftp://ftp.rfc-editor.org/in-notes/rfc3246.txt*, March 2002.

[17] L. Georgiadis, P. Jacquet, and B. Mans, "Bandwidth reservation in multihop wireless networks: Complexity and mechanisms," *http://www.inria.fr/rrrt/rr-4876.html*.

[18] R. Sivakumar, P. Sinha, and V. Bharghavan, "Cedar: A core-extraction distributed ad hoc routing algorithm," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1454–1465, August 1999.

[19] C. Perkins and E. Belding-Royer, "Quality of service for ad hoc on-demand distance vector routing," *http://www.ietf.org/html.charters/manet-charter.html*, October 2003.

[20] J. Sobrinho and A. Krishnakumar, "Quality-of-service in ad hoc carrier sense multiple access wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, August 1999.

[21] Gahng-Seop Ahn, A. Campbell, A. Veres, and Li-Hsiang Sun, "Supporting service differentiation for real-time and best-effort traf£c in stateless wireless ad hoc networks (swan)," *IEEE Transactions on Mobile Computing*, vol. 1, no. 3, pp. 192–207, July-Sept. 2002.

[22] S.-B. Lee, G.-S. Ahn, and A. Campbell, "Improving udp and tcp performance in mobile ad hoc networks with insignia," *IEEE Communications Magazine*, vol. 39, no. 6, June 2001.

[23] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, August 1993.

[24] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on demand distance vector routing protocol," *http://www.ietf.org/rfc/rfc3561.txt*, July 2003.

[25] D. Johnson, D. Maltz, and Yin-Chun Hu, "The dynamic source routing protocol," *Internet Draft, IETF MANET Working group*, April 2003.

[26] "http://comet.columbia.edu/swan," .

[27] H. Arora and H. sethu, "A simulation study of the feasibility of differentiated services architecture for qos in mobile ad hoc networks," in *Proceedings of Applied Telecommunications Symposium*, San Diego, CA, April 2002.

[28] J. Novatnack, L. Greenwald, and H. Arora, "Evaluating ad hoc routing protocols with respect to quality of service," in *Under Review*.