Network Delay Control Through Adaptive Queue Management

by

Lee Booi Lim

A Doctoral Thesis

Submitted in partial fulfilment
of the requirements for the award of

Doctor of Philosophy
of
Loughborough University

5th August 2011

Abstract

Timeliness in delivering packets for delay-sensitive applications is an important QoS (Quality of Service) measure in many systems, notably those that need to provide real-time performance. In such systems, if delay-sensitive traffic is delivered to the destination beyond the deadline, then the packets will be rendered useless and dropped after received at the destination. Bandwidth that is already scarce and shared between network nodes is wasted in relaying these expired packets. This thesis proposes that a deterministic per-hop delay can be achieved by using a dynamic queue threshold concept to bound delay of each node. A deterministic per-hop delay is a key component in guaranteeing a deterministic end-to-end delay. The research aims to develop a generic approach that can constrain network delay of delay-sensitive traffic in a dynamic network. Two adaptive queue management schemes, namely, DTH (Dynamic THreshold) and ADTH (Adaptive DTH) are proposed to realize the claim. Both DTH and ADTH use the dynamic threshold concept to constrain queuing delay so that bounded average queuing delay can be achieved for the former and bounded maximum nodal delay can be achieved for the latter. DTH is an analytical approach, which uses queuing theory with superposition of N MMBP-2 (Markov Modulated Bernoulli Process) arrival processes to obtain a mapping relationship between average queuing delay and an appropriate queuing threshold, for queue management. While ADTH is an measurement-based algorithmic approach that can respond to the time-varying link quality and network dynamics in wireless ad hoc networks to constrain network delay. It manages a queue based on system performance measurements and feedback of error measured against a target delay requirement. Numerical analysis and Matlab simulation have been carried out for DTH for the purposes of validation and performance analysis. While ADTH has been evaluated in NS-2 simulation and implemented in a multi-hop wireless ad hoc network testbed for performance analysis. Results show that DTH and ADTH can constrain network delay based on the specified delay requirements, with higher packet loss as a trade-off.

Keywords: adaptive queue management, dynamic queue threshold, delay-sensitive, queuing delay, nodal delay, end-to-end delay, wireless ad hoc networks

Acknowledgement

This work is sponsored by the Engineering and Physical Science Research Council (EPSRC), United Kingdom, under grant CASE/CNA/07/82 and the Royal Society, United Kingdom. I am most grateful to my supervisors, Dr Lin Guan, Dr Iain Phillips from Computer Science Department and Dr Alan Grigg from BAE Systems, for their support and guidance throughout the journey of my PhD research. It was a joyous experience to work with them.

I would like to thank Dr Colin Machin for his support in purchasing Gumstix and sourcing of a cheaper alternative. I would also like to thank Dr Olaf Maennel for his feedback during the annual review meetings and the thesis.

I would like to extend my gratitude to the support staff in the Computer Science Department for their help in facilitating my research. I would like to thank Mr~Gurbinder Singh Samra for his assistance in lab equipments purchases and thesis printing, Mr Jeff Pitchers for his help in restoring data from my crashed hard drive and setting up the servers in lab, Mr Kip Sahnsi for his help in setting up the equipments and cabling, Mr Satish Bedi for his help in making my work place more ergonomically suitable and Mr Richard Mee for dealing with the laptop repair. I would like to thank Ms Christine Bagley and Mrs Judith Poulton for their help in administrative jobs. I would like to thank Ms Jo McOuat for her help in raising purchase orders for lab equipment.

I would like to thank the members of my research group, other research students and staff in my department for making the office a great place to work and their friendships.

Finally, I would like to thank my family members for their support especially to my husband, Dr Tze Yang Hin, for keeping me motivated and his company throughout the journey. I would like to dedicate this thesis to my beloved parents who have provided their best support to me since I was born.

Publications

- L. B. Lim, L. Guan, A. Grigg, I. W. Phillips, X. G. Wang, I. U. Awan, "Bounding Queuing Delay in a Router Based on Superposition of N MMBP Arrival Process," ICCCN, pp.1-6, 2009 Proceedings of 18th International Conference on Computer Communications and Networks, 2009.

- Lee Booi Lim, Lin Guan, Alan Grigg, Iain W. Phillips, Xingang Wang, Irfan U. Awan, "RED and WRED Performance Analysis Based on Superposition of N MMBP Arrival Process," AINA, pp.66-73, 2010 24th IEEE International Conference on Advanced Information Networking and Applications, 2010.

- Jie Wang, Lin Guan, Lee Booi Lim, X.G. Wang, A. Grigg, I. Awan, I. Philips, X. Chi, QoS enhancements and performance analysis for delay-sensitive applications, Journal of Computer and System Sciences, In Press, Corrected Proof, Available online 25 February 2010, ISSN 0022-0000, DOI: 10.1016/j.jcss.2010.02.007.

- L.B. Lim, L. Guan, A. Grigg, I.W. Phillips, X.G. Wang, I.U. Awan, Controlling mean queuing delay under multi-class bursty and correlated traffic, Journal of Computer and System Sciences, In Press, Corrected Proof, Available online 17 August 2010, ISSN 0022-0000, DOI: 10.1016/j.jcss.2010.08.007.

Contents

List of Figures

List of Tables

List of Abbreviations

ACK Acknowledgment

ADTH Adaptive Dynamic THreshold

AIMD Additive Increase Multiplicative Decrease

AODV Ad Hoc On-Demand Distance Vector

AP Access Point

AQM Active Queue Management

bps bits per second

CBR Constant Bit Rate

CSMA/CA Carrier Sense Multiple Access With Collision Avoidance

CTS Clear To Send

CW Contention Window

CWND Congestion WiNDow

DCF Distributed Coordination Function

DDP Delay Differentiation Parameter

DiffServ Differentiated Services

DSDV Destination-Sequenced Distance-Vector

DSR Dynamic Source Routing

DF Drop front

DT Drop Tail

DTH Dynamic THreshold

E2E End-to-end

EDCA Enhanced Distributed Channel Access

EPSRC Engineering and Physical Sciences Research Council

EWMA Exponentially Weighted Moving Average

FIFO First In First Out

HOL Head Of Line

IEEE Institute of Electrical and Electronic Engineering

IFQ InterFace Queue

IntServ Integrated Services

IP Internet Protocol

kbps kilo bits per second

LAN Local Area Networks

MAC Medium Access Control

MANET Mobile Ad hoc NETwork

Mbps Mega bits per second

MIMD Multiplicative Increase Multiplicative Decrease

MIPS Millions Instruction Per Second

MMBP Markov Modulated Bernoulli Process

MMPP Markov Modulated Poisson Process

MSE Mean Squared Error

OWD One Way Delay

PAD Proportional Average Delay

PDD Proportional Delay Differentiation

PFIFO Packet FIFO

PGPS Packet Generalized Processor Sharing

PI Proportional Integral

PQ Priority Queuing

QoS Quality of Service

RED Random Early Detection

RSVP Resource ReSerVation Protocol

RTCP Real-time Transport Control Protocol

RTP Real-time Transport Protocol

RTT Round Trip Time

RTS Request To Send

SCV Squared Coefficient of Variation

SD Standard Deviation

SE Standard Error

SLA Service Level Agreement

TCP Transmission Control Protocol

TDMA Time Division Multiple Access

UDP User Datagram Protocol

VBR Variable Bit Rate

WLAN Wireless Local Area Network

WTP Waiting Time Priority

Chapter 1

Introduction

Wireless networking is a booming technology today and an active research field in academia and industry. Wireless networks can be divided into two main categories: those with infrastructure support, and those without (also known as wireless ad hoc networks). For wireless networks with infrastructure support, there will be a few special nodes acting as access points to manage communication among other nodes. All the communication between nodes will go through access points. While a wireless ad hoc network is a decentralized wireless network. Each node in the network can communicate with its immediate neighbours without the need of access points. Each node in the network is willing to forward data from its neighbour nodes to their destinations [130, 150].

Their self-configuration and self-organizing nature makes wireless ad hoc networks suitable for deployment at areas that lack infrastructure. Dynamic and adaptive routing protocols enable a wireless ad hoc network to be formed and deployed quickly for various purposes, such as communications in battlefields, disaster scenes and areas lack of networking infrastructure.

Typically, applications can be divided into two classes: delay-sensitive and non delay-sensitive. The need for network delay control is of high concern with the emergence of delay-sensitive applications for emergency response, battlefield communication, event monitoring, surveillance, media streaming, multi-player online games and Internet telephony [33, 76, 91, 118, 129, 162]. Timeliness of data delivery is a vital quality of service (QoS) performance measure for such applications. Delay in data delivery for real-time communication may annoy users. In hard real-time systems, data are rendered useless or redundant if they arrive later than their deadlines and even cause system failures. There are several transport protocols, such as TCP (Transmission Control Protocol) and UDP (User Datagram Protocol), that can be used to provide end-to-end (E2E) communication for the applications. TCP is usually used to transport traffic for applications that requires reliable communication but not timing sensitive, whereas UDP is used to transport traffic for applications that is timing sensitive but sacrificing reliability. Packet loss is preferable compared to unbounded delay of packets for delay-sensitive applications [103, 105].

Typically in wired networks, over-provisioning is a common strategy used by telecommunication service providers to ensure QoS requirements are met [162]. However, over-provisioning becomes a luxury in wireless ad hoc networks especially for those networks formed by a group of resource-constrained devices with limited processing power, limited bandwidth and limited energy [162]. Additional mechanisms to support QoS in wireless ad hoc networks therefore becomes an inevitable task [30, 143, 147, 166].

QoS provisioning for wireless ad hoc networks is typically more challenging compared with wired networks owed to the nature of wireless communication. Such technologies are prone to errors such as signal fading or attenuation problem, multi-path propagation problem, interference from neighbour nodes and environmental interference [130, 143, 194]. These factors cause the link quality to vary over time. Node orientation, shadowing of objects, hidden terminal and exposed terminal problems also contribute to degradation of link quality [11, 63, 147].

Guaranteeing QoS, such as bounded end-to-end delay, maximum jitter and limited loss rate, for delay-sensitive applications in such networks is a great challenge [76,169]. End-to-end delay is an aggregation of nodal delays from a source to a destination (see Section 2.2), it increases with number of hops between a source and a destination in the networks [76]. A deterministic end-to-end delay can be achieved indirectly if nodal delay of each intermediate hop is bounded. This is consistent with the views of Burbank *et al.* [26], Vergados *et al.* [170], Yang and Kravets [183] that the end-to-end QoS can only be achieved through a consistent, predictable and deterministic per-hop behaviour.

In wireless ad hoc networks, nodal delay is mainly caused by queuing delay and MAC (Medium Access Control) contention delay [122]. A predictable delay is necessary to overcome the delay variation caused by contention in MAC layers, interference, fluctuation in link quality [63,78]. Queuing delay is easier to constrain compared to MAC layer delay. To constrain MAC layer delay, tuning of MAC layer contention mechanism or MAC layer scheduling mechanism are required [64,65,83,85,101,108,161,163,170,172,183]. Co-operation of nodes in the network or modification of MAC firmware or hardware is needed to achieve this. Therefore, interoperability and legacy issues may hinder the effectiveness of these approaches. Whereas constraining queuing delay requires a proper queue management scheme to decide when to enqueue or to drop packets. The maximum queuing delay experienced by a system is only determined by the maximum queue size and system throughput. Inappropriate setting of the queue size or thresholds may lead to system performance degradation such as large queuing delay if the queue size is too large or high packet dropping rate if the queue size is too small [110,123]. Besides these, achievable system throughput highly depends the link quality and interference of surroundings.

Even though queue management schemes have been actively researched in wired networks, most wireless ad hoc networks still use a Drop Tail (DT) queue discipline [40, 97, 99, 135]. Some researchers [3, 40, 62, 97–99, 109–111, 125, 126, 135, 136,166,179,180] borrow the concept of active queue management (AQM) from the wired network domain. However, these queue management schemes mainly aim to alleviate network congestion, to improve link utilization, to improve throughput and to provide fairness for traffic flows. Most of these schemes drop packets probabilistically to achieve the goals with the assumption that traffic sources can response to packet loss events. For such schemes, low queuing delay is maintained via rate adaptation at TCP senders (transport agents of traffic sources) resulting from packet loss events used as congestion indicators. However, UDP senders, which are transport agents of delay-sensitive traffic sources, are non-responsive and do not adapt their sending rate based on these events. Queuing delays are not bounded with the existing schemes. Surprisingly, there is a lack of consideration to delay-sensitive traffic in queue management schemes. Furthermore, most of the queue management schemes ignore MAC layer delays.

There is a need to constrain both queuing delay and MAC layer delay to bound nodal delay. In view of this, a generic and adaptive queue management approach that can adapt to fluctuation of system performance and takes care of MAC layer delay is required to bound nodal delay of wireless nodes.

1.1 Aims

The aim of this research is to develop a generic queue management approach that can constrain network delay for delay-sensitive traffic in a multi-hop wireless ad hoc network. The IEEE 802.11 standard [79] is dominant in wireless networks [11, 63, 77], therefore the research idea proposed is based on an IEEE 802.11 multi-hop wireless ad hoc network. The objectives of the research are:

- To investigate the factors which contribute to network delay in wireless ad hoc networks at a node level perspective.

- To investigate a generic approach that can give a deterministic per-hop delay, including:

  - queue management solution to bound queuing delay and nodal delay

  - scalable and lightweight solution that can adapt to network dynamics autonomously
  - independent from MAC layer but aware of MAC layer delay

- To investigate the impact of queue sizing on network delay.

The contributions of the thesis are outlined as below:

- An adaptive queue management scheme, namely, DTH (Dynamic THreshold), is proposed to constrain average queuing delay to a specified delay requirement. DTH relies on the mapping relationship between queuing delay and queuing threshold to manage a queue. A discrete-time queuing model, which uses superposition of MMBP-2 (Markov Modulated Bernoulli Process) arrival process to model aggregated Internet traffic, is developed to derive the mapping relationship. The simulation results shows that DTH is able to bound average queuing delay to the specified value. More details are presented in Chapter 3. The discrete time model has been extended for RED (Random Early Detection) and WRED (Weighted RED) performance analysis [114].

- A detailed delay analysis for a multi-hop wireless ad hoc network has been carried out to analyze the factors that may affect network delay of delay sensitive traffic. The analysis shows that queuing delay and MAC layer delay are two major components contributing to large nodal delay in a contention-based multi-hop wireless network. The simulation also shows that factors, such as traffic load, packet size and queue size, contribute to network dynamics apart from the factors of wireless network characteristics. All these factors cause fluctuation in system performance and lead to variation in queuing delay and MAC layer delay. More details are presented in Chapter 4. The findings from the analysis become the design factors of the scheme proposed next.

- An adaptive queue management scheme, namely, ADTH (Adaptive Dynamic THreshold), is proposed to bound per-hop nodal delay to a specified delay requirement in a multi-hop wireless ad hoc network. ADTH is a measurement-based queue management scheme that can adapt to network dynamics autonomously. The ADTH design is generic and independent from the underlying MAC layer. Nodal delay of a node is bounded through an adaptive dynamic queue threshold to compensate for variation in MAC layer delay. The simulation results show that ADTH is able to bound nodal delay to the specified delay value. The feasibility of adopting ADTH to bound nodal delay and end-to-end delay in a multi-hop wireless network has been validated in a testbed. The processing overhead incurred by the ADTH controller is low and has minimal impact to the system performance. More details are presented in Chapter 5 and Chapter 6.

1.2 Thesis Outline

The rest of the thesis is organized as following:

Chapter 2 gives an overview of existing QoS provisioning schemes in maintaining network QoS with the focus in network delay area. These approaches are studied and summarized. The literature review is then concluded with a gap analysis and hence motivates the direction of this research.

Chapter 3 presents an analytical-based queue management scheme that uses a dynamic threshold concept to constrain average queuing delay at core routers. The proposed scheme is named DTH. A discrete-time queuing model is developed to calculate the optimum queuing threshold under predetermined network condition. The proposed scheme has been simulated and validated with a Matlab simulation.

Chapter 4 presents a detailed delay analysis in a small multi-hop wireless ad hoc network in order to analyze the important factors that contribute to huge variation in network delay under different network conditions. The delay analysis shows that an online-based adaptive queue management scheme is needed and more effective in bounding the per-hop delay in wireless domain.

Chapter 5 presents an online-based adaptive queue management scheme based on the findings in Chapter 4 that improves the DTH scheme and applies the dynamic threshold concept in the wireless domain to constrain per-hop nodal delay. The scheme, named ADTH, has been simulated and validated using the NS-2 network simulator.

Chapter 6 presents a performance analysis of ADTH in a testbed. The analysis shows that ADTH can feasibly be implemented and adopted for network delay control in a multi-hop wireless ad hoc network. ADTH is implemented as a queue management scheme in an embedded Linux environment and real-time traffic is injected to wireless nodes through a hardware traffic generator.

Chapter 7 discusses the possible application of ADTH with other QoS provisioning schemes in the domain of wireless ad hoc networking. This chapter shows that the delay bounding nature of ADTH and its internal states can facilitate other QoS schemes, such as routing, to find an optimum path and admission control to admit or reject traffic flows.

Chapter 8 summarizes the thesis to show how the aims have been achieved and gives suggestions for future work.

Chapter 2

An Overview of Network Delay Control

2.1 Introduction

This chapter presents a comprehensive literature review on network delay control schemes. The network delay control schemes reviewed are categorized and summarized in Tables 2.1 - 2.9. Understanding of network latency (also known as network delay) characteristics and controlling network latency have become critical with the growing demands of soft real-time and hard real-time applications. Data communication for applications such as voice and video streaming, military command and control, surveillance and monitoring are delay-sensitive [33, 76, 91, 118, 129, 162].

QoS requirements are tightly coupled with applications' attributes. Some applications require low bandwidth, but are stringent on delay, such as VoIP (Voice over Internet Protocol). While for Internet web browsing high throughput is required but delay is tolerable. QoS requirements can be divided into three broad categories: non real-time, soft real-time and hard real-time [78].

- **Non real-time**: There is no specified time constraint on the end-to-end delay of data communication. This is also known as best-effort communication. Packets can arrive at a destination out of sequence and any time. Examples of non real-time applications are email service, FTP (File Transfer Protocol) application and Internet web browsing.

- **Soft real-time**: There is a time constraint on the end-to-end delay of data communication. There is a deadline associated with packets arriving time. Soft real-time applications, such as video streaming application, can tolerate some lateness. A video streaming application requires packets to be received at fix interval. Video packets are buffered before playout to compensate for latency and jitter.

- **Hard real-time**: There is a strict time constraint on the end-to-end delay of data communication. Packets are rendered useless if the packets arrive after the deadline. This may deteriorate quality significantly or cause disastrous impact. The significance of the impact depends on the application requirements. Taking an example of room temperature monitoring application, sensor reading of room temperature is reported over the network periodically for fire alert monitoring. If the reporting of sensor data misses the deadline then the sensor data are useless but the impact is not significant; newer sensor data are coming on their way to represent the latest room temperature. However, if the room caught fire and the control room depends on the sensor data for fire detection, then the impact is serious as it causes delay in rescue.

Traffic from non real-time applications is classified as delay-tolerant traffic, while traffic from soft real-time and hard real-time applications are classified as delay-sensitive traffic. Soft real-time and hard real-time applications are also known as delay sensitive applications. There are a few transport protocols, such as TCP and UDP, that can be used to provide end-to-end communication services for applications. TCP cannot be used to transport delay-sensitive traffic due to its reliability and congestion control features. Typically, UDP is used since packet loss is preferably compared to unbounded delay of packets for delay-sensitive applications [103, 105].

The remainder of the chapter is organized as follows: Section 2.2 describes network delay components briefly; Section 2.3 gives an overview of existing network delay control schemes; and lastly, the literature survey is summarized and discussed in Section 2.4.

2.2 What is Network Delay?

Network delay refers to the time delay experienced by a packet that travels from a source to a destination. Network delay metrics are foundation of many other metrics measurements, such as bandwidth, jitter, and packet loss [173].

Network delay comprises of stochastic delay components and deterministic delay components (Fig. 2.1). The deterministic delay components (e.g. propagation delay, transmission) are normally not run-time tuneable. Magnitudes of these components vary based on hardware design, medium type, transceiver capacity,

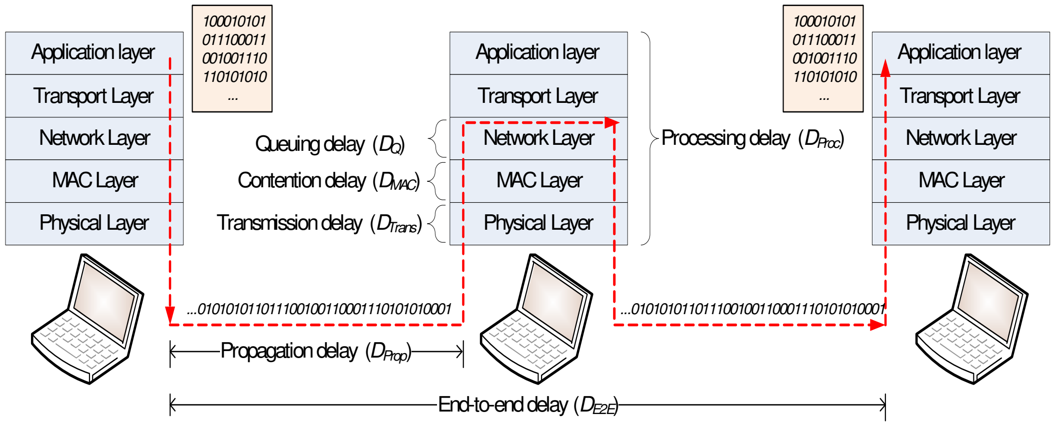


Figure 2.1: Network Delay Components

distance between nodes, packet size, etc. These delay components could be easily derived or estimated, whereas the stochastic delay components (e.g. queuing delay, processing delay, MAC contention delay) are harder to be estimated as there are many factors that may cause the variation. Definitions of each of the delay components [103] are listed as below:

- **Contention Delay,** D_{Con}: Contention delay refers to the time taken for a wireless node to gain access to a shared physical communication channel for a packet transmission. Each node needs to arbitrate for the wireless channel access before it can start transmitting a packet. The node backoff and arbitrates for the channel again if collision occurs. Thus, contention delay is the interval time between the time that a packet is at head-of-line for transmission and the time that the packet actually starts to be transmitted out.

- **Transmission Delay,** D_{Trans}: Transmission delay refers to the time taken to push a packet onto a communication link. The transmission delay is proportional to the length of a packet. It is calculated as packet length in bits (N) divided by the transmission rate (R) of the network interface (Eq. 2.1).

$$D_{Trans} = \frac{N}{R} \tag{2.1}$$

- **Propagation Delay,** D_{Prop}: Propagation delay refers to the time taken for a packet to be transmitted or propagated over the communication medium. Propagation delay is calculated as the distance between two nodes (d) divided by the propagation speed of the medium (s) (Eq. 2.2).

$$D_{Prop} = \frac{d}{s} \tag{2.2}$$

- **Queuing Delay,** D_Q: Queuing delay refers to the waiting time for a packet in a queue before it is being processed and transmitted out. Traffic intensity, congestion, system throughput, processing load and processing speed are major causes of queuing delay.

- **Processing Delay,** D_{Proc}: Processing delay refers to the time taken by a node to process a packet. Packet processing may be incurred at all layers. Processing delay may vary according to the processing load and the processor speed of the node. Examples of packet processing are data encoding, data compression, packet header encapsulation, packetization, encryption, and route lookup.

Nodal delay (Eq. 2.3) is typically known as a single hop delay; it is the sum of all delay components discussed above at a node. While end-to-end delay (Eq. 2.4) or one-way delay (OWD) refers to the time taken for a packet to be transmitted across a network from a source to a destination, which is the aggregation of nodal delay along its path to the destination.

$$D_{Nodal} = D_Q + D_{Proc} + D_{Prop} + D_{Trans} + D_{Con} \tag{2.3}$$

$$D_{E2E} = \sum_{i=1}^{n} D_{Nodal}(i),\ n = number\,of\,hops \tag{2.4}$$

2.3 Network Delay Control Schemes

This section provides a brief review on network delay control mechanisms from various QoS control categories. All QoS control mechanisms discussed in this section contribute to network delay control either implicitly or explicitly.

2.3.1 Admission Control

Admission control is used to admit or reject flows into networks based on the link status and resources available along the path. Admission control ensures the newly admitted flows will not deteriorate the QoS of the existing flows and at the same time the QoS for the newly admitted flow is guaranteed.

Some researchers [167, 183] state that admission control is not effective in wireless networks especially for mobile ad hoc networks (MANETs). Links are setup and tore down dynamically in such networks and the link quality varies in different time scales. Admission of a new flow may cause other nodes to violate their QoS requirements such as packet delay. This is owed to wireless nodes share a common medium for packet transmission. However, some researchers [7, 93, 94, 121, 151, 164] argue that admission control is important to provide service differentiation. Ahmed *et al.* [5] emphasize that admission control is important to guarantee QoS parameters such as signal quality and packet-level QoS parameters (packet delay and throughput) in wireless networks. This is because the more loaded a network is, then the more deteriorated is the signal quality resulting from interference. Admission control can prevent excessive packet loss and long packet delay resulting from congestion and collision triggered by over-subscription of bandwidth.

Various admission control schemes have been designed to take care of wireless characteristics. The admission control decision for wireless networks normally involves neighbour nodes (e.g. within carrier sensing range) and MAC layer information. CACP (Contention-aware Admission Control Protocol) [185] is a contention-aware admission control scheme for IEEE 802.11 wireless networks, the admission control decision is made by estimating locally available resources of itself and knowing available resources of neighbour nodes within its carrier sensing range (c-neighbours). Available bandwidth is estimated via measurement of channel utilization. Whereas available bandwidth of neighbour nodes can be obtained by querying c-neighbours or through monitoring the wireless medium to detect if c-neighbours are idle. When a node receives a query message, the node checks its available bandwidth to determine whether the flow can be admitted. If the bandwidth is not sufficient, a message is sent back to the source to reject the flow.

PAC (Perceptive Admission Control) [29] is also a contention-aware admission control scheme. Available bandwidth is estimated via measurement of channel utilization for an extended carrier sensing range. The range of measurement is increased to cover the minimum distance required for two simultaneous transmissions to occur without a collision. Therefore, this enables a node to make the admission decision without querying other nodes. Each source monitors the available bandwidth continuously to ensure that the bandwidth is sufficient to support active traffic flows. If the available bandwidth drops below a threshold, then the source throttles or stops the transmission. The source can then attempt to re-admit the flow after a randomly selected backoff time.

MACMAN (Multi-path Admission Control for Mobile Ad hoc Networks) [119] estimates available bandwidth via channel utilization measurement. It builds on CACP and PAC with an additional support of multi-path routing. Multiple paths that can support the required QoS between a source and a destination are discovered and maintained. This enables the source to switch to an alternate path if the existing path becomes unusable.

CACR (Call Admission and Rate Control) [192] and ACA (Admission Control Algorithm) [157] are approaches that make admission control decision based on channel busyness ratio. ACA takes care of hidden terminal issue that was omitted in CACR for available bandwidth estimation. CACR mainly focuses on wireless local area networks (WLAN), while ACA focuses on multi-hop wireless ad hoc networks. The channel busyness ratio is derived from the average time of a successful packet transmission and the average time of collisions based on the probability of a time slot in idle state, successful transmission state and collision state. The bandwidth requirement of a real-time flow is converted into channel utilization that the flow will occupy. Checking of aggregated channel utilization against the estimated available bandwidth is then carried out to admit or reject that real-time flow. CACR and ACA also control sending rates of non real-time flows to prevent network congestion.

Other than MAC layer approach, some researchers use probing approaches to estimate end-to-end available bandwidth [7,41,52,94,155] and end-to-end delay [27] for decision-making of admission control. However, there are some drawbacks, such as additional delay for a call setup is introduced by the probing process and congestion caused by the probing process, for probing approaches [186]. To understand how a probing approach works, an example of probing method is described here. In [7], UDP control packets are used to check on available bandwidth along a transmission path. Each intermediate node compares its available bandwidth with the requested bandwidth and updates the bandwidth field in UDP control packets if the available bandwidth is lower. When UDP control packets reach the destination, the minimum available bandwidth along the path is extracted and relayed back to the source for decision-making.

Yasukawa *et al.* [186] propose to use a delay estimation instead of a bandwidth estimation or a probing approach for decision-making. The proposed scheme is targeted for a WLAN. Wireless nodes in a WLAN monitor the shared medium for frequency of idle times and calculate the time between idle times (TBIT) of an access point to estimate queuing delay of the access point. The authors show that delays experienced by a real-time flow can be limited if a network is not congested as a result of a proper admission control decision. Wireless nodes check if the frequency of idle times observed from TBIT mechanism is higher than the packet rate of a new real-time flow before admitting the flow. This is to ensure there is room for packet transmission from the new flow without causing congestion.

Table 2.1: Summary of Network Delay Control Schemes (Part 1)

QoS Schemes	Goals	Approach	Delay Control
Admission control	• Prevent over-subscription of bandwidth • Prevent network congestion • Guarantee QoS of admitted traffic flows	Admit or reject traffic flows based on: • Bandwidth estimation via channel utilization [29, 119, 185] • Channel busyness ratio [157, 192] • Bandwidth estimation via probing approach [7, 41, 52, 94, 155] • Delay estimation via probing approach [27] • Delay estimation via channel idle time monitoring [186]	• Preventive approach • Contribute to network delay control implicitly • No constraint on queuing delay or MAC layer delay • Minimize overall network delay through preventing network congestion and collisions

Table 2.1 gives a brief summary of admission control schemes reviewed. The effectiveness and efficiency of admission control schemes rely on the accuracy of available resources estimation. These schemes play a preventive role in network delay control. Network resources are mostly not reserved explicitly in admission control schemes, but controlling the amount of traffic admitted into a network can mitigate network congestion problem. Network delay is minimized as a result of optimum utilization of bandwidth available. If such an optimum utilization cannot be achieved or maintained due to network dynamics, a network may become congested. Network congestion causes packet loss, larger queuing delay and larger MAC layer delay. There is no explicit control mechanism in these schemes to constrain the upper bound of nodal delay or end-to-end delay.

2.3.2 Bandwidth Reservation / Allocation

Bandwidth allocation can be grouped into two broad approaches: static allocation and dynamic allocation [160]. Bandwidths are reserved along transmission paths and dedicated to real-time traffic flows to prevent QoS deterioration. For static allocation, bandwidth reserved for a particular traffic flow cannot be used by other traffic flows before the reservation is released. While for dynamic allocation, bandwidth is not reserved per traffic flow. Bandwidth is allocated adaptively for aggregated traffic flows to achieve the required QoS.

Signalling is normally used in a bandwidth reservation process. A signalling mechanism is responsible for establishment, maintenance, and termination of a connection for an application to reserve required resources. Signalling protocols can be divided into two types: out-of-band signalling and in-band signalling. For out-of-band signalling protocols, control information is carried in a separate control packet. While for in-band signalling protocols, control information is carried along with a data packet. Out-of-band signalling introduces higher overhead to wireless ad hoc networks as the bandwidth in such networks is scarce. Control packets may compete with data packets for bandwidth and cause congestion and collision in networks. Traditionally, RSVP (Resource ReserVation Protocol) [23] is used in IntServ networks to reserve per-flow bandwidth. However, RSVP signalling overhead is too high for wireless ad hoc networks and RSVP cannot adapt to network dynamics in wireless ad hoc networks [143,147]. A few signalling protocols, such as INSIGNIA [107], dRSVP (dynamic RSVP) [127] and MRSVP [165], have been proposed to adapt to network dynamics of wireless ad hoc networks.

RSVP is an out-of-band signalling protocol. It enables an application to request for different level of QoS support via reservation of network resources along transmission paths for their data flows. A flow specification is used to communicate QoS requirements of the application during the reservation process. The flow specification consists of a service class, a reservation specification that defines the QoS and a traffic specification that describes the data flow. The reservation process is initiated by a source via sending a RSVP path message through pre-established route to a destination. When the destination receives the path message, reservation is made based on the requested parameters. Admission control and policy control are used to determine whether a node has sufficient available resources to fulfil the requested QoS. An error notification is returned to the source if reservation failed. If the reservation is granted, the destination sends a RSVP reservation message specifying the flow specification back to the source along the reverse data path. Each node in the path can either accept or reject the request. Once the source receives the reservation message, it can start sending data packets. Reservation at each node is maintained via a soft state table, the reservation soft state needs to be refreshed periodically by path and reservation request messages. Otherwise, the soft-state automatically times out and the reservation is released.

dRSVP [127] is a variant of RSVP that is designed for wireless ad hoc networks by considering the factors of variable application demands and wireless network characteristics, such as variable link characteristics and node mobility. A reservation is specified as a range of values (minimum and maximum) instead of a single point value to create the flexibility needed to deal with factors above. When the topology changes or available resources change, the updated available resources are discovered. This allows the QoS level to be adjusted accordingly within the reservation range. A few modifications have been made to RSVP in order to support the dynamic adaptation feature of dRSVP, such as: an additional flow specification in reservation messages, an additional traffic specification in path messages to allow a description of the specification in a range, a measurement specification is added to the reservation message and a new reservation notification message is introduced to enable nodes to learn about the resource bottleneck at downstream and upstream. Admission control is modified to deal with the bandwidth range admission, and a bandwidth allocation algorithm is introduced to divide the available bandwidth among admitted flows.

MRSVP is another variant of RSVP that supports mobility in wireless ad hoc networks. To overcome the hand off impact of mobility, Talukdar *et al.* [165] propose to make advance reservation from a set of locations where a mobile node might visit in near future. A mobility specification is used to record the set of locations to be visited. The protocol supports two types of reservation: active and passive. An active reservation is made from the current location of a mobile node, while passive reservation is made from the locations set in the mobility specification. MRSVP requires local proxy agents and remote proxy agents for the reservation. Proxy agents make reservation along the paths from locations in the mobility specification of a sender to locations in the mobility specification of a receiver. When a mobile node moves to a new location, MRSVP changes the state of passive reservation of the new location into an active state and alters the original active reservation to a passive state.

INSIGNIA [107] is an in-band signalling protocol that makes use of OPTIONS field of IP (Internet Protocol) header to carry control information, such as service mode, payload type, bandwidth indicator and bandwidth requested, for operations of resource reservation and adaptation. It supports fast reservation, restoration and QoS adaptation. Fast reservation is carried out by sending a reservation request from a source with the requested bandwidth and service embedded in the IP option field of data packets to its destination. Admission control is carried out at intermediate nodes to determine if the reservation request can be supported. The decision of admission control is reflected in the IP option field to be carried along to next hops. Reservation is granted if there are sufficient resources; otherwise packets are treated as best-effort packets. This process is repeated on a hop-by-hop basis until the reservation request reaches the destination. If the end-to-end reservation is successful, the source changes its internal state from best-effort to reserved. When there is a route failure resulting from host mobility, INSIGNIA reestablishes reservation quickly via a flow restoration operation that involves rerouting packets, admission control and resource reservation at the new path. The destination node actively monitors the QoS of active flows and reports the status back to the source periodically so that the source can respond to network conditions by adjusting its transmission rate.

Besides signalling-based bandwidth reservation schemes, some researchers focus on MAC layer approach [4, 113, 115, 116, 124] to reserve bandwidth in wireless networks. One of the weaknesses of MAC layer approaches for bandwidth reservation is lack of interoperability between different platforms. This approach requires all nodes implement the same MAC layer. A few examples of MAC layer approach are described subsequently.

RT-MAC (Real-Time MAC) [124] is a MAC layer bandwidth reservation approach that builds on the IEEE 802.11 DCF (Distributed Coordination Function) standard. It allocates free slots in a superframe for transmission of real-time data from neighbour nodes upon receiving the reservation request. Three-way handshaking is used for the reservation process. The reservation process can be explained as following three steps: 1) Node A sends a request to node B with reservation and timing information 2) Upon receiving the request at node B, node B checks its reservation table and replies to Node A if reservation is permitted. 3) Node A sends acknowledgement to Node B after receiving the reply from Bode B.

The free slots in node B's superframe are then reserved for the traffic flow from node A. All node B's neighbours update their reservation table upon overhearing request granted message and the same for node A's neighbours when they overhear the acknowledgement message.

MACA/PR (Multi-hop Access Collision Avoidance with Piggyback Reservations) [115, 116] is also a MAC layer bandwidth reservation scheme that builds on the IEEE 802.11 standard. The first data packet of a real-time traffic flow is used to reserve MAC bandwidth along a path. The source initiates a RTS–CTS (Ready to Send - Clear To Send) dialog to reserve bandwidth for the real-time flow. When the receiver replies with CTS, the source can start to transmit data. Reservation information is piggybacked in the data packet. The receiver acknowledges the data packet received if it received the data packet correctly and records the reservation into its reservation table. The next transmission time is carried in data packets and acknowledgement packets as a way to inform their neighbours who can overhear the packets to avoid collision. Each node maintains a reservation table to keep track of reserved windows of neighbour nodes within its range.

A synchronous type MAC layer approach is also used for bandwidth reservation, such as TDMA (Time Division Multiple Access) [113]. For the TDMA-based approach, the bandwidth requirement is realized by reserving time slots on a link. Bandwidth is divided into a limited number of time slots. Each node can only transmit data during the time slots allocated to it. The nodes need to synchronize on time and agree on the time slot allocation. Time slot scheduling for nodes is carried out during route discovery phase. Each node maintains a time-slot allocation table for the purpose of reservation. However, TDMA approach incurs high overhead for time synchronization [124].

Dynamic bandwidth management (dBM) [4] is another bandwidth reservation scheme targeted for IEEE 802.11 ad hoc networks. Each node tracks the required bandwidth and transmission rate of neighbours that are one and two hops away within the same contention region. The decision-making for the bandwidth reservation of a particular session in a network is based on the available bandwidth estimated from the bandwidth required by neighbour nodes in the same contention region. The bandwidth reservation is rejected if the available bandwidth is insufficient. The bandwidth allocation information is then distributed to all nodes within the same contention region. dBM regulates the packet transmission of admitted flows according to a token bucket mechanism to prevent over-subscription of bandwidth.

Table 2.2: Summary of Network Delay Control Schemes (Part 2)

QoS Schemes	Goals	Approach	Delay Control
Bandwidth reservation / allocation	• Dedicate bandwidth for specific traffic flows to prevent QoS deterioration	Reserve or allocate bandwidth: • Via signalling approach to discover bandwidth available along the path [107, 127, 165] • Via hand-shaking approach and time slots reservation at MAC layer [113, 115, 116, 124] • Via tracking of bandwidth and transmission rate of neighbours [4] • Based on queue length, packet loss or network latency observed [92, 144, 160]	• Preventive approach • Contribute to network delay control implicitly • No constraint on queuing delay and MAC layer delay • Minimize overall network delay through bandwidth management; network delay is high when the bandwidth is over-subscribed

Adaptive bandwidth control (ABC) is used in dynamic bandwidth allocation approach to improve network QoS and to reduce wastage of bandwidth [160]. Depending on the QoS requirements and the goals to be achieved, different techniques are used for ABC. A closed-loop control technique is frequently used to control the chosen QoS metrics to the targets required by applications. Bandwidth allocation [92, 144, 160] at each node is adjusted adaptively based on queue length, packet loss or network latency observed.

Similarly to admission control scheme, bandwidth reservation or allocation schemes play a preventive role in network delay control. Table 2.2 gives a brief summary of bandwidth reservation schemes reviewed. There is no explicit control on network delay with this approach. Adopting bandwidth reservation or allocation approaches to control QoS may be difficult for wireless ad hoc networks. The allocation of bandwidth at each node may impact neighbour nodes since the nodes in the networks share the same medium and the available bandwidth of each node varies over time. A global coordination is needed in order to reserve bandwidth efficiently. The bandwidth reservation may be violated due to network dynamics in networks [175]. Issues such as bandwidth fluctuation, network dynamics and scalability remained as open issues and need further attention in order to reserve bandwidth efficiently and maintain the target QoS effectively.

2.3.3 QoS-aware Routing

Routing is a process of selecting transmission paths in a network to send data packets from a source to a destination through intermediate nodes. Routing mechanisms in wireless ad hoc networks [148] can be divided into two major categories: 1) proactive approach and 2) reactive approach. The proactive approach is also known as table driven approach. For routing protocols fall under this category, each node needs to maintain one or more tables to store routing information. Consistent and up-to-date routing information in response to network topology changes is propagated to other nodes in the network in order to maintain a consistent network view. Examples of proactive approaches are DSDV (Destination-Sequenced Distance-Vector), CGSR (Clusterhead Gateway Switch Routing) and WRP (Wireless Routing Protocol). While for the reactive approach, routes are created only when desired by a source node. When a node requires a route to a destination, a route discovery process is initiated. After the route is established, it is maintained until either the destination becomes inaccessible or until the route is no longer needed. TORA (Temporally-Ordered Routing Algorithm), AODV (Ad hoc On-Demand Distance Vector) and DSR (Dynamic Source Routing) are examples of reactive routing protocols. Routing protocols such as DSDV, AODV, DSR and TORA are best-effort routing protocols. These protocols become the foundation of QoS-aware routing protocols for wireless ad hoc networks.

Best-effort routing protocols use minimum hop count as the routing metric for route setup. However, a path with a minimum hop count does not necessarily give the shortest delay and the best QoS. Therefore, these protocols are not suitable for applications with specific QoS constraint requirements. QoS-aware routing protocols, which based on routing metrics such as delay, bandwidth, jitter and packet loss, are used to select a path that may fulfil the QoS requirements of applications. According to RFC2386 [39], QoS-aware routing is defined as a routing mechanism that uses some knowledge of resource availability in a network as well as the QoS requirement of the flows in decision-making for path selection.

From the surveys on the QoS-aware routing protocols [17, 70], routing metrics commonly used in MANET routing protocols are: 1) minimum required throughput or capacity, 2) maximum tolerable delay, 3) maximum tolerable delay jitter, and 4) maximum tolerable packet loss ratio. The routing metrics can be associated with QoS metrics in different protocol stack layer. Some of the routing protocols are based on network layer metrics, such as achievable throughput or residual capacity, end-to-end delay, node buffer space, delay jitter and packet loss ratio. There are routing protocols that based on link and MAC layer metrics, such as MAC delay, packet delivery ratio, link stability, and node relative mobility. There are also routing protocols that based on physical layer metrics, such as signal-to-interference ratio, bit error rate and node residual battery charge. Only delay-based QoS-aware routing protocols [15, 20, 21, 32, 82, 122, 128, 131, 133, 142, 152, 158, 176, 181, 190] are discussed below.

Chen *et al.* [32] has proposed a QoS-aware routing protocol that establishes routes based on end-to-end delay requirements. A source sends out several probes to its neighbour nodes to discover a feasible route that satisfies the delay constraint. Each probe is responsible to accumulate delay of the path it has traversed towards a destination from a source. Each probe contains at least one ticket. Each intermediate node updates the delay field carrying in the probe packet by adding the link delay experienced between itself and the previous hop. Intermediate nodes may split a single probe into multiple probes and distribute the received tickets among these new probes. The new probes can then be used to search for a different downstream subpaths. The ticket is invalidated if the accumulated delay along the path violates the delay requirement. If multiple probes arrive at the destination node with a valid ticket, the least cost path is selected as the primary route and other paths as backup routes to be used when the primary route is broken. Queuing delay and processing delay at each intermediate node are not taken into account during the path discovery, only link delay is accumulated. Therefore, this becomes a drawback of this approach. This may cause some data packets to miss their deadlines if the network is highly loaded or congested.

AQOR (Ad hoc Qos On-demand Routing) [181] is a QoS-aware routing protocol for MANET. AQOR discovers the best available route that has the smallest end-to-end delay with a bandwidth guarantee. A route request packet that carries the requested bandwidth and the end-to-end delay constraint is sent out via flooding communication to its next hop. When an intermediate node receives the route request packet, it rebroadcasts the route request to its next hop only if the bandwidth requested can be fulfilled and the delay constraint is not violated. Since the flooding approach is used, there might be multiple request packets arrive at a destination node. The destination node will send back a reply packet along each of these routes. When the source node receives the route replies, the route with the least delay is chosen by the source. The delay estimation at the source node is based on round trip time (RTT) of the route discovery process. One way delay is assumed to be symmetric in this proposal. However, this assumption may not hold in MANET. Therefore, the end-to-end delay might not be satisfied in actual case in addition to delay variation caused by network dynamics.

Perkin and Royer [142] have extended AODV to provide QoS by including delay and bandwidth as routing metrics for the route discovery process. It is known as QoS-AODV. During the route discovery process, the requested maximum delay is added into a route request packet. While the route request is broadcasted and forwarded to reach its destination, intermediate nodes subtract NODE_TRAVERSAL_TIME (processing time for the route request packet) from the maximum delay. Queuing delay and MAC layer contention delay are not considered in the route discovery process of this approach. A route is established if the route request reaches the destination with the remaining time greater than zero.

EDC-AODV (Energy and Delay-Constrained AODV) [152] is an extension of AODV to consider residual energy and current queue size in the route discovery process. These two parameters are included in the cost function to discover routes. When a network is congested, queuing delay and MAC contention delay are large. The approach aims to avoid congested nodes in a route based on the backlogs in the queue. This approach prolongs system lifetimes and minimizes the end-to-end delay as congested routes are avoided.

Both DOSPR (Delay-Oriented Shortest Path Routing) [158] and AODV-D (AODV-Delay) [122] use the same mechanism in estimating forwarding delay for route selection. Forwarding delay is estimated based on the timing and backoff mechanism in IEEE802.11, therefore only MAC layer delay is considered. Queuing delay is omitted in the route selection process. DOSPR is a proactive approach, whereas AODV-D is a reactive approach. For AODV-D, a route request is flooded out with the maximum delay requirement. Intermediate nodes only forward the route request packet until reaches its destination if the accumulated forwarding delay along the path has not exceeded the required maximum delay. AODV-D tries to maintain end-to-end delay by monitoring the end-to-end delay of packets received at the destination; alternate paths are selected if the delay constraint is violated.

DA-AODV (Delay Aware AODV) [20] is another extension of AODV that takes the delay requirement from applications. During the route discovery process, the accumulated delay along the path from a source to a destination is recorded in the routing table of each node. When an application requests a route to a destination; the delay requirement of the application is compared to the delay recorded in the routing table to check if such a route exists. The route will be selected if the delay requirement is fulfilled. DA-AODV has been extended to include multi-path support and the extension is named DAAM (Delay Aware AODV-Multi-path) [21].

Asokan and Natarajan [15] have proposed a similar extension to AODV and DSR, namely, EDAODV (Energy and Delay aware AODV) and EDDSR (Energy and Delay aware DSR), to consider residual energy and maximum delay constraints. A minimum energy field and a maximum delay field are added into route request packets so that a route that satisfies the maximum delay permitted between a source and a destination can be established. If the remaining energy of a node is less than the required minimum energy or the cumulative delay estimation is greater than the maximum delay, the route request will be dropped.

RTD-DSR (Real Time DSR Protocol with Delay constraints) [82] is an extension to DSR routing protocol that includes delay constraint and admission control process to satisfy the guaranteed QoS requirement of real-time traffic. IntServ principle is adopted for admission control and resources reservation in this protocol to ensure that a newly admitted flow will not cause QoS degradation for existing flows. The delay constraint is checked during the route discovery process. The expiration delay to the deadline specified by the real-time flow is checked. The remaining time of expiration delay is updated at each intermediate node by subtracting transmission delay and processing delay (including queuing delay) along the path. The new flow will be admitted if the route request packet reaches the destination before the deadline and the delay constraints of existing flows will not be violated after the admission of that flow. The protocol checks the validity of paths and satisfaction of delay constraints with a prediction of topology changes to address the mobility issue in MANET.

QOLSR (QoS Optimized Link State Routing) [128] is an extension to OLSR with QoS constraints for delay and bandwidth. The delay metric used for routing table computation is measured by comparing the receiving time of a HELLO message to the creation time of the HELLO message. Therefore, the measured delay includes queuing delay, MAC contention delay and propagation delay. However, this approach requires time synchronization in the network to measure the delay between nodes accurately. Routes selection is carried out based on the delay metric and bandwidth metric calculated to get a path that has the most available bandwidth and a shorter delay.

Nagarajan *et al.* [133] propose to consider link delay metric (media access delay) for a selection of routes based on OLSR routing protocol. The routing table calculation of OLSR is adapted to include link delay metric so that only routes with good delay characteristics are selected. Media access delay is obtained from the average of nodal delay measurement experienced by packets being transmitted at a node. The average media access delay of each link to neighbour nodes is determined and being advertised to the neighbour nodes for routing tables computation. Only links with media access delay less than a specified threshold are considered in the computation. Therefore, end-to-end delay is minimized by using this approach.

LDAR (Link Delay-aware Routing) [131] is a delay-based routing protocol for multi-hop wireless ad hoc networks. LDAR is built on OLSR protocol with the extension of expected transmission count as a routing metric. Delay estimation is carried out to estimate the link quality so that a decision on route selection can be made. Delay experienced by a node (transmission delay, processing delay and queuing delay) is estimated via real-time measurement at driver level with a consideration of different link data rates. The delay estimated is dispersed to one and two hops away neighbour nodes for route computations to reach other nodes in the network. From this approach, routes with a shorter link delay can be discovered and being selected to relay delay-sensitive traffic.

AAQR [176] is an application-aware QoS routing that is targeted for multimedia applications in MANET. Routes are discovered on demand and selected based on the QoS constraint specified by the applications. The proposed solution estimates transmission delay between two nodes by obtaining the difference between the transmission timestamp of a RTCP (Real-time Transport Control Protocol) packet and the receiving timestamp of the RTCP packet. The proposed scheme assumes that RTP (Real-time Transport Protocol) is used to carry voice and video traffic. Besides the transmission delay estimation and available bandwidth estimation, the variance of transmission time is calculated and included in the selection of routes.

Table 2.3: Summary of Network Delay Control Schemes (Part 3)

QoS Schemes	Goals	Approach	Delay Control
QoS-aware routing (delay-based)	• Discover a single or multiple routes that can satisfy end-to-end requirements • Establish routes • Maintain routes	Routes selection based on: • End-to-end delay estimation via probing method [32, 181] • Aggregation of processing time for route request packets [142] • Residual energy, current queue size, delay constraint [15, 152] • Forwarding delay estimated from the timing and back off mechanism in IEEE 802.11 [122, 158] • Tracking of accumulated delay or the remaining time of expiration delay [20, 21, 82] • Delay measurement from beacons or packets transmitted [128, 131, 133, 176]	• Reactive approach • Contribute to network delay control implicitly • No constraint on end-to-end delay and nodal delay • Control end-to-end delay of traffic flows via an active route discovery and a route maintenance

Table 2.3 gives a brief summary of delay-based QoS-aware routing protocols reviewed. The delay-based routing protocols can contribute to network delay control by choosing routes that can satisfy the end-to-end delay requirements of traffic flows. However, end-to-end delay may not be guaranteed. Most of the delay-based routing protocols discussed above are implemented without resource reservation and routes are established based on delay estimation. End-to-end delay and nodal delay for routes are not bounded. The delay estimation may become invalid due to time-varying link quality and interference. Consequently, frequent rerouting may be required to overcome the issue. If nodal delay can be bounded, then violation of delay requirements can be avoided. Hence, the effectiveness and efficiency of routing protocols can be improved.

2.3.4 MAC Layer Approach

A lot of QoS provisioning schemes for multi-hop wireless ad hoc networks revolve around MAC layer. These schemes either gather information from MAC layer to achieve their goal or modify MAC layer to overcome weaknesses of wireless network characteristics. The schemes, such as admission control and bandwidth reservation, probe the MAC layer to gather information on channel utilization for available bandwidth estimation. This section gives a brief overview of approaches that require to tweak MAC layer parameters or modify MAC layer to control delay. Bandwidth sharing and collision are the main focus of these schemes.

Many of the QoS-aware MAC schemes proposed are based on the IEEE 802.11 DCF standard [64,65,101,183]. MAC parameters used in controlling packets transmission and contention access are tweaked in these proposals to fulfil the requirements of real-time traffic in terms of bandwidth and delay. For such approaches, different inter-frame spacing, different backoff contention value and different contention window value are assigned to traffic from different priority classes. One of the proposals is the IEEE 802.11e EDCA (Enhanced Distributed Channel Access) standard [79,154]. It is an extension to the IEEE 802.11 DCF standard. It divides traffic flows into eight priority levels, higher priority traffic flows are assigned with a shorter AIFS (Arbitration Inter Frame Spacing) and a smaller backoff contention window. This enables the higher priority traffic flows to gain medium access faster than lower priority traffic flows.

Ge and Li [64] propose to use different values for contention parameters for real-time traffic and best-effort traffic. The proposal is similar to IEEE 802.11e but it takes the impact of congestion into consideration for random backoff mechanism. Higher priority traffic uses a smaller contention window value in order to have a higher chance in gaining medium access. The backoff mechanism of each node is adjusted according to the sensed network load; the congestion level is reflected by the collision probability that is computed periodically.

AMP (Adaptive QoS MAC protocol) [65] is a MAC protocol that builds on the IEEE 802.11 DCF standard. It aims to provide a better QoS for higher priority traffic. The nodes inside a network compete for channel access adaptively according to the network load. The network load is estimated at each node based on the packet loss rate over a sampling period. The contention parameters for the backoff mechanism and the waiting time for medium idle are adjusted dynamically based on the network load for different priorities traffic. Higher priority traffic gets a higher chance to access the medium from the adaptive mechanism. Nodes that encounter high packet loss rate are held off from transmission and contention for medium access for a certain period to reduce the network load. High priority traffic has a higher threshold of packet loss rate before giving up the participation in channel contention.

DDA (Distributed Delay Allocation) [183] is an IEEE 802.11 MAC based delay control algorithm that targets to provide average delay guarantee to real-time multimedia applications in wireless ad hoc networks. The end-to-end delay requirements are broken down to per-hop delay (average queuing delay, average transmission delay and contention delay). In this approach, contention delay is deemed as the dominant per-hop delay component. The MAC contention window size of a relaying node is adapted dynamically based on an average contention delay. Average queuing delay is assumed to be bounded by bounding the packet service rate that is controllable at node by adapting the contention window size.

QMA (QoS-based Multiple Access) [172] is another contention based MAC protocol that divides the medium access into contention phase and transmission phase. The wireless nodes compete for medium access by broadcasting forecast bursts before packet transmission. The node with the most forecast bursts will win the channel. The number of forecast bursts is calculated based on the priority and the deadline of packets. Consequently, packets with an earlier deadline and a higher priority have a larger forecast bursts. This approach reduces the possibility of collision to support QoS guarantee for the real-time traffic flows.

For most of the approaches discussed above, delay-sensitive traffic is given higher priority through adaptation of MAC layer parameters to gain access to the medium so that MAC contention delay is shortened. Hence, end-to-end delay experienced is minimized. However, these approaches lack in per-hop delay constraint, upper layer delay such as queuing delay is ignored. MAC layer does not have insight of queue backlogs. Therefore, nodal delay could not be bounded. Furthermore, tweaking MAC layer may require co-operation of other nodes in a network for its effectiveness and may require modification to MAC firmware or hardware for the implementation. Modification of MAC layer may not cost effective and require more effort to realize it. Interoperability and legacy issues may also hinder the effectiveness of this approach.

Some other MAC layer approaches aim to reduce network latency by avoiding collisions in networks [85, 101, 161, 163, 170]. Collisions could be avoided by having a sleep scheduling at MAC layer, each node only wakes up at its scheduled time slot to receive and transmit packets. Collisions could also be avoided by using different radio coding techniques such as TDMA, FDMA (Frequency Division Multiple Access) and CDMA (Code Division Multiple Access). For TDMA approach, optimal TDMA schedules need to be designed for slots assignment by taking care of bandwidth sharing among nodes and delay requirements of traffic flows. Specific time slots are assigned to each node for data transmission. The bandwidth is considered wasted if a node does not have data to be sent out when reaches its turn. Time synchronization is needed among nodes for TDMA scheduling. For FDMA approach, the frequency channel is split into several frequency sub-channels. Each node can transmit simultaneously with this approach. Nodes are assigned to different radio frequency sub-channels to avoid collisions. However, the nodes have to be equipped with complex radio systems to use such approach. While for CDMA approach, a different code sequence is assigned to each node for communication. This enables simultaneous transmission with minimal interference. However, CDMA approach imposes heavy computation and requires a lot of memory to store the code sequences for all nodes. Besides that, TDMA and CDMA schemes are difficult to be implemented in wireless ad hoc networks due to lack of centralized control and network dynamics [70].

Other than aforementioned approaches, some researchers [47, 96] propose to adjust transmission rate and transmission power to minimize end-to-end delay and to improve QoS. Kim *et al.* [96] propose to use a mathematical model to calculate an optimized achievable data rate for IEEE802.11 networks. The transmission rate of MAC can be changed adaptively based on the data rate calculated with respect to end-to-end delay constraint. In this way, end-to-end delay is minimized and energy can be conserved as different transmission power is needed for different transmission rate. Dogahe and Murthy [47] have stated that queuing delay is a dominant component delay in wireless network. Therefore, they propose to use a network utility function based on the average delay derived from queuing model to adjust the transmission rate and transmission power in order to meet the average queuing delay constraint. The adaptation of transmission rate and transmission power may not be scalable as both parameters may affect the contention level and interference level in the network and cause performance variation to other nodes.

Table 2.4 gives a brief summary of MAC layer approaches reviewed.

Table 2.4: Summary of Network Delay Control Schemes (Part 4)

QoS Schemes	Goals	Approach	Delay Control
MAC layer approaches	• Overcome weaknesses of wireless network characteristics to improve network performance especially network delay	Control medium access contention or packet transmission by: • Tweaking parameters of IEEE 802.11 MAC [64, 65, 79, 101, 154, 183], such as assigning different inter-frame spacing, different backoff contention value, different contention window value to traffic from different traffic classes. Higher priority traffic is prioritized to have higher chances of channel access • Broadcasting forecast bursts [172], higher priority traffic is assigned with larger forecast bursts so get higher chance to access the medium • Using scheduling approach and different radio coding techniques (such as TDMA, FDMA and CDMA) to avoid collisions [85, 101, 161, 163, 170] • Adjusting transmission rate and transmission power at MAC layer [47, 96]	• Control MAC layer delay • Delay-sensitive traffic is given higher priority so that the MAC contention delay is minimized • No bound on MAC layer delay for contention-based MAC • Lack of control on upper layer delay

2.3.5 Buffer Management / Active Queue Management

Queue management is a mechanism to decide whether to enqueue or drop incoming packets based on the current queue state or QoS constraints. Congestion may occur if traffic arrives into network nodes is faster and more than the network nodes can handle. Hence, queuing delay is large and packets may be dropped. Queuing delay is proportional to backlogs of a queue in a network node, the longer the backlogs in a queue, the longer the waiting time for a packet resides in the node. Therefore, a proper queue management may constrain queuing delay. Queue management schemes for both wired and wireless networks domains are surveyed here as most of the existing queue management schemes in wireless ad hoc networks are based on the queue management approach in wired networks.

2.3.5.1 Wired Networks

Each router or network node has an interface queue (IFQ) that holds data packets scheduled to go out on that network interface. Before AQM is introduced, Drop Tail (DT) discipline is used to manage an interface queue. In DT discipline, data packets are dropped only if a queue is full. This results in large queuing delay if the queue size is large. AQM is a mechanism that actively managing a network queue to keep the average queuing delay low. The role of AQM in controlling network delay is very obvious by comparing DT and AQM schemes [24]. An AQM scheme manages a queue by dropping data packets probabilistically. Packet loss serves as an early congestion indicator to a source to enable the source to regulate its transmission rate. Most of the queue management schemes in wired networks target for delay-tolerant traffic (TCP traffic), but AQM plays an important role in controlling queuing delay. Therefore, AQM schemes are included in the review.

RED [61] is the most well known congestion avoidance mechanism for packet-switched networks that was recommended in RFC2309 [22]. RED detects congestion by estimating the average queue size and marks packets when the average queue size exceeds preset thresholds. RED is designed to accompany a transport-layer congestion control protocol such as TCP. Average queue size (Q_{AVG}) is calculated in RED using a low pass filter (exponentially weighted moving average (EWMA) approach). There are two thresholds for a RED queue, which are minimum threshold (TH_{MIN}) and maximum threshold (TH_{MAX}). Packets are marked or dropped probabilistically when the average queue size falls between TH_{MIN} and TH_{MAX}. If the Q_{AVG} is greater than TH_{MAX}, then all packets are marked or dropped.

The RED concept becomes the core of design for most of the AQM schemes introduced after that. Most researchers use the same concept as RED to drop or mark packets between minimum and maximum thresholds but tweaking the way of dropping or marking packets and the way of calculating dropping probability [9, 13, 55, 60, 117, 141, 195]. These AQM schemes are known as RED variants. FRED (Fair RED) [117] and BRED (Balanced RED) [13] aim to improve fairness among the traffic flows using the basic RED mechanism but maintain per flow state variables such as per flow queue threshold. DSRED (Double Slope RED) [195] maintains the same mechanism in RED but uses double slope of dropping probability to drop or mark packets fall between the minimum and the maximum thresholds of a queue. Another threshold (TH_{MID}) is introduced between the minimum and the maximum threshold to decide when to change the function of dropping probability slope. When congestion increases (exceeds threshold of TH_{MID}), the dropping rate is higher. Adaptive RED proposed by Feng *et al.* [55] adjusts the maximum dropping probability adaptively using multiplicative increase multiplicative decrease (MIMD) approach; while ARED (Adaptive RED) proposed by Floyd *et al.* [60] adjusts the maximum dropping probability using additive increase multiplicative decrease (AIMD) approach. CHOKe [141] is also based on RED. When the average queue length falls between the minimum and the maximum thresholds, a packet is dropped if the packet is from the same flow of previous packet. Otherwise, the packet is dropped based on the RED dropping probability. The other RED variant, which is proposed by Al-Raddady *et al.* [9], uses an adaptive dropping probability based on the target arrival rate and average arrival rate, instead of linear growing dropping probability.

There are also some AQM schemes [16,54,56,57,74,75,88,102,106] that are not RED-based. Hong *et al.* [75] propose an AQM algorithm that aims to achieve high utilization and low queuing delay. A target queue length is set in this approach. The average queue length is then being controlled via an adaptive dropping probability to match the target queue length to achieve the goals above. The proposed algorithm estimates the packet arrival and calculates the dropping probability based on the average queue length, the target queue length and the estimated packet arrival. Feng *et al.* [54, 57] propose an AQM scheme called BLUE. Packet loss and link idle events are used to manage congestion instead of queue length in this scheme. Packet dropping probability is increased if a queue continually experiences packet loss due to buffer overflow. As a result, a congestion notification is sent back to the source at higher rate. If the link is idle, the packet dropping probability will be decreased. SFB (Stochastic Fair BLUE) [56] is a variant of BLUE algorithm that incorporates a BLOOM filter to identify unresponsive flows and then rate-limit the unresponsive flows. It consists of BLUE algorithm that adaptively updating packet dropping probability based on packet loss and link idle events, and a Bloom filter that classifies traffic flows into different bins. The packet dropping probability is updated based on occupancy of the bins. The flow is non-responsive if the packet dropping probability reaches 1. Thus, the flow is being identified and rate-limit. GREEN [88] aims to provide fairness during congestion at edge router. The dropping probability is calculated based on throughput of incoming traffic and the number of active flows in a router. The throughput of incoming traffic is estimated through RTT estimation. The dropping probability is increased when the active flows increase or a flow has a shorter RTT, thus provides fairness to all traffic flows indirectly.

Some AQM schemes [16, 74, 102, 106] take control theoretic approach. PI (Proportional-Integral) controller introduced by Hollot *et al.* [74] is one of the well known control theoretic AQM approaches. In this approach, the queue length is regulated to match the target queue length. The dropping probability is updated periodically based on the queue length. The dropping probability is increased if the queue length is higher than the target queue length and is decreased otherwise. REM (Random Exponential Marking) [16] is an AQM scheme that drops packet based on a cost function. It aims to achieve high utilization besides maintaining low queuing delay. The cost function is used to determine the dropping probability. It is updated periodically (either increasing or decreasing) based on rate mismatch observed between packet arrivals and packet departures, and queue length mismatch observed between the actual queue length and the target queue length. When the mismatch is high, the cost increases and hence the dropping probability also increases. A rate-based queue management scheme, namely, AVQ (Adaptive Virtual Queue) has been proposed by Kunniyur *et al.* [102]. AVQ algorithm maintains a virtual queue with the queue size less than the actual queue capacity. The virtual queue size is calculated adaptively based on the desired utilization of the link against the arrival rate. The virtual queue size is inverse proportional to the arrival rate. Packets are marked or dropped when the virtual queue overflows. The marking or dropping becomes more aggressive when the link utilization exceeds the desired utilization. LQD (Loss and Queuing Delay control) [106] is an AQM scheme that calculates dropping probability based on a target queue length and a target loss rate. The dropping probability is increased when the queue length is larger than the target queue length and is decreased otherwise. The dropping probability is also decreased if the loss rate is higher than the target loss rate so that the packet loss rate is kept within the threshold.

Some AQM schemes [36, 37, 156, 191] use fuzzy logic approach. Chrysostomou *et al.* [36,37] propose to use a fuzzy logic controller to calculate marking probability for DiffServ traffic (best-effort and assured) based on the error of queue length (difference between a target queue length and an instantaneous queue length). Fuzzy-Green [191] is a modification version of GREEN queue management algorithm. A fuzzy logic controller is used to replace the formulas that are being used to calculate the adaptation parameter in GREEN algorithm. The adaptation parameter is meant to enable the algorithm adapting to uncertainty in the system. ADT-FL (Adaptive Drop Tail-Fuzzy Logic) [156] uses fuzzy logic to control the queue size adaptively based on traffic intensity and available bandwidth with a set of fuzzy rules.

All the AQM schemes discussed above manage queues by dropping packets probabilistically based on current queue states such as queue length and load, packet loss and link utilization. These approaches need to work with TCP protocols to adjust the sending rate accordingly so that congestion can be alleviated. For such approaches, lower queuing delay is promised but queuing delay is not bounded as there is no constraint of delay associated to these approaches. If these schemes are used for delay-sensitive traffic; delay could not be bounded as UDP traffic is non-responsive and the UDP protocol cannot adapt the sending rate. So queuing delay is only bounded by the maximum queue size.

There are only a few queue management schemes [8,68,174] focus on constraining network delay in terms of average queuing delay. Guan *et al.* [68] propose a queue management scheme that constrains average queuing delay to a specified value by adjusting the queuing threshold dynamically using analytical model results based on a single MMBP-2 arrival process. Average queuing delay experienced by a node is calculated periodically to compare with the target queuing delay required. The difference between the current average queuing delay and the target queuing delay is summed up to obtain next target average queuing delay. Based on the next target average queuing delay and the arrival rate; an analytical formula has been derived to map a queuing threshold into average queuing delay required at the next time step. Therefore, the queuing threshold is either increased if lower queuing delay is observed in the previous time step to allow larger queuing delay in the next time step, or the threshold is decreased to constrain queuing delay. Through the threshold adjustment, packets are dropped when the queue length exceeds the queuing threshold. Al-Jabber *et al.* [8] and Wang *et al.* [174] extended [68] to use a multi-source arrival process in establishing the delay maintaining mechanism, the arrival process used is based on Binomial distribution.

Table 2.5 gives a brief summary of queue management schemes in wired domain.

Table 2.5: Summary of Network Delay Control Schemes (Part 5)

QoS Schemes	Goals	Approach	Delay Control
Queue management (wired domain)	• To mitigate congestion in networks • To lower or constrain queuing delay	Manage a queue based on approaches below: • Drop or mark packets between minimum and maximum thresholds probabilistically based on a dropping probability function [9,13,55,60,61,117,141,195] • Drop or mark packets based on dropping probability calculated against target QoS metrics (e.g. queue length, loss rate, link utilization, etc) [16, 54, 56, 57, 74, 75, 88, 102, 106] • Drop or mark packets by using a fuzzy logic approach [36, 37, 156, 191] • Drop packets based on the dynamic queue threshold calculated according to delay constraint [8, 68, 174]	• Control queuing delay • Queuing delay is minimized by alleviating congestion in the network and maintaining lower backlogs in a queue • Mainly target for TCP traffic • Queuing delay are mostly not bounded except for schemes [8, 68, 174]

2.3.5.2 Wireless Networks

Drop Tail is still the main queuing discipline being used in wireless nodes for ad hoc networks [40, 97, 99, 100, 135]. Compared to wired networks, not that many AQM schemes are proposed for congestion control. AQM approaches in wired networks are not that efficient in controlling congestion in wireless networks due to different characteristics exhibited in wireless networks, such as dynamic topology, error-prone link, high interference, collision and resource constraints. Most of the wireless networks characteristics and behaviors exhibited could be derived from information gathered at a lower layer especially MAC layer. Therefore, wireless network congestion control schemes are moving towards a cross-layer approach and involve MAC layer in the design.

In wireless ad hoc networks, all wireless nodes need to act as routers to relay packets of other nodes. Therefore, the router-based queue management concept from the wired domain has been adopted by some researchers into wireless domain. These AQM schemes take into account of wireless characteristics and MAC layer information to calculate dropping probabilities and then drop packets probabilistically.

LRED (Loss ratio based RED) [62,120] is a RED-based AQM scheme but based on link layer information to mark or drop packets. The dropping probability of LRED is calculated based on the number of attempts at the MAC layer to transmit a packet. The average MAC retry count is estimated using an EWMA estimator. If MAC retry count increases in transmitting a packet then local congestion is implied. Packets are dropped based on the dropping probability calculated when the retry count exceeds the minimum threshold of MAC retries.

NRED (Neighbourhood RED) [180] is a distributed RED-based AQM scheme designed for wireless ad hoc networks. Nodes detect congestion based on the threshold of queue length at neighbourhood. The neighbourhood queue length is estimated via passive measurement of channel utilization instead of having each node to broadcast its queue length periodically or probing neighbourhood nodes. The channel utilization measurement is then being translated into average queue length. If the utilization exceeds the threshold, it indicates early congestion. The dropping probability is then calculated and is broadcasted to all neighbours. Each neighbour node calculates a local drop probability based on the notification above and drop packets accordingly.

Chen *et al.* [31] propose a RED-like queue management scheme for the base station of third generation wireless networks. The proposed scheme aims to reduce packet dropping of real-time traffic caused by expiration of deadline and to lower queuing delay. The minimum and the maximum queue thresholds used for packet dropping decision and the packet dropping probability are derived based on the worst case and the best case of queuing delay respectively. The worst case scenario assumes a packet gets transmitted or dropped from a queue after the maximum retries in retransmitting the packet. While the best case scenario assumes that a packet gets transmitted for the first attempt. The derivation is based on the assumptions of propagation delay, Internet delay and packet size are constant. The dropping probability is calculated similarly to DSRED based on the thresholds.

AREED (A Random Early Expiration Detection) [34] is an extension of queue management scheme proposed by Chen *et al.* [31] which uses the same derivation of the minimum and the maximum thresholds but added a median threshold. Instead of using a fixed maximum dropping probability (max_p) as in [31], the max_p is adapted based on the channel trend. Other than that, the minimum and the median thresholds are recalculated when the user rate is changed. For both schemes, packets are dropped probabilistically to increase link utilization and lower queuing delays are maintained.

AHRED (Ad hoc Hazard RED) [3] is another RED variant that has been proposed for wireless ad hoc networks. AHRED uses mechanism similar to RED to mark or drop packets except the dropping probability function is based on Weibull model of hazard rate function. The parameter of hazard function changes according to the queue length. The author claims that AHRED performs better than Drop Tail, RED, SRED and REM in terms of packet loss, throughput and delay. However, it has no constraint on the maximum queuing delay.

PSRED (Priority Self-adaptive RED) is proposed by Kong *et al.* [97] to avoid starving to death phenomenon of lower priority queues. It is a RED-based queue management scheme that drops packets probabilistically but adds in the support of priority adaptation for packets in lower priority queues. The priorities of packets from lower priority queues are upgraded to higher priorities after the highest priority queue has been served for a specific amount of time. Instead of starving to death, those packets get transmitted after being upgraded to higher priority queues and thus their end-to-end delay is reduced.

MADR (Media Access Delay Regulator) [179] is a control theoretic AQM scheme to control congestion at a WLAN AP. It is an extension of PI-controller AQM scheme proposed for wired networks with the difference that media access delay is used as a control target instead of queue length. It aims to maintain the average media access delay of TCP traffic entering WLAN. The target queue length is calculated based on the target average media access delay and the average service time of TCP packets in the sliding window from measurement. The dropping probability is then calculated based on the current and the target queue length. Packets are dropped based on the dropping probability calculated.

Marbach *et al.* [125, 126] propose an AQM approach for congestion control in wireless ad hoc networks by monitoring backlogs at MAC layer. The proposed scheme randomly drops incoming packets to keep backlogs at the target level. The dropping probability is calculated based on the backlogs level at a node. The larger the backlogs the higher probability that the channel is busy. Therefore, the dropping probability is increased or decreased based on the channel status that can be derived from the backlogs level. Besides the random drops, a node uses busy tone (in a separate channel) to inform other nodes on channel busy when the node senses its 1-hop neighbour is transmitting.

PAQMAN (Prediction bAsed Queue MANagement) [98] is a predictive queue management that uses recursive least squares (RLS) to predict average queue length from the past samples and then regulates the packet dropping probability based on the predicted average queue length. A target queue length needs to be carefully selected for the dropping probability regulation and for packets dropping or marking. Packets are dropped or marked probabilistically between the target queue length and the maximum queue size. It mainly aims to alleviate congestion and to maintain low queuing delay in wireless networks by providing early congestion indicators to the sources to regulate their sending rates based on the random loss events.

Natsheh *et al.* [136] highlight the need of a highly adaptive AQM scheme for wireless ad hoc networks due to time-varying link quality of wireless channels. Fuzzy-AQM has been proposed to adapt the dropping probability for a queue based on the current queue length and neighbour nodes density through fuzzy rules defined. It aims to achieve high queue utilization, low packet loss and low delay through congestion control for TCP flows.

QMMN (Queue Management scheme for Multihop Networks) [135] is proposed for multi-hop wireless mesh networks to avoid bias towards longer hops packets. Shorter hops packets enjoy higher performance due to less loss probability. The proposed scheme allows fair share of buffer allocation for different traffic flows at intermediate mesh points. The buffer occupancy and fair share of traffic flows from other mesh points are maintained at each mesh point. The fair share of buffer is updated based on the moving average of inter-arrival time and service time of the traffic flows from the particular mesh points. Packets are dropped when the buffer occupancy of the particular mesh point exceeds the fair share calculated and the residual share is used up.

Kalil *et al.* [86] propose a queue management scheme that provides service differentiation to long hops (LH) flows and short hops (SH) flows to reduce loss probability of LH flows. Higher impacts observed when LH flows get dropped compared to SH flows as more nodes involved in relaying packets for LH flows.

Bandwidth and energy wastage will be more significant in this case. Therefore, LH packets are prioritized and enqueued if the queue is not overflow. While SH packets are dropped probabilistically based on the number of LH packets and SH packets in the queue besides the queue threshold. SH packets are dropped when the queue length above the queue threshold configured for SH flows.

Similar to approaches in wired domain, all these approaches assume that sources can respond to packet loss events. These approaches drop packets probabilistically and require rate adaptation from the sources in order to mitigate congestion or some other goals such as fairness. They are not designed to constrain queuing delay. Queuing delay may be lowered if the sources are responsive; otherwise queuing delay is only bound by the maximum queue size.

There are a few AQM schemes [48, 110, 153, 166, 182, 189] which do not use probabilistic dropping approach. These AQM schemes use approaches such as transmission rate adjustment, queue size adjustment, etc. CAPEL (Channel state Aware Packet discard on Expiration Likelihood) [189] is an AQM scheme that involves MAC layer in the design. CAPEL is a channel aware algorithm that takes into account of the wireless channel status and packets' lifetime when enqueuing packets. A packet is discarded if the estimated lifetime is below the lifetime threshold. With this approach, wastage of bandwidth due to packet expiration is reduced and queuing delay is also decreased.

Tang and Li [166] have proposed three queue management schemes, namely, SQM (Simple Queue Management), UQM (Utilization based Queue Management) and FCQM (Fast-Convergence based Queue Management), based on analytical models to increase link utilization and to regulate the packet loss rate at the target threshold by adjusting the queue size. SQM, UQM and FCQM have the same goal of regulating the packet loss rate; but different methods are used in adjusting the queue size based on the packet loss rate derived from the analytical models. Simple addition and deduction of queue size based on predefined values are used in SQM. While UQM adjusts the queue threshold adaptively with the aim to maintain the same link utilization after the adjustment. FCQM enhances UQM to achieve quick convergence to the target packet loss rate.

Dousse [48] propose to push buffering of packets from network layer or MAC layer of intermediate nodes to the application layer of a source node. The proposed scheme mainly aims to improve link utilization and to reduce packet loss caused by collisions for TCP traffic. The author has shown that a bigger buffer does not increase the system performance, but leads to larger end-to-end delay. Therefore, the author proposes that packets should not be buffered at intermediate nodes; the size of the transmission queue is set to one packet. Co-operation of MAC layer is needed to implement this scheme. Each node needs to overhear packet transmission of its target node and ignore incoming RTS packets when the buffer of the target node is occupied.

CXCC (Cooperative cross-layer Congestion Control) [153] is a cross-layer congestion control approach for multi-hop wireless networks. It uses overhearing mechanism as implicit acknowledgment for packet forwarding to the next hop. Each node listens to its channel to ensure that a packet has been forwarded to the next hop by its neighbour before it proceeds to forward next packet to its neighbour. A timeout mechanism is used to detect for packet loss in order to prevent the node from waiting for implicit acknowledgment infinitely. For this proposal, network interface queue is eliminated; the per-flow queuing is maintained inside CXCC. It maintains per-flow queue with maximum one packet in the queue. A downstream node that wishes to send packets from same flow will need to wait for its upstream node to forward the packet to its next hop successfully before it can send the next one. This is known as implicit back pressure mechanism.

A* algorithm [110] is a hybrid algorithm which consists of eBDP (emulating Bandwidth-Delay Product) algorithm [111] and ALT (Adaptive Limit Tuning) algorithm [109] to adapt buffer size at wireless access point in order to improve TCP throughput and link utilization while maintaining low queuing delay. The eBDP algorithm measures mean service rate at MAC layer and then uses the mean service rate and the maximum queuing delay allowed at AP to calculate the AP buffer size. Additional buffers are added to the calculation for over-provisioning to absorb bursty traffic. While for ALT algorithm, AP buffer size is calculated based on link idle time, link busy time and a few design parameters which factor in the additive increase multiplicative decrease (AIMD) mechanism of TCP. The buffer size is then chosen between the minimum size calculated by eBDP and ALT. The authors have demonstrated that the use of fixed-size buffers lead to channel under utilization or high delays. Although A* algorithm has resulted in lower RTT from AP to wireless nodes; nodal delay is not bounded since there is an over-provision factor for eBDP algorithm and MAC layer delay is not compensated by adjusting allowable queuing delay. The maximum queuing delay stays the same for the algorithm as the main focus is to improve TCP throughput and link utilization.

Table 2.6 gives a brief summary of queue management schemes in wireless domain. None of the queue management schemes discussed above takes delay requirement as an explicit input to constrain queuing delay to a specified value. Bounded delay is needed for delay-sensitive traffic. Since that queuing delay is one of the major contributors to end-to-end delay, queue management should focus on constraining queuing delay with the ability to adapt to time-varying link quality and network dynamics.

Table 2.6: Summary of Network Delay Control Schemes (Part 6)

QoS Schemes	Goals	Approach	Delay Control
Queue management (wireless domain)	• To mitigate congestion in networks • To lower queuing delay	Manage a queue based on approaches below: • Drop or mark packets between minimum and maximum thresholds probabilistically based on dropping probability function [3, 31, 34, 62, 97, 180] • Drop or mark packets based on dropping probability calculated against target QoS metrics (e.g. media access delay, backlog in queue, queue length, packet lifetime, loss rate, link utilization, etc) [98, 125, 126, 166, 179, 189] • Drop or mark packets by using fuzzy logic approach [136]. • Drop or mark packets based on fair share of bandwidth or number of hops [86, 135] • Eliminate buffering at transmission queue [48, 153] • Adjust transmission rate or queue size based on link utilization, channel condition, queuing delay, etc [109–111, 182]	• Control queuing delay • Queuing delay is minimized by alleviating congestion in networks and maintaining lower backlogs in the queue • No bound on queuing delay • Mainly target for TCP traffic, lack of queue management to constrain delay for delay-sensitive traffic • Lack of knowledge on MAC layer delay

2.3.6 Packet Scheduling

Packet scheduling is a function or an algorithm that selects a packet from head-of-line (HOL) of queues for transmission based on the priority of packets, delay differentiation among traffic flows, the waiting time of packets, etc. This subsection focuses only on packet scheduling above MAC layer. For MAC layer scheduling, refers to priority queuing in MAC layer such as in IEEE 802.11e and TDMA scheduling as in Subsection 2.3.4.

Packet scheduling is used to ensure efficient link utilization and to provide delay bound guarantee for delay-sensitive applications besides fair resource sharing [53]. Scheduling algorithms can be divided into two categories: 1) work-conserving discipline and 2) non-work-conserving discipline [38]. Work-conserving schedulers are never idle if there are packets awaiting transmission. FCFS (First-Come-First-Served), PQ (Priority Queuing), WFQ/PGPS (Weighted Fair Queuing / Packet Generalized Processor Sharing), VC (Virtual Clock), WRR (Weighted Round Robin), SCFQ (Self-Clocked Fair Queuing), Delay-EDD (Delay- Earliest Due Date) and DRR (Deficit Round Robin) are examples of work-conserving schedulers [38, 51, 53, 140]. While for non-work-conserving schedulers, the schedulers may enter idle state even there are packets awaiting transmission as the schedulers may be expecting other higher priority packets to arrive or handling jitter requirements.

Generally, a system that uses non-work-conserving scheduler experiences higher average delay as compared to a system that uses work-conserving scheduler. HRR (Hierarchical Round Robin), SGQ (Stop-and-Go Queuing), Jitter-EDD (Jitter-Earliest Due Date) are examples of non-work-conserving schedulers. From the survey done by Cottet *et al.* [38], WFQ, VC, Delay-EDD, HRR, SGQ and Jitter-EDD are able to guarantee delay bound if all sessions obey their traffic specification. This implies that by using scheduling approach alone, the delay bound is not guaranteed. A traffic shaper such as a leaky bucket or a token bucket may be used to ensure the sessions conform to resources allocated.

The FCFS scheduler [38] is the most basic scheduling algorithm that serves packets in the arrival order of the packets. It is simple to implement, but it does not provide any bound on delay. While for the PQ scheduler [38], it serves packets from queues according to priority. Incoming packets are put into different queues based on their priority. The scheduler serves packets from the highest priority first then followed by the next lower priority. As long as there are packets at higher priority queues, they are scheduled to be transmitted first. So high priority packets experience lower delay. However, there is no deadline associated with the priority. This scheduling discipline may lead to starvation of lower priority queue.

WFQ [44] is a scheduling discipline that uses separate queues for packets from different sessions or connections. It is an extension of the FQ discipline [134]. FQ aims to maintain equal share of bandwidth for all connections. Each queue is served in a round-robin fashion. Therefore, FQ is not suitable for bounding and controlling delay. WFQ allows different sessions to have different service shares indicated by the weight of that session. A traffic flow with higher priority will gain a bigger share of bandwidth. Therefore, queuing delay for that flow is shortened. The drawback of WFQ is the computation complexity of the algorithm. WF2Q [18] and SCFQ [66] are variants of WFQ that aim to solve the drawback of WFQ.

The WRR scheduler [90] also uses separate queues for different traffic flows. WRR scheduler serves several packets for each non-empty queue in round robin fashion according to the normalized weight of traffic flow against the mean packet size. DRR [159] is a modified WRR scheduling discipline. It overcomes the issue of which the mean packet size must be known for scheduling decision. The WRR scheduler serves all non-empty queues according to the weights and the mean packet size whereas the DRR scheduler serves packets at HOL of every non-empty queue that the deficit counter is greater than the packet size.

The VC scheduler [193] uses virtual finish time to schedule packets. Each packet is timestamped with the virtual clock. The timestamp is calculated based on the virtual clock and the average reserved throughput for that flow. The packets are then transmitted in the order of increasing timestamps.

The Delay-EDD scheduler [58] assigns scheduling deadlines to delay-sensitive packets. The deadline of a packet is the time at which the packet should be sent out with conformance to the service level agreement (SLA). The priority of a packet is determined by the deadline assigned. Therefore, packets are sent out in increasing order of deadlines. Jitter-EDD [171] is an extension of Delay-EDD that aims to guarantee jitter bound besides delay bound. Packets are scheduled based on deadline but with jitter included for the deadline determination. This is to make sure packets receive same delay at each intermediate hop.

Table 2.7: Summary of Network Delay Control Schemes (Part 7)

QoS Schemes	Goals	Approach	Delay Control
Packet scheduling	• To ensure efficient link utilization • To provide delay bound guarantee for delay-sensitive applications • To provide fair resource sharing	Selects packet from HOL of queues for transmission [38, 51, 53, 140] based on : • priority of packets • delay differentiation among traffic flows • deadline of packets • waiting time of packets	• No explicit constraint on delay components • Minimize end-to-end delay for delay-sensitive traffic through prioritization of packet transmission

Table 2.7 gives a brief summary of packet scheduling schemes. From the operation of scheduling algorithms discussed above and the findings of Antila *et al.* [14], obviously accuracy of delay estimation and calculation of deadlines for packets are important in order to give the best scheduling decision. Multiple queues are also needed to implement these schedulers. Therefore, the scheduling approaches discussed above are actually designed for intermediate core routers with a lot of resources (e.g. buffers and processing power). Such approaches are not suitable for resource constrained nodes in wireless ad hoc networks. Even though the scheduler could bound delay by making right decision in picking packets from HOL of queues to be transmitted, the transmission of packets is still dependent on the gaining of medium access for that node. Medium access contention delay is not taken into account for these scheduling disciplines.

2.3.7 Service Differentiation

Service differentiation is an important QoS provisioning scheme that enables coexisting of real-time traffic flows and non real-time traffic flows in networks. Contrary to integrated service that provides end-to-end QoS to real-time traffic through a static resource allocation or a dedicated transmission path; service differentiation model provides different treatment towards delay-tolerant and delay-sensitive traffic flows by prioritizing delay-sensitive traffic handling. This is to prevent deterioration of QoS to real-time traffic caused by best-effort traffic. Service differentiation is normally achieved via hop-by-hop treatment on traffic flows based on their QoS requirements with different QoS provisioning schemes, such as admission control, traffic conditioning, traffic classification, etc.

DiffServ (Differentiated Services) is one of the most popular QoS models for wired networks falls under this category. Some researchers claim that the DiffServ model is not suitable for wireless ad hoc networks [7, 178]. Several other models are introduced to achieve service differentiation in wireless ad hoc networks are discussed subsequently. Some of the proposals [6, 7, 72, 95, 164, 178] involve admission control, traffic conditioning and resource reservation to provide service differentiation; while some other proposals [35, 42, 43, 49, 84, 104, 151] only involve scheduling and queue management mechanisms to provide service differentiation. Besides that, fuzzy logic approach is slowly gaining its importance in service differentiation proposals [36, 93, 94] in order to adapt to dynamic behaviors of wireless networks.

FQMM (Flexible QoS Model for MANET) [178] is a hybrid model of IntServ and DiffServ that aims to provide per flow QoS guarantee to the highest priority traffic and per class QoS guarantee to all others traffic. Three main features in FQMM are: dynamic roles of a node, hybrid provisioning to determine and allocate resources based on priority classes of traffic, and adaptive conditioning that consists of traffic profile, meter, marker and dropper. Traffic conditioning is applied to the traffic source to make sure that traffic generated conforms to its traffic profile. HQMM (Hybrid QoS Model for MANET) [72] is similar to FQMM which combines per flow and per class QoS guarantees. The different between HQMM and FQMM is the signalling protocol used in providing per flow QoS guarantee. INSIGNIA [107] is used in HQMM scheme instead of RSVP.

SWAN (Service differentiation in stateless Wireless Ad hoc Networks) [6] is a stateless differentiated service model in MANET that does not require per-flow information and need not using any signalling mechanism to reserve resources along the path. SWAN comprises of sender based admission control, dynamic regulation of real-time traffic and rate control for best-effort traffic. Sender-based admission control is used to decide whether to admit a new flow of UDP real-time traffic. Probing packets are sent to find out the instantaneous end-to-end bandwidth availability (bottleneck) along the path to the receiver. Each intermediate node intercepts the probing packets and updates the available bandwidth in UDP control packet field if the local bandwidth estimation is smaller than the available bandwidth carried in the control packets. This approach forces the sender to reestablish the real-time session when a network experiences congestion instead of regulating maximum allowable delay at each hop to maintain the real-time session. This causes a lot of overhead in terms of session setup (packet probing) for fast and dynamic changing wireless networks.

Sun *et al.* [164] propose to use service differentiation to support real-time traffic in large scale MANET. The proposed approach provides service differentiation for real-time and best-effort traffic using call admission mechanism. Admission control is assisted by a modified AODV routing protocol to include path selection of fixed router for real-time traffic. Besides that, adaptive priority scheduling at MAC layer is used to provide service differentiation by having different backoff time for different priority traffic. This approach is only applicable to wireless mesh networks with fixed router installed and not suitable for other types of wireless networks that any nodes can be a router and nodes may be mobile.

De Vuyst *et al.* [42, 43] propose a scheduling mechanism that provides differentiated service to delay-sensitive and delay-tolerant traffic through reservation of multiple slots in a queue. Delay-sensitive traffic has higher priority than delay tolerant traffic. The proposed scheme aims to reduce queuing delay of delay-sensitive traffic at the cost of allowing higher queuing delay for delay-tolerant traffic. A fixed amount of slots is reserved at HOL of a queue during initialization. Whenever a packet of delay-sensitive type enters a queue, it is enqueued into the reserved slot that is nearest to HOL of the queue. Packets of delay-tolerant type are enqueued into the empty slots of the queue that are not reserved. Therefore, packets of delay-sensitive type are given priority in the scheduling of transmission.

PDD (Proportional Delay Differentiation) [49] is another scheduling approach that provides differentiation of queuing delay to traffic from different priority classes with respect to the proportional delay ratio of traffic classes. PDD controls the ratios of average queuing delay between traffic classes according to delay differentiation parameters (DDP) defined in SLA. PDD maintains separate queues for different traffic classes. A scheduler is used to determine which queue to be served next by measuring delay of each queue. Three scheduling disciplines based on proportional delay ratio have been introduced in this model. The PAD (Proportional Average Delay) scheduler serves the queue of traffic class with the largest normalized average delay, while the WTP (Waiting Time Priority) scheduler serves the queue of traffic class with the largest normalized HOL waiting time. HPD (Hybrid Proportional Delay) is a hybrid approach of WTP and PAD. PDD does not provide delay bound since the scheduling decision is based on proportional delay ratio of traffic classes and not absolute deadline of data packets. With this approach, QoS for each traffic class may vary with network conditions but differentiation QoS received by each class should remain constant. However, this assumption is only true if there is no contention delay at nodes. The proportional delay ratio cannot be achieved at each node resulting from additional random probabilistic waiting time at MAC layer [151].

Scheduling disciplines introduced in PDD become a foundation of other approaches [84, 104, 151, 175]. WTP focuses on the accuracy of achieved proportional delay between different traffic classes without considering packet transmission time. MWTP (Maximum-WTP) and VWTP (variance-WTP) [104] are the extension of WTP to take into account of packet transmission time. Virtual waiting time is calculated in MWTP to include transmission time as a decision point. Transmission time may vary depends on the packet size at HOL of each queue. Therefore, the proportional delay ratio may vary depending which queue is served next. Similarly, VWTP scheduler makes scheduling decision based on the smallest variance between HOL waiting time ratio that includes transmission time and the DDP. From the simulation results presented by Lai, MWTP and VWTP perform better than WTP in terms of accuracy and lower queuing delay.

MWTP (Multi-hop WTP) [84] is another extension of WTP to guarantee proportional delay differentiation for a multi-hop wireless network. It takes care of the accumulated contention delay effect suffered at multi-hop links. A multi-hop wireless network has higher collisions and thus contention delay may be high. The waiting time calculation is based on the departure time at an upstream node, therefore MAC contention delay from the upstream node and queuing delay at the node itself are included. The scheduling decision is then made based on the maximum normalized delay calculated. This approach needs clock synchronization among the nodes in the network.

EPDS [151] is a QoS framework that builds on the WTP scheduler in the PDD model to provide end-to-end QoS in wireless ad hoc networks with admission control and congestion control schemes included in the framework. The IEEE 802.11e EDCA standard is used in this framework to overcome the random probabilistic waiting time problem in WTP. Admission control is used to control the number of traffic flows being admitted into a network. A traffic flow is admitted only if the estimated end-to-end delay can meet the application's requirement; otherwise the connection is refused. End-to-end delay is estimated through the route discovery process. When a route reply packet is received, the RTT is halved to obtain one way delay. No resources are reserved during the admission control process, therefore congestion may still happen if the network is highly loaded. The network load status, which is derived from bandwidth estimation through idle channel time measurement, is used as a congestion indicator. If the idle time falls below a predefined threshold, it indicates a network is highly loaded and congested. Consequently, a flow is randomly picked by the congestion control algorithm to be rejected. In the framework, the priority of the traffic flow and the contention window of IEEE 802.11e are adjusted adaptively based on the network load in order to make sure the traffic flow adhere to proportional delay differentiation required.

RED-RT (RED Real-Time) [35] is an extension of RED congestion control scheme that includes different treatment for real-time and non real-time traffic. Two different dropping probabilities are used to mark or drop real-time and non real-time traffic when congestion occurs. Non real-time traffic is treated with the normal RED dropping policy, packets are dropped probabilistically when the queue length exceeds TH_{MIN}. All packets are dropped when the queue length exceeds TH_{MAX}. Real-time traffic is only dropped probabilistically when the queue length exceeds TH_{MAX}. In this case, real-time traffic is given priority to be transmitted during congestion.

The other service differentiation technique via congestion control in combination with traffic conditioning technique is proposed by Kim *et al.* [95]. Real-time traffic is passed through a token bucket for traffic shaping to control the rate of real-time traffic flow and to ensure the flows conform to their traffic profiles. During congestion, best-effort traffic is conceded to give way to real-time traffic using back pressure mechanism. A congestion notification is sent to upstream nodes to concede the bandwidth allocated for best-effort traffic. Congestion detection is done at each node via delay and bandwidth utilization of real-time traffic flows.

FuzzyMARS [94] and FuzzyCCG [93] are fuzzy logic approaches that complement each other to provide service differentiation for wireless ad hoc networks. Admission control and temporary resource reservation are used to admit a real-time traffic flow. A probing request is sent out from a source to a destination to find out the minimum bandwidth available along the transmission path. If the available bandwidth is sufficient for the new real-time flow, the destination node informs the source node via a short reply message. At the same time, the bandwidth required is reserved temporary for that flow. The reserved bandwidth is released after the expiration. Best-effort traffic is regulated dynamically in order to alleviate congestion in the network. The regulation rate is determined by the fuzzy logic controller based on the MAC layer feedback on delay experienced by packets. Real-time traffic is regulated based on priority of the traffic flows and the QoS requirements. The fuzzy logic approach enables the framework to operate with imprecise information due to network dynamics in wireless ad hoc networks.

FIO (Fuzzy explicit marking controller In/Out) [36] also uses fuzzy logic approach to control the queue length of real-time traffic and best-effort traffic at a predefined level. A fuzzy logic controller is normally used to design a closed-loop feedback control where a rigorous control theoretic approach cannot be used due to difficulty in obtaining the analytical model. The target queue length for best-effort traffic is lower than real-time traffic in order to have a higher marking or dropping probability for best-effort traffic. The fuzzy logic controller calculates the marking probability for both queues based on difference between the target queue length and the instantaneous queue length. The simulation results show large standard deviation for delays. This mechanism constrains the delay for higher priority traffic but it fails to provide guarantee of service for delay limit.

Service differentiation is important to provide a certain level of QoS guarantees to real-time traffic with less overhead as compared to integrated service. Delay-sensitive traffic may get delivered to destinations with a shorter OWD, but there is no guarantee on the delay bound. The approach may improve network link utilization as no static resource allocation is used for real-time service, this enables left over bandwidth to be shared by best-effort traffic. Table 2.8 gives a brief summary of service differentiation schemes reviewed.

Table 2.8: Summary of Network Delay Control Schemes (Part 8)

QoS Schemes	Goals	Approach	Delay Control
Service differentiation	• Provide different treatment towards delay-tolerant and delay-sensitive traffic by prioritizing delay-sensitive traffic handling • Prevent deterioration of QoS towards real-time traffic caused by best-effort traffic	Provide hop-by-hop treatment to traffic flows based on their QoS through: • Combinations such as admission control, traffic conditioning and/or resource reservation [6, 72, 95, 164, 178]. • Combination of scheduling and queue management mechanisms [35,42,43,49,84,104,151] • Fuzzy logic approach to adapt to dynamic behaviors of wireless networks [36, 93, 94].	• Contribute to network delay control implicitly • No bound on queuing delay and MAC delay. • Minimize network delay of delay-sensitive traffic through prioritization of delay-sensitive traffic • Mostly focus on lowering queuing delay of delay-sensitive traffic

2.3.8 Cross-Layer Approach

A cross-layer approach is an emerging approach to provide optimized QoS provisioning solutions in wireless ad hoc networks [26,108,137,167]. Because of complex interdependency of layers in wireless nodes, an optimal design or configuration can only be achieved by sharing information from other layers in the protocol stack. Cross layer approach facilitates information sharing and feedback between different layers in protocol stack and enables QoS components from different layers to react on information gathered to achieve end-to-end QoS provisioning.

Wang and Ramanathan [175] propose a cross-layer approach to provide class-based service differentiation for TCP and UDP traffic in order to achieve throughput assurance for TCP traffic and delay assurance for UDP traffic. The authors propose to use the neighbourhood proportional delay differentiation (NPDD) service model with a dynamic class selection at the application layer to give a higher priority to packets with longer waiting time with respect to its DDP through the WTP scheduling. The solution requires coordination from MAC layer to map packets into different MAC layer priority in IEEE 802.11e. The medium access priority selection mechanism adapts a node's priority based on the average normalized waiting time of packets transmitted from that node and packets transmitted within its transmission range. The proposed solution only ensures traffic of different classes get relayed to their destination based on its DDP and waiting time; but not bounding the end-to-end delay.

A cross layer approach of transport protocol (LATP) for multimedia streaming applications is introduced by Navaratnam *et al.* [137]. In this proposal, adaptation of the sending rate is done at the application layer to reduce network congestion and network delay. Streaming applications adapt their sending rate based on the rate feedback from receivers. The allowable sending rate is calculated based on MAC layer feedback at intermediate nodes, while the permissible throughput is calculated based on channel busy ratio and throughput at MAC layer of intermediate nodes. The allowable sending rate is then feedback to the receiver via IP option header if the calculated rate is smaller than the sending rate stated in IP header. When the data packet reaches the receiver, the minimum allowable sending rate is obtained. The receiver will send the feedback to the sender to enable it to adapt to the minimum allowable sending rate. Through this approach, congestion is reduced and network latency is reduced indirectly too. However, this approach might not be optimum for wireless ad hoc networks which normally use multi-path routing. There will be multiple paths for a source to reach its destination. This may lead to inaccurate rate adaptation and the fast changing topology may cause the rate feedback to be outdated very fast. The proposed solution relies on the ability of application layer to adapts its sending rate; however there are possibility that some applications are incapable to do so due to the traffic generation nature or other QoS requirements.

He *et al.* [71] claim that upper layer adaptation, is more appropriate. This is because QoS requirements are application dependent, and some applications such as audio applications may have multi-level QoS requirements depending on the network conditions. The proposed framework comprises of WTP, delay monitor, classifier, priority adaptor and requirement adaptor at the application layer. The delay information needed by priority and requirement adaptors are estimated via RTT measured from packets delivery and ACK for the packets. It is assumed that users prefer to tolerate quality degradation instead of dropping packets when the network condition is bad. The requirement adaptor lowers the delay requirements when the delay monitor indicates that network condition is bad. Besides, priority adaptor, which is a PI controller, is used to adjust the priority of the application to control the end-to-end delay within its target. The output of priority adaptor is fed into the classifier to dynamically change the service class of multimedia flow to meet required delay and jitter. WTP scheduler uses the mapping of classifier to schedule packets. This approach is only suitable for multimedia traffic that can tolerate quality degradation.

Li *et al.* [108] propose an adaptive per-hop differentiation (APHD) scheme to ensure the end-to-end delay requirement is met based on the EDCA scheme in the IEEE 802.11e standard. The approach relies on cross-layer information, such as end-to-end delay requirements from the application layer, node channel status and per class delay from MAC layer, to enable priority mapping at the network layer. Packets carry the end-to-end delay requirement in the header. Per-hop delay experienced by a packet is accumulated while the packet traverses to its destination. The cumulative delay of each packet is checked at intermediate nodes, the priority level of the packet is then adjusted based on the cumulative delay and the current channel status. Packets are assigned into different priority queues at MAC layer that uses different MAC contention parameter sets. This approach enables per-hop adaptation to bound end-to-end delay. However, the approach is MAC layer dependent; it is not transparent and not suitable for heterogeneous networks.

QPART (QoS Protocol for Ad hoc Real-time Traffic) [184] is a distributed QoS provisioning approach that consists of a QoS-aware scheduler for flow scheduling and a QoS Manager. The QoS manager manages admission control and resolves conflicts when running out of resources to guarantee the target QoS. Real-time traffic flows are separated into different queues. Each queue has its own backoff timer. The size of contention window is calculated and adapted at the network layer instead of MAC layer based on the per-hop delay requirement of real-time flows. The backoff time based on the contention window is passed to MAC layer; the MAC layer backoff mechanism has been replaced by this network layer mechanism. All flows update their contention window independently and control their medium access frequency through the calculated contention window size. The QoS manager will pick victim flows to be rejected when the network is congested. The proposed solution aims to constrain end-to-end delay via the per-hop scheduling based on the per-hop delay requirements. However, the solution is not scalable since it requires per flow adaptation and MAC layer modification.

DCLQ (Distributed Cross-Layer QoS) [112] is a distributed cross-layer QoS approach that aims to provide QoS guarantee for real-time traffic. DCLQ implements a per-hop delay QoS-aware priority scheduling and MAC layer service differentiation using IEEE 802.11e MAC. The priority scheduling is carried out by keeping a table that records the residual waiting time of real-time packets. The table is updated after each successful packet transmission. A packet with the smallest residual waiting is chosen to be delivered to MAC layer for transmission. The per-hop delay requirement is obtained by divided the maximum allowed delay for the real-time flow with the number of hops for that particular packet to reach its destination. Admission control is carried out to check if a route from the source to the destination exists and the queuing delay is below its per-hop delay requirement before accepting a new real-time flow. This framework is not scalable as it requires per packet tracking.

Though the cross layer approach has attracted a lot of attention in wireless ad hoc networks, the performance gain versus complexity of the design is yet to be further evaluated [194]. Most of the network nodes adopt a well-defined layering concept as in OSI (Open Systems Interconnection) model or TCP/IP model, therefore information sharing across layers is not a trivial task. In addition, modularity concept is compromised which may lead to issues such as maintainability and re-usability. Table 2.9 gives a brief summary of cross-layer approaches reviewed.

Table 2.9: Summary of Network Delay Control Schemes (Part 9)

QoS Schemes	Goals	Approach	Delay Control
Cross-layer approach	• To gather information and feedback from multi layers in the protocol stack for network conditions assessment • To provide end-to-end QoS provisioning based on the information and feedback gathered	Provide end-to-end QoS provisioning through information sharing and feedback between layers, such as: • Based on priority adaptation at application layer or network layer, delay differentiation at network layer and medium access adaptation at MAC layer [71, 108, 175] • Based on admission control and per-flow backoff time adaptation at network layer [184] • Based on sending rate adaptation at application layer. The sending rate and permissible throughput are calculated based on MAC layer feedback [137] • Based on priority scheduling, MAC layer service differentiation and residual time tracking [112]	• Minimize network delay of delay-sensitive traffic through service differentiation and prioritization of delay-sensitive traffic • Mostly focus on lowering queuing delay or MAC layer delay of delay-sensitive traffic

2.4 Summary and Discussions

Tables 2.1 - 2.9 summarize the network control schemes surveyed. Obviously, MAC layer approaches receive a lot of attention from researchers. For examples, admission control in wireless ad hoc networks mainly depends on MAC layer channel utilization (Subsection 2.3.1) to reject or admit a flow, bandwidth reservation is carried out at MAC layer (Subsection 2.3.2), congestion control approaches are contention-aware (Subsection 2.3.5), and other MAC layer approaches discussed in Subsection 2.3.4. However, MAC layer approaches may not be the best way in controlling network delay. It is more difficult to design and implement a MAC that might involve modification of firmware or hardware. Besides, changing or enhancing a MAC layer may impact other approaches at the upper layer that rely on the information gathering and scheduling at MAC layer to control QoS. Different kind of MAC or MAC variants might not be able to talk to each other directly. Therefore, interoperability issue is a concern for heterogeneous networks. Upper layer delay is mostly ignored in MAC layer approaches, whereas queuing delay is the most significant nodal delay component besides MAC layer delay. Therefore, bounded nodal delay is not guaranteed.

Admission control and bandwidth reservation approaches also receive a lot of attention to tackle shared bandwidth and fluctuation of network performance issues. These approaches prevent over-subscription of bandwidth to avoid congestion and contention in a network, so that end-to-end delay could be minimized. However, the accuracy of resource estimation and allocation are hard to achieve due to the decentralized nature of such networks and wireless characteristics. Network delay may grow drastically when these approaches fail to maintain an optimum resource utilization in the network. Therefore, a delay constraint mechanism is needed to react on this scenario before the optimum resource utilization is obtained again.

QoS-aware routing protocols have received a lot of attention due to the nature of self-configuration in wireless ad hoc networks. Dynamic routing is needed to establish paths for traffic flows. Delay-based QoS routing protocols, which rely on delay estimation and RTT, are proposed to discover routes that can satisfy the delay requirements for delay-sensitive traffic. Even with QoS routing in place, a deterministic end-to-end delay is needed to increase the efficiency of routes discovery and to minimize overhead of route maintenance. If network delay is not constrained corresponding to network dynamics; the delay requirements of the routes established may be violated and cause frequent re-routing. Bandwidth is shared among wireless nodes within the same proximity of transmission range so the changes of routes may disrupt other existing flows in the network due to imposed delay overhead [76]. This has motivated the development of a new approach to constrain network delay to cope with this problem.

While packet scheduling approach is more suitable to be used in core routers of wired networks instead of in wireless ad hoc networks. Packet scheduling approach normally needs to maintain multiple queues, this may become a burden to resource constrained wireless nodes. Besides that, wireless communication poses special channel-specific problems such as time-varying link capacities and location-dependent errors. There is no efficient scheduling for wireless ad hoc networks so far; issues such as how to improve bandwidth efficiency and maintain goodput fairness with various link qualities for power-constrained mobile hosts remain unresolved [28].

Service differentiation approach aims to provide different treatment for packets based on the packet classification. Real-time traffic and higher priority traffic are processed and transmitted faster than non real-time traffic and lower priority traffic. End-to-end delays of real-time traffic flows and higher priority traffic flows are minimized through this approach. However, there is no checking on violation of the delay bound at each node before a packet reaches its destination. The deterministic end-to-end delay may not be guaranteed since the per-hop delay is not bounded.

Queue management is one of the popular approaches to control queuing delay in the domains of wired and wireless networking. Buffers are used to improve link utilization and system performance; however the size of buffer does matter. Buffer sizing is very important as the buffer size has a direct impact on the QoS of the network such as throughput and delay. Inappropriate setting of the buffer size may lead to system performance issues such as large queuing delay if the buffer size is too large or high packet dropping rate if the buffer size is too small [123]. It is non-trivial to find an optimum buffer size, an optimum target queue length or optimum minimum and maximum thresholds in AQM.

Most of the existing buffer and queue management schemes aim to reduce queuing delay but not bounding the delay. This is because most of the researchers use an average queue length in queue management instead of an instantaneous queue length to absorb and tolerate the burstiness of traffic. The end-to-end delay needs to be guaranteed for delay-sensitive traffic. Therefore, the instantaneous queue length must be taken into account in order to bound the maximum queuing delay. Furthermore, most of the AQM schemes focus on dropping packets probabilistically based on current queue states and QoS constraints for delay-tolerant traffic but not for delay-sensitive traffic. The maximum queuing delay is only constrained by the maximum queue size or a constant target queue length. It is not the goal of AQM to control the worst case delay. Control theoretic AQM approaches are able to control the target queue length to a specified threshold by using a closed-loop feedback, but time-varying link quality and wireless characteristics in wireless ad hoc networks may cause fluctuation of system throughput and thus cause variation in queuing delay.

In addition, nodal delay is not bounded by using existing AQM approaches as other important delay components such as contention delay are not taken into consideration. The maximum queuing delay can be fine-tuned to compensate other delay components experienced by a node to achieve bounded nodal delay. This can be done by having a dynamic maximum queue size. This motivates a need of an adaptive queue management scheme that can react to network changes by adjusting the queue threshold adaptively in order to fulfil the nodal delay requirement at each node. So that a deterministic per-hop delay can be guaranteed explicitly. Consequently, the end-to-end delay can be bounded if nodal delays from a source to a destination are all bounded.

Chapter 3

Dynamic Threshold Queue Management

3.1 Introduction

This chapter presents an adaptive queue management scheme to bound queuing delay of network nodes to a required level based on a comprehensive analytical model under aggregated Internet traffic flows from various traffic classes. The arrival process is based on superposition of multiple Markov Modulated Bernoulli Process (MMBP-2). The proposed scheme is named as Dynamic Threshold (DTH).

Some control theoretic AQM schemes aim to control the average queue length to its required target, but determining an optimum target queue length remains an unresolved issue. PI controller [74] aims to regulate the queue length to a required value using a closed-loop feedback control. The dropping probability is updated periodically based on the queue length. The same approach is taken by Hong *et al.* [75]; the proposed AQM scheme aims to control the average queue length via an adaptive dropping probability which is calculated based on the average queue length, the target queue length and the estimated packet arrival. LQD [106] is an AQM scheme that adapts the dropping probability based on a target queue length and a target loss rate. While REM [16] uses a cost function to determine the dropping probability based on rate mismatch observed between packet arrival and packet departure; and queue length mismatch observed between an actual queue length and a target queue length. Although these approaches may control the average queue length to its target level, the average queue length is bounded with the assumption of maximum system utilization.

Bounding the queue size to a specified target queue length does not mean that the average queuing delay is bounded. The same target queue length may be translated into different average queuing delay with different system utilization and traffic characteristics. None of the aforementioned queue management schemes has the goal to constrain network delay to a specified value. There are only a few queue management schemes [8, 68, 174] focusing on constraining network delay in terms of average queuing delay. The dynamic queue size concept introduced is interesting and looks promising to constrain queuing delay. DTH is an extension to the scheme proposed by *Guan et al.* [68] with a more representative arrival process.

The remainder of the chapter is organized as follows: Section 3.2 gives an overview of the DTH system; Section 3.3 describes the system model of the DTH queue management scheme; Section 3.4 describes the limitations and assumptions of DTH; Section 3.5 then presents the analysis from analytical and simulation results; and lastly, Section 3.6 discusses on the feasibility and extensibility of the proposed scheme in wireless domain.

3.2 DTH (Dynamic THreshold) Overview

The DTH mechanism relies on the relationship mapping between queuing thresholds and average queuing delays that are generated a priori from a queuing analysis for potential scenarios based on targeted traffic profiles and system utilization. The traffic profiles and system utilization are assumed to be approximated from the past measurements collected by the service provider at the routers. A N-MMBP-2/Geo/1/K queuing model is proposed for the derivation of target queue thresholds. N MMBP-2 arrival processes are superposed to represent N number of aggregated Internet traffic classes.

The queuing model is an extension from previous work [68] to use multiple Markovian arrival source instead of single Markovian source as proposed by Guan *et al.* [68]; a single traffic source is not sufficient to model aggregated Internet traffic that consists of multi-class traffic. The aggregated network traffic is bursty and exhibits Long Range Dependence (LRD) characteristic [45, 89]. Although Al-Jabber *et al.* [8] and Wang *et al.* [174] propose to use multi-source in establishing the delay maintaining mechanism, the arrival process used is based on Binomial distribution. A Binomial arrival process cannot capture the burstiness and LRD characteristics of Internet traffic. Therefore, the proposed approach extends the previous work by giving a comprehensive investigation on the discrete-time queuing model to use multiple Markovian arrival processes in order to capture multi-class aggregated Internet traffic.

Markovian arrival process is a popular arrival process for bursty and correlated traffic modeling. For example, MMBP-2 [138, 139] and Markov Modulated Poisson Process (MMPP) [12, 132] can be used to serve the purpose. It was shown that superposition of MMPP can be used to model variable packet traffic with LRD [12, 132]. MMBP is the discrete-time counterpart of MMPP. Thus, superposition of multiple (N) MMBP is a good candidate for traffic arrival process to model aggregated Internet traffic in discrete-time queuing analysis for DTH.

A discrete-time approach is used for the proposed scheme instead of a continuous-time approach as computers and communications are discrete in nature. Only a single event can occur at any time instant for a continuous-time approach; whereas a discrete-time approach can allow multiple events to take place [177]. Although continuous-time approach is less complex, it cannot represent the digitalized communication world well. DTH aims to constrain average queuing delays at network routers to a required average queuing delay (TD) by adjusting the queuing threshold (L) dynamically (Fig. 3.1) through a closed-loop feedback controller based on the relationship derived from the queuing model. The feedback control loop results in a movable queue threshold for the system queue and maintains the system average queuing delay around the TD specified.

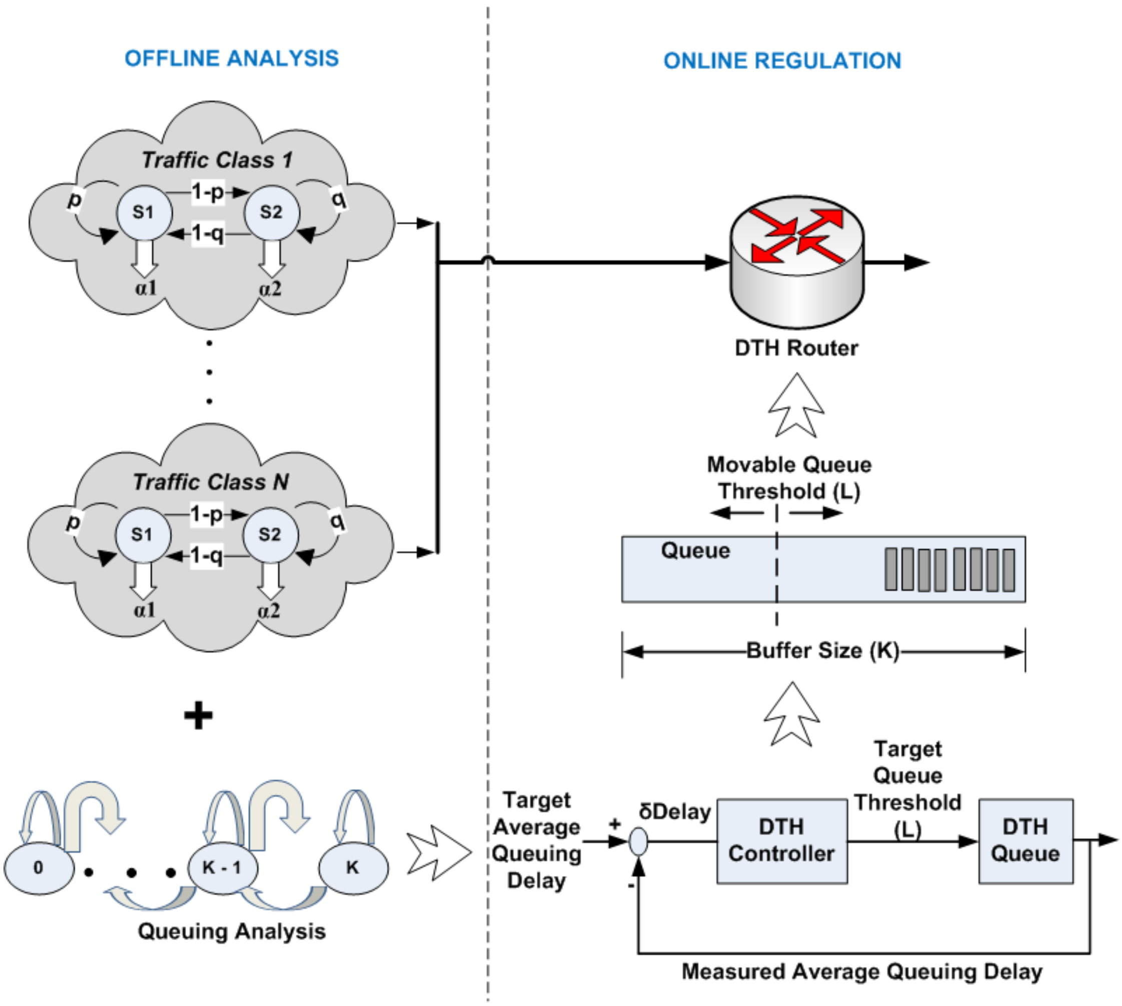


Figure 3.1: DTH System Diagram

The average queuing delay (D_k) is measured and calculated at a fixed time interval denoted as time window (TW) with associated index k ($k = 0, 1, 2, ...$). The D_k measured is used to obtain the delay delta (G_k) between the measured and target average queuing delay. The delay delta, G_k, is then used to predict the target average queuing delay ($\hat{D}_{k+1}$) for the next TW based on Eqs. 3.1 - 3.3 [68].

$$\begin{aligned} G_k &= kTD - \sum_{i=1}^{k} D_i\,;\ k = 1, 2, ... \\ &= TD - D_k + G_{k-1} \end{aligned} \tag{3.1}$$

$$TD = \frac{D_1 + D_2 + \cdots + D_k + \hat{D}_{k+1}}{k+1} \tag{3.2}$$

$$\hat{D}_{k+1} = 2 \times TD - D_k + G_{k-1}\,;\ k = 1, 2, ...\ \&\ G_0 = 0 \tag{3.3}$$

After obtaining the $\hat{D}_{k+1}$, the $\hat{D}_{k+1}$ is then being translated into a queue threshold, L, based on the mapping relationship obtained from the queuing analysis in Section 3.3. The queuing threshold is either increased, if a lower queuing delay is observed in the previous time step to allow larger queuing delay in the next time step; or decreased to constrain queuing delay. With the DTH queue management scheme, packets are dropped when the current queue length ($QLen$) exceeds the target queue threshold obtained from the mapping table. Packet loss events can serve as implicit congestion indicators for the sources to regulate their sending rates in order to avoid overloading the routers and also avoid a high packet loss rate.

3.3 DTH System Model

From the DTH mechanism explained in Section 3.2, the mapping relationship between queuing delay and queuing threshold is important in maintaining average queuing delay to its target level. A discrete-time queuing model is used to generate the mapping relationship with the assumption of packet departure probability (service rate) and packets arrival probability (arrival rate) are known. The queuing model introduced here uses N MMBP-2 with $N > 0 \in \mathbb{Z}$ to represent aggregated Internet traffic.

The queuing analysis is carried out by varying the queue threshold with superposition of N MMBP-2 sources. The queuing analysis with the same arrival process and the same departure probability or service rate (β) is carried out for

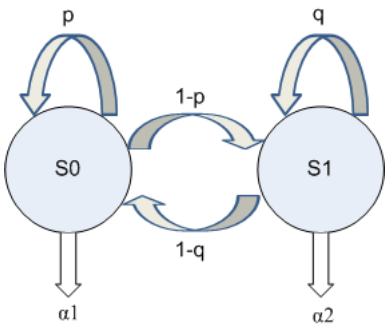


Figure 3.2: MMBP-2 Diagram

all queue thresholds, which ranges from 1 to K (maximum queue size). Average queuing delay observed for each queuing threshold is then derived through the steady state queue length probabilities obtained. After K iterative of queuing analysis; a mapping relationship between the threshold and delay is obtained from the analytical results.

With the periodic feedback of average queuing delay measurement, the DTH controller adjusts the queue threshold for next time unit to maintain the average queuing delay to the required value. The following subsections describe the traffic model and queuing model for this queue management scheme.

3.3.1 MMBP-2 Traffic Model

MMBP-2 source model (Fig. 3.2) is used as a traffic source model to represent bursty traffic [68, 138, 139]. There are two distinct states in a MMBP-2 arrival process, which are state *S0* and *S1*. The MMBP-2 source model is characterized by a transition probability matrix $\tilde{P}$ and a diagonal arrival probability matrix $\tilde{\Lambda}$ as given in Eqs. 3.4 and 3.5.

$$\tilde{P} = \begin{bmatrix} p & 1-p \\ 1-q & q \end{bmatrix} \tag{3.4}$$

$$\tilde{\Lambda} = \begin{bmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{bmatrix} \tag{3.5}$$

The MMBP-2 source can be used to model traffic flow with different characteristics by setting the state transition parameters (p, q) and arrival probabilities (α_1, α_2) of the model appropriately. It can become a Bernoulli source by setting the same arrival probability of both states in MMBP-2 to generate smooth and constant traffic patterns. It can also be used to model voice or video traffic. By setting arrival probability in either *S0* or *S1* to zero, MMBP-2 source becomes an IBP (Interrupted Bernoulli Process) [188] source which has ON and OFF state.

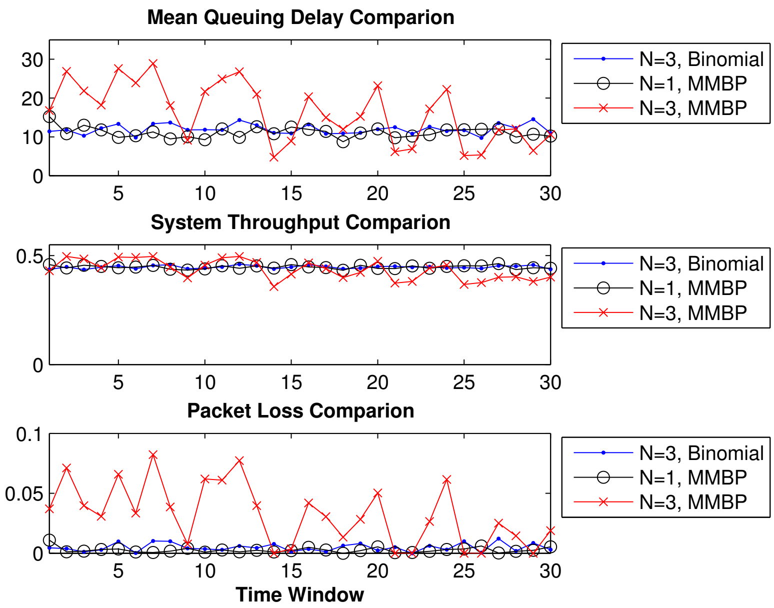


Figure 3.3: System Performance with Different Arrival Process

OFF state represents the silence state in the voice traffic. In short, MMBP-2 source can be adjusted to generate different traffic patterns with different traffic intensity or burstiness for traffic flows from different traffic classes.

Nevertheless, a single MMBP-2 arrival process is not able to represent multi-class aggregated Internet traffic. Internet traffic in the core network is an aggregation of multiple traffic flows from different traffic classes, such as constant bit rate (CBR), variable bit rate (VBR), voice, video, etc. Therefore, superposition of multiple MMBP-2 models is proposed to model the aggregated traffic flows at core networks.

A simple simulation using Drop Tail queue management ($queue\ size = 20$) with a different arrival process has been carried out to show why superposition of N MMBP-2 is chosen as the sources for the DTH model instead of single MMBP-2 [68] or Binomial sources [174]. From Fig. 3.3, it can be observed that the mean queuing delays measured from the simulation for single MMBP-2 traffic source and Binomial traffic sources show the similar trend with smooth average queuing delays. This implies that single MMBP-2 or Binomial sources cannot capture the burstiness of multi-class Internet traffic and thus may not give an accurate queuing analysis for DTH. Besides the mean queuing delay, the other system performance metrics, such as throughput and packet loss, would fluctuate with the burstiness of the traffic also.

3.3.2 Queuing Model

A discrete-time queuing model is adopted for the proposed queue management scheme. The model proposed is a N-MMBP-2/Geo/1/K queue followed the Kendall's notation. In a discrete-time queuing model, time is divided into slots of fix length interval. Inside a time slot, a packet arrival or/and departure may take place with the departure probability of β. For the queuing model proposed, a packet departure is assumed to take place before arrival in any time slots and at maximum one packet departure is allowed (see Fig. 3.4 for conventions of discrete-time queue). The queue size is finite for the model proposed, which means the queue can hold up to maximum K packets (where $K > 0$). The queuing discipline for the model is FIFO (First In First Out). Therefore when the number of packets in the system queue exceeds the queue threshold, packets are dropped.

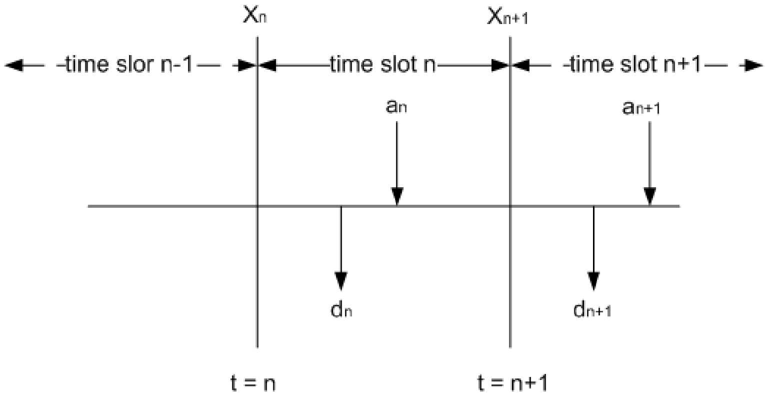


Figure 3.4: Conventions for Arrivals(a_n), Departures(d_n) and States(X_n) of a Discrete-Time Queue

The arrival process consists of N MMBP-2 source (where $N > 0$) that represents different traffic flows from either the same or different traffic classes. Each MMBP-2 represents aggregated traffic flows instead of individual traffic flow from the same traffic class. The arrival probabilities are assumed to be aggregation of arrival probabilities of all traffic flows from a particular class. All MMBP-2 sources in the arrival process are superposed together in order to generate an aggregated traffic flow. The number of packets generated at each time slot ranges from 0 to N with each MMBP-2 source generates at maximum one packet per time slot. Each of the MMBP-2 sources can be configured separately.

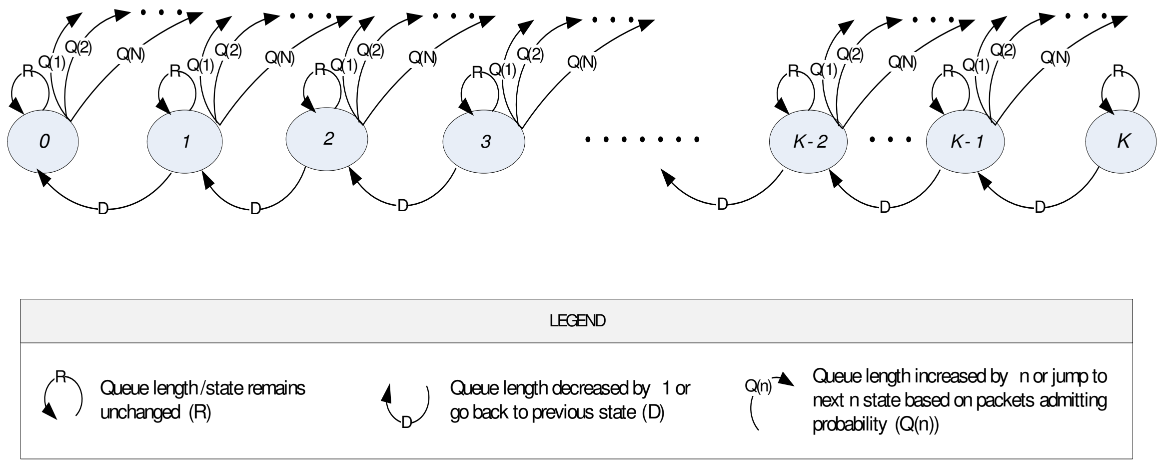


Figure 3.5: State Transition Diagram for Superposition of N MMBP-2 Arrival Process in Single Dimension

Fig. 3.5 shows the state transition diagram of the proposed queuing model. The diagram only captures queue length state transition in single dimension. In fact, the queuing model is a complex model with multi-dimensional transition. Figs. 3.6 - 3.9 gives an example of multi-dimensional transition breakdown diagram with $N = 2$ for each transition type in Fig. 3.5. The multi-dimensional diagram becomes complex as N increases.

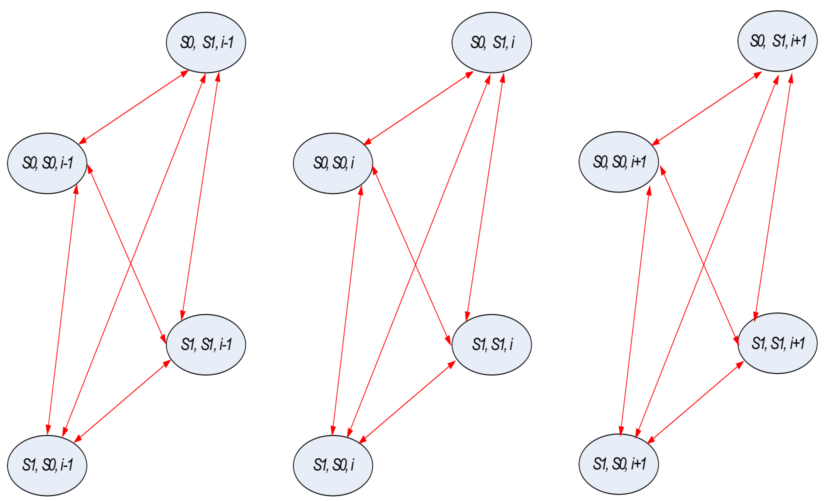


Figure 3.6: R elements in $\tilde{QT}$

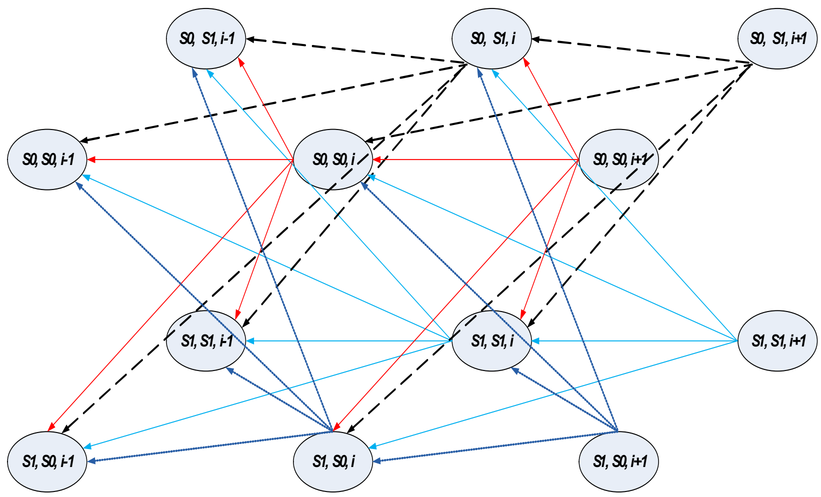


Figure 3.7: D elements in $\tilde{QT}$

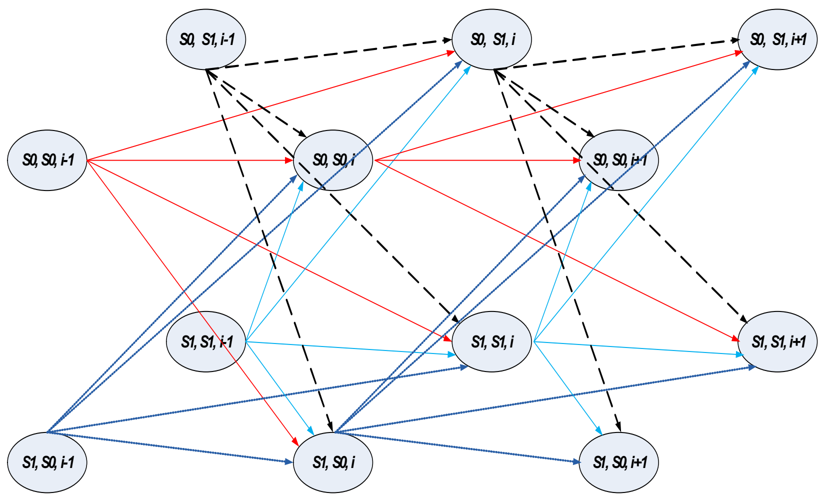


Figure 3.8: $Q(1)$ elements in $\tilde{QT}$

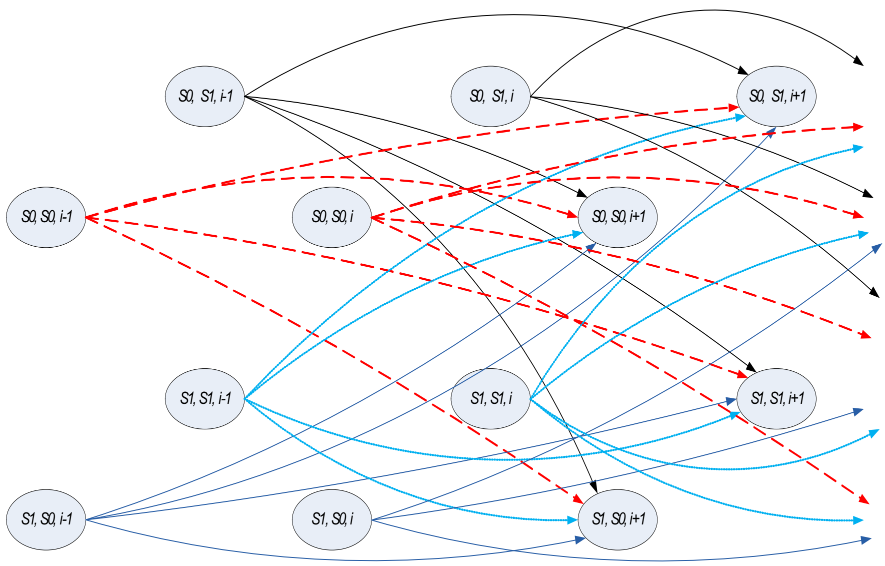


Figure 3.9: $Q(2)$ elements in $\tilde{QT}$

To give a clearer picture of the state transition of the queue length in the system, the queue length state transition matrix $(\tilde{QT})$ is shown in Eq. 3.6. The queue size K can be any size greater than zero. For simplicity of explanation and illustration of state transition matrix, $K = 7$ is used. The state transition matrix can be divided into three cases that are decided by the number of MMBP-2 sources (N) being superposed and the queue size (K). The notations of the queue length state transition are explained in Table 3.1.

$$
\tilde{QT} = \begin{cases}
\begin{bmatrix}
R' & Q(1)' & Q(2)' & Q(3)' & Q(4)' & 0 & 0 & 0 \\
D & R & Q(1) & Q(2) & Q(3) & Q(4) & 0 & 0 \\
0 & D & R & Q(1) & Q(2) & Q(3) & Q(4) & 0 \\
0 & 0 & D & R & Q(1) & Q(2) & Q(3) & Q(4)'' \\
0 & 0 & 0 & D & R & Q(1) & Q(2) & Q(3)^* \\
0 & 0 & 0 & 0 & D & R & Q(1) & Q(2)^* \\
0 & 0 & 0 & 0 & 0 & D & R & Q(1)^* \\
0 & 0 & 0 & 0 & 0 & 0 & D & R^*
\end{bmatrix}_{(K+1,K+1)} & for\ N < K \\
\begin{bmatrix}
R' & Q(1)' & Q(2)' & Q(3)' & Q(4)' & Q(5)' & Q(6)' & Q(7)' \\
D & R & Q(1) & Q(2) & Q(3) & Q(4) & Q(5) & Q(6)^* \\
0 & D & R & Q(1) & Q(2) & Q(3) & Q(4) & Q(5)^* \\
0 & 0 & D & R & Q(1) & Q(2) & Q(3) & Q(4)^* \\
0 & 0 & 0 & D & R & Q(1) & Q(2) & Q(3)^* \\
0 & 0 & 0 & 0 & D & R & Q(1) & Q(2)^* \\
0 & 0 & 0 & 0 & 0 & D & R & Q(1)^* \\
0 & 0 & 0 & 0 & 0 & 0 & D & R^*
\end{bmatrix}_{(K+1,K+1)} & for\ N = K \\
\begin{bmatrix}
R' & Q(1)' & Q(2)' & Q(3)' & Q(4)' & Q(5)' & Q(6)' & Q(7)'^* \\
D & R & Q(1) & Q(2) & Q(3) & Q(4) & Q(5) & Q(6)^* \\
0 & D & R & Q(1) & Q(2) & Q(3) & Q(4) & Q(5)^* \\
0 & 0 & D & R & Q(1) & Q(2) & Q(3) & Q(4)^* \\
0 & 0 & 0 & D & R & Q(1) & Q(2) & Q(3)^* \\
0 & 0 & 0 & 0 & D & R & Q(1) & Q(2)^* \\
0 & 0 & 0 & 0 & 0 & D & R & Q(1)^* \\
0 & 0 & 0 & 0 & 0 & 0 & D & R^*
\end{bmatrix}_{(K+1,K+1)} & for\ N > K
\end{cases}
\tag{3.6}
$$

Table 3.1: Notations for Queue Length Transition Probability Matrix (*Eq.* 3.6)

Symbols	Explanation
$R^{'}$	Special case where queue length will remain unchanged as no packet arrival and no packet departure (empty queue).
R	Queue length will remain unchanged in the following scenarios: 1) No packet departure and no packet arrival at that TS 2) 1 packet arrival and 1 packet departure at that TS
R^{*}	Special case where queue length will remain unchanged in the following scenarios: 1) At least $\geqslant 1$ packets arrive into the queue at that TS with or without packet departure 2) No packet departure and no packet arrival at that TS
D	1 packet departure with no packet arrival at that TS.
$Q(n)^{'}$	Special case where queue length will be incremented by n with n packets arrival, no packet departure (empty queue).
$Q(n)^{'*}$	Special case where no packet departure (empty queue) and queue length will be incremented by n with $> n$ packets arrive into the queue at that TS.
$Q(n)$	Queue length is incremented by n in the following scenarios: 1) n packets arrival and no packet departure at that TS 2) $n + 1$ packets arrival and one packet departure at that TS
$Q(n)^{"}$	Queue length is incremented by n where $n = N$ and no packet departure. Note: N is the maximum number of MMBP-2 sources.
$Q(n)^{*}$	Special case where queue length will be incremented by n even with $\geqslant n$ packets arrive into the queue at that TS in the following scenarios, the equation ($current\ queue\ length + n = K$) must be satisfied. Note: K is the maximum queue size. 1) n packets arrival and no packet departure 2) With $> n$ packets arrival with or without packet departure

Each of the state transition elements ($R/D/Q$) in the queue length state transition matrix is a 2^N by 2^N matrix which are represented as $R^{'}$ / R / R^{*}, D, $Q(n)^{'}$ / $Q(n)$ / $Q(n)^{"}$ /$Q(n)^{*}$ / $Q(n)^{'*}$ in $\tilde{QT}$ (Eq. 3.6). There are a total of N MMBP-2 sources with each MMBP-2 has two distinct states that can be encoded or represented by a one bit binary value 0 (for state $S0$) and 1 (for state $S1$), so that there will be 2^N combinations of state for N MMBP-2. The complexity of the queuing analysis increases with N increases. R represents the state transition where queue length remain unchanged; D represents the state transition where queue length is decreased by one due to a packet departure; while $Q(n)$ state transition means that queue length is increased by n ($1 \leqslant n \leqslant N$). Figs. 3.6 - 3.9 illustrate the queue length state transition elements with 2 MMBP-2 as arrival sources.The queue length state transition element can be obtained through steps below:

- Derivation of state transition probabilities matrix ($\tilde{S}$) (Eq. 3.7) for N MMBP-2 step by step through Eqs. 3.11 - 3.12.

$$\tilde{S} = \begin{bmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,2^N} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,2^N} \\ \vdots & \vdots & \cdots & \vdots \\ s_{2^N,1} & s_{2^N,2} & \cdots & s_{2^N,2^N} \end{bmatrix}_{2^N \times 2^N} \tag{3.7}$$

- Coupling $\tilde{S}$ with arrival or no-arrival probability depending on the state of each MMBP-2 source to derive packet arrivals matrices $\tilde{A}(k)$ of N MMBP-2 (Eq. 3.8) through Eqs. 3.13 - 3.21. k is the number of packets generated at each time slot. Since that the arrival process consists of N MMBP-2 sources, the number of packets generated at each time slot can range from 0 to N. $\tilde{A}(k)$ is the packet arrival matrices for all possibilities of scenarios.

$$\tilde{A}(k) = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,2^N} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,2^N} \\ \vdots & \vdots & \cdots & \vdots \\ a_{2^N,1} & a_{2^N,2} & \cdots & a_{2^N,2^N} \end{bmatrix}_{2^N \times 2^N} \quad ; for\, k = [0..N] \tag{3.8}$$

- Derive each of the queue length state transition elements ($QT_{i,j}$) (a.k.a. $R/D/Q(n)$) by coupling the arrivals matrix with departure (β) or no-departure ($1-\beta$) probability (Eqs. 3.22 - 3.24).

The notations shown in Eqs. 3.9 and 3.10 are the state transition matrix ($\tilde{P}_i$) and arrival probability matrix ($\tilde{\Lambda}_i$) of each MMBP-2 source represented by i with $i = [1..N]$. The state of ith MMBP-2 is denoted as ST_i. The next state transition probability of ith MMBP-2 is denoted as SP_i. It is determined by its current state and next state as shown in Eq. 3.11.

$$\tilde{P}_i = \begin{bmatrix} p_i & 1-p_i \\ 1-q_i & q_i \end{bmatrix}, \, where\, i = 1..N \tag{3.9}$$

$$\tilde{\Lambda}_i = \begin{bmatrix} \alpha_{i1} & 0 \\ 0 & \alpha_{i2} \end{bmatrix}; \, where\, i = 1..N \tag{3.10}$$

$$SP_i = \begin{cases} p_i & if\, current\, ST_i = S0 \\ & \&\, current\, ST_i = next\, ST_i \\ 1-p_i & if\, curent\, ST_i = S0 \\ & \&\, current\, ST_i \neq next\, ST_i \\ 1-q_i & if\, curent\, ST_i = S1 \\ & \&\, current\, ST_i \neq next\, ST_i \\ q_i & if\, curent\, ST_i = S1 \\ & \&\, current\, ST_i = next\, ST_i \end{cases} \tag{3.11}$$

The derivation of next state transition probability element $S_{m,n}$ is shown in Fig. 3.10. Indexes of each element in $\tilde{S}$, which are m and n, actually refer to the binary aggregation of current state and next state for all MMBP-2 sources. ith-bit of binary indexes m and n represents the current or next state of ith MMBP-2 source. Both indexes are turned into binary encoding in order to represent the state of each MMBP-2 source. The state transition probability $S_{m,n}$ is the product of transition probability of all MMBP-2 sources as shown in Eq. 3.12. In Fig. 3.10, the derivation of $s_{5,22}$ with six MMBP-2 sources is taken as an example for the derivation.

$$s_{m,n} = \prod_{i=1}^{N} SP_i \tag{3.12}$$

The packet arrivals matrix ($\tilde{A}(k)$) can be derived by associating the arrival probability or no-arrival probability to each MMBP-2 according to its state and number of arrivals. SRC is a set contains of all MMBP-2 sources. Each MMBP-2 source has the same equality in being selected as the source for packet generation. The MMBP-2 combination sets are subsets of SRC as shown in Eqs. 3.15 and 3.16. The MMBP-2 combinations sets are the list of sources that are randomly selected from the given set of MMBP-2 sources(SRC) for packets generation. The number of possible combinations (G) can be determined by Eq. 3.14. Following this, arrival probabilities of each MMBP-2 source from Eqs. 3.15 - 3.16 are determined to derive $\tilde{A}(k)$ as in Eqs. 3.17 - 3.21.

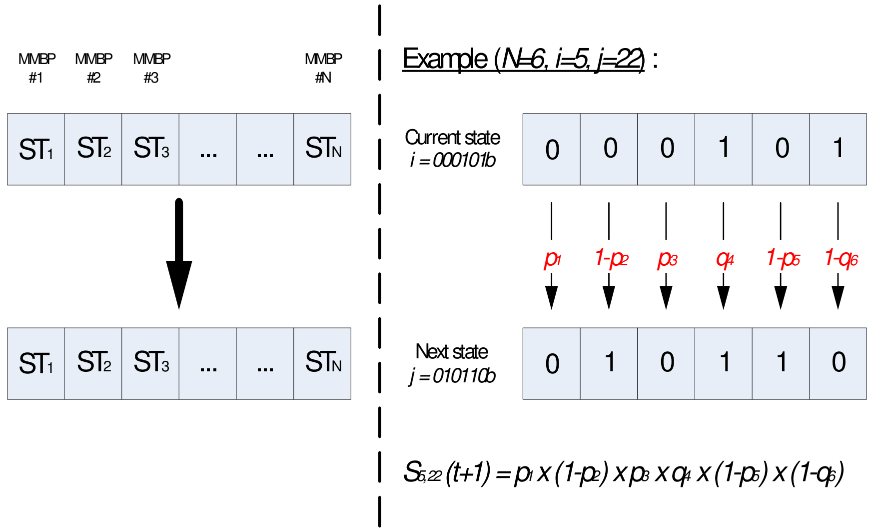


Figure 3.10: Derivation of State Transition Probability Elements, $S_{m,n}$ (Eq. 3.12)

$$SRC = \{1,\, 2,\, ,3,\, \ldots\, ,N\} \tag{3.13}$$

$$G = C_k^N \; ; for\; k = [0..N] \tag{3.14}$$

$$SRC\_C(k) = \begin{bmatrix} x_{1,1} & \cdots & x_{1,k} \\ \vdots & \ldots & \vdots \\ x_{G,1} & \cdots & x_{G,k} \end{bmatrix}_{G,k} \quad ; for\; x_{i,j} \in SRC \tag{3.15}$$

$$\overline{SRC\_C}(k) = \begin{bmatrix} y_{1,1} & \cdots & y_{1,N-k} \\ \vdots & \ldots & \vdots \\ y_{G,1} & \cdots & y_{G,N-k} \end{bmatrix}_{G,N-k} \quad ; for\; y_{i,j} \in SRC \tag{3.16}$$

$$AP(k) = \begin{bmatrix} ap_1 \\ \vdots \\ ap_G \end{bmatrix} \tag{3.17}$$

$$ap_i = \begin{cases} \prod_{j=1}^{k} \alpha_{m1} & ; for\; m = x_{i,j}\; \&\; ST_m(t) = S0 \\ \prod_{j=1}^{k} \alpha_{m2} & ; for\; m = x_{i,j}\; \&\; ST_m(t) = S1 \end{cases} \tag{3.18}$$

$$NAP(k) = \begin{bmatrix} nap_1 \\ \vdots \\ nap_G \end{bmatrix} \tag{3.19}$$

$$nap_i = \begin{cases} \prod_{j=1}^{N-k} (1 - \alpha_{m1}) & ; for\; m = y_{i,j}\; \&\; ST_m(t) = S0 \\ \prod_{j=1}^{N-k} (1 - \alpha_{m2}) & ; for\; m = y_{i,j}\; \&\; ST_m(t) = S1 \end{cases} \tag{3.20}$$

$$a_{i,j} = \sum_{m=1}^{G} \{s_{i,j} * ap_m * nap_m\} \tag{3.21}$$

After all arrivals matrices are obtained, queue length state transition matrix elements $QT_{i,j}$ (from Eq. 3.6 and Table 3.1) can be derived by combining the arrivals matrices with departure or no-departure probability (Eqs. 3.22 - 3.24). The derivation also depends on the current queue state QL (current queue length) and the maximum queue size K.

$$R = \begin{cases} \tilde{A}(0) \equiv R' & if\, QL = 0 \\ \tilde{A}(0) * (1-\beta) + \tilde{A}(1) * \beta & if\, 0 < QL < K \\ \tilde{A}(0) * (1-\beta) + \sum_{i=1}^{N} A(i) \equiv R^* & if\, QL = K \end{cases} \tag{3.22}$$

$$D = \tilde{A}(0) * \beta \tag{3.23}$$

$$Q(n) = \begin{cases} \tilde{A}(n) \equiv Q(n)' & if\, QL = 0 \\ & for\, all\, n\, in\, case\, N \leqslant K;\, and \\ & for\, n < K\, in\, case\, N > K \\ \sum_{i=n}^{N} \tilde{A}(i) \equiv Q(n)'^{*} & if\, QL = 0 \\ & for\, n = K\, in\, case\, N > K \\ \tilde{A}(n) * (1-\beta) + \tilde{A}(n+1) * \beta & if\, QL > 0\, \&\, QL + n < K \\ & for\, all\, cases \\ \tilde{A}(n) * (1-\beta) \equiv Q(n)'' & if\, QL > 0\, \&\, QL + n = K \\ & for\, n = N\, in\, case\, N < K \\ \tilde{A}(n) * (1-\beta) + \sum_{i=n+1}^{N} A(i) \equiv Q(n)^* & if\, QL > 0\, \&\, QL + n = K \\ & for\, case\, n < N\, in\, all\, cases \end{cases} \tag{3.24}$$

The $\tilde{QT}$ matrix (Eq. 3.6) derived from Eqs. 3.7 - 3.24 will then be used in performing steady state queuing analysis. The steady state queuing analysis is performed by using the Markov Chain Solver algorithm [59, 68] to solve the joint steady state probability vector of queue length $\tilde{\pi} = \pi_i(0 \leqslant i \leqslant K)$, which satisfies Eqs. 3.25 and 3.26 where $e = [1, 1, ..., 1]_K^T$ is the column vector of length K. Solving these two equations yields the steady state vector $\tilde{\pi}$ (Eq. 3.28) as derived in Markov Chain Solver algorithm where u is an arbitrary row vector of X and I is an identity matrix with size of K x K. A closed-form function derivation approach is not adopted due to the complexity of the queuing model and also a huge number of parameters involved. Numerical analysis in Matlab is used instead.

$$\tilde{\pi}.\tilde{QT} = 0 \tag{3.25}$$

$$\tilde{\pi}.e = 1 \tag{3.26}$$

$$X = I + \tilde{QT}/min\{QT_{i,j}\} \tag{3.27}$$

$$\tilde{\pi} = u(I - X + eu)^{-1} \tag{3.28}$$

3.4 DTH Limitations and Assumptions

Here is a list of limitations and assumptions for the DTH queue management scheme on top of the assumptions made for the queuing analysis:

1. DTH is designed to bound average queuing delay for aggregated traffic flows; per flow treatment is not part of the goal. Therefore, the fairness issue is not considered by DTH.

2. It is not the main aim of DTH to alleviate congestion, however congestion control can be achieved as a side goal from DTH which drops packets that stay in the queue beyond the required average queuing delay. The packet loss event can serve as a congestion indicator for responsive traffic.

3. The focus of DTH is to show that dynamic queue threshold concept can be used to bound queuing delay if such a relationship between queuing thresholds and queuing delays can be derived. The mapping of the MMBP-2 traffic models to real traffic traces is not the focus of this research.

4. The analytical model proposed for the derivation of mapping relationship between a target queue threshold and queuing delay is complex, thus a closed form equation could not be obtained. The analytical analysis was carried out on a Pentium 4 personal computer, which is equipped with a $2.4GHz$ processor and $1GB$ RAM, under Windows XP operating system. The computation time for the analytical model with $N = 8$ is around 15 minutes. Due to the complexity of the model and the limitation of computer specification, the derivation process in Matlab cannot be solved when N greater than 8.

3.5 Simulation Results

Simulation of the DTH scheme with multiple MMBP-2 sources has been carried out using Matlab simulation. Common performance metrics used in the discrete-time queuing model, such metrics are system utilization (ρ), average throughput ($\bar{S}$), average queue length ($\bar{L}$) and average queue delay ($\bar{D}$), can then be derived easily with the resolution of queue length steady state probability vector [67, 68, 177].

System utilization can be derived easily by knowing the probability distribution of the queue length. As long as the queue is not empty then the system is not idle. Thus system utilization equation can be written as Eq. 3.29 where p_i denotes the probability with i packets in the queue and p_0 denotes the probability that the queue is empty.

$$\rho = 1 - p_0 \tag{3.29}$$

Throughput of the system can then be derived using system utilization and packet departure probability (β) as in Eq. 3.30.

$$\bar{S} = \rho.\beta \tag{3.30}$$

Average queue length can also be obtained easily from probability distribution of queue length in that system where L is the maximum number of packets in the queue (maximum queue length). The calculation is given in Eq. 3.31.

$$\bar{L} = \sum_{i=1}^{L} i.p_i \tag{3.31}$$

Little's law [177] is then used to derive average queue delay in the system. Little's law is a theorem that specifies a relation between the mean waiting time of a customer in a system, the long-term average number of customers arrives into that system, and the mean number of customers held in the system queue. By applying the Little's law into communication systems, average number of customer arrivals is average packets arrival rate (λ), mean waiting time is referring to mean queuing delay (W) and lastly the number of customers held in the queue is equivalent to mean queue length (L). The relationship can be stated as Eq. 3.32. From the Eq. 3.33, average queue delay is derived and can be written as Eq. 3.34.

$$\bar{L} = \lambda \bar{W} \tag{3.32}$$

$$\bar{W} = \frac{\bar{L}}{\lambda} \tag{3.33}$$

Table 3.2: Configuration for DTH Queuing Analysis and Simulation

Scenarios	Experiment Configurations
Scenario 1	MMBPs with same configuration, N = [1..5] $\alpha_1 = \frac{0.4}{N}$; $\alpha_2 = \frac{0.5}{N}$; $p = 0.9999$; $q = 0.9999$ Departure probability, $\beta = 0.5$ Required delay, $TD = 7$
Scenario 2	MMBPs with different configuration, N = 3 MMBP #1: $\alpha_1 = 0.1$; $\alpha_2 = 0.25$; $p = 0.9999$; $q = 0.9999$ MMBP #2: $\alpha_1 = \alpha_2 = 0.2$; $p = 0.9$; $q = 0.9$ MMBP #3: $\alpha = 0.15$; $\alpha_2 = 0$; $p = 0.5$; $q = 0.5$ Departure probability, $\beta = 0.5$ Required delay, $TD = [5..9]$ with stepping 1
Scenario 3	MMBPs with different configuration, N = 3 MMBP #1: $\alpha_1 = 0.30$; $\alpha_2 = 0.45$; $p = 0.9999$; $q = 0.9999$ MMBP #2: $\alpha_1 = \alpha_2 = [0.1..0.3]$ with stepping 0.05; $p = 0.9$; $q = 0.9$ MMBP #3: $\alpha = 0.15$; $\alpha_2 = 0$; $p = 0.5$; $q = 0.5$ Departure probability, $\beta = [0.6..0.8]$ with stepping 0.05 Required delay, $TD = 7$

$$\bar{D} = \frac{\bar{L}}{\bar{S}} \tag{3.34}$$

In order to validate and test the feasibility of the proposed scheme, the simulation has been carried out based on the scenarios listed in Table 3.2 with the following configurations: $K = 60$, $TW = 200$. Note that each $TW = 10{,}000$ *TS*. Three simulation scenarios are discussed in the following subsections.

3.5.1 Scenario 1: Different number of sources

Theoretical results have been validated via simulation results as shown in Fig. 3.11. The figure presents the mean queuing delay experienced in DTH versus the target delay with aggregated traffic flows from same traffic class and with increasing number of traffic flows. The simulation results match the analytical result with small Squared Coefficient of Variation (SCV) and Mean Squared Error (MSE) (Table 3.3). DTH is able to maintain the average queuing delay to the target delay with different number of sources. It can be seen that the queuing threshold is adjusted dynamically from time to time in Fig. 3.12. The fluctuation is due to the adaptive nature of queue threshold.

The simulation results show that the system could achieve the expected throughput for different number of traffic sources (Fig. 3.12). However, the packet loss

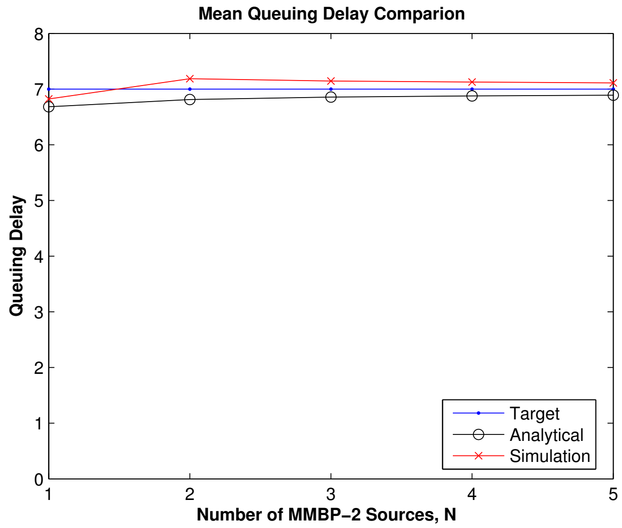


Figure 3.11: Scenario 1: DTH Validation

Table 3.3: Scenario 1: SCV and MSE for DTH

N	1	2	3	4	5
SCV	0.1379	0.1619	0.3065	0.3482	0.3144
MSE	0.1697	0.1963	0.3274	0.3639	0.3267

ratio increases as the number of traffic sources increases (Fig. 3.13). This can be explained by the burstiness of the arrival process in terms of number of packets generated at each time slot. With the increasing number of traffic sources, more packets are generated at each time slot and more packets are being dropped due to queue full.

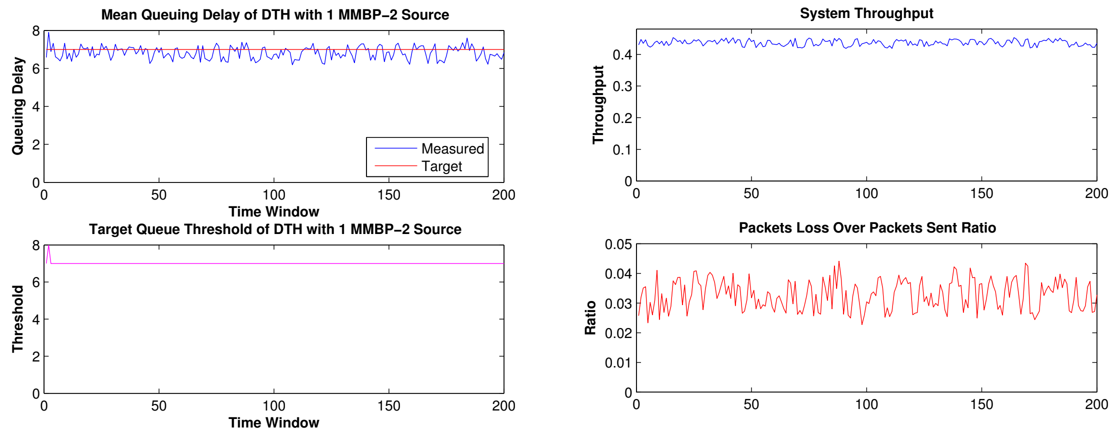
3.5.2 Scenario 2: Different required average queuing delay

Apart from supporting multiple traffic sources, DTH is capable of maintaining the average queuing delay to different TD as shown in Fig. 3.14. This is further proven with small SCV and MSE values (Table 3.4) from the simulation.

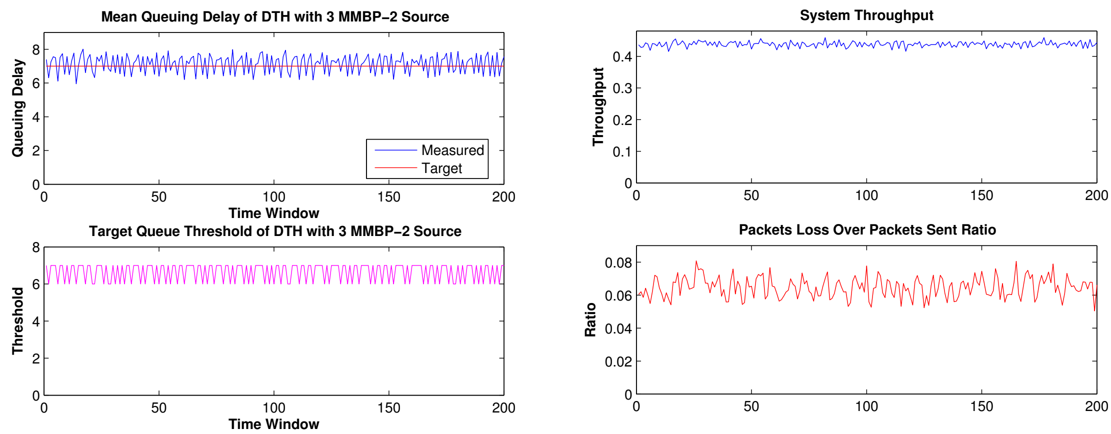
Table 3.4: Scenario 2: SCV and MSE for DTH

TD	5	6	7	8	9
SCV	0.3907	0.3456	0.2585	0.2638	0.4475
MSE	0.6395	0.4318	0.2636	0.2914	0.6097

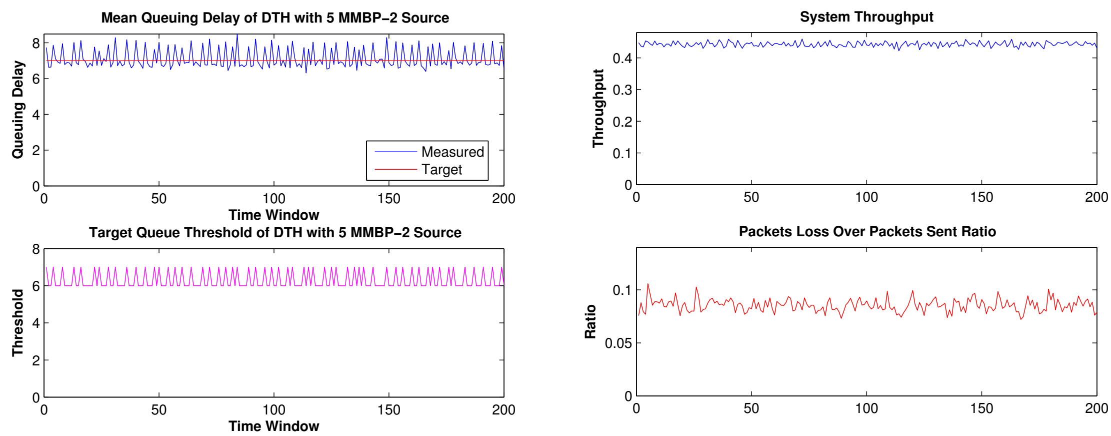
For Scenario 2, Figs. 3.15 and 3.16 show that packet loss ratio decreases when the required delay value increases. This is due to the upper queuing threshold is relaxed along with the increasing of required delay value. This means more rooms available in the buffer queue for packets enqueuing. Consequently, packet loss ratio decreases.



(a) N = 1



(b) N = 3



(c) N = 5

Figure 3.12: Scenario 1: System Performance of DTH

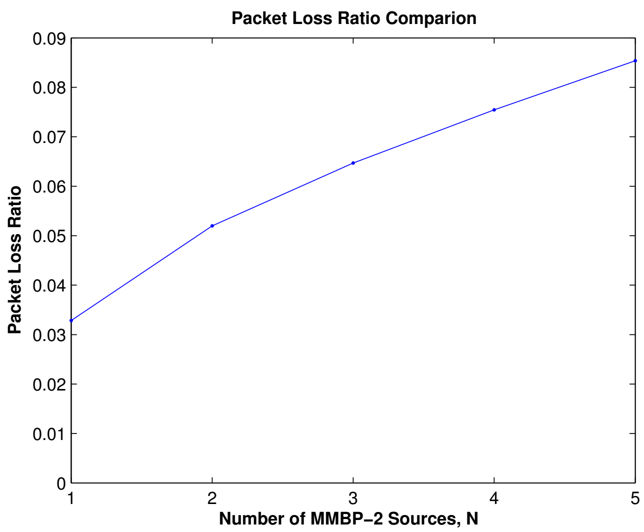


Figure 3.13: Scenario 1: Packet Loss Ratio Comparison

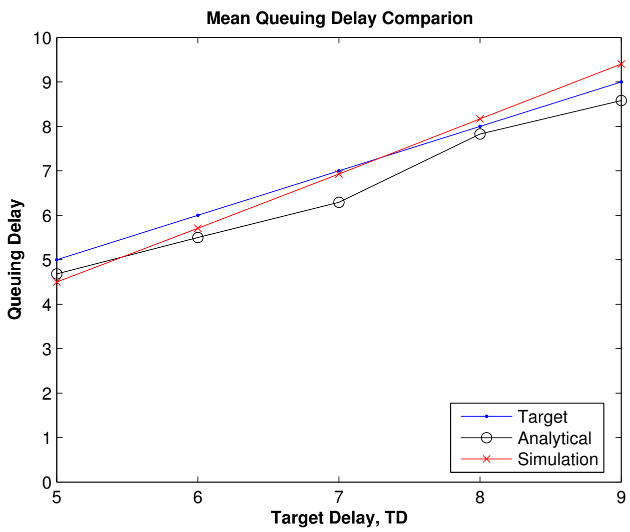
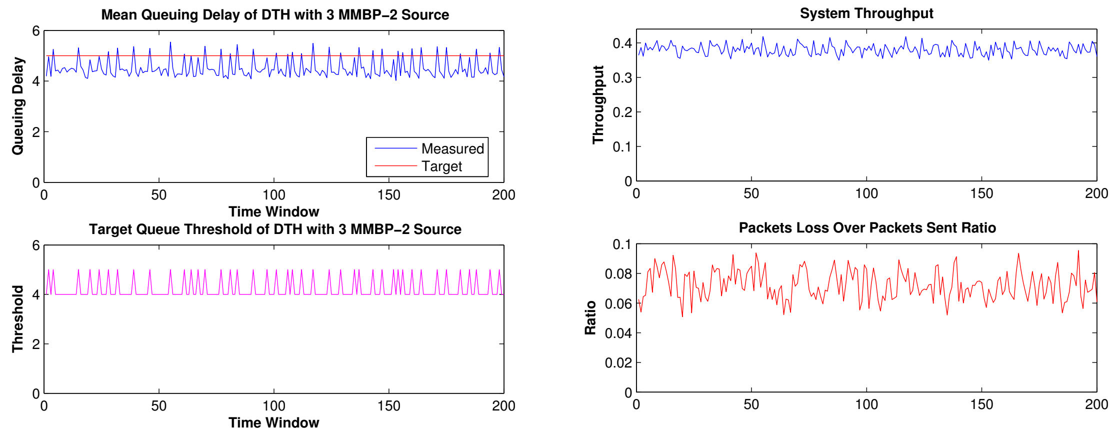


Figure 3.14: Scenario 2: DTH Validation

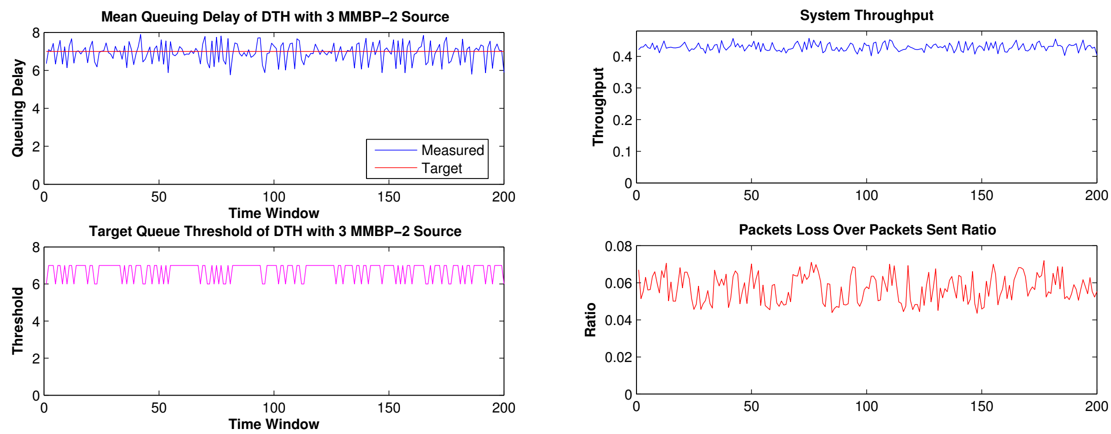
3.5.3 Scenario 3: Different service rate

Scenario 3 (Fig. 3.17) shows the feasibility of the DTH scheme to maintain the average queuing delay by taking into account of service rate. These are further proven with small SCV and MSE values (Table 3.5) from the simulation.

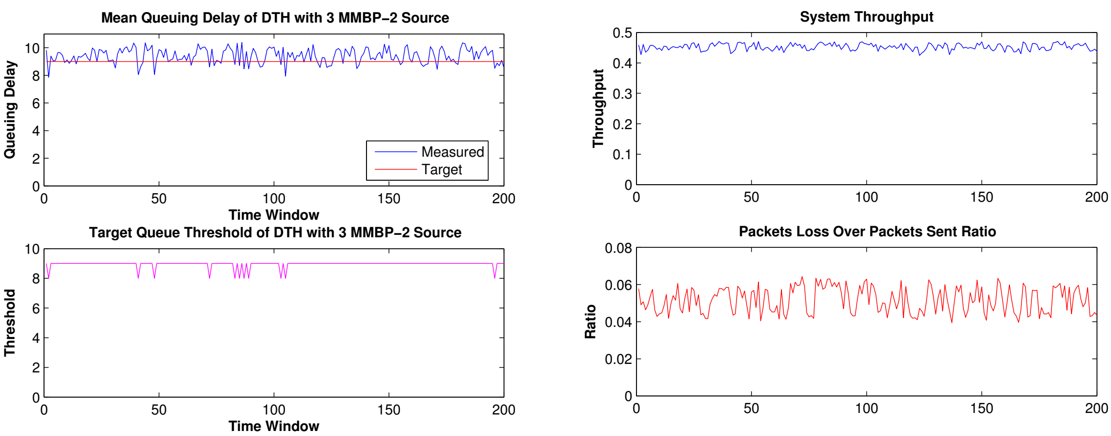
Similarly for Scenario 3, Figs. 3.18 and 3.19 show that packet loss ratio has a trend of decreasing when the system service rate increases. With the faster rate of packets processing in the system, the queue is drained up faster and leaves more rooms in the buffer queue for new incoming packets. Packet loss ratio decreases as a consequence of this.



(a) $TD = 5$



(b) $TD = 7$



(c) $TD = 9$

Figure 3.15: Scenario 2: System Performance of DTH

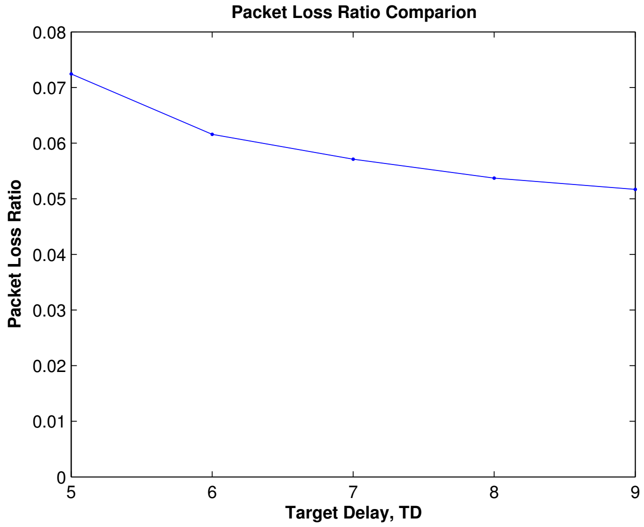


Figure 3.16: Scenario 2: Packet Loss Ratio Comparison

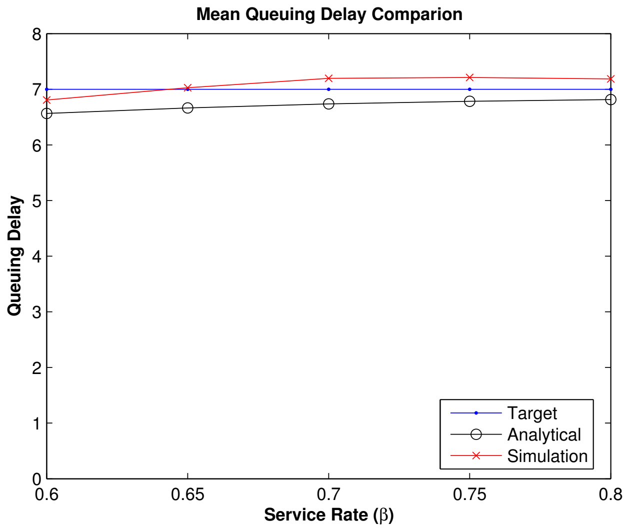
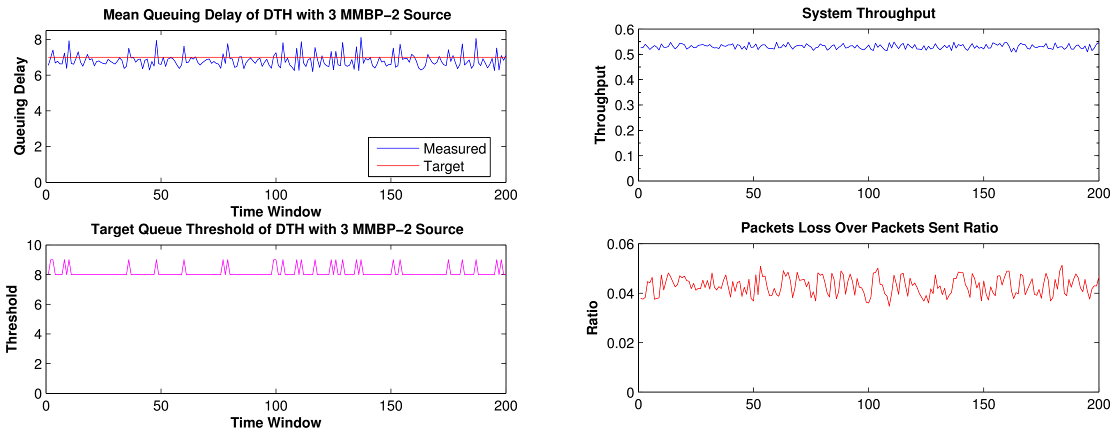


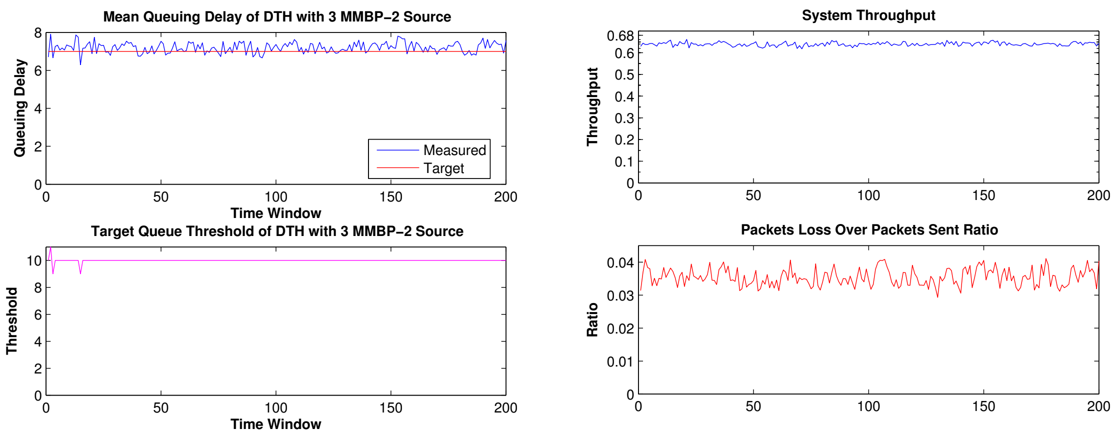
Figure 3.17: Scenario 3: DTH Validation

Table 3.5: Scenario 3: SCV and MSE for DTH

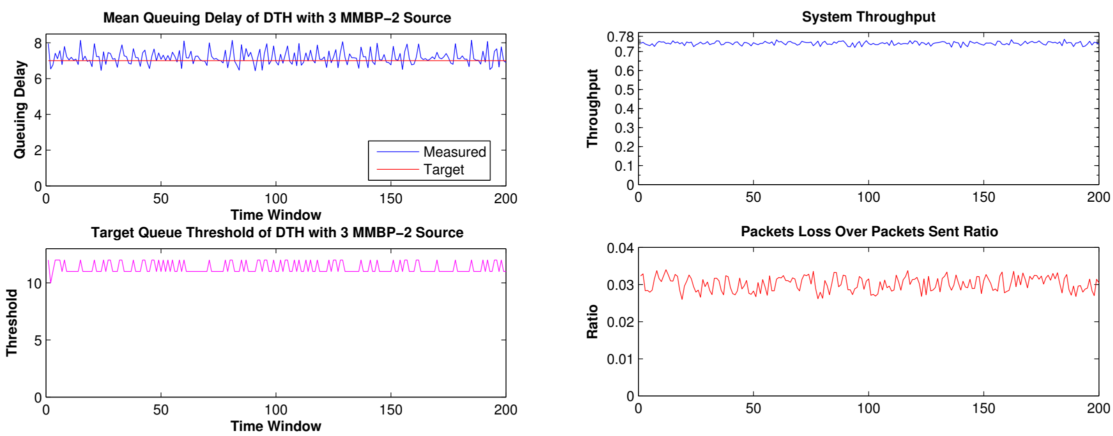
β	0.60	0.65	0.70	0.75	0.80
SCV	0.1784	0.0508	0.1112	0.1959	0.1919
MSE	0.2154	0.0516	0.1497	0.2408	0.2261



(a) Service Probability = 0.6



(b) Service Probability = 0.7



(c) Service Probability = 0.8

Figure 3.18: Scenario 3: System Performance of DTH

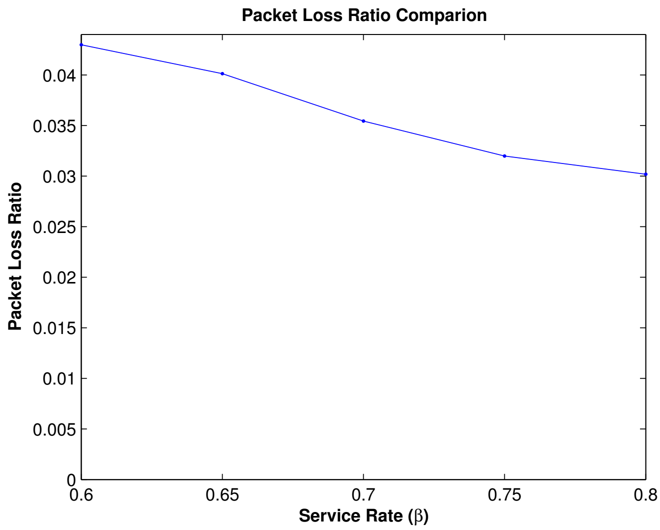


Figure 3.19: Scenario 3: Packet Loss Ratio Comparison

3.6 Summary and Discussions

This chapter describes an adaptive queue management scheme (DTH) that constrains the average queuing delay of a queue to a required value by adjusting the target queue threshold dynamically. The target queue threshold is estimated from a discrete time queuing analysis that uses superposition of N MMBP-2 arrival processes to model aggregated Internet traffic. Packets are dropped when the queue length exceeds the target queue threshold. The packet loss events can serve as implicit congestion indicators for responsive sources but not for delay-sensitive traffic sources that are non-responsive to the packet loss event.

The simulation results show that the proposed DTH solution can bound average queuing delay in a router via a movable queuing threshold using a closed-loop feedback control mechanism. However, the implementation highly relies on the mapping relationship obtained from the discrete-time queue analysis, which is complex and can only be solved by using a numerical analysis, since a closed-form equation could not be derived. The mapping tables need to be generated a priori for all possible scenarios.

Predicting arrival process to get an accurate model of real traffic profiles is a non-trivial task. Wireless traffic measurement and profiles are hard to be obtained due to difficulties in traffic capturing and the network operators may not willing to disclose all the information for the traffic traces [69]. Besides that, data streams for a wireless ad hoc network are unpredictable as the wireless ad hoc network may be setup for various purposes such as data sharing in a meeting, data communication at a disaster scene, etc. Therefore, the analytical approach of DTH may not be suitable for wireless ad hoc networks.

Furthermore, system performance of a wireless ad hoc network is not stable as compared to a wired network. The system performance of a wireless ad hoc network is highly affected by environmental interference and interference caused by neighbour nodes due to the nature of shared wireless medium for transmission. Therefore, DTH needs to be extended in order to take into consideration of wireless characteristics.

If the analytical approach similar to DTH is taken for queue management in the wireless domain, the complexity of the analytical model will be multi-folded by considering aforementioned factors. This may not be a good solution for wireless nodes with limited resources. Besides the complexity, a self-adaptive queue management scheme is needed in order to respond to the network dynamics in wireless domain. DTH regulates the queue based on the estimated queuing threshold derived a priori from the analytical model. Hence, DTH is not self-adaptable to the changes in the network. Nevertheless, the concept of regulating a queue with a dynamic threshold should be brought forward into the wireless domain.

DTH needs to be extended from the analytical analysis approach to an online measurement-based in order to make it adaptive to wireless networking environment. Therefore, a detailed delay analysis on a small wireless ad hoc network has been carried out and discussed in the next chapter to find out the impact factors and design factors that should be considered for the online measurement-based version of DTH.

Chapter 4

Delay Analysis for Wireless Ad Hoc Network

4.1 Introduction

In this chapter, OWD for a small wireless ad hoc network has been analyzed. The delay analysis aims to show the significance of queuing delay that contributes to nodal delay and OWD. The analysis highlights why a queue management scheme with a dynamic threshold is indeed a promising mechanism to constrain network delay in wireless ad hoc networks.

At present, wired communication technology is more matured than wireless communication technology. Data transfers are faster and links capacity are greater in wired networks compared to wireless networks. Wired networks are not susceptible to interference from neighbour nodes and environment. Therefore, throughput of a wired network is more predictable and stable.

In contrast to wired networks, wireless networks are dynamic. The network performance of wireless networks fluctuates over time. Therefore, network delay experienced by packets sent over wireless networks is harder to be controlled. Understanding the network latency introduced at node level is important to design an adaptive queue management scheme for wireless networks especially wireless ad hoc networks.

The delay analysis has been carried out with the NS-2 [80] simulator. NS-2 is chosen since it is the most popular network simulation tool among the networking research community [73]. Transceiver configurations in the simulation follow the datasheet of MRVL-88W8385 [168] to match the configurations of wireless nodes in the testbed (Chapter 6).

The remainder of the chapter is organized as follows: Section 4.2 estimates delay components based on the theoretical limit; Section 4.3 describes briefly the simulation setup; Section 4.4 presents the simulation results and finally Section 4.5 summarizes the findings.

4.2 Impact Factors on Network Delay

Section 2.2 shows that nodal delay consists of a few delay components, such as processing delay, queuing delay, MAC contention delay, transmission delay and propagation delay. Therefore, a few factors that affect the magnitude of OWD are analyzed briefly based on an ideal scenario and theoretical limits in this section. Processing delay is not considered in this delay analysis since that NS-2 assumes no processing overhead ($D_{Proc} = 0$). Other than that, propagation delay is taken as a constant and negligible in this analysis since that propagation delay is only impacted by the distance between two nodes and propagation speed of the medium ($\equiv$ light speed in this case) as shown in Eq. 2.2. It is hard to derive MAC contention delay theoretically, therefore this section only focuses on queuing delay and transmission delay.

4.2.1 Transmission Delay

Based on Eq. 2.1, transmission delay is only affected by the packet size and the transfer data rate. Fig. 4.1 shows the transmission delay for a packet with different packet sizes and under different transfer data rates. The transmission delay increases significantly when the transfer data rate is low and when the packet size becomes larger. Nevertheless, the transmission delay can be considered as minimal impact to the OWD since the magnitude of the transmission delay is quite small.

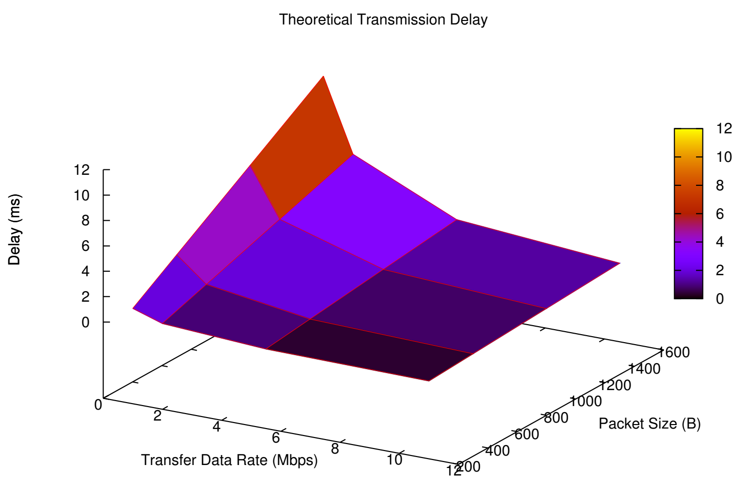


Figure 4.1: Theoretical Transmission Delay

4.2.2 Queuing Delay

The maximum queue size and packet size are important elements for queuing delay calculation. Apart from that, system throughput (which is how fast can a packet be dequeued and transmitted out) is another important factor that contributes to queuing delay.

By taking the default maximum queue size as 1000 packets, theoretical queuing delay for a wireless node under different transfer data rates and packet sizes are calculated. The maximum queuing delay is calculated by assuming ideal maximum throughput for a wireless node. Under ideal scenario (Fig. 4.2), the maximum queuing delay observed has the same trend as transmission delay. When a wireless node transfers data at a lower rate, such as 1 mega bits per second ($Mbps$) or $2Mbps$, the maximum queuing delay can range from 2 seconds (s) to $11s$ even the destination is just one hop away when the IFQ is full and the packet size is $>=$ 500 bytes (B). This is unacceptable for delay-sensitive traffic especially the traffic may have to travel multiple hops before reaching its destination. If the cumulative of queuing delay along the path from a source node to a destination node is large, then delay-sensitive traffic may not reach its destination on time. This leads to degradation of QoS.

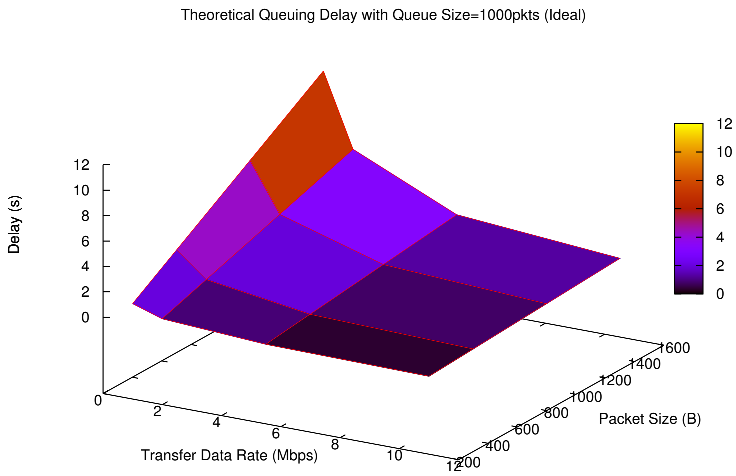


Figure 4.2: Theoretical Queuing Delay with Queue Size = 1000 Packets

Moreover, the ideal maximum throughput could not be achieved since that packets are transmitted over a shared and error-prone medium. The usable throughput of 802.11 system is lesser than the raw bandwidth; almost half of the bandwidth is wasted in various aspects such as encoding, protocol overhead, channel contention, etc [11, 46, 87]. Taking the performance measurements done by Anastasi *et al.* [11] for UDP traffic, the maximum queuing delay with packet size of $1024B$ for different queue sizes are calculated and shown in Fig. 4.3. It can be concluded that the maximum queue size and the actual throughput of a wireless node are important factors in controlling queuing delay apart from packet size.

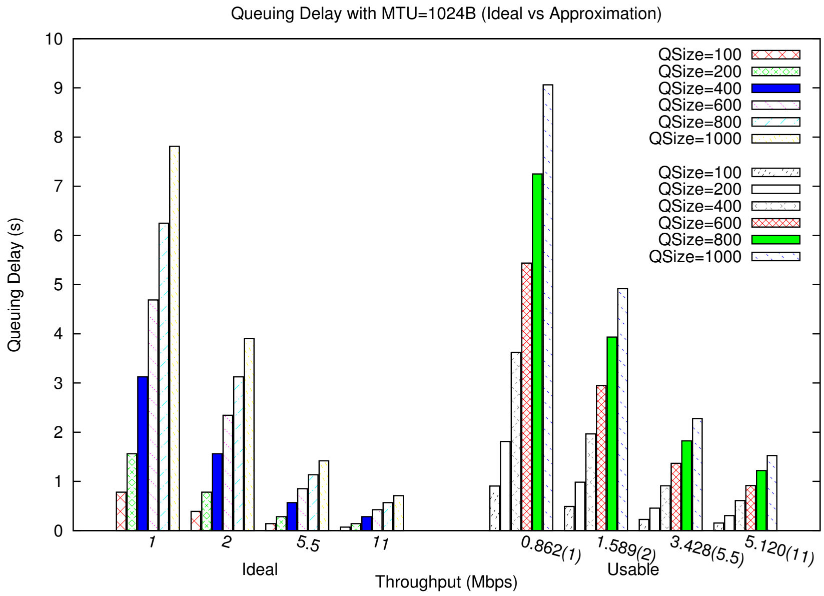


Figure 4.3: Queuing Delay Comparison for Different Queue Sizes

4.3 Simulation Configurations

NS-2 simulation configurations are listed in Table 4.1. Fig. 4.4 shows the simulation network topology. Node 0 (left most) competes for channel access with Node 1 (middle) and Node 2 (right most) competes for channel with Node 1. Therefore, Node 1 needs to compete with Node 0 and Node 2 for channel access.

Table 4.1: NS-2 Simulation Configurations

Parameters	Configurations
Radio Propagation Model	TwoRayGround
Wireless Mode	IEEE 802.11b
Interface Queue	DropTail/PriQueue
Routing Protocol	AODV
Virtual Carrier Sensing	OFF
Transmit Power	15dBm
Transmission Range	30m
Carrier Sensing Range	$\gg$ 2x transmission range
Transmission Data Rate	$11Mbps$ (no auto-fallback)

Most of the WLAN cards support multiple transfer data rates with an auto-fallback mechanism when interference increases. A transceiver will try to send

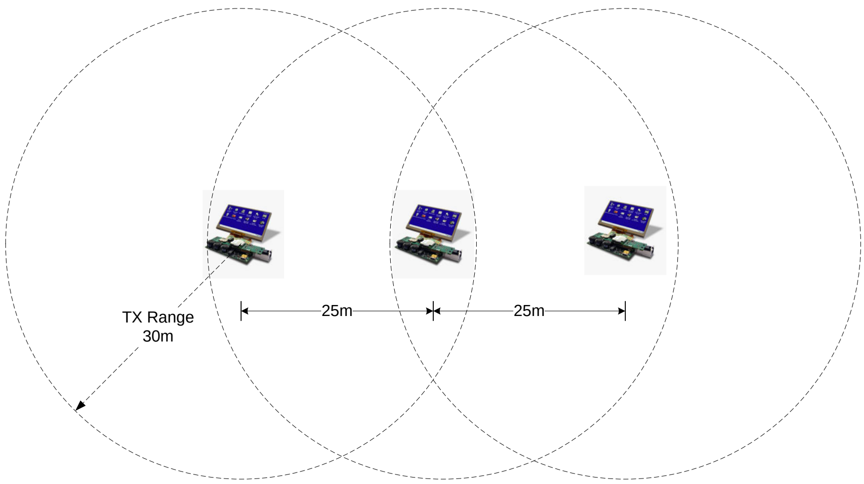


Figure 4.4: Simulation Network Topology

data at the highest transfer rate supported by the hardware and the standard; but the transfer rate will be stepped down when noise increases in order to reduce packet loss. For the IEEE 802.11b standard, the basic transfer rate ($1Mbps$) is the lowest transfer rate supported. All management or control frames (e.g RTS/CTS frames, ACK frame) are sent using the basic transfer rate. The data transfer rate may take the value of 1, 2, 5.5 or $11Mbps$. However, NS-2 simulation does not support the auto-fallback feature. Therefore, the delay analysis has been carried out by using a fixed transfer data rate that is $11Mbps$.

The default queue size in NS-2 is 50 packets. Whereas the default queue size for a Gumstix node, which runs Embedded Linux operating system, is 1000 packets. Therefore, the default IFQ size in the simulation is changed to Gumstix configuration. Default queue discipline is Drop Tail FIFO.

Fig. 4.5 shows the traffic load being injected into the network during the simulation. The simulation starts with Node 0 sending CBR traffic at rate R to Node 2. After $25s$ Node 2 starts the CBR transmission to Node 0 at the same data rate. At $50s$, Node 0 reduces its rate to half and then continue at this rate to the end of the simulation. While Node 2 reduces its rate to half at $75s$, and continue this rate towards the end of the simulation. From this simulation, impacts of single directional and bi-directional traffic and intensity of traffic load are simulated.

4.4 Simulation Results

Wireless nodes transmit packets on a shared medium. Therefore, neighbour nodes within the sensing range of a wireless node and traffic load in the network may cause interference to that wireless node. This reduces the chance of that wireless

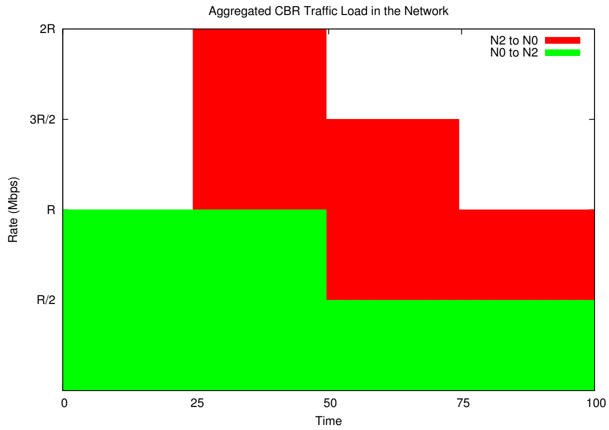


Figure 4.5: Traffic Load for Simulation

node to grasp the channel for transmission. In this subsection, the performance impact due to interference from neighbour nodes and traffic load in the network has been analyzed. Besides that, scenarios with different queue sizes and packet sizes have also been simulated. The OWD, queuing delay, UDP goodput, and packet loss are analyzed for various scenarios based on the common simulation configurations stated in Section 4.3.

4.4.1 Scenario 1: Different traffic load

For scenario 1, the simulation has been carried out with different traffic loads but with a fixed queue size (QS) of 1000 packets and a constant packet size (PS) of $1000B$. CBR data rate (R) is varied from $1.0Mbps$ to $2.0Mbps$ to simulate scenarios with light to heavy traffic load in the network.

The simulation results are summarized in Table 4.2. When $R > 1.2Mbps$, the network starts saturating gradually as shown by the increase of collision counts. Backlogs start to build up in the interface queue (IFQ) when the traffic load is higher. The wireless nodes cannot cope with the faster data rate. As a result, packet loss increases due to queue overflow.

Besides packet loss, the magnitude of OWD is another indicator of network congestion. OWD increases drastically due to channel contention and backlogs in the IFQ. The OWD is insignificant when the traffic load is light. When $R \geqslant 1.8Mbps$, the network becomes highly saturated and the maximum OWD recorded is greater than $8s$.

Throughout the simulation, Node 1 has the lowest collision count (Fig. 4.6) and hence it can be deduced that Node 1 wins in channel contention. From Fig. 4.7, IFQ is saturated for Node 1 when $R \geqslant 1.3Mbps$. The IFQ of Node 2 and Node 0 are only saturated when $R \geqslant 1.6Mbps$ and $R \geqslant 1.8Mbps$. Node 1's IFQ is saturated before Node 0 and Node 2 as Node 1 needs to buffer and forward packets from

Table 4.2: Scenario 1: Overall System Performance with Different Traffic Loads

R ($Mbps$)	Collision Count	Full Q Drop	Max OWD (s) N0→N2	Max OWD (s) N2→N0	Packets Delivery (%) N0→N2	Packets Delivery (%) N2→N0
1.0	0	0	0.0117	0.0232	100	100
1.2	0	0	0.0137	0.0330	100	100
1.4	274	360	3.2748	3.2971	98.6	98.5
1.6	479	2118	4.7655	4.9781	93.1	92.0
1.8	629	3936	8.2236	8.5012	91.2	83.7
2.0	601	6271	8.8317	8.9273	87.8	76.1

both Node 0 and Node 2.

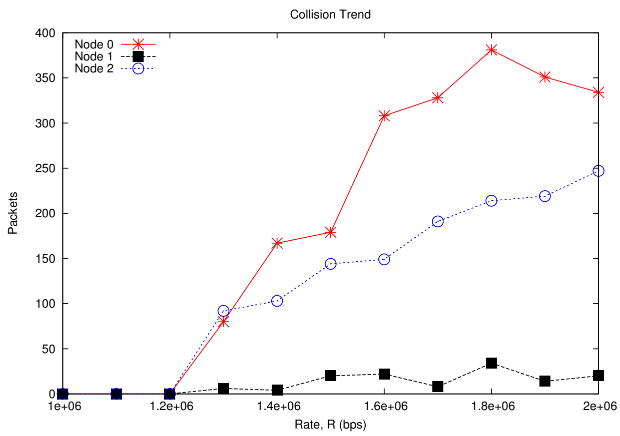


Figure 4.6: Scenario 1: Collision Trend for Different Traffic Loads

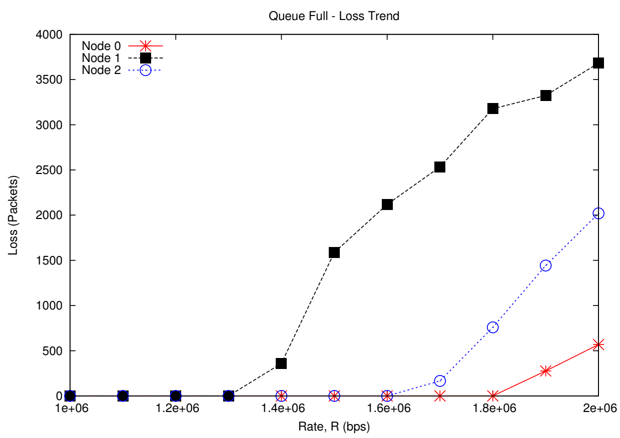
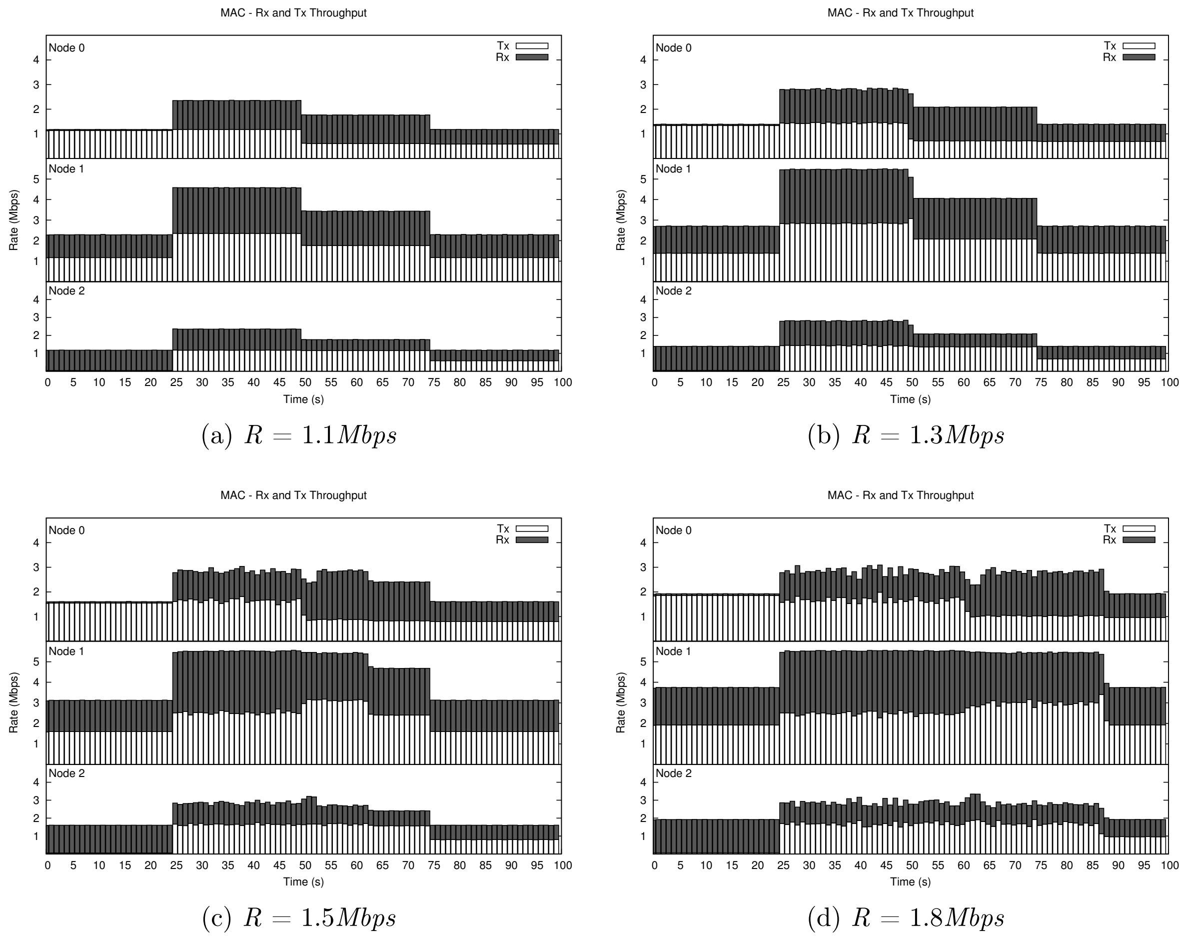


Figure 4.7: Scenario 1: Queue Full Loss Trend for Different Traffic Loads

A closer look into the system performance is then taken to give a full picture on why the OWD increase tremendously after certain thresholds. A few simulation sets with different R rate ($=1.1$, 1.3, 1.5, 1.8$Mbps$) are chosen and discussed to give a snapshot of the network at different traffic loads.

Fig. 4.8 shows that Node 1's throughput at MAC level is almost twice of Node 0 and Node 2 for all cases. This is due to Node 1 is a relay node which forwards packets at both directions to Node 0 and Node 2. When traffic load is light (R $=1.1Mbps$), there is no congestion and therefore the MAC throughput is similar to the traffic load. The UDP goodput matches the traffic load (Fig. 4.9) . When R is increased to $1.3Mbps$, MAC throughput and UDP goodput could not match the traffic load as the wireless nodes start dropping packets due to congestion. There is a spike at $51s$ for UDP goodput (Fig. 4.9(b)) for CBR flow from Node 2 to Node 0. This is because of the total traffic load in the network is reduced to $3R/2$ at $50s$ and leads to less MAC collisions. Consequently, backlogs in IFQ are drained out faster and resulting in higher UDP throughput.



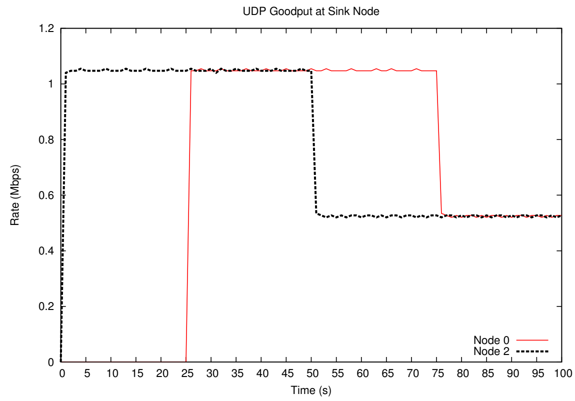
(a) $R = 1.1Mbps$

(b) $R = 1.3Mbps$

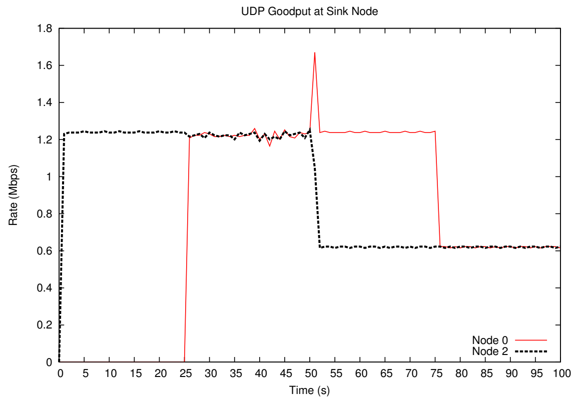
(c) $R = 1.5Mbps$

(d) $R = 1.8Mbps$

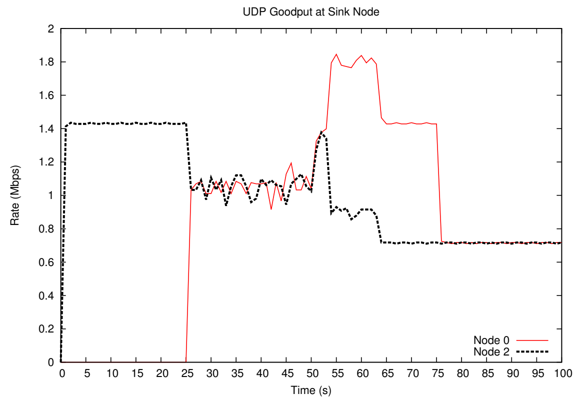
Figure 4.8: Scenario 1: MAC Throughput for Different Traffic Loads



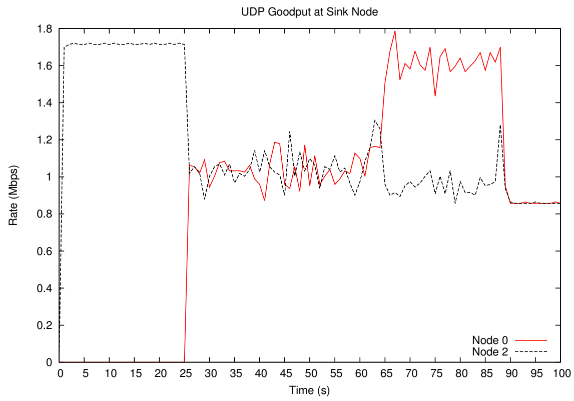
(a) $R = 1.1Mbps$



(b) $R = 1.3Mbps$



(c) $R = 1.5Mbps$

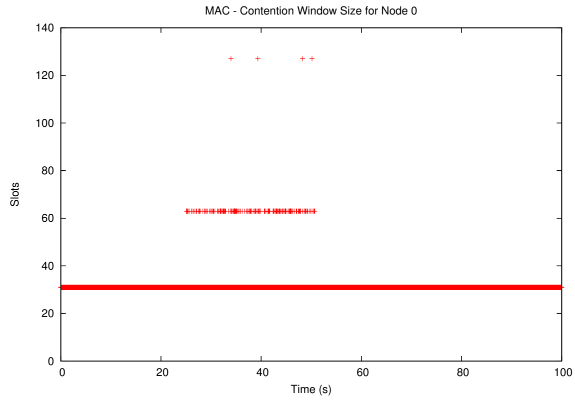


(d) $R = 1.8Mbps$

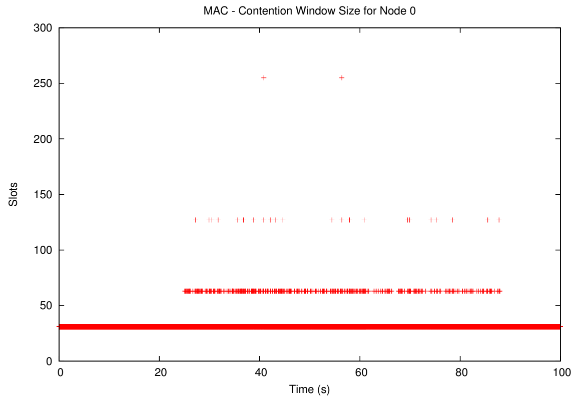
Figure 4.9: Scenario 1: UDP Goodput for Different Traffic Loads

The network becomes congested when R is $1.5Mbps$ and becomes saturated when R is $1.8Mbps$. In both cases, MAC throughput shoots up after the network is less congested as the backlogs are drained faster from the IFQ after $50s$ (traffic load is stepped down to $3/2\ R$). The throughput becomes better when the traffic load is stepped down as the frequency of nodes backoff is reduced. The UDP goodput is much lower than its data rate for these two cases (Figs. 4.9(c) and (d)) due to higher level of packet collisions when traffic load is doubled at $25s$. Although the CBR rate for both directions is $1.5Mbps$ or $1.8Mbps$ each at $25s$ to $50s$, the UDP goodput achieved is only around $0.9Mbps$ - $1.2Mbps$. The network performance is deteriorated seriously when the network is congested.

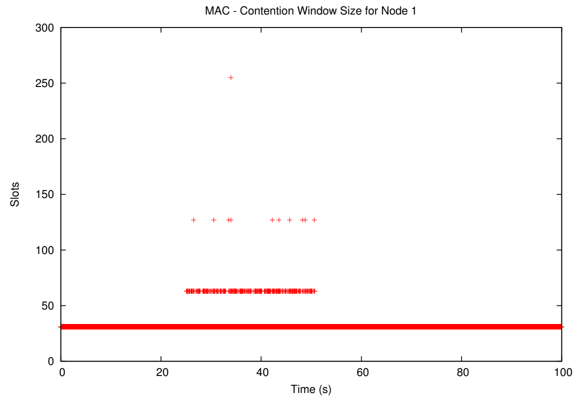
Random backoff frequency of the nodes and the severity of congestion can be inferred from the MAC contention window (CW) size shown in Fig. 4.10. The minimum CW value is 31. Whenever a collision occurs, the nodes that collide backoff for a few time slots in the range of [1.. CW]. The CW value is increased exponentially for each time of collision. If the CW values for all nodes always stay at 31 then there is no collision in the network. Obviously, the network is congested between $25s$ to $50s$ as the MAC CW size increases exponentially. The higher the frequency of backoff and CW value, the larger the MAC layer delay. When collision is minimal, packets get sent out faster so the system throughput is higher. Consequently, packets will not stuck in the queue for long so the queuing delay is lower.



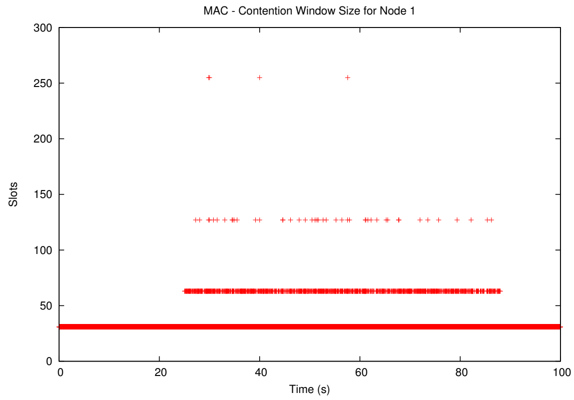
(a) Node 0 ($R = 1.3Mbps$)



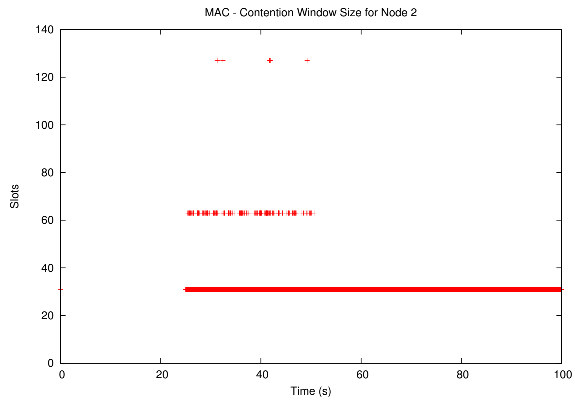
(b) Node 0 ($R = 1.8Mbps$)



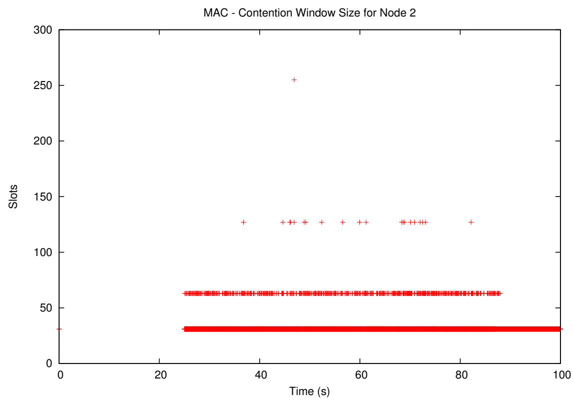
(c) Node 1 ($R = 1.3Mbps$)



(d) Node 1 ($R = 1.8Mbps$)



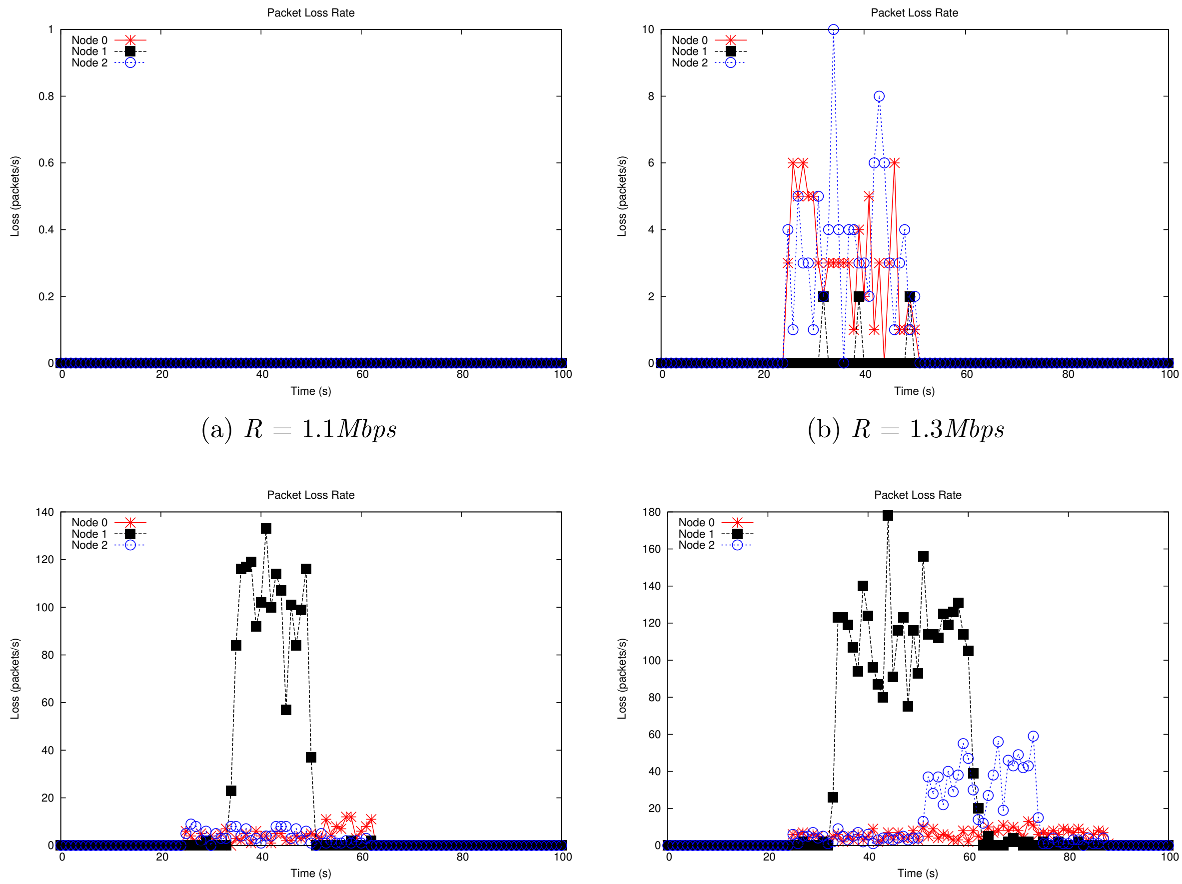
(e) Node 2 ($R = 1.3Mbps$)



(f) Node 2 ($R = 1.8Mbps$)

Figure 4.10: Scenario 1: MAC Contention Window Size for Different Traffic Loads

Fig. 4.11 shows the packet loss rate for all nodes at different traffic loads. There is no packet loss when R is $1.1Mbps$. When the traffic load is increased beyond $1.3Mbps$, packet loss is observed at all nodes. This is due to the IFQ overflows as a result of network congestion. The packet loss is high especially for Node 1 when network is congested (Figs. 4.11(c) - (d)). Node 1 observes higher loss as the traffic intensity is higher at Node 1.

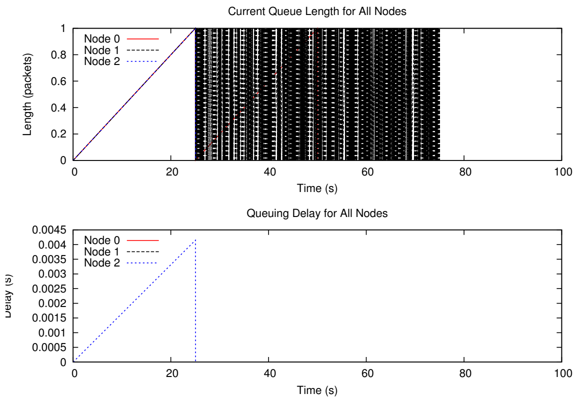


(a) $R = 1.1Mbps$ (b) $R = 1.3Mbps$

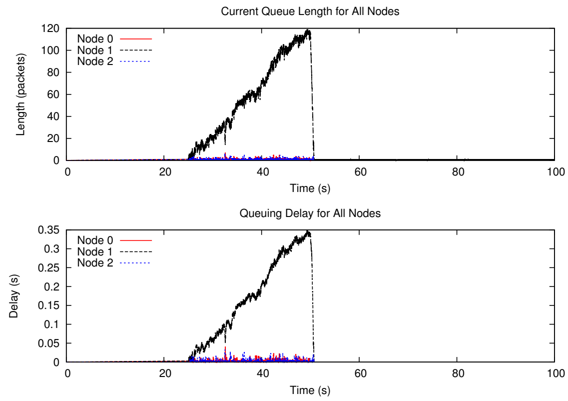
(c) $R = 1.5Mbps$ (d) $R = 1.8Mbps$

Figure 4.11: Scenario 1: Packets Loss Rate for Different Traffic Loads

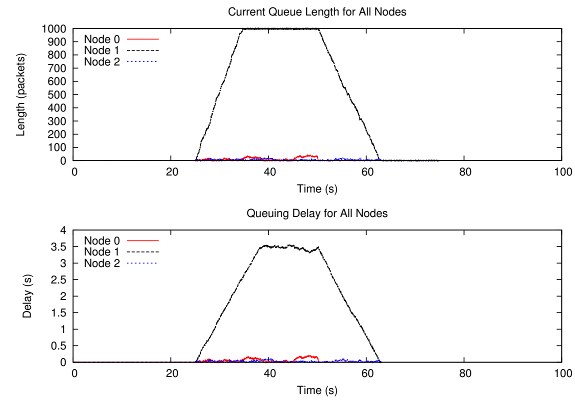
Fig. 4.12 shows the instantaneous queue length and queuing delay measured during the simulation for all nodes. The figure has validated that the high packet loss is due to queue overflow. Basically, there is no backlog when no congestion. When congestion starts, all wireless nodes start to buffer packets in their IFQ while waiting for the chance to grasp channel access. Node 1 experiences higher congestion due to higher traffic load at Node 1. Therefore, backlogs start to build up at Node 1 when $R = 1.3Mbps$ and then hit full queue when $R \geqslant 1.5Mbps$. Higher backlogs are also seen at Node 0 and Node 2 when network is saturated.



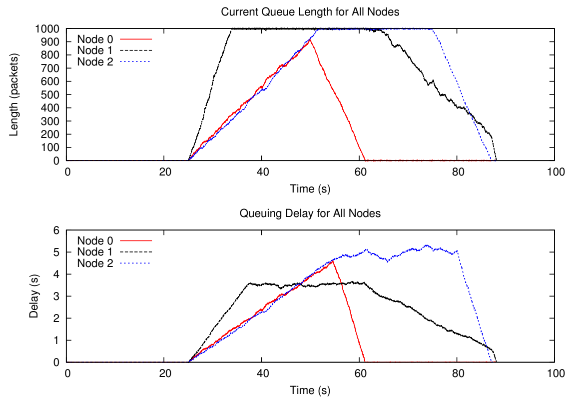
(a) $R = 1.1Mbps$



(b) $R = 1.3Mbps$



(c) $R = 1.5Mbps$

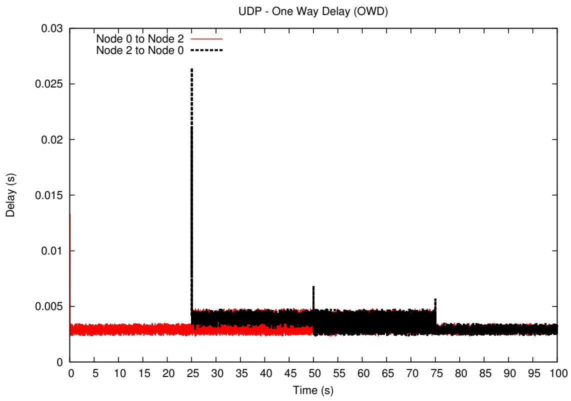


(d) $R = 1.8Mbps$

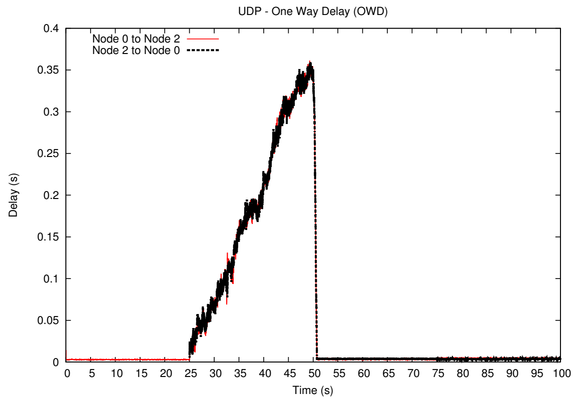
Figure 4.12: Scenario 1: IFQ Instantaneous Queue Length versus Queuing Delay for Different Traffic Loads

When the queue backlog is larger, queuing delay observed is larger. The throughput of nodes is not the same although the nodes are identical. Node 1 has higher opportunities in winning the channel access so it has higher throughput. Therefore, Node 1 has the lowest queuing delay when compared to other nodes at the same instantaneous queue length (Fig. 4.12(d)). The queuing delay observed at Node 1 is $< 4s$ when the IFQ is full. Whereas the queuing delay observed at Node 2 is $> 5s$ when the IFQ is full.

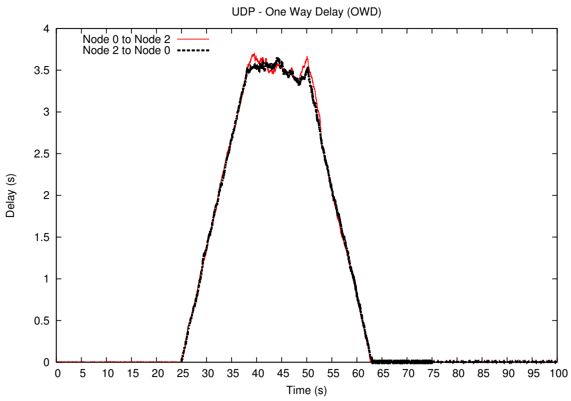
Fig. 4.13 shows the OWD of both CBR flows over time. The OWD (Fig. 4.13(a)) is in the range of 2 miliseconds (ms) - $5ms$ throughout the simulation except a spike at $25s$ with OWD value $\geqslant 25ms$ when the traffic load increases due to traffic load from the opposite direction. The OWD increases drastically when backlog built up quickly at the IFQ due to network congestion during $25s$ - $50s$. The OWD decreases when the rate of CBR flow 1 and CBR flow 2 are reduced at $50s$ and $75s$ respectively. The OWD becomes insignificant after the backlogs in IFQ have been drained and also less collisions at MAC layer. Therefore, queuing delay is the key delay component that causes large OWD and then followed by MAC layer delay.



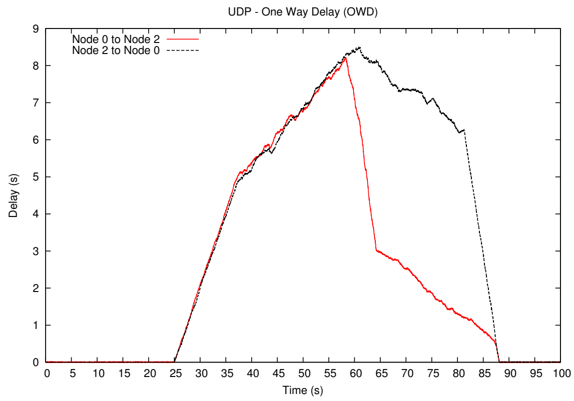
(a) $R = 1.1Mbps$



(b) $R = 1.3Mbps$



(c) $R = 1.5Mbps$



(d) $R = 1.8Mbps$

Figure 4.13: Scenario 1: UDP One Way Delay for Different Traffic Loads

4.4.2 Scenario 2: Different queue size

The simulation configuration for this scenario is similar to Scenario 1 except the queue size is varied now. In this scenario, $R = 1.5Mbps$ (moderate congestion) and $R = 1.8Mbps$ (heavy congestion) are selected for detailed discussion.

The impact of queue size towards collisions is not obvious (Tables 4.3 - 4.4). Collisions increase when the queue size is larger. The wireless nodes with a larger queue size always have packets to transmit due to more backlogs in the queue during network congestion. This increases the chance of collision indirectly. The impact of traffic load towards collision is greater than the queue size, collision count is higher when the traffic load is higher (Figs. 4.14(a) - (b)).

Tables 4.3 - 4.4 also show the trade-off between OWD and packet loss for different queue sizes. The packet loss is higher when the queue size is smaller. The packet loss resulting from queue full can be reduced by increasing the queue size. However, the OWD increases too when the queue size is increased. When $R = 1.5Mbps$, the packet loss is 1587 packets and the maximum OWD is $> 3.5s$ if the

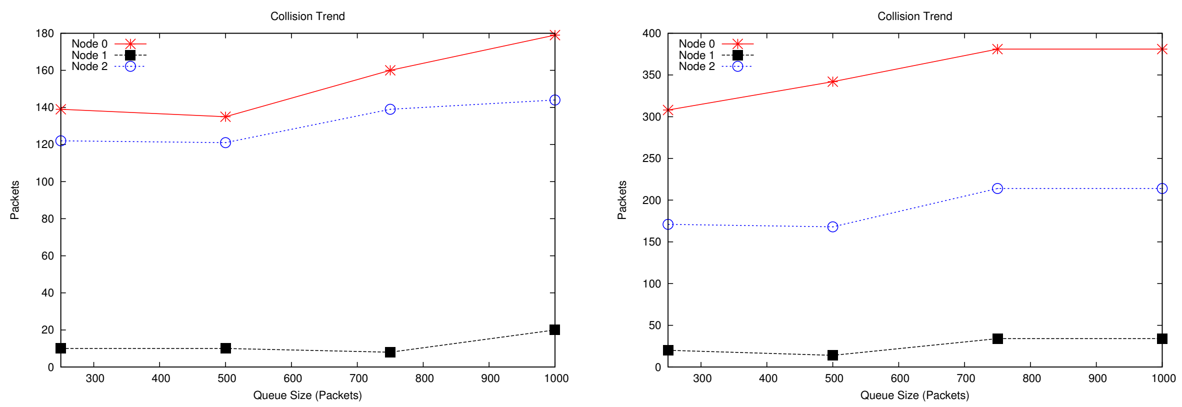
Table 4.3: Scenario 2: Overall System Performance with Different Queue Sizes (R = $1.5Mbps$)

Queue Size	Collision Count	Full Q Drop	Max OWD (s)		Packets Delivery (%)	
			N0→N2	N2→N0	N0→N2	N2→N0
250	271	2397	1.1159	1.0692	91.79	90.21
500	266	2056	1.8856	1.9170	92.92	91.65
750	307	1831	2.8401	2.7193	93.61	92.67
1000	343	1587	3.7020	3.6544	94.62	93.46

Table 4.4: Scenario 2: Overall System Performance with Different Queue Sizes (R = $1.8Mbps$)

Queue Size	Collision Count	Full Q Drop	Max OWD (s)		Packets Delivery (%)	
			N0→N2	N2→N0	N0→N2	N2→N0
250	499	4983	2.2907	2.2660	87.5	80.9
500	524	4789	4.5148	4.3673	88.6	81.0
750	547	4297	6.4521	6.5232	90.2	82.4
1000	629	3936	8.2236	8.5012	91.2	83.7

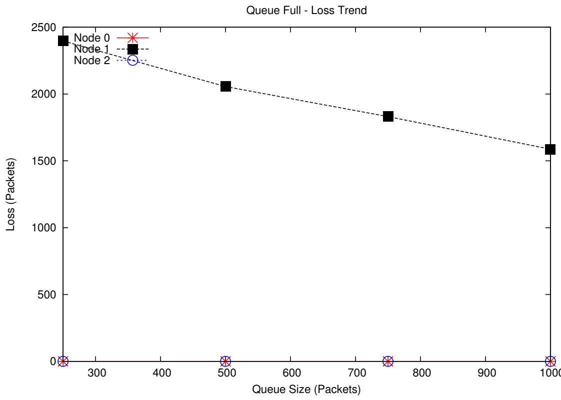
queue size is 1000 packets. If the queue size is 250 packets, the maximum OWD is reduced to $< 1.2s$ but the packet loss is nearly 2400 packets. A similar trend is observed for $R = 1.8Mbps$, but both packet loss and queuing delay increase at a larger magnitude.



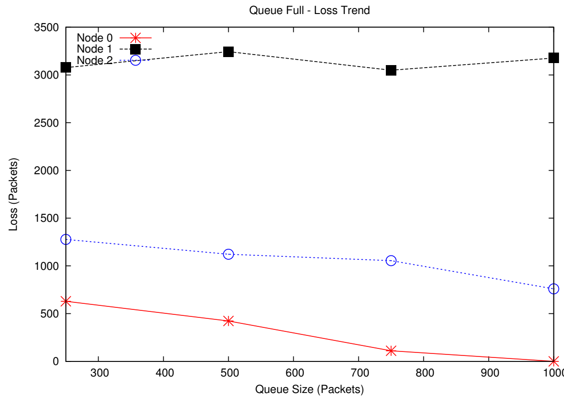
(a) $R = 1.5Mbps$ (b) $R = 1.8Mbps$

Figure 4.14: Scenario 2: Collision Trend for Different Queue Sizes

Fig. 4.15 shows that only Node 1 drops packets due to queue full when $R = 1.5Mbps$. While for $R = 1.8Mbps$, all nodes drop packets due to queue full when network is highly congested. The number of packet losses decreases when the queue size is larger since that more packets can be buffered in the queue when the network is congested.

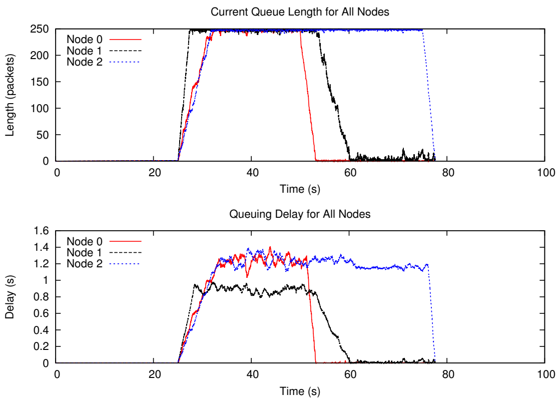


(a) $R = 1.5Mbps$

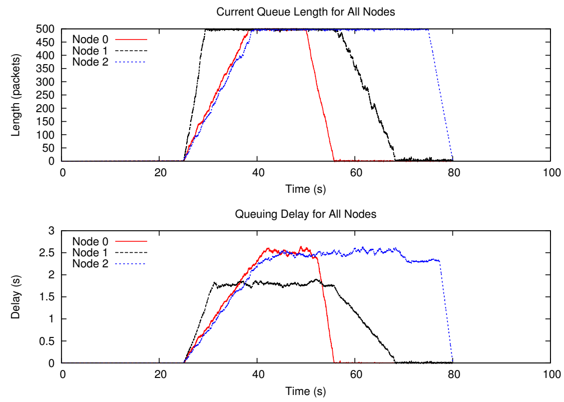


(b) $R = 1.8Mbps$

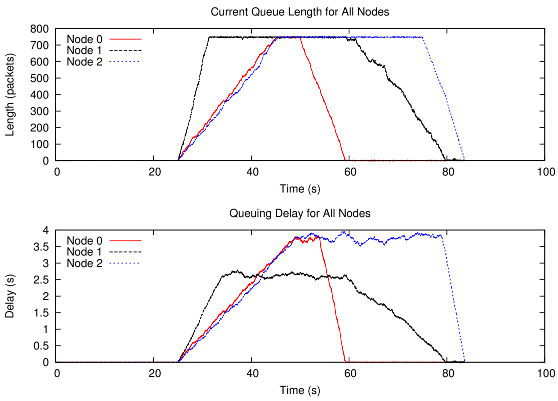
Figure 4.15: Scenario 2: Queue Full Loss Trend for Different Queue Sizes



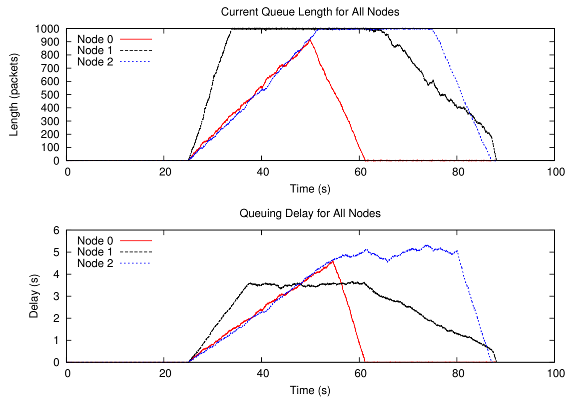
(a) Queue Size = 250



(b) Queue Size = 500



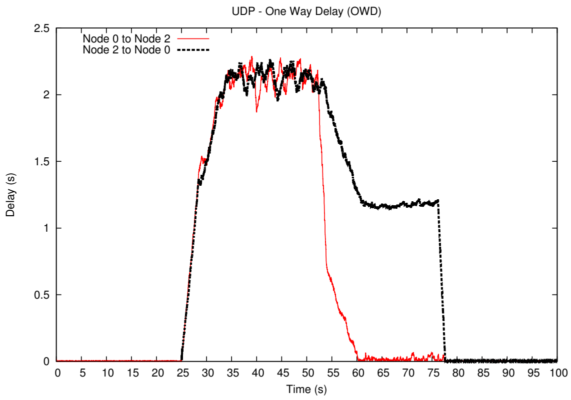
(c) Queue Size = 750



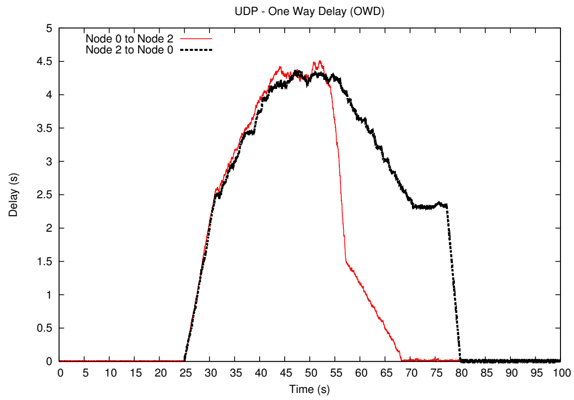
(d) Queue Size = 1000

Figure 4.16: Scenario 2: IFQ Instantaneous Queue Length versus Queuing Delay for Different Queue Sizes ($R = 1.8Mbps$)

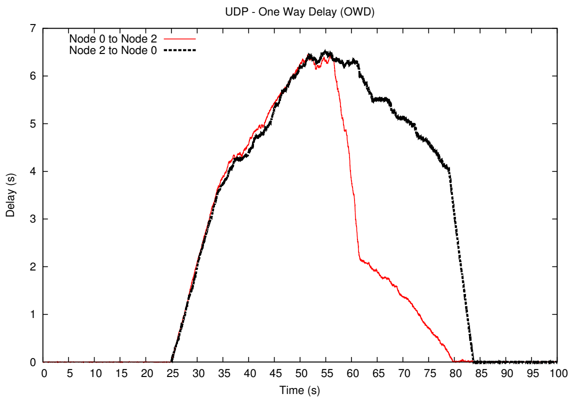
By looking at the instantaneous queue length versus queuing delay over time, the queue size impact on the performance is obvious. The simulation scenario for $R = 1.8Mbps$ is discussed here. When the network is heavily congested, the IFQ is always maintained in full or nearly full state (Fig. 4.16). The more backlogs in a queue the more time is needed to drain the queue, therefore a new packet entering the queue waits for a longer time to get transmitted over the wireless medium. As a result, this leads to longer queuing delay which contributes to longer OWD as shown in Fig. 4.17. Although the current queue lengths of all queues from time $25s$ to $50s$ are almost the same; the queuing delay observed at each queue is different as the MAC throughput of each node is different. Other than that, the time taken to drain a full queue is longer for a bigger queue size. Therefore, the OWD is maintained at high values for a longer period for a larger queue size.



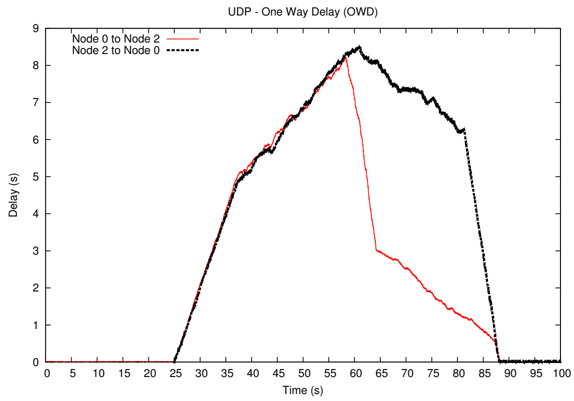
(a) Queue Size = 250



(b) Queue Size = 500



(c) Queue Size = 750

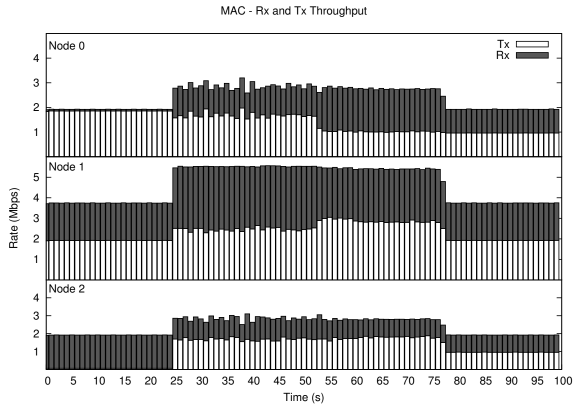


(d) Queue Size = 1000

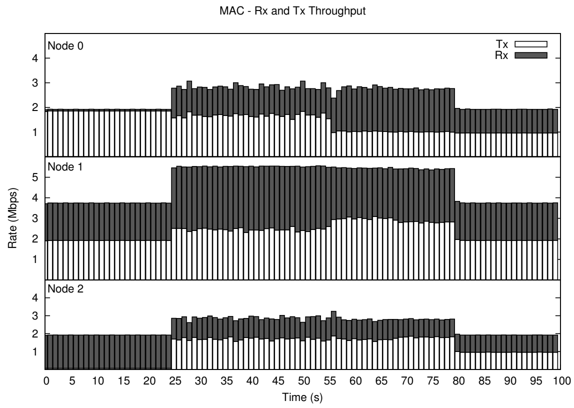
Figure 4.17: Scenario 2: UDP One Way Delay for Different Queue Sizes ($R = 1.8Mbps$)

Reducing the queue size by a stepping of 250 packets, the magnitude of OWD is reduced by approximately $2s$ for each stepping. The maximum queuing delay is bounded by the queue size. The queues are saturated for all nodes when the congestion is peak (Fig. 4.16). The performance gain is far greater than the packet loss in terms of magnitude. For example, when the queue size is reduced by half from 1000, the gain in OWD is $>$ 1.8x (Fig. 4.17(b) versus Fig. 4.17(d)); while the packet loss is only increased by 1.2x and the packet delivery ratio is only reduced by around 2.5% (Table 4.4).

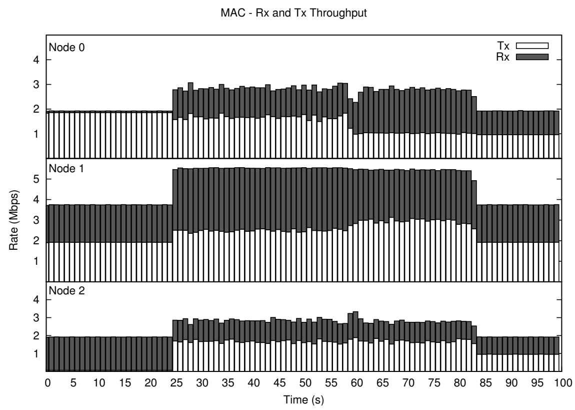
Queue size does not give obvious impact to the MAC throughput (Fig. 4.18), the MAC throughput trend is similar for all queue sizes except MAC throughput is maintained at higher throughput after $75s$ for larger queue size. This is the consequence of draining full queue from IFQ with larger backlogs. The trends of UDP goodput are similar for different queue sizes. However, it takes a longer time for UDP throughput to resume normal for a larger queue size even though the traffic load has been stepped down at $50s$ (Fig. 4.19). It takes a longer time to clear backlogs from a larger queue, therefore this cause longer congestion in the network until the queues are drained.



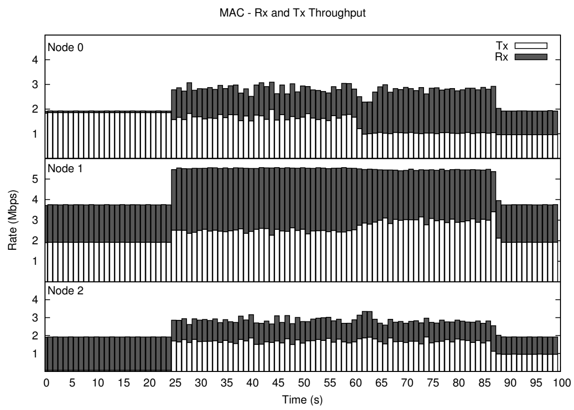
(a) Queue Size = 250



(b) Queue Size = 500

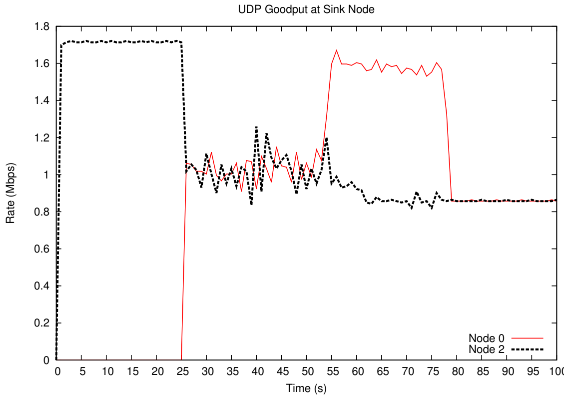


(c) Queue Size = 750

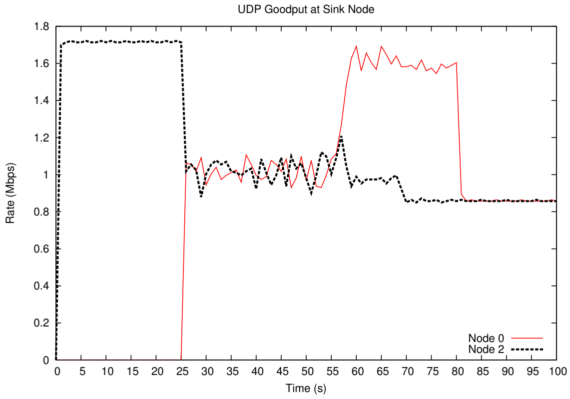


(d) Queue Size = 1000

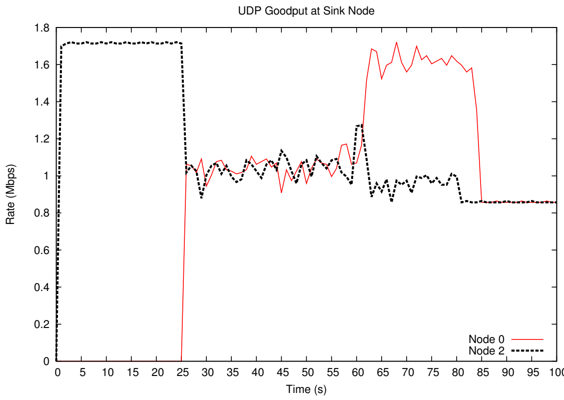
Figure 4.18: Scenario 2: MAC Throughput for Different Queue Sizes ($R = 1.8Mbps$)



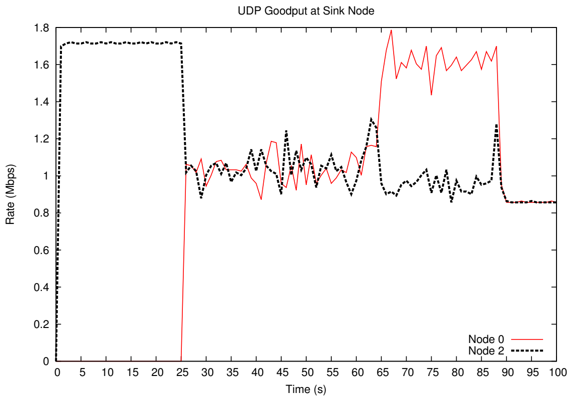
(a) Queue Size = 250



(b) Queue Size = 500



(c) Queue Size = 750



(d) Queue Size = 1000

Figure 4.19: Scenario 2: UDP Goodput for Different Queue Sizes ($R = 1.8Mbps$)

4.4.3 Scenario 3: Different packet size

Data packets sent over networks may vary in size, the packet size depends on the device settings (such as encoder), protocol stack and application needs. Therefore, the impact of packet size on network performance has been analyzed in this simulation scenario. The settings are similar to Scenario 1 except the packet size is varied from $200B$ to $1200B$ with stepping of $200B$. Only $R = 1.1Mbps$ and $R = 1.8Mbps$ are picked for the simulation in this scenario.

The simulation results are summarized in Tables 4.5 and 4.6. Based on the performance analysis in Scenario 1 with packet size $1000B$ under $R = 1.1Mbps$, the network is lightly loaded and not congested. However, the network experiences congestion at $R = 1.1Mbps$ when the packet size is smaller.

Table 4.5: Scenario 3: Overall System Performance with Different Packet Sizes (R = $1.1 Mbps$)

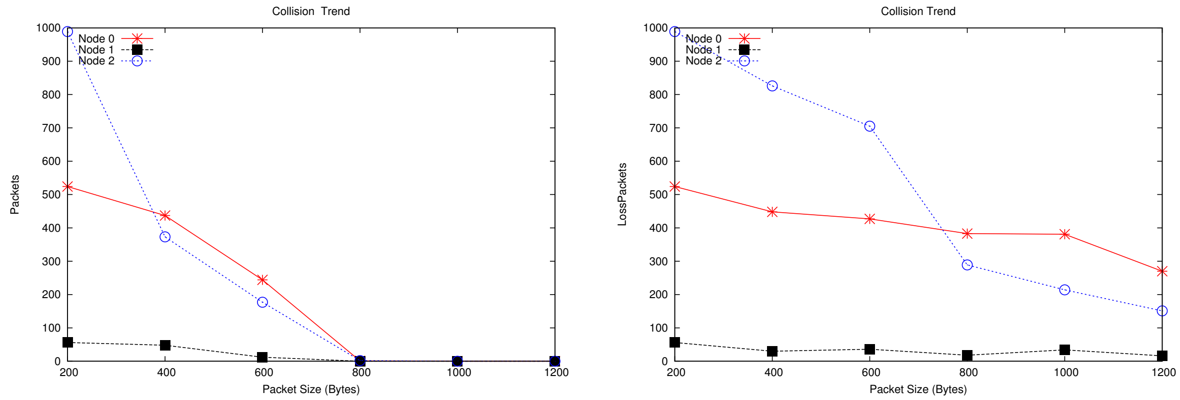
Packet Size	Collision Count	Full Q Drop	Max OWD (s)		Packets Delivery (%)	
			N0→N2	N2→N0	N0→N2	N2→N0
200	1569	67918	5.2964	5.3036	45.79	33.02
400	858	10266	6.2744	6.1827	86.29	73.44
600	433	2339	2.9296	2.9122	93.62	92.40
800	3	0	0.0165	0.0227	100	100
1000	0	0	0.0133	0.0264	100	100
1200	0	0	0.0121	0.0224	100	100

Table 4.6: Scenario 3: Overall System Performance with Different Packet Sizes (R = $1.8 Mbps$)

Packet Size	Collision Count	Full Q Drop	Max OWD (s)		Packets Delivery (%)	
			N0→N2	N2→N0	N0→N2	N2→N0
200	1569	143107	5.2968	5.3050	28.0	20.2
400	1304	42109	6.2845	6.1554	54.7	38.9
600	1168	17307	7.0345	7.1341	75.7	57.0
800	690	7819	8.0442	7.9665	86.7	73.7
1000	629	3936	8.2236	8.5012	91.2	83.7
1200	437	2159	5.7883	5.6104	92.9	90.7

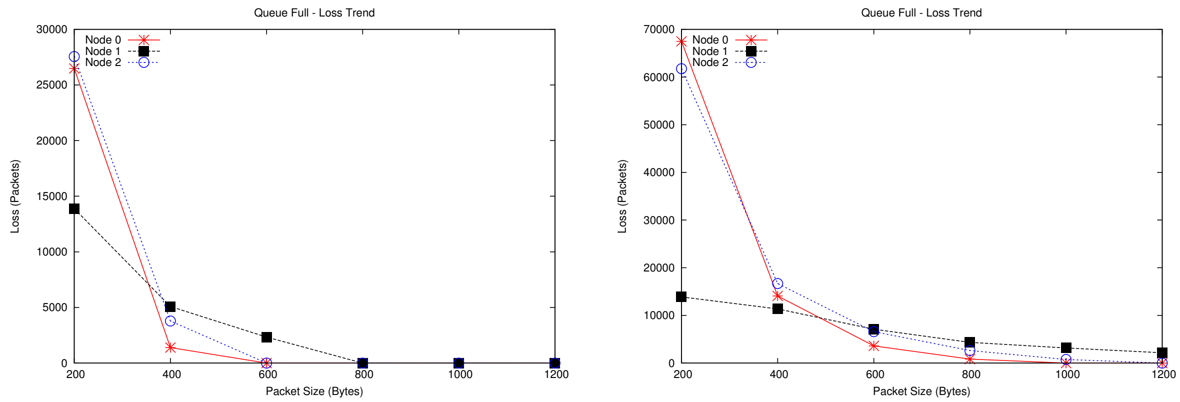
When the packet size is small, collision is high (Fig. 4.20). This is due to more packets need to be sent under the same data rate. The nodes need to compete for channel access and randomly backoff more frequent when compared to the scenario that has the same traffic load but with a larger packet size. For $R = 1.1 Mbps$, the network is heavily congested when the packet size is $200B$. The congestion condition is then eased when the packet size becomes bigger. Therefore, packet size is an important factor that contributes to network dynamics. For the case of $R = 1.8 Mbps$, the trend of collision count is also decreasing when the packet size is bigger. However, the decreasing rate is not as fast as $R = 1.1 Mbps$ case. This is because the data rate is higher and the congestion is partly contributed by the fast data rate.

Packet loss due to queue full also increases drastically when the packet size becomes smaller. This is because more packets are generated at the same data rate. This causes the queue to fill up quickly. Fig. 4.21 shows the packet loss count decreases with an inverse exponential trend when the packet size gets larger. The packet delivery ratio also increases drastically for a larger packet size due to fewer packet drops.



(a) $R = 1.1Mbps$ (b) $R = 1.8Mbps$

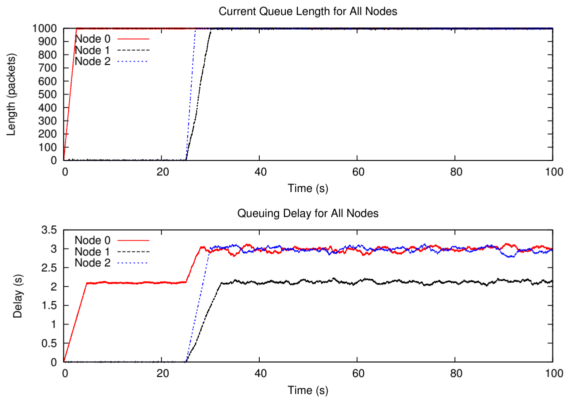
Figure 4.20: Scenario 3: Collision Trend with Different Packet Sizes



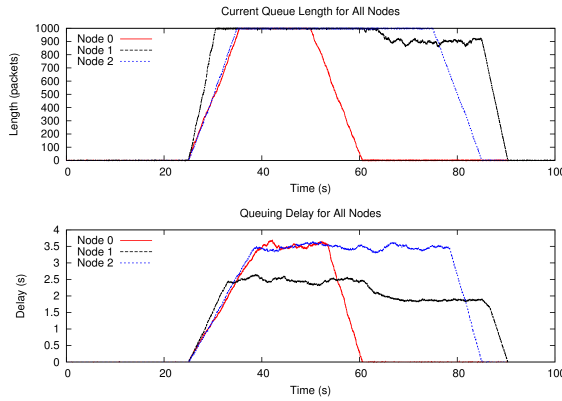
(a) $R = 1.1Mbps$ (b) $R = 1.8Mbps$

Figure 4.21: Scenario 3: Queue Full Loss Trend with Different Packet Sizes

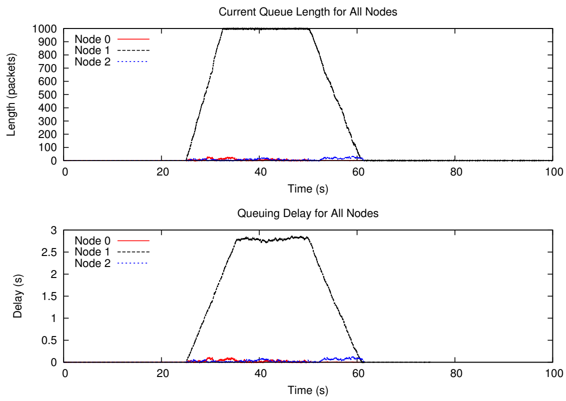
As discussed earlier, less collision means less congestion in the network. Time is not wasted to compete for channel access. This also means that throughput is higher and then incoming packets can be transmitted faster. This leads to fewer backlogs in a queue. Therefore, the OWD is lower (Table 4.5). However, the OWD is increasing even the collision count is decreasing when $R = 1.8Mbps$ (Table 4.6). The contradiction is due to the network is over saturated at this rate. Although the packet size is as large as $1000B$, the network is still heavily congested. Most of the time the IFQ is nearly full or full. It takes a longer time to clear a full queue with bigger sized packets. Nevertheless, the network is not necessarily more congested when the OWD is larger. In fact, the network is less congested as the collision count and packet loss are reducing when the packet size is larger. The turning point of OWD for $R = 1.8Mbps$ when the packet size is $1200B$ is due to the network congestion level is much lower compared to when the packet size is $1000B$. Consequently, packets are drained faster and not all IFQ are always maintained in full queue. Therefore, the OWD becomes lower.



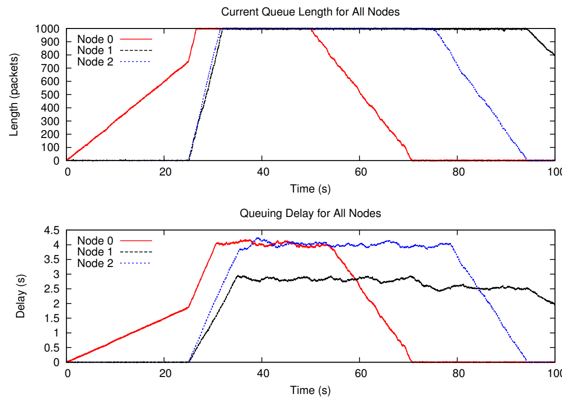
(a) Packet Size = $200B$ ($R = 1.1Mbps$)



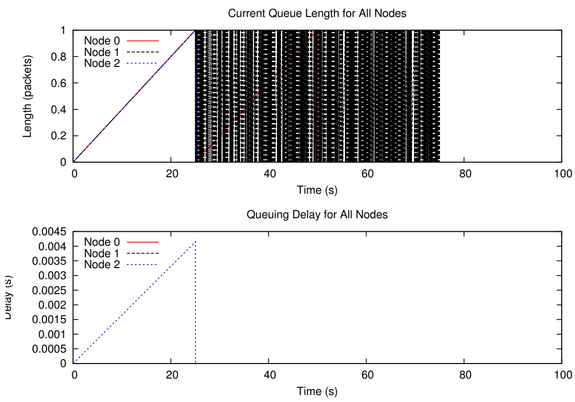
(b) Packet Size = $200B$ ($R = 1.8Mbps$)



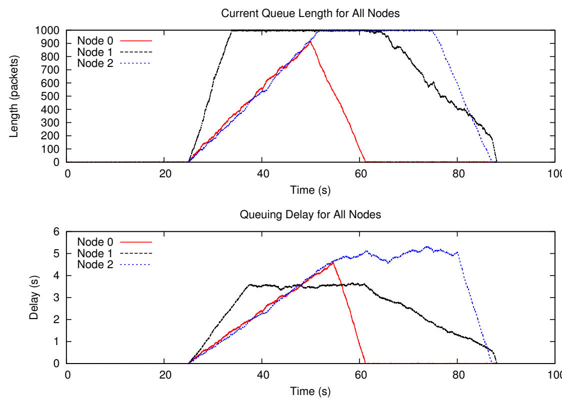
(c) Packet Size = $600B$ ($R = 1.1Mbps$)



(d) Packet Size = $600B$ ($R = 1.8Mbps$)



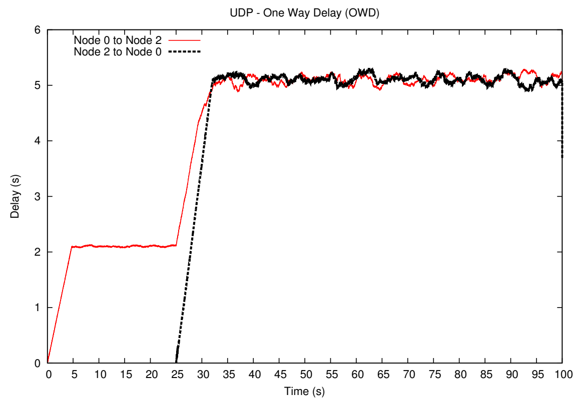
(e) Packet Size = $1000B$ ($R = 1.1Mbps$)



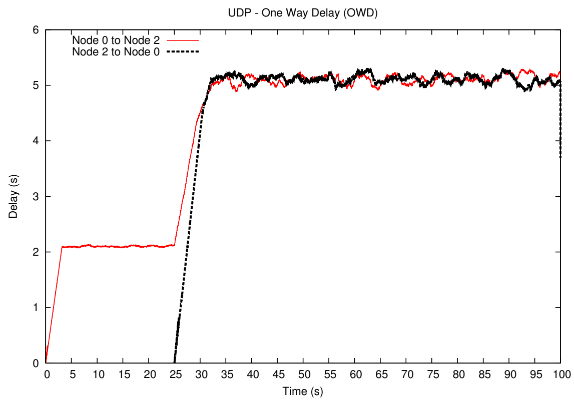
(f) Scenario 3: Packet Size = $1000B$ ($R = 1.8Mbps$)

Figure 4.22: Scenario 3: IFQ Instantaneous Queue Length versus Queuing Delay for Different Packet Sizes

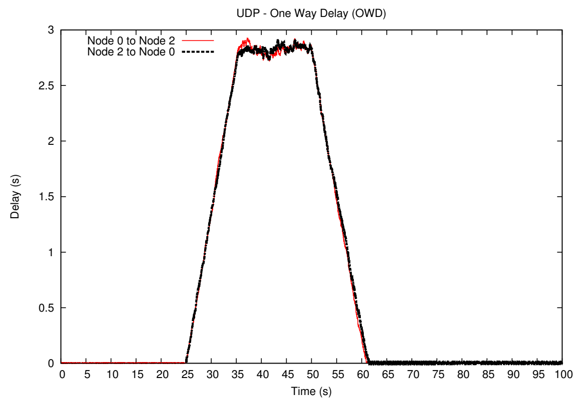
The magnitude of queuing delay is proportional to packet size and queue backlogs. Under full queue condition, queue with smaller packet size has smaller queuing delay. Fig. 4.22 shows that the same instantaneous queue length may be translated into different magnitude of queuing delay due to network dynamics caused by the packet size and traffic load. Therefore, OWD is higher when the packet size increases during network congestion.



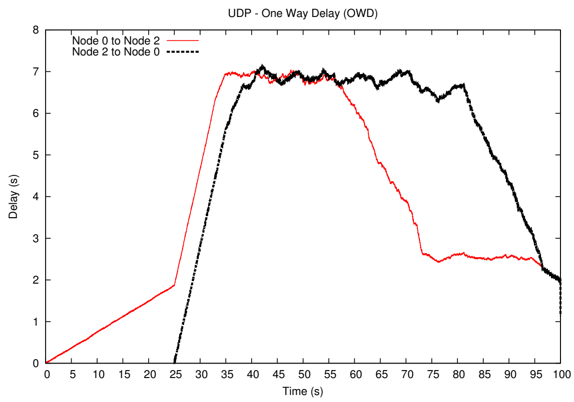
(a) Packet Size $= 200B$ ($R = 1.1Mbps$)



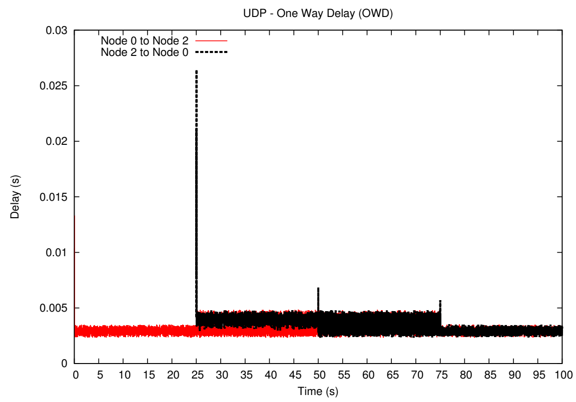
(b) Packet Size $= 200B$ ($R = 1.8Mbps$)



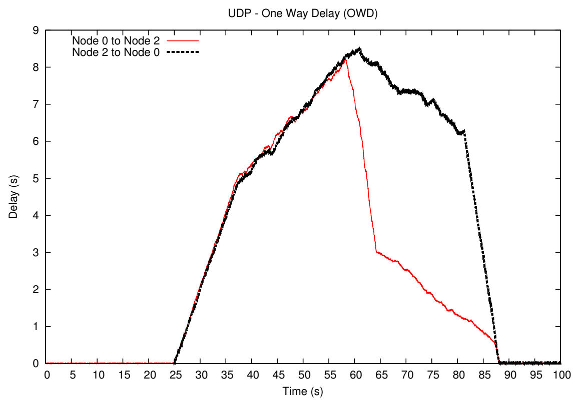
(c) Packet Size $= 600B$ ($R = 1.1Mbps$)



(d) Packet Size $= 600B$ ($R = 1.8Mbps$)



(e) Packet Size $= 1000B$ ($R = 1.1Mbps$)

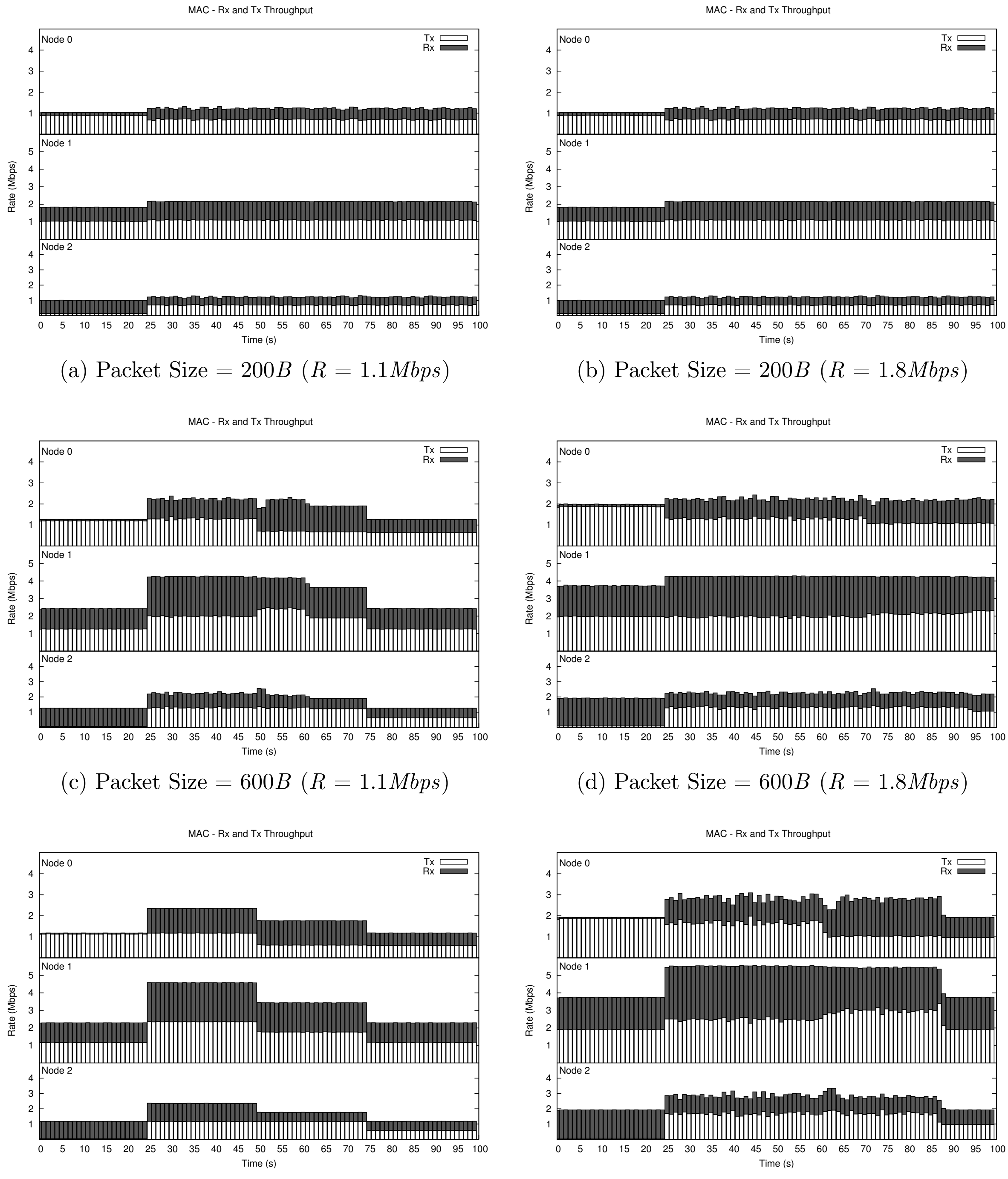


(f) Packet Size $= 1000B$ ($R = 1.8Mbps$)

Figure 4.23: Scenario 3: UDP One Way Delay for Different Packet Sizes

The OWD trends follow the trends of queuing delay due to queuing delay is the major contributor to OWD in this simulation (Fig. 4.23).

Fig. 4.24 shows that MAC throughput is lower when the network is more congested resulting from smaller packet size under the same traffic load. Time is wasted in competing for channel access and retransmission due to collision.



(a) Packet Size = $200B$ ($R = 1.1Mbps$)

(b) Packet Size = $200B$ ($R = 1.8Mbps$)

(c) Packet Size = $600B$ ($R = 1.1Mbps$)

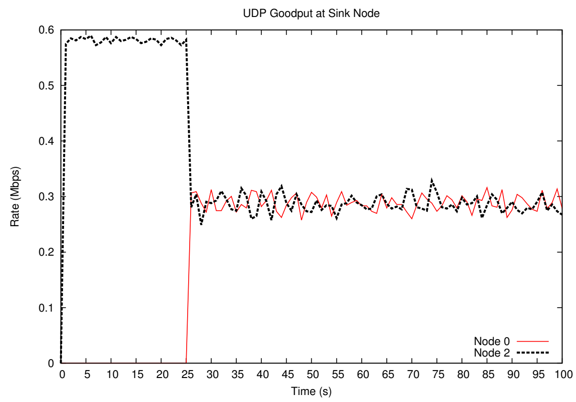
(d) Packet Size = $600B$ ($R = 1.8Mbps$)

(e) Packet Size = $1000B$ ($R = 1.1Mbps$)

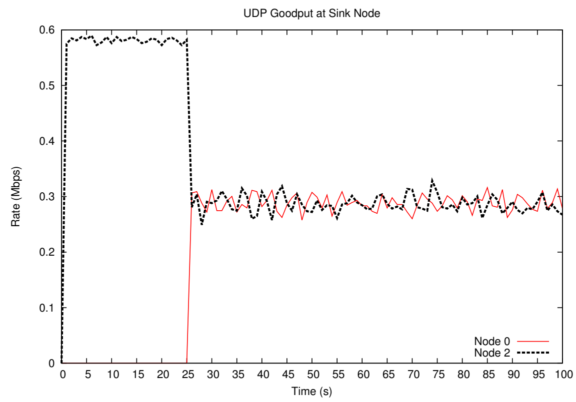
(f) Packet Size = $1000B$ ($R = 1.8Mbps$)

Figure 4.24: Scenario 3: MAC Throughput for Different Packet Sizes

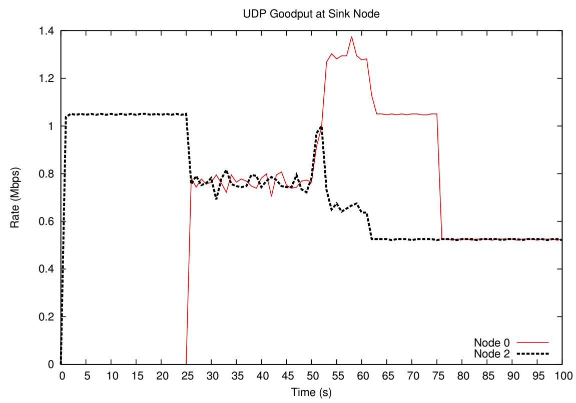
Similar to MAC throughput, UDP goodput is much lower when the network is highly congested (Fig. 4.25). When the traffic intensity is the same but the traffic load consists of smaller packets, the UDP goodput is seriously deteriorated due to higher collision and higher packet drops.



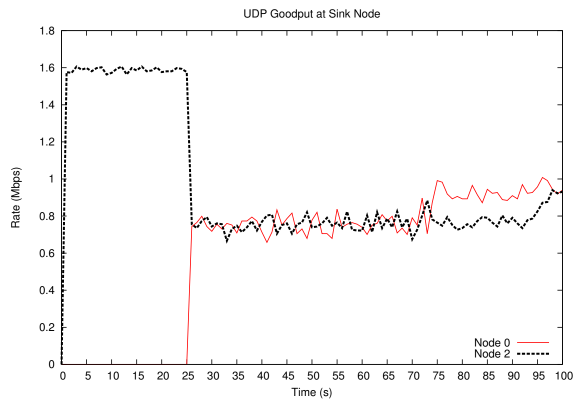
(a) Packet Size $= 200B$ $(R = 1.1Mbps)$



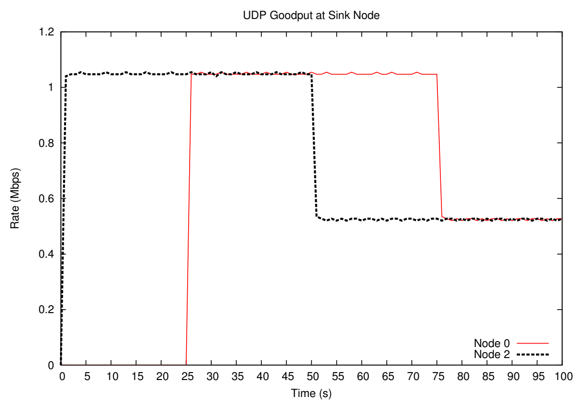
(b) Packet Size $= 200B$ $(R = 1.8Mbps)$



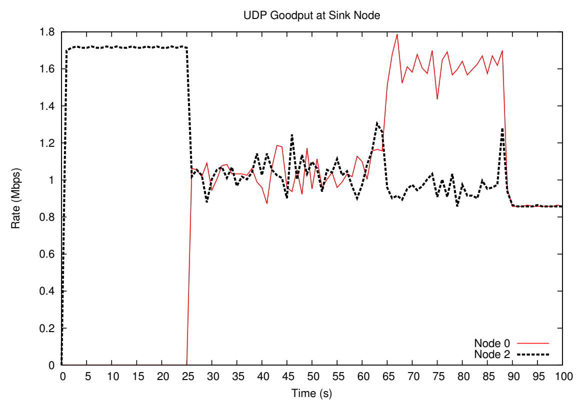
(c) Packet Size $= 600B$ $(R = 1.1Mbps)$



(d) Packet Size $= 600B$ $(R = 1.8Mbps)$



(e) Packet Size $= 1000B$ $(R = 1.1Mbps)$



(f) Packet Size $= 1000B$ $(R = 1.8Mbps)$

Figure 4.25: Scenario 3: UDP Goodput for Different Packet Sizes

4.5 Summary and Discussions

From this delay analysis, obviously the queue size is one of the significant factors that contribute to large OWD in wireless ad hoc networks that carry real-time data. Nodes that are identical with the same level of queue backlogs may experience different queuing delay due to each node has different throughput, different channel access opportunity and different level of interference. Therefore, a proper tuning of IFQ threshold is a crucial factor to control end-to-end delay.

Network dynamics and interference are also the causes of OWD variation. There are many factors that contribute to network dynamics and interference, for example number of nodes sharing the same transmission medium, distance between nodes, network load variation due to different application needs, packet size variation, etc. It can be seen that network dynamics bring huge impacts to the network performance. Therefore, all these factors need to be considered in order to improve the network performance.

However, most of these factors are difficult to be controlled. The traffic load in a network depends on the active nodes in the network within the same proximity of transmission range, while the packet size and the data rate are usually application specific. Besides that, environmental interference and interference from neighbour nodes are hard to be controlled. Some may suggest to use QoS provisioning scheme such as scheduling, bandwidth reservation in order to control the network load and interference from neighbour nodes. However, all these schemes may need a greater picture of the network and may be complex or high overhead to be implemented. It is also hard to control the MAC throughput and the random backoff at the MAC layer.

The simulation results in this chapter show that queuing delay is the major component of the OWD. Therefore, queuing delay can be manipulated to offset other delay components imposed by various factors. This means that the maximum OWD for delay-sensitive traffic flows can be regulated by controlling and maintaining a dynamic queue threshold. The key findings from this delay analysis are considered in the ADTH queue management scheme that is proposed in the next chapter.

Chapter 5

Adaptive Dynamic Threshold Queue Management

5.1 Introduction

This chapter presents an adaptive queue management scheme to bound nodal delay of wireless nodes to a required level. The proposed scheme is named after DTH, and is known as ADTH (Adaptive DTH). It is a generic and scalable adaptive queue management which can self-adapt to network dynamics without the need of reconfiguration after being deployed.

The delay analysis in Chapter 4 shows that constraining the queue length or queuing delay to a constant required value is not sufficient to constrain network delay for delay-sensitive traffic in a wireless ad hoc network. MAC layer contention and interference may cause QoS deterioration that lead to large variation in queuing delay and MAC layer delay. To guarantee a deterministic per-hop delay, both MAC layer delay and queuing delay should be taken care of. End-to-end delay for delay-sensitive traffic could be bounded by bounding per-hop delay of intermediate nodes along the path towards a destination. Thus, an adaptive queue management scheme which can constrain nodal delay of wireless nodes under dynamic conditions is needed.

The analytical approach in DTH has shown that the average queuing delay can be bounded. However, the analytical approach in DTH is not suitable for wireless ad hoc networks where the interference and link quality vary over time. DTH could not respond to interference and network dynamics autonomously as it relies on the results of a priori queuing analysis for the queuing threshold estimation. Furthermore, MAC layer delay should be considered in wireless ad hoc networks. Therefore, DTH has evolved to an adaptive online approach as proposed in this chapter.

Most of the queue management schemes proposed [34, 62, 97, 100, 120, 125, 126, 136, 179, 189] for wireless ad hoc networks focus on congestion problem for delay-tolerant traffic but not delay-sensitive traffic. These schemes aim to alleviate congestion by dropping packet probabilistically so that traffic sources can respond to packet loss events by adapting their sending rate. However, delay-sensitive traffic that is typically carried over UDP cannot respond to packet loss events. Therefore, queuing delay and nodal delay for delay-sensitive traffic cannot be bounded with these schemes.

The ADTH queue management scheme proposed in this chapter has filled in the gap to constrain the network delay for delay-sensitive traffic in wireless ad hoc networks. The aim of ADTH is to adapt the queuing threshold on wireless nodes autonomously so that a deterministic per-hop delay can be achieved. The implications on the network delay, UDP goodput, loss and data yield of ADTH have been assessed with a NS-2 simulation.

The remainder of the chapter is organized as follows: Section 5.2 gives an overview of ADTH; Section 5.3 describes the system design of ADTH in details; Section 5.4 discusses on assumptions and limitations of ADTH; Section 5.5 presents the results analysis from the NS-2 simulation; finally a summary is given in Section 5.6.

5.2 ADTH (Adaptive Dynamic THreshold) Overview

ADTH is an adaptive queue management that aims to control the maximum nodal delay of a wireless node to a required nodal delay (D_{Nr}) by adjusting the target queuing threshold (L_Q) dynamically (Fig. 5.1). The feedback control loop results in a movable queue threshold for the system queue and maintains nodal delay around the D_{Nr} specified when a network is congested.

ADTH measures system performance metrics of interest periodically and then uses the system metrics collected to estimate the queuing threshold for next sampling period based on the current situation of the network. Changes of network situation may contribute to variation of nodal delay that comprises of queuing delay and MAC layer delay. For examples, larger queuing delay caused by larger backlogs in a queue and lower system throughput, larger MAC layer delay caused by collision and retransmission. Nodal delay of wireless nodes increases due to aforementioned factors and this leads to larger OWD. Therefore, ADTH aims to monitor the changes of network situation to bound nodal delay by actively regulating the queuing threshold of IFQ.

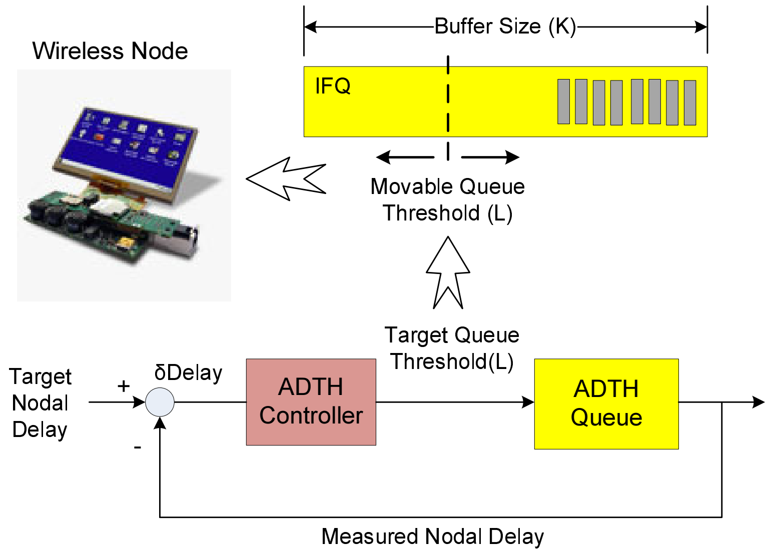


Figure 5.1: ADTH System Diagram

Several important findings from Chapter 4 have been taken into consideration for the core design of ADTH. These important findings and design considerations are summarized as below:

1. Not all wireless nodes get equal opportunities to access the shared medium for transmission. This leads to an uneven distribution of available shared bandwidth to each wireless node. Wireless nodes with higher opportunities to gain medium access will have higher system throughput. Even the wireless nodes may have the same queue length or backlogs in queues but variation in queuing delay is observed due to different system throughput. Hence, queue throughput (system throughput at queue level) becomes a key parameter in the design of the ADTH scheme.

2. Network load level and packet size contribute to network dynamics and network congestion. A higher network load causes network congestion. This leads to a higher packet loss rate and larger OWD. It is difficult to predict traffic load in a wireless ad hoc network due to interference from nodes within the proximity of transmission range. Therefore, traffic load is not considered in the ADTH design. Smaller packet size incurs more overhead in channel access contention. The packet size may impact the magnitude of queuing delay also. Therefore, packet size becomes one of the parameters for the ADTH design.

3. Maximum queue size is a major player in determining the OWD. The magnitude of queuing delay is much larger than the magnitude of propagation

delay, transmission delay and backoff delay. Different queue size leads to different maximum queuing delay under the same system throughput. Besides that, traffic load, system throughput and packet size cause variation in queuing delay. In a highly dynamic network, finding an optimum queue size that can fulfil the delay requirement is very important. This motivates the need of an adaptive queue management approach that can bound nodal delay per system or user requirement based on the current network situation.

4. MAC layer delay is a stochastic delay component as there are various factors which can influence the magnitude of MAC layer delay. MAC layer delay is mainly contributed by random backoff and retransmission in the network when nodes compete for medium access to transmit packets. It is difficult to predict and control MAC layer delay due to uncertainties in network load, interference, etc. Therefore, MAC layer delay should be offset by regulating the maximum queuing delay at node level.

Based on the findings and considerations above, the ADTH controller is designed to bound nodal delay of wireless nodes through active monitoring and measuring of system performance metrics. With the ADTH queue management scheme in place, packets are dropped when the current queue length (QLen) exceeds the target queue threshold estimated. Detailed design of ADTH is elaborated in Section 5.3.

5.3 ADTH Design

The ADTH queue management scheme reacts to network changes by tuning the target queue threshold to bound nodal delay. Active measurements and statistics collection are carried out periodically for the estimation of target queuing threshold and performance monitoring. The sampling time (t_s) for the ADTH controller can be fine-tuned to balance the trade-off required between accuracy of the controller and system overhead.

At each sampling interval (k), the ADTH controller estimates the target queue threshold by using Eqs. 5.1 - 5.10. Maximum allowable queuing delay (D_{Qr}) is estimated by subtracting measured mean MAC delay (D_M) from the target nodal delay (D_{Nr}). MAC delay measurement is further explained in Subsection 5.3.2.

$$D_{Qr}(k+1) = D_{Nr} - \bar{D}_M(k) \tag{5.1}$$

After D_{Qr} is obtained, the maximum allowable queue size (S_{Qr}) for next sampling interval ($k+1$) is estimated based on the queue throughput (T_Q) at the current sampling interval. T_Q is calculated from the number of bytes sent

(n_B) over the current sampling interval. Mean packet size (S_{Pr}) is calculated from the queue statistics collected by averaging the number of bytes sent with the number of packets sent (n_P), so that the estimation of maximum allowable queue threshold (L_{Qr}) can be done in the unit of packets (Eqs. 5.4 and 5.5).

$$T_Q(k) = n_B(k)/t_s \tag{5.2}$$

$$S_{Qr}(k+1) = D_{Qr}(k) * T_Q(k) \tag{5.3}$$

$$S_{Pr}(k+1) = n_B(k)/n_P(k) \tag{5.4}$$

$$L_{Qr}(k+1) = S_{Qr}(k+1)/S_{Pr}(k+1) \tag{5.5}$$

To increase the robustness of the ADTH scheme, a queue length factor (f_Q) is used to adapt L_{Qr} based on a feedback control loop to get a final target queue threshold (L_Q) (Eqs. 5.6 - 5.10). This is to absorb the impact of network changes that causes QoS deterioration (e.g. overshooting of measured nodal delay (D_N) against D_{N_r}). At each sampling interval, D_N is compared to D_{Nr} to obtain error (e_N). The normalized error (β) is used to update f_Q. If D_N overshoots D_{Nr}, f_Q is deducted by β. This results in a smaller f_Q and hence lower the L_Q. When $D_N < D_{Nr}$, f_Q is increased slowly by the magnitude of ${}^{\beta}\!/_{2}$ to anticipate network changes in preventive manner. Through this mechanism, the target queue threshold is not solely based on the performance metrics at the current sampling interval but also from the previous sampling interval.

$$D_N(k) = D_Q(k) + D_M(k) \tag{5.6}$$

$$e_N(k) = D_{Nr} - D_N(k) \tag{5.7}$$

$$\beta = e_N(k)/D_{Nr} \tag{5.8}$$

$$f_Q(k+1) \quad = \quad \begin{cases} f_Q(0) & ; if\ k = 0 \\ f_Q(k) + \beta & ; if\ D_{Nr} \leqslant D_N \\ f_Q(k) + \beta/2 & ; if\ D_{Nr} > D_N \end{cases} \tag{5.9}$$

$$L_Q(k+1) = f_Q(k+1) * L_{Qr}(k+1) \tag{5.10}$$

Table 5.1: ADTH Controller Pseudocodes

```
ADTH Controller Timer Interrupt (interval) {
    if (packets received != 0) // Calculate target queue threshold
        mean packet size := number of bytes sent / number of packets sent;
        queue throughput := number of bytes sent / sampling period;
        estimated queuing delay := target nodal delay - mean MAC delay;
        estimated queue size := estimated queuing delay * queue throughput;
        estimated queue length := estimated queue size / mean packet size;

        // Measure error and normalize it for adaptation of estimation (feedback loop)
        error := target nodal delay - measured nodal delay;
        beta := error / target nodal delay; // normalized error
        if (target nodal delay ⩽ measured nodal delay)
            queue length factor += beta;
            queue length factor := MIN(MIN_ QLEN_ FACTOR,  queue length factor);
        else
            queue length factor += beta/2
            queue length factor := MAX(queue length factor, MAX_ QLEN_ FACTOR);
        end if
        //Calculate target queue threshold for next interval
        target queue length := queue length factor * estimated queue length;

        //Sanity check to make sure target queue length is within the limit range
        if (target queue length < MIN_ Q_ LIMIT)
            target queue length := MIN_ Q_ LIMIT;
        end if
        if (target queue length > MAX_ Q_ LIMIT)
            target queue length := MAX_ Q_ LIMIT;
        end if
    else // Reset queue threshold if no traffic
        target queue length := MAX_ Q_ LIMIT;
    end if

    // Remove packets if current queue length is greater than target length
    if (target queue length < current queue length)
        purge queue
        update enqueIndex or dequeIndex
    endif
    Reset statistics and performance metrics
}
```

Table 5.1 is a simplified version of pseudocodes which capture the core design of the ADTH controller during a timer interrupt for each sampling interval. There are two important parameters that control the accuracy and overhead of the ADTH controller. They are the queue length factor (f_Q) and also the sampling time (t_s). These two parameters are discussed further in Subsections 5.3.3 and 5.3.4.

The target queue threshold is estimated at each interval and compared to current queue length. If the target queue threshold is smaller than the current queue length, all packets beyond the target queue threshold are discarded from the tail of queue (drop tail) or the head of queue (drop front). This action discards packets which possibly cannot arrive at their destinations within the required end-to-end delay at an earlier stage. Hence, the early discard not only can alleviate congestion indirectly but can also reduce wastage of bandwidth for transmitting the packets which will be discarded eventually at the destinations. Drop front policy is preferred and recommended for ADTH, refer to Subsection 5.3.5 for further details.

5.3.1 Queuing Delay Measurement

Queuing delay (D_Q) is the amount of time that a packet stays inside a queue before the packet is dequeued and transmitted out. A circular buffer (Fig. 5.2) is used to record the entry time of each packet into a queue. Two indexes (producer and consumer indexes) are used to keep track of packets entering or leaving the queue in order to obtain per packet queuing delay. The ring buffer is set to the same size as the queue. The indexes wrap over when they reach the top of the ring buffer. The pseudocodes for this simple mechanism are listed in Table 5.2.

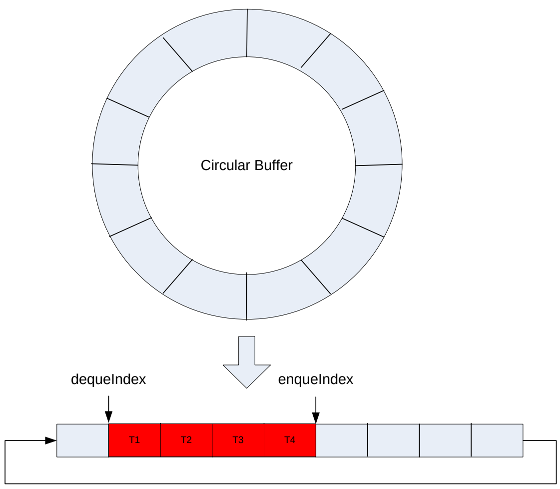


Figure 5.2: Circular Buffer

Table 5.2: Queuing Delay Measurement Pseudocodes

```
Initialize parameters {
        enqueIndex := 0; //producer index
        dequeIndex := 0; //consumer index
        ringBuffer[MAX_QUEUE_SIZE] := 0;
        queueDelay := 0;
}

Enqueue packet {
        ...
        if (IFQ not full)
                ringBuffer[enqueIndex] := current time;
                enqueIndex++;
                enqueIndex := enqueIndex mod MAX_QUEUE_SIZE;
                Insert packet into IFQ
        end if
        ...
}

Dequeue packet {
        ...
        if (IFQ not empty)
                queueDelay := current time - ringBuffer[dequeIndex];
                dequeIndex++;
                dequeIndex := dequeIndex mod MAX_QUEUE_SIZE;
                Remove packet from IFQ
        end if
        ...
        Record maximum queuing delay
}
```

5.3.2 MAC Delay Measurement

The pseudocodes for MAC delay measurement mechanism are listed in Table 5.3. MAC delay (D_M) is taken by measuring the interval between 2 consecutive packets being dequeued from a queue (Fig. 5.3). This approach is transparent to all MAC layer access method and enables MAC delay to be measured above MAC layer. MAC delay (Eq. 5.11) here refers to the aggregation of MAC contention delay, transmission delay, processing delay at MAC and propagation delay.

Table 5.3: MAC Delay Measurement Pseudocodes

```
Initialize parameters {
    macDelay := 0;
    macDelayStart := 0;
    macDelayMean := 0;
    queueState := idle;
}

Dequeue packet {
    ...
    Remove packet from IFQ
    ...
    // Measure MAC delay
    if (previous queue state not idle)
        t := current time;
        macDelay := t - macDelayStart;
    else
        macDelay := macDelayMean;
    end if
    // Calculate mean MAC delay
    macDelayMean += macDelay;
    macDelayMean /= 2;

    // Reset MAC delay start time
    macDelayStart := t;

    // Record current queue state
    if (queue empty)
        queueState := idle;
    else
        queueState:=busy;
    end if

    Send packet down to MAC layer
}
```

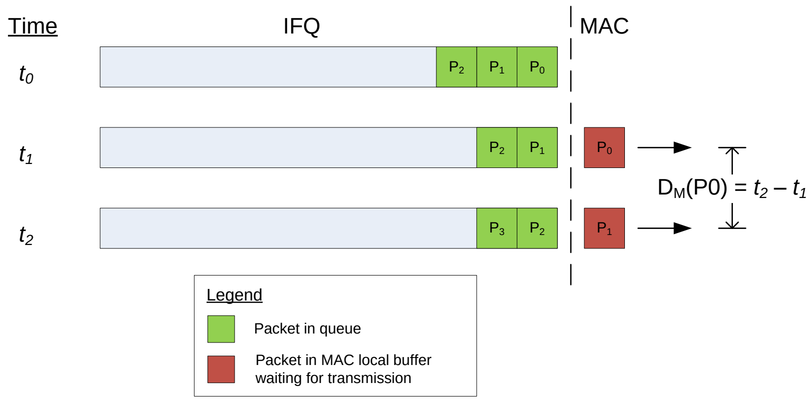


Figure 5.3: MAC Delay Measurement

$$
\begin{aligned}
D_M \quad = \quad & contention\, delay \\
& + propagation\, delay \\
& + transmission\, delay \\
& + retransmission\, delay\, (if\; any) \\
& + acknowledgement\, delay\, (if\; any) \\
& + MAC\, processing\, delay\, (= 0\, in\, NS - 2)
\end{aligned}
\tag{5.11}
$$

$$
\bar{D}_M = \frac{\bar{D}_M + D_M}{2} \tag{5.12}
$$

Mean MAC delay ($\bar{D}_M$) is calculated for ADTH target queue threshold estimation. The previous queue state is tracked to counter measure the queue idle time. Without checking the queue state, queue idle time will be included as part of MAC delay. This will lead to wrong measurement of MAC delay.

5.3.3 Feedback Control Loop

The feedback control loop in the ADTH controller aims to adjust the queue length factor (f_Q) based on the normalized error measured from control variable (D_N) (Eqs. 5.7 - 5.10). The initial queue length factor ($f_Q(0)$) is used as a starting point for the target queue threshold estimation before the feedback control kicks start. $f_Q(0)$ is set to 1, so the first target queue threshold estimation is 100% based on the system performance profile from the current sampling interval.

After the initial sampling interval, f_Q is updated based on the normalized error (β) measured at each sampling interval. f_Q is bounded by MIN_QLEN_FACTOR (lower bound) to prevent it from becoming a negative value. When the target nodal delay is set beyond the realistic range (which means too stringent to be achieved), the normalized error will be huge and in that case f_Q might be negative.

If that happened, f_Q is set to *MIN_ QLEN_ FACTOR.*

MAX_ QLEN_ FACTOR (upper bound) is used to prevent f_Q from being increased too aggressively due to false positive normalized error as a result of low traffic load or no congestion. When no congestion or moderate congestion, there will be either no backlog in the queue or the backlog is small. Hence, nodal delay would be low and causes huge normalized error when compared to the target nodal delay. Therefore, the queue length factor should be bounded to avoid underestimate of target queue threshold in the next sampling interval that may cause overshooting of nodal delay when the queue built up.

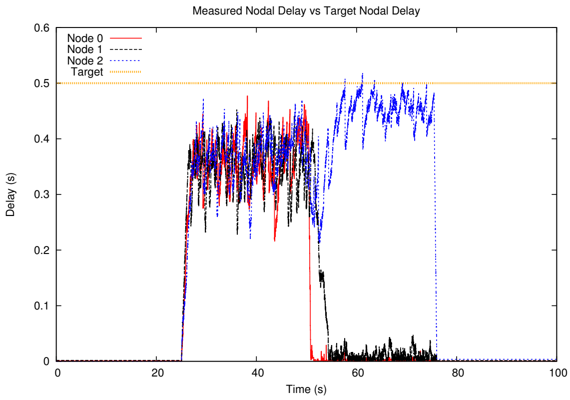
5.3.4 Sampling Interval

Since the system performance metrics needed for the ADTH controller are measured at a fixed sampling interval (t_s). The sampling interval can be fine-tuned in order to balance the trade-off required between the controller's accuracy and system overhead. If the sampling interval is small, the ADTH controller is run at a higher frequency ($f_s = \frac{1}{t_s}$) and thus incurs higher overhead in terms of computational power and execution time. The system generates more interrupts with a smaller sampling interval, higher interrupt produces more overhead to the system. Therefore, the sampling frequency cannot be set too high. Nevertheless, ADTH is a lightweight queue management scheme, which has the run time complexity of $O(1)$, therefore the overhead is low. If f_s is too low, the accuracy of the ADTH controller would be compromised due to slower response to network changes.

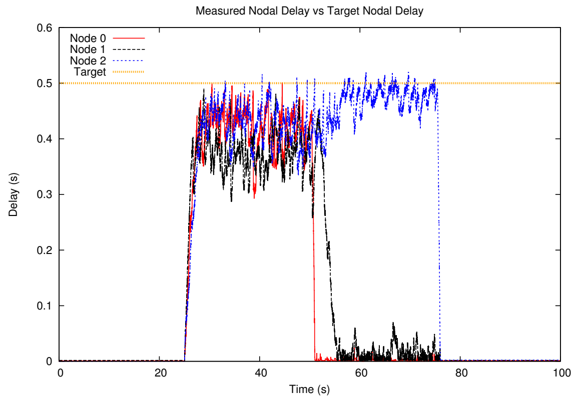
In addition, the setpoint of the controller (target nodal delay) is one of the important factors in deciding the sampling frequency. If the magnitude of target nodal delay is small, then the sampling rate needs to be high as fast response to the output error is needed. Otherwise, nodal delay may largely overshoot the target due to the system cannot cope with the changes. While for larger target nodal delay, there is enough room for the ADTH controller to react to the network changes. Therefore, the sampling frequency can be lowered. If the sampling frequency is too high, it may cause the system under utilization.

Fig. 5.4 shows the system performance based on the nodal delay measured with different sampling intervals for target nodal delay of $0.5s$. The simulation configurations and traffic profiles from Chapter 4 (Section 4.3) are used for this simulation. Most of the time the nodal delay is maintained much lower than the target when $t_s \ll D_{Nr}$. When $t_s = \frac{1}{2}D_{Nr}$, the nodal delay is regulated around the target. However, when $t_s > \frac{1}{2}D_{Nr}$, nodal delay overshoots the target at a higher frequency. Therefore, the rule of sampling interval is shown in Eq. 5.13.

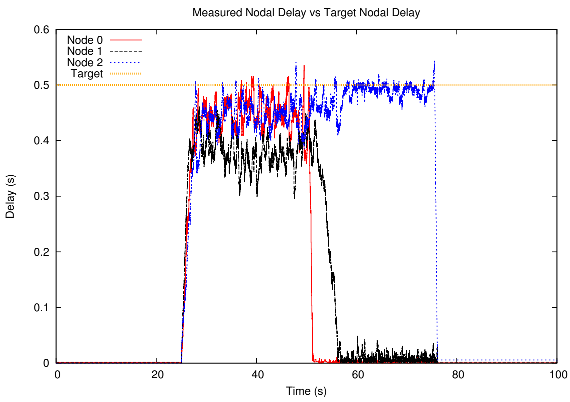
$$t_s \leqslant \frac{1}{2} D_{Nr} \tag{5.13}$$



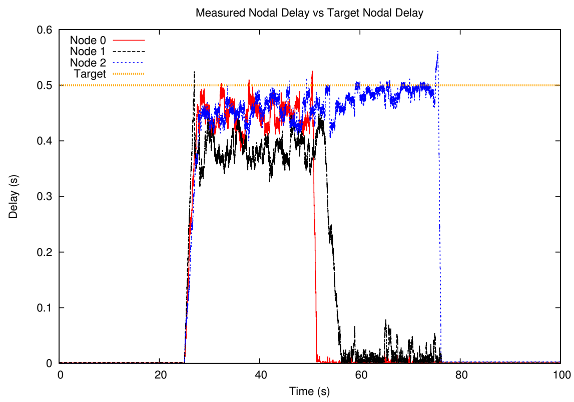
(a) $t_s = 0.1s$



(b) $t_s = 0.25s$



(c) $t_s = 0.5s$



(d) $t_s = 1s$

Figure 5.4: ADTH Controller Performance with Different Sampling Intervals (D_{Nr} $= 0.5s$)

5.3.5 Dropping Policy

Most of the IFQ of wireless nodes is a FIFO with drop tail (FIFO-DT) dropping policy by default. New packets are dropped when the queue is full for FIFO-DT. If the IFQ is configured to use drop front policy (FIFO-DF) then old packets from the head of the queue are discarded when the queue is full. New packets are enqueued into IFQ after the old packets are discarded. Different type of dropping policy may have different impacts on the system performance especially network delay and packet loss. These two performance metrics are used to decide which dropping policy is the best choice for ADTH.

Both the FIFO-DT and FIFO-DF queue disciplines can be coupled with the ADTH scheme to get nodal delay bounded to the target delay requirement. Without the ADTH scheme, FIFO-DT and FIFO-DF have no control over nodal delay.

Nodal delays for FIFO-DT and FIFO-DF are only constrained by the constant maximum queue limit and queue throughput. When ADTH is enabled, FIFO-DT and FIDO-DF have a movable queue threshold based on the current system performance and network dynamics instead of a constant maximum queue limit. ADTH manipulates the maximum limit of the FIFO to bound nodal delay. The movable maximum limit of the queue is known as target queue threshold. When the target queue threshold is smaller than the current queue length, packets are discarded from the head of the queue (ADTH-DF) or the tail of the queue (ADTH-DT) depending on the dropping policy chosen for the ADTH scheme.

In this subsection, the system performance has been investigated for IFQ with or without ADTH enabled and with different dropping policies. For this investigation, the simulation configurations for delay analysis in previous chapter (Sec. 4.3) are used. If the maximum OWD required (OWD_R) for UDP packets is $1s$, then the target nodal delay (D_{Nr}) is set to $0.5s$ since the maximum hops are 2. R is set to $1.8Mbps$ to simulate congested network. The sampling interval (t_s) of the ADTH controller is set to $\frac{1}{2}D_{Nr}$.

The dropping policy has a direct impact on the maximum nodal delay experienced by a packet. The simulation (Table 5.4) shows that the drop front policy has lower nodal delay when compared to the drop tail policy. However, both FIFO-DT and FIFO-DF have a similar packet loss rate and UDP delivery ratio. By examining OWD of each UDP packet, it can be seen that most of the packets get delivered to the destination exceed the required OWD_R. If the packets are time sensitive and need to reach the destination before the deadline then $\simeq 48\%$ of packets for CBR flow from Node 0 to Node 2 (*F1*) and $\simeq 83\%$ of packets for UDP flow from Node 2 to Node 0 (*F2*) are rendered useless. The lower the OWD_R, the higher the deadline miss ratio.

Nodal delay can be bounded by applying the ADTH queue management scheme to FIFO-DT and FIFO-DF, the simulation results have strongly supported the claim on this. The results in Table 5.4 show that network delay can be controlled without sacrificing the packet delivery ratio and not exacerbating the packet loss rate too much. The overall packet delivery ratio is only decreased by 3% - 4%, whereas the performance gain is far more valuable compared to the packet loss count. The deadline miss ratio for the flow from N0 to N2 (F1) has improved by at least 40% - 48% depending on the dropping policy of the FIFO queue and the ADTH scheme; for flow from N2 to N0 (F2) 75% - 83% improvement has been observed in deadline miss ratio.

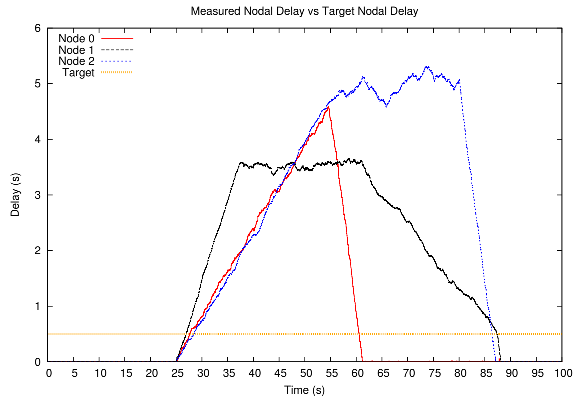
The simulation results suggest that drop front policy should be used for delay-sensitive traffic instead of drop tail policy. Although nodal delay is constrained for FIFO-DT with ADTH enabled, the combination of FIFO-DT with either ADTH-

Table 5.4: Overall Performance for Different Queue Management Schemes

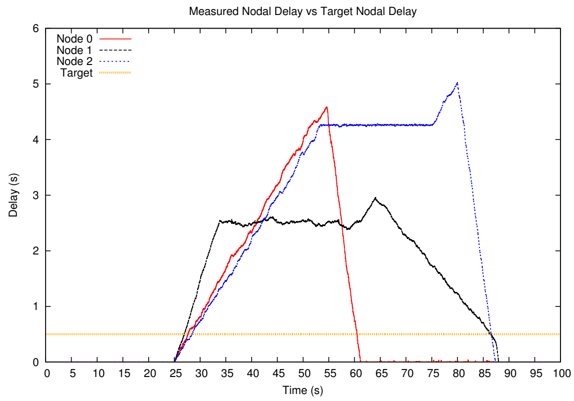
Queue Management Schemes	Average OWD (s)		Maximum OWD (s)		Packets Delivery (%)		Deadline Miss (%)	
	F1	F2	F1	F2	F1	F2	F1	F2
FIFO-DT	2.1653	5.0934	8.2236	8.5012	91.16	83.73	48.54	83.61
FIFO-DF	1.8681	4.4835	7.1301	7.2341	90.69	83.95	47.05	83.49
FIFO-DT & ADTH-DT	0.2270	0.5107	1.1542	1.2316	87.40	80.75	6.76	8.22
FIFO-DT & ADTH-DF	0.2245	0.5225	1.0598	1.0673	86.78	81.11	1.15	2.39
FIFO-DF & ADTH-DT	0.2073	0.4858	0.9626	0.9551	87.50	80.71	0	0
FIFO-DF & ADTH-DF	0.1945	0.4696	0.9281	0.9427	87.10	81.12	0	0

DT or ADTH-DF cannot control nodal delay and OWD within the required range efficiently (Figs. 5.5 and 5.6). It is reflected by the non-zero deadline miss ratio. The deadline miss ratio for FIFO-DT with ADTH-DT is higher than coupling of FIFO-DT with ADTH-DF. This is because if the previous packet has exceeded the target nodal delay, then there is a high possibility that the following packets in the queue also exceed the target nodal delay. Packets are dropped earlier instead of being passed on to the next hop and wasting the network resources if the packets cannot meet the deadline. This explains why drop front policy has lower deadline miss ratio.

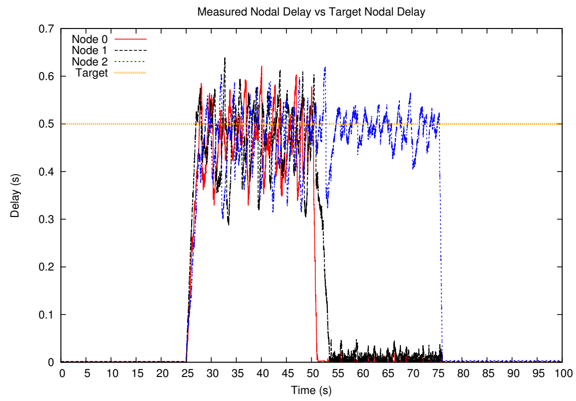
When FIFO-DF is coupled with the ADTH scheme, all packets received at the destination meet the deadline. The deadline miss ratio is 0% for both the ADTH-DF and ADTH-DT schemes. The packet loss ratio and packet delivery ratio are similar for both the ADTH-DF and ADTH-DT schemes. However, FIFO-DF with ADTH-DF would be recommended since this combination gives better results in terms of bounded nodal delay and bounded OWD.



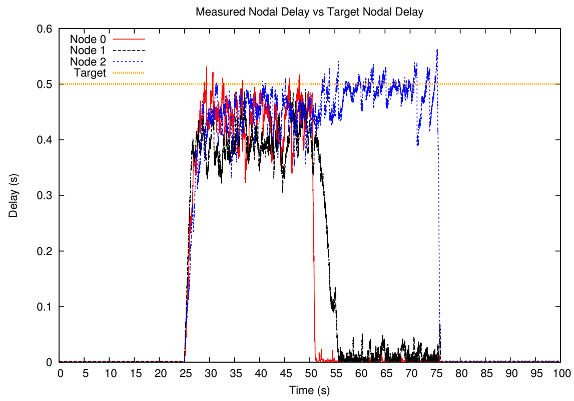
(a) FIFO-DT



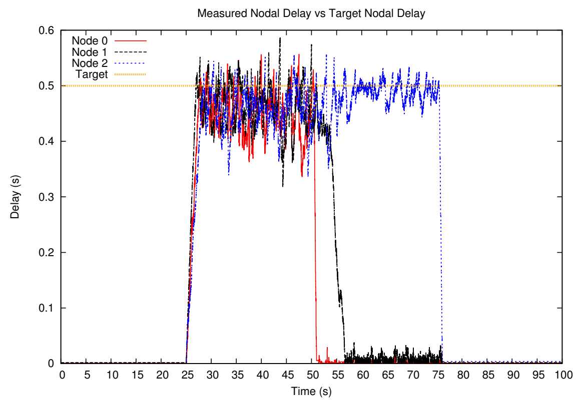
(b) FIFO-DF



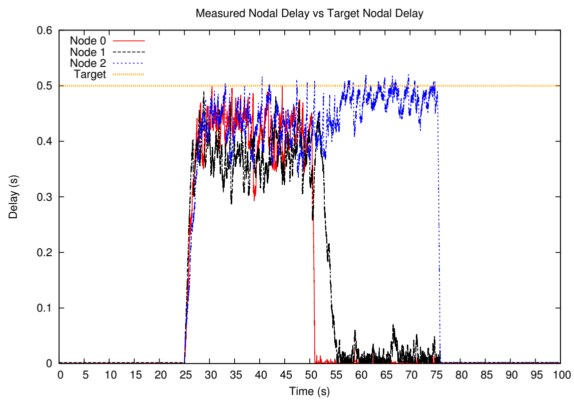
(c) FIFO-DT with ADTH-DT



(d) FIFO-DF with ADTH-DT

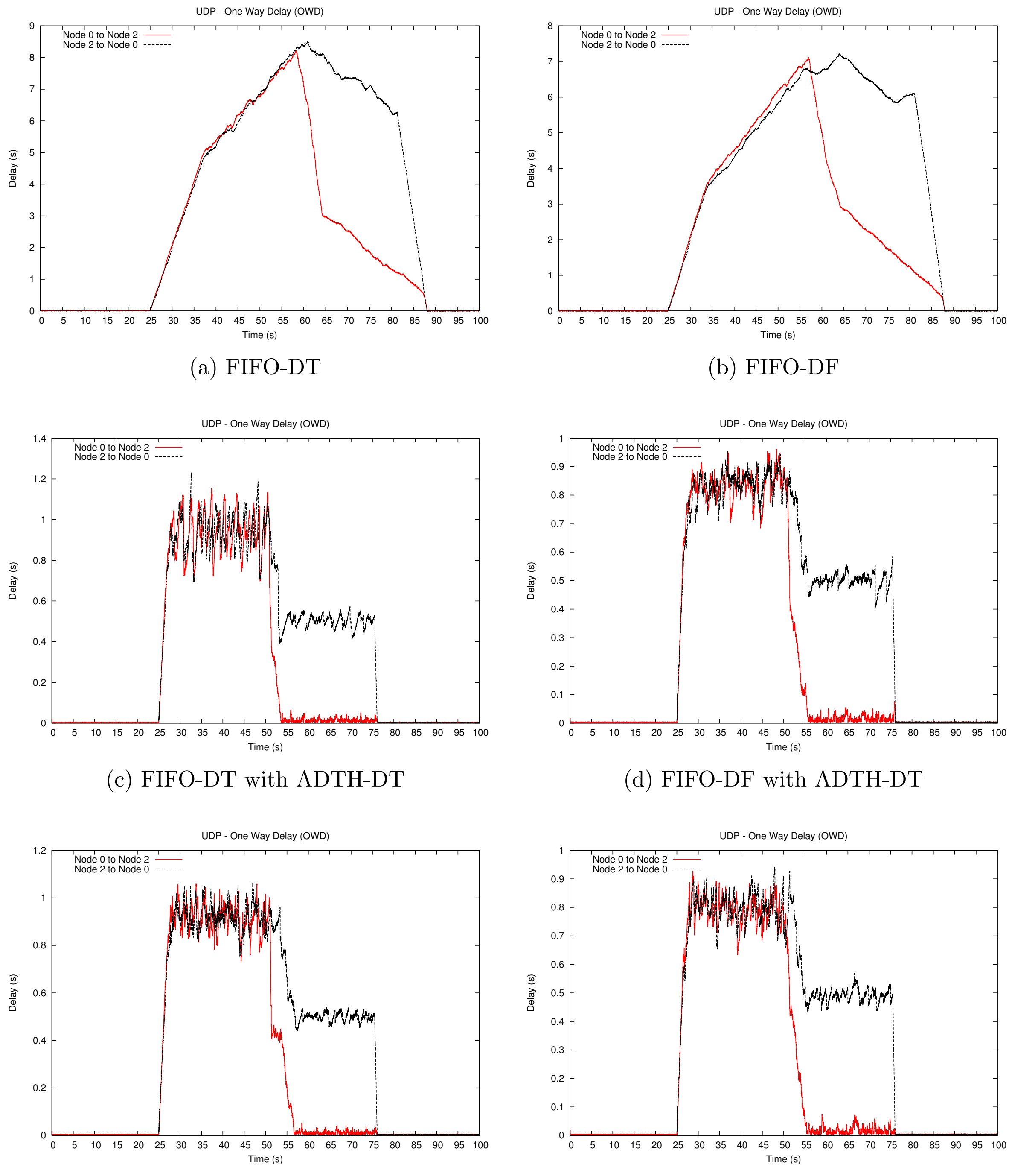


(e) FIFO-DT with ADTH-DF



(f) FIFO-DF with ADTH-DF

Figure 5.5: Nodal Delay for Different Queue Management Schemes ($D_{Nr} = 0.5s$)



(a) FIFO-DT

(b) FIFO-DF

(c) FIFO-DT with ADTH-DT

(d) FIFO-DF with ADTH-DT

(e) FIFO-DT with ADTH-DF

(f) FIFO-DF with ADTH-DF

Figure 5.6: UDP OWD for Different Queue Management Schemes ($OWD_r = 1s$)

5.4 ADTH Limitations and Assumptions

ADTH is a lightweight and adaptive queue management scheme, however with the ADTH scheme alone cannot solve network-wide delay related issues. Here is a list of limitations and assumptions for the ADTH queue management scheme:

1. The target nodal delay requirement cannot be too stringent. ADTH bounds nodal delay via queuing delay constraint to compensate for MAC layer delay observed at each node. ADTH has no control on MAC layer delay. Therefore, if MAC layer delay exceeds the target nodal delay requirement, ADTH is incapable to constrain nodal delay to the target. The lower bound of the target queue threshold is 1 packet (MIN_Q_LIMIT) when this happens. Packets are dropped aggressively in this case until MAC layer delay is reduced (e.g. less congestion) or the delay requirement is relaxed.

2. There is no upper bound for the target nodal delay requirement. However, there is an upper bound for the target queue threshold estimated by ADTH, which is the maximum limit of the queue size. If the estimated target queue threshold > maximum queue limit (MAX_Q_LIMIT), it will be capped to MAX_Q_LIMIT.

3. ADTH is designed to constrain nodal delay based on delay requirements from users or network administrators with higher packet loss as a trade-off. If no congestion in a network, ADTH behaves similarly to Drop Tail or Drop Front schemes. When congestion started, ADTH regulates its queue limit to constrain nodal delay by dropping packets.

4. ADTH is designed to bound nodal delay in order to enable bounded end-to-end delay for delay-sensitive traffic flows if such routing paths exist between a source and a destination that can satisfy the end-to-end delay requirement.

5. It is the role of routing agents to discover paths that can meet end-to-end delay requirements for particular real-time traffic flows. It is also the responsibility of routing agents to handle changes of paths due to mobility of nodes to ensure the end-to-end delay requirement is met. Delay-based QoS-aware routing can be adopted to discover paths. Therefore, only nodes with ADTH capability are advised to be considered for routes selection so that a deterministic end-to-end delay can be achieved.

6. Processing delay and queuing delay above network layer are considered as application layer delay and are not considered by the ADTH controller in the estimation of target queue threshold. These delays are application specific and introduced by source nodes. Packets received at intermediate nodes are processed at network layer for next hop route lookup before being enqueued to the IFQ. The processing delay is minimal when compared to queuing delay and MAC layer delay. Therefore, the processing delay incurred is not considered by the ADTH controller.

7. ADTH is a generic approach that is transparent to MAC variants. Thus, this approach avoids interoperable issues between ADTH enabled nodes and legacy nodes. ADTH can be adopted by any nodes that supports timer interrupt and timestamp capability.

8. ADTH is not designed to maintain per flow QoS; all delay-sensitive traffic is treated as an aggregated traffic and thus fairness among flows is not considered for the design. However, scalability becomes the advantage of ADTH since no effort is needed to track per flow performance.

9. It is not the aim of ADTH to alleviate congestion in a network. Delay-sensitive traffic is mostly carried over UDP transport agents that are known to be non-responsive to packet loss events and unable to adapt the sending rate. It is the responsibility of an application to adapt its sending rate.

10. Although ADTH can constrain nodal delay regardless of packet types as no classification is done at ADTH to differentiate the packet types. It is impractical to constrain nodal delay or end-to-end delay of TCP packets by dropping packets since no deadline is associated with TCP packets. Packet loss events may cause throughput degradation for TCP traffic. The TCP agent reduces its sending rate and retransmits a packet when packet loss is detected. Hence, ADTH is not favoured for traffic carried over TCP if throughput is a concern.

5.5 Simulation Results

The ADTH scheme has been validated in NS-2 with a small stationary wireless ad hoc network from various aspects. Drop front policy gives the best performance from the findings in Subsection 5.3.5, therefore ADTH-DF is used for all simulation scenarios. ADTH and ADTH-DF are used interchangeability throughout the text for the rest of the sections.

NS-2 simulation configurations are listed in Table 5.5. Fig. 5.7 shows the simulation network topology. The default queue discipline is Drop Tail FIFO and the default queue size is 1000 packets.

Fig. 5.8 shows the traffic load being injected into the network during the simulation. The simulation starts with Node 0 sending CBR traffic at rate R to Node 2. After $25s$ Node 2 starts the CBR transmission to Node 0 at the same data rate. At $50s$, Node 0 reduces its rate to half and then continue at this rate to the end of the simulation. While Node 2 reduces its rate to half at $75s$, and continue this

Table 5.5: NS-2 Simulation Configurations

Parameters	Configurations
Radio Propagation Model	TwoRayGround
Wireless Mode	IEEE 802.11b
Interface Queue	DropTail/PriQueue
Routing Protocol	AODV
Virtual Carrier Sensing	OFF
Transmit Power	15dBm
Transmission Range	30m
Carrier Sensing Range	$\gg$ 2x transmission range
Transmission Data Rate	11$Mbps$ (no auto-fallback)

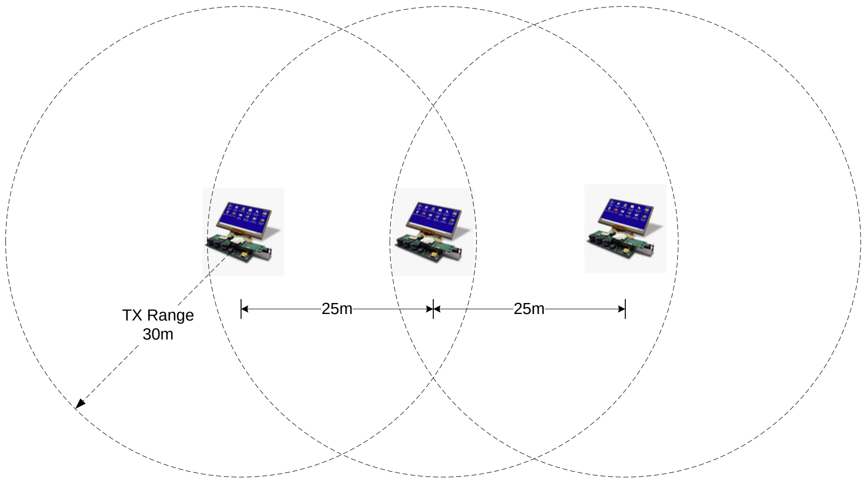


Figure 5.7: Simulation Network Topology

rate towards the end of the simulation. From this simulation, impacts of single directional and bi-directional traffic and intensity of traffic load are simulated.

The level of congestion changes when the traffic load injected into the network varies according to the traffic profile in Fig. 5.8. Consequently, drastic changes on the system performance are observed (at simulation time 25s, 50s and 75s) in the simulation results for most of the simulation scenarios in this section. The changes are such as a sudden increase or decrease of queuing delay, nodal delay, throughput, etc.

The effectiveness and efficiency of the ADTH scheme have been analyzed in the simulation scenarios with QoS metrics, such as packet loss ratio, OWD, miss target nodal delay ratio, deadline miss ratio, UDP goodput and data yield. Miss target nodal delay ratio is the ratio for the number of packets that experience nodal delay greater than D_{Nr} over the number of packets sent at a particular node. Deadline miss ratio is the ratio for the number of packets that get delivered to a destination after the deadline over the number of packets received at the

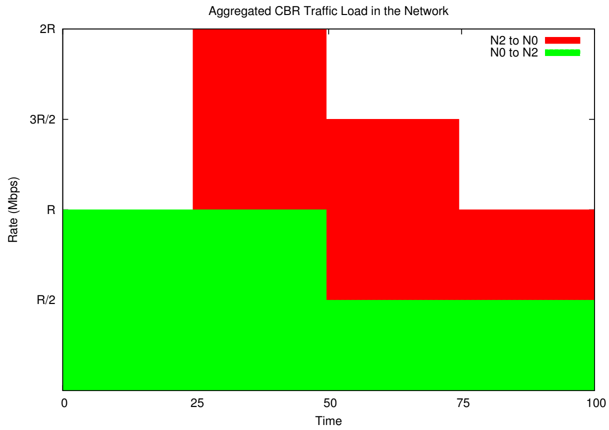


Figure 5.8: Traffic Load for Simulation

destination. While data yield is the ratio for the number of packets that get delivered to a destination before the deadline over the number of packets sent from the source. It is used to show the usefulness of packets arrived at the destination. High packet delivery ratio does not guarantee high data yield if network delay is not controlled effectively.

5.5.1 Scenario 1: Different target nodal delay

5.5.1.1 (a) Same D_{Nr} for all wireless nodes

In Scenario 1(a), R is configured to $1.8Mbps$ to create a highly congested network when bi-directional CBR traffic started. ADTH has been validated with different D_{Nr} ranging from [0.2s..1.0s] with stepping of $0.2s$. All nodes are assigned with the same D_{Nr} in each stepping. The maximum allowable OWD (OWD_R) of CBR flows for both directions are $2 \times D_{Nr}$ since the sources are 2 hops away from the destinations. This simulation scenario is to show that ADTH-DF is able to constrain nodal delay to different target value specified by the delay requirement.

Tables 5.6 and 5.7 show the system performance for FIFO-DF and FIFO-DF with ADTH-DF enabled. When ADTH-DF is enabled, none of the packets received at the destinations exceeds the OWD_R. So the deadline miss ratio is 0%. Those packets that cannot be delivered to the destination on time have been dropped either at the source node or the intermediate nodes. Consequently, less collision occurs when ADTH-DF is enabled. However, the packet loss ratio is higher by 2% - 4% depending on the D_{Nr} value. Although the packet loss ratio is higher when ADTH is enabled, the performance gain is very convincing from the network delay perspective.

Table 5.6: Scenario 1(a): Overall System Performance for ADTH-DF

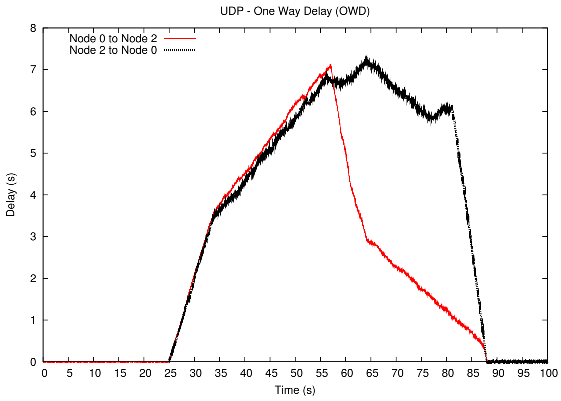
D_{Nr}	Collision	Full Q	Max OWD (s)		Packet Loss (%)	
(s)	Count	Drop	N0→N2	N2→N0	N0→N2	N2→N0
0.2	417	5104	0.3680	0.3921	13.26	18.93
0.4	446	5063	0.7504	0.7398	12.70	19.32
0.6	436	4945	1.1358	1.1197	12.43	18.85
0.8	448	4899	1.4804	1.4523	12.24	18.75
1.0	450	4805	1.8899	1.8626	12.18	18.19

Table 5.7: Scenario 1(a): Overall System Performance for FIDO-DF

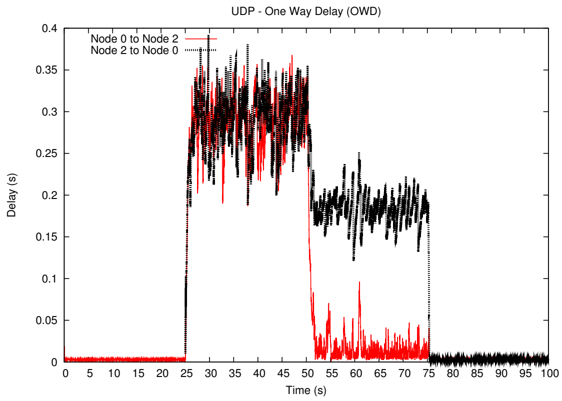
Collision	Full Q	Max OWD (s)		Packet Loss (%)	
Count	Drop	N0→N2	N2→N0	N0→N2	N2→N0
603	3988	7.1301	7.2341	9.31	16.05

Fig. 5.9 shows the OWD measured for both CBR flows throughout the simulation. For FIFO-DF, OWD increases when the network starts congested from 25s. When the congestion is eased at 50s as a result of reduced traffic load in the network, the OWD decreases. The peak of OWD is $> 7s$ for FIFO-DF queue discipline. When ADTH-DF is enabled, OWD is bounded below OWD_R for all D_{Nr}.

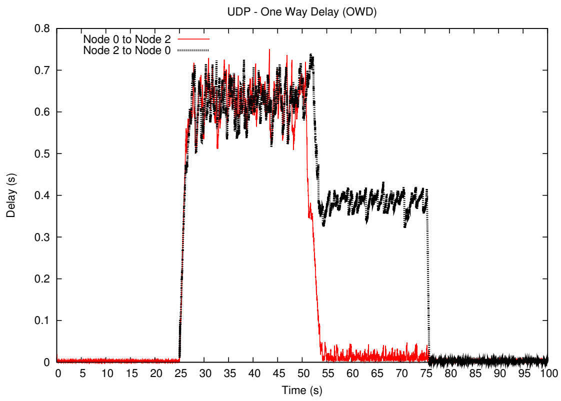
The deadline miss ratio for FIFO-DF is high. The deadline miss ratio increases from 38% to 53% for flow 0 and from 79% to 87% for flow 1 when D_{Nr} decreases from 1.0s to 0.2s.



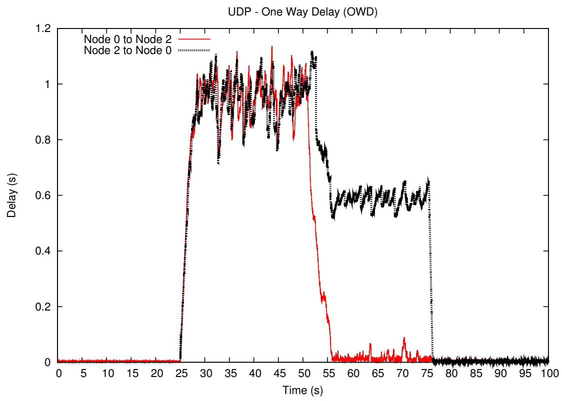
(a) FIFO-DF ($D_{Nr} = 0.2s - 1.0s$)



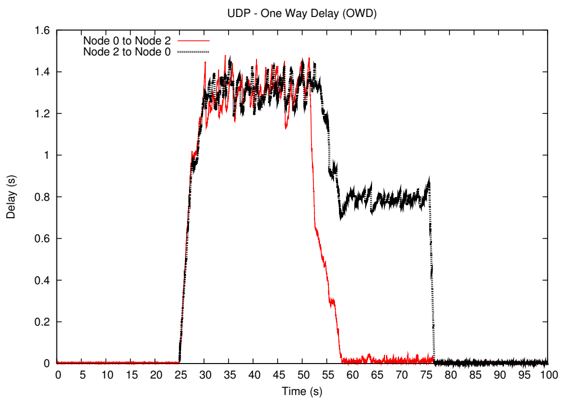
(b) ADTH-DF ($D_{Nr} = 0.2s$)



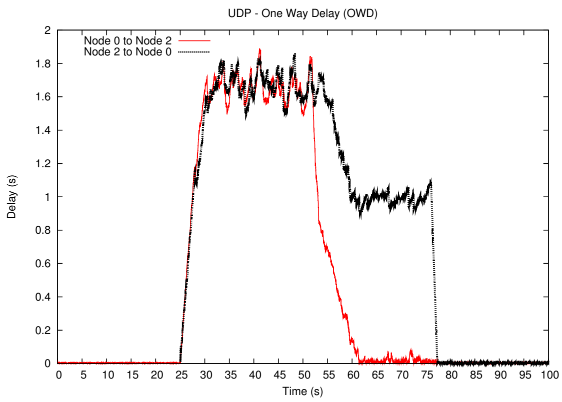
(c) ADTH-DF ($D_{Nr} = 0.4s$)



(d) ADTH-DF ($D_{Nr} = 0.6s$)



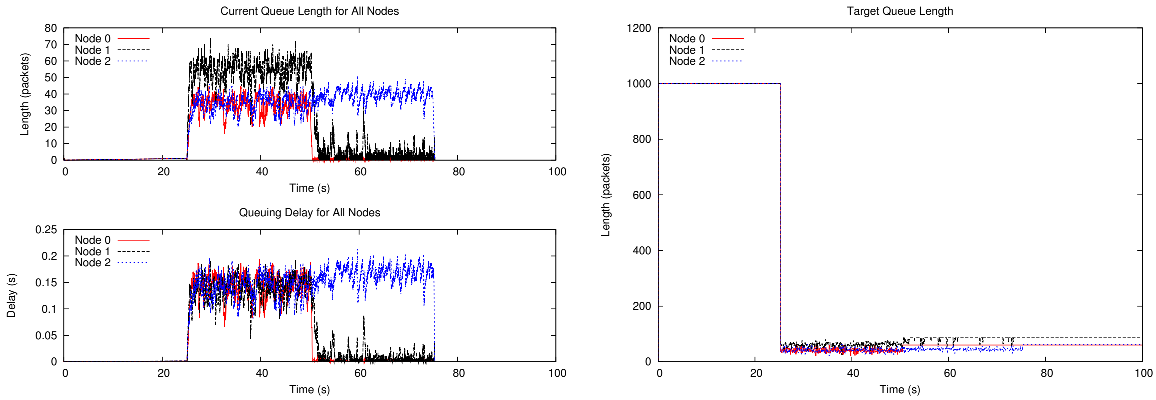
(e) ADTH-DF ($D_{Nr} = 0.8s$)



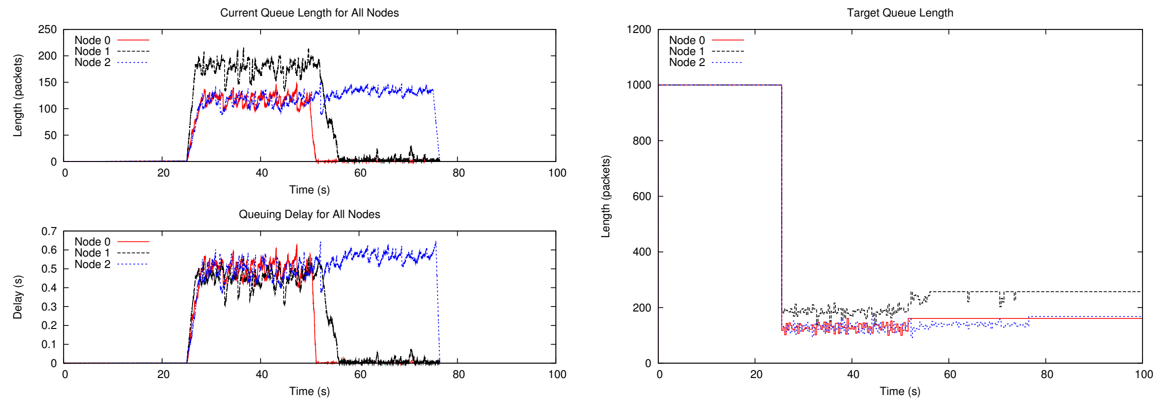
(f) ADTH-DF ($D_{Nr} = 1.0s$)

Figure 5.9: Scenario 1(a): Overall OWD Trend (ADTH-DF versus FIFO-DF)

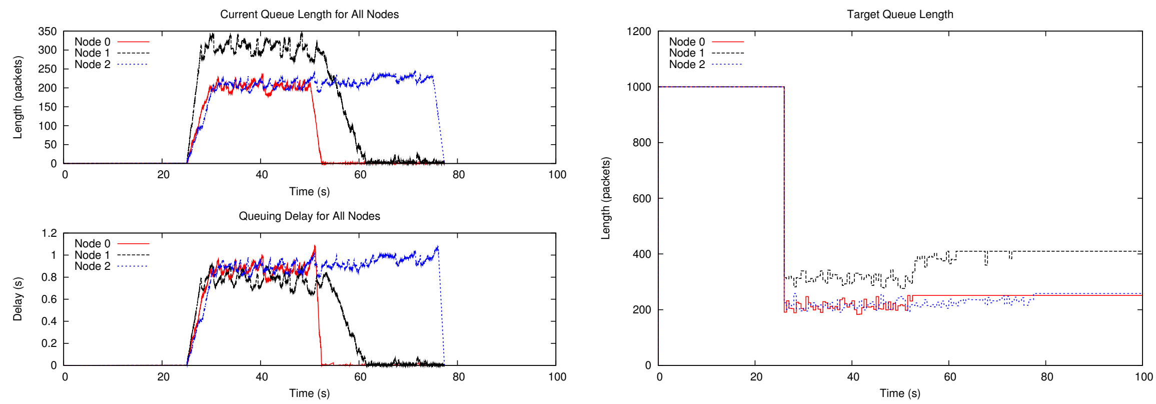
When ADTH is enabled, the target queue threshold is varied and estimated periodically (Fig. 5.10). The target queue threshold estimated is different for each node. The estimation is done solely based on the system performance profile of each node and D_{Nr} without relying on information gathering from other nodes. Before $25s$, the target queue threshold is the same as default maximum limit because no congestion detected. Network starts congested after $25s$ due to bi-directional traffic and the traffic load is double. Therefore, a drastic change on the target queue threshold is observed. At 50s, the total traffic load in the network is reduced to *3R/2*. Although network congestion is eased at 50s, the target queue threshold is not allowed to bounce back to default maximum limit. It is maintained at the maximum estimated queue threshold from the past for anticipation of network congestion.



(a) $D_{Nr} = 0.2s$



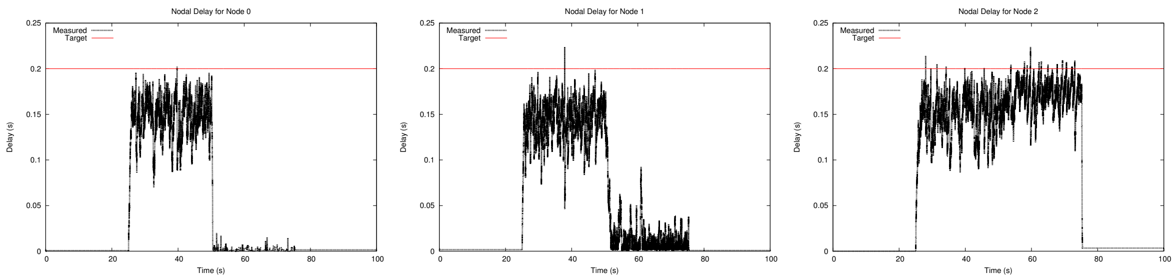
(b) $D_{Nr} = 0.6s$



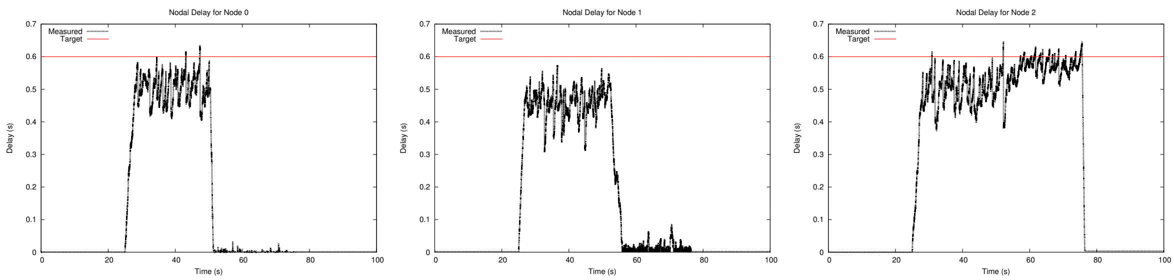
(c) $D_{Nr} = 1.0s$

Figure 5.10: Scenario 1(a): Queuing Delay Regulation via a Movable Target Queue Threshold Estimated by ADTH

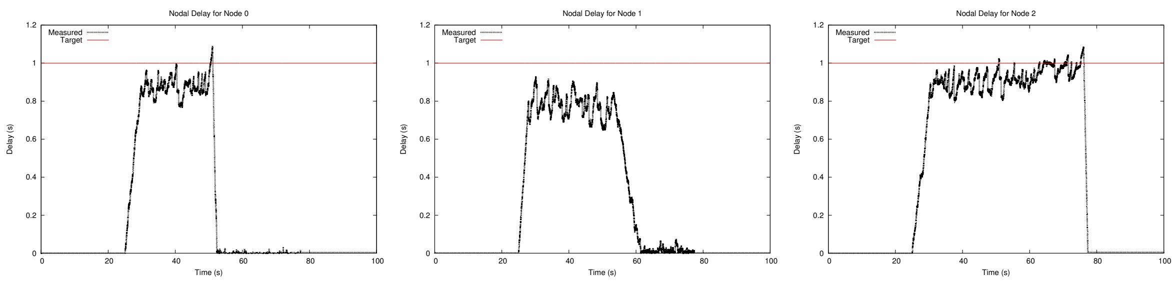
Simulation results in Fig. 5.10 also show that when the network is congested at $25s$, backlogs are built up quickly in the queue. The instantaneous queue length increases fast and the same to the queuing delay. After the network congestion is eased at $50s$, backlogs in the queue are drained quickly. The time taken for the system to drain a queue depends on the number of backlogs in queue and the system throughput. When target nodal delay is lower, the target queue threshold is lower. Consequently, less packets are backlogged in the queue. Thus, a shorter time is needed to drain the queue for $D_{Nr} = 0.2s$ compared to $D_{Nr} = 1.0s$ as shown in the figure.



(a) $D_{Nr} = 0.2s$



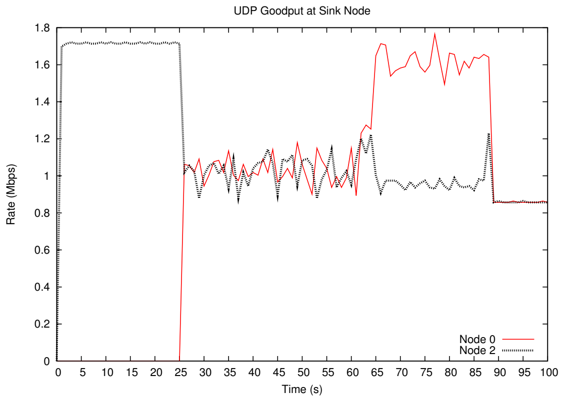
(b) $D_{Nr} = 0.6s$



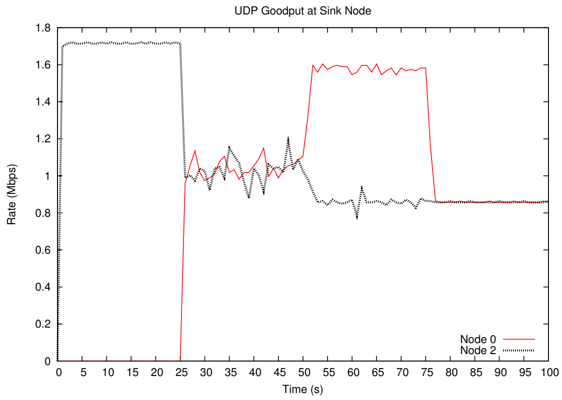
(c) $D_{Nr} = 1.0s$

Figure 5.11: Scenario 1(a): Nodal Delay for Wireless Nodes with ADTH Enabled

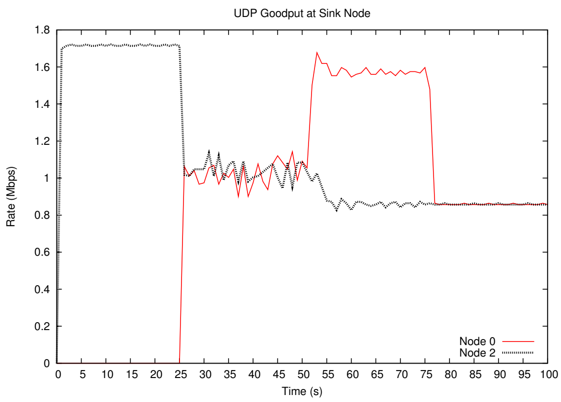
The OWD is constrained indirectly via bounded nodal delay (Fig. 5.11) at each hop. The sampling frequency can be increased to reduce overshoots observed. There is no packets missing the OWD_R deadline even though a few packets overshot the target nodal delay especially for Node 0 and Node 2. This is because of OWD is an aggregation of nodal delays from a source to a destination; the overshooting effect may be cancelled out at intermediate nodes which experience lower nodal delay. Even if it is not cancelled out; the miss target nodal delay ratio is small, such as 0.0059% for Node 0, 0.0036% for Node 1 and 0.7202% for Node 2 when $D_{Nr} = 0.2s$.



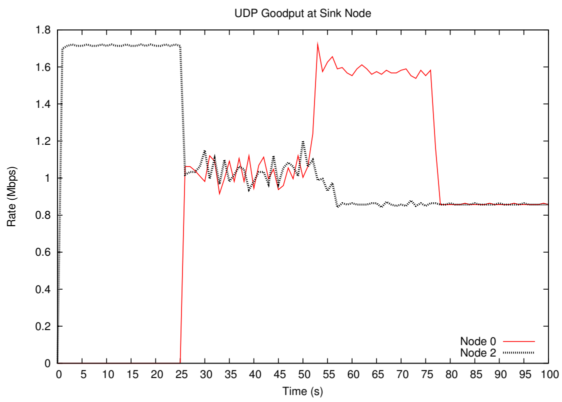
(a) FIFO-DF ($D_{Nr} = 0.2s - 1.0s$)



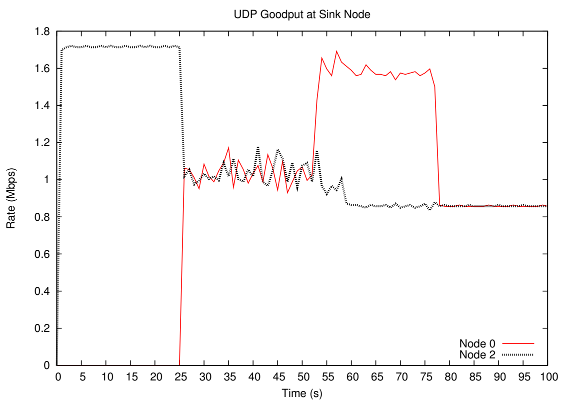
(b) ADTH-DF ($D_{Nr} = 0.2s$)



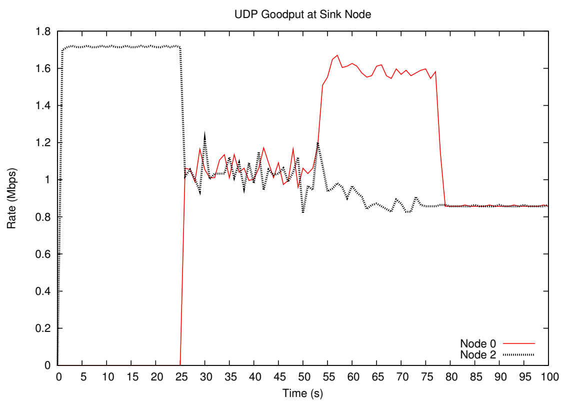
(c) ADTH-DF ($D_{Nr} = 0.4s$)



(d) ADTH-DF ($D_{Nr} = 0.6s$)



(e) ADTH-DF ($D_{Nr} = 0.8s$)



(f) ADTH-DF ($D_{Nr} = 1.0s$)

Figure 5.12: Scenario 1(a): UDP Goodput for ADTH-DF versus FIFO-DF

Both ADTH and FIFO-DF show the similar trend for UDP goodput (Fig. 5.12), but ADTH performs better. The UDP goodput of the ADTH scheme catches up very soon after the network is less congested. While for FIFO-DF, the UDP goodput catches up slower due to a longer congestion state. It takes a longer time to drain the full FIFO-DF queue, thus the network load is not reduced until the queue is drained even the sources have reduced the rate at time $50s$.

5.5.1.2 (b) Different D_{Nr} for each wireless node

After looking at the simulation case that D_{Nr} is homogeneous for all nodes, heterogeneous D_{Nr} has been investigated in this case. The same simulation configuration as above is used except that all nodes are set with different D_{Nr} with Node 0 is $1.0s$, Node 1 is $1.5s$ and Node 2 is $0.5s$. Therefore, OWD_R for CBR flow from Node 0 to Node 2 is $2.5s$, whereas OWD_R for CBR flow from Node 2 to Node 0 is $2s$. t_s is set to $\frac{1}{2}D_{Nr}$.

Only the simulation results of the ADTH-DF scheme are discussed in detailed as the performance results of FIFO-DF is the same as in Scenario 1(a). Fig. 5.13 shows that when ADTH-DF is enabled, the maximum OWD recorded is $\leqslant OWD_R$. The deadline miss ratio is 0% for both CBR flows when ADTH-DF enabled; it is 32.83% and 78.82% otherwise for FIFO-DF. While the packet loss ratio is only 3% higher when ADTH-DF is enabled.

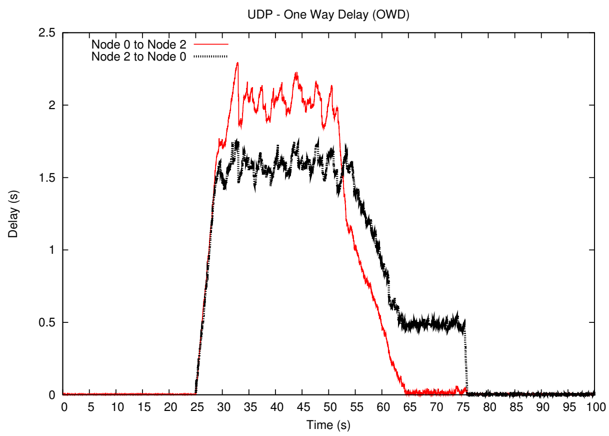


Figure 5.13: Scenario 1(b): Overall OWD Trend for ADTH-DF

This simulation shows that every node acts independently to bound nodal delay (Fig. 5.14) to achieve bounded OWD regardless of D_{Nr} for other nodes. This is because ADTH-DF is a stand-alone scheme. It estimates the target queue threshold for the nodes based on their own system performance profiles and nodal delay requirements without needing global information from surrounding nodes. Through the adaptation of target queue threshold, queuing delay (Fig. 5.15) on each node is constrained in order to bound nodal delay. The miss target nodal delay ratio is negligible.

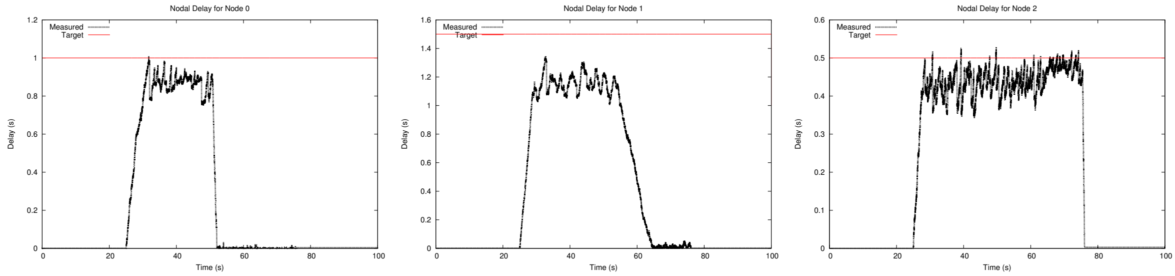
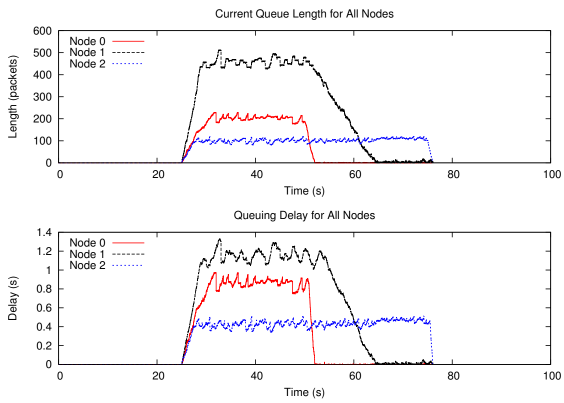
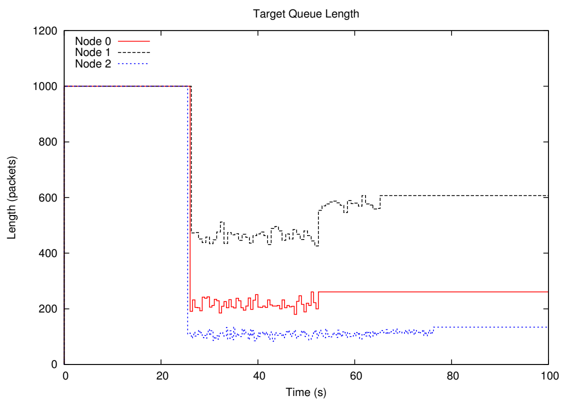


Figure 5.14: Scenario 1(b): Nodal Delay for Wireless Nodes with ADTH Enabled



(a) Instantaneous Queue Length versus Queuing Delay



(b) Target Queue Threshold

Figure 5.15: Scenario 1(b): Queuing Delay Regulation via a Movable Target Queue Threshold Estimated by ADTH

5.5.1.3 (c) Stringent D_{Nr}

As mentioned in the limitations, D_{Nr} cannot be too stringent as network congestion may cause large MAC layer delay. MAC layer delay is possibly higher than D_{Nr}. If this happens, the ADTH scheme will fail to bound nodal delay. A snapshot of this scenario is captured by setting $D_{Nr} = 0.01s$ with $R = 1.8Mbps$.

It is expected that the packet loss ratio is higher when compared to the Scenario 1(a) due to a more stringent delay requirement. The packet loss ratio is 14.05% for CBR flow from Node 0 to Node 2 and 21.37% for CBR flow from Node 2 to

Node 0 for this scenario. Nevertheless, the packet loss ratio is only higher by 4% to 6% if compared to FIFO-DF.

Fig. 5.16 shows that nodal delay cannot be controlled effectively due to the stringent D_{Nr}. Miss target nodal delay ratio is 3.01% for Node 0, 1.99% for Node 1 and 5.43% for Node 2. While the miss deadline ratio is 0.95% for CBR flow from Node 0 to Node 2 and 1.60% for CBR flow from Node 2 to Node 0. The miss target nodal delay ratio and miss deadline ratio will be higher if the network is more congested or D_{Nr} is further decreased. Therefore, it is important to have a reasonable target nodal delay requirement. Fig. 5.17 shows the overall OWD trend for both CBR flows. The miss deadline ratio is not as high as miss target nodal delay ratio as some overshoots are cancelled out by the aggregation of nodal delays of other nodes.

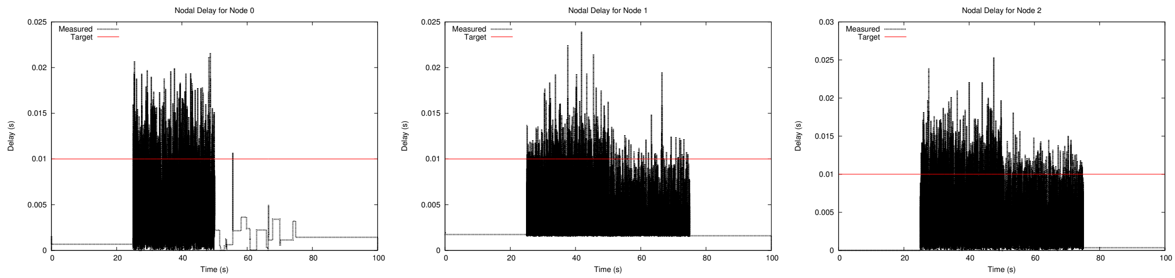


Figure 5.16: Scenario 1(c): Nodal Delay for Wireless Nodes with ADTH Enabled

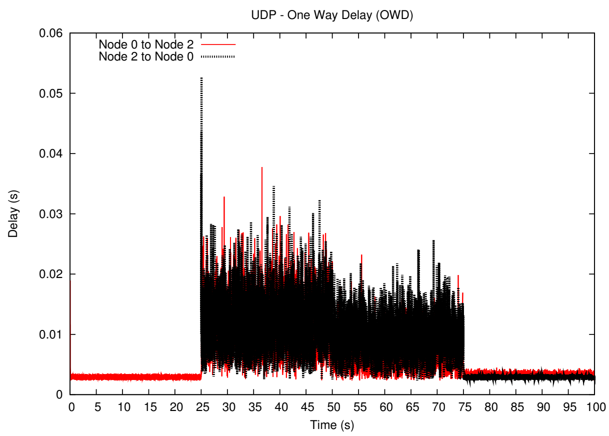
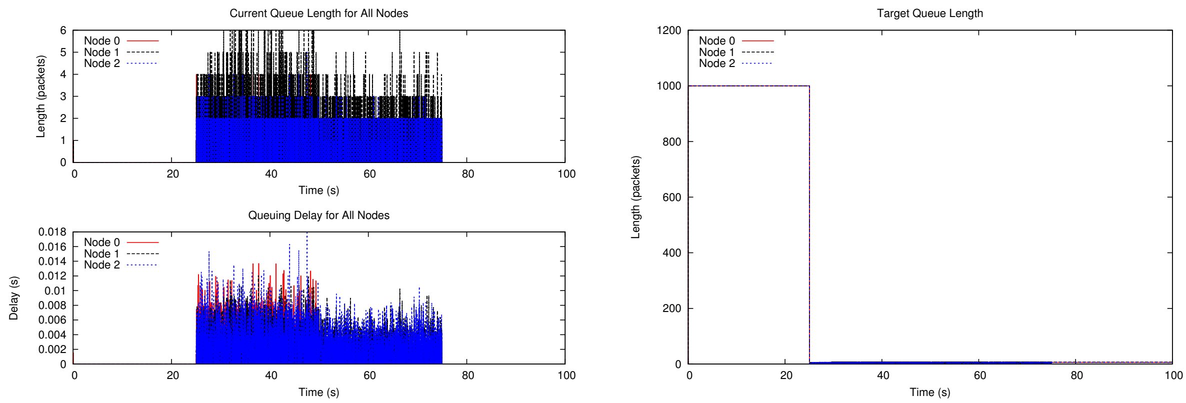


Figure 5.17: Scenario 1(c): Overall OWD Trend for ADTH-DF

Fig. 5.18 shows that ADTH has regulated queuing delay by shrinking the target queue threshold to a very small limit and constrained by the lower bound *MIN_ Q_ LIMIT*.



(a) Instantaneous Queue Length vs Queuing Delay (b) Target Queue Threshold

Figure 5.18: Scenario 1(c): Queuing Delay Regulation via a Movable Target Queue Threshold Estimated by ADTH

5.5.2 Scenario 2: Different traffic load

D_{Nr} is fixed at 0.1s in this scenario, thus $OWD_R = 0.2s$. ADTH has been validated with different traffic loads in this scenario to show that ADTH-DF is able to adapt to network changes caused by network load. R ranges from $1.0Mbps$ to $2.0Mbps$ with stepping of $0.25Mbps$.

Tables 5.8 - 5.9 show the system performance recorded for all sub-cases in the simulation. When the network is lightly loaded and not congested, the OWD recorded is minimal for both schemes. There is no packet loss when the network is not congested. When the traffic rate is increased, the overall packet loss ratio also increases for both FIFO-DF and ADTH-DF.

Table 5.8: Scenario 2: Overall System Performance for ADTH-DF

R	Collision	Full Q	Max OWD (s)		Packet Loss (%)	
($Mbps$)	Count	Drop	N0→N2	N2→N0	N0→N2	N2→N0
1.00	0	0	0.0117	0.0232	0	0
1.25	0	0	0.0139	0.0234	0	0
1.50	234	2501	0.1650	0.1505	8.66	10.10
1.75	404	4572	0.1773	0.1844	12.61	16.98
2.00	456	7374	0.1766	0.1775	15.03	27.28

Table 5.9: Scenario 2: Overall System Performance for FIFO-DF

R	Collision	Full Q	Max OWD (s)		Packet Loss (%)	
($Mbps$)	Count	Drop	N0→N2	N2→N0	N0→N2	N2→N0
1.00	0	0	0.0117	0.0232	0	0
1.25	0	0	0.0139	0.0234	0	0
1.50	316	1601	2.7405	2.7295	5.50	6.51
1.75	516	3393	6.2643	7.0489	8.89	13.16
2.00	631	6187	7.5983	7.4679	12.07	23.52

Deadline miss ratio for ADTH is 0% while for FIFO-DF is very high and intolerable when the traffic rate is higher (Fig. 5.19). Deadline miss ratio for FIFO-DF ranges from 36% - 57% for flow 0 and 55% - 94% for flow 1 when R is increased from 1.5$Mbps$ to 2.0$Mbps$.

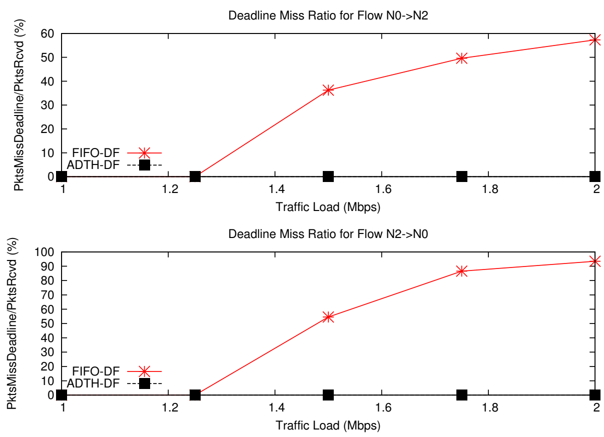
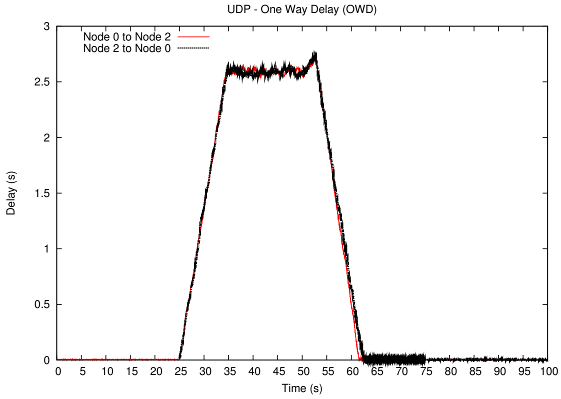
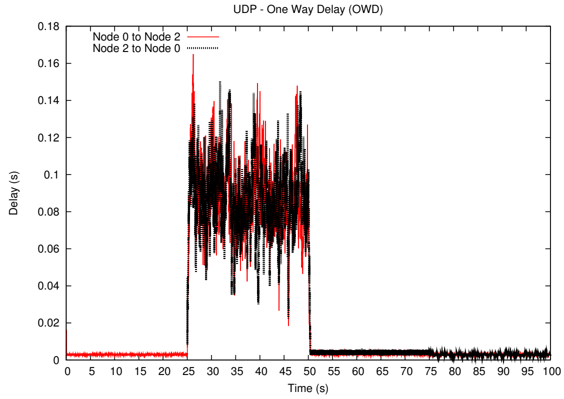


Figure 5.19: Scenario 2: Overall Deadline Miss Ratio (ADTH-DF versus FIFO-DF)

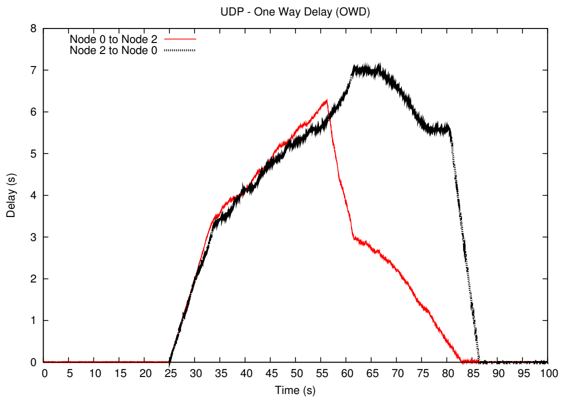
ADTH performs well under different network loads by bounding the measured OWD to be $\leqslant OWD_R$ for all cases despite the target nodal delay is only 100ms (Fig. 5.20). FIFO-DF fails to do so with the observation of OWD measured far beyond the maximum allowable value (Fig. 5.20). This has shown that an adaptive queue management scheme like ADTH is needed. When ADTH-DF is enabled, the node experiences higher packet loss. Nevertheless, the difference is only 3% - 4% higher compared to FIFO-DF.



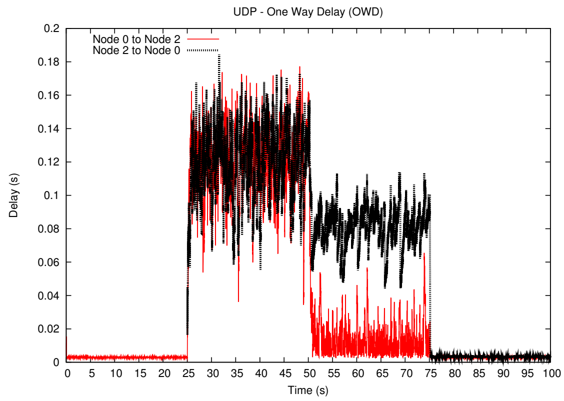
(a) FIFO-DF ($R = 1.5Mbps$)



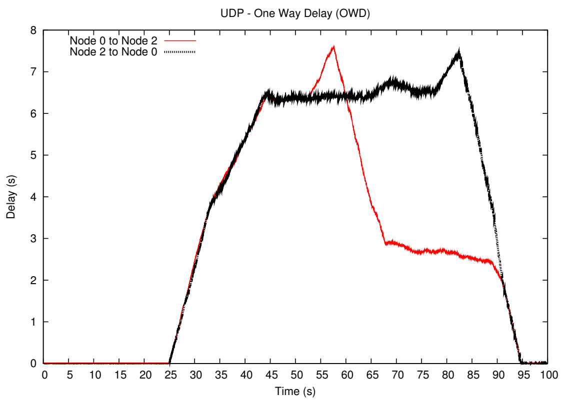
(b) ADTH-DF ($R = 1.5Mbps$)



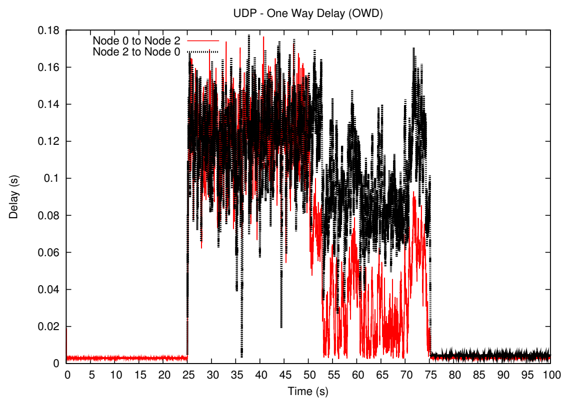
(c) FIFO-DF ($R = 1.75Mbps$)



(d) ADTH-DF ($R = 1.75Mbps$)



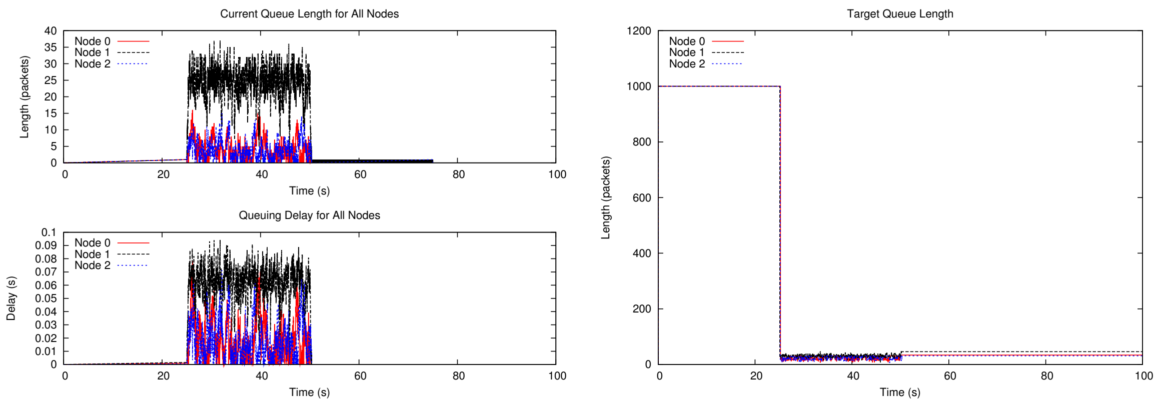
(e) FIFO-DF ($R = 2.0Mbps$)



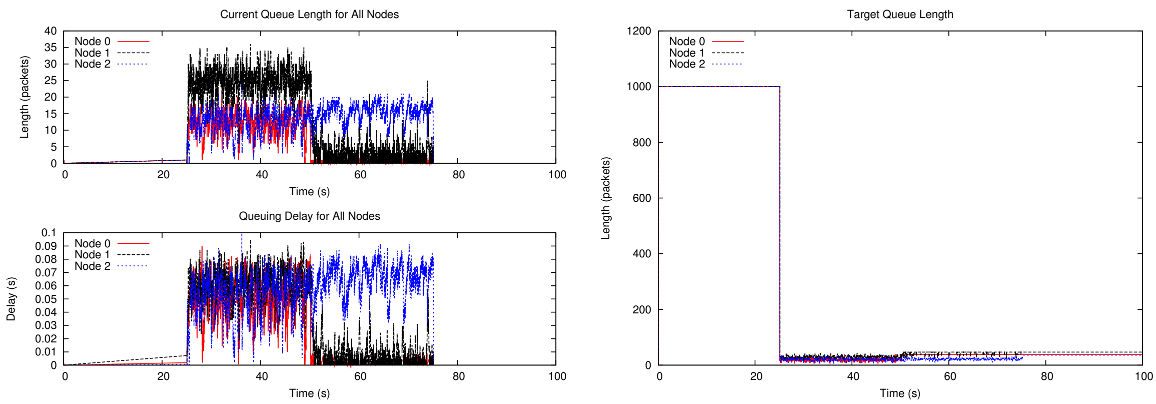
(f) ADTH-DF ($R = 2.0Mbps$)

Figure 5.20: Scenario 2: Overall OWD Trend (ADTH-DF versus FIFO-DF)

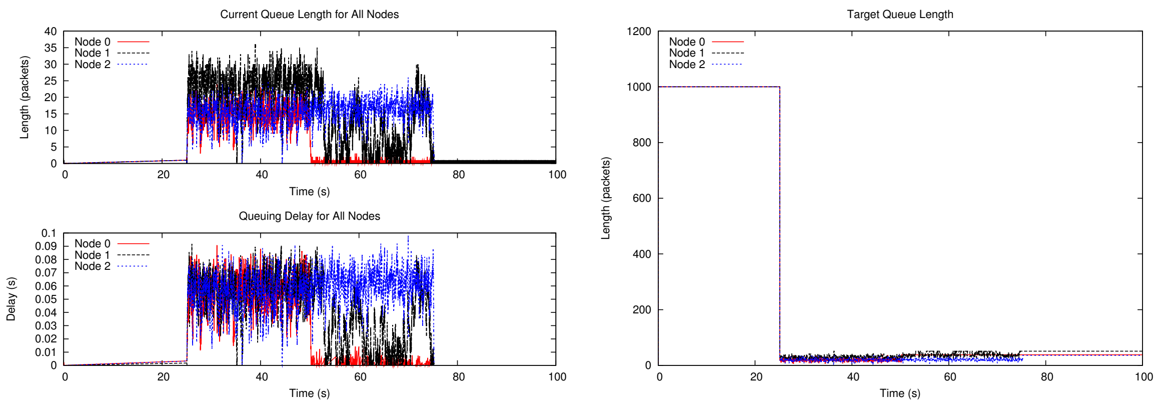
When the network starts congested ($R \geqslant 1.5Mbps$), the backlog in queue starts increasing and hence contributes to large queuing delay (Fig. 5.21). With ADTH-DF enabled, the backlog in queue is constrained by the target queue threshold. Consequently, nodal delay and OWD for the packets are bounded.



(a) $R = 1.5Mbps$



(b) $R = 1.75Mbps$

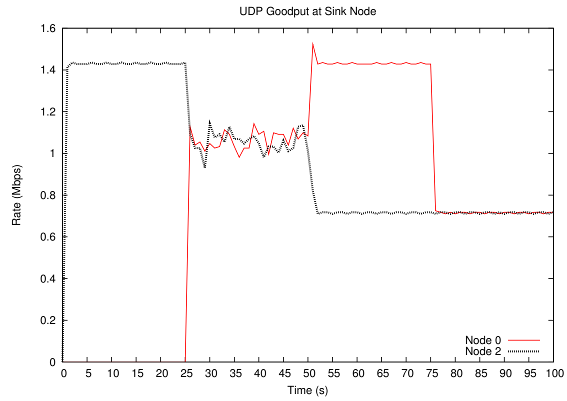


(c) $R = 2.0Mbps$

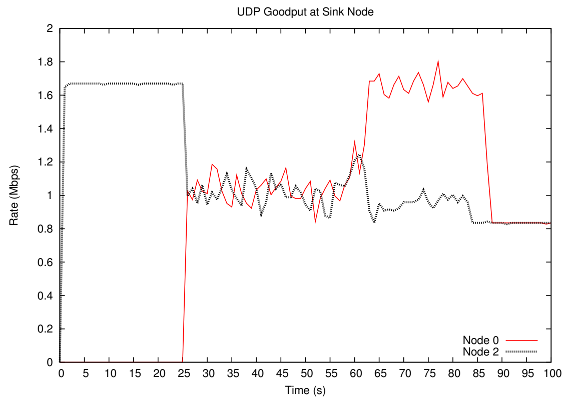
Figure 5.21: Scenario 2: Queuing Delay Regulation via Movable Target Queue Thresholds Estimated by ADTH

Fig. 5.22 shows the UDP goodput of ADTH-DF and FIFO-DF for $R \geqslant 1.5Mbps$. Higher UDP goodput is observed for FIFO-DF when $R = 1.5Mbps$. A spike in UDP goodput is observed from $50s$ - $65s$, packets buffered are drained from the full IFQ during this period. However, this does not mean that FIFO-DF performs better as most of the packets reach the destination after severe congestion are recorded with huge OWD.

(a) FIFO-DF ($R = 1.5Mbps$)

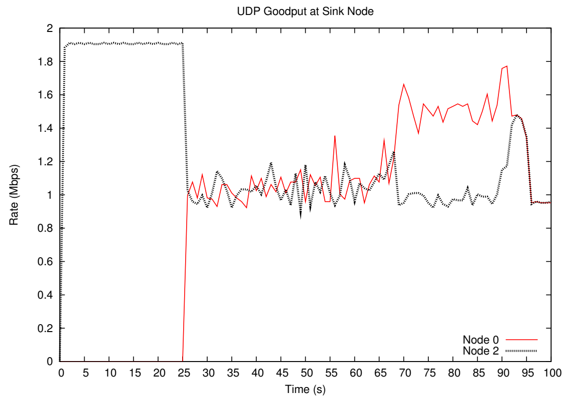


(b) ADTH-DF ($R = 1.5Mbps$)

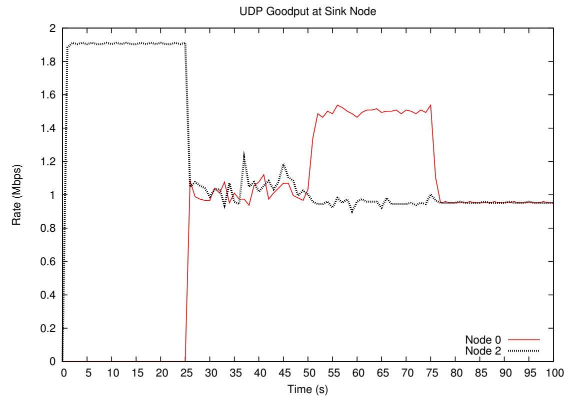


(c) FIFO-DF ($R = 1.75Mbps$)

(d) ADTH-DF ($R = 1.75Mbps$)



(e) FIFO-DF ($R = 2.0Mbps$)

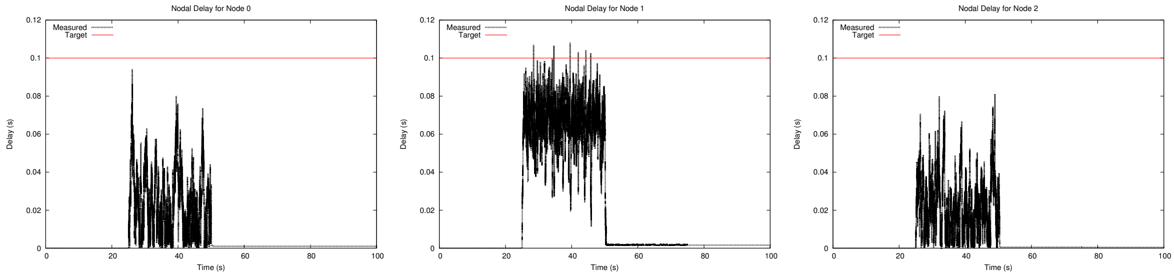


(f) ADTH-DF ($R = 2.0Mbps$)

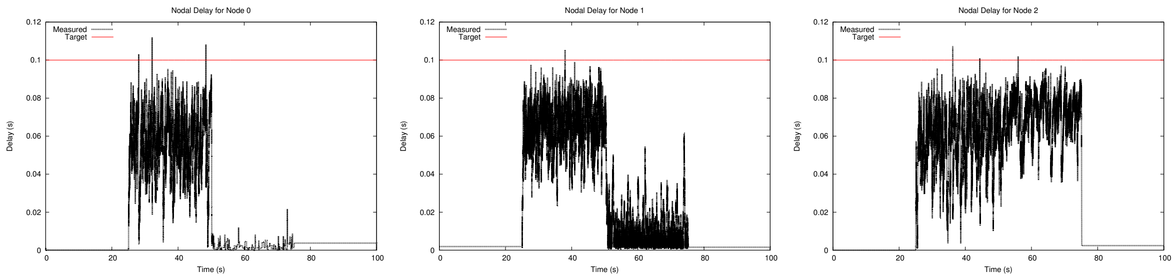
Figure 5.22: Scenario 2: UDP Goodput for ADTH-DF versus FIFO-DF

IFQ size for the ADTH-DF scheme is controlled at a lower limit; therefore it takes lesser time to drain the IFQ. Packets are discarded earlier and thus this enables following new packets to be forwarded to the destinations within the delay requirement. This mechanism contributes to network congestion mitigation implicitly as the UDP goodput for ADTH-DF catches up faster than FIFO-DF when the traffic load is reduced.

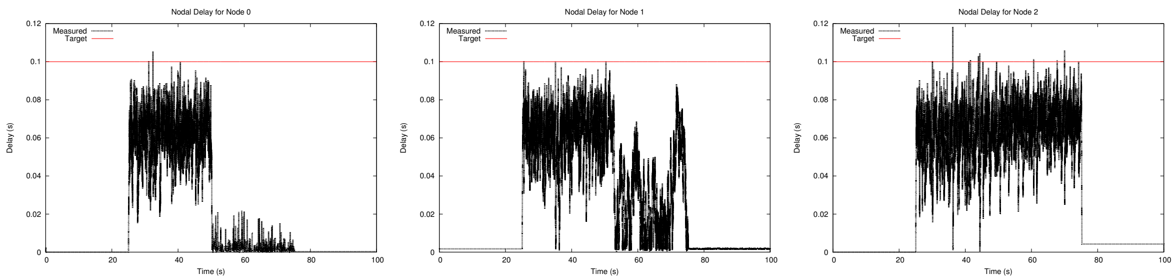
Fig. 5.23 shows that nodal delays are bounded at all wireless nodes with ADTH-DF enabled regardless of traffic load in the network. The miss target nodal delay ratio is negligible.



(a) ADTH-DF ($R = 1.5Mbps$)



(b) ADTH-DF ($R = 1.75Mbps$)



(c) ADTH-DF ($R = 2.0Mbps$)

Figure 5.23: Scenario 2: Nodal Delay for Wireless Nodes with ADTH Enabled

5.5.3 Scenario 3: Different packet size

This scenario is to validate the ability of ADTH-DF to react to network dynamics caused by packet size. The packet size (PS) is varied from $300B$ to $1500B$ with stepping of $300B$. All nodes are preset with the same D_{Nr} that is $1.5s$. Hence, OWD_R of CBR flows for both directions are $3.0s$. For this scenario, $R = 1.6Mbps$ and $t_s = \frac{1}{2}D_{Nr}$ are used.

The network is more congested when the data are transported with smaller sized packets under the same traffic load. Therefore, the packet loss ratio is expected to be high when the packet size is small. Simulation results in Tables 5.10 and 5.11 show that packet loss ratio is only increased by 2% - 3% in certain cases when ADTH-DF is enabled. The performance drop is negligible in this case when compared to the performance gain in other aspects that are discussed later.

Table 5.10: Scenario 3: Overall System Performance for ADTH-DF

PS (B)	Collision Count	Full Q Drop	Max OWD (s)		Packet Loss (%)	
			N0→N2	N2→N0	N0→N2	N2→N0
300	1471	60211	2.7425	2.6293	52.29	66.06
600	859	11631	2.8859	3.0637	16.53	32.27
900	587	3752	2.7786	2.8266	9.71	14.16
1200	236	1457	1.4171	1.4165	5.66	6.73
1500	182	0	0.8408	0.8244	0	0

Table 5.11: Scenario 3: Overall System Performance for FIFO-DF

PS (B)	Collision Count	Full Q Drop	Max OWD (s)		Packet Loss (%)	
			N0→N2	N2→N0	N0→N2	N2→N0
300	1431	58960	4.3094	4.2941	52.28	66.13
600	923	11149	6.0070	6.2936	15.84	30.94
900	604	3062	4.9467	5.8189	8.55	10.81
1200	296	890	2.9657	2.9680	3.40	4.19
1500	182	0	0.8408	0.8244	0	0

The maximum OWD recorded are mostly $\leqslant OWD_R$ when ADTH-DF is enabled except for the case with packet size $600B$ (Table 5.10). For this case, the OWD measured exceeds OWD_R slightly with deadline miss ratio of 0.58%. However, using a smaller value of t_s can eliminate this. In this simulation, t_s is set to $0.75s$ that is considered quite high. Therefore, it can be lowered in order to enable ADTH-DF to react to network changes more aggressively.

Without ADTH-DF scheme, the OWD recorded is almost 2× compared to ADTH-DF scheme (Table 5.11). In addition, the deadline miss ratio for FIFO-DF is very high for most of the cases (Fig. 5.24). If the D_{Nr} is smaller, then the deadline miss ratio will be even higher for FIFO-DF.

OWD is bounded to $\leqslant OWD_R$ when ADTH-DF is enabled through bounded nodal delay (Fig. 5.25). The miss target nodal delay ratio is negligible. Fig. 5.26 shows an interesting view of OWD trend for both ADTH-DF and FIFO-DF with different packet sizes. The OWD trends differ due to the network dynamics introduced by different packet sizes. The congestion level and the queue fullness are impacted. This causes variation in MAC layer delay and queuing delay experienced by packets. OWD is high when the network is severely congested with network load comprises of small sized packets. However, enabling ADTH-DF can mitigate these impacts.

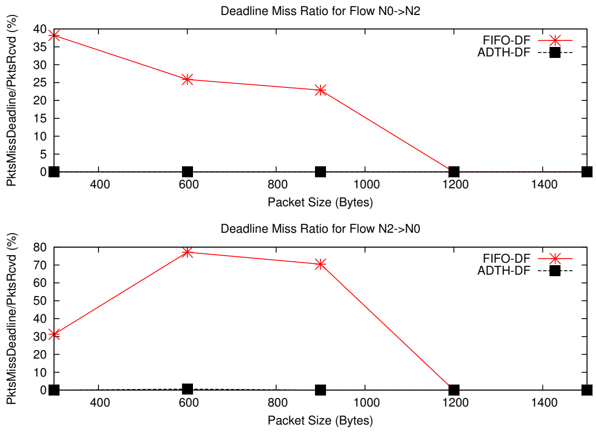
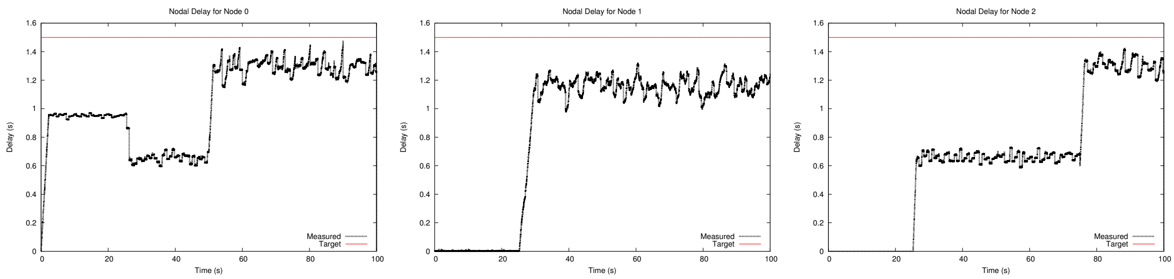
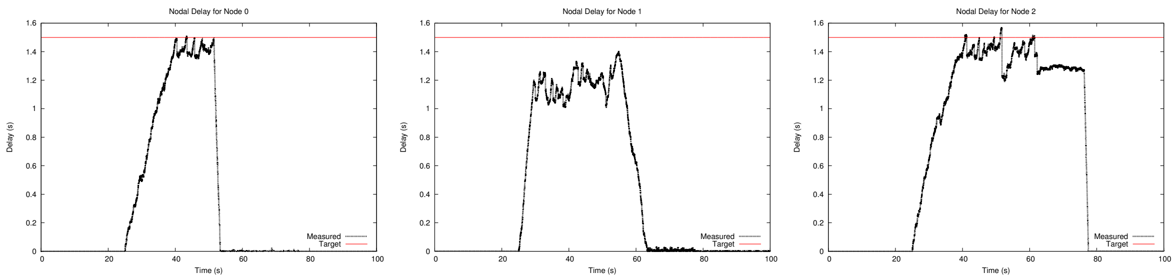


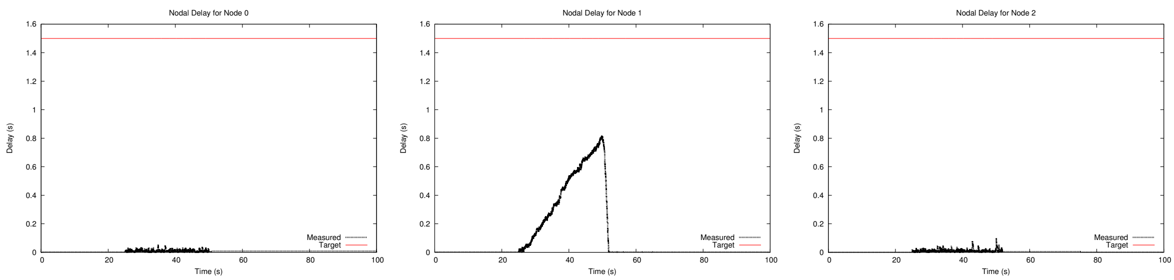
Figure 5.24: Scenario 3: Overall Deadline Miss Ratio (ADTH-DF versus FIFO-DF)



(a) ADTH-DF (Packet Size = $300B$)

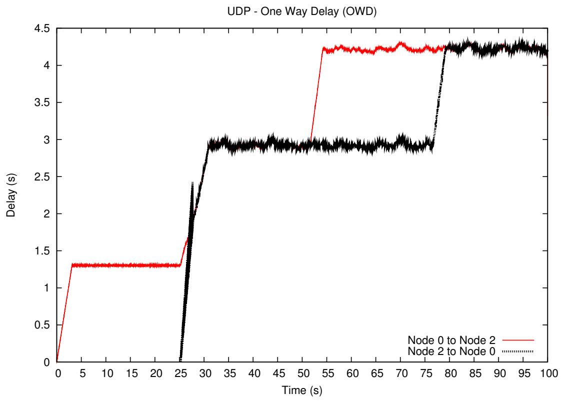


(b) ADTH-DF (Packet Size = $900B$)

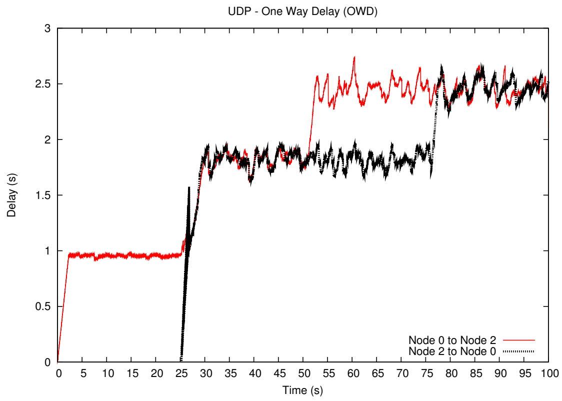


(c) ADTH-DF (Packet Size = $1500B$)

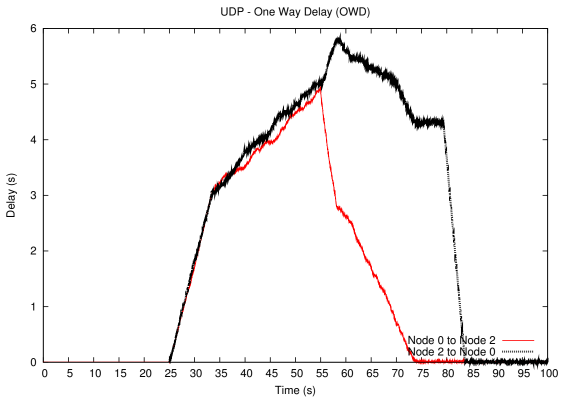
Figure 5.25: Scenario 3: Nodal Delay for Wireless Nodes with ADTH Enabled



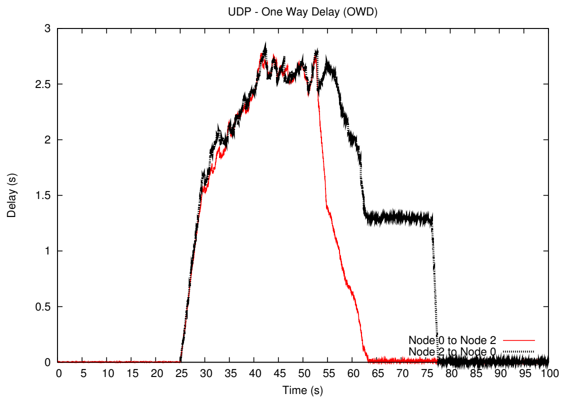
(a) FIFO-DF (Packet Size = 300B)



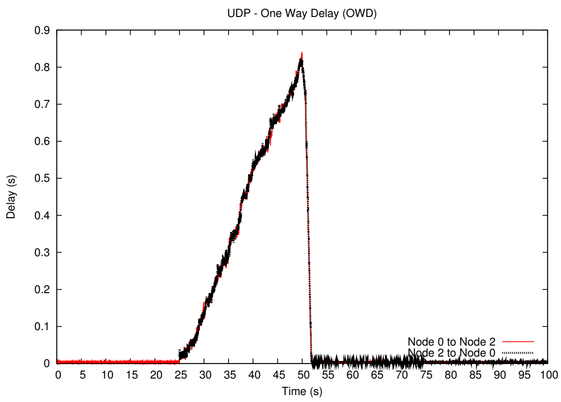
(b) ADTH-DF (Packet Size = 300B)



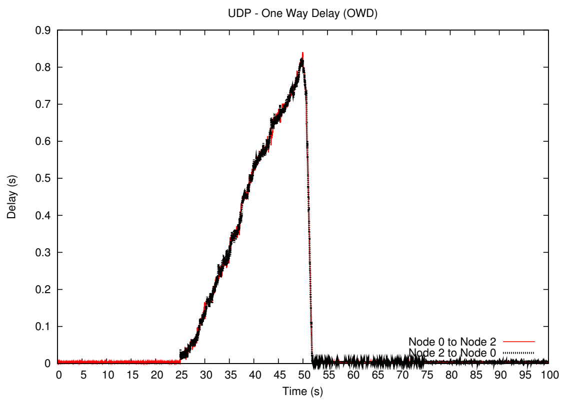
(c) FIFO-DF (Packet Size = 900B)



(d) ADTH-DF (Packet Size = 900B)



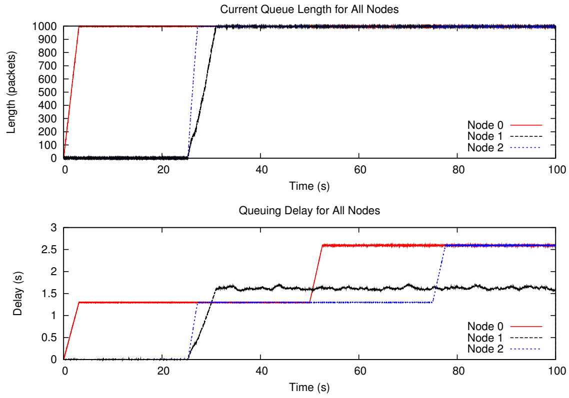
(e) FIFO-DF (Packet Size = 1500B)



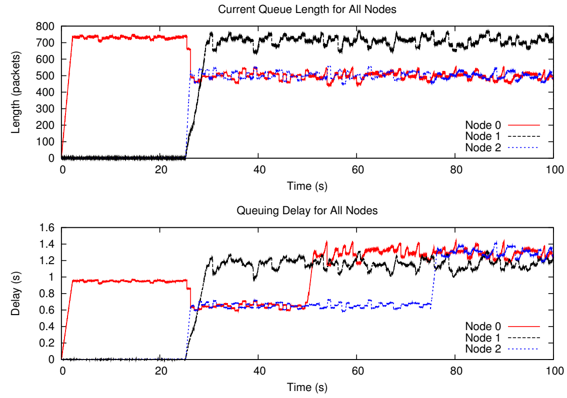
(f) ADTH-DF (Packet Size = 1500B)

Figure 5.26: Scenario 3: Overall OWD Trend (ADTH-DF versus FIFO-DF)

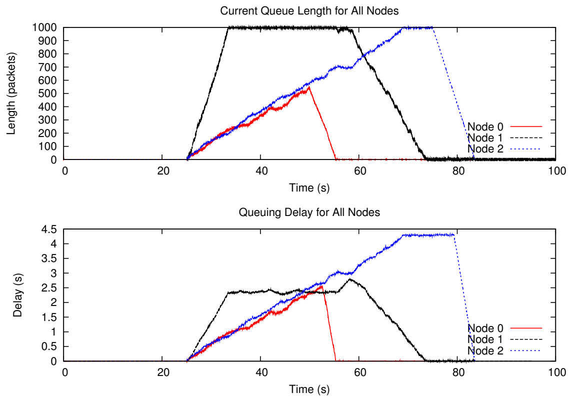
Fig. 5.27 presents the trend of queue backlog and the trend of queuing delay for both schemes. Without the ADTH-DF scheme, backlog in queue can grow to the maximum queue limit. Hence, queuing delay and OWD can be high. When ADTH-DF is enabled, queuing delay is constrained by allowing different level of queue backlog in each IFQ. The node with higher throughput (Node 1) is estimated with a higher queue threshold and yet able to constrain queuing delay around the same target as others.



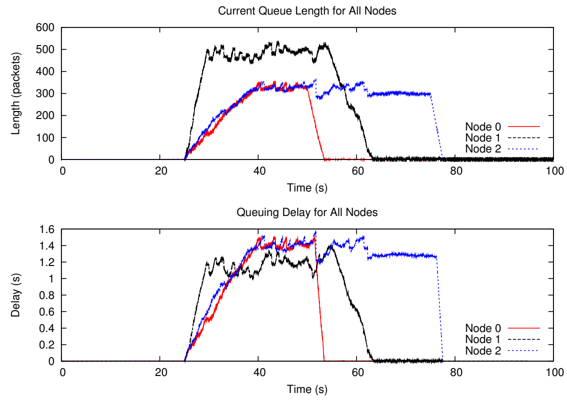
(a) FIFO-DF (Packet Size = $300B$)



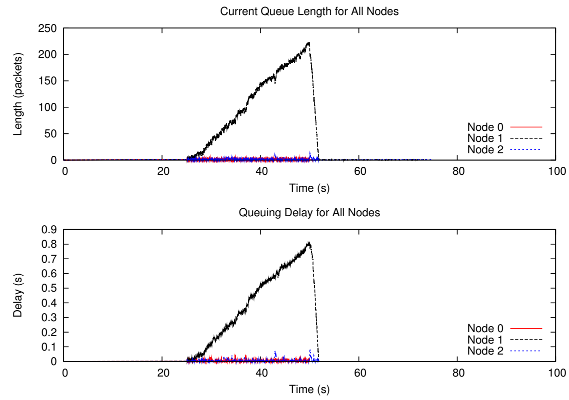
(b) ADTH-DF (Packet Size = $300B$)



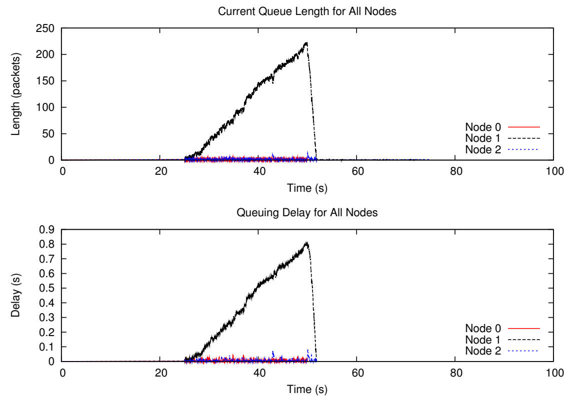
(c) FIFO-DF (Packet Size = $900B$)



(d) ADTH-DF (Packet Size = $900B$)



(e) FIFO-DF (Packet Size = $1500B$)

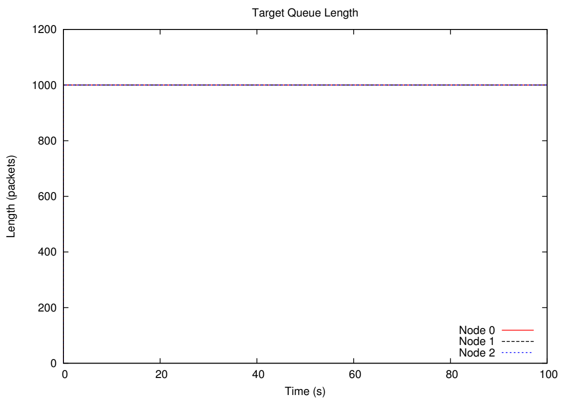


(f) ADTH-DF (Packet Size = $1500B$)

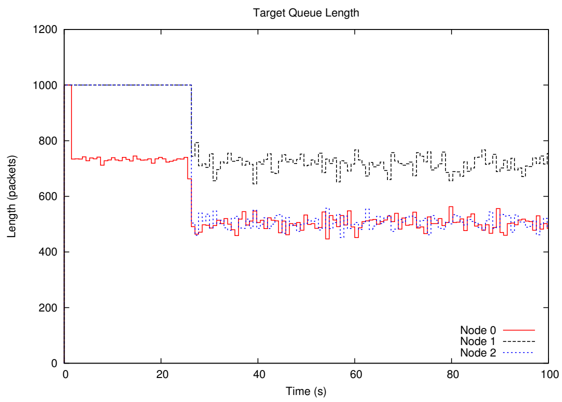
Figure 5.27: Scenario 3: Instantaneous Queue Length and Queuing Delay for ADTH-DF versus FIFO-DF

Based on the simulation results above, ADTH-DF is able to bound nodal delay regardless of the packet size and congestion level. This is due to ADTH-DF adapts to the changes in the network and changes the target queue threshold dynamically (Fig. 5.28). The self-adaptation of queue size does not have big impact on the

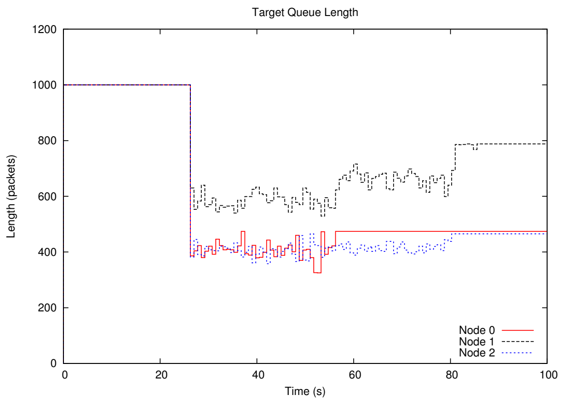
UDP goodput (Fig. 5.29). The UDP goodput for the case of PS $= 1500B$ is almost identical for FIFO-DF and ADTH-DF. This is because the network is not severely congested given a lenient D_{Nr}, the backlog in queue does not hit the target queue threshold estimated. So ADTH-DF behaves similarly to FIFO-DF in this particular case.



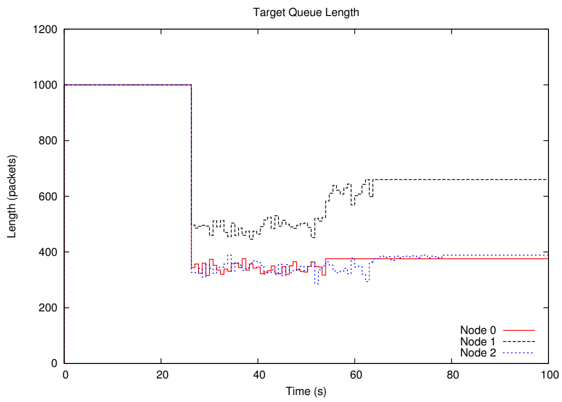
(a) FIFO-DF (Packet Size $= 300B$ - $1500B$)



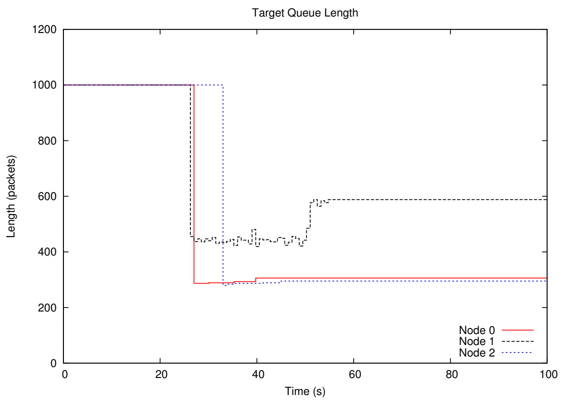
(b) ADTH-DF (Packet Size $= 300B$)



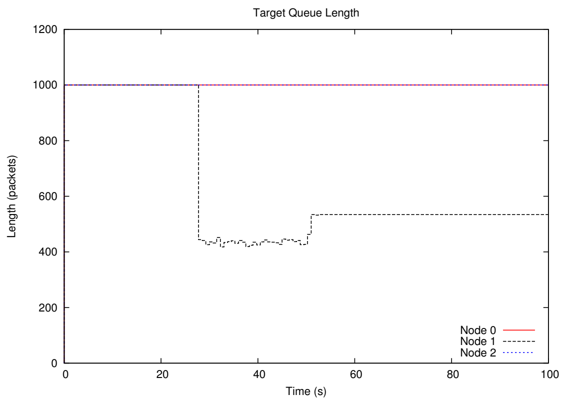
(c) FIFO-DF w ADTH-DF (Packet Size $= 600B$)



(d) FIFO-DF w ADTH-DF (Packet Size $= 900B$)

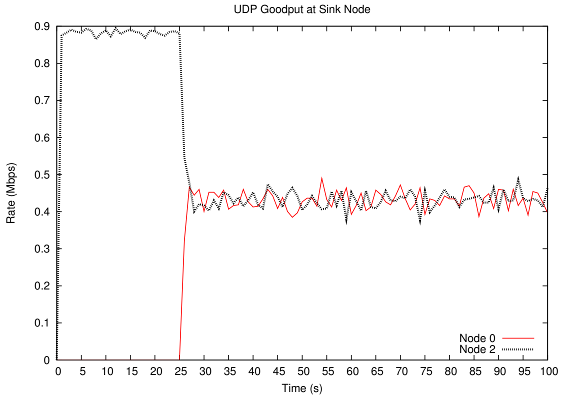


(e) FIFO-DF w ADTH-DF (Packet Size $= 1200B$)

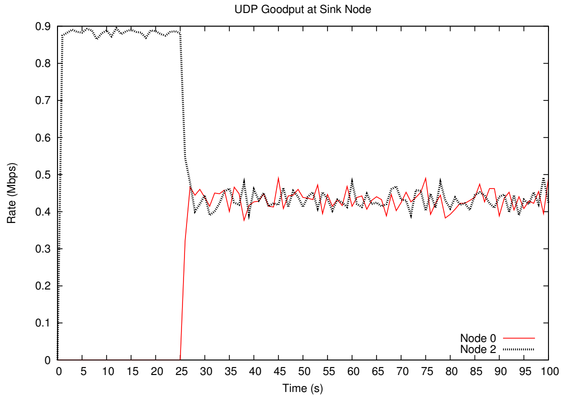


(f) FIFO-DF w ADTH-DF (Packet Size $= 1500B$)

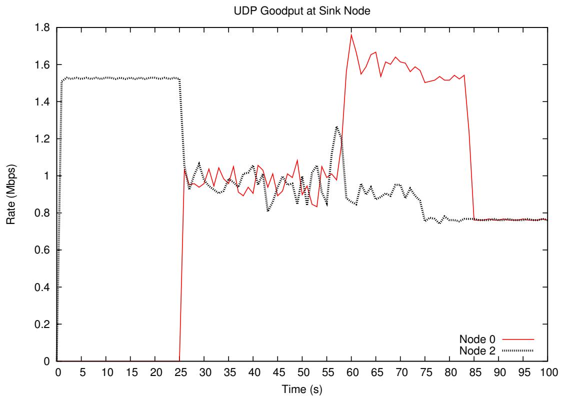
Figure 5.28: Scenario 3: ADTH Target Queue Threshold



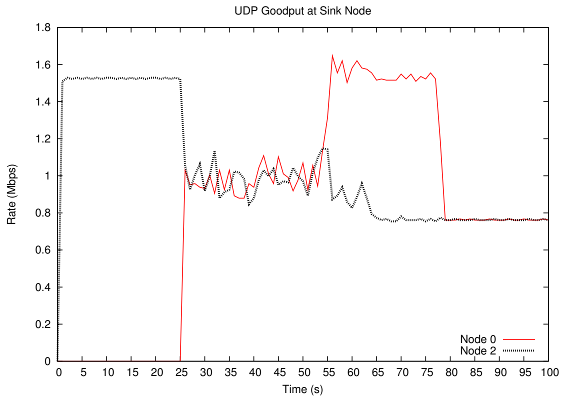
(a) FIFO-DF (Packet Size = $300B$)



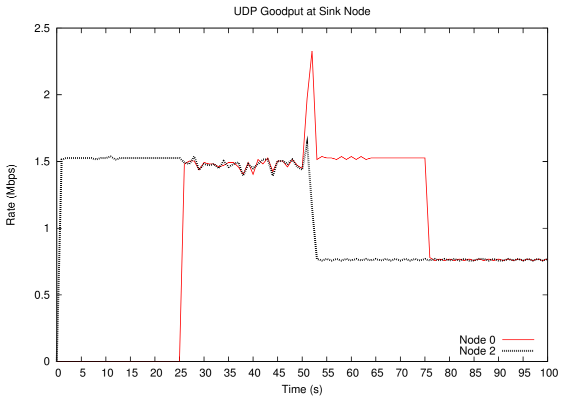
(b) ADTH-DF (Packet Size = $300B$)



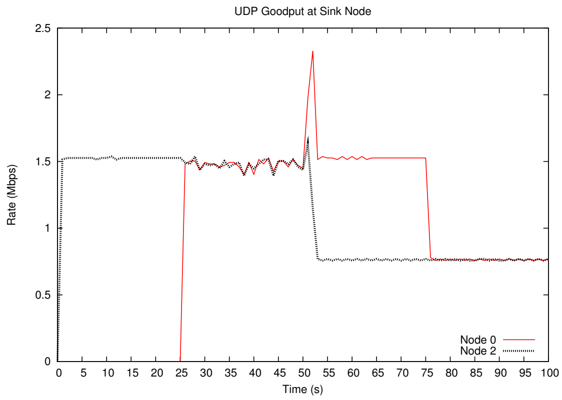
(c) FIFO-DF (Packet Size = $900B$)



(d) ADTH-DF (Packet Size = $900B$)



(e) FIFO-DF (Packet Size = $1500B$)



(f) ADTH-DF (Packet Size = $1500B$)

Figure 5.29: Scenario 3: UDP Goodput for ADTH-DF versus FIFO-DF

5.5.4 Scenario 4: Multiple CBR flows with mix packet size

This simulation is to show that ADTH-DF is a generic and scalable queue management scheme that does not require per-flow state or statistics. In this simulation scenario, multiple CBR traffic flows carried over the UDP transport agent have been simulated. Packet size for the CBR flows are not uniform and the rates also vary. In total, five CBR flows with each flow $0.3Mbps$ but with different packet sizes (2 flows @ $200B$, 2 flows @ $600B$ and 1 flow @ $1500B$) are sent from the source Node 0 to Node 2. While for the source Node 2, eight CBR flows with different packet sizes (2 flows @ $400B$, 2 flows @ $600B$, 2 flows @ $800B$ and 2 flows @ $1000B$) and $0.2Mbps$ each are sent towards Node 0. Delay requirements are $D_{Nr} = 0.5s$ and $OWD_R = 1s$. The simulation has been carried out for the ADTH-DF scheme with the maximum queue limit of 1000 packets; while FIFO-DF with 50 packets and 1000 packets as the maximum queue limit.

Table 5.12 shows the performance results of FIFO-DF queue with queue limit of 50 packets (FIFO-DF-50) and 1000 packets (FIFO-DF-1000). The performance of FIFO-DF-50 is better than FIFO-DF-1000 in terms of data yield and maximum OWD. Packet loss ratio is lower when the queue limit is 1000 packets, but the data yield becomes extremely low. Most of the packets reach the destination after the deadline and are rendered useless.

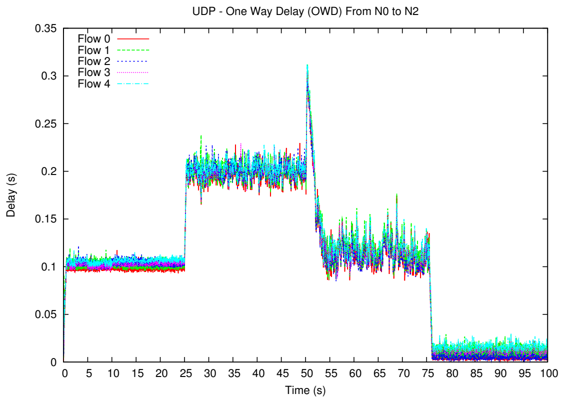
Table 5.12: Scenario 4: Performance of FIFO-DF with Different Queue Limit

FIFO-DF	Queue Limit = 50		Queue Limit = 1000	
	N0→N2	N2→N0	N0→N2	N2→N0
Packet loss ratio (%)	31.59	33.80	30.47	17.03
Packet delivery ratio (%)	68.41	66.20	69.53	82.97
Deadline miss ratio (%)	0	0	89.62	0.13
Data yield (%)	68.41	66.20	7.22	1.43
Max OWD (s)	0.3128	0.2950	5.7141	5.9209

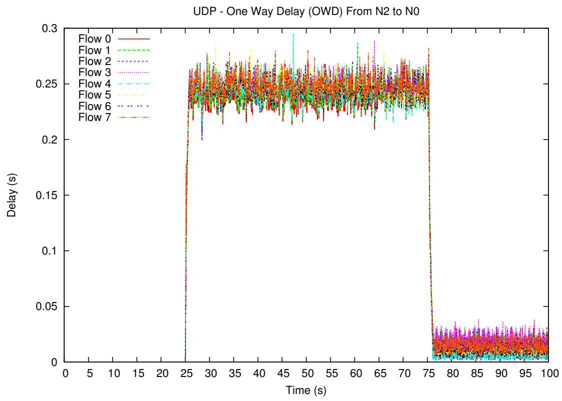
Table 5.13: Scenario 4: Performance of ADTH-DF with Different t_s

ADTH-DF	$t_s = 0.2s$		$t_s = 0.25s$	
	N0→N2	N2→N0	N0→N2	N2→N0
Packet loss ratio (%)	30.13	33.04	30.24	33.24
Packet delivery ratio (%)	69.87	66.96	69.76	66.76
Deadline miss ratio (%)	0	0	≈0	0.13
Data yield (%)	69.87	66.96	69.76	66.67
Max OWD (s)	0.8991	0.9198	1.0088	1.3981

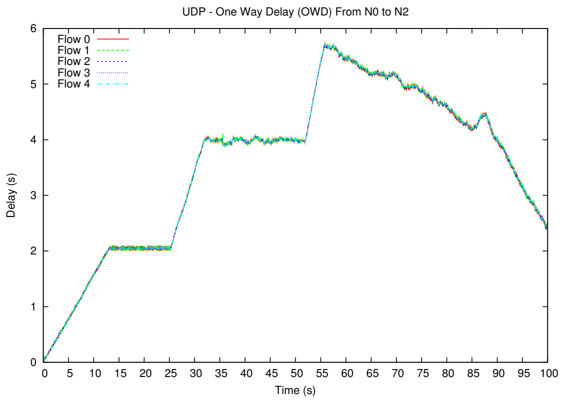
ADTH-DF performs much better than FIFO-DF especially when the queue limit of FIFO-DF is 1000 packets (Table 5.12 and Table 5.13). Even multiple CBR flows traverse across the network; the ADTH-DF scheme is still able to bound nodal delay to the required value with appropriate sampling frequency. Thus, OWD is bounded. When the sampling time is $0.25s$, a few packets reach the destination later than their deadline. The deadline miss ratio is only 0.13% for CBR flow from Node 2 to Node 0. It is merely 0% for CBR flow from Node 0 to Node 2 as only 1 out of 25507 packets received exceeds OWD_R. The value is insignificant as compared to the delivery ratio. If the sampling time is set to slightly lower than $\frac{1}{2}D_{Nr}$, then the deadline miss ratio becomes 0%. ADTH-DF responds faster to the network changes with a higher sampling frequency.



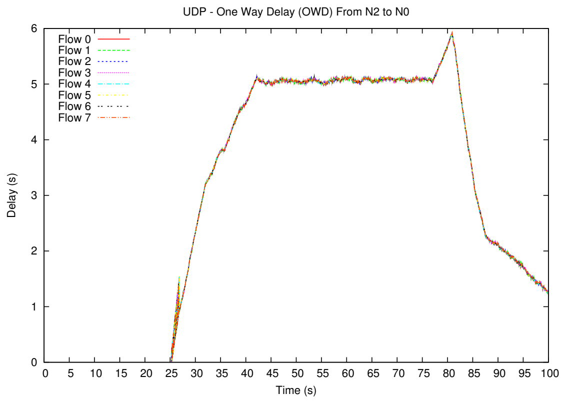
(a) FIFO-DF (Queue Limit = 50)



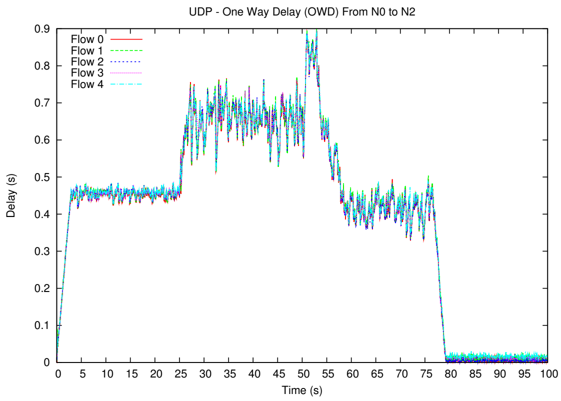
(b) FIFO-DF (Queue Limit = 50)



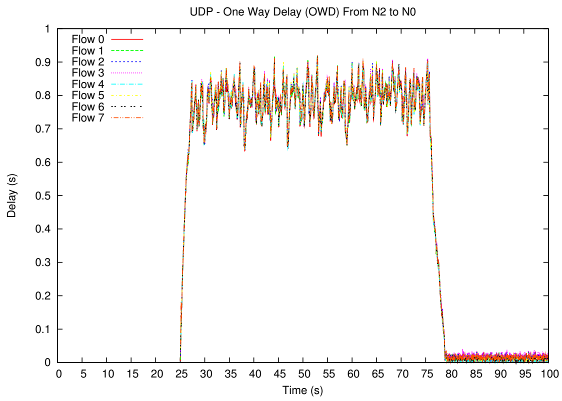
(c) FIFO-DF (Queue Limit = 1000)



(d) FIFO-DF (Queue Limit = 1000)



(e) ADTH-DF ($t_s = 0.2s$)

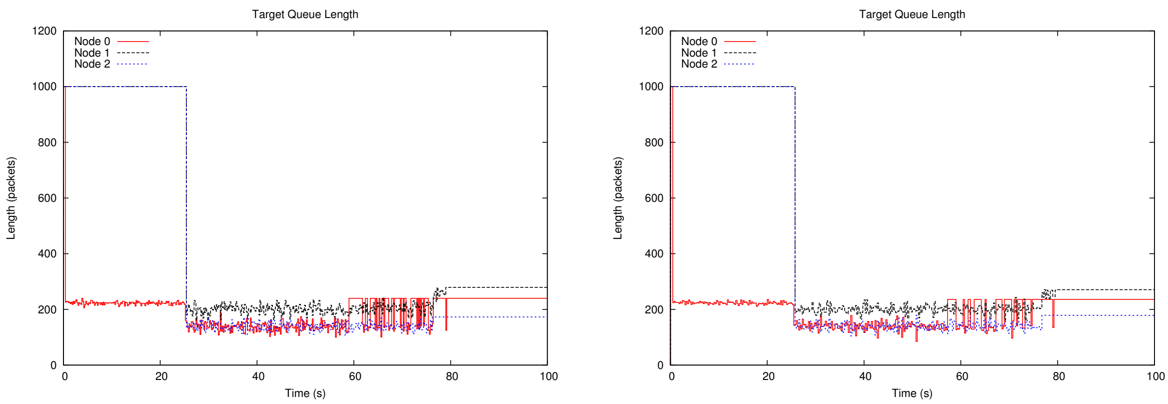


(f) ADTH-DF ($t_s = 0.2s$)

Figure 5.30: Scenario 4: Overall OWD trend for ADTH-DF versus FIFO-DF

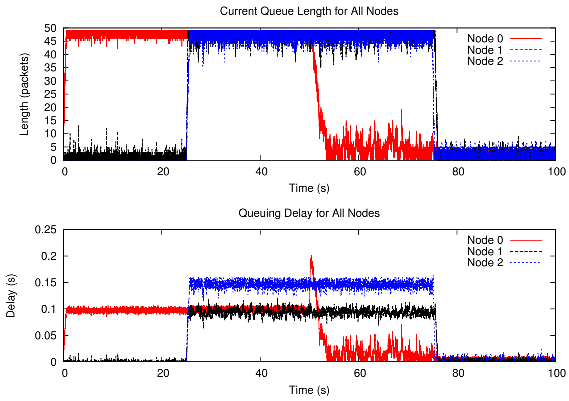
The maximum OWD recorded for ADTH-DF is higher than FIFO-DF-50, but it is within the required bound. It is coincident that the queue limit 50 works well for this simulation scenario. If there are any changes in traffic load, number of neighbour nodes, required delay, etc., then this queue limit might not be optimum to give the best system performance. If the queue limit is set too low, this may cause unnecessary packet loss and affect the data yield. In contrast, if the queue limit is set to too high then packets might arrive at the destination later than their deadlines due to huge queuing delay. Therefore, adaptive target queue threshold is needed. Overall, ADTH-DF still performs better than FIFO-DF-50 as its packet deliver ratio and data yield are slightly higher than FIFO-DF-50. In addition, more buffer spaces should be allowed to buffer packets when D_{Nr} is higher. Then in this case, queue limit of 50 packets is not optimum anymore. Fig. 5.30 shows the overall OWD trend throughout the simulation for scenario discussed above.

Fig. 5.31 shows the target queue thresholds for all nodes are adjusted dynamically to bound nodal delay through queuing delay constraint (Fig. 5.32). With the requirement of $D_{Nr} = 0.5s$, the target queue thresholds roughly range from 125 to 225 packets. This shows that why a constant queue threshold is not optimum for wireless ad hoc network. This also explains why ADTH-DF has higher data yield than FIFO-DF-50 as more packets can be buffered.

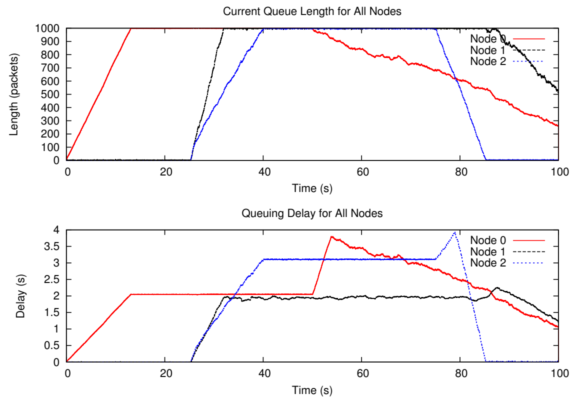


(a) ADTH-DF ($t_s = 0.2s$) (b) ADTH-DF ($t_s = 0.25s$)

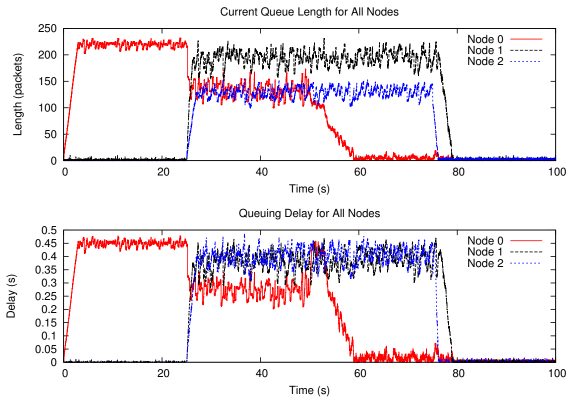
Figure 5.31: Scenario 4: Target Queue Threshold of ADTH-DF



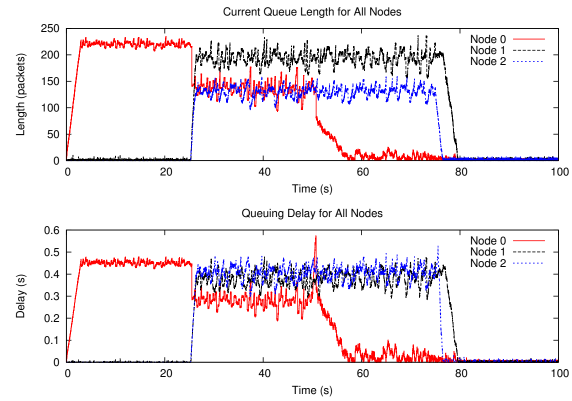
(a) FIFO-DF (Queue Limit = 50)



(b) FIFO-DF (Queue Limit = 1000)



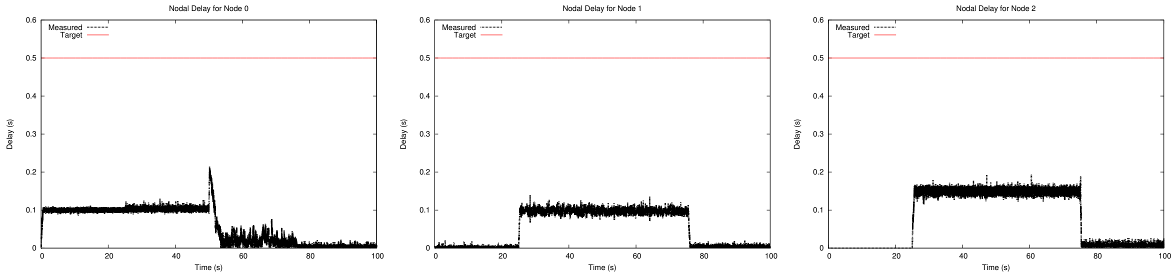
(c) ADTH-DF ($t_s = 0.2s$)



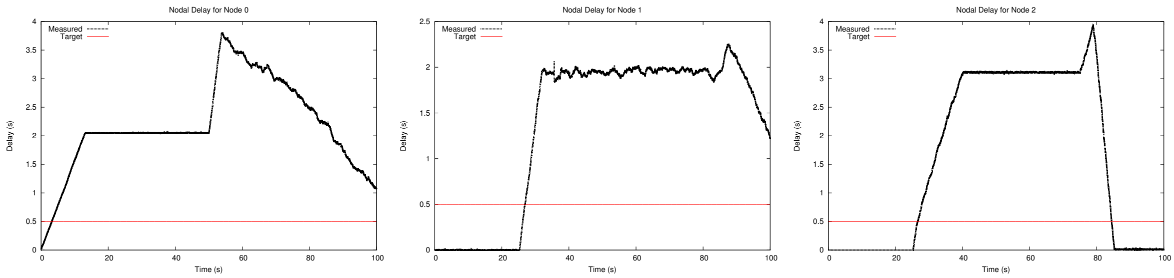
(d) ADTH-DF ($t_s = 0.25s$)

Figure 5.32: Scenario 4: Instantaneous Queue Length versus Queuing Delay (ADTH-DF versus FIFO-DF)

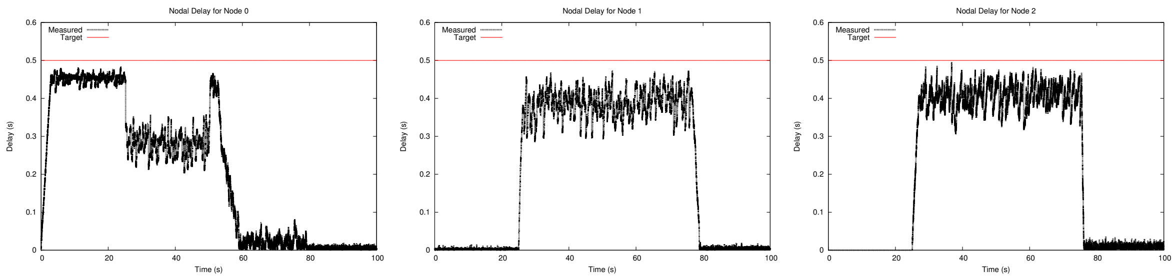
Fig. 5.33 shows nodal delays are bounded as a result of adaptive target queue threshold when ADTH-DF is enabled. The miss target nodal delay ratio is either 0% or negligible.



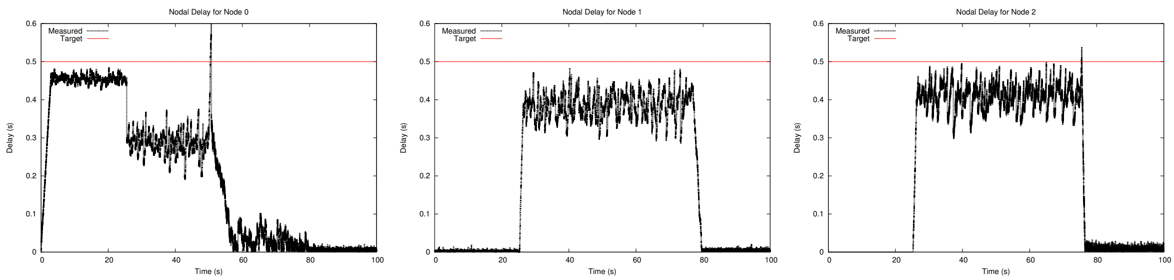
(a) FIFO-DF (Queue Limit = 50)



(b) FIFO-DF (Queue Limit = 1000)



(c) ADTH-DF ($t_s = 0.2s$)



(d) ADTH-DF ($t_s = 0.25s$)

Figure 5.33: Scenario 4: Nodal delay for FIFO-DF versus ADTH-DF

5.5.5 Scenario 5: Different number of hops

ADTH is a standalone queue management scheme that bound nodal delay based on the nodal information gathered. A global view of the network is not needed. In this scenario, ADTH has been simulated with different number of hops (H) ranging from 3 to 15 hops with stepping of 3 hops to show that ADTH is scalable. R is fixed at $0.8Mbps$ with $OWD_R = 1.5s$. Thus, $D_{Nr} = OWD_R/H$.

ADTH-DF with queue limit of 1000 packets has been simulated and compared to FIFO-DF with queue limit of 50 (FIFO-DF-50) and 1000 packets (FIFO-DF-1000). Generally, the packet loss ratio increases when packets need to traverse more hops to reach the destination (Fig. 5.34). With the number of hops increases, more nodes involve in forwarding packets and bandwidth available for each node becomes lesser. There are more nodes competing for the channel access and the nodes interfering each other. Consequently, the congestion level is exacerbated. MAC delay can be high due to retransmission at MAC layer and medium access contention. Overall, ADTH-DF has the highest loss ratio among the 3 schemes simulated. However, the difference of packet loss ratio between ADTH-DF and FIFO-DF-1000 is not significant except for the cases of 12 hops and 15 hops. However, this does not imply that ADTH-DF has worse performance. If the OWD is observed, then obviously the network performance is improved when ADTH-DF is adopted.

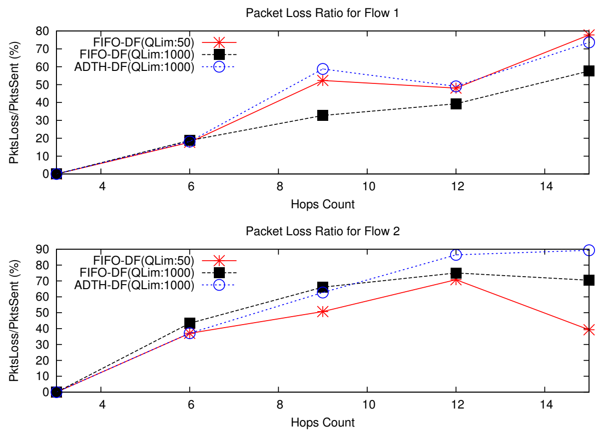


Figure 5.34: Scenario 5: Overall Packet Loss Ratio (ADTH-DF versus FIFO-DF)

Fig. 5.35 shows that the OWD recorded for ADTH-DF is the lowest. FIFO-DF-1000 gives the worst OWD performance. The maximum OWD recorded for this case can be $>50s$ when the source is 15 hops away from the destination. The network latency recorded is unacceptable for delay-sensitive traffic. One may argue that by setting the queue limit to a lower limit then the OWD can be controlled. However, finding an optimum queue limit is non-trivial. FIFO-DF-50 has similar maximum OWD when compared to ADTH-DF at smaller hops. After 9 hops, the OWD of FIFO-DF-50 is higher than ADTH-DF, the OWD recorded for FIFO-DF-50 is $> 2.5s$. The network load, contention level and interference level become dynamic with the changes of number of nodes in the network. The optimum queue limit might vary with the requirement of D_{Nr}, OWD_R and the number of hops between a source and a destination. Therefore, the queue limit of 50 packets is no longer optimum in this case.

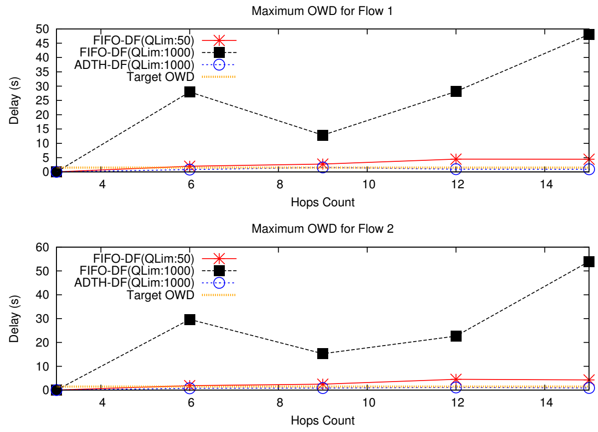


Figure 5.35: Scenario 5: Overall Maximum OWD (ADTH-DF versus FIFO-DF)

Fig. 5.36 shows that the deadline miss ratio for ADTH-DF scheme is 0%, all packets get delivered to the destination within the required OWD bound. In contrast, deadline miss ratio for FIFO-DF is very high and increases drastically when the number of hops between the source and the destination is increased.

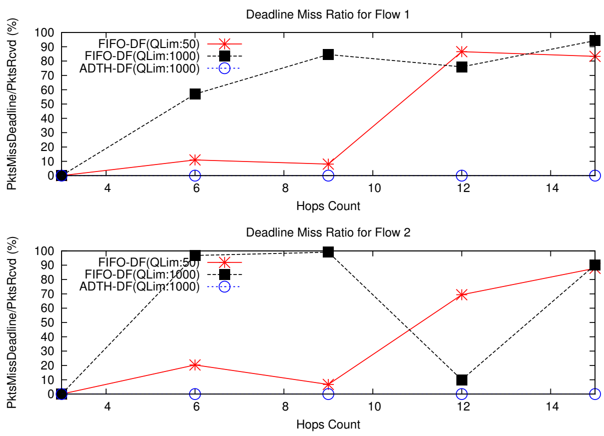


Figure 5.36: Scenario 5: Overall Deadline Miss Ratio (ADTH-DF versus FIFO-DF)

The data yield is very high when there is no congestion in the network. This is because packets are delivered to the destination on time and can be used by the application. Those packets reach the destination beyond the required OWD will be discarded and this results in lower data yield. The packet loss ratio of ADTH-DF is higher but the data yield for ADTH-DF is the best even the data yield decreases with the incremental of number of hops (Fig. 5.37).

There are very few backlogs in the queues when the destination is just 3 hops away from the source as the network is not congested. However, the situation changes when more nodes involve in the simulation and the destination is distant away from the source. Backlogs start to build up at FIFO-DF queues. The farther the destination node from the source node, the backlogs observed is higher even the

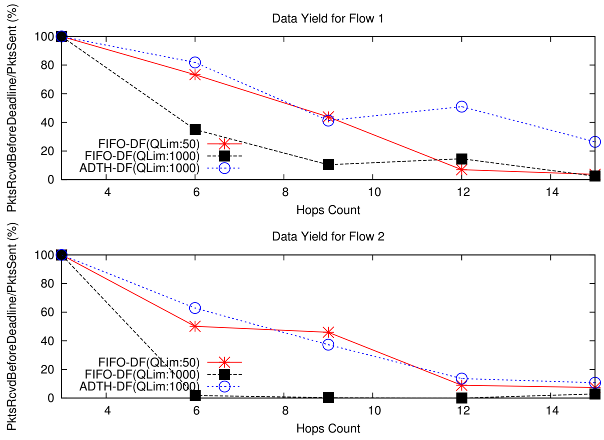
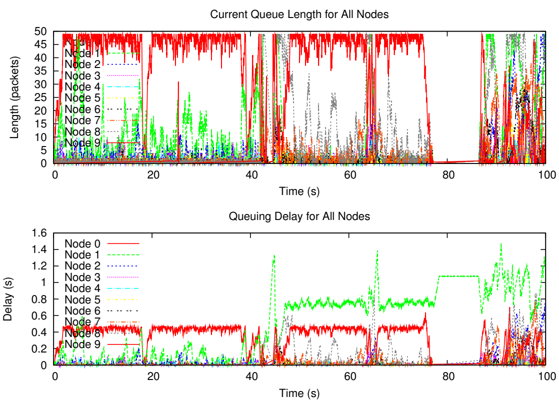


Figure 5.37: Scenario 5: Overall Data Yield (ADTH-DF versus FIFO-DF)

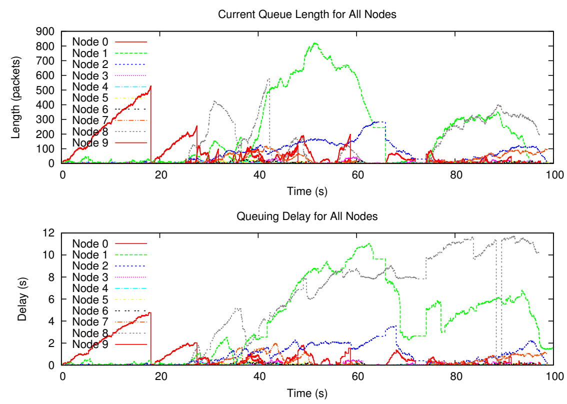
traffic rate from the source is maintained the same. This is due to the network is severely congested as a result of interference of neighbour nodes and the contention over shared bandwidth. When the queue limit is 1000 packets, the queuing delay observed is of course higher than the case where the queue limit is 50 packets as there are more backlogs in the queue when the queue limit is higher. Queuing delay and MAC layer delay contribute to large OWD as captured in Fig. 5.35. Fig. 5.38 shows the queue backlogs trend of FIFO-DF-50 and FIFO-DF-1000 for 9, 12 and 15 hops.

Fig. 5.39 shows the trend of queuing delay and queue length regulation for ADTH-DF. When H grows larger, D_{Nr} becomes more stringent in this simulation. Therefore, chances of packets overshoot D_{Nr} are higher. However, the chances of the overshot being cancelled out are also increased as some nodes may experience lower nodal delay. Overall, ADTH-DF is able to bound nodal delay by maintaining lower backlogs in queue. The adaptive nature of ADTH-DF enables packets to be dropped earlier when congestion is detected instead of wasting the resources to buffer and forward the packets. Indirectly, the network performance is improved and more bandwidth is available.

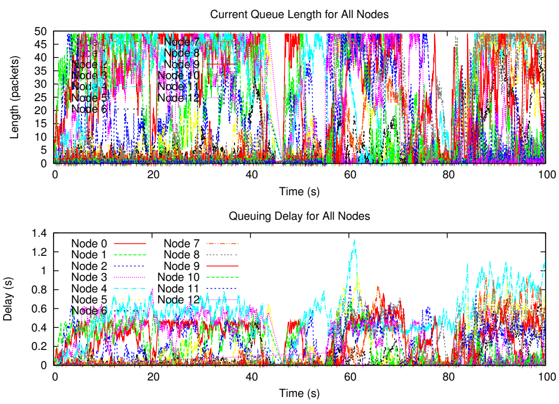
The simulation results show that when the number of hops increases, then network congestion is more intense. Packet loss becomes higher for ADTH-DF as ADTH-DF regulates the queue size actively to bound nodal delay. Higher packet loss is partly due to a more stringent D_{Nr} requirement when the number of hops increases. As a result, UDP goodput is lower when compared to FIFO-DF scheme. The trend of UDP goodput is not the same for all three schemes (Fig. 5.40). FIFO-DF-50 gives higher UDP goodput, but the bias on the UDP goodput towards 2 different flows is very obvious. This is bad for the network wide performance as only one flow monopolizes the bandwidth. The trends for FIFO-DF-1000 and ADTH-DF are quite similar. Under certain periods, FIFO-DF-1000 has higher throughput, however the gain in throughput is at the expense of extremely large OWD. Therefore, ADTH-DF is still a better choice for real-time traffic even with lower UDP goodput.



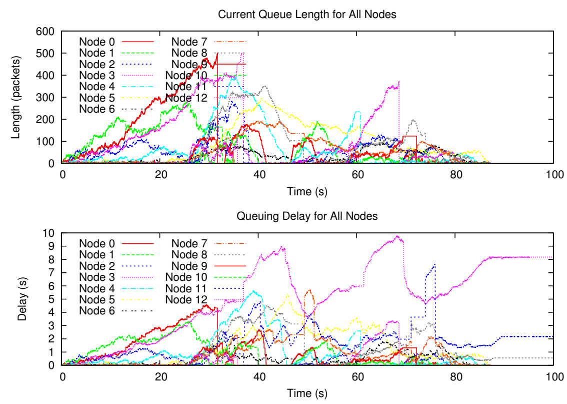
(a) FIFO-DF-50 (Hops Count = 9)



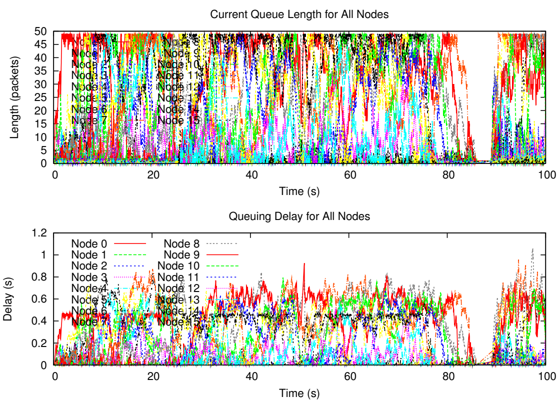
(b) FIFO-DF-1000 (Hops Count = 9)



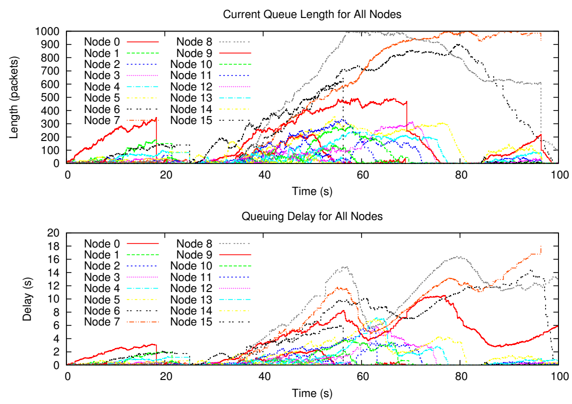
(c) FIFO-DF-50 (Hops Count = 12)



(d) FIFO-DF-1000 (Hops Count = 12)

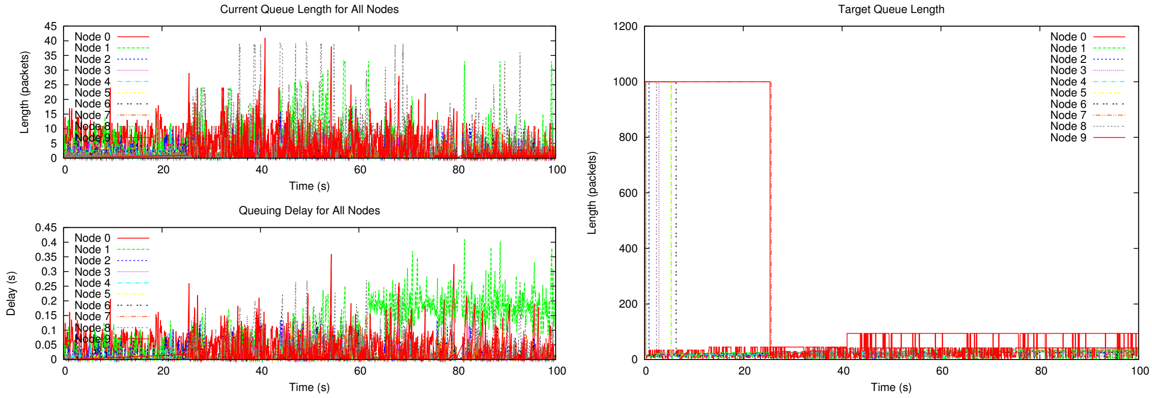


(e) FIFO-DF-50 (Hops Count = 15)

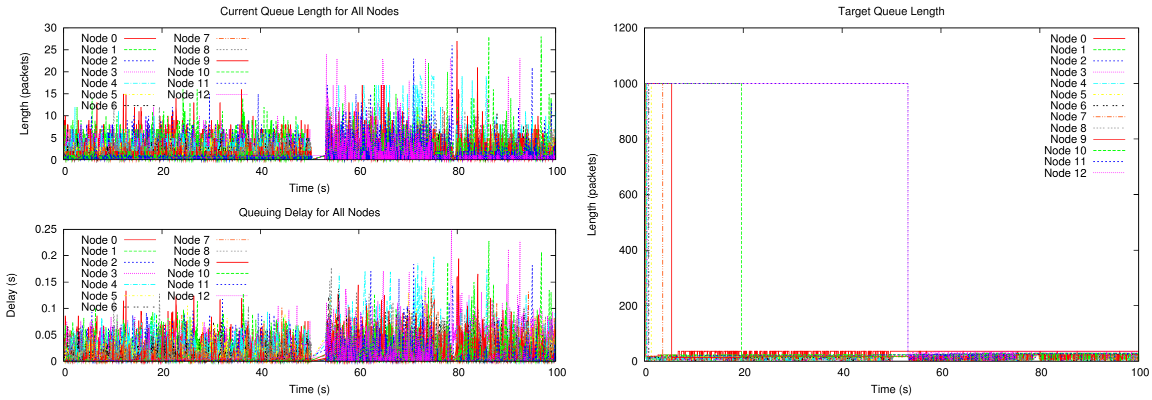


(f) FIFO-DF-1000 (Hops Count = 15)

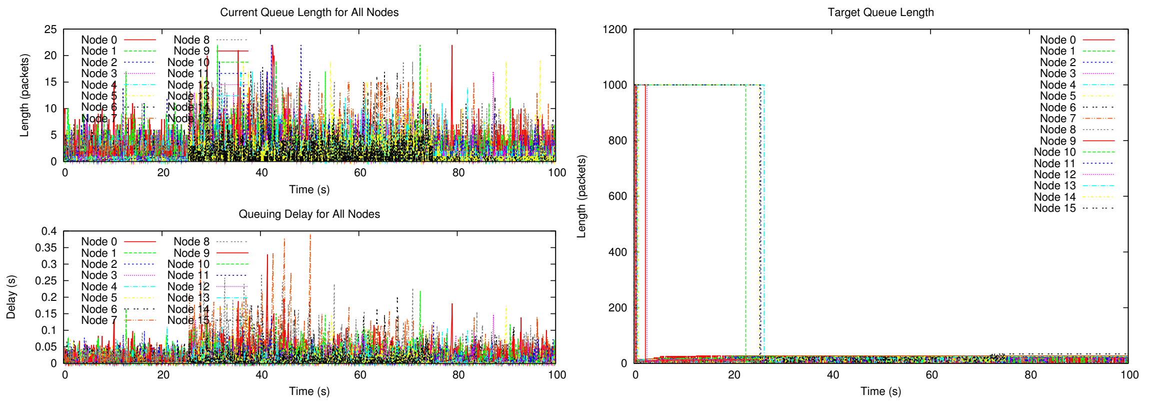
Figure 5.38: Scenario 5: Instantaneous Queue Length and Queuing Delay for FIFO-DF



(a) ADTH-DF (Hops Count = 9)

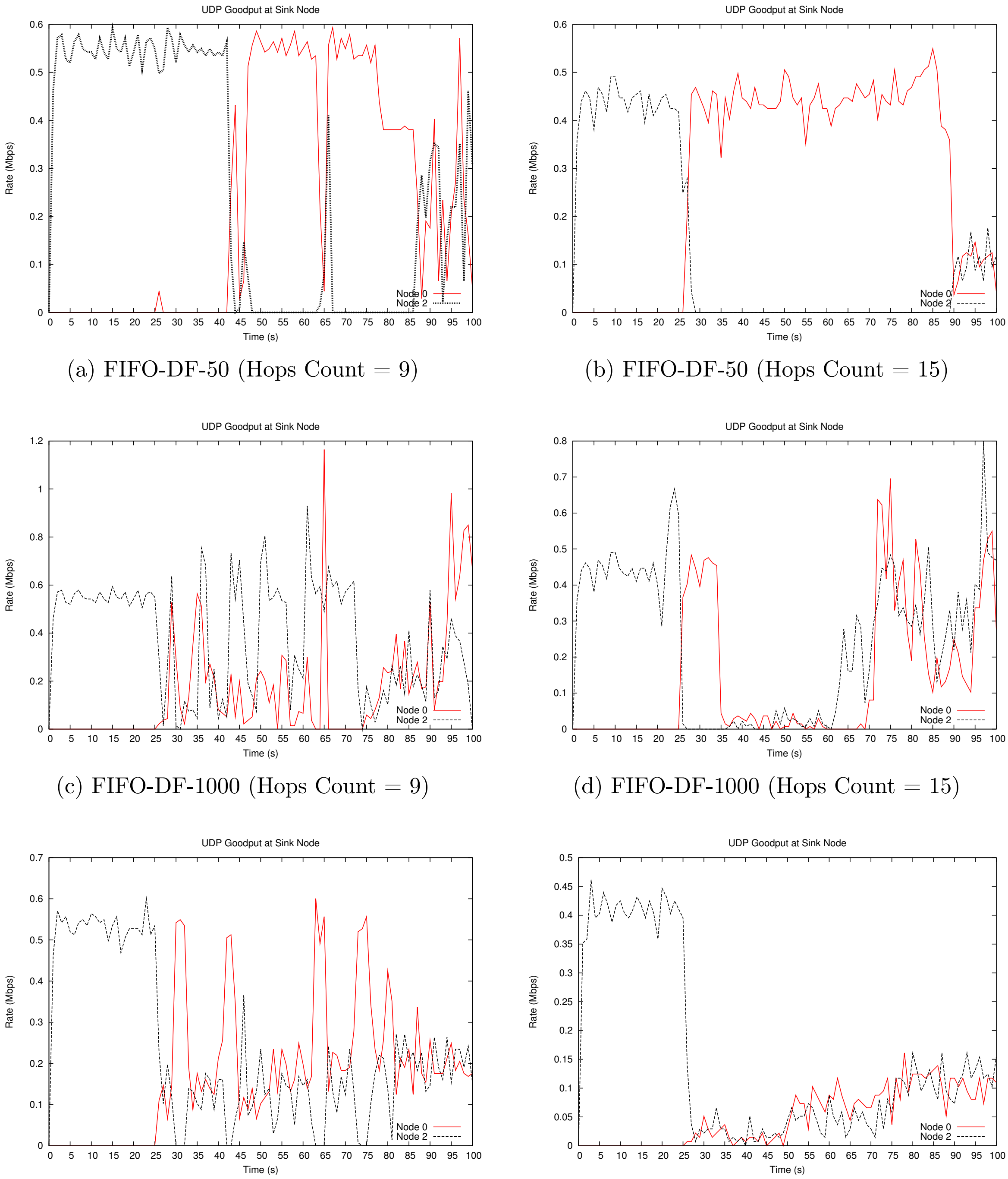


(b) ADTH-DF (Hops Count = 12)



(c) ADTH-DF (Hops Count = 15)

Figure 5.39: Scenario 5: Instantaneous Queue Length and Queuing Delay for ADTH-DF



(a) FIFO-DF-50 (Hops Count = 9)

(b) FIFO-DF-50 (Hops Count = 15)

(c) FIFO-DF-1000 (Hops Count = 9)

(d) FIFO-DF-1000 (Hops Count = 15)

(e) ADTH-DF (Hops Count = 9)

(f) ADTH-DF (Hops Count = 15)

Figure 5.40: Scenario 5: UDP Goodput for ADTH-DF versus FIFO-DF

5.6 Summary and Discussions

This chapter describes an adaptive queue management scheme (ADTH) that can constrain nodal delay of wireless nodes to a required value through an adaptive queuing threshold mechanism. The queuing threshold is estimated based on active system performance measurements with regards to the specified delay requirement and also the normalized error measured from the feedback control loop. From the simulation results, drop front policy is recommended for the ADTH scheme in order to achieve better system performance and efficiency. The aim of ADTH is to constrain nodal delay for delay-sensitive traffic (such as UDP traffic), but not delay-tolerant traffic (such as TCP traffic).

Wireless bandwidth is scarce especially for a multi-hop wireless network. All nodes within the same proximity of transmission and carrier sensing range share the bandwidth. Therefore, the bandwidth must be fully utilized in transmitting useful packets. Packets that potentially miss deadlines are discarded at earlier stage with ADTH in place. Early discard may alleviate congestion indirectly. However, mitigating congestion is not the goal of ADTH since that delay-sensitive traffic is not responsive to packet loss events. Nevertheless, ADTH contributes to congestion mitigation indirectly by reducing bandwidth wastage from transmitting packets which will be discarded eventually at the destination. Consequently, power consumption and wastage of network resources can be reduced by not processing and transmitting those packets.

Only localized information is gathered in ADTH for the queuing threshold regulation. There is no overhead in terms of messaging or signalling to gather information for the estimation process. The same queuing operation is applied to all packets entering the ADTH queue without differentiating the flows. ADTH does not maintain per flow information and thus makes it a lightweight design and scalable. In addition, ADTH is easy to be deployed as the configuration of ADTH is simple; only two parameters need to be configured in order to adopt the ADTH scheme. The parameters are target nodal delay (D_{Nr}) and sampling interval (t_s). The sampling interval is recommended to be $\leqslant \frac{1}{2}D_{Nr}$. While the D_{Nr} needs to be a reasonable target. There is not much effort needed in tuning the parameters to get optimized system performance since that ADTH is a self-adaptive queue management scheme.

The NS-2 simulation results show the ADTH scheme is able to react to network changes and constrain nodal delay of wireless nodes to give deterministic per-hop latency. It is able to bound nodal delay within the range of maximum allowable delay requirement at the expense of higher packet loss. The trade off between packet loss and latency is worthwhile if network delay is more critical than packet loss, as nodal delay at each hop has been significantly reduced. The result analysis shows that OWD can be bounded implicitly via per-hop control approach. Besides, the simulation results also show that ADTH is scalable. Nevertheless, NS-2 simulation is not able to capture network dynamics caused by wireless characteristics and also environmental interference precisely. Therefore, ADTH is implemented and validated on a testbed as detailed in Chapter 6.

Chapter 6

ADTH Testbed Implementation

6.1 Introduction

This chapter presents a performance analysis carried out on a testbed to prove that the ADTH queue management scheme is feasible to be implemented on hardware and yet achieves the goals of bounding per-hop nodal delay explicitly and end-to-end delay implicitly. The testbed has been setup in the laboratory environment to evaluate the ADTH scheme under real network conditions.

The ADTH scheme is shown to be effective and efficient in constraining nodal delay in NS-2 simulation. NS-2 simulation is widely used, but the results obtained from the simulation are still deterministic. In addition, the simulation scenarios in Chapter 5 only capture neighbour nodes interference but not environmental interference. The link quality may deteriorate or attenuate due to environmental interference such as obstruction of objects, wireless interference from other networks or equipment, etc. These signal fading and path loss characteristics are hard to be modelled in the simulation. Besides that, variation in processing speed, traffic inter-arrival time and hardware timing may also cause randomness in the network and contribute to network dynamics on top of link quality factor. Consequently, the performance of each wireless node and the performance of whole network are not deterministic.

ADTH is a standalone adaptive queue management scheme that does not rely on neighbour nodes information or a global view of network for it to work. The scheme mainly relies on internal performance metrics and feedback loop to bound nodal delay. These characteristics make it scalable. The scalability of the scheme is proven by the NS-2 simulation results in Chapter 5. Therefore, even the wireless ad hoc network setup in this testbed is small but is adequate to show the feasibility of the ADTH implementation.

The remainder of the chapter is organized as follows: Section 6.2 captures the equipment needed for the testbed setup and also configurations of the testbed; Section 6.3 describes the constraints and deviations of the testbed implementation; Section 6.4 presents the performance analysis carried out on the testbed; and finally the feasibility and efficiency of the ADTH queue management scheme are discussed in Section 6.5.

6.2 Testbed Setup

Equipment used for testbed setup and also the network topology are explained in the following subsections.

6.2.1 Testbed Equipment

6.2.1.1 Gumstix

Gumstix [1] is a single board computer that is equipped with the capability of embedded processing and network communication. Gumstix boards are setup as wireless nodes in the testbed. It is a linux-based miniature computer with a small form factor. Gumstix comprises of parts as shown in Fig. 6.1.

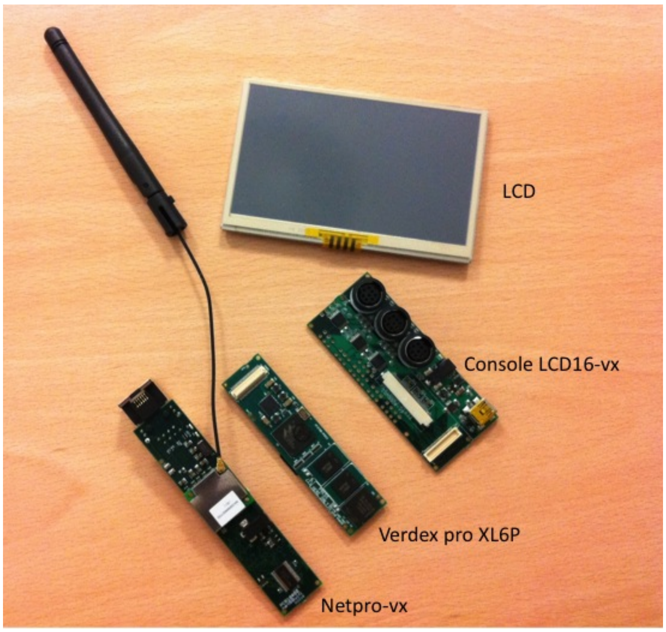


Figure 6.1: Gumstix Nodes

Verdex Pro XL6P is the motherboard of Gumstix. The processor on board is XScale PXA-270. The processing speed of PXA-270 is 624MHZ. It is an embedded processor that comes with 128MB RAM and 32MB flash. The processor supports 800MIPS (Millions Instruction Per Second). Netpro-vx is an expansion board for Gumstix that is used to enable wired and wireless communication. It supports

Fast Ethernet (10/100$Mbps$) wired communication via the Ethernet Port, and IEEE 802.11 b/g wireless communication via Marvell 88W8385 WiFi module. While Console LCD16-vx is an expansion board to connect a LCD (Liquid Crystal Display) to Gumstix and provides connectivity to serial ports and USB (Universal Serial Bus) port. The serial port is mainly used for debugging purpose and to re-flash Gumstix with a new kernel image.

6.2.1.2 Traffic Generator and Traffic Analyzer

Spirent Test Center (STC) traffic generator (Fig. 6.2) is used as a source of traffic generation for Gumstix wireless ad hoc network. It can send traffic at a required rate and to capture traffic with performance metrics. The traffic generator modules support up to 1Gbps traffic generation.

Figure 6.2: Spirent Test Center Traffic Generator

The accuracy of timing is one of the key factors for STC being selected instead of a software traffic generator. STC can capture the timing in the granularity of nano second. The STC chassis has a time reference generator which generates timestamps that are inserted into test packets for latency measurement. Latency is measured across the paired transmitting and receiving ports. This feature enables the performance analysis to be carried out accurately.

6.2.2 Testbed Configurations and Assumptions

The testbed has been setup according to the topology shown in Fig. 6.3 in the laboratory. Three Gumstix nodes are used to act as wireless nodes (WNs) in the wireless ad hoc network for most of the experiments except for the last experiment. Six wireless nodes are used to show that ADTH is scalable in the last experiment.

STC is used as traffic sources to generate UDP traffic flows. There is no wireless transceiver on STC, therefore the traffic generated is relayed to Gumstix at the edge over Ethernet cable and then being transmitted out via the wireless transceiver on Gumstix to next hop. Performance analysis has been carried out based on the results captured by traffic analyzer of STC.

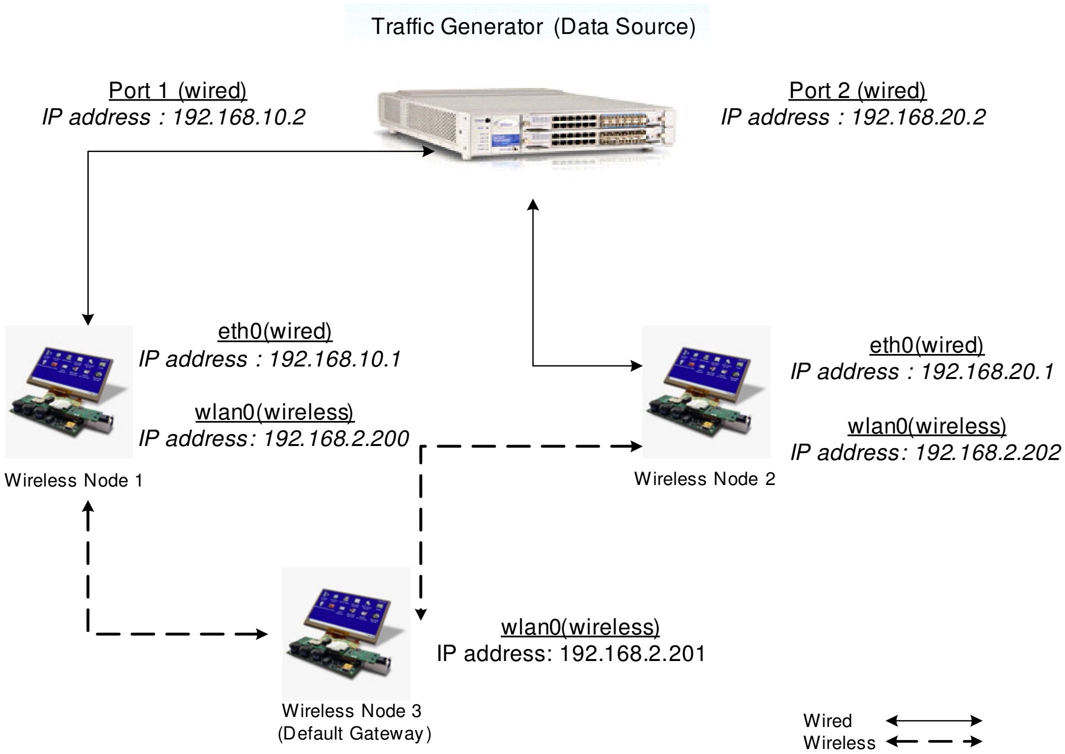


Figure 6.3: ADTH Testbed Topology

Communication between Gumstix nodes is wireless communication. All wireless nodes are within the transmission range and carrier sensing range of each other. Therefore, the nodes may interfere each other during transmission. This may cause network contention and congestion under heavy load. Besides interference from neighbour nodes, the testbed also subjects to environmental interference in the laboratory such as wireless access points within the vicinity of the testbed and other laboratory equipment. WN3 is placed in between WN1 and WN2 to create a multi-hop network. WN3 is configured as the gateway for these two nodes. Static routing is used to enforce the forwarding policy.

Each network interface on the WNs is assigned with a static IP address. Route tables of all wireless nodes are listed in Tables 6.1 - 6.3. IP forwarding is enabled at each node, so that each wireless node can act as a router to forward packets in a multi-hop environment. In this testbed setup, WN1 and WN2 are 2 hops away from each other. WN3 is responsible to forward packets for both directions.

The settings of transceiver are configured to be the same as NS-2 simulation in

Table 6.1: Routing Entries for Wireless Nodes 1 (WN1)

Destination	Gateway	Netmask	Interface
192.168.10.0	0.0.0.0	255.255.255.0	eth0
0.0.0.0	192.168.2.201	0.0.0.0	wlan0

Table 6.2: Routing Entries for Wireless Nodes 2 (WN2)

Destination	Gateway	Netmask	Interface
192.168.20.0	0.0.0.0	255.255.255.0	eth0
0.0.0.0	192.168.2.201	0.0.0.0	wlan0

Chapter 4 and Chapter 5. The auto-fallback feature is disabled and the maximum transfer data rate is set to 11Mbps. Power management and transmit power control are also disabled, so that the transmit power is always constant for each wireless node. The transmit power is configured to the minimum transmission power, which is 5dBm.

The delay between STC and Gumstix node is estimated via '*ping*' response from WN1 and WN2 to STC. RTT returned by the '*ping*' program is $0.5ms$, thus the delay estimated is only $0.25ms$ by assuming the transmission path is symmetric for the wired link. Therefore, the delay introduced between STC and wireless nodes are ignored. The testbed is mainly setup to transmit and forward packets, therefore the processing delay can be considered as minimal based on the processing speed of 800MIPS. The processing delay for receiving packets from Ethernet port (eth0) and forwarding to wireless interface (wlan0) are ignored as well with the assumption that the processing delay is minimal and also the magnitude of processing delay is far smaller than the magnitude of queuing delay and MAC layer delay. The Ethernet ports for Gumstix nodes are $100Mbps$, while the Ethernet ports on STC can support up to $1Gbps$. Therefore, wired transmission can never become a bottleneck here. The IFQ of eth0 is always drained fast and no backlog in the queue, therefore queuing delay contributed by eth0 can be ignored.

Table 6.3: Routing Entries for Wireless Nodes 3 (WN3)

Destination	Gateway	Netmask	Interface
192.168.10.0	192.168.2.200	255.255.255.0	wlan0
192.168.20.0	192.168.2.202	255.255.255.0	wlan0
192.168.2.0	0.0.0.0	255.255.255.0	wlan0
0.0.0.0	0.0.0.0	0.0.0.0	wlan0

6.3 Testbed Implementation

There are some constraints and deviations of implementation compared to the simulation version. The following subsections describe the constraints and deviations of the implementation on the testbed.

6.3.1 ADTH Implementation on Gumstix

Gumstix wireless nodes used in the testbed are running on Linux operating system with a kernel version 2.26.21. When Gumstix is powered on, the transmission queues of wireless and wired network interfaces are configured with a default queue size of 1000 packets. The transmission queue is known as IFQ in NS-2.

ADTH is implemented as a new queue discipline for network interfaces. It is a generic implementation that can be adopted for wired and wireless network interfaces. The queuing operations of ADTH are similar to the default pfifo (packet FIFO) queue discipline in Linux kernel [10, 19, 145] except the queue threshold is regulated by the ADTH controller. The ADTH controller is invoked periodically to estimate the target queue threshold. Instead of using the maximum queue size, the target queue threshold is used for decision-making in queuing operations.

The ADTH controller is invoked in a timer interrupt context when the timer is timeout. A kernel timer [149] is used to schedule a periodic timeout based on the sampling interval specified. The granularity of timer interrupt for the ADTH controller is at minimum one clock tick. The clock tick of Linux kernel is 100 per second for Gumstix, therefore the minimum granularity of timer interrupt is $10ms$. This means that the minimum sampling interval is $10ms$. Hence, the lower bound of D_{Nr} is two clock ticks ($20ms$) based on the sampling rule of $t_s \leqslant \frac{1}{2}D_{Nr}$. The sampling interval and the target nodal delay can be configured in multiple of $10ms$ (1 clock tick $\equiv 10ms$) in this implementation.

There are several deviations of the ADTH implementation in Linux kernel from the design detailed in Section 5.3. Queuing delay measurement in Linux kernel makes use of the timestamp field in the data structure of *skb_buff* (Linux socket buffer) header to store the timestamp instead of the ring buffer implementation. When a packet is enqueued into a transmission queue, the timestamp of enqueuing time is stored into the buffer header. The queuing delay is then calculated based on this timestamp when the packet is dequeued.

Besides that, a locking mechanism is very important in the kernel space. A locking mechanism is used to protect critical section of codes in order to provide mutual exclusive access and to prevent race conditions. The transmission queue access is protected by a queue lock (*dev->queue_lock*) in Linux kernel. Whenever packets are enqueued or dequeued, the queue lock must be acquired. When the

ADTH queue needs to be shrunk based on the target queue threshold estimated by the ADTH controller, the queue lock is acquired and released after discarding those packets potentially miss deadline.

Timing is very sensitive in kernel space, the execution time of timer interrupt cannot be too long. Otherwise, this may block other important interrupts such as device driver interrupts. Therefore, the operation to shrink the queue in the ADTH controller is moved from the timer interrupt function into the dequeue function. The queue lock is already acquired when dequeue operation is invoked, thus shrinking queue can be safely carried out in the dequeue function.

Probes are not inserted into the ADTH queue for performance result collection, as ADTH runs in the kernel space. A detailed analysis for internal variables is not recommended because these operations are intrusive and may cause changes of timings in a system. This may impact the system performance. Therefore, performance results are collected at the end points from STC traffic analyzer.

The ADTH queue discipline is built into the Linux kernel, Gumstix nodes are re-flashed with the new kernel image. Gumstix wireless nodes with the new image can then be configured to use the ADTH queue discipline via a user space utility that is known as Traffic Control (TC) Utilities. For details of TC, refer to Section 6.3.2.

6.3.2 Traffic Control Utility

'tc' [145] is a user space utility that is used to associate queues with the output devices for packets transmission. *'tc'* is needed to associate the ADTH queue to the wireless interface of Gumstix. The *'tc'* utility can be built into Linux by including the iproute2 package. The communication between *'tc'* utility and a ADTH queue is via messaging between a user space and a kernel space through *rtnetlink* sockets [145]. *rtnetlink* is based on *netlink* and being used to exchange traffic control parameters between the user space and the kernel space. Here is a list of *'tc'* commands for queue configuration.

- Associate a queue discipline to an output device

  tc qdisc add dev <interface> root <queue discipline> options

- Change the parameters of queue discipline for an output device

  tc qdisc change dev <interface> root <queue discipline> options

- Show the parameters of queue discipline for output devices

  tc -s -d qdisc show

- Disassociate the queue discipline from an output device

  tc qdisc del dev <interface> root

- Help text on usage of '*tc*'

  tc qdisc help

'*tc*' utility is modified to configure a ADTH queue. The modifications enable the ADTH queue to be used as an interface queue. Parameters of ADTH that can be configured by users are target nodal delay (D_{Nr}) in unit *ms*, the ADTH controller's sampling interval (t_s) in unit *ms* and maximum queue limit (*MAX_ Q_ LIMIT)* in unit packets. Here is the syntax of the ADTH configurations:

- *tc qdisc add dev <interface> root adth_ pfifodf limit <maximum queue limit> target_ nd <target nodal delay> interval <sampling interval>*

- *tc qdisc change dev <interface> root adth_ pfifodf limit <maximum queue limit> target_ nd <target nodal delay> interval <sampling interval>*

6.3.3 Processing Overhead

The elapsed time (t_e) for the ADTH estimator at each node is average $14\mu s$ per sampling period. The elapsed time is measured by inserting timestamps at the beginning and the end of the ADTH estimator function. The overhead incurred by the ADTH estimator can be estimated via Eq. 6.1.

$$OH_{ADTH} = t_e * 1/t_s \tag{6.1}$$

Taking an example of $t_s = 50ms$, the sampling frequency per second is 20. Therefore, the total processing overhead incurred is $280\mu s$ per second. The overhead incurred becomes smaller if the processing speed is higher or if the sampling frequency is lower.

The overhead of nodal delay measurement is around 3 to $4\mu s$ per packet. The overhead incurred to each packet is insignificant when compared to end-to-end delay. The current implementation on the testbed is per packet. The overhead can be reduced by sampling nodal delay at lower frequency. Other than that, faster processor speed would bring down the overhead incurred.

ADTH can be claimed as a lightweight algorithm based on the overhead analysis above. Overhead incurred is insignificant when compared to end-to-end delay. Furthermore, the processing power is not the bottleneck in the communication world since the processor speed is catching up fast and also with the introduction of multi-cores in a processor.

6.4 Performance Analysis

Fig. 6.4 shows the testbed setup in the laboratory. WN1 and WN2 are connected to external keyboards, while WN3 is connected to a laptop via a serial port. The laptop is used to control STC for traffic generation and to collect performance results captured by STC. The performance analysis has been carried out based on the performance metrics captured by the STC traffic analyzer. The performance analysis is done at the end points instead of at intermediate nodes. The end points in this performance analysis are STC traffic generator modules. QoS metrics such as OWD, overall packet loss, miss deadline ratio and packet delivery ratio are compiled from the results obtained from the STC traffic analyzer. Node level performance is not captured to avoid performance impact on the systems due to intrusive performance measurement in the kernel space at each node.

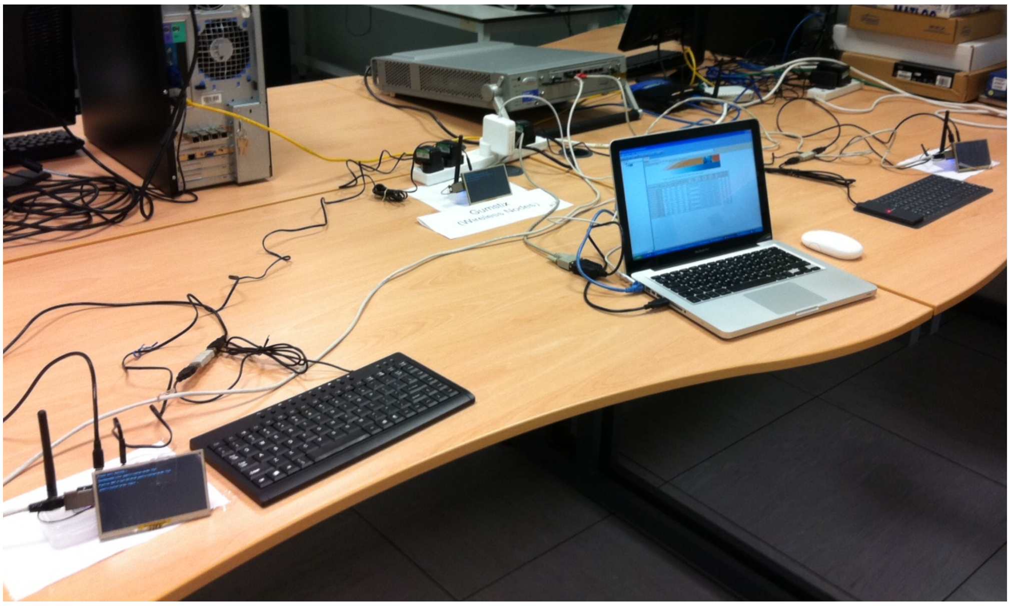
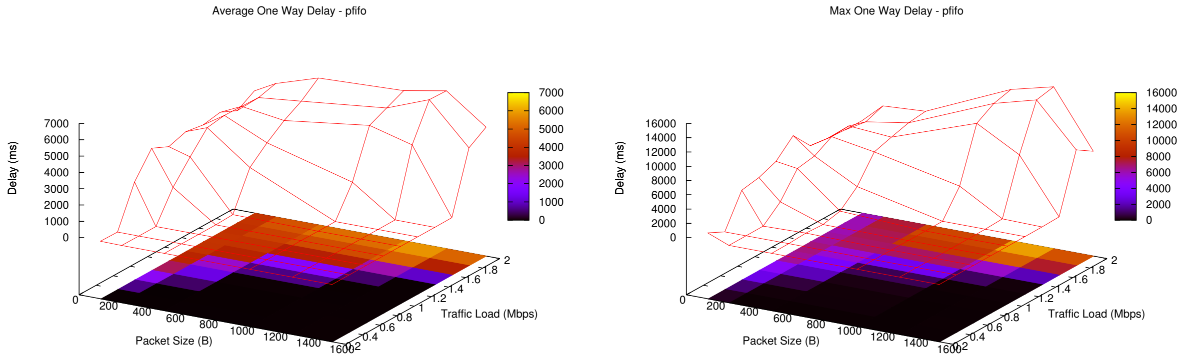


Figure 6.4: Testbed Setup in Lab

Bi-directional traffic is applied to all experiments. Traffic flows are started immediately from both directions at both source ports of STC when the experiment begins. The traffic duration is set to 100s for each run. The ADTH queue discipline is compared against pfifo queue discipline. The maximum queue limit for both queue disciplines is set to 1000 packets (default queue size).

6.4.1 Experiment 1: ADTH validation

ADTH has been validated and compared against pfifo with a custom list of packet sizes and incremental traffic loads. The default custom packet sizes of STC are $128B$, $256B$, $512B$, $1024B$, $1280B$ and $1518B$. The traffic load (R) is [200..2000]



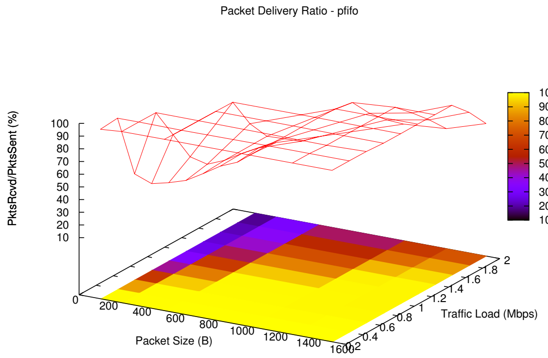
(a) Average OWD (b) Maximum OWD

Figure 6.5: Experiment 1: Overall OWD Trend for pfifo

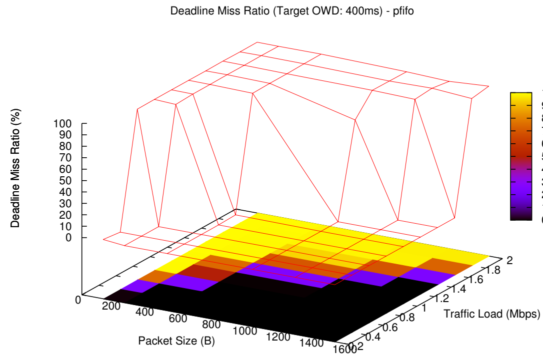
kbps with stepping of $200kbps$. This experiment is to validate the claim of ADTH being able to constrain nodal delay regardless of packet sizes and traffic load provided that the network is not too saturated. D_{Nr} is set to $200ms$ for each node, thus OWD_R is $400ms$. The effects of sampling interval are also analyzed in this experiment. The experiment is repeated with $t_s = 100ms \, (\equiv \frac{1}{2}D_{Nr})$ and $t_s = 50ms \, (\equiv \frac{1}{4}D_{Nr})$.

Fig. 6.5 shows the average OWD and maximum OWD recorded for pfifo under different packet sizes and different traffic loads. The average OWD recorded can be $> 6s$ and the maximum OWD recorded can be $> 14s$ when the network is saturated. The network starts congested even R is just $0.2Mbps$ for small sized packets (e.g. $128B$ and $256B$). The network congestion becomes more severe with the increment of traffic load. While for bigger sized packets, the network starts congested when $R \geqslant 0.6Mbps$. This is consistent with the observations in NS-2 simulations that the network becomes congested at high traffic load and also if the traffic load consists of small sized packets. Although the network is more saturated at high traffic load for small sized packets compared to big sized packets, the OWD observed for big sized packet is larger as a result of larger queuing delay contributed by the packet size.

The packet delivery ratio of pfifo decreases when network starts congested and becomes extremely low when the network is saturated for packet size $128B$ and $256B$ with high traffic load (Fig. 6.6(a)). The packet delivery ratio is 100% if no congestion in the network. Although the packet delivery ratio is $\geqslant 50\%$ for most of the cases when network contention is high, the deadline miss ratio can be as high as 100% (Fig. 6.6(b)). This means that none of the packets gets delivered to the destination within the delay requirement specified. This is unacceptable for delay-sensitive applications. The ADTH scheme becomes a remedy for this.

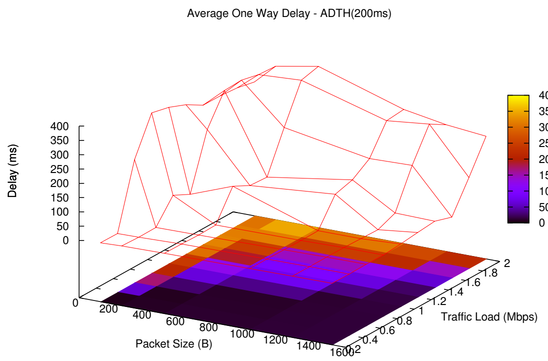


(a) Packet Delivery Ratio

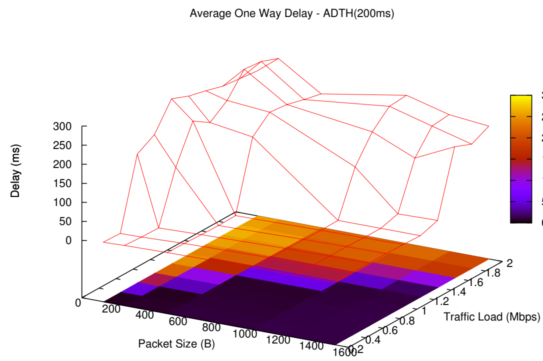


(b) Deadline Miss Ratio

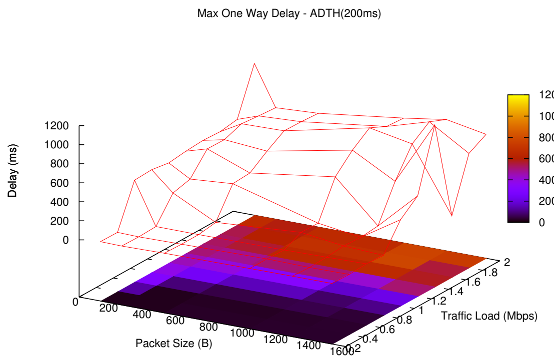
Figure 6.6: Experiment 1: Overall Packet Delivery Ratio and Deadline Miss Ratio Trend for pfifo



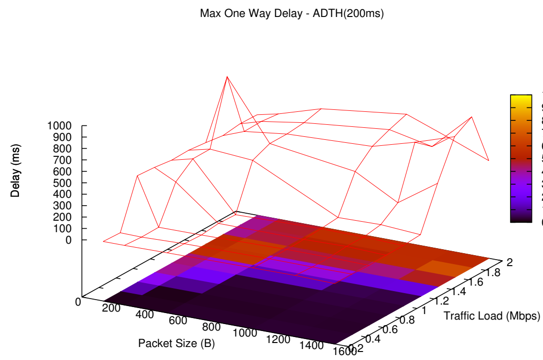
(a) Average OWD ($t_s = 100ms$)



(b) Average OWD ($t_s = 50ms$)



(c) Maximum OWD ($t_s = 100ms$)

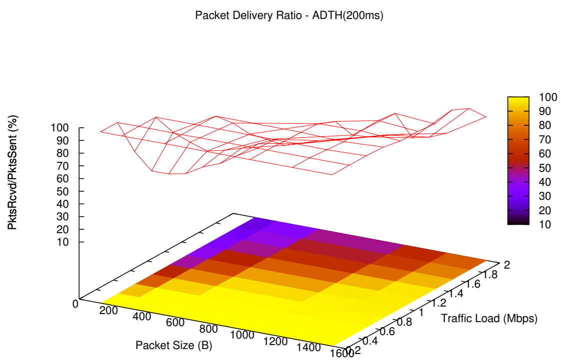


(d) Maximum OWD ($t_s = 50ms$)

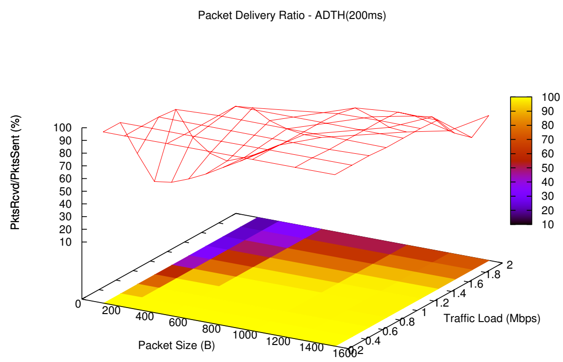
Figure 6.7: Experiment 1: Overall OWD Trend for ADTH ($D_{Nr} = 200ms$)

When the ADTH scheme is used, a significant improvement over network delay is observed. The average OWD recorded is $\leqslant OWD_R$ for both sampling intervals (Fig. 6.7). However, some packets reach the destination beyond the required

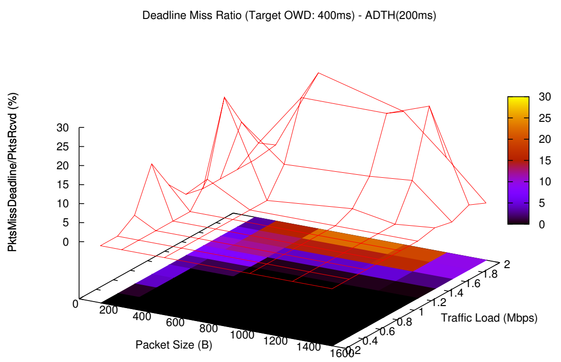
OWD. When the network is too saturated, MAC layer delay is potentially greater than the target nodal delay. Thus, ADTH is unable to bound the nodal delay in this circumstances. The maximum OWD recorded is $> 600ms$ for the case of $t_s = \frac{1}{2}D_{Nr}$ and $> 400ms$ for the case of $t_s = \frac{1}{4}D_{Nr}$ when the network is severely congested. Nevertheless, the deadline miss ratio is low (Fig. 6.8). Most of the packets get delivered to the destination within the bound of OWD_R except when the network is highly saturated.



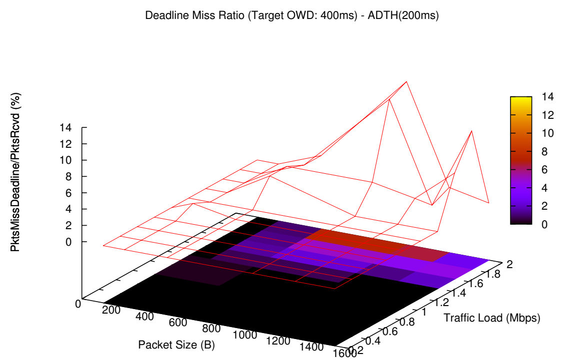
(a) Packet Delivery Ratio ($t_s = 100ms$)



(b) Packet Delivery Ratio ($t_s = 50ms$)



(c) Deadline Miss Ratio ($t_s = 100ms$)



(d) Deadline Miss Ratio ($t_s = 50ms$)

Figure 6.8: Experiment 1: Overall Packet Delivery Ratio and Deadline Miss Ratio Trend for ADTH

The packet delivery ratio for ADTH is similar to pfifo, but the deadline miss ratio is significantly improved with ADTH enabled. The advantage of having a finer sampling interval is highlighted here. The deadline miss ratio for ADTH with $t_s = \frac{1}{2}D_{Nr}$ is higher than the case with $t_s = \frac{1}{4}D_{Nr}$ (Fig. 6.8). The ADTH controller needs to be invoked at a higher frequency due to a higher level of interference in laboratory environment. So that it can respond faster to errors measured and hence mitigates the overshooting of nodal delays. For most of the cases, the deadline miss ratio for ADTH with $t_s = \frac{1}{4}D_{Nr}$ is either 0% or $\leqslant 3\%$; only a few sub-cases (e.g. packet sizes are $1024B$ and $1280B$ at traffic load $\geqslant 1.8Mbps$) are recorded with the deadline miss ratio $\geqslant 6\%$. The deadline miss ratio for ADTH with $t_s = \frac{1}{4}D_{Nr}$ is only 0.99% with standard deviation (SD) 2.67% but 4.01% with SD 7.47% for the case of $t_s = \frac{1}{2}D_{Nr}$. Nevertheless, the deadline miss ratio is still far better than pfifo which has 47.51% of deadline miss ratio with SD 49.51%. The pfifo is just a normal interface queue with a constant queue size. Therefore, there is no control over network delay.

Packet loss ratio trends are similar for pfifo and ADTH (Figs. 6.9 and 6.10). However, the packet loss ratio for ADTH is slightly higher than pfifo in general due to early discard nature of ADTH. The packet loss ratio becomes higher when the traffic load is increased and the packet size is decreased. Packet loss is caused by overflowing of the transmission queue and collisions. The packet loss ratio is 0% when the network load is light as the network is not congested. All nodes get the chance to transmit packets fast. Although the packet loss ratios are similar for ADTH and pfifo here, the packet loss ratio will be higher for ADTH if the target nodal delay is more stringent. Nevertheless, the packet loss ratio of ADTH is lower for the case with $t_s = \frac{1}{4}D_{Nr}$ when compared to the case with $t_s = \frac{1}{2}D_{Nr}$ as ADTH performs better with a finer sampling interval.

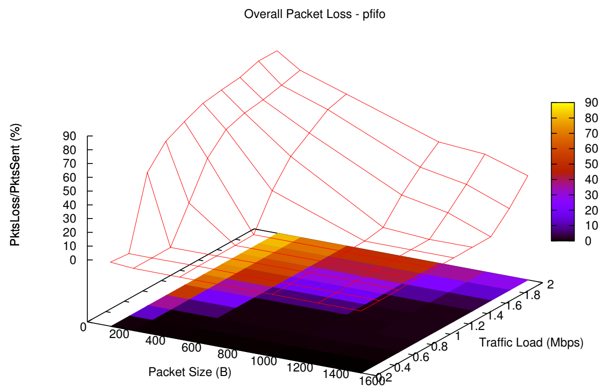
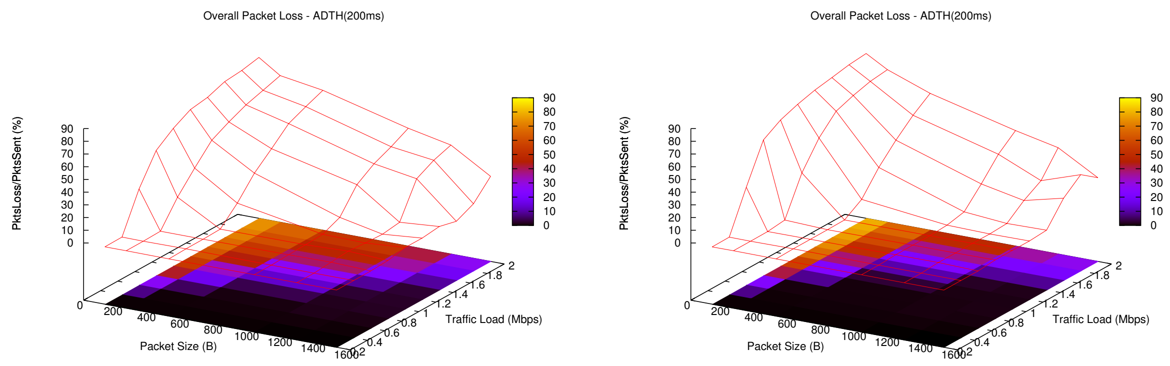


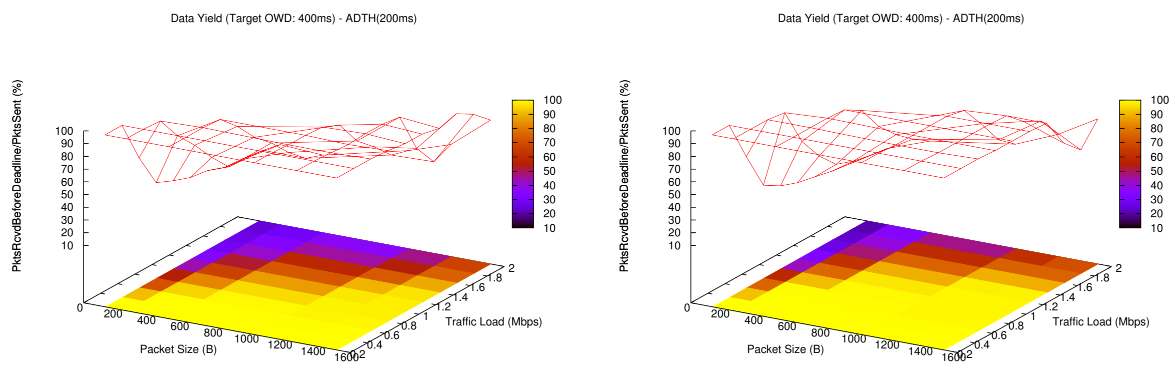
Figure 6.9: Experiment 1: Overall Packet Loss Ratio Trend for pfifo



(a) $t_s = 100ms$ (b) $t_s = 50ms$

Figure 6.10: Experiment 1: Overall Packet Loss Ratio Trend for ADTH ($D_{Nr} = 200ms$)

Overall, the ADTH scheme has significantly improved the control over network delay as shown by the data yield (Fig. 6.11). The data yield of ADTH is almost the same as the packet delivery ratio since the deadline miss ratio is either 0% or very low for most of the cases even when the network is congested. While for pfifo, the data yield drops drastically when the network is severely congested and the data yield can be 0% (Fig. 6.12). The data yield for ADTH ($t_s = \frac{1}{4}D_{Nr}$) is average 76.19% and for pfifo is only 51.94%.



(a) $t_s = 100ms$ (b) $t_s = 50ms$

Figure 6.11: Experiment 1: Overall Data Yield Ratio Trend for ADTH

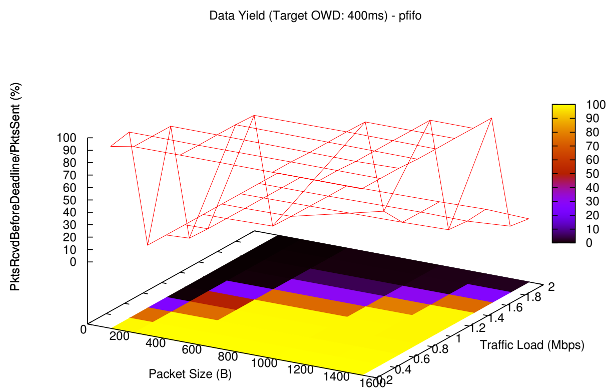


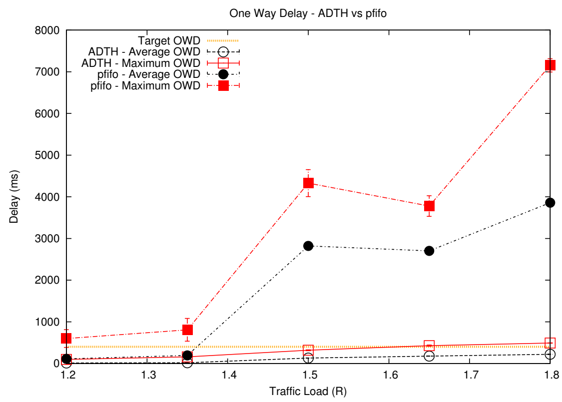
Figure 6.12: Experiment 1: Overall Data Yield Ratio Trend for pfifo

The experiment shows the ADTH controller has better control over network delay with a finer sampling interval. This is because with smaller sampling interval, the ADTH controller is invoked more frequent and thus the queue size is regulated more frequent. Thus, the ADTH controller can respond to network dynamics faster to prevent overshooting of nodal delay for packets in the queue. The interference is greater in the testbed environment as compared to simulation environment. Therefore, a finer sampling interval should be used to enable faster response. The sampling interval is suggested to be $\leqslant \frac{1}{4}D_{Nr}$ for the rest of the experiments in this chapter.

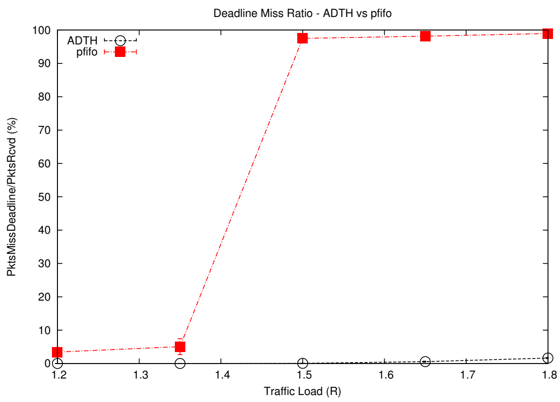
6.4.2 Experiment 2: Different traffic loads with random packet sizes

ADTH has been further evaluated with an incremental of traffic load but with random packet sizes instead of a constant packet size. Traffic generated by applications can be of any sizes depending on traffic type and also the applications' attributes. The packet sizes range from $128B$ to $1518B$ for this experiment. The bi-directional traffic load (R) is in the range of [1.2..1.8] $Mbps$ with stepping of $150kbps$. D_{Nr} is set to $200ms$ for each wireless node, thus OWD_R is $400ms$. The sampling interval is set to $50ms$ ($\equiv \frac{1}{4}D_{Nr}$).

Fig. 6.13(a) shows the average OWD and the maximum OWD of each configuration for both the pfifo and ADTH schemes. Each configuration in this experiment was run for 50 times, the error bars in the graph show the standard errors (SE) of the experiments run. When $R \leqslant 1.35Mbps$, the network is not yet highly congested as the average OWD recorded for pfifo is $\leqslant OWD_R$. However, the deadline miss ratio is not negligible (Fig. 6.13(b)). The ratio is average 3.43% with SD 9.99% for $R = 1.2Mbps$ and average 5.05% with SD 16.86% for $R = 1.35Mbps$. The deadline miss ratio increases drastically after $R \geqslant 1.5Mbps$ to an unacceptable range which is $>$ 97% in average. Compared to pfifo, the deadline miss ratio for ADTH is insignificant. The average OWD for pfifo change from $< 1s$ to nearly $4s$ when $R \geqslant 1.5 Mbps$. This shows how serious that the impact of network contention in wireless environment towards network delay.



(a) Overall OWD Trend



(b) Overall Deadline Miss Ratio Trend

Figure 6.13: Experiment 2: Overall OWD and Deadline Miss Ratio Trend (ADTH versus pfifo)

The experiment results (Fig. 6.14) show that ADTH has successfully bounded OWD to the required OWD_R. This is the effect of queuing threshold regulation at each wireless node to achieve bounded nodal delay and thus bounded OWD. The average OWDs recorded for ADTH are $\leqslant OWD_R$, this implies that most of the packets get transmitted to the destination timely. Only some of the packets get transmitted beyond the required OWD at $R = 1.65 Mbps$ and $R = 1.8 Mbps$. However, the deadline miss ratio is very low. The ratio is average 0.55% with SD 1.01% for $R = 1.65 Mbps$ and average 1.64% with SD 1.12% for $R = 1.8 Mbps$. There are chances that packets being transmitted at the next sampling interval exceeds queuing delay due to lower queue throughput or MAC layer delay exceeds the estimate. Therefore, some packets may overshoot the target nodal delay before the ADTH controller is invoked.

Since that ADTH drops packets more aggressively compared to pfifo in order to constrain nodal delay, it is expected that the packet loss ratio of ADTH is higher than pfifo especially when the network is very congested (Fig. 6.15). When the network is very congested, MAC layer delay become larger, so queuing delay needs to be smaller to make sure the bounded nodal delay is still achievable. Therefore, higher packet drop rate is expected for ADTH. The packet loss ratio of ADTH is only higher for around 2% to 3% but more than 10x gain in network delay control is obtained based on the average OWD recorded. If the network is not congested or the contention level is light, packet loss ratio is similar for both pfifo and ADTH.

An interesting observation is shown when $R = 1.5Mbps$, the network contention is high at this rate but the network is not yet saturated as the packet loss ratio is not too high. The packet loss ratio for ADTH is indeed lower than pfifo in this case. The average for ADTH is 6.53% and 15.98% for pfifo. This shows that early discard attribute of ADTH plays its role in easing the network contention since less bandwidth is wasted to transmit overdue packets. Fewer packets in the network means less contention and thus more packets get transmitted to the destination.

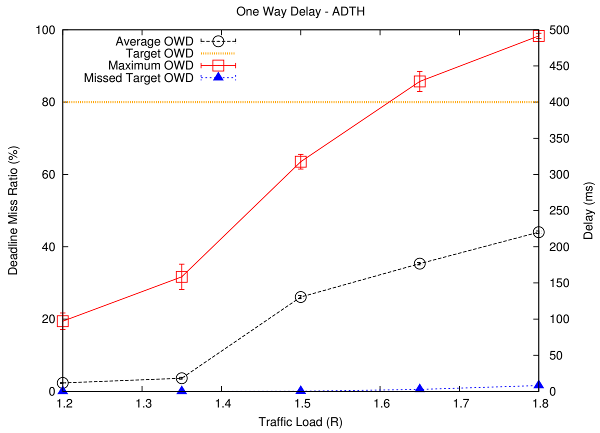


Figure 6.14: Experiment 2: Overall OWD with Deadline Miss Ratio Trend (ADTH)

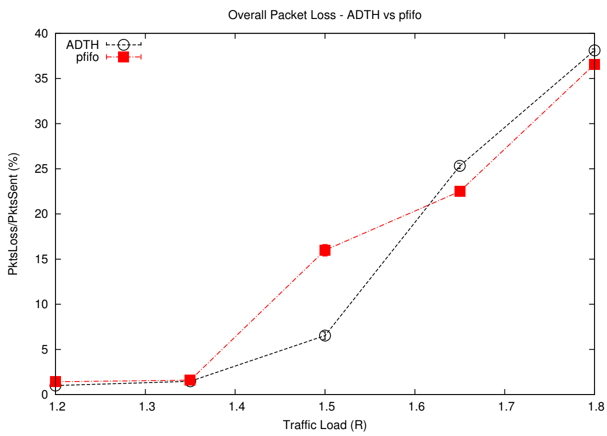


Figure 6.15: Experiment 2: Overall Packet Loss Ratio Trend (ADTH versus pfifo)

The average throughput for both schemes (Fig. 6.16) are closed to the expected throughput when the traffic load is light. When $R \geqslant 1.5Mbps$, the system performance deteriorates drastically. The average throughput decreases with the incremental of traffic load. Higher packet loss ratio and larger latency are also observed. The average throughput for ADTH is slightly lower than pfifo when $R \geqslant 1.65Mbps$, but higher than pfifo when $R = 1.5Mbps$ as less contention in the network due to regulation of ADTH as explained. The average throughput is much lower than the expected throughput when the contention is high. Bandwidth is wasted due to medium contention and packet retransmission at MAC layer.

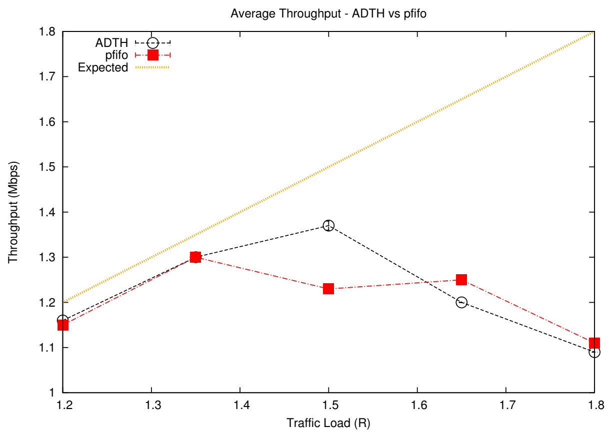
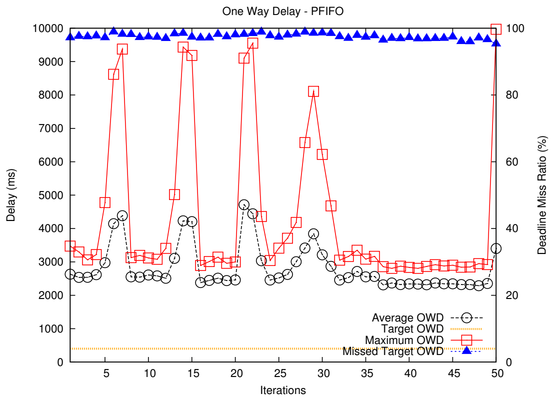
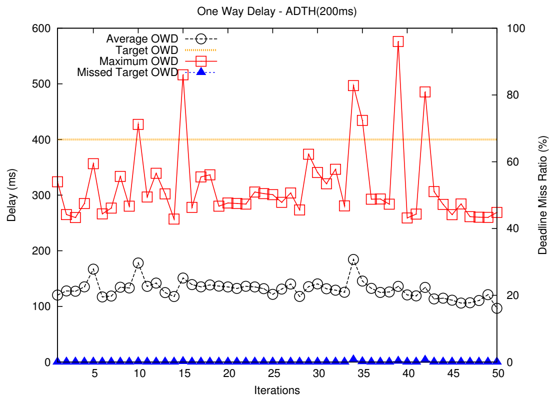


Figure 6.16: Experiment 2: Overall Average Throughput Trend (ADTH versus pfifo)

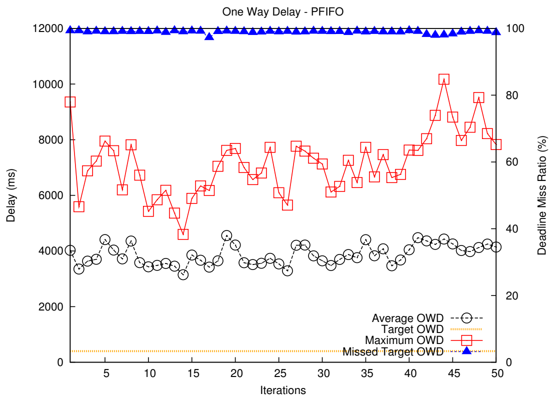
Fig. 6.17 shows the snapshots of OWD trends of pfifo and ADTH for 50 runs at $R = 1.5Mbps$ and $R = 1.8Mbps$. The deadline miss ratio trend is also shown in the figure. There are some spikes observed in OWD recorded throughout the 50 runs of experiments. This is due to interference in the network varies over time. The snapshot simulation results show that the deadline miss ratios of pfifo for all runs are $> 97\%$. In addition, the average OWDs of pfifo for all runs are huge when compared to ADTH. The average OWD is reduced significantly when ADTH is enabled. The average OWD is reduced from $> 2s$ to $< 200ms$ when $R = 1.5Mbps$ and from $\simeq 4s$ to $\simeq 200ms$ when $R = 1.8Mbps$.



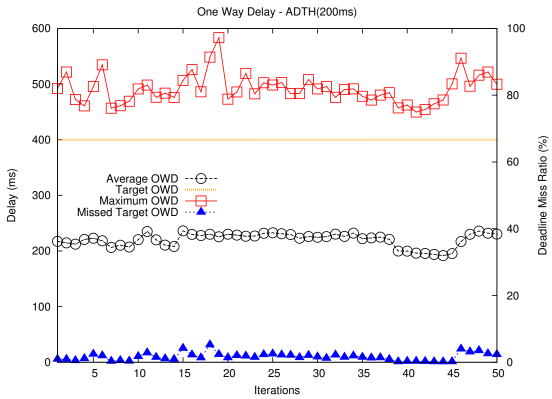
(a) pfifo ($R = 1.5Mbps$)



(b) ADTH ($R = 1.5Mbps$)



(c) pfifo ($R = 1.8Mbps$)



(d) ADTH ($R = 1.8Mbps$)

Figure 6.17: Experiment 2: OWD and Deadline Miss Ratio for Each Run @ $R = 1.5Mbps$, $1.8Mbps$ (ADTH versus pfifo)

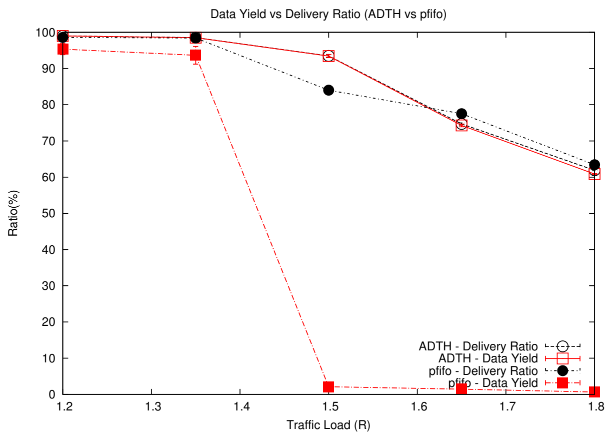
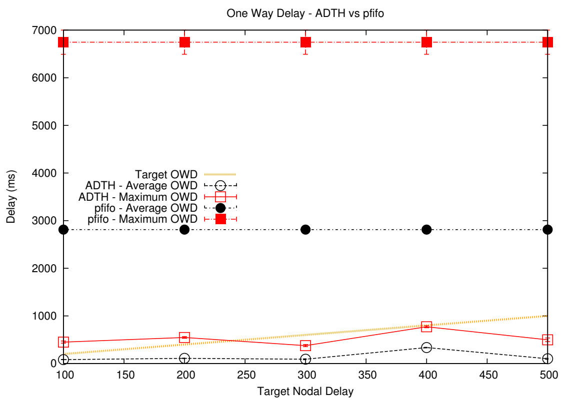


Figure 6.18: Experiment 2: Overall Data Yield and Packet Delivery Ratio Trend (ADTH versus pfifo)

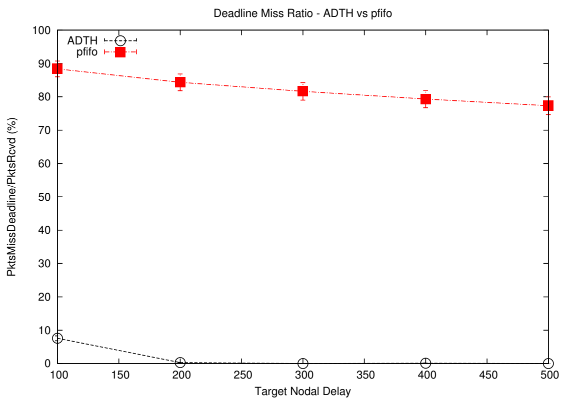
Overall, the experiment results show that ADTH can constrain network delay with slightly higher packet loss for certain cases. The data yield of ADTH is very closed to the data delivery ratio (Fig. 6.18), whereas for pfifo the data yield is extremely low when network contention is high due to large nodal delay and OWD.

6.4.3 Experiment 3: Different target nodal delays with random packet sizes and random traffic loads

ADTH has been evaluated with different target nodal delay requirements in this experiment. Besides packet sizes are random (range from $128B$ to $1518B$), traffic load is varied randomly between $1Mbps$ to $1.8Mbps$ to emulate time-varying traffic load and contention. Experiment 2 shows the level of contention is light when $R \leqslant 1.35Mbps$ as the packet loss ratio is very low and the OWD for pfifo is also low. The level of contention becomes more intense and the network becomes saturated when R increases beyond $1.5Mbps$. The experiment intends to show that ADTH can adapt to dynamic changes in the network efficiently and autonomously. It also intends to show that ADTH can be configured to bound nodal delay at a different required value in a dynamic environment. D_{Nr} is configured to [100..500] ms with the stepping of $100ms$. The sampling interval is set to $\frac{1}{4}D_{Nr}$.



(a) Overall OWD

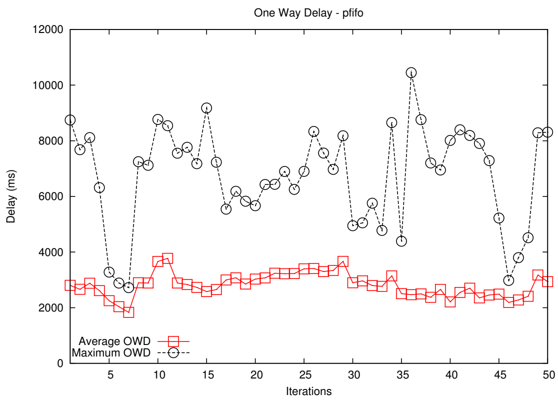


(b) Overall Deadline Miss Ratio

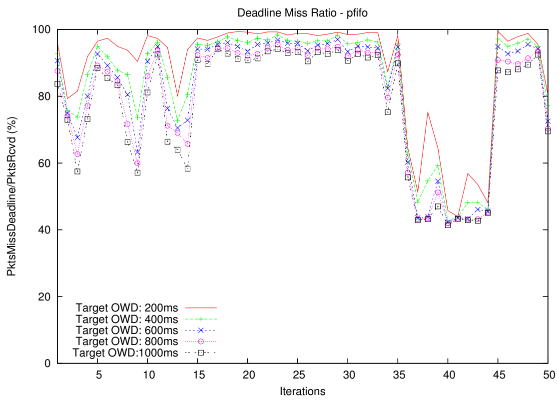
Figure 6.19: Experiment 3: Overall OWD and Deadline Miss Ratio Trend (ADTH versus pfifo)

The experiment has been carried out for ADTH with 50 runs for each D_{Nr}; whereas pfifo has been carried out 50 runs for once since that D_{Nr} value is meaningless to pfifo. Therefore, the performance results of pfifo for all D_{Nr} are the same. Fig. 6.19(a) shows the average and maximum OWD for both ADTH and pfifo. pfifo is unable to satisfy any of the delay requirements specified by D_{Nr}. The average OWD recorded is nearly $3s$ and the maximum OWD recorded is nearly $7s$ for pfifo. The latency is considered very high, as the destination is just 2 hops away from the source for both directions. The deadline miss ratio for pfifo decreases slightly with the relaxation of delay requirement (Fig. 6.19(b)); but it is still very high and unacceptable for delay-sensitive traffic. The deadline miss ratio range is average 88.37% with SE 2.36 for $D_{Nr} = 100ms$ and average 77.32% with SE 2.63 for $D_{Nr} = 500ms$.

Fig. 6.20 shows the OWD trend and deadline miss ratio trend of pfifo for 50 runs. The results show that the performance of pfifo fluctuates. For most of the repetitions, deadline miss ratio is very high except for the runs from 35 to 45. The variation observed is owed to the contention level is dynamic in the network resulting from time-varying environmental interference and also interference created by the random traffic load and random packet sizes.



(a) OWD Trend



(b) Deadline Miss Ratio

Figure 6.20: Experiment 3: OWD and Deadline Miss Ratio Trend for Each Run (pfifo)

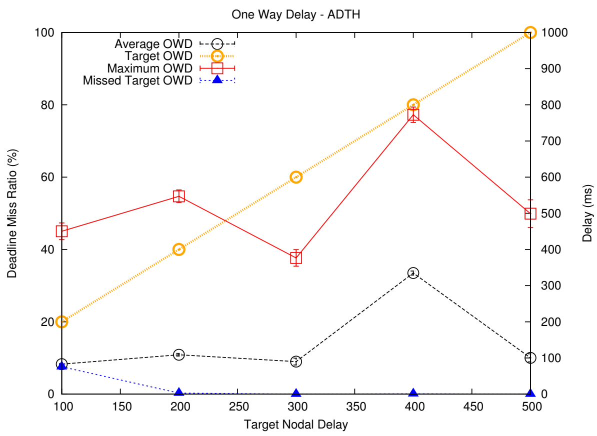


Figure 6.21: Experiment 3: Overall OWD with Deadline Miss Ratio Trend (ADTH)

For ADTH, the average OWD is $< OWD_R$ for all D_{Nr}. There are some packets reach the destination with the maximum OWD recorded $> OWD_R$ (Fig. 6.21).

However, the deadline miss ratio is not high. The ratio is average 7.57% with SE 0.73 for $D_{Nr} = 100ms$ and average 0.29% for $D_{Nr} = 200ms$ with SE 0.04. The deadline miss ratio for $D_{Nr} = 200ms$ is actually negligible. When the target nodal delay requirement is $> 200ms$, the deadline miss ratio is merely 0%. The deadline miss ratio is higher at a smaller delay requirement as MAC layer delay potentially higher than the required value when high contention.

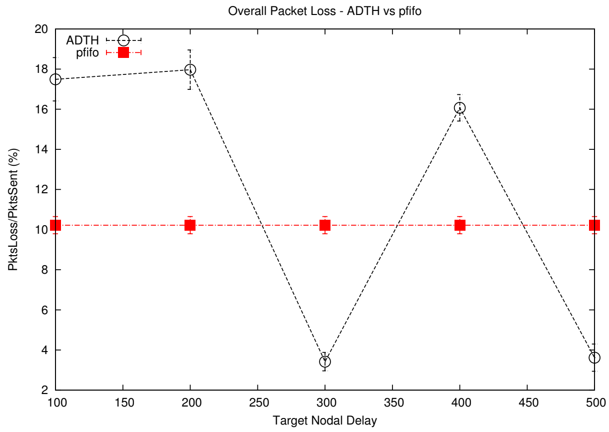
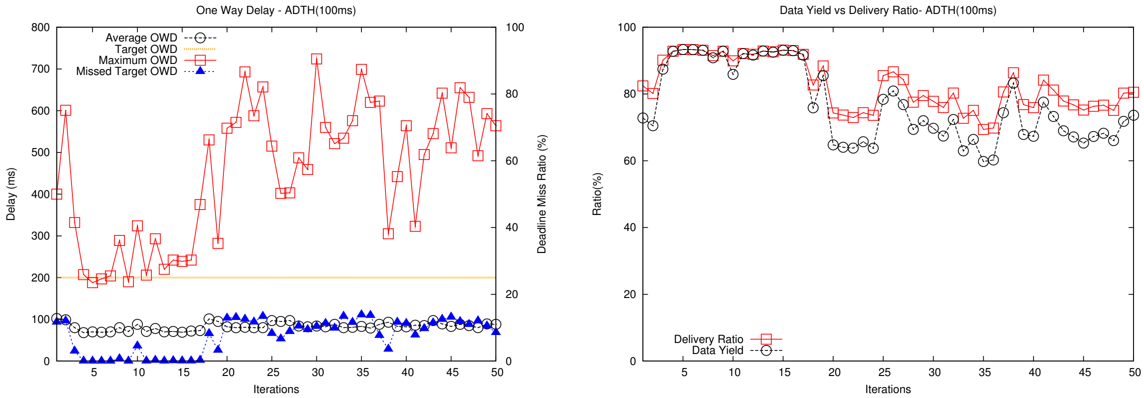


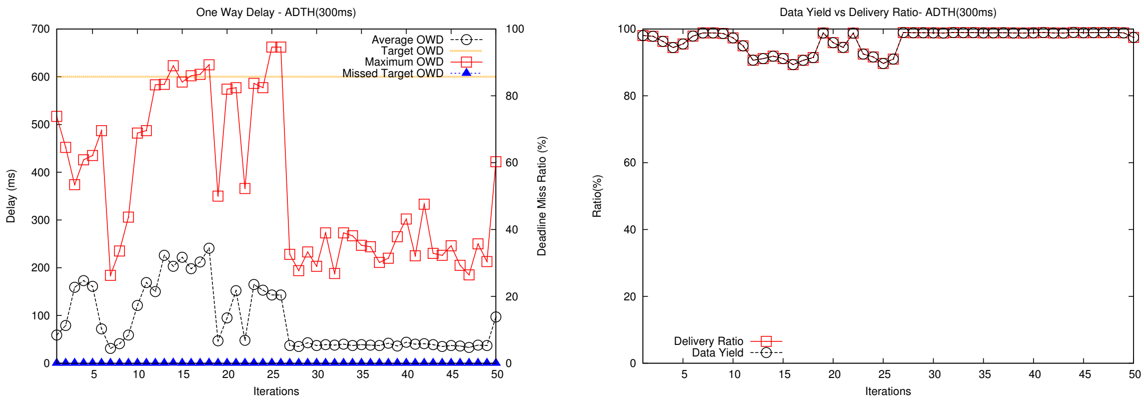
Figure 6.22: Experiment 3: Overall Packet Loss Ratio Trend (ADTH versus pfifo)

By examining the packet loss ratio trend (Fig. 6.22), it can be seen that the packet loss ratio for ADTH is ±8% when compared to pfifo. For most of the cases, packet loss ratio is higher for ADTH and the trend is supposed decreasing when the delay requirement is relaxed. However, not only the delay requirement has impact on packet loss. There are other factors, such as environmental interference, network dynamics caused by random traffic load and packet sizes, and network dynamics caused by ADTH due to the regulation of queue threshold, may change the contention level in the network. Therefore, the packet loss ratio for ADTH is lower than pfifo when $D_{Nr} = 300ms$ and $500ms$ and not abide to the factor caused by relaxation of delay requirement.

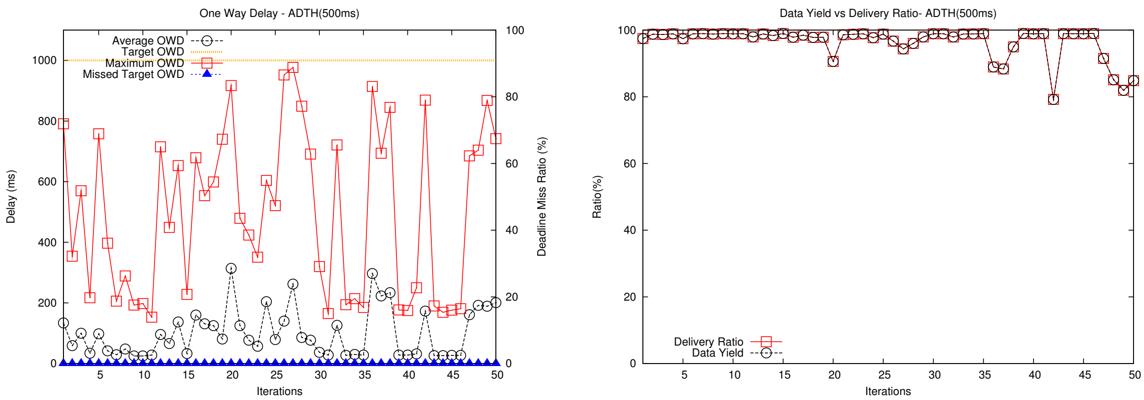
The observation above can be further supported by taking snapshots into the performance results of each run for different configurations (Fig. 6.23). For the first few runs of experiments with $D_{Nr} = 100ms$, the maximum OWD recorded is lower for the repetition from 3 to 18. The deadline miss ratio is low for these repetitions. The variation of interference contributes to the variation of the performance. However, the average OWD does not fluctuate much, as ADTH regulates nodal delay of each node actively. When the network quality is lower due to higher contention and interference, the data yield is lower than the packet delivery ratio resulting from a higher deadline miss ratio. Similarly for the case when $D_{Nr} = 300ms$, the contention level and interference level are lower after 27 runs; thus packet loss decreases as observed in Fig. 6.22. It is clearly shown in Fig. 6.23 that packet delivery ratio becomes higher when D_{Nr} is higher.



(a) ADTH ($D_{Nr} = 100ms$)



(b) ADTH ($D_{Nr} = 300ms$)



(c) ADTH ($D_{Nr} = 500ms$)

Figure 6.23: Experiment 3: OWD with Deadline Miss Ratio Trend and Data Yield for ADTH @ $D_{Nr} = 100ms$, $300ms$, $500ms$

The packet delivery ratio is high for pfifo, but the data yield is extremely low (Fig. 6.24). The data yield ranges from 12% to 23% when D_{Nr} is increased from $100ms$ to $500ms$. While for ADTH, the data yield is quite closed to the packet delivery ratio for all cases (Fig. 6.24). For the cases of $D_{Nr} = 300ms$ and $500ms$, the packet delivery ratios are higher than pfifo. ADTH has reduced the contention level indirectly by adapting the target queue threshold dynamically and enabled early dropping of packets to reduce bandwidth wastage.

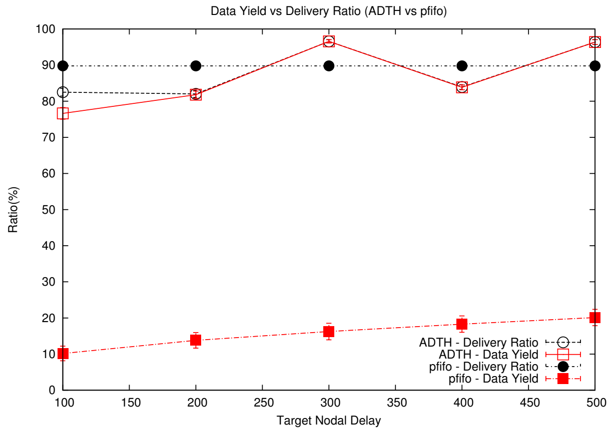


Figure 6.24: Experiment 3: Overall Data Yield Trend (ADTH versus pfifo)

6.4.4 Experiment 4: Minimum target nodal delay

The experiments above have shown that ADTH can adapt to the dynamic changes in the network effectively. However, there is a limitation of ADTH as stated in Section 5.4 that the target nodal delay cannot be too stringent. When the network contention is very high, the nodes compete for the channel access for packet transmission. High collisions will occur and the nodes need to backoff randomly and frequently. This causes MAC layer delay becomes large and potentially larger than D_{Nr} if D_{Nr} is set too low. Therefore, ADTH will fail to constrain the nodal delay within the range required. D_{Nr} is set to the lowest bound of the ADTH implementation that is $20ms$; while the sampling interval is also set to the minimum value $10ms$. OWD_R is $40ms$ in this case. Bi-directional traffic is generated from STC at the rate of $1.8Mbps$ with random packet sizes.

Under such a stringent delay requirement and yet the network is saturated with high traffic load, ADTH is unable to bound nodal delay to the required value (Fig. 6.25). The average deadline miss ratio is as high as 99.13%. This is because the MAC layer delay alone has exceeded D_{Nr}. When the contention is high, the nodes need to wait for the channel becomes idle before can start transmission. Packets may collide and be discarded. This causes the nodes to backoff with random waiting time [25,79]. The waiting time is randomly chosen from the range of 1 to CW in unit of time slot with each time slot equal to $20\mu s$. The minimum CW is 31 and the maximum CW is 1023. The CW is increased exponentially each time the nodes collide and reset after a packet is transmitted successfully. Therefore, the random backoff time can be high if the maximum waiting time is chosen each time the node backoff. For each packet transmitted, the node will wait to get an acknowledgement (ACK) if it is not a broadcast packet. If ACK is not received, the packet will be retransmitted until reaches the retry limit. The retries limit is default to 8 times for Gumstix. MAC layer delay can be large for a saturated or lossy network due to the retransmissions. Therefore, this explains why ADTH is not able to bound the delay at a high traffic load.

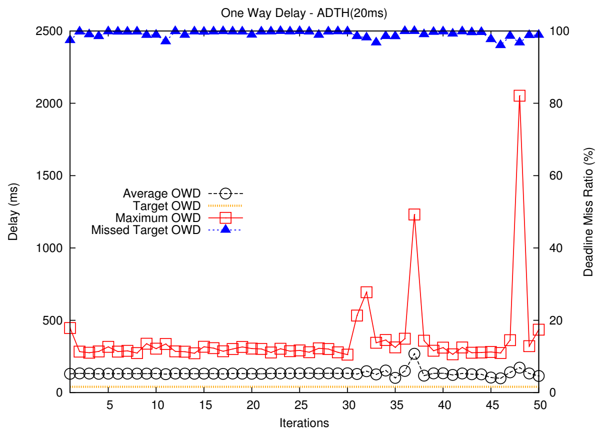


Figure 6.25: Experiment 4: OWD Trend versus Deadline Miss Ratio for ADTH ($D_{Nr} = 20ms$)

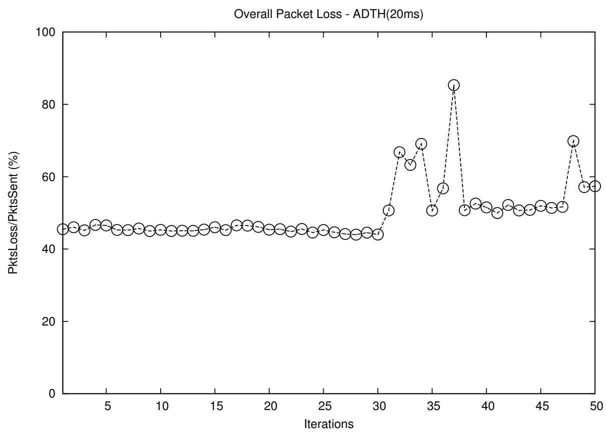


Figure 6.26: Experiment 4: Packet Loss Ratio Trend for ADTH ($D_{Nr} = 20ms$)

Although ADTH cannot constrain nodal delay to the target nodal delay here due to an unrealistic delay requirement; the nodal delay is still constrained to a lower possible value with higher packet loss as a trade-off. ADTH has shrunk the queue size aggressively with its best attempt to meet the delay requirement. The mean of average OWD recorded is $133ms$ with SD $22ms$. While the mean of maximum OWD is $373ms$. By comparing the packet loss ratio (Fig. 6.26) to the packet loss ratio in Experiment 2 at the same traffic load, it can be seen that a more stringent delay requirement contributes to higher packet loss ratio under the same experiment configurations. The packet loss ratio in this experiment is 49.99% with SD 8.22% compared to 38.11% with SD 1.08% in Experiment 2.

The packet delivery ratio is low (average 50.01%) and the data yield is extremely low which is $< 1\%$ (Fig. 6.27) due to high packet loss ratio and deadline miss ratio. Therefore, the delay requirement cannot be too stringent and unrealistic.

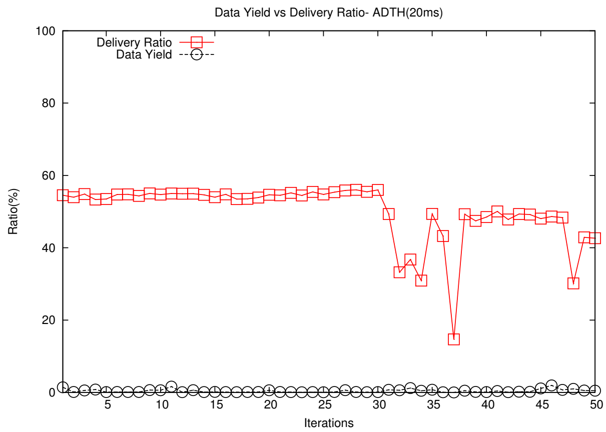
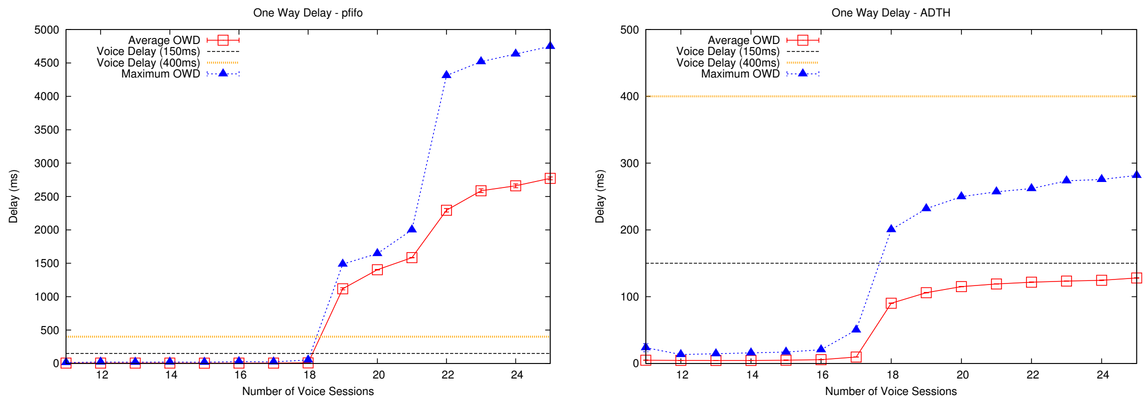


Figure 6.27: Experiment 4: Data Yield versus Packet Delivery Ratio for ADTH ($D_{Nr} = 20ms$)

6.4.5 Experiment 5: End-to-End Delay for Voice Data

According to ITU-T G.114 [2], the acceptable delay for conversational voice application is preferably $< 150ms$ and the limit is $< 400ms$. The impacts of ADTH and pfifo on OWD and packet loss ratio for voice sessions have been investigated in this experiment. A voice session is a two-way communication; therefore bidirectional voice traffic is generated. CBR traffic is used to represent the voice traffic to simplified the experiment setup. The experiment has been carried out with incremental number of concurrent voice sessions ranging from 11 to 25 sessions. Each voice traffic flow consists of 20 packets per second based on G.711 codec rate and the packet size is $218B$. ADTH parameters are configured as: D_{Nr} $100ms$ and $t_s = \frac{1}{4}D_{Nr}$. End user QoS or the application layer QoS is not investigated since the focus of the thesis is on network delay control and also the test setup is simplified with no involvement of application layers.

The experiment results show that when the number of concurrent voice calls surpasses the network capacity; all voice flows suffer high end-to-end delay for pfifo queue discipline (Fig. 6.28(a)). The trend of OWD for the voice sessions has changed drastically when the network is congested. A sharp turning point occurs when the number of voice session is increased from 18 to 19 sessions. The average OWD recorded is greater than $1s$. The magnitude of OWD recorded is unacceptable for conversational voice sessions. The OWD continue to rise with the incremental of voice sessions and can be $> 2.5s$ for the average and $> 4.5s$ for the maximum when the number of voice sessions is 25.



(a) pfifo (b) ADTH

Figure 6.28: Experiment 5: Overall OWD Trend (ADTH versus pfifo)

When ADTH queue discipline is adopted, the OWD is controlled within the acceptable range for the conversational voice traffic (Fig. 6.28(b)). Overall, the average OWDs recorded are $< 150ms$ for all cases and the maximum OWDs recorded are $< 400ms$. This shows that ADTH has improved the network performance significantly in terms of network latency. However, the improvement comes at the expense of a higher packet loss ratio (Fig. 6.29).

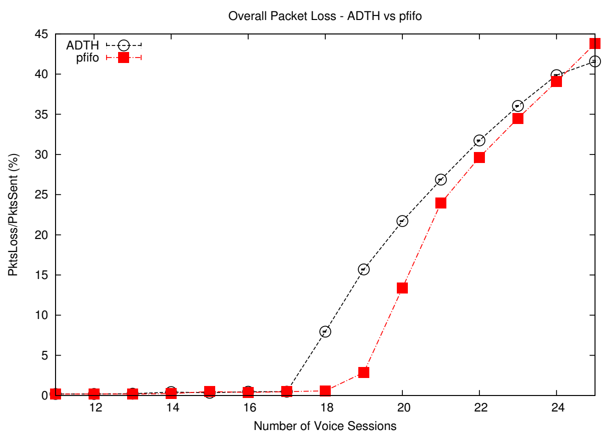


Figure 6.29: Experiment 5: Overall Packet Loss Ratio Trend (ADTH versus pfifo)

The turning point of packet loss ratio is at 17 sessions for ADTH, but at 18 sessions for pfifo. ADTH starts to regulate the queue size to bound nodal delay when congestion is detected. Consequently, packets are dropped aggressively in order to meet the per-hop delay requirement. According to ITU-T G.1010 [81], conversational voice traffic is sensitive to packet loss and preferably to have $<$ 3% loss ratio. Although ADTH can constrain the network delay, it could not constrain the packet loss. This is a known limitation of ADTH design. A proper admission control is needed to deal with this issue. Fig. 6.30 shows the ratio of packets received at the destination per delay requirements in ITU-T G.114. pfifo performs badly when the network is congested.

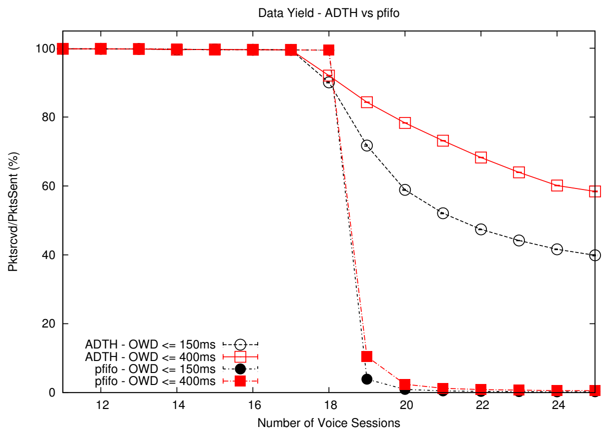


Figure 6.30: Experiment 5: Data Yield against Delay Requirements of ITU-T G.114 (ADTH versus pfifo)

Nevertheless, if an application is not sensitive to packet loss ratio then ADTH is a perfect candidate for it. Even ADTH fails to constrain the packet loss ratio, the queue fullness and the packet loss ratio can be good indicators of network congestion. Aggressive packet dropping is due to fulfilment of delay constraint and queue overflow. Therefore, these indicators can be used in admission control process for decision-making. pfifo cannot provide these indicators as the delay is constrained by a constant maximum queue size. When the queue overflows, it is too late for the admission control or applications to react on the congestion.

6.4.6 Experiment 6: Scalability

Three more wireless nodes are added to form a bigger wireless ad hoc network in this experiment. The intention of this experiment is to show that ADTH is a scalable and standalone scheme. All wireless nodes are crammed into a limited space in the laboratory. A high contention network is created as all nodes are within the transmission range of each other. However, the network has been setup as a multi-hop network that the nodes cannot reach all other nodes directly. Fig. 6.31 shows the setup for this experiment. Fig. 6.32 shows that there are 5 hops in total from WN1 (IP: 192.168.2.200 / 192.168.10.1) to WN2 (IP: 192.168.2.202 / 192.168.20.1) and vice versa. WN3 (IP: 192.168.2.201), WN4 (IP:

192.168.2.203), WN5 (IP: 192.168.2.204) and WN6 (IP: 192.168.2.205) are intermediate hops between WN1 and WN2. Bi-directional traffic consists of random sized packets (ranges from $128B$ to $1518B$) with a random load (ranges from $0.2Mbps$ to $0.8Mbps$) are sent to WN1 and WN2.

Figure 6.31: Wireless Ad Hoc Network with Six Wireless Nodes

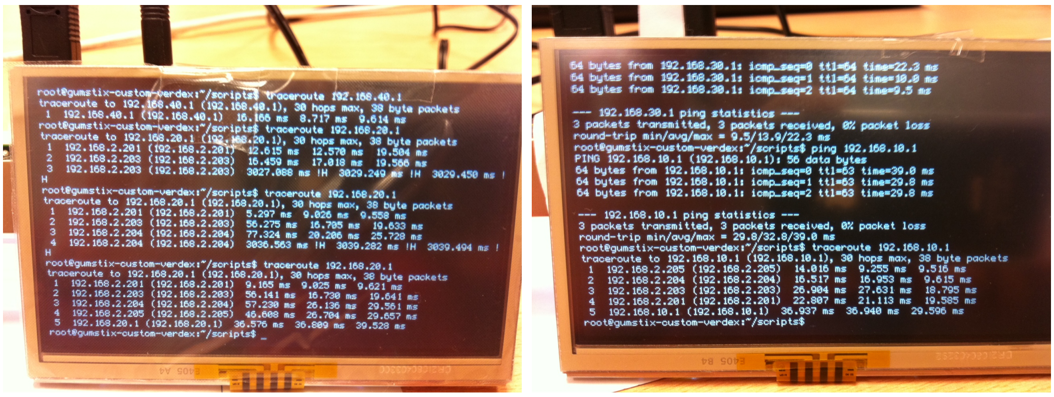
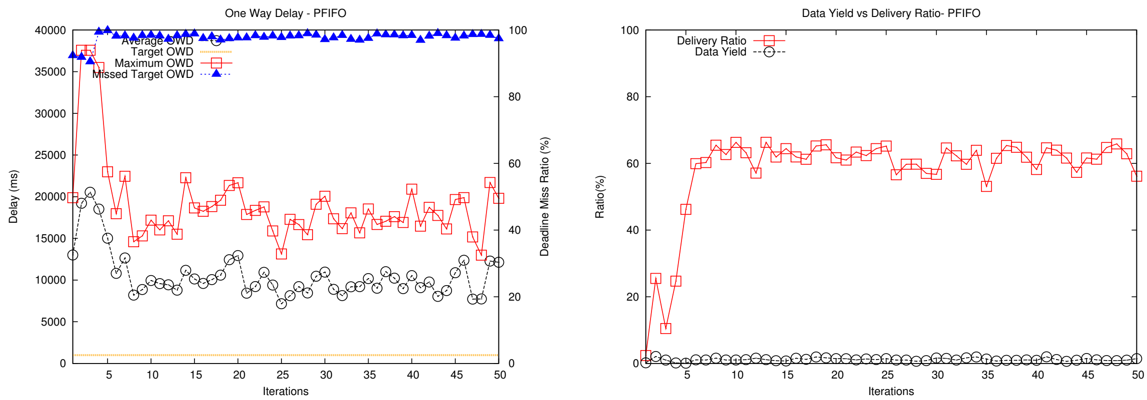


Figure 6.32: Traceroute from WN1 to WN2 and vice versa

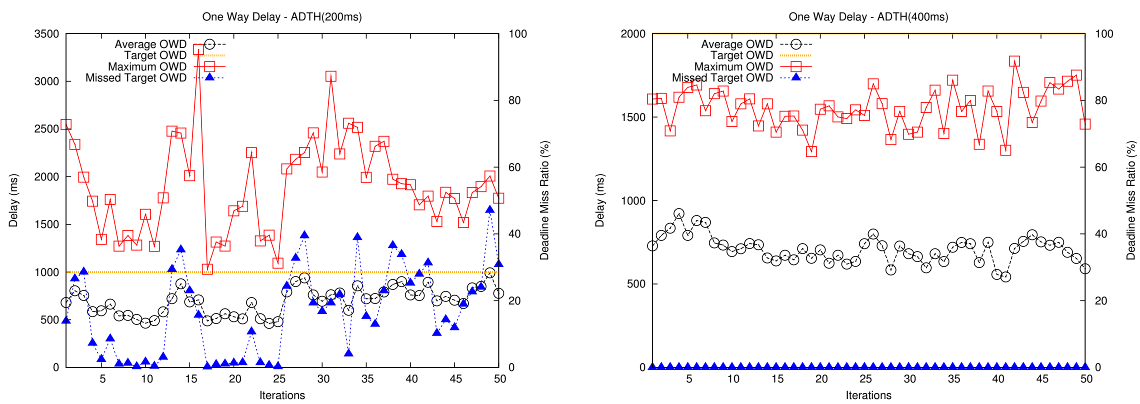
The network contention is very high in this experiment, all nodes interfere each other and compete for the channel access to transmit packets. Therefore, the average OWD and the maximum OWD recorded for pfifo are very large (Fig. 6.33(a)). The mean of average OWD is $10.56s$ with SD $2.79s$; while the mean of the maximum OWD is $19.13s$ with SD $5.08s$. There are a few runs that the maximum OWD are $> 20s$ and as high as $37s$. The OWD of packets is definitely unacceptable for delay-sensitive traffic. If the required nodal delay of 200ms, then the deadline miss ratio is 97.73% for pfifo. This results in extremely low data yield even though the mean of packet delivery ratio is 57.89% (Fig. 6.33(b)).



(a) OWD with Deadline Miss Ratio Trend (b) Data Yield versus Data Delivery Ratio

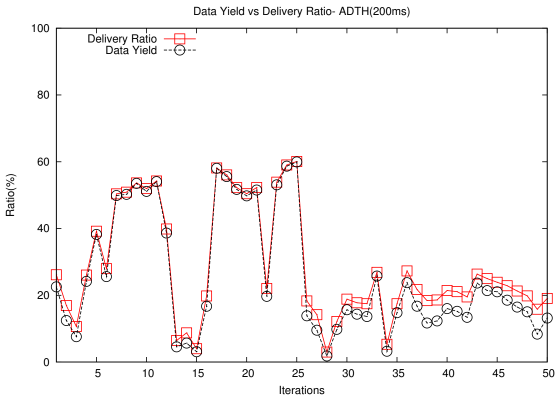
Figure 6.33: Experiment 6: System Performance for pfifo (5 hops)

The experiment is repeated with $D_{Nr} = 200ms$ and $400ms$ for ADTH. The sampling interval for both cases is set to $\frac{1}{4}D_{Nr}$. For the case of $D_{Nr} = 200ms$, ADTH is able to bound network delay but not very efficient. This is because the network contention level is too high and MAC layer delay can be large at each node. The deadline miss ratio is average 16.37% with SD 13.24% (Fig. 6.34(a)). Nevertheless, the deadline miss ratio is still much lower than pfifo and the average OWD recorded is $< OWD_R$. When the delay requirement is changed to $400ms$, ADTH can constrain nodal delay and OWD effectively with the maximum OWD $<$ OWD_R for all 50 runs. Consequently, the deadline miss ratio is 0% (Fig. 6.34(b)).

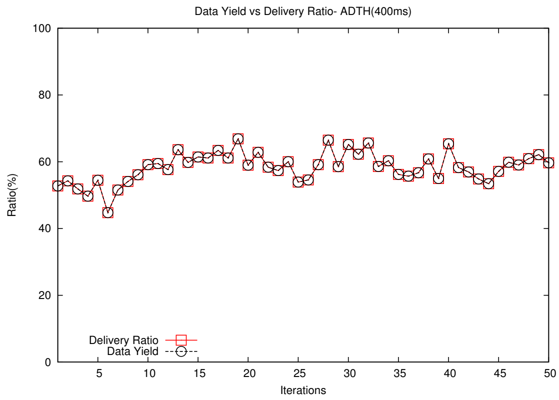


(a) $D_{Nr} = 200ms$, $OWD_R = 1000ms$ (b) $D_{Nr} = 400ms$, $OWD_R = 2000ms$

Figure 6.34: Experiment 6: OWD Trend for ADTH (5 hops)



(a) $D_{Nr} = 200ms$, $OWD_R = 1000ms$



(b) $D_{Nr} = 400ms$, $OWD_R = 2000ms$

Figure 6.35: Experiment 6: Data Yield versus Data Delivery Trend for ADTH (5 hops)

Besides, the system performance fluctuates a lot when $D_{Nr} = 200ms$ (Figs. 6.34(a) and 6.35(a)). This is because ADTH regulates the queue actively and drops packets aggressively for a smaller delay requirement. Suppression of queue in one node may lead to lower queuing delay of its next hops as the packets are dropped instead of being passed on to next hops. Other than that, the fluctuation is highly dependent on interference and contention level in the network. If MAC layer delay is large, it leaves lesser rooms for ADTH to regulate the queue as queuing delay must be small to compensate for large MAC layer delay. Thus, only a few packets can be buffered in this case. The data yield of ADTH (Fig. 6.35) is the same as the packet delivery ratio for the case $D_{Nr} = 400ms$, whereas the data yield is slightly lower than the packet delivery ratio for $D_{Nr} = 200ms$.

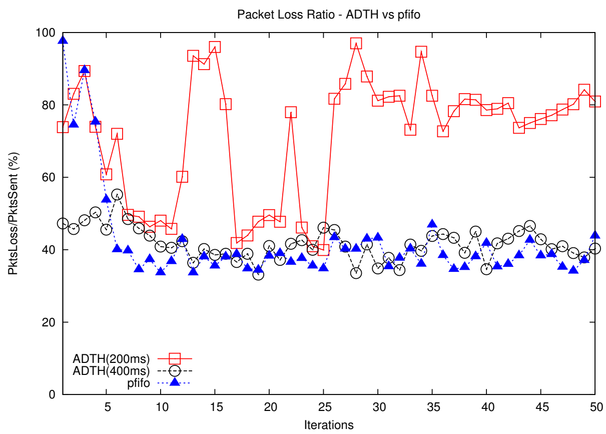


Figure 6.36: Experiment 6: Packet Loss Ratio for ADTH versus pfifo (5 hops)

The packet loss ratio is then compared against pfifo and ADTH in Fig. 6.36. The packet loss ratio for ADTH is higher than pfifo especially when $D_{Nr} = 200ms$. Whereas when $D_{Nr} = 400ms$, the packet loss ratio of ADTH is similar to pfifo which are average 41.65% and 42.11% respectively. However, the average OWD for ADTH is only $707ms$ compared to $10.56s$ for pfifo. The average OWD is nearly 15x smaller than the OWD observed in pfifo.

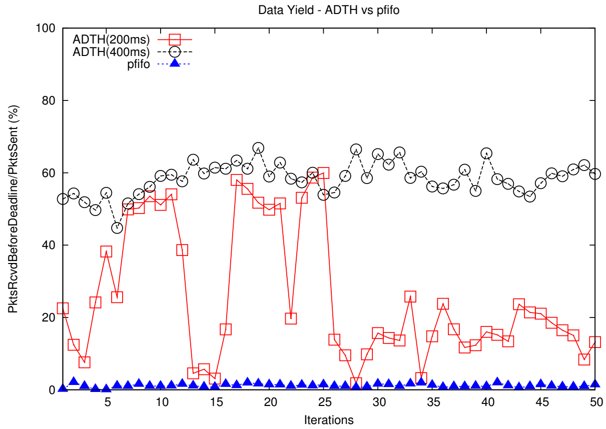


Figure 6.37: Experiment 6: Data Yield for ADTH versus pfifo (5 hops)

The efficiency of ADTH is highlighted by comparing the data yield of pfifo and ADTH (Fig. 6.37). The data yield of pfifo is <2%. Although the performance fluctuates for the case of $D_{Nr} = 200ms$, the average of data yield is 25.71%. A very promising result is shown by ADTH when $D_{Nr} = 400ms$, the achievable of data yield is average 58.35%.

In conclusion, ADTH is able to constrain network delay regardless of number of hops in between a source and a destination based on the results shown in this experiment and also previous experiments as long as the target nodal delay required is within the reasonable range.

6.5 Summary and Discussions

ADTH has been validated on a small wireless ad hoc network. Gumstix nodes are setup as wireless nodes in the testbed, the experiments have been carried out with 3 wireless nodes for the validation and performance analysis of ADTH against pfifo. This proves that ADTH is feasible to be implemented on real wireless nodes. It can be adopted widely as a generic queuing discipline for network nodes to constrain per-hop nodal delay. The processing overhead incurred by the ADTH controller is low and has minimal impact to the system performance. The wireless ad hoc network has been expanded to become a six nodes network in the last experiment to show that ADTH is scalable.

ADTH is a generic and standalone queuing discipline that does not require gathering information from neighbour nodes or from cross layers within the node to to bound nodal delay. The implementation of ADTH only involves changes at the network layer to add in ADTH as a new queue discipline and changes at user space utilities to configure the ADTH queue discipline. The main parameters to be configured for ADTH from the user space are target nodal delay (D_{Nr}) and sampling interval (t_s). Based on the experiment results, the sampling interval is recommended to abide to the rule of $t_s \leqslant \frac{1}{4}D_{Nr}$ in order to improve the efficiency of ADTH. This is because interference is greater in the real world as compared to the simulation. Therefore, a higher sampling frequency is needed for the ADTH controller to enable ADTH to response faster to network changes. ADTH can adapt to network dynamics autonomously in real testbed environment even the network contention level and the link quality vary over time.

The experiment results show that ADTH is able to constrain network delay effectively via the per-hop queue regulation mechanism. The trade-off between higher packet loss ratio and lower network delay is expected with ADTH enabled when the network is severely congested or the target nodal delay requirement is stringent. Nevertheless, the loss ratio is only higher by average of 3% - 5% and yet the network delay is lower by a factor of X depending on the target nodal delay requirement and the network contention level. More importantly, wastage of bandwidth and resources are reduced as a result of early dropping of packets that potentially reach destination after the deadline. The packet loss ratio could be better with ADTH under certain circumstances (e.g. network is moderately congested) as more bandwidth can be used to transmit useful packets. The effect of early discard may reduce the network contention level in the network.

ADTH guarantees per-hop nodal delay with the assumption that the target nodal delay requirement is not too stringent. ADTH cannot provide guarantee on end-to-end delay without the assistance of QoS-aware routing to discover appropriate routes that can satisfy the specified end-to-end delay requirement. However, the deterministic per-hop latency characteristic enables a deterministic end-to-end latency to be achieved. This may facilitate the QoS-aware routing protocols to discover routes easily without the need of latency estimation from intermediate nodes during routes discovery.

Processing delay for packets receiving and routing at network layer are assumed to be minimal for the ADTH implementation judging from the capacity of the processor speed. While the upper layer processing delay is considered as application layer overhead and not being taken into account by ADTH for nodal delay regulation. If a wireless ad hoc network is setup as a secured network, packets are encrypted before being sent out and decrypted after they are received at the destination. The processing delay can be quite large especially when the nodes perform complex encryption algorithms and examine or modify packet content [146]. The processing delay can be safely ignored if these operations are carried out at the sources and destinations, as this overhead will not affect the network delay of other packets directly. This overhead is considered as application layer overhead. However, if the complex operations are carried out at the intermediate nodes that act as gateways to the sources and the destinations; then the processing overhead should be taken care of at the gateways to ensure nodal delays is bounded. Therefore, ADTH can be enhanced to include such type of processing overhead in future work for the target queue threshold estimation.

Chapter 7

ADTH Applications and Implications in E2E QoS

7.1 Introduction

This chapter presents potential application scenarios to show how can ADTH contribute to end-to-end delay and end-to-end QoS guarantee when combined with other QoS schemes. In depth discussion on these QoS schemes are not presented here, a comprehensive survey on each of the proposed application domains and its feasibility are left for future work. However, a brief overview of these QoS schemes can be found in Chapter 2.

For delay-sensitive applications, such as interactive online gaming, media streaming, Internet telephony, surveillance, etc., timeliness in sending and receiving data are very important to make sure the QoS perceived is within a tolerable range. Bounded end-to-end delay and a tolerable packet loss ratio are two key metrics in guaranteeing the QoS for these applications. From the results achieved in this thesis, it has been shown that ADTH is effective in constraining nodal delay. A deterministic per-hop delay can be achieved with the adoption of ADTH for transmission queue of network interfaces provided that the nodal delay requirement is reasonable. Therefore, the ADTH queue discipline should be adopted by network nodes which need to transfer delay-sensitive data.

ADTH could not guarantee a deterministic end-to-end delay on its own. It is also not able to alleviate congestion and control the traffic load on its own. However, the internal variables and parameters of ADTH can be manipulated to facilitate other QoS schemes to achieve the goals. ADTH can be combined with other QoS schemes, such as QoS-aware routing, admission control and service differentiation, to provide solutions for delay-sensitive applications. Fig. 7.1 shows an overview of the ADTH queue discipline with its parameters and internal variables listed. These parameters and internal variables can be exploited to facilitate decision-making and operations of other QoS Schemes.

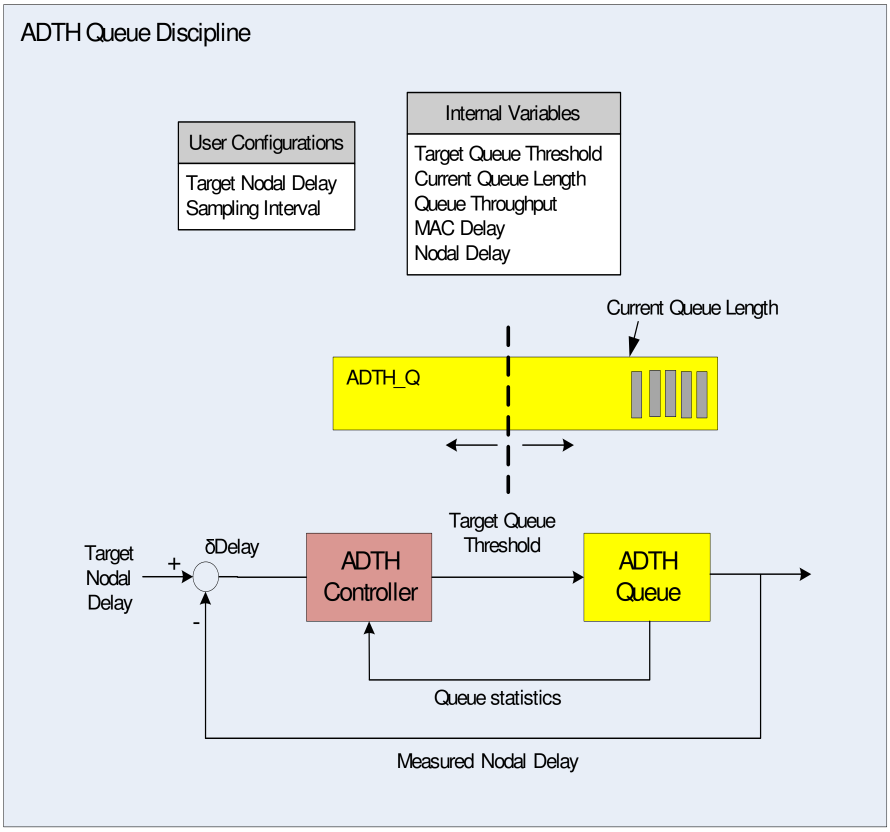


Figure 7.1: ADTH Queue Discipline Overview

The remainder of the chapter is organized as follows: Section 7.2 shows that ADTH can be coupled with QoS-aware routing to provide end-to-end delay guarantee; Section 7.3 describes how can ADTH be utilized for admission control; Section 7.4 describes how can changes of network conditions be reflected via the ADTH internal status; Section 7.5 shows the application of ADTH with priority queuing in order to provide service differentiation for delay-sensitive and delay-tolerant traffic; Section 7.6 describes a way of utilizing the ADTH internal queue status to rate-limit delay-tolerant traffic while maintaining delay for delay-sensitive traffic with single transmission queue; finally a summary is given in Section 7.7.

7.2 ADTH with QoS-aware Routing

Traditionally, the shortest path is selected as the route for a particular flow from a source to a destination. With the emerging of wireless ad hoc networks, QoS-aware routing protocols have been invented to discover appropriate paths that can satisfy QoS requirements. QoS metrics, such as throughput, loss ratio, delay and jitter are commonly used for route discovery and decision-making.

A delay-based routing protocol discovers paths that can satisfy the end-to-end delay requirement for a particular flow or an application. Most of the delay-based routing protocols [32, 82, 122, 131, 158, 181] use delay estimation and packet probing method to decide the selection of paths. However, end-to-end delay for the paths chose may vary over time. The delay estimated or probed during the route discovery process may not hold after the paths being established due to network dynamics resulting from environmental interference, link quality, medium contention, etc. ADTH can facilitate the route discovery process and also ensure the end-to-end delay requirements are met after the paths are established. Therefore, nodes with ADTH support should be chosen in the route discovery process to enable deterministic end-to-end delay to be achieved.

The target nodal delay parameter of ADTH can be used in the route discovery process to discover paths that can meet the end-to-end delay requirements. The delay estimation or delay probing method can be replaced by aggregating the target nodal delay of nodes from a source to a destination. The selection of path is then decided by comparing the aggregated target nodal delay to the delay constraint. A simple wireless ad hoc network shown in Fig. 7.2 is used to illustrate the proposed application scenario of ADTH in delay-based routing protocol. The wireless ad hoc network consists of eight wireless nodes with only six of them have ADTH support. For those wireless nodes with ADTH support, the target nodal delay has been configured for the ADTH queue discipline. For legacy wireless nodes without ADTH support, nodal delay is not bounded.

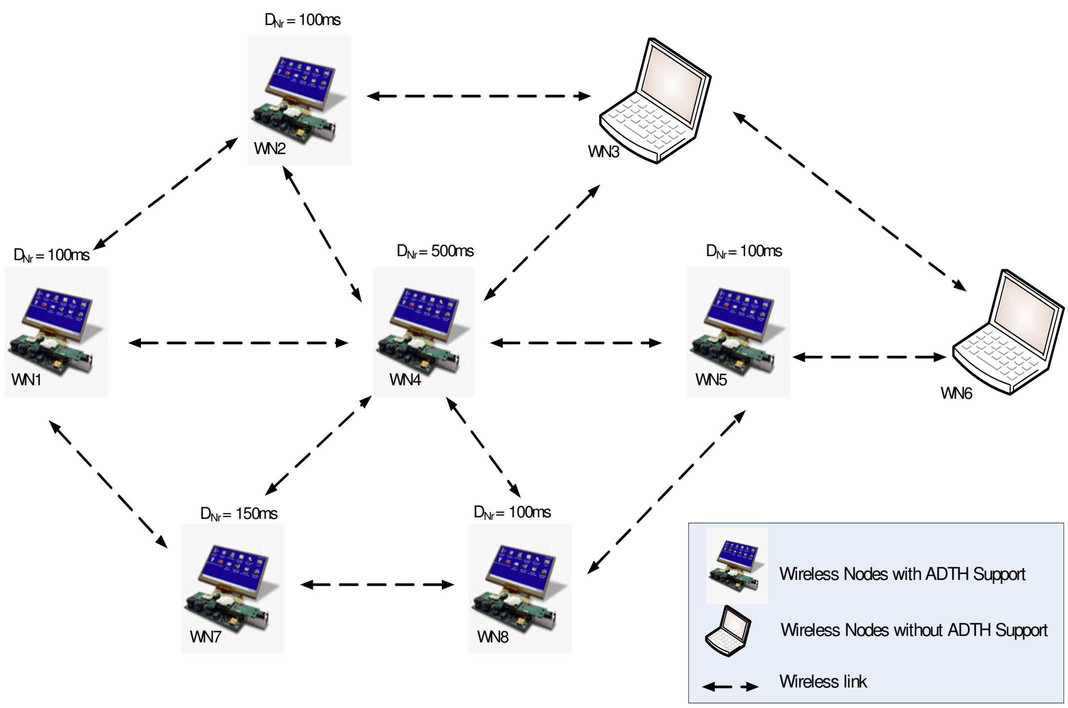


Figure 7.2: Example of a Wireless Ad Hoc Network Topology

For example, an application on WN1 would like to transfer delay-sensitive data to WN6 with a requirement of $OWD_R \leqslant 500ms$. At least one route needs to be established before WN1 can communicate to WN6 if no static routes exist between WN1 and WN6. WN6 can be reached via multiple paths from WN1 with the assumption that all nodes are willing to forward packets for others. So a few possible routes are picked for the discussion here, the aggregated delay for the paths are listed below:

- Estimated OWD (Path 1: WN1→WN2→WN3→WN6)

  $= 100ms + 100ms +$ unknown

- Estimated OWD (Path 2: WN1→WN4→WN5→WN6)

  $= 100ms + 500ms + 100ms = 700ms$

- Estimated OWD (Path 3: WN1→WN7→WN8→WN5→WN6)

  $= 100ms + 150ms + 100ms = 450ms$

From the estimated OWD above, path 3 will be selected as it can fulfil the requirement of $OWD_R \leqslant 500ms$ even though the destination is 4 hops away from the source. Path 1 is not recommended, as WN3 has no constraint on nodal delay, therefore OWD can be greater than OWD_R when the network is congested or the link quality is poor. Large nodal delay may be observed at WN3.

For another example, if the requirement of OWD_R from WN2-$>$WN7 is $\leqslant 800ms$; both the paths of WN2→WN1→WN7 and WN2→WN4→WN7 can satisfy the delay requirement. WN2→WN1→WN7 may be preferred in this case due to smaller bounded OWD.

If there exists a few paths with the same estimated OWD from a source to a destination that can meet the end-to-end delay requirement; other internal variables of ADTH can be made visible to the routing protocol so that a better route can be chosen. Such decision-making can base on backlog level in a queue (current queue length), throughput of a queue, MAC layer delay or nodal delay sampled in the ADTH controller.

In conclusion, ADTH can be coupled with delay-based routing protocols to ensure the end-to-end delay requirements for delay-sensitive applications are met.

7.3 ADTH with Admission Control

Although one or more routes may exist for the transfer of delay-sensitive traffic over a network; by satisfying just the delay constraint is not sufficient for certain delay-sensitive traffic, such as VoIP. Conversational traffic is sensitive to packet loss. ADTH can only constrain nodal delay but not packet loss. Packet loss at nodes may be caused by queue overflow and retransmissions at MAC layer. Queue overflow happens when a node receives packets from applications or neighbour nodes faster than the rate it can transmit or forward the packets, so the queue backlog exceeds the queue limit. Packet loss also happens after the maximum retries at MAC layer to retransmit a particular packet failed.

Admission control based on the current network load or network contention level should be employed to mitigate this issue. The network load and network contention level can be reflected from the internal states of a ADTH queue. Packets are backlogged in a queue before the queue overflows. Queue throughput is low, MAC layer delay and nodal delay are high when the network load or the contention level is high. This implies that the admission control scheme can poll the ADTH internal states during decision-making process to decide the admission of new flows into the network.

An admission control threshold which varies based on the target queue threshold can be introduced as a decision point to prevent packet loss caused by queue overflow. The target queue threshold is adjusted adaptively by ADTH to constrain nodal delay. Therefore, the admission control threshold should be movable also following the adjustment of target queue threshold (Fig. 7.3). For example, if the admission control threshold is set to 80% of the target queue threshold; then the admission control scheme should reject a flow when the current queue length exceeds 80% of target queue threshold. It can either be perceived as the node almost hits its capacity in constraining nodal delay within the delay requirement or packet loss is to be triggered as the queue capacity is almost full.

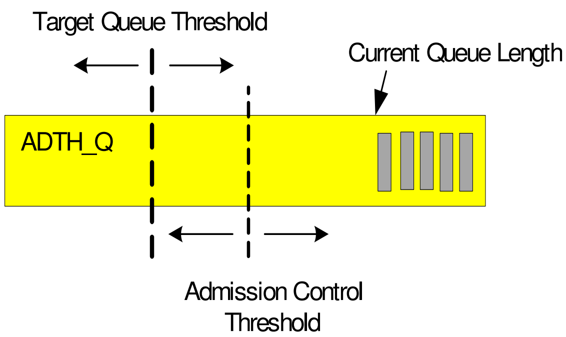


Figure 7.3: ADTH with Admission Control Threshold

Furthermore, the mean MAC layer delay sampled by the ADTH controller can also be exploited for decision-making process in the admission control scheme. The severeness of network contention and link quality can be reflected by the MAC layer delay measurement. If the channel is busy or the link quality is poor; it will take longer time for a node to transmit a packet successfully due to a longer random backoff period and more retries to retransmit a packet. Without the need to probe into MAC layer to measure the channel busy ratio or available bandwidth; an admission control profile can be built based on the trend of MAC layer delay measurement or the magnitude of MAC layer delay. Therefore, a node may make a better decision to admit or reject a flow based on this profile in addition to the admission control threshold discussed above.

With these exploitations, a contention-aware and yet delay-aware admission control can be designed to take care the trade-off between packet loss and maximum delay experienced for a particular flow.

7.4 ADTH with Explicit Notifications on Network Conditions

Delay-sensitive traffic (e.g. UDP traffic) is generally unable to respond to packet loss events. Therefore, traffic sources continue to inject the same amount of traffic generated from applications into a network without knowing the network conditions. This may exacerbate the severeness of network contention and network congestion. Other than that, the network conditions vary over time due to interference, degradation of link quality, etc. Hence, an explicit notification mechanism can be established to inform the sources on the network changes or when the delay constraint can merely be met.

Link quality deterioration or throughput degradation due to interference and contention may be reflected by the status of a ADTH queue implicitly. When the link quality degrades or the network contention is more severe, backlog in the queue grows rapidly as a result of higher collisions and retransmissions with the assumption that the arrival of packets into the ADTH queue is roughly the same or even higher. Therefore, an appropriate notification to the sources is needed in order to regulate their sending rates or trigger route maintenance to look for alternative paths. In this scenario, the number of packets in the queue against the ADTH target queue threshold can be used as an indicator for such changes. A queue backlog threshold can be used to decide when to trigger the notification process. When the queue backlog exceeds the queue backlog threshold, then the explicit queue backlog indicator must be sent to the sources. A signalling mechanism is needed since that there is no return path (acknowledgment packets) to the sources for UDP packets.

Instead of using a single queue backlog threshold, multiple thresholds can be used to indicate different level of backlogs in the queue. Different level of backlogs in the queue against the target queue threshold can be taken as congestion level in the network. The idea of multi-level ECN notification for TCP traffic [50] can be borrowed here for delay-sensitive traffic. For multi-level ECN, the congestion level is estimated based on RED thresholds. In this case, the thresholds for queue backlogs level can be set with respect to the target queue threshold. For example $TH_i = \{\frac{1}{4}L_Q, \frac{1}{2}L_Q, \frac{3}{4}L_Q\}$ as shown in Fig. 7.4.

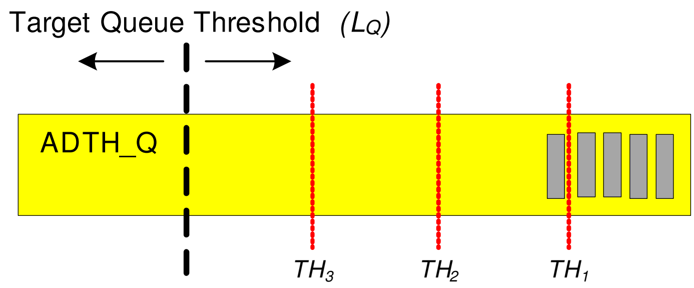


Figure 7.4: ADTH with Multiple Thresholds for Network Conditions Notification

Other than the level of queue backlogs at a node, MAC layer delay or nodal delay measured in ADTH can be used to monitor if the delay requirements can be met after the network deployment. If MAC layer delay becomes very large and causes the delay constraint to be violated, then the signalling process can be used to inform the delay-sensitive sources to look for alternative paths.

With the explicit notification mechanism proposed above, the network congestion issue could be mitigated. This also enables the network to provide a proactive QoS provisioning with anticipation of network changes to assure end-to-end QoS for delay-sensitive applications.

7.5 ADTH with Priority Queuing

Most of the time, the traffic flows in a network consists of delay-sensitive traffic and delay-tolerant traffic. Therefore, delay-sensitive traffic may be impacted by delay-tolerant traffic. In order to provide better QoS and to guarantee end-to-end delay of delay-sensitive traffic; packets from these traffic classes need to be relayed to their destinations as soon as possible. Therefore, delay-sensitive traffic and delay-tolerant traffic should be separated into different transmission queue with the highest priority given to delay-sensitive traffic [112, 187].

However, maintaining a separate queue for delay-sensitive traffic with the highest priority does not bound end-to-end delay or nodal delay. The priority queuing mechanism can only guarantee service differentiation to provide a better service for delay-sensitive traffic. Hence, the ADTH mechanism can be applied to the priority queue which carries delay-sensitive traffic (Fig. 7.5) to bound nodal delay assuming that priority 0 is the highest priority. However, the lower priorities queues may be starved by higher priority queues in this case. Excessive packets dropping will happen for those starved queues when the higher priority queues are never empty. This is a known issue of priority queuing, it is not related to the ADTH application here.

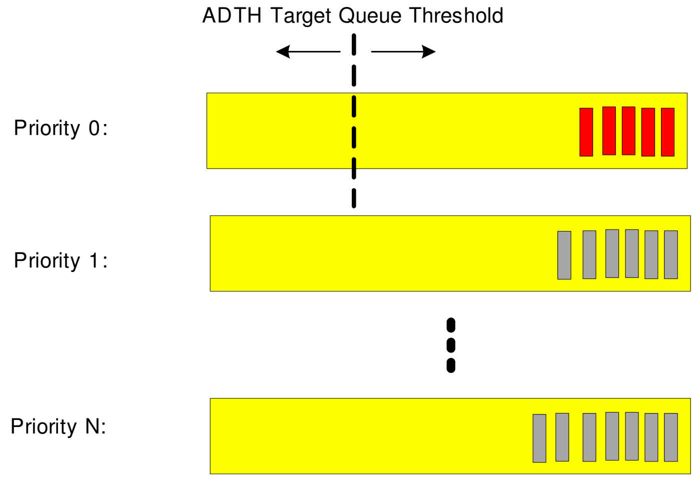


Figure 7.5: ADTH with Priority Queuing

With the service differentiation added and delay constraint capability of ADTH, it is possible to achieve deterministic end-to-end delay for delay-sensitive traffic if routes exist between the sources and the destinations that can satisfy the end-to-end delay requirements.

7.6 ADTH with Queue Buffers Reservation

It has been briefly discussed in Section 7.5 that priority queuing can be used to support service differentiation between delay-sensitive and delay-tolerant traffic with multiple transmission queues. If such mechanism is not preferable due to potential starvation of delay-tolerant traffic; a single transmission queue may be used to enqueue both types of traffic. However, delay-sensitive traffic should get bigger share of buffers available on a node.

The target queue threshold calculated by the ADTH controller can be used as the variable to control the amount of delay-tolerant traffic being admitted into a queue when a network is highly loaded or congested. Similar to the concept of admission control or explicit notification mechanism discussed above, a threshold is introduced into the ADTH queue with proportion to the buffer size available. The available buffer size is adapted dynamically by the ADTH controller. Therefore, the new threshold (TH) introduced is also adapted accordingly to the target queue threshold. Whenever the current queue length (CQL) exceeds TH and incoming packets are delay-tolerant type then the packets will be dropped. Delay-sensitive packets are allowed to occupy the rest of the queue buffers. Fig. 7.6 shows the flow chart of queuing operations when enqueuing packets.

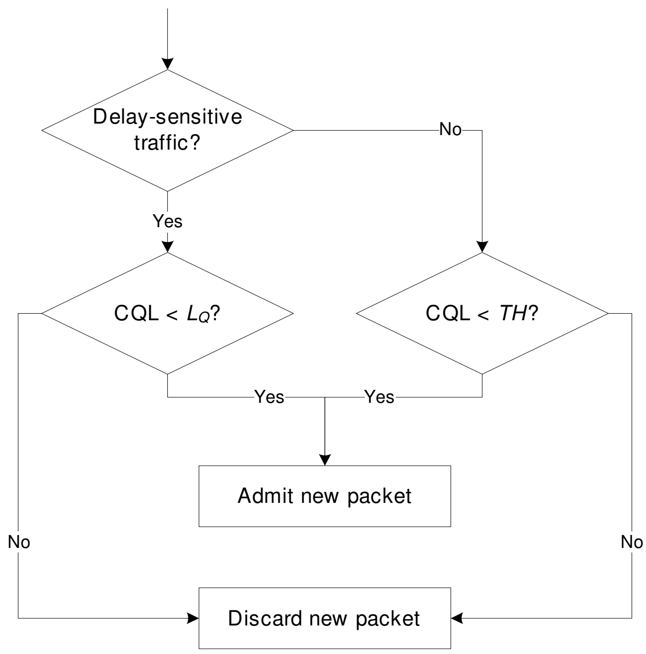


Figure 7.6: ADTH Queuing Operation with Queue Buffer Reservations

Higher bandwidth is allocated implicitly to delay-sensitive traffic by rate-limiting delay-tolerant traffic with the assumption that packet classification is adopted to differentiate the traffic type. The concept is similar to RED-RT [35] which differentiates real-time traffic and non real-time traffic with different dropping profiles. Instead of using different dropping probabilities based on thresholds; delay-tolerant traffic is discarded after exceeding the queue admission threshold.

From the mechanism proposed above, delay-sensitive traffic is given priority if only a single transmission queue is used. This can prevent delay-tolerant traffic from occupying the transmission queue and causes excessive drops of delay-sensitive packets.

7.7 Summary and Discussions

Different type of applications requires different end-to-end QoS guarantee, such as bounded delay, bounded jitter, bounded loss ratio, high throughput, etc. With just the ADTH queue management scheme alone, a guaranteed end-to-end QoS could not be realized. However, the ADTH scheme can be adopted to manage the transmission queues of wireless nodes that require to carry delay-sensitive traffic. ADTH can be combined with delay-based QoS routing protocols to provide end-to-end delay guarantee via the deterministic per-hop nodal delay characteristic featured by ADTH.

ADTH can also be combined with admission control to take care of other important QoS metrics especially the loss ratio and to avoid over-subscription of bandwidth in a network. Although ADTH is unable to constrain the packet loss ratio, its internal states can be used to facilitate the admission control process as suggested in this chapter.

With a proper design of a signalling mechanism and inference of network load, contention level and link quality based on the internal states of ADTH, traffic sources could be informed of the network conditions. The nodes can then react to the network conditions by adjusting the sending rate, or looking for alternative paths in order to ensure the end-to-end QoS can be guaranteed.

Besides that, service differentiation can be combined with ADTH in order to give priority to delay-sensitive traffic over delay-tolerant traffic. It is important to classify and treat traffic of different types properly to achieve required QoS with the assumption that a wireless ad hoc network is setup for various purposes. Therefore, the network load is a mix of traffic with different priorities and QoS requirements. TCP traffic should not be constrained with the ADTH queue management scheme since that TCP traffic is not timing sensitive but favour for higher throughput. However, TCP traffic can be rate-limited to occupy partial buffers of the queue based on the target queue threshold estimated. So that TCP traffic will not monopoly all the bandwidth.

Nevertheless, end-to-end QoS for different applications require a proper design of QoS provisioning solution. This chapter only explores a few potential application scenarios of ADTH with other QoS schemes to get better end-to-end QoS control. A list of future work is presented in Chapter 8.

Chapter 8

Conclusions and Future Work

This thesis focuses on research into adaptive queue management schemes to bound network delay. In this chapter, a summary of work completed is given in Section 8.1 and potential future work is described in Section 8.2.

8.1 Conclusions

Wireless ad hoc networks are normally deployed in areas that have a lack of wired network infrastructure to supporting the task of communication. Their self-configuration and self-organizing nature makes them suitable for communication in battlefields, disaster scenes, etc. Communication in these areas mostly involves voice, video or event data that from delay-sensitive applications. There arises a need to relay delay-sensitive traffic from a source to a destination in timely manner.

Most of the existing queue management schemes focus on congestion control for responsive traffic. Queuing delay is not bounded with the existing approaches. Therefore, this thesis proposes two adaptive queue management schemes, namely, DTH and ADTH, to constrain network delay for delay-sensitive traffic.

DTH is an adaptive queue management scheme which aims to constrain average queuing delay to a specified delay requirement at routers. It applies a dynamic queue threshold concept in regulating queuing delay to the target delay based on the relationship between average queuing delay and queuing threshold derived from a discrete time analytical model. The analytical model is an extension of the existing work [68] to use a more representative arrival process to model aggregated traffic by superposing multiple MMBP-2 arrival process. The proposed solution has been validated using Matlab numerical analysis and simulation to show that average queuing delay is bounded to the target as claimed. DTH is then evolved from the analytical analysis approach to an online algorithmic approach to cope with network dynamics exhibited in wireless ad hoc networks.

The delay analysis in Chapter 4 shows that queuing delay and MAC layer delay are two major components contributing to large nodal delay in a multi-hop contention-based wireless network. Transmission delay and propagation delay are insignificant when compared to queuing delay and MAC layer delay. Processing delay is also insignificant for basic packet processing. Besides that, each wireless node observes different queuing delay even if all wireless nodes have the same amount of backlogs in their queues. Factors such as traffic load, packet size and queue size besides time-varying link quality and interference contribute to network dynamics. All these factors cause fluctuation in system performance and lead to variation in queuing delay and MAC layer delay. Therefore, ADTH aims to bound nodal delay instead of just bounding queuing delay as in DTH. A deterministic nodal delay is needed to achieve a deterministic end-to-end delay.

ADTH is an adaptive queue management scheme which suits fast changing conditions in wireless ad hoc networks. It can respond to time-varying link quality and network dynamics autonomously. ADTH uses an algorithmic approach based on system performance measurements to estimate the target queue threshold for a queue in order to bound nodal delay of the queue. Queuing delay is varied and constrained through an adaptive dynamic queue threshold to compensate for variation in MAC layer delay, so that bounded nodal delay can be achieved.

ADTH is a simple and generic scheme which only needs a nodal delay requirement and sampling interval as configuration parameters. It eliminates the need of parameter configuration to obtain optimum performance and its overhead is minimal. ADTH is agnostic to MAC layer and application layer. The queuing mechanism is transparent to MAC layer as it does not require information nor co-operative mechanism from MAC layer to bound nodal delay. ADTH has been simulated with NS-2 and also implemented on a testbed for validation and performance analysis purposes. The performance analysis has shown that nodal delay can be bounded via ADTH and also bounded end-to-end delay can be achieved if such a route exists either via static routing or dynamic routing. This suggests that nodes with ADTH enabled should be included for route discovery for delay-sensitive traffic. The testbed implementation has convinced the feasibility of ADTH in bounding nodal delay in real world.

Both the DTH and ADTH schemes constrain average queuing delay and maximum nodal delay respectively with higher packet loss rate as a trade-off. Although packet loss ratio is higher when these schemes are enabled, the performance gain is worthwhile as the network delay is significantly lowered. Bandwidth in a wireless ad hoc network is scarce and shared among nodes. These schemes avoid bandwidth wastage and reduce congestion implicitly by dropping packets earlier if the packets potentially miss their deadlines. The potential applications of ADTH and its implications in end-to-end QoS provisioning are briefly discussed in Chapter 7 to cope with the limitations and constraints of ADTH.

There is still room to improve the proposed schemes and complete the proposed solutions with a combination of other QoS schemes to guarantee end-to-end QoS. A list of potential extensions and future work is presented in the next section.

8.2 Future Work

Recommendations for future work are presented here based on the discussions from the previous chapters and illustrations of potential applications of ADTH in a wider aspect with other QoS schemes:

- To extend the ADTH queue management scheme to include processing delay for target queue threshold estimation when complex operations (such as encryption and decryption) are carried out for packets at intermediate nodes.
- To investigate further on the sampling interval of the ADTH controller and enhance the ADTH queue management scheme to self-tune the sampling interval based on the delay requirement and the feedback error of controller. This will reduce the configurable parameters of ADTH to a single parameter which is target nodal delay.
- To incorporate the ADTH scheme with a QoS routing protocol with least modification on the routing protocol in order to discover routes that can satisfy end-to-end delay requirements for delay-sensitive applications.
- To apply the ADTH scheme into priority queuing for service differentiation and assess the performance for delay-sensitive traffic and its impact to delay-tolerant traffic.
- To adopt the ADTH scheme in multiple transmission queues with different delay constraints on each queue and then investigate if the delay constraints are fulfilled for all queues. This may require queuing mechanism other than strict priority queuing to prevent starving of lower priority queues.
- To explore the feasibility of using the internal states of ADTH for admission control or combine ADTH with existing admission control schemes to constrain packet loss ratio.
- To explore the feasibility of manipulating the internal states of ADTH to infer current network conditions and relay the information back to traffic sources for adaptive actions.

- To explore the feasibility of rate-limiting delay-tolerant traffic in a network based on the internal states of ADTH.
- To explore the possibility of incorporating ADTH with other AQM schemes to constrain other important QoS metrics such as packet loss, fairness, etc.
- To further assess the performance and efficacy of ADTH on different type of delay-sensitive applications, such as VoIP, video streaming, teleconference, etc. This may require to adopt other QoS schemes for the performance evaluation.

References

[1] Gumstix. `http://www.gumstix.com/`.

[2] ITU-T G.114 One way transmission time, 2003.

[3] B. Abbasov. AHRED: A robust AQM algorithm for wireless ad hoc networks. In *Application of Information and Communication Technologies, 2009. AICT 2009. International Conference on*, pages 1 –4, oct. 2009.

[4] A. Agbaria, G. Gershinsky, N. Naaman, and K. Shagin. A bandwidth management approach for quality of service support in mobile ad hoc networks. pages 1–5, March 2009.

[5] M.H. Ahmed. Call admission control in wireless networks: a comprehensive survey. *Communications Surveys & Tutorials, IEEE*, 7(1):49–68, Qtr. 2005.

[6] Gahng-Seop Ahn, A.T. Campbell, A. Veres, and Li-Hsiang Sun. Supporting service differentiation for real-time and best-effort traffic in stateless wireless ad hoc networks (SWAN). *Mobile Computing, IEEE Transactions on*, 1(3):192–207, Jul-Sep 2002.

[7] Gahng-Seop Ahn, A.T. Campbell, A. Veres, and Li-Hsiang Sun. SWAN: service differentiation in stateless wireless ad hoc networks. *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, 2:457–466, 2002.

[8] O. Al-Jaber, L. Guan, X.G. Wang, I. Awan, A. Grigg, and X. Chi. Delay restraining of combined multiple input cross core router. In *Advanced Information Networking and Applications, 2008. AINA 2008. 22nd International Conference on*, pages 355–362, March 2008.

[9] F. Al-Raddady and Mike Woodward. A new adaptive congestion control mechanism for the Internet based on RED. *Advanced Information Networking and Applications Workshops, 2007, AINAW '07. 21st International Conference on*, 2:934–939, May 2007.

[10] W. Almesberger et al. Linux network traffic control - implementation overview. In *Sixth IEEE Symposium on*, pages 296–301. Citeseer, 1999.

[11] Giuseppe Anastasi, Eleonora Borgia, Marco Conti, and Enrico Gregori. IEEE 802.11b ad hoc networks: Performance measurements. *Cluster Computing*, 8(2-3):135–145, 2005.

[12] A.T. Andersen and B.F. Nielsen. A Markovian approach for modeling packet traffic with long-range dependence. *Selected Areas in Communications, IEEE Journal on*, 16(5):719–732, Jun 1998.

[13] F.M. Anjum and L. Tassiulas. Fair bandwidth sharing among adaptive and non-adaptive flows in the Internet. In *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1412–1420, Mar 1999.

[14] J. Antila and M. Luoma. Robust delay estimation of an adaptive scheduling algorithm. *Proceedings of QoS-IP*, 2005.

[15] R. Asokan and AM Natarajan. Performance Evaluation of Energy and Delay Aware Quality of Service (QoS) Routing Protocols in Mobile Adhoc Networks. *International Journal of Business Data Communications and Networking (IJBDCN)*, 4(1):52–63, 2008.

[16] S. Athuraliya, S.H. Low, V.H. Li, and Qinghe Yin. REM: active queue management. *Network, IEEE*, 15(3):48–53, May 2001.

[17] Philipp Becker. QoS routing protocols for mobile ad-hoc networks - a survey. Technical Report 368/08, University of Kaiserslautern, August 2007.

[18] J.C.R. Bennett and Hui Zhang. WF2Q: worst-case fair weighted fair queueing. *INFOCOM '96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE*, 1:120–128 vol.1, Mar 1996.

[19] C. Benvenuti. *Understanding Linux network internals*. O'Reilly Media, Inc., 2006.

[20] JN Boshoff and ASJ Helberg. Improving End-to-End Delay for Real-time Multimedia Services in Mobile Ad-hoc Networks. In *Southern African Telecommunications and Networks Conference*, 2007.

[21] J.N. Boshoff and A.S.J. Helberg. Improving QoS for real-time multimedia traffic in Ad-hoc Networks with delay aware multi-path routing. In *Wireless Telecommunications Symposium, 2008. WTS 2008*, pages 1 –8, april 2008.

[22] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang. Recommendations on queue management and congestion avoidance in the Internet. RFC 2309 (Informational), April 1998.

[23] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource reservation protocol (RSVP) – version 1 functional specification. (2205), September 1997. Updated by RFCs 2750, 3936, 4495.

[24] C. Brandauer, G. Iannaccone, C. Diot, T. Ziegler, S. Fdida, and M. May. Comparison of tail drop and active queue management performance for bulk-data and web-like Internet traffic. *Computers and Communications, 2001. Proceedings. Sixth IEEE Symposium on*, pages 122–129, 2001.

[25] P. Brenner. A technical tutorial on the IEEE 802.11 protocol. *BreezeCom Wireless Communications*, 1996.

[26] J.L. Burbank and W.T. Kasch. Cross-layer design for military networks. *Military Communications Conference, 2005. MILCOM 2005. IEEE*, pages 1912–1918 Vol. 3, Oct. 2005.

[27] C.T. Calafate, P. Manzoni, and M.P. Malumbres. Supporting soft real-time services in MANETs using distributed admission control and IEEE 802.11e technology. *Computers and Communications, 2005. ISCC 2005. Proceedings. 10th IEEE Symposium on*, pages 217–222, June 2005.

[28] Yaxin Cao, Ka-Cheong Leung, and V.O.K. Li. Bandwidth-guaranteed fair scheduling with effective excess bandwidth allocation for wireless networks. *Wireless Communications, IEEE Transactions on*, 7(6):2094–2105, June 2008.

[29] I.D. Chakeres and E.M. Belding-Royer. PAC: perceptive admission control for mobile wireless networks. *Quality of Service in Heterogeneous Wired/Wireless Networks, 2004. QSHINE 2004. First International Conference on*, pages 18–26, Oct. 2004.

[30] Satyabrata Chakrabarti and Amitabh Mishra. Quality of service challenges for wireless mobile ad hoc networks. *Wireless Communications and Mobile Computing*, 4(2):129–153, 2004.

[31] Jian Chen and V.C.M. Leung. Applying active queue management to link layer buffers for real-time traffic over third generation wireless networks.

Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE, 3:1657–1662, March 2003.

[32] Shigang Chen and K. Nahrstedt. Distributed quality-of-service routing in ad hoc networks. *Selected Areas in Communications, IEEE Journal on*, 17(8):1488–1505, Aug 1999.

[33] Xiuzhong Chen, Chunfeng Wang, Dong Xuan, Zhongcheng Li, Yinghua Min, and Wei Zhao. Survey on QoS management of VoIP. In *Computer Networks and Mobile Computing, 2003. ICCNMC 2003. 2003 International Conference on*, pages 69 – 77, oct. 2003.

[34] Yuan Chen and Lemin Li. A random early expiration detection based buffer management algorithm for real-time traffic over wireless networks. *Computer and Information Technology, 2005. CIT 2005. The Fifth International Conference on*, pages 507–511, Sept. 2005.

[35] Hyon-Young Choi and Sung-Gi Min. RED-RT for congestion control and real time traffic protection. *Computer and Information Technology, 2006. CIT '06. The Sixth IEEE International Conference on*, pages 115–115, Sept. 2006.

[36] C. Chrysostomou, A. Pitsillides, G. Hadjipollas, M. Polycarpou, and A. Sekercioglu. Congestion control in differentiated services networks using fuzzy logic. *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, 1:549–556 Vol.1, Dec. 2004.

[37] C. Chrysostomou, A. Pitsillides, G. Hadjipollas, A. Sekercioglu, and M. Polycarpou. Fuzzy logic congestion control in TCP/IP best-effort networks. 12, 2003.

[38] F. Cottet, J. Delacroix, Z. Mammeri, and C. Kaiser. *Scheduling in real-time systems*, chapter 7. Wiley, 2002.

[39] E. Crawley, R. Nair, B. Rajagopalan, and H. Sandick. A framework for QoS-based routing in the Internet. RFC 2386 (Informational), August 1998.

[40] M. de Barros and M. El Koutbi. AQM applications in MAODV ad hoc network. In *Telecommunications, 2009. ICT'09. International Conference on*, pages 37–40. IEEE, 2009.

[41] R. de Renesse, V. Friderikos, and H. Aghvami. Cross-layer cooperation for accurate admission control decisions in mobile ad hoc networks. *Communications, IET*, 1(4):577–586, August 2007.

[42] S. De Vuyst, S. Wittevrongel, and H. Bruneel. Place reservation: Delay analysis of a novel scheduling mechanism. *Computers & Operations Research*, 35(8):2447–2462, 2008.

[43] S. De Vuyst, S. Wittevrongel, D. Fiems, and H. Bruneel. Controlling the delay trade-off between packet flows using multiple reserved places. *Performance Evaluation*, 65(6-7):484–511, 2008.

[44] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. In *ACM SIGCOMM Computer Communication Review*, volume 19, pages 1–12. ACM, 1989.

[45] C. Di Cairano-Gilfedder and R.G. Clegg. A decade of Internet research - advances in models and practices. *BT Technology Journal*, 23(4):115–128, 2005.

[46] K. Diwakar and S. Iyer. Supporting real-time applications with better QoS guarantees in 802.11. *ISWCS, Mauritius*, pages 373–377, 2004.

[47] B.M. Dogahe, M.N. Murthi, X. Fan, and K. Premaratne. A distributed congestion and power control algorithm to achieve bounded average queuing delay in wireless networks. *Telecommunication Systems*, pages 1–14, 2010.

[48] O. Dousse. Revising buffering in multihop CSMA/CA wireless networks. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2007. SECON '07. 4th Annual IEEE Communications Society Conference on*, pages 580 –589, june 2007.

[49] C. Dovrolis, D. Stiliadis, and P. Ramanathan. Proportional differentiated services: delay differentiation and packet scheduling. *Networking, IEEE/ACM Transactions on*, 10(1):12–26, Feb 2002.

[50] A. Durresi, M. Sridharan, C. Liu, M. Goyal, and R. Jain. Traffic management using multilevel explicit congestion notification. *Proceedings of SCI*, 2001.

[51] Hesham El-Sayed, Abdelhamid Mellouk, Laurent George, and Sherali Zeadally. Quality of service models for heterogeneous networks: overview and challenges. *Annals of Telecommunications*, 63(11):639–668, 2008.

[52] M.A. Ergin, M. Gruteser, L. Luo, D. Raychaudhuri, and H. Liu. Available bandwidth estimation and admission control for QoS routing in wireless mesh networks. *Computer Communications*, 2008.

[53] H. Fattah and C. Leung. An overview of scheduling algorithms in wireless multimedia networks. *Wireless Communications, IEEE*, 9(5):76–83, Oct. 2002.

[54] W. Feng, K.G. Shin, D.D. Kandlur, and D. Saha. The BLUE active queue management algorithms. *IEEE/ACM Transactions on Networking (TON)*, 10(4):513–528, 2002.

[55] W.-C. Feng, D.D. Kandlur, D. Saha, and K.G. Shin. A self-configuring RED gateway. In *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1320–1328, Mar 1999.

[56] W.C. Feng, DD Kandlur, D. Saha, and KG Shin. Stochastic fair blue: a queue management algorithm for enforcing fairness. *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 3:1520–1529, 2001.

[57] Wu-chang Feng, D.D. Kandlur, Debanjan Saha, and K.G. Shin. BLUE: A new class of active queue management algorithms. *Michigan: University of Michigan*, 1999.

[58] D. Ferrari and D.C. Verma. A scheme for real-time channel establishment in wide-area networks. *Selected Areas in Communications, IEEE Journal on*, 8(3):368–379, 1990.

[59] Wolfgang Fischer and Kathleen Meier-Hellstern. The Markov-modulated Poisson Process (MMPP) cookbook. *Performance Evaluation*, 18(2):149–171, 1993.

[60] S. Floyd, R. Gummadi, S. Shenker, et al. Adaptive RED: An algorithm for increasing the robustness of RED's active queue management, 2001. Preprint, available at http://www. icir. org/floyd/papers. html, August.

[61] Sally Floyd and Van Jacobson. Random early detection gateways for congestion avoidance. *Networking, IEEE/ACM Transactions on*, 1(4):397–413, Aug 1993.

[62] Zhenghua Fu, Haiyun Luo, P. Zerfos, Songwu Lu, Lixia Zhang, and M. Gerla. The impact of multihop wireless channel on TCP performance. *Mobile Computing, IEEE Transactions on*, 4(2):209–221, March-April 2005.

[63] G. Gaertner and V. Cahill. Understanding link quality in 802.11 mobile ad hoc networks. *Internet Computing, IEEE*, 8(1):55 – 60, jan-feb 2004.

[64] Zhihui Ge and Taoshen Li. An Adaptive QoS-Supported Backoff Mechanism for IEEE 802.11. In *Young Computer Scientists, 2008. ICYCS 2008. The 9th International Conference for*, pages 633 –637, nov. 2008.

[65] Rong Geng, Lei Guo, and Xingwei Wang. A new adaptive MAC protocol with QoS support based on IEEE 802.11 in ad hoc networks. *Computers & Electrical Engineering*, In Press, Corrected Proof:–, 2010.

[66] S.J. Golestani. A self-clocked fair queueing scheme for broadband applications. In *INFOCOM'94. Networking for Global Communications., 13th Proceedings IEEE*, pages 636–646. IEEE, 1994.

[67] L. Guan, I.U. Awan, and M.E. Woodward. Stochastic modelling of random early detection based congestion control mechanism for bursty and correlated traffic. *Software, IEE Proceedings* -, 151(5):240–247, 2004.

[68] L. Guan, M.E. Woodward, and I.U. Awan. Control of queueing delay in a buffer with time-varying arrival rate. *Journal of Computer and System Sciences*, 72(7):1238–1248, 2006.

[69] E. Halepovic, C. Williamson, and M. Ghaderi. Wireless data traffic: a decade of change. *Network, IEEE*, 23(2):20–26, March 2009.

[70] L. Hanzo and R Tafazolli. A survey of QoS routing solutions for mobile ad hoc networks. *Communications Surveys & Tutorials, IEEE*, 9(2):50–70, Quarter 2007.

[71] Wenbo He, Klara Nahrstedt, and Xue Liu. End-to-end delay control of multimedia applications over multihop wireless links. *ACM Trans. Multimedia Comput. Commun. Appl.*, 5:16:1–16:20, November 2008.

[72] Yan He and H. Abdel-Wahab. HQMM: A hybrid QoS model for mobile ad-hoc networks. *Computers and Communications, 2006. ISCC '06. Proceedings. 11th IEEE Symposium on*, pages 194–200, June 2006.

[73] Luc Hogie, Pascal Bouvry, and FrÈdÈric Guinand. An overview of MANETs simulation. *Electronic Notes in Theoretical Computer Science*, 150(1):81 – 101, 2006. Proceedings of the First International Workshop on Methods and Tools for Coordinating Concurrent, Distributed and Mobile Systems (MTCoord 2005).

[74] C.V. Hollot, V. Misra, D. Towsley, and Wei-Bo Gong. On designing improved controllers for AQM routers supporting TCP flows. In *INFOCOM*

2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, volume 3, pages 1726–1734, 2001.

[75] Jaesung Hong, Changhee Joo, and Saewoong Bahk. Active queue management algorithm considering queue and load states. In *Computer Communications and Networks, 2004. ICCCN 2004. Proceedings. 13th International Conference on*, pages 140–145, Oct. 2004.

[76] Jorge Hortelano, Juan-Carlos Cano, Carlos T. Calafate, and Pietro Manzoni. Evaluating the performance of real time videoconferencing in ad hoc networks through emulation. In *Proceedings of the 22nd Workshop on Principles of Advanced and Distributed Simulation*, PADS '08, pages 119–126, Washington, DC, USA, 2008. IEEE Computer Society.

[77] B. Hughes, M. Gleeson, M. Karpinski, R. Cunningham, and V. Cahill. Real-Time Communication in IEEE 802.11 Mobile Ad hoc Networks A Feasibility Study. *Trinity College Dublin's Computer Science Technical Report, Ireland*, 2006.

[78] Barbara Hughes and Vinny Cahill. Achieving real-time guarantees in mobile ad hoc wireless networks. *Proc. Work-in-Progress Session 24th IEEE Real-Time Systems Symp*, 2003.

[79] ieee. IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2007.

[80] T. Issariyakul and E. Hossain. *Introduction to Network Simulator NS2*. Springer Verlag, 2008.

[81] itu t. ITU-T G.1010 End user multimedia QoS categories, 2001.

[82] A.K. Jagadev, B.K. Pattanayak, M.K. Mishra, and M.R. Nayak. A QoS solution to hard real time delay satisfaction in ad hoc networks. *IJCSNS International Journal of Computer Science and Network Security*, 9(12):238, 2009.

[83] Hai Jiang, Ping Wang, and Weihua Zhuang. A distributed channel access scheme with guaranteed priority and enhanced fairness. *Wireless Communications, IEEE Transactions on*, 6(6):2114 –2125, june 2007.

[84] H. Jung, H.J. Yeon, and J. Lee. Proportional delay differentiation in multi-hop wireless networks. In *Information Networking, 2008. ICOIN 2008. International Conference on*, pages 1–5. IEEE, 2008.

[85] R. Jurdak, C.V. Lopes, and P. Baldi. A survey, classification and comparative analysis of medium access control protocols for ad hoc networks. *Communications Surveys Tutorials, IEEE*, 6(1):2 –16, quarter 2004.

[86] M. Kalil, H. Al-Mahdi, and A. Mitschele-Thiel. Modeling of an efficient queue management scheme for multihop ad hoc networks. In *Wireless Conference, 2008. EW 2008. 14th European*, pages 1–5, June 2008.

[87] M. Kang, D.I. Kang, and J. Suh. Real-time support on IEEE 802.11 wireless ad-hoc networks: Reality vs. theory. *IEICE Transactions on Communications*, 92(3):737–744, 2009.

[88] A. Kapadia, W. Feng, and R.H. Campbell. GREEN: A TCP equation-based approach to active queue management. Technical report, University of Illinois, 2004. UIUCDCS-R-2004-2408.

[89] T. Karagiannis, M. Molle, and M. Faloutsos. Long-range dependence ten years of Internet traffic modeling. *Internet Computing, IEEE*, 8(5):57–64, Sept.-Oct. 2004.

[90] M. Katevenis, S. Sidiropoulos, and C. Courcoubetis. Weighted round-robin cell multiplexing in a general-purpose ATM switch chip. *Selected Areas in Communications, IEEE Journal on*, 9(8):1265–1279, 1991.

[91] H. Kazemitabar, A. Ali, K. Nisar, M. Said, and H. Hasbullah. A comprehensive review on VoIP over wireless LAN networks. *Computer Science Letters*, 2(2), 2010.

[92] George Kesidis. Bandwidth adjustments using online packet-level measurements. *SPIE Conf. Performance and Control of Network Systems*, pages 230–238, 1999.

[93] Lyes Khoukhi and Soumaya Cherkaoui. FuzzyCCG: a fuzzy logic QoS approach for congestiosn control in wireless ad hoc networks. pages 105–111, 2005.

[94] Lyes Khoukhi and Soumaya Cherkaoui. FuzzyMARS: a fuzzy logic approach with service differentiation for wireless ad hoc networks. *Wireless Networks, Communications and Mobile Computing, 2005 International Conference on*, 2:839–844, June 2005.

[95] J.N. Kim, K.J. Kim, and K.J. Han. Supporting differentiated service in mobile ad hoc networks through congestion control. *Lecture notes in computer science*, pages 643–651, 2005.

[96] Sang-Chul Kim. An end-to-end packet delay optimization for QoS in a MANET. *Lecture Notes in Computer Science*, 4523:653, 2007.

[97] Cuixiang Kong, Zhonghua Guo, Liao Ping, and Xuange Peng. PSRED: A Queue Management Algorithm with Priority Self-adaptive Random Early Detection for Ad Hoc Network. *IWISA 2009. International Workshop on Information Security and Application*, 2009.

[98] P. Kulkarni, M. Nazeeruddin, S. McClean, G. Parr, M. Black, B. Scotney, and P. Dini. Deploying lightweight queue management for improving performance of mobile ad-hoc networks (manets). In *Networking and Services, 2006. ICNS '06. International conference on*, page 102, july 2006.

[99] P.G. Kulkarni, S.I. McClean, G.P. Parr, and M.M. Black. Proactive predictive queue management for improved QoS in IP networks. In *Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies, 2006. ICN/ICONS/MCL 2006. International Conference on*, page 7, april 2006.

[100] K.D. Kumar, I. Ramya, and M.R. Masillamani. Queue management in mobile ad-hoc networks (MANETS). *The 4th Asia-Pacific Embedded Systems Education and Research Conference (APESER-2010)*, Dec 2010.

[101] S. Kumar, V.S. Raghavan, and J. Deng. Medium access control protocols for ad hoc wireless networks: A survey. *Ad Hoc Networks*, 4(3):326–358, 2006.

[102] Srisankar S. Kunniyur and R. Srikant. An adaptive virtual queue (AVQ) algorithm for active queue management. *IEEE/ACM Transactions on Networking*, 12(2):286–299, 2004.

[103] J.F. Kurose, K.W. Ross, and K. Ross. *Computer Networking: A Top-down Approach Featuring the Internet*. Addison-Wesley, international edition edition, 2005.

[104] Y.C. Lai. Two schedulers to provide delay proportion and reduce queueing delay simultaneously. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 51(11):3220–3231, 2007.

[105] L. Larzon, M. Degermark, and S. Pink. UDP lite for real time multimedia applications. *Proceedings of the IEEE International Conference of Communications*, 1999.

[106] Long Le, K. Jeffay, and F.D. Smith. A loss and queuing-delay controller for router buffer management. In *Distributed Computing Systems, 2006. ICDCS 2006. 26th IEEE International Conference on*, pages 4–4, 2006.

[107] S.B. Lee, G.S. Ahn, X. Zhang, and A.T. Campbell. INSIGNIA: An IP-based quality of service framework for mobile ad hoc networks. *Journal of Parallel and Distributed Computing*, 60(4):374–406, 2000.

[108] Jian Li, Zhi Li, and Prasant Mohapatra. Adaptive per hop differentiation for end-to-end delay assurance in multihop wireless networks. *Ad Hoc Networks*, 7(6):1169 – 1182, 2009.

[109] T. Li and D. Leith. Adaptive buffer sizing for TCP flows in 802.11 e WLANs. In *Communications and Networking in China, 2008. ChinaCom 2008. Third International Conference on*, pages 7–11. IEEE, 2008.

[110] T. Li, D. Leith, and D. Malone. Buffer sizing for 802.11-based networks. *Networking, IEEE/ACM Transactions on*, 19(1):156–169, 2011.

[111] Tianji Li and D.J. Leith. Buffer Sizing for TCP Flows in 802.11e WLANs. *Communications Letters, IEEE*, 12(3):216 –218, march 2008.

[112] Xuefei Li and L. Cuthbert. Node-disjoint multipath routing and distributed cross-layer QoS guarantees in mobile ad hoc networks. In *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2006. SNPD 2006. Seventh ACIS International Conference on*, pages 243 – 248, june 2006.

[113] Wen-Hwa Liao, Yu-Chee Tseng, and Kuei-Ping Shih. A TDMA-based bandwidth reservation protocol for QoS routing in a wireless mobile ad hoc network. *Communications, 2002. ICC 2002. IEEE International Conference on*, 5:3186–3190 vol.5, 2002.

[114] L.B. Lim, L. Guan, A. Grigg, I.W. Phillips, X. Wang, and I.U. Awan. RED and WRED performance analysis based on superposition of N MMBP arrival proccess. In *2010 24th IEEE International Conference on Advanced Information Networking and Applications*, pages 66–73. IEEE, 2010.

[115] Chunhung Richard Lin and Mario Gerla. Real-time support in multihop wireless networks. *Wirel. Netw.*, 5(2):125–135, 1999.

[116] C.R. Lin and M. Gerla. Asynchronous multimedia multihop wireless networks. *INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE*

Computer and Communications Societies. Proceedings IEEE, 1:118–125, Apr 1997.

[117] D. Lin and R. Morris. Dynamics of random early detection. *ACM SIGCOMM Computer Communication Review*, 27(4):127–137, 1997.

[118] Morten Lindeberg, Stein Kristiansen, Thomas Plagemann, and Vera Goebel. Challenges and techniques for video streaming over mobile ad hoc networks. *Multimedia Systems*, 17:51–82, 2011. 10.1007/s00530-010-0187-8.

[119] Anders Lindgren and Elizabeth M. Belding-Royer. Multi-path admission control for mobile ad hoc networks. pages 407–417, 2005.

[120] Christian Lochert, Bjorn Scheuermann, and Martin Mauve. A survey on congestion control for mobile ad hoc networks: Research articles. *Wirel. Commun. Mob. Comput.*, 7(5):655–676, 2007.

[121] M. Ma and Q. Zhu. Providing delay guaranteed service in CDMA wireless networks. *Computer Communications*, 2007.

[122] P. Macharla, R. Kumar, A.K. Sarje, and M. Misra. A QoS routing protocol for delay-sensitive applications in mobile ad hoc networks. pages 720–727, 2008.

[123] D. Malone, P. Clifford, and D.J. Leith. On buffer sizing for voice in 802.11 WLANs. *Communications Letters, IEEE*, 10(10):701–703, Oct. 2006.

[124] B.S. Manoj and C. Siva Ram Murthy. Real-time traffic support for ad hoc wireless networks. *Networks, 2002. ICON 2002. 10th IEEE International Conference on*, pages 335–340, 2002.

[125] P. Marbach. Distributed scheduling and active queue management in wireless networks. *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 2321–2325, May 2007.

[126] P. Marbach and Y. Lu. Active queue management and scheduling for wireless networks: The single-cell case. *Information Sciences and Systems, 2006 40th Annual Conference on*, pages 1560–1565, March 2006.

[127] M Mirhakkak, N. Schult, and D. Thomson. Dynamic quality-of-service for mobile ad hoc networks. Technical report, MITRE Corporation, 2000.

[128] A. Munaretto, H. Badis, K. Al Agha, and G. Pujolle. A link-state QoS routing protocol for ad hoc networks. In *Mobile and Wireless Communications Network, 2002. 4th International Workshop on*, pages 222–226. IEEE, 2002.

[129] S. Munir and I. Ahmad. VOIP on wireless LAN: a comprehensive review. In *Proceedings of the 7th WSEAS International Conference on Electronics, Hardware, Wireless and Optical Communications*, pages 225–230. World Scientific and Engineering Academy and Society (WSEAS), 2008.

[130] C.S.R. Murthy and BS Manoj. *Ad hoc wireless networks: architectures and protocols*. Prentice Hall PTR Upper Saddle River, NJ, USA, 2004.

[131] S. Murthy, P. Hegde, and A. Sen. Design of a delay-based routing protocol for multi-rate multi-hop mobile ad hoc networks. In *Communications, 2009. ICC'09. IEEE International Conference on*, pages 1–6. IEEE, 2009.

[132] L. Muscariello, M. Mellia, M. Meo, M. Ajmone Marsan, and R. Lo Cigno. Markov models of Internet traffic and a new hierarchical MMPP model. *Computer Communications*, 28(16):1835–1851, 2005.

[133] R. Nagarajan, S.P. Parekh, and K.M. Rege. Link access delay aware routing over mobile ad hoc networks (MANETs). December 29 2006. US Patent App. 20,080/159,142.

[134] J. Nagle. On packet switches with infinite storage. *Communications, IEEE Transactions on*, 35(4):435–438, 1987.

[135] N.S. Nandiraju, D.S. Nandiraju, D. Cavalcanti, and D.P. Agrawal. A novel queue management mechanism for improving performance of multihop flows in IEEE 802.11s based mesh networks. In *Performance, Computing, and Communications Conference, 2006. IPCCC 2006. 25th IEEE International*, pages 7 pp.–168, April 2006.

[136] Essam Natsheh, Adznan B. Jantan, Sabira Khatun, and Shamala Subramaniam. Fuzzy active queue management for congestion control in wireless ad-hoc. *The International Arab Journal of Information Technology*, 4(1), Jan 2007.

[137] Pirabakaran Navaratnam, Haitham Cruickshank, and Rahim Tafazolli. A link adaptive transport protocol for multimedia streaming applications in multi hop wireless networks. pages 1–6, 2007.

[138] C.H. Ng and Boon-Hee Soong. *Queueing Modelling Fundamentals: With Applications in Communication Networks*. Wiley Publishing, 2nd edition edition, 2008.

[139] CH Ng, L. Yuan, W. Fu, and L. Zhang. Methodology for traffic modeling using two-state Markov-modulated Bernoulli Process. *Computer Communications*, 22(13):1266–1273, 1999.

[140] N. Olifer and V. Olifer. *Computer Networks: Principles, Technologies and Protocols for Network Design.* John Wiley, 2006.

[141] Rong Pan, B. Prabhakar, and K. Psounis. CHOKe - a stateless active queue management scheme for approximating fair bandwidth allocation. *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2:942–951 vol.2, 2000.

[142] C. Perkins and E. Belding-Royer. Quality of Service for Ad-hoc On-demand Vector Routing. *IETF Draft, October*, 2003.

[143] D.D. Perkins and H.D. Hughes. A survey on quality-of-service support for mobile ad hoc networks. *Wireless Communications and Mobile Computing*, 2(5):503–513, 2002.

[144] A. Pitsillides, P. Ioannou, and L. Rossides. Congestion control for differentiated-services using non-linear control theory. *Computers and Communications, 2001. Proceedings. Sixth IEEE Symposium on*, pages 726–733, 2001.

[145] V.S. Radhakrishnan and S. Radhakrishnan. Linux-Advanced Networking Overview (Version 1). Technical report, Department of Electrical Engineering & Computer Science, The University of Kansas, August 1999.

[146] R. Ramaswamy, Ning Weng, and T. Wolf. Characterizing network processing delay. volume 3, pages 1629–1634, Nov.-3 Dec. 2004.

[147] T.B. Reddy, I. Karthigeyan, BS Manoj, and C.S.R. Murthy. Quality of service provisioning in ad hoc wireless networks: a survey of issues and solutions. *Ad Hoc Networks*, 4(1):83–124, 2006.

[148] E.M. Royer and Chai-Keong Toh. A review of current routing protocols for ad hoc mobile wireless networks. *Personal Communications, IEEE*, 6(2):46–55, Apr 1999.

[149] A. Rubini and J. Corbet. *Linux Device Drivers.* O'Reilly Media, 2nd ed edition, 2001.

[150] M.G. Rubinstein, I.M. Moraes, M.E.M. Campista, L. Costa, and O. Duarte. A survey on wireless ad hoc networks. *INTERNATIONAL FEDERATION FOR INFORMATION PROCESSING-PUBLICATIONS-IFIP*, 211:1, 2006.

[151] O. Salem and A. Benzekri. Towards absolute end-to-end QoS in ad hoc networks. *Telecommunications, 2006. AICT-ICIW '06. International Conference on Internet and Web Applications and Services/Advanced International Conference on*, pages 18–18, Feb. 2006.

[152] L. Sánchez-Miquel, N. Vesselinova-Vassileva, F. Barceló, and P. Carbajo-Flores. Energy and delay-constrained routing in mobile ad hoc networks: an initial approach. pages 262–263, 2005.

[153] B. Scheuermann, C. Lochert, and M. Mauve. Implicit hop-by-hop congestion control in wireless multihop networks. *Ad Hoc Networks*, 6(2):260–286, 2008.

[154] J. Sengupta and G.S. Grewal. Performance evaluation of IEEE 802.11 MAC layer in supporting delay sensitive services. *International Journal of Wireless & Mobile Networks (IJWMN*, 2(1):42–53, 2010.

[155] S.H. Shah, K. Chen, K. Nahrstedt, et al. Available bandwidth estimation in IEEE 802.11-based wireless networks. In *The 1st Bandwidth Estimation Workshop (BEst 2003)*. Citeseer, 2003.

[156] S.M. Shalinie, G. Preetha, S.D. Nidhya, and K.D. Subramanian. Fuzzy adaptive tuning of router buffers for congestion control. *International Journal of Advancements in Technology*, 1(1):85–94, 2010.

[157] Qiang Shen, Xuming Fang, Pan Li, and Yuguang Fang. Admission control based on available bandwidth estimation for wireless mesh networks. *Vehicular Technology, IEEE Transactions on*, 58(5):2519–2528, Jun 2009.

[158] Shiann-Tsong Sheu and Jenhui Chen. A novel delay-oriented shortest path routing protocol for mobile ad hoc networks. *Communications, 2001. ICC 2001. IEEE International Conference on*, 6:1930–1934 vol.6, 2001.

[159] M. Shreedhar and G. Varghese. Efficient fair queueing using deficit round robin. *ACM SIGCOMM Computer Communication Review*, 25(4):231–242, 1995.

[160] P. Siripongwutikorn, S. Banerjee, and D. Tipper. A survey of adaptive bandwidth control algorithms. *IEEE Communications Surveys and Tutorials*, 5(1):14–26, 2003.

[161] Marcelo M. Sobral and Leandro B. Becker. A wireless hybrid contention/TDMA-based MAC for real-time mobile application. In *Proceedings of the 2008 ACM symposium on Applied computing*, SAC '08, pages 284–288, New York, NY, USA, 2008. ACM.

[162] J. Soldatos, E. Vayias, and G. Kormentzas. On the building blocks of quality of service in heterogeneous IP networks. *Communications Surveys Tutorials, IEEE*, 7(1):69 –88, qtr. 2005.

[163] S. Srivathsan and SS Iyengar. Minimizing latency in wireless sensor networks: a survey. *Proceedings of the third conference on IASTED International Conference: Advances in Computer Science and Technology table of contents*, pages 159–164, 2007.

[164] Y. Sun, E.M. Belding-Royer, X. Gao, J. Kempf, and CA San Jose. Real-time traffic support in large-scale mobile ad hoc networks. *Proc. of BroadWIM*, 2004.

[165] A.K. Talukdar, BR Badrinath, and A. Acharya. MRSVP: A resource reservation protocol for an integrated services network with mobile hosts. *Wireless Networks*, 7(1):5–19, 2001.

[166] Shensheng Tang and Wei Li. QoS provisioning and queue management in mobile ad hoc networks. *Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE*, 1:400–405, 0-0 2006.

[167] V.S.Y. To, B. Bensaou, and S.M.K. Chau. Quality of service framework in MANETs using differentiated services. *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, 5:3463–3467 Vol.5, Oct. 2003.

[168] Amy Tu. Market Specifications for Product DRCM-81 (WLAN 802.11g Module), March 2005.

[169] Pedro Velloso, Marcelo Rubinstein, and Otto Duarte. Evaluating voice traffic requirements on IEEE 802.11 ad hoc networks. *Annals of Telecommunications*, 63:321–329, 2008. 10.1007/s12243-008-0032-7.

[170] D.D. Vergados, D.J. Vergados, C. Douligeris, and S.L. Tombros. QoS-aware TDMA for end-to-end traffic scheduling in ad hoc networks. *Wireless Communications, IEEE*, 13(5):68–74, October 2006.

[171] D.C. Verma, H. Zhang, and D. Ferrari. Delay jitter control for real-time communication in a packet switching network. In *Communications Software, 1991,'Communications for Distributed Applications and Systems', Proceedings of TRICOMM'91., IEEE Conference on*, pages 35–43. IEEE, 1991.

[172] Dapeng Wang, Kai Liu, Lianzhen Cheng, and Yan Zhang. A novel multiple access protocol with QoS support for mobile ad hoc networks. In Keqiu Li, Chris Jesshope, Hai Jin, and Jean-Luc Gaudiot, editors, *Network and*

Parallel Computing, volume 4672 of *Lecture Notes in Computer Science*, pages 260–266. Springer Berlin / Heidelberg, 2007.

[173] J. Wang, M. Zhou, and Y. Li. Survey on the end-to-end Internet delay measurements. *Lecture Notes in Computer Science*, pages 155–166, 2004.

[174] Jie Wang, Lin Guan, Lee Booi Lim, X.G. Wang, A. Grigg, I. Awan, I. Philips, and X. Chi. Qos enhancements and performance analysis for delay sensitive applications. *Journal of Computer and System Sciences*, In Press, Corrected Proof:–, 2010.

[175] Kuang-Ching Wang and P. Ramanathan. QoS assurances through class selection and proportional differentiation in wireless networks. *Selected Areas in Communications, IEEE Journal on*, 23(3):573–584, March 2005.

[176] M. Wang and G.S. Kuo. An application-aware QoS routing scheme with improved stability for multimedia applications in mobile ad hoc networks. In *Vehicular Technology Conference, 2005. VTC-2005-Fall. 2005 IEEE 62nd*, volume 3, pages 1901–1905. IEEE, 2005.

[177] M.E. Woodward. *Communication and computer networks*. IEEE Computer Society Press Los Alamitos, Calif, 1994.

[178] Hannan Xiao, W.K.G. Seah, A. Lo, and K.C. Chua. A flexible quality of service model for mobile ad-hoc networks. *Vehicular Technology Conference Proceedings, 2000. VTC 2000-Spring Tokyo. 2000 IEEE 51st*, 1:445–449 vol.1, 2000.

[179] Heng Xu, Qi Xue, and A. Ganz. Adaptive congestion control in infrastructure wireless LANs with bounded medium access delay. *Mobility and Wireless Access Workshop, 2002. MobiWac 2002. International*, pages 44–49, 2002.

[180] K. Xu, M. Gerla, L. Qi, and Y. Shu. TCP unfairness in ad hoc wireless networks and a neighborhood RED solution. *Wireless Networks*, 11(4):383–399, 2005.

[181] Q. Xue and A. Ganz. Ad hoc QoS on-demand routing (AQOR) in mobile ad hoc networks. *Journal of Parallel and Distributed Computing*, 63(2):154–165, 2003.

[182] Yuan Xue, Hoang Nguyen, and K. Nahrstedt. CA-AQM: Channel-aware active queue management for wireless networks. *Communications, 2007. ICC '07. IEEE International Conference on*, pages 4773–4778, June 2007.

[183] Y. Yang and R. Kravets. Achieving delay guarantees in ad hoc networks through dynamic contention window adaptation. *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–12, April 2006.

[184] Yaling Yang and R. Kravets. Distributed QoS guarantees for realtime traffic in ad hoc networks. *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pages 118–127, Oct. 2004.

[185] Yaling Yang and R. Kravets. Contention-aware admission control for ad hoc networks. *Mobile Computing, IEEE Transactions on*, 4(4):363–377, July-Aug. 2005.

[186] K. Yasukawa, A.G. Forte, and H. Schulzrinne. Distributed delay estimation and call admission control in IEEE 802.11 wireless lans. In *Communications, 2009. ICC'09. IEEE International Conference on*, pages 1–6. IEEE, 2009.

[187] Jeonggyun Yu, Sunghyun Choi, and Jaehwan Lee. Enhancement of VoIP over IEEE 802.11 WLAN via dual queue strategy. In *Communications, 2004 IEEE International Conference on*, volume 6, pages 3706 – 3711 Vol.6, june 2004.

[188] Xiaohong Yuan and M. Ilyas. Modeling of traffic sources in ATM networks. In *SoutheastCon, 2002. Proceedings IEEE*, pages 82–87, 2002.

[189] Ching-Wan Yuen, Wing-Cheong Lau, and On-Ching Yue. CAPEL: A packet discard policy for real-time traffic over wireless networks. *Communications, 2007. ICC '07. IEEE International Conference on*, pages 158–163, June 2007.

[190] S.M. Zaki, M.A. Ngadi, and S.A. Razak. A review of delay aware routing protocols in MANET. *Computer*, 1:1, 2009.

[191] S.T. Zargar and M.H. Yaghmaee. Fuzzy Green: A modified TCP equation-based active queue management using fuzzy logic approach. *IJCSNS International Journal of Computer Science and Network Security*, 6(5A):50, 2006.

[192] Hongqiang Zhai, Xiang Chen, and Yuguang Fang. A call admission and rate control scheme for multimedia support over IEEE 802.11 wireless LANs. *Wirel. Netw.*, 12(4):451–463, 2006.

[193] L. Zhang. Virtual clock: a new traffic control algorithm for packet switching networks. *ACM SIGCOMM Computer Communication Review*, 20(4):19–29, 1990.

[194] Qian Zhang and Ya-Qin Zhang. Cross-layer design for QoS support in multihop wireless networks. *Proceedings of the IEEE*, 96(1):64–76, Jan. 2008.

[195] B. Zheng and M. Atiquzzaman. DSRED: A new queue management scheme for the next generation Internet. *IEICE Transactions on Communications*, pages 764–774, 2006.