ACHIEVING SOFT REAL-TIME GUARANTEES FOR INTERACTIVE APPLICATIONS IN WIRELESS MESH NETWORKS

by

Cecil M. Reid

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2008

Author's Declaration For Electronic Submission of a Dissertation

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

ABSTRACT

The use of 802.11-based multi-hop wireless mesh networks for Internet access is extensive and growing. The primary advantages of this approach are ease of deployment and lower cost. However, these networks are designed for web and e-mail applications. Highly interactive applications, such as multiplayer online games and VoIP, with their requirements for low delay, present significant challenges to these networks. In particular, the interaction between real-time traffic and TCP traffic tends to result in either a failure of the real-time traffic getting its needed QoS or the TCP traffic unnecessarily experiencing very poor throughput. To solve this problem we place real-time and TCP traffic into separate queues. We then rate-limit TCP traffic based on the average queue size of the local or remote real-time queues. Thus, TCP traffic is permitted to use excess bandwidth as long as it does not interfere with real-time traffic guarantees. We therefore call our scheme Real-time Queue-based Rate and Admission Control, RtQ-RAC. Extensive simulations using the network simulator, ns-2, demonstrate that our approach is effective in providing soft real-time support, while allowing efficient use of the remaining bandwidth for TCP traffic.

ACKNOWLEDGEMENTS

First, I need to acknowledge my parents for their love, support and encouragement from day one. Second, I thank my supervisor, Paul Ward. His continual guidance has been instrumental and encouraging. Moreover, his wide knowledge and experience kept me focused and I am very grateful for our many discussions. Next, I need to thank all my friends, new and old. My time in Waterloo has been amazing because of them. I must also thank Kamran Jamshaid for his extensive help through our many discussion. There have been times when my understanding of some material was unclear. Kamran was always able to point me in the right direction to figuring it out. I need to thank my readers, Raouf Boutaba and Kshirasagar Naik. Finally, I thank Diane Cyr and the Canadian Bureau for International Education for their financial support through the Commonwealth Scholarship Programme.

CONTENTS

FIGURES

TABLES

1 INTRODUCTION

Interactive multimedia, particularly Voice over Internet Protocol (VoIP) and Multiplayer Online Games (MOG) are experiencing tremendous growth. VoIP applications and services have seen extensive growth in the residential and corporate arena [69]. VoIP service providers such as Skype [95] have seen tremendous growth in users [96] and business corporations are aggressive in moving towards VoIP-based Private Branch Exchange (PBX) deployments [21]. The ubiquity of broad-band access to the Internet and low deployment costs have been influential in driving growth of VoIP services. MOG interest is evident not only in franchised PC-based online games such as Doom, Quake, and Unreal but in the increased sales of console-based games supported by XBox, Play Station, and Nintendo. Video-on-demand services are also bandwidth-demanding and are provided by websites such as You Tube, MSN Video, and MTV. Although the use of interactive multimedia applications is growing its world-wide Internet use is minimal at 3% when compared to best-effort applications at 95% [43]. Best-effort or elastic-traffic services mostly use TCP as transport and include: email, news groups, HTTP, and long-lived file retrieval applications such as FTP, Bittorrent, and other peer-to-peer (P2P) services.

The increasing demand for multimedia applications is driving the need for, and roll-out of, broadband access networks to the Internet. Broadband access networks to the Internet are called first- or last-mile access networks as shown in Figure 1.1. Examples access technology include: DSL (Digital Subscriber Line), (coaxial) cable access, T1 (T-carrier or DS1 - Digital Service - Level 1) or E1 (E-carrier), and optical fiber. In the consumer and small-business space, DSL and cable primarily provide broadband access. Large companies use T1 or aggregated T1s (*e.g.,* T3) [92, 100, 102] and optical fiber. Initially, optical fiber was used as distribution back-haul for cross-oceanic transport; cable companies used it as the transport media between central-office switches and distribution hubs. Now, optical fiber is used to deliver access to businesses and homes [14, 23].

In the wireless arena, broadband access technologies include: third generation (3G) cellular (*e.g.,* UMTS and W-CDMA) for mobile users, satellite access - VSAT (Very Small Apeture Terminal) and DBS (Direct Broadcast Satellite, *e.g.,* DirecTV or DISH Network that provide Internet access), WiMAX, WLANs.

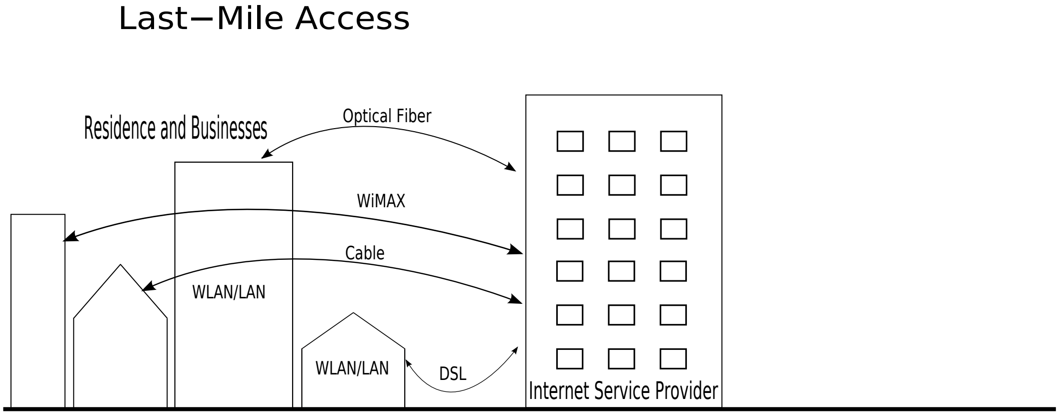


Figure 1.1: Last-mile Access

Access to the backhaul is shared within organizations using a LAN (Local Area Network) or Wireless LAN (WLAN); users and hosts interface with the LAN (via Ethernet or 802.11) but the access to the Internet is provided by an access technology such as optical fiber. Similarly, consumers can set-up their own wired or wireless area networks that access the Internet via cable or DSL. In general, a wireless or wired LAN extends the last-mile access.

The above access technologies have a common drawback of expensive deployment costs, especially with respect to the general consumer market. DSL and cable, for example, require an extensive physical installation of wire for every home and between the central switch and distribution hub. The expense is not only because of the cost of wires, but the cost in logistics and time and construction costs associated with ploughing and tunnelling roads for the installation. Costs are more significant barriers in rural communities. ; satellite access can provide rural communities with broadband Internet access, however, significant link latencies, cost of equipment and lack of competition make it unattractive for average consumers. Wired and wireless LANs have significant wiring costs as each host or access point (AP) requires an Ethernet cable connection while 3G and WiMAX are expensive alternatives because of expensive spectrum licensing and base transceiver station (BTS) deployment.

A wireless mesh network (WMN) is an attractive option for last-mile broadband Internet access. As shown in Figure 1.2, a WMN is a multihop network; mesh clients and static mesh routers wirelessly communicate with each other to provide communication even when they are not directly within radio-transmission range. One or more clients

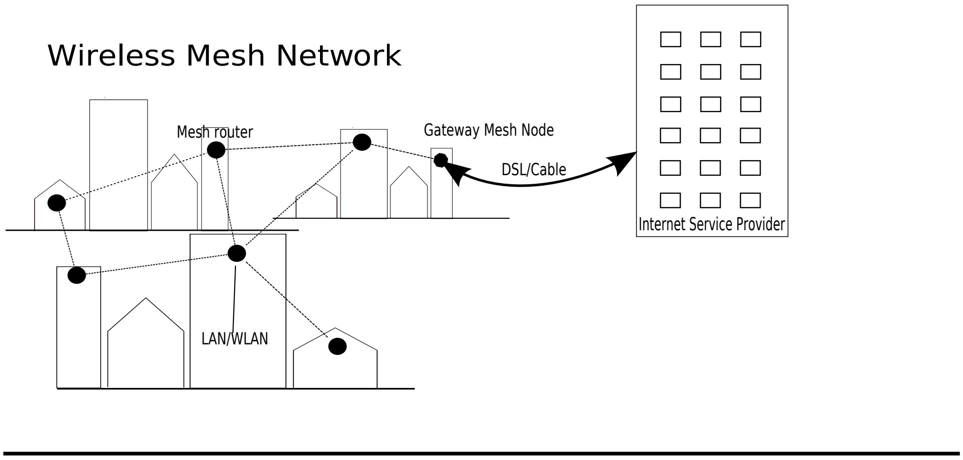


Figure 1.2: Last-mile Access via Wireless Mesh Network

directly connect to a mesh router using Ethernet or an orthogonal connection. Mesh routers are capable of operating as gateways providing Internet access to the clients. A mesh gateway is a mesh node that serves as an entrance to the Internet and vice-versa transferring data between the mesh network and the Internet. Wireless mesh networks are cheap compared to the alternatives. They have lower deployment costs with respect to time and wiring. When built from IEEE 802.11 equipment, costs are even lower because of the low cost and ready availability of its components and its adequate tranmission range. Meraki Networks [77], for example, use 802.11 components to build cheap mesh nodes that cost US $49. Blue-tooth is also cheap and ubiquitous, however, because of its relatively short transmission range, low transmission rate and medium access control, it is impractical for building a WMN.

In a WMN, wiring costs are cheap because only a few mesh routers need to be gateways with wired connections to the Internet. Depending on the particular deployment, reconfiguration is also easy as mesh routers do not require fixed mounting and can always be easily removed. WMNs are also attractive for rural areas because of reduced wiring costs. They are able to scale or extend a single or few Internet connections reducing access costs as a few dedicated connections can be shared amongst a very large community, for example, without extensive wiring. Moreover, a WMN can be used to extend a WLAN as mesh routers can forward data to an access point via a wireless hop [65]. Other key advantages of wireless mesh networks include ease of installation, automatic connection among all nodes, network flexibility, automatic discovery of newly added nodes, redundancy, and self-healing reliability [3]. All these qualities make WMNs an interesting and attractive technology for home, community, wireless metropolitan area, and enterprise networking [3]. There are a significant number of WMN equipment vendors such as Intel, Motorola, CISCO, Nortel, and Strix Systems; companies such as Microsoft Research are developing WMN software technology. Moreover, there are many WMN deployments such as CUWiN (Champaign-Urbana Community Wireless Network), Wireless Philadelphia, and MIT RoofNet [22] that validate the interest in and viability of wireless mesh networks.

1.1 Interactive Real-time Application Support

Interactive applications, such as VoIP, multiplayer online games, and Interactive Video, require a guaranteed level of service. Their traffic is bidirectional; at least two parties are interacting, talking to each other or competing in a game, for example. If a packet carrying voice, game state, or an image-update is delayed or lossed, then interactivity is affected. Thus, it is important to ensure that such messages are delivered within certain delay and loss bounds. An 802.11-based WMN can cause large delay and significant packet loss, making the support of QoS-sensitive applications challenging. These problems are because of the susceptible nature of wireless transmission to interference that adversely affects capacity, the relative packet size to the 802.11 frame header overhead, and the default 802.11 scheduling which causes interflow and intraflow interference. In addition, the default 802.11 scheduling operates regardless of competing traffic classes; real-time traffic is unable to attain desired QoS in the presence of competing TCP traffic.

Many researchers have studied the problems in supporting QoS in WMNs, but a few have proposed solutions to improve the QoS for interactive applications. Proposed solutions include: MAC-layer and MAC contention window (CW) modifications, QoS-routing, packet-aggregation, and admission control and rate-limiting governed by MAC-layer delays or an estimated fraction of air-time (FAT). In this thesis, it is our objective to achieve real-time QoS while ensuring an adequate share of bandwidth for TCP traffic, regardless of topology. We intend to achieve soft, or statistical, and not hard real-time QoS guarantees. We use existing commodity 802.11 hardware and as such our solution must not modify the existing the 802.11 MAC which would limit practicability.

1.2 Contributions

In this thesis, we study the efficacy of a node's real-time queue in influencing rate control of elastic traffic to improve interactive real-time application QoS. In addition, we study the use of such per-node queues for real-time traffic admission control. Our rate and admission control mechanisms are combined into a scheme called Real-time Queue-based Rate and Admission Control, RtQ-RAC. The main contributions of this work are as follows:

1. A simple mechanism for ensuring quality of service for interactive real-time multimedia in WMNs. This mechanism operates between layers 2 and 3 of the network stack and requires no modification to existing hardware and MAC.

2. A comprehensive evaluation that supports the view of using an orthogonal real-time traffic queue and class to effectively and directly affect the elastic traffic class. In particular, we showed that a real-time queue can be used to explicitly enforce rate control on elastic traffic to attain real-time QoS. Our real-time queues mark their own packets to signal other nodes to perform rate control. In addition, packet marking is used as a metric to help determine admission control decisions.

3. A demonstration and discussion of the difficulty in achieving real-time QoS with competing elastic traffic.

4. An evaluation of our mechanism in allowing elastic traffic to efficiently use residual bandwidth.

1.3 Thesis Organization

In Chapter 2 we describe wireless communication and our wireless mesh network model. Further, we review QoS related literature, discuss and demonstrate the difficulty in supporting interactive real-time applications in wireless mesh networks. We end by reviewing previous approaches to guaranteeing real-time QoS. We describe Real-time RED which is our initial approach in solving the problem in Chapter 3. In Chapter 4 we build on our previous approach, describing the design and operation of Real-time Queue-based Rate and Admission Control. We evaluate its performance via extensive simulations in Chapter 5 and end the thesis by presenting our conclusions and future work in Chapter 6.

2 Background and Related Work

In this chapter, first we present our communication model and media access control (MAC) for wireless networks, describing the IEEE MAC protocol as example. Second, we describe our WMN model and present a mathematical model for computing per-flow residual capacity. After describing interactive applications, we present general QoS background. We then discuss the effects of 802.11 MAC operation on WMNs that make soft real-time application support challenging and discuss the efficacy of service disciplines in solving that problem. We present a taxonomy of existing QoS approaches in multihop networks and finally, we critique related work that attempts to solve the problem of achieving soft real-time guarantees.

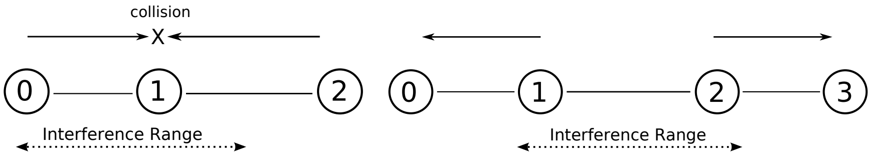
2.1 Wireless Communication

Wireless communication is the transfer of data between a sender and receiver, via radio waves, without the use of wires or fibres. Senders and receivers are called nodes and form a wireless communication network. Examples include fixed-wireless, cellular, and wireless LAN (WLAN). Wireless channels (*i.e.* radio waves of a particular frequency range) are detectable and disrupted by receivers and transmitters of similar frequency responses. In addition, the reliability and efficiency of wireless communication depend on the wireless transmission and interference ranges and the relative node locations. We explain using the following model.

2.1.1 Wireless Communication Model

A receiver is in transmission range of a sender when the transmitted signal is of sufficient fidelity that it is correctly decoded. If a receiver is in interference range, but out of transmission range a signal is detected but lacks enough fidelity to be correctly decoded. A receiver out of interference range of the sender cannot detect or interfere with each other.

The interplay of node location and the various wireless communication ranges potentially lead to hidden-terminal and exposed-terminal problems [94, 97]. Figure 2.1(a)



(a) Hidden Terminal (b) Exposed Terminal

Figure 2.1: CSMA MAC Coordination Problems

illustrates three nodes, 0, 1, and 2. Nodes 0 and 2 are out of interference range of each other. A solid line indicates nodes are within transmission range. Therefore, nodes 0 and 1 are in transmission range of each other and nodes 1 and 2 are in transmission range of each other. Node 0 can continually send a packet to node 1 or node 2 can continually send a packet to node 1 because node 0 cannot detect node 2 transmissions (*i.e.* carrier sense) and vice versa. The sender sees an idle channel. If node 0 transmits to node 1, then node 2 transmits to node 1, the signal reception at node 1 is potentially garbled because of interference by node 2's signal with node 0's signal. A hidden terminal problem occurs when senders are unable to detect other senders transmissions, causing collisions at a receiver.

An exposed terminal problem occurs when sending nodes are in interference range of each other and continually defer their transmission to avoid collisions even though successful receptions are possible. Figure 2.1(b) illustrates four nodes, 0, 1, 2, and 3. Nodes 1 and 2 are in interference range and defer their transmissions if the other transmits first. If node 2 transmits to node 3, node 1 will defer its transmission to node 0 and vice versa, increasing delay. However, node 1's transmission to node 0 and node 2's transmission to node 3 can occur simultaneously without failure. The incorrect perception by node 1 and 2 is the exposed terminal problem. The exposed terminal problem occur with certain medium access control protocols, for example, MACAW [9].

2.1.1.1 Wireless Medium Access Control

Wireless channels can be randomly accessed or disrupted prompting the need for an arbitration mechanism that allow access while minimizing contention and interference. Such a mechanism is defined in the medium access control (MAC) sub-layer.

MAC mechanisms are categorized according to the method of access and or sharing of the wireless channel(s) between nodes and can be classified as contention-free and contention-based. In contention-free access, nodes do not compete for wireless media as their share is allocated by a central arbitrator. Examples are TDMA (Time Division Multiple Access), FDMA (Frequency Division Multiple Access), and CDMA (Code Division Multiple Access). In contention-based access, nodes randomly contend with each other for an opportunity to access wireless media forming a distributed arbitration mechanism. CSMA (Carrier Sense Multiple Access)[1] and its variants (*e.g.*, CSMA with Collision Detection (CSMA/CD) and CSMA with Collision Avoidance (CSMA/CA)are contention-based.

Networks that have a central coordinator or base station, such as cellular require the use of a TDMA MAC for example, in which time slots are determined by the base station. When central control is hardware resource intensive or unfeasible, a CSMA (*e.g.*, ALOHA, MACAW) MAC is appropriate. Performance related issues such as guaranteed access to the media and QoS vary between MAC classes. A contention-free MAC is immune to node interference as the wireless medium is divided and nodes only access their share. Interactive real-time data such as voice are effectively supported because bandwidth is guaranteed and interference is minimized. However, a network using such a MAC is inefficient in supporting best effort and bursty IP traffic; frequencies in FDMA, for example, will not be re-allocated to other nodes if they are unused. Although reliability is worse because of possible packet collisions, a contention-based MAC is more efficient for best-effort traffic.

In CSMA, nodes listen for a carrier (*i.e.*, an electric magnetic signal that implies data transmission or a busy channel). Nodes access the channel only when its free (*i.e.*, idle) of a carrier, but may react in a specified way on detection of one to guarantee access on subsequent attempts. There are two classes of CSMA protocols: p-persistent and non-persistent [94]. The IEEE 802.11 standard defines a mandatory non-persistent CSMA with collision avoidance (CSMA/CA) scheme implemented as its Distributed Coordination Function (DCF). We now discuss the 802.11 MAC protocol in more detail.

2.1.2 IEEE 802.11 MAC

IEEE 802.11 uses a carrier sense multiple access with collision avoidance (CSMA/CA) protocol with binary exponential backoff (BEB) [37]. The standard defines two modes

[1] CSMA-based MAC protocols are also used in wired environments *e.g.*, Ethernet.

of operation, Distributed Coordination Function (DCF) and Point Coordination Function (PCF). There are three defined important waiting times for media access: DIFS (DCF inter-frame spacing, PIFS (PCF inter-frame spacing) and SIFS (short inter-frame spacing). The following describes the mandatory DCF. When a station has a frame to transmit, it monitors the channel's busy status for a DIFS period. After the channel is idle for DIFS, it waits a random backoff time before transmitting. The backoff timer decrements by one slot time while the channel remains idle. It pauses when the channel is busy. The frame is transmitted when the timer expires. The random backoff time is uniformly selected from the range of $[0, CW - 1]$ where CW is the current contention-window size. Each unsuccessful unicast transmission, implied when an ACK (frame acknowledgement) is not recieved from the destination, doubles CW until CW_{max} is reached. CW is reset after a successful transmission or when the retransmission counter is at the retry limit, dropping the frame. A successful receipt of a frame at the destination causes it to transmit an ACK after SIFS time. The doubling operation of the CW is the BEB algorithm and by default, the minimum and maximum CW is 32 and 1024, respectively.

If request-to-send/clear-to-send (RTS/CTS) is enabled, a station transmits an RTS frame before transmitting a data frame. The destination, if it senses the medium is idle, replies after SIFS time with a CTS frame reserving the channel. After receiving the CTS frame, the station waits SIFS time and then the data frame is sent. If a sender does not receive the CTS frame, it assumes a collision occurred and executes BEB. The RTS and CTS frame identify their receivers and contain a duration field that specifies the required time for transmission. The fields and mechanism are designed to reduce the occurrence of the hidden terminal problem; nodes other than the sender and destination do not send and adjust their earliest time-to-transmit from the duration fields whenever they receive RTS and CTS packets.

The 802.11 standard also defines the optional PCF which is a centrally controlled access mechanism. A point coordinator (PC) arbitrates the PCF and is usually located in an access point (AP). In PCF mode, the PC regularly polls stations for traffic and simultaneously deliver traffic to them. PCF is layered above the DCF, and uses the PIFS to help prevent stations operating in DCF from accessing the medium. A contention free period (CFP) begins when the PC access the medium via the DCF at which the PCF operation begins.

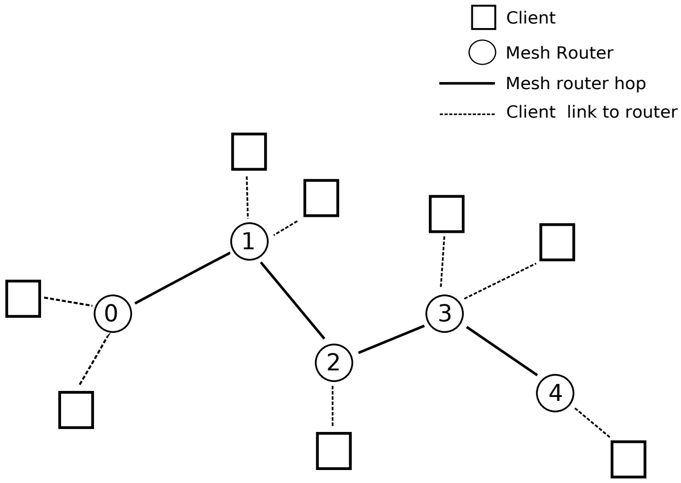


Figure 2.2: Multihop Wireless Network: WMN

2.1.3 WMN Model

A wireless mesh networks (WMN) is a form of multihop network in which communication is possible and extended to nodes that are not in transmission range with each other. Communication is made possible via intermediate nodes on each link (a hop), from the source, between intermediate nodes, to the destination. Figure 2.2 illustrates a multihop wireless network. Node 0 sends a message to node 4. Nodes 0 and node 4 are not in transmission range with each other, however node 1 is, and thus relays the message to node 2; node 2 relays the message to node 3 and so on until the node 4 receives the message. This wireless relayed method of communication causes three forms of contention that do not in single-hop networks. First, maximum throughput and capacity are not determined by the capacity of a single link, but is constrained by the bottleneck link on route that allocates the least bandwidth to that flow. Second, because the wireless medium is broadcast, originated traffic and forwarded traffic contend with each other. Moreover, network flows in the same spatial area contend for media and limit each other's achieved throughput. Third, a flow self-contends across multiple hops because each hop uses the same channel.

In a WMN, a node is either a fixed mesh router or mobile mesh clients. Mesh routers form the backbone of the network, communicating with each other, while routing and forwarding mesh clients' data. In this thesis, we assume that clients do not participate in forwarding, but only send messages to, or receive messages from, mesh routers via a wired connection (*e.g.* Ethernet) or an orthogonal wireless channel. A router is thus a wired network gateway or wireless access point to a client, enabling traditional LAN or WLAN operation.

Clients include desktop PC, laptops, smart phones and other network devices. Mesh routers are specially configured computers with wireless NICs connected to antennae. Routers can be configured with multiple wireless NICs, enabling multichannel communications. They are powered by an electrical outlet, which makes them practically immobile. Routers form the core of the WMN creating topologies that are mostly static. Changes occur during incremental router deployment and node failure which are rare. Routing is also effectively static as we assume re-routing is significantly less frequent than changes in user traffic activity. Mesh client intra-communication is very limited; clients mostly communicate with remote hosts. Messages to or from remote hosts to nodes within the network must traverse a mesh router gateway node. A single WMN may comprise multiple gateways; however, assuming static routing, we partition these into disjoint sets, each with a single gateway. We assume that interference between disjoint sets does not occur.

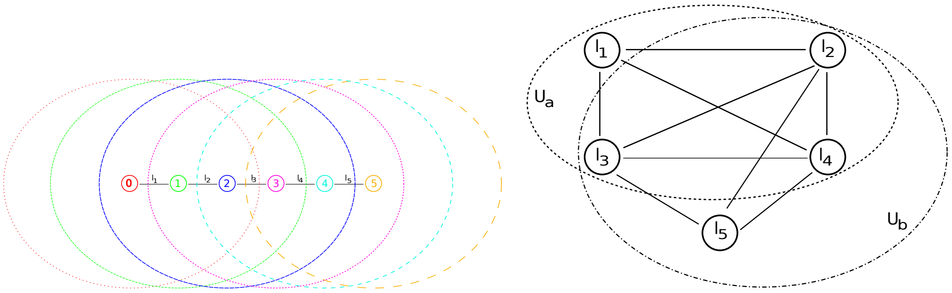
Our view of a WMN is similar to a WLAN in terms of client access, however, the distribution network is a single- or multi-channel multihop wireless network. Our model is a generalization of the Transit-Access-Point (TAP) model [35], and is similar to architectures such as that of Lee *et al.* [65]. WMNs of this type are suited to a community environment, where mesh routers are deployed on houses, serving client devices in the household. Mesh routers communicate with each other, forwarding traffic to and from a gateway router connected to a high bandwidth (wired) connection to the Internet.

2.1.3.1 MODELLING WMN CAPACITY

We represent a WMN as a connectivity graph $G = (V, E)$, where V is the set of vertices that represents mesh routers and E is the set of edges that represents the links (hops) between mesh routers. There is an edge between two vertices, v_i and v_j, if they can potentially transmit successfully between each other.

We use the clique-graph model [26, 46, 73][2] to determine the capacity of a WMN.

[2] Collision-domain theory [52] is another model that is used to model WMN capacity, however it has been shown by Li [67] to be less accurate than clique theory for single-channel WMNs.



(a) 6-node Chain Topology, each node with corresponding interference range

(b) Link Contention Graph

Figure 2.3: Clique Model

Using this model, we represent the contention area for links with a link-contention graph $G_c = (V_c, E_c)$, where V is the set of all links in the connectivity graph, G, and $\{u, v\} \in E_c$ iff links u and v contend. Two links contend if they are within interference range of each other. Specifically, they contend if either node from one link is in the interference range of either node from the other link.

A clique in a contention graph is a set of vertices that contend with each other for medium. At any instance, only one link (*i.e.* the link that wins the contention) may be active in a clique. A clique is termed maximal, if no other clique has more vertices. Links in independent cliques can transmit simultaneously.

A maximal clique is used to compute an upper bound of network capacity. Let $C(l)$ be the maximum aggregate throughput that link l carries; $B(u)$ is the available bandwidth in each maximal clique, u. Assuming all rates are equal, $B(u)$ is approximated as the one-hop theoretical maximum throughput (TMT) [50, 52]. All links in a clique share the same channel bandwidth and thus the channel resource constraint is defined as:

$$\sum_{i:i \in u} C(l_i) \leq B(u) \tag{2.1}$$

We now illustrate using Figure 2.3. We define a chain topology as shown in Figure 2.3(a). Nodes are spaced 200 m, with a transmission and interference range of 250 m and 550 m respectively. In this topology, links of nodes that are two hops apart interfere with each other. As an example, link l_1 contends with link l_4 because their respective nodes, node 1 and node 4 is less than 550 m apart and are thus in interference range of each other. Figure 2.3(b) shows the resulting contention graph, with two maximal cliques, u_a and

u_b, each of four links. Their respective channel resource constraints are:

$$\begin{array}{c}\sum_{i=1}^{4} C(l_i) \leq B(u_a) \\ \sum_{i=2}^{5} C(l_i) \leq B(u_b)\end{array}$$

Because links within a contention area share the network resource at an instance of time, it is effective to express the shared network resource as time that needs to be shared, rather than bandwidth in bits per second. In essence, we would like to know the fraction of time a flow is allocated to communicate its data (See [64] and [103]). Consider two nodes, a and b, operating at the same link rate, and transmitting packets of the same size to each other. Each node attempts to gain its maximum throughput. Now considering Equation 2.1 and the fact that the nodes share a clique, we get $C(l_i) \leq B$, where l_i is a link in the clique. Moreover, because there are two flows $a{\rightarrow}b$ and $b{\rightarrow}a$, at any instance, the probability of that either node will capture the medium (link) is 0.5. The time share (probability) per flow is thus 0.5. Using Equation 2.1, the throughput per flow f as [64]:

$$p_{l_i}^{f} = t_{l_i}^{f} B^{f}(u) \tag{2.2}$$

Flows from the set of flows, F, that are in the same contention or maximal clique, u, share the resource and are thus constrained by [64]:

$$\sum_{f=1, i \in u}^{F} t_{l_i}^{f} \leq 1 \tag{2.3}$$

With Equations 2.2 and 2.3, we derive the time share of a flow, ε, within a maximal clique, u, and among contending flows as:

$$t_{l_i}^{\varepsilon} = 1 - \sum_{f \neq \varepsilon, i \in u}^{F} \frac{p_{l_i}^{f}}{B^{f}(u)}, \tag{2.4}$$

where the fraction $\frac{p_{l_i}^{f}}{B^{f}(u)}$ is the time share, $t_{l_i}^{f}$, of another flow, f derived from Equation 2.2. We denote the clique capacity as B^f to take in consideration the change in capacity with packet size per flow as established in [50].

As an example, consider Figure 2.3(a) with a bidirectional flow between nodes 0 and 5 of a 60-byte packet, 48 kbps load in either direction. We need to model the remaining time share and capacity for flow $4 \rightarrow 5$ on link, l_5. Using Equation 2.2, the time share per flow ($0 \rightarrow 5$ or $5 \rightarrow 0$) per link is:

$$\begin{aligned} t_{l_i}^{f} &= \frac{p_{l_i}^{f}}{B^{f}(u)} \\ &= \tfrac{48}{513.717} \\ &= 0.0934 \end{aligned}$$

Using Equation 2.4, the time share for flow $4 \rightarrow 5$ is:

$$\begin{aligned} t_{l_5}^{4\rightarrow 5} &= 1 - \sum \frac{p_{l_i}^{f}}{B^{f}(u)} \\ &= 1 - (t_{l_2}^{0\rightarrow 5} + t_{l_3}^{0\rightarrow 5} + t_{l_4}^{0\rightarrow 5} + t_{l_5}^{0\rightarrow 5} + t_{l_2}^{5\rightarrow 0} + t_{l_3}^{5\rightarrow 0} + t_{l_4}^{5\rightarrow 0} + t_{l_5}^{5\rightarrow 0}) \\ &= 1 - 0.747 \\ &= 0.253 \end{aligned}$$

With this time share value, we use Equation 2.2 to convert to bits per second, if necessary.

2.2 Interactive Real-time Applications

In this section, we describe interactive real-time applications and present their requirements. Further, we present First-person Shooter games as an example.

An interactive application requires a guaranteed quality of service throughout its execution. There are three main types of interactive implications: VoIP, multiplayer online games (MOG), and Interactive Video (*e.g.*, video-conferencing). An Interactive application's data is mostly transported using the UDP or UDP/RTP protocols. Further, traffic is bidirectional because at least two parties are interacting. For example, in a VoIP session, the voice of a speaker is encoded as UDP/RTP packets, delivered over the network, and decoded into sound as it reaches the ear of the other participant and vice versa. The interactivity which determines the perceived enjoyment of the service is affected by delay and loss if they are not within certain bounds. It is recommended for voice traffic that one-way delay (mouth to ear) should be no more than 150 ms with the average one-way jitter less than 30 ms [44]. Depending on the sampling codec, voice traffic may require a bandwidth of 21–320 kbps [15]. If the required level of service is not guaranteed, in VoIP, for example, the participants may experience voice clipping, skips and long silences that give the impression of a disconnected call. Interactive-video requirements are similar, however, the bandwidth demands are greater, requiring at least 460 kbps [93]. Multiplayer online game QoS requirements can be more demanding; we present First-Person Shooter game as example in the following section.

2.2.1 First-person Shooter

A first-person shooter (FPS) game requires players to move around a defined game space in real-time, interacting with (typically, shooting at) other players to achieve the game's objectives. Each player-action requires two state-update messages: one from the player's computer to the game server; the second from the server to all players affected by the action. Such state-update messages are small, but frequent. Client messages are 50–90 bytes every 10–50 ms; server messages are 60–300 bytes every 50–60 ms [63]. In many games a state update is a complete state description, and therefore subsumes any prior state updates [8].

Multiplayer online games are characterized by two main interdependent requirements: interactivity and consistency. Interactivity refers to the delay between event generation and receipt of that event by other nodes, and its perception is dependent on some human-related threshold. Interactivity is inversely related to delay; higher interactivity requires smaller delays and vice-versa. Consistency means that all participants should have an uniform game-state view amongst all participants. When consistency is achieved, the game is perceived as fair. Consistency is hard to achieve if there are large differences in participants' end-to-end delay. A player in an FPS game is at a competitive disadvantage if the RTT for that participant exceeds 150 ms, while 75 ms is visually noticeable [6, 80]. Loss rates of 5% or less are rarely noticed and have no statistical effect on the game's outcome [6]. There is no study we are aware of to indicate at what point loss rates impact player performance. Subjective experiments within our lab suggest that it can be quite large (10% or higher). Other real-time multiplayer online games (MOGs) have similar, but slightly less stringent requirements than FPS games. We therefore focus on ensuring network QoS requirements are met for FPS games.

2.3 Quality of Service

In this section we present general QoS background on performance metrics, service disciplines, congestion management, active queue management, and admission control. In addition, we discuss and illustrate the difficulty of supporting interactive real-time applications in wireless mesh networks and the efficacy of service disciplines in guaranteeing QoS.

2.3.1 Performance Metrics

As we have seen from Section 2.2.1, for multimedia applications, QoS performance is subject to human perception. Human perception enables us to support soft real-time, rather than hard real-time requirements. Hard requirements are deterministic, in that QoS is guaranteed and strictly enforced by the network based a contract between the user and network provider. Soft requirements are statistical, in that not every single instance is strictly enforced. For example, a hard real-time guarantee might be delivering within n milliseconds while a soft real-time guarantee would be delivering within n milliseconds 95% of the time and with a certain known statistic distribution of non-compliance. Humans can tolerate such variances to a point for multimedia applications.

In providing guaranteed QoS, it is of great importance to recognize and meet the performance requirements that vary with application traffic characteristics. For interactive applications that have stringent real-time requirements, end-to-end delay is very important. Throughput is the amount of messages successfully transmitted per unit time. The third important metric, is the delay jitter which is the maximum difference between delays experienced by any two packets [30, 99]. Jitter is thus the standard deviation of the end-to-end delay. Jitter has important implication in regards to buffer occupancy as smaller jitter requires less buffer space. The last important metric is the loss rate. The loss rate is the ratio of successfully received messages to the total amount of messages sent. Loss occur due to buffer overflows, channel errors, or explicit discard because of delay bound violations.

2.3.2 Service Disciplines

One of the most important issues in providing real-time guarantees is based on the packet service discipline deployed [114, 116]. Service disciplines determine the order that packets from different flows or connections are served. Connections may be grouped as per class or per flow, where a class is a general application field such as best-effort and real-time and per flow is with respect to particular source-destination pairs. Three types of resources are being governed by service disciplines: bandwidth, promptness, and buffer space which, in turn, influences throughput, delay, and loss rate [25].

An appropriate service disciplines per application should be efficient, protective, flexible, and simple [114]. One service discipline is more efficient than another if it results in higher utilization of the network, under heavier load, while achieving the same end-to-end performance guarantee. A protective service discipline guards well behaving flows from ill-behaving clients, network load variances, and unconstrained best-effort traffic. A service discipline's flexibility is a measure of its ability to adapt its support to applications with diverse traffic characteristics and QoS requirements. Therefore, it is able to allocate different delay, bandwidth, and loss rates to flows with differing requirements.

A service discipline can be classified as work-conserving or non-work-conserving. A work-conserving discipline is never idle when there is a message (*i.e.* packet, etc.) to send. With a non-work-conserving discipline, each message is sent according to an eligibility time, that is implicitly or explicitly assigned [27, 114]. Even, if the link is idle, packets will not be transmitted if they are ineligible. Examples of work-conserving disciplines are: fair queueing (FQ) [25], weighted fair queueing (WFQ) [81], self-clocked fair queueing (SCFQ) [39], worst-case fair weighted fair queueing (WF^{2}Q) [7], round-robin (RR), Deficit Round Robing (DRR) [87], priority queueing [71], virtual clock [117], first-come first-served (FCFS), and delay earliest-due date (delay EDD) [31, 54]. Non-work-conserving disciplines include: jitter earliest-due-date (jitter-EDD) [99], hierarchical round robin (HRR) [53], rate controlled static priority (RCSP) [115], and stop-and-go [38].

There is a coupling between end-to-end delay and bandwidth [114] in work conserving disciplines because the server allocates resources in proportion to the flows' loads. Because of this coupling, the resulting QoS is susceptible to distortions from network-load variances causing burstiness and instantaneously higher rates. Non-work conserving disciplines are immune to these effects as some packet are held even if there is extra capacity. Non-work-conserving service disciplines have properties that support real-time service requirements. In non-work-conserving disciplines, because the rate-control mechanism and scheduler are separated, delay and bandwidth are decoupled and can be achieved without the use of priority queueing. Its paced control helps to reduce the required buffer space at each forwarding node. Moreover, traffic have desirable bounded rates, delay, and jitter.

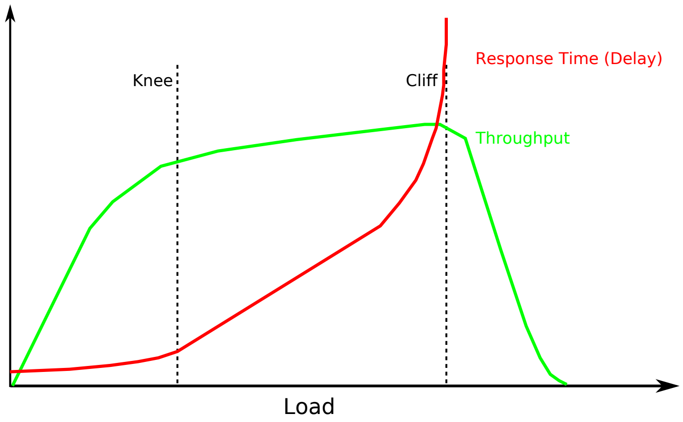


Figure 2.4: Network performance and congestion as a function of load

2.3.3 Congestion Avoidance and Control

In IP-based networks, congestion occurs when the aggregate rate of input traffic *i.e.* load to the network (or subset of the network) temporarily exceeds the capacity of the network causing large delays and high loss rates [114].

Without congestion, achieving soft real time QoS is a function of how fast the network can transmit packets. Given that congestion occurs because of various factors (including heterogeneous traffic and mismatched link speeds), achieving soft-real time guarantees depends on the congestion management mechanisms. Congestion avoidance ensures that a network operates in the best range of low delay and high throughput without the network becoming congested; congestion control recovers the network from a congested state of large delay and low throughput [19]. Figure 2.4 illustrates the knee and cliff [19] operation points with the network load and shows the response time (delay) variation. A congestion avoidance scheme keeps the network operating at or below the knee while a congestion control scheme keeps the network left of the cliff. Throughput and load vary linearly until just below network capacity (*i.e.*, knee) after which throughput slowly increases. When the load is increased further, the queues build-up and packet loss begin because the network is congested. After cliff point, throughput suddenly decreases as congestion collapse occurs. For the delay curve, the delay slowly increases with load. As queue build up increases, the delay is linear; when packet loss occur the delay exponentially increases.

Congestion management is either reactive or proactive. The latter determines the available resources with a computation scheme that is used to control load below capacity, avoiding congestion. A reactive algorithm detects and reacts to congestion based on feedback data from the network. The efficacy of a reactive congestion management depends on the speed of relaying the congestion signal (*i.e.*, feedback) that influences a reaction while the proactive scheme is dependent on accurately computing the resources [114].

Chiu and Jain [19] analyzed reactive congestion management algorithms and came to the conclusion that simple additive-increase multiplicative-decrease (AIMD) algorithm is the most stable as it satisfies the metrics of efficiency, fairness, convergence time, and size of oscillations regardless of starting state in the network. The ubiquitous transport protocol, TCP, utilizes AIMD for flow and congestion control while many papers in the literature that attempt to achieve QoS are based on AIMD and a combination of other established algorithms and techniques such as Explicit Congestion Notification (ECN) packet marking [33], fair queueing [25], and priority queueing [71]. Congestion management mechanisms continually adjust rate allocations making it suitable for elastic or best-effort traffic; variations in rate allocations are inappropriate for real-time traffic because it is mostly inelastic.

2.3.4 Active Queue Management

An active queue management (AQM) scheme is the method and mechanisms that determine when or how a buffered packet should be marked or discarded. AQM schemes detect the onset of congestion via a threshold parameter on the queue size. The marking of, or discarding of, packets is used as explicit or implicit signals of congestion to flow and congestion control and avoidance schemes. Random early detection (RED) [32] uses an exponentially-weighted moving average (EWMA) of the queue size and a linear probability function of the this average to determine the marking or dropping probabilities for packets. There are many variants of RED, such as: adaptive RED, balanced RED, flow RED [4, 29, 72]. Specifically, to wireless multihop networks are Link RED and neighbourhood RED [34, 107]. Virtual Queue (VQ) and Adaptive Virtual Queue (AVQ) [62] use a virtual queue of smaller capacity than the real queue, marking packets if the virtual queue overflows [57]. Fundamentally different AQM schemes are BLUE and stochastic fair BLUE (SFBLUE) [28] that use packet loss and idle link events instead of queue sizes to signal incipient congestion. While BLUE, RED, and variants indirectly characterize or estimate load, Load-Based Marking (LBM) [88] calculates marking probabilities from directly measuring link loads [57].

2.3.5 Admission Control

Admission control is the process of determining resource availability and using this information to allow or deny a requesting connection into the network. It limits the amount of flows into the network ensuring that the network is operating within capacity which in turn protects and enables flows to attain their requested QoS.

There are three main approaches to admission control: deterministic, stochastic, and measurement-based [86]. Traditionally, admission control have taken the deterministic or stochastic approach (*e.g.*, [24, 70]), where an a priori traffic specification defining parameters of the deterministic or stochastic model is required. The admittance decision is determined only from the specifications of new and existing connections. The main disadvantage of these approaches is that traffic such as variable bit-rate (VBR) can be hard to characterize a priori. As a result, traffic may be over- or underestimated which if admitted, leads to inefficient utilization or insufficient resource allocation.

Measurement-based admission control (MBAC) (*e.g.*, [40, 47, 57, 59, 83]) simplifies the traffic specification requirement to a simple peak rate, for example, but determines the statistics of existing traffic by indirect observations or direct measurements.

In addition to how the admittance decision is made, admission control schemes are classified by where the decision is made. The decision can be made only at edge (egress) nodes (*e.g.* [13, 16, 57]) or at internal (ingress) nodes along the path (*e.g.*, [12]).

2.3.6 Challenges to Soft Real-time Support

The overhead per packet can cause the operation of an 802.11-based network to be very inefficient, particularly when the payload is small, as is the case with FPS games or VoIP applications. The overhead is caused by the 802.11 DCF MAC, PHY Header and preamble, MAC ACK, and collision avoidance. The theoretical maximum throughput (TMT), *i.e.*, capacity over one hop, is [50]:

$$TMT(x) = \frac{8x}{ax+b} \times 10^6 bps \tag{2.5}$$

MSDU (bytes)	TMT (Mbps)	SMT (Mbps)
60	0.514	0.521
120	0.982	0.995
1500	6.056	6.094

Table 2.1: Maximum One-hop Throughput of 802.11 at 11 Mbps

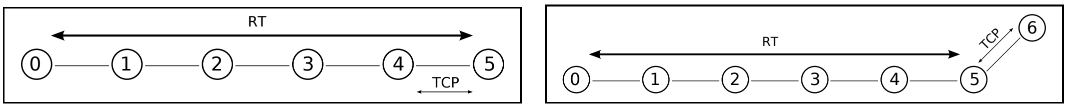
where x is the MAC service data unit (MSDU; *i.e.,* the MAC payload size). At 11 Mbps, $a = 0.72727$ and $b = 890.73$. Table 2.1 shows the TMT and ns-2 simulated maximum throughput (SMT) of typical FPS game-size and TCP-size packets.

In addition to low throughput for small packets, the 802.11 MAC causes significant delay, especially in multi-hop scenarios. There are three major causes of delay: offered load, number of hops to the gateway, and hidden-terminal effects. The offered load increases delay not merely because capacity is spread over multiple senders, but also because the contention window, CW, badly reacts to a higher load. After every successful transmission, CW_{max} is reset to 32. As such, a node that succeeds in transmitting is likely to succeed again, while one that perceives a collision, will increase its CW, and thus is likely to fail again. The result is bursts of transmission opportunities [42].

The number of hops increases delay partly because there are more transmissions required. However, it also increases delay because each such transmission competes not only with traffic transmitting through that mesh router, but also with all traffic within interference range, including its own.

Finally, various hidden-terminal effects [97], combined with BEB, can dramatically increase delay. When two senders are out of carrier-sense range, but within interference range of the others receiver, repeated collisions can occur [36]. Li has shown that with source rate limiting this will not affect throughput [67], but the collision rate is comparable to the sending rate, resulting in a significant increase in delay, as each packet has to be transmitted, on average, twice. The effects of RTS/CTS has been repeatedly shown to be poor in multi-hop wireless networks and does not solve the problem (*e.g.,* [67]). QoS-enhanced MACs are also not immune. IEEE 802.11e, for example, has a enhanced DCF (EDCF) and Hybrid Coordination Function (HCF) which have the ability to adjust the CW and inter-frame spacing (IFS) and prioritize channel access for certain frames. However, in a multihop setting, frames even of the highest priority suffer from unpredictable delay and throughput degradation caused by hidden terminals and other interference [75].

In addition to the above problems, competing best-effort TCP traffic, without addi-



(a) RT and TCP share paths

(b) RT adn TCP on disjoint paths

Figure 2.5: Challenges to Real-time Traffic

tional control prevents real-time traffic from acheiving its desired QoS. We illustrate this with the following ns-2 simulations.

Figure 2.5 shows two meshes. Each node has an 802.11b card operating at a link speed of 11 Mbps. Carrier sense and communication range are 550 m and 250 m, respectively, with node separated by 200 m. We define a bidirectional real-time flow, RT of 60-byte packets every 10 ms (48 kbps) from node 0 to node 5. In addition, there is an infinite TCP flow from node 4 to 5 (Figure 2.5(a)) and 6 to 5 (Figure 2.5(b)). Each node has the ns-2 default drop-tail queue with a 50-packet capacity.

The resulting performance is shown in Table 2.2. Absent TCP traffic, the real-time flows receive their desired bandwidth, with an average delay of 6.7 ms ± 1.2 ms. However, with TCP traffic, it is immediately apparent that the RT flows achieves very poor performance, achieving neither their required bandwidth nor acceptable delays and loss rates for interactive applications. In both cases, the poor real-time flow performance is caused by TCP being able to continually increase its sending window as it is reacting to a single packet loss with fast retransmit, rather than with slow start [79], and thus it does not slow down appreciably. In addition, the TCP sender in both cases is out of carrier-sense range of various of the nodes that are transmitting real-time flows, but within range of their respective receivers of real-time traffic. As such, the TCP flow can successfully acquire the wireless medium, while at the same time causes the real-time flows to experience packet loss and delay. The real-time flows experience information asymmetry [36] leading to significant delay and packet loss. TCP experiences no congestion and continuously increases its sending window, monopolizing node 4's queue (Figure 2.5(a)) with its packets, reducing the queue capacity for forwarded RT traffic, resulting in increased packet loss at the queues [51].

It should be noted that poor performance is possible with non-TCP traffic. Consider flow $4 \rightarrow 5$, or flow $6 \rightarrow 5$, in scenario 1 or 2. If these flows were non-TCP (*e.g.,* UDP) and operating at unfairly high loads, similar poor real-time performance occurs.

(a) QoS for Scenario A - Figure 2.5(a)

$S \rightarrow D$	Throughput (kbps)	Delay (ms)	Jitter (ms)	Loss Rate (%)
4→5	2092	86	32	0.2
0→5	38.3	414	127	20
5→0	33.0	443	141	31

(b) QoS for Scenario B - Figure 2.5(b)

$S \rightarrow D$	Throughput (kbps)	Delay (ms)	Jitter (ms)	Loss Rate (%)
6→5	2727	10	10	0
0→5	26.0	663	124	45.5
5→0	29.5	721	152	38.5

Table 2.2: TCP traffic reduces real-time QoS

2.3.7 Service Discipline Efficacy

Service disciplines are known to be very effective in wireline networks. However, the unavoidable and unique issues of wireless communication challenges the effectiveness of service disciplines in these networks. Specifically, from Section 2.3.6 we see that the channel capacity is dynamically varying; channel errors are location-dependent as mobile stations sharing the medium will perceive different interference levels and fading. In addition, these are compounded by the lack of global channel state; nodes contend to discover availability to transmit.

Fair queueing, which includes all the disciplines that simulate the fluid fair queueing model [81], is often used to provide bounded-delay channel access and flow separation in wireline networks. We illustrate the efficacy of fair queueing by considering the scenario shown in Figure 2.5(a). Each node has a deficit round robin (DRR) scheduler. The simulation results are shown in Table 2.3. DRR is ineffective in guaranteeing real-time QoS. DRR simulates the fluid fair queueing model. The model defines the following property that must be satisfied with respect to each flow, i:

$$\forall i, j \in B(t_1, 2_2), \left| \frac{W_i(t_1, t_2)}{r_i} - \frac{W_j(t_1, t_2)}{r_j} \right| = 0, \tag{2.6}$$

$B(t_1, t_2)$ is the set of backlogged flows, $W_i(t_1, t_2)$ is the channel capacity granted to flow i and r_i is weight of flow $i's$ rate. The explanation is as follows [10]. Consider three flows during the time interval $[0, 2]$. Flows 1 and 2 have access to an idle and error-free

$S{\rightarrow}D$	Throughput (kbps)	Delay (ms)	Jitter (ms)	Loss Rate (%)
$4{\rightarrow}5$	1970	55	35	0.8
$0{\rightarrow}5$	39.2	198	405	16.2
$5{\rightarrow}0$	39.3	196	386	15.9

Table 2.3: Deficit Round-Robin does not Improve Fairness or Real-time QoS

channel, while flow 3 experience a busy channel during the time interval $[0, 1)$ and defers it transmission. Using Equation 2.6, the following capacity allocations are made:

$$W_1[0,1) = W_2[0,1) = \tfrac{1}{2}; W_3[0,1) = 0$$
$$W_1[1,2] = W_2[1,2] = W_3[1,2] = \tfrac{1}{3}.$$

However, over the time window $[0, 2]$:

$$W_1[0,2] = W_2[0,2] = \tfrac{5}{6}, W_3[0,2] = \tfrac{1}{3}$$

which fails the fair criterion. Using this explanation, it is seen that because it is only one hop away from the gateway, flow $4{\rightarrow}5$ will likely experience better (location-dependent) channel conditions than the other flows (0$\rightarrow$5 and $5{\rightarrow}0$) and consequently violate fairness.

Traditional service disciplines used in wireline networks do not take location dependent channel conditions into account. When applied in a wireless environment where channel conditions are much more dynamic, they fail to be effective. There are wireless fair queueing disciplines (WFQD) [10] such as: wireless fair service (WFS), server-based fairness approach (SBFA), idealized wireless fair queueing (IWFQ), and channel condition independent fair queueing (CIFQ). These algorithms use a compensation model to compensate flows that experienced bad channel conditions. In addition, they have methods that continually monitor and predict channel conditions that increase scheduling efficiency and support for delay-sensitive and error-sensitive flow decoupling.

If the service discipline is channel-condition aware such as the above WFQDs and it is work-conserving, low delay is not guaranteed because of the coupling between bandwidth and delay as discussed in Section 2.3.2. Flow $4{\rightarrow}5$ in Figure 2.5(a) is an infinite TCP flow that is causing congestion in the network. A (channel-condition aware) work-conserving fair scheduler would simply distribute the effects of congestion, increasing the delays and packet loss rates of every flow. What is needed is a non-work-conserving discipline which inherently decouples bandwidth and delay, such that the rates of TCP flows are controlled regardless of the bandwidth (capacity) available. Guaranteeing QoS for real-time flows becomes a factor of controlling the load in the network below congestion.

For wireless mesh networks, it has been shown that, by themselves, fair-queueing, priority queueing and AQM mechanisms such as RED (and its variants) do not solve the problem [35, 36, 48, 51].

2.3.8 QoS in Multihop Networks

QoS provision primarily relies on reserving resources and ensuring that the reservation is met throughout the lifetime of the QoS sensitive traffic [18]. Research on QoS support in multihop networks include [104]: QoS models, resource reservation signalling and or admission control, QoS routing, QoS MAC, and more recently, packet aggregation.

QoS models such as [105], define the architecture to provide certain services within the network. Traditional QoS models include: Integrated Services (IntServ) [85], Differentiated Services (DiffServ) [11] and Resource Reservation Protocol (RSVP) [118]. A QoS model for a multihop wireless network should consider the inherent challenges of time varying link capacity due to fading channels and broadcast nature of the wireless spectrum, topology, and the effects from type of traffic.

Signalling whether implicit or explicit is necessary for QoS reservation. A QoS signal is a control message to an entity in the network that directs any operation that achieves reservation or negotiation of resources. QoS signalling coordinates the behaviour of QoS routing and MAC and the admitting and scheduling of flows. It is necessary that the signal is reliably and efficiently transmitted with minimal delay because it is the transmission of a control message. Signalling can be in-band or out-of-band. In-band refers to signalling that is done on the same channel or path as the data. For better efficiency control packets can be piggybacked with data packets. Out-of-band signalling uses explicit control packets over an exclusive channel or path.

Admission control is an integral part of QoS reservation. An admission control protocol or algorithm must accurately assess the current resource utilization of flows in the system to determine whether new flows can be allowed without impacting the QoS guarantee of the existing flows. In a wired network, where the view of the communication medium or physical wire is the same for all nodes sharing it, admission control is simpler. The shared wireless medium do not provide such a unified view; each node perceive different levels of contention and interference from another. Admission control is thus more difficult in wireless multihop networks as the assessment of resource usage now requires the communication of each node's view and or neighbouring nodes' views. For example, a wireless multihop network may be lightly loaded then suddenly overloaded if a new flow's hop-count is large, consuming plenty channel resource. For example, the admission control scheme in [58] that used packet-marking to characterize load by computation, admitting a connection if lightly loaded is not directly applicable to wireless multihop networks. Lee *et al.* [66] proposed INSIGNIA, which is a framework based on in-band signalling and soft-state setup for ad hoc networks. The framework utilizes existing systems for admission control, packet forwarding, routing, and scheduling and is transparent to any MAC. The signalling control data is carried in the IP option of every IP data packet to enforce per flow resource reservation. Yang and Kravets [111] propose a contention-aware admission control protocol (CACP) for mobile ad hoc networks. The idea in that paper is that a node should consider both local resources and resources of its contending neighbours (c-neighbours) since it may consume their resources through contention. Nodes within carrier sense range are queried to determine whether new flows can be admitted. Chakeres and Belding-Royer [17] proved that CACP makes unnecessarily bandwidth reservations. They instead proposed perceptive admission control (PAC) that estimates available bandwidth by adjusting the carrier sense range to measure the channel busy time.

The intent of QoS routing is to search for network paths that have sufficient resources to meet the QoS requirements of admitted flows. Paths should be selected such that resource utilization is globally efficient [18]. QoS routing is different from routing protocols such as destination-sequenced distance vector (DSDV) [82], dynamic source routing (DSR) [49], zone routing protocol (ZRP) [41], and optimized link-state routing (OLSR) [20] that focus on shortest-path and achieving a high availability due to topology dynamics. QoS routing protocols must coordinate with a resource manager to establish paths that meet end-to-end QoS requirements, including costs. Sivakumar *et al.* [89] proposed a core-extraction distributed routing algorithm (CEDAR) for QoS routing. A self-organizing routing infrastructure called the core covers the network as every node in the core covers its n-neighbourhood. The core enables efficient route computation, and along with link state propagation enables QoS support in ad hoc networks. Chen and Narhstedt [18] proposed ticket-based algorithms for QoS routing. The algorithms are made to utilize imprecise state information to find routes that satisfy delay or bandwidth requirements. A ticket is considered permission on behalf a node to search paths that could satisfy the required QoS. Multiple paths are searched in parallel to find the best one. The number of tickets issued determine the amount of paths to search and overhead. Xue and Ganz [108] proposed ad hoc QoS on-demand routing (AQOR) that combines admission control and bandwidth reservation in routing. QoS is maintained through temporary reservation and destination-initiated recovery procedures. Yin *et al.* [113] proposed a traffic aware routing metric called path predicted transmission time (PPTT). PPTT is feasible with single and multi-radio configurations. It considers the impact of interference from self-traffic and neighbouring real time communication (RTC) traffic which results in a more accurate estimation of path transmission delay. PPTT capable of choosing high quality paths for RTC flows as it is able to distinguish links using wireless channels and radios.

In addition to the operation of resolving medium access contention and reducing hidden and exposed terminals effects, a QoS MAC such as [5, 17, 55, 56, 74, 90, 91, 98, 109, 101] provides resource reservation and QoS guarantees to real-time traffic. The 802.11e [76] draft extends the 802.11 protocol to achieve QoS functionality within the DCF by appropriately adjusting the CW and inter-frame spacing. In this thesis, we focus on achieving soft real-time guarantees with ubiquitous 802.11 hardware. A new MAC would require new or upgraded hardware which incurs a significant increase in costs. Kravets [112] model the delay caused by 802.11 MAC contention and state that a significant portion of the end-to-end delay is caused by the contention delay.The probabilistic model was used to create a Distributed Delay Allocation (DDA) algorithm that provide average delay guarantees by allocating a contention window size for each node according to the delay requirements of each flow.

As previously discussed, interactive real-time applications have very small packet payloads that can result in bandwidth under-utilization or inefficiencies. Aggregation, which is the combination of smaller packets into a larger packet is a effective method of improving the utilization. Aggregation can occur at the source or at ingress nodes. A simple implementation such as in [60] can incur additional delay over-head when waiting for packets to aggregate. An improved implementation such as [45] utilizes the natural queueing or system delays to aggregate back-logged packets. Niculescu *et al.* [78] improved the latter scheme with header compression to achieve a significant increase in the number of VoIP calls supported in their 802.11b testbed. In 802.11-based multihop networks, packet aggregation is particularly beneficial as there are significant gains in bandwidth utilization with just a small increase in packet transmission times.

2.4 Related Work

In this section we review related work that take a practical approach to realizing soft-real time guarantees with competing best-effort traffic. It is important to note that much research have focused on guaranteed throughput (*e.g.*, [51]) or have ignored the effects of competing traffic classes. In addition, as presented in our background, guaranteed throughput is not sufficient for guaranteed real-time performance as the delay requirement could still be unachieved. We therefore discuss work that focus on providing real-time performance with respect to delay, jitter, and packet loss.

Ahn *et al.* [2] proposed a service differentiation scheme that achieves real-time QoS by source rate limiting best-effort traffic with probe-based admission control. Their approach assumes a stateless wireless ad hoc networks (SWAN) model. SWAN performs local rate control for UDP real-time and TCP traffic, and sender based admission control for UDP traffic. ECN marking is used as a signal to regulate UDP traffic if the network becomes overloaded. TCP is rate limited with respect to the per-hop MAC delay measurements from packet transmission using an AIMD discipline. SWAN does not depend on per flow or aggregate state information and rely only on measurements derived from querying the MAC.

It has been shown in [110] that an admission and rate control scheme based on delay measurements can perform poorly since traffic delay is related to the packet scheduling between competing traffic at neighbouring nodes [112]. Delay measurements can be inaccurate whenever a new flow is admitted since the previous measured packet delay is much smaller. When the new packet is admitted it can severely increase overall packet delay which can fail to guarantee real-time QoS. Moreover, because it is stateless, SWAN can be inefficient. Different real-time applications require different delay metrics to be met. However, because there is no per flow state, the smallest delay have to be unnecessarily maintained across all real-time applications with different requirments. This could cause an unnecessarily low TCP rate.

Wu *et al.* [103] proposed SoftMAC, which is a collaborative software MAC position between layer 2 and 3, that supports multimedia for 802.11-based multihop wireless networks. The scheme uses distributed soft-state admission control and strict priority queueing for real-time traffic and rate limiting for best-effort (BE) TCP traffic. These mechanisms are based on the notion of fraction of air time (FAT) that considers the time costs to deliver a packet from a source to destination. The costs include the overhead time for carrier sensing, back-off, MAC-frame ACK, and retransmissions. FAT essentially translates into an estimation of bandwidth requirement for flows. FAT is of two forms: consumed and residual FAT, which are defined as the fraction of total air time consumed or available in a given time interval to the length of the interval. On request for admittance, the consumed FAT is computed and compared to the residual FAT. If the consumed FAT is less than the residual FAT, the flow is admitted. Similarly, the residual FAT is used to gauge the rate limiting of the BE traffic. The FAT computation takes into consideration the impact on existing flows within a neighbourhood of nodes (*i.e.*, nodes within communication range). Each node periodically broadcasts its FAT allocations. A broadcast is also triggered when consumed FAT is reserved or released for flows. Each node uses the broadcasted information to estimate bandwidth availability and govern flow admittance. Priority queueing is used for service differentiation. Real-time traffic is given priority over BE traffic, while the control signalling traffic is of the highest priority. Though the FAT concept is very useful, for implementation, there is trade-off between signalling overhead and guaranteed performance. SoftMAC broadcasts information which communicates channel conditions, such as frame loss probability and link capacity that is necessary for the FAT prediction.

SoftMAC is similar to equation-based congestion control mechanisms. It has the advantage of providing smooth rate control because bandwidth allocations are computed. However, such a schemee is slow to respond to network dynamics and incipient congestion. In addition, successful bandwidth allocation is dependent on accurate measurements and fast and reliable transmission of these measurements. Accurate measurements require suitably large observation periods. However, large observation periods cannot be granted if we want a system to be suitably responsive. Therefore, SoftMAC must trade off accurate measurements for responsiveness which reduces its efficient. SoftMAC admission control can cause inefficiencies because it wrongly denies flows that should be admitted as presented in the paper. Similarly, the measurement of physical link capacity via probing underestimates the true link capacity as the link rates increases (see Fig. 6

in [103]). For example, at 54 Mbps link capacity, the measured capacity is 45 Mbps reducing the bandwidth for real-time and BE flows. It is evident that the delay results for the real-time flows are influenced greatly by the use of strict priority queueing. Priority queueing presents a disadvantage to BE flows.

A drawback of SoftMAC and SWAN are the assumptions of interference. SWAN does source-based admission control, while the intermediate nodes have the ability to rate limit flows whether TCP or real-time flows. Based on that paper, there is the implicit assumption that TCP traffic is always on the same path as the real-time flows, ignoring interference or busy conditions from TCP traffic that is not local to that path. If the interfering TCP traffic is not rate controlled, real-time QoS can be disrupted. Admission control is similarly affected. A smaller number of flows is admitted because of interfering and unlimited elastic TCP traffic that consume significant bandwidth.

The authors of SoftMAC, during the analysis of FAT, assume that interference, carrier sense and communication range are the same. In their simulation experiments (see Fig. 7 in [103]) the distance between neighbouring nodes, communication range, and carrier sense range is 24 m, 25 m, and 30 m, respectively. This set-up significantly diminishes the effect of interference. Interference effects are far more significant in a set-up with an interference or carrier sense range that is much larger than the transmission range and would cause poorer performance in rate and admission control. In addition, it is unclear if SoftMAC is operational in an 802.11-based multichannel network since the calculations of FAT would then involve non-neighbouring nodes.

2.5 Summary

Our critique of related work suggests the need for a system that is first, very responsive and reactive enabling soft real-time QoS, while ensuring that elastic TCP traffic throughput is optimal. The system must also achieve efficient network load management via rate and or admission control. Load control and general operation of the system must based on realistic assumptions about interference. The efficacy of the scheme should be transparent to the underlying MAC; it must be easily installed via software upgrades and easily ported for use with future MACs such as 802.11e or 802.11n [106]. In this thesis, we present mechanisms that try to achieve these objectives.

3 Real-time RED

RtRED is inspired by the original RED algorithm [32]. Like RED, an exponentialy-weighted moving average (EWMA) and linear probability curve is calculated and compared with respect to the minimum and maximum queue size thresholds. However, unlike RED, RtRED defines two classes, real-time and elastic, with respective queues. Voice, interactive-video, and MOG traffic of UDP and RTP protocols are classified as real-time, while long-lived FTP, HTTP and other TCP traffic are classified as elastic. The EWMA is maintained for the real-time queue, rtq. When the EWMA of the rtq exceeds the queue thresholds, following original RED, a drop or marking probability is calculated, but instead of marking or dropping packets from its own real-time queue, packets are dropped or marked from the elastic queue, elq as shown in Algorithm 1. In essence, rtq size influences packet drops or marking on the elq. When deployed, each mesh router node would have to be RtRED capable. The ns-2 implementation is shown in Figure 3.1.

As discussed in Section 2.3.3, queue size is a direct indicator of congestion. A large

Algorithm 1 RtRED Algorithm - Modified RED algorithm from [32]

```
For each packet arrival:
calculate the average real-time queue size, $avg$, of $rtq$
$avg = (1 - W_q)avg + W_q \times rtq$
if $min_{th} \leq avg < max_{th}$ then
    calculate probability $p_a$
    with probability $p_a$:
    if arriving packet is elastic then
        mark or drop the arriving packet
    end if
else if $max_{th} \leq avg$ then
    if arriving packet is elastic then
        mark or drop the arriving packet
    end if
end if
```

queue size means that packets have long waiting times before they are serviced. In this respect, RtRED monitors its rtq size to detect the congestion caused by competing TCP

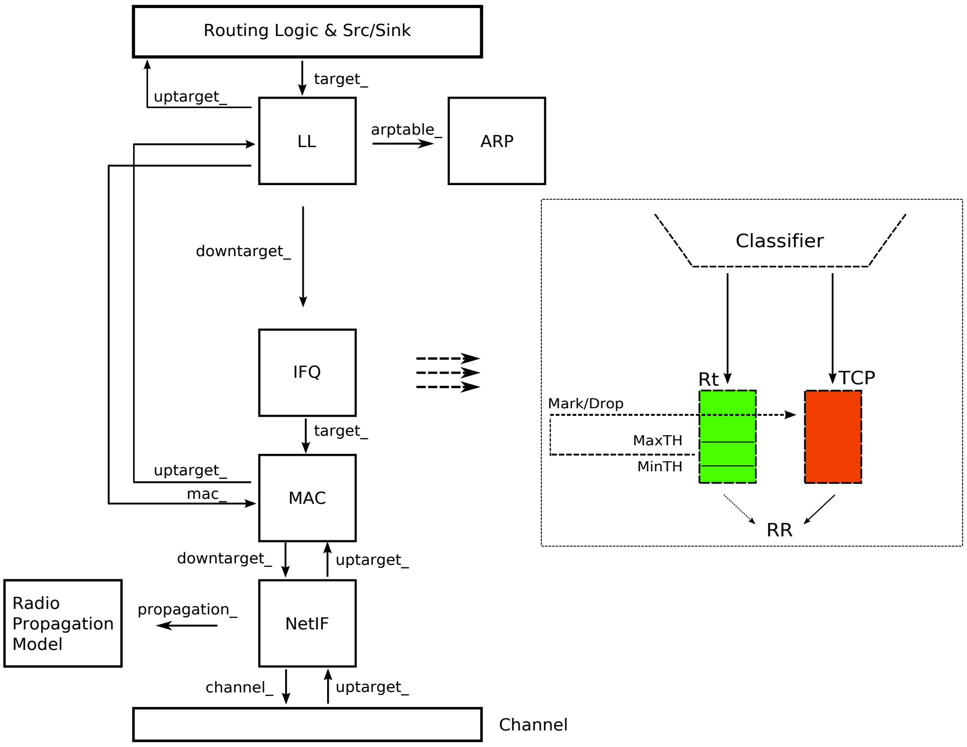


Figure 3.1: Ns2 RtRED Implementation

Parameter	**Setting**
Physical rate	11 Mbps
RTS/CTS	Off
Transmission Range	250 m
Carrier-sense Range	550 m
Radio Propagation	Two-ray Ground
Area	1000 × 1000 m^2
Queue Type	Queue/RtRED
Routing protocol	DSDV
Traffic Type	UDP, Infinite TCP(of 1500-byte packets)
Interactive real-time traffic	48 Kbps/60-byte packets per 10 ms
min_{th}	0
max_{th}	1
W_q	0.002

Table 3.1: RtRED Experiment Parameters

$S \rightarrow D$	Throughput (kbps)	Delay (ms)	Jitter (ms)	Loss Rate (%)
4→5	984.6	15	15	15.9
0→5	47.6	56	61	1.2
5→0	47.7	64	69	1.1

Table 3.2: The QoS achieved with RtRED

traffic. We leverage the best-effort nature of TCP, reducing its throughput via packet drops to increase the residual bandwidth for real-time traffic improving its QoS.

Using the set-up in Table 3.1, we evaluated RtRED's performance by simulating the simple scenario shown in Figure 2.5(a) [1] . The results are shown in Table 3.2. RtRED is not effective as desired as seen from the large one-way delays of the real-time flow. TCP also experiences significant packet loss. In an 802.11 environment, such packet loss wastes resources transmitting packets that will be dropped eventually, at an intermediate (forwarding) node, for example. RtRED performs poorly because of the TCP effects as discussed in Section 2.3.6. In addition, TCP does not reduce its sending rate quickly enough to allow the real-time flows sufficient access to the medium. TCP packet loss at the queues is detected by non-receipt of an ACK (or duplicated ACKs) within a certain

[1] min_{th} and max_{th} are set to 0 and 1, respectively. Although this may seem aggressive it results in the best real-time delay.

time period. Controlled by the sender's timer, when this period expires and an ACK has not been received for a packet, congestion is perceived and the sending window is reduced. The response time for TCP to reduce its sending window is dependent on its timer setting which is course-grained or not sufficiently rapid, causing back-log in the real-time queues.

Although real-time traffic QoS is within the desirable bounds for interactive applications, RtRED's performance is undesirable, considering the additional delay, jitter, and loss that would be encountered from the Internet and last-mile access to the other end-host. The resulting real-time traffic delay, using RtRED, is at least eight times as the real-time traffic delay, absent TCP (see Section 2.3.6). Moreover, when we simulate this scenario (Figure 2.5(a)), setting the TCP flow to the the residual maximum throughput (calculated from the theory presented in Section 2.1.3.1), real-time traffic delay is less than 15 ms. Further, RtRED fails to achieve real-time QoS when we simulate the scenario in Figure 2.5(b), where the TCP flow does not share paths with the real-time traffic, but interfers with the real-time traffic.

The back-logged behaviour in the real-time queues, RtRED's ineffectiveness when there is interfering TCP traffic (that do not intersect with real-time traffic), and the desirable real-time QoS when we explicitly rate-limit TCP, give insight for a better solution. RtRED cannot slow down elastic traffic that may be causing interference because its design assumes that real-time and elastic traffic traverse the same nodes. Any approach to the problem must assume interference from elastic traffic transmissions. Further, it is necessary to communicate with the nodes carrying elastic traffic to coorperatively slow down their transmissions. We use this as motivation for a better approach which we discuss next.

4 Real-time Queue Rate and Admission Control

While RtRED was not as effective as desired, it did point to a possible solution, *viz.* using the real-time queue behaviour to control elastic traffic queue behaviour. We expand on that solution and thus propose Real-time Queue-based Rate and Admission Control, RtQ-RAC.

Improving on the RtRED scheme, RtQ-RAC is designed to support interactive multimedia application traffic from multiplayer online games *e.g.* FPS and VoIP services in wireless mesh networks[1]. In this regard, we assume that all traffic (or communication sessions) is between a wireles mesh node and an end-host in the Internet which implies that every flow traverses the wireless mesh gateway. RtQ-RAC is designed to work with commodity off-the-shelf 802.11a/b/g and future wireless network interface cards such as 802.11e and 802.11n without any modification to the standard MAC DCF. RtQ-RAC is a reactive QoS scheme. By using increase-decrease rate control algorithms and packet marking admission control,RtQ-RAC maintains network load of best effort and real-time traffic below congestion, allowing real-time traffic to achieve its desired QoS. The main idea and novelty is that rate and admission control is directed and influenced by real-time queues.

4.1 Architecture

RtQ-RAC employs a number of mechanisms to enforce soft real-time QoS as depicted in Figure 4.1. There are six major components of RtQ-RAC. The scheme is centered around the queue management component. By inspecting packet headers, the classifier sorts received packets, differentiating between control (*e.g.* routing messages, ACKs *etc.*), real-time, and elastic packets, buffering them in respective queues at the queue

[1] RtQ-RAC is applicable to other multimedia applications *e.g.* streaming-video and streaming-audio. However, these applications lack strigent interactivity requirements and use large buffering to smooth playback during significant delay and jitter. Large buffering is inappropriate for interactive applications as it increases delays which diminishes interactivity.

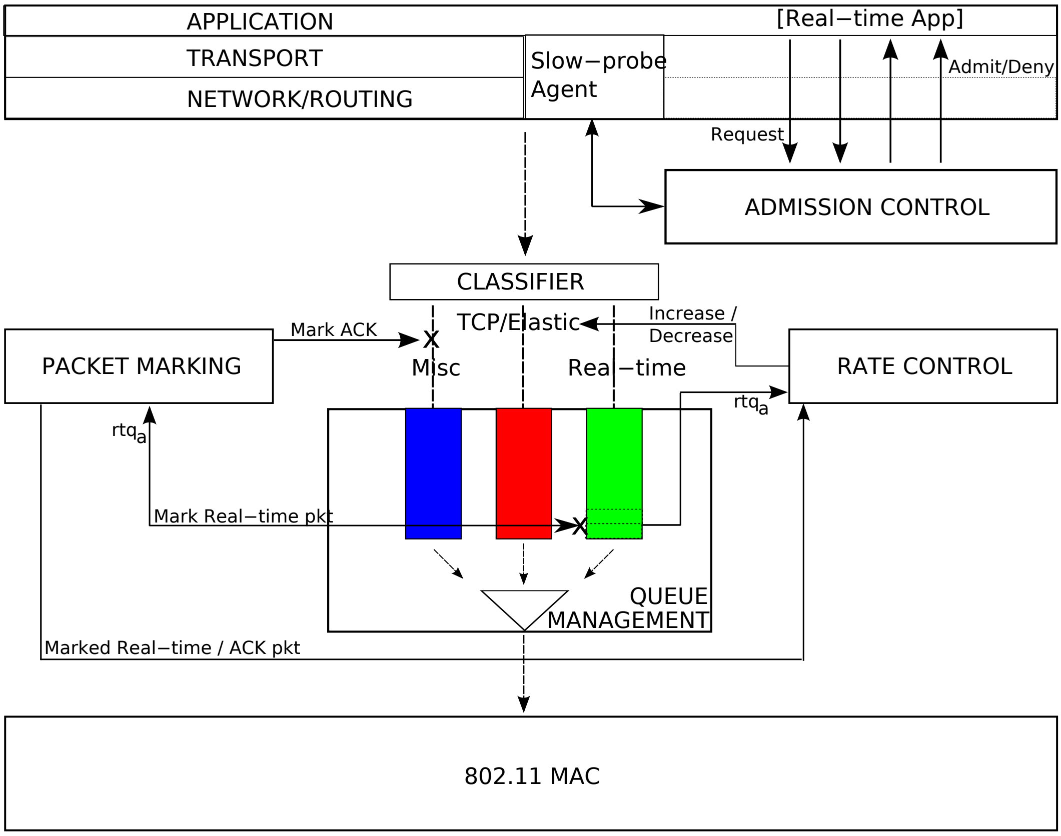


Figure 4.1: RtQ-RAC Architecture

management component. The average real-time queue size, rtq_a governs the rate control and shaping of elastic traffic and packet marking. The information gathered from packet marking is used in the admission control component. It is also used as a signal to enforce rate control of elastic sources or flows that do not share similar nodes with real-time traffic, but whose transmission may interfere with real-time traffic. A priority queueing discipline schedules control and data (real-time and elastic) traffic. Within the data traffic class, real-time and elastic traffic is round-robin scheduled. RtQ-RAC is designed such that rate and admission control exist in the control plane, while packet classification, queue management, and packet marking are data-plane functions. In addition, while rate-control functions exist on every mesh node, admission decisions are determined solely by egress mesh nodes that infer network state. The rate control mechanism continually reacts with respect to variation of rtq_a Elastic traffic is regulated by increasing or decreasing its shaping rate when the average real time queue size is below or above some threshold, minimizing its impact on existing real-time flows. The traffic shaper is a simple token bucket filter. By delaying the elastic packets with respect to the rate calculated by the rate controller, the traffic shaper reduces contention and congestion, ensuring that real-time traffic gets more air-time to transmit its packet.

The admission control component achieves two goals. First, it determines if there are enough network resources or bandwidth to fully support new requests. Second, it ensures that admitted flows are protected and conformant to their traffic specification. In wired networks, bandwidth, link, and path utilization are determined from the aggregate load of existing flows compared to the link capacity which are easy to obtain or to estimate accurately. In an 802.11-based multihop wireless network, it is more difficult because of continual variation of capacity caused by interference, hidden and exposed terminal problems, collisions, and MAC frame overheads. As discussed in Chapter 2, schemes that explicitly compute capacity under or over estimate residual capacity causing poor performing admission control. To address this, we use a reactive admission control scheme. Admittance is determined from the result of sending slow-rise bidirectional probe packets between the mesh source and mesh gateway if the connection originates within the WMN, and between the mesh gateway and mesh destination, if the connection originates from the Internet. In both cases, the gateway acts as a proxy for the remote host in the Internet.

The scheduling mechanism gives non-preemptive priority to control packets. If control packets are available they are sent before data packets. Packet-based round-robin scheduling is done for the elastic and real-time traffic. Non-preemptive priority queueing could also be used to give real-time traffic higher priority. However, it is unnecessary because network load is held under capacity and elastic traffic is in conformance with the rate governed by the real-time traffic.

There is no need for error-prone estimation of capacity and shaping rates in RtQ-RAC because it is completely reactive. The exact value of capacity or load is irrelevant; we simply need to know if real-time QoS is being degraded or if load is approaching the network's capacity. Admission control decisions and shaping rates are derived from real-time queue sizes that reflect residual capacity, whatever it is, without overloading the network. In other words, the correct characterization of residual capacity of a path is inferred from the real-time queues. Exceeding the real-time queue threshold size, rtq_{Th}

signals the onset of degraded real-time QoS and congestion which are communicated with other nodes by marking real-time packets and slow-rise probes.

4.2 ALGORITHMS

We now present the main control algorithms used in RtQ-RAC: 1) real-time queue directed rate control, RtQ-RC and 2) real-time queue influenced admission control, RtQ-AC.

4.2.1 RTQ-RC

The rate controller's purpose is to regulate the elastic traffic with respect to real-time (RT) traffic. In others words, the remaining bandwidth unconsumed by the real-time traffic is allocated to the elastic traffic. The rate controller sets the dynamic rate of the traffic shaper *i.e.* the token bucket filter using an increase-decrease algorithm, such as AIMD. Each node allocates a single real-time queue, rtq and an elastic queue, elq, if real-time and elastic traffic traverses it. rtq provides feedback to the rate controller. Each node independently regulates elastic traffic when shared with real-time traffic as shown in Algorithm 2. Nodes also operate cooperatively to regulate elastic traffic, if elastic traffic exists on different paths or nodes as shown in Algorithm 3. If real-time and elastic traffic do not share the same nodes or path, real-time packets are possibly marked based on the feedback from the rtq size, if that elastic traffic transmission causes contention. The marking operation is similar to what is done in AQM. However, unlike traditional AQM schemes, the queue size feedback enforces rate control. In addition, where traditional queue-size based AQM schemes mark or discard packets of that queue experiencing congestion, in this work a real-time queue representing a different traffic class is used to explicitly influence another.

Algorithm 2 allows elastic traffic rate to rise if the real-time queue is not experiencing congestion which occurs when the average real-time queue size, $rtq_a < rtq_{Th}$. If congestion is experienced then elastic traffic is decreased progressively with respect to the factor, fac. If elastic traffic does not traverse that node, real-time packets are marked for use in Algorithm 3. In Algorithm 3, the marked real-time packets are used to influence remote rate control. The gateway node, for example, when it receives a marked real-time packet will directly decrease elastic traffic, if it is the source, or mark the ce-bit in the

TCP ACK packet. A decrease of TCP rate is continually signalled until $rtcongested_i$ is false for all real-time flows, i. When a node receives an ACK, if its ce-bit is set, elastic traffic rate is decreased; otherwise elastic traffic rate is increased.

RtQ-RC's algorithms are effective when the aggregate load of the existing real-time traffic is below the network's capacity. It is possible, absent appropriate mechanisms, that too many real-time flows exist in the network, causing congestion and disrupting the QoS. Therefore, to sufficiently support real-time QoS, admission control is necessary. We describe our admission control scheme next.

4.2.2 RTQ-AC

RtQ-AC exploits the packet marking functionality of the above algorithm to influence admission control. Admission of a flow is negotiated between a mesh source and mesh gateway or between a mesh destination and mesh gateway. The gateway acts a proxy for the Internet destination or source, respectively. When a new flow needs to be admitted, the node (*i.e.* proxy gateway or mesh source) sends slow-rise probes up to the flow traffic specification (tspec). Reverse slow-rise probes are also sent from the destination towards the gateway since an interactive multimedia flow is simultaneously bidirectional. The slow-rising of the probes purpose is to influence the packet marking rate. The probe packets are treated as real-time packets, not control packets and thus share the same queue as real-time traffic. When an application request admittance, probe packets are generated to emulate the application's packets. The slow-rise mechanism is an AIMD function that is capped by the tspec rate and a slow probe timeout.

On a request, the source and destination each maintain an exponential weighted average of the ratio of marked real-time packets and total real-time packets, γ, received during the slow probe period; marked and unmarked real-time packets include the slow-rise probe packets. For each received real-time packet destined at the node, γ is compared with a defined admittance threshold, α as shown in Algorithm 4. If $\gamma < \alpha$, an increase of the slow-rise probes is performed, otherwise it is decreased according to the AIMD function. When the slow probe period expires the current rate of the AIMD is compared with the tspec. The flow is admitted if that rate is close (within 5%) to the tspec; otherwise it is denied.

Algorithm 2 RtQ-RC : Local

```
Init:
$rtq_{Th} = 1; rtq_a = 0$
$W_{rtq} = 0.125; fac = 0$
On enqueuing a packet:
calculate EWMA $rtq_a$ of $rtq$:
if $rtq \neq null$ then
    $rtq_a = W_{rtq} * rtq + (1 - W_{rtq}) \times rtq_a$
    $rtcongested = false$
    if $rtq_a \leq rtq_{minTh}$ then
        $W_{rtq} = 0.125$
        $fac = rtq_{minTh} - rtq_a$
    else if $rtq_{minTh} < rtq_a < rtq_{Th}$ then
        $W_{rtq} = 0.6$
        $fac = 0$
    else
        $W_{rtq} = 0.875$
        if $rtq_a > rtq_{maxTh}$ then
            $fac = 0.5$
            $rtcongested = true$
        else
            $fac = 1 - (rtq_a/rtq_{maxTh}) \times 0.5)$
            $rtcongested = true$
        end if
    end if
    if $elq \neq null$ then
        if $rtcongested = true$ then
            decrease rate based on $fac$
        else
            increase rate based on $fac$
        end if
    else
        mark real-time packets:
        if $rtcongested = true$ then
            mark $rtq$ head packet
        end if
    end if
end if
```

Algorithm 3 RtQ-RC : Remote

```
On packet arrival:
if RT packet then
    if marked then
        $rtcongested_i = true$
    else
        $rtcongested_i = false$
    end if
end if
if TCP-ACK and destination then
    if ce-bit set then
        decrease rate
    else
        increase rate
    end if
end if
On sending a packet:
if TCP-ACK then
    if any $rtcongested_i$ then
        set ce-bit
    end if
end if
if TCP then
    if any $rtcongested_i$ then
        decrease rate
    end if
end if
```

Algorithm 4 RtQ-AC

```
On new admission request with $tspec$:
start slow-rise probe timer
while slow-rise probe timer pending do
  if RT-PROBE packet received then
    AIMD discipline of $rate$:
    if $\gamma < \alpha$ then
      if $rate < tspec$ then
        $rate = rate + additive$
      end if
      if $rate \geq tspec$ then
        align to $tspec$ rate:
        $rate = tspec$
      end if
    end if
    if $\gamma > \alpha$ then
      aggressively decrease rate as state of affairs is bad and to protect existing flows:
      $rate = rate \times 0.5$
    end if
  end if
end while
admit or deny flow:
if $rate \approx tspec$ then
  admit flow
else
  deny flow
end if
```

4.3 Increase-Decrease Algorithms

In this section we present the rate control disciplines. An increase-decrease algorithm direct the rate increase or decrease of elastic traffic and thus affects its utilization of residual bandwidth. Moreover, the increase-decrease algorithm affects real-time QoS since it determines the degree of rate penalization to elastic traffic that is causing congestion. We use additive-increase multiplicative-decrease (AIMD) and additive-increase additive-decrease (AIAD) disciplines for RtQ-RC. These algorithms determine a rate in bits per second for elastic traffic that is enforced by a respective token bucket filter (TBF).

4.3.1 Additive-Increase Multiplicative-Decrease

Additive-increase multiplicative-decrease (AIMD) is primarily used for congestion control algorithms. It is stable regardless of initial value [19]. It is used in TCP's congestion control mechanisms [79]. By default, we use an AIMD algorithm, shown in Algorithm 5 to direct rate control of elastic traffic. AIMD is usually used for self-traffic congestion management. In TCP, for example, its ACK return rate determines the increase or decrease its sending window. However, in RtQ-RC , the $additive$ and fac parameters that influence the amount of increase or decrease are relative to rtq_a which is orthogonal. On a node, for every packet arrival, rtq_a is updated; elastic traffic is allowed to gradually (*i.e.*, additively) increase until large real-time traffic delays are detected via rtq_a (*i.e.*, the average real-time queue size is large; $rtq_a > 1$). When this occurs, TCP elastic traffic rate is quickly lowered by multiplicative decrease relieving the congestion and improving real-time QoS. On nodes where only TCP traffic exists, AIMD control is governed indirectly by rtq_a via real-time packet marking. Nodes that receive marked real-time packets set respective flow variables in $rtcongested_i$ to maintain state and set the ce-bit of TCP ACK packets. Elastic TCP traffic is multiplicatively reduced, if any $rtcongested_i$ variable or an received ACK packet's ce-bit is set.

4.3.2 Additive-Increase Additive-Decrease

Additive-increase additive-decrease (AIAD) is not as common as AIMD. In RtQ-RC, it is used as an alternative rate control discipline. The AIAD decrease of elastic traffic rate is less aggressive than that of AIMD because it is such that the rate is reduce by negative additive instead of a multiplicative factor. As shown in Algorithm 6, with respect to

Algorithm 5 AIMD for RtQ-RC

```
On RT packet arrival:
if INCREASE then
    $rate = rate + additive \times fac$
end if
if DECREASE then
    $rate = rate * fac$
end if
return $rate$
```

rtq_a, its operation is similar to the AIMD operation in Algorithm 5; the rate decrease in AIAD is determined by a $decrement$ which is calculated from the TCP packet size in bits, $TCPPKTSIZE$ and the inverse of fac.

Algorithm 6 AIAD for RtQ-RC

```
On RT packet arrival:
if INCREASE then
    $rate = rate + additive \times fac$
end if
if DECREASE then
    $decrement = TCPPKTSIZE \times \frac{1}{fac}$
    if $decrement < rate$ then
        $rate = rate - decrement$
    end if
end if
return $rate$
```

4.4 Discussion

In this section, we extend our presentation of RtQ-RAC to address issues related to the assumptions, parameter variables, advantages and implementation of its algorithms.

4.4.1 Parameter Settings

The average, rtq_a is used to influence the increase-decrease algorithms for elastic traffic rate control.

Denote $\alpha(t)$ and $\delta(t)$ as the number of arrivals and departures in the interval $(0,t)$ at some time, t. Now,

$$N(t) = \alpha(t) - \delta(t) \tag{4.1}$$

is the number of jobs or items in the system at time, t [61]. The total area between $\alpha(t)$ and $\delta(t)$ is the total time the jobs have spent in the system *i.e.*, job-seconds during the interval $(0,t)$ and is denoted as $\gamma(t)$.

$$\lambda_t = \frac{\alpha(t)}{t} \tag{4.2}$$

is the average arrival rate during $(0,t)$. Since $\gamma(t)$ is the accumulated job-seconds up till t, the system time per job averaged over all jobs in $(0,t)$, T_t is:

$$T_t = \frac{\gamma(t)}{\alpha(t)} \tag{4.3}$$

Now, the average number of jobs in the queueing system during $(0,t)$, $\bar{N}_t$ is:

$$\bar{N}_t = \frac{\gamma(t)}{t} \tag{4.4}$$

Therefore,

$$\bar{N}_t = \lambda_t T_t \tag{4.5}$$

i.e., the average number of jobs in the queue is the average arrival rate times the system time per job.

Assuming, $\lambda = \lim_{t\to\infty} \lambda_t$ and $T = \lim_{t\to\infty} T_t$,

$$\bar{N}_t = \lambda T \tag{4.6}$$

Equation 4.6 is Little's result *i.e.*, the average number of jobs in a queue system is equal to the average arrival rate of jobs to that system times the average time spent in that system. It is independent of the inter-arrival time, the service time, the number of servers and the queueing discipline [61].

Representing jobs as packets, rtq_a represents equation 4.6 and captures the total system time since a real-time queue is layered above the underlying 802.11 MAC operations (*e.g.,* packet transmissions, backoffs and busy-channel timer deferrals). We want minimal delays subject to these MAC operations. When $\bar{N}$ *i.e.,* rtq_a, is greater than 1, it means the system is unable to service jobs (in our case, real-time packets) as fast as they arrive, creating a back-log. In other words, if the utilization factor, ρ, which is the average arrival rate times the average service time (*i.e.,* $\frac{load}{capacity}$) is greater than 1, then the system is unstable, increasing delays [15, 61]. Therefore, local rate control (Algorithm 2) operates by allowing the elastic traffic rate to rise as long at the real-time queue is not experiencing congestion (*i.e.,* the average real-time queue size is not large; $rtq_a \leq rtq_{minTh}$). If the real-time queue size grows but the real-time traffic is still being serviced ($rtq_{minTh} < rtq_a < rtq_{Th} = 1$), then no further increases are permitted in elastic traffic. If the average real-time queue size exceeds 1, then the arrival rate is greater than the service rate; congestion is declared and the elastic traffic rate is cut, progressively further as the average real-time queue size increase. Elastic TCP traffic is maximally reduced (by half) if $rtq_a > rtq_{maxTh}$. We have found by experiment that $rtq_{minTh} = 0.6$ and $rtq_{maxTh} = 5$ yield good results.

In addition, we have found it necessary, when computing the average queue size, to use an exponentially weighted moving average, where the weighting factor, W_{rtq}, is adjusted upward according to the queue size, causing a faster reaction to congestion. Similarly, the factor, fac, by which we increase and decrease the TCP traffic rate is also adjusted according to the queue size. We have found that varying these parameters improves the efficiency of TCP traffic in using residual bandwidth.

The value of α determines the strictness of RtQ-AC. We have found that when $\alpha > 0.3$, admission control is too lenient, admitting more RT flows than the network can handle. An α of between 0.25 and 0.3 was usually sufficient, preventing RT flows from being admitted that would interfere with the QoS requirements of existing flows, while allowing those that would not cause problems. When $\alpha < 0.2$ admission control is stringent, denying flows that could be admitted and achieve their desired QoS. The $additive$ parameter is kept low, typically 1000 bps, to slowly probe for bandwidth. Finally, we usually set the slow-rise probe time to 2 seconds; however, it can be derived from an initial slow-rise rate (with respect to tspec rate) and the value of $additive$.

4.4.2 System Implementation

In this section, we briefly describe a feasible implementation of RtQ-RAC according to the architecture shown in Figure 4.1. The rate control and admission control modules can be implemented using current mechanisms available in the Internet protocol stack and by modifying existing 802.11 drivers *e.g.*, Madwifi Atheros [84]. We describe the relevant module implementation below.

4.4.2.1 Queue Management and Rate Control

We remove the buffering at the MAC layer to support the implementation of the queue management and rate control modules. Therefore, implemented in the driver, a packet will be pushed down from our managed queues to the MAC layer only when there is no pending packet. In other words, we keep the NIC's buffer length at a maximum size of one packet, pushing packets one-by-one from RtQ-RAC's queue to the NIC's buffer. We classify packets using the type-of-service (TOS) field in the IP header, which supports differentiated services (DiffServ) and explicit congestion notification (ECN). Classified packets are placed into respective queues as shown in Figure 4.1 and are non-preemptive priority and round-robin scheduled. We use a token bucket to enforce rate control of TCP elastic traffic according to Algorithms 2, 3, and 5 or 6.

4.4.2.2 Packet Marking for Remote Rate and Admission Control

We utilize packet marking for admission and remote rate control. We implement packet marking using the congestion experienced bit (ce-bit) of the IP header. Marking can also be done on real-time packets that have the RTP (Real-time protocol) header. Using Algorithm 2, a real-time packet is marked by setting the ce-bit, if real-time traffic is perceived to be experiencing congestion. Similarly, a TCP ACK packet ce-bit is set, if at least one $rtcongested_i$ state variable is true. We implement our slow-rise probing module in the network layer parallel to routing, however, only egress mesh routers perform slow-rise probing. Slow-rise probing generates UDP packets that emulate the real-time packets of applications that are requesting admittance. It maintains the exponential weighted average of the ratio of marked probe packets and total probe packets received during the slow probe period, which is used as the metric to decide on admittance. Rate and admission control is enforced using packet marking as discussed in Section 4.2.1.

4.4.2.3 State Clean-up

There must be a mechanism for cleaning up per-flow state, in particular for our remote rate control algorithm (Algorithm 3). Cleaning-up the state variables in $rtcongested_i$ is very important to prevent the over-penalization of elastic TCP traffic if a real-time flow, i (or a respective node) dies unexpectedly leaving the respective $rtcongested_i$ variable set to true. A state clean-up operation can be done based on a course granular time-out mechanism that checks the amount of packets per flow it received in an interval. Rather, a state clean-up could simply just delete $rtcongested_i$ in random time intervals, between 100 ms to 500 ms, for example. Once another real-time packet is received from any active real-time flow, $rtcongested_i$ will be re-created, automatically deleting state information for inactive real-time flows.

4.4.3 Benefits and Drawbacks

As is evident from its algorithms, RtQ-RAC is simple. Fundamentally, it is based on monitoring and reacting (via rate and admission control) to real-time traffic queue sizes. In comparison, SoftMAC, for example, needs to measure the frame loss probability and the physical link capacity and exchanges, communicating this data via broadcasts. Accurately measuring such metrics is not trivial in comparison to measuring queue lengths. Moreover, the need to broadcast information in a neighbourhood of nodes, increases the communication and media contention overhead reducing capacity. Similarly, SWAN imposes communication overhead in its admission control scheme. RtQ-RAC is easily implemented above and works in general with any CSMA/CA MAC because it is queue-based. A queue can always be maintained above the wireless NIC's buffer directing its QoS functions. In addition, unlike SoftMAC, RtQ-RAC's efficacy is unaffected by multichannel deployments. Moreover, because an average real-time queue size is maintained, efficiency is unaffected by real-time flows of different loads. SWAN's efficiency is affected by flows of different loads as discussed in Section 2.4. Using the queue size as feedback enables RtQ-RAC to react to incipient and transient congestion conditions. Moreover, efficiency is improved during rate control as residual bandwidth can be effectively reserved without being conservative. In addition, other techniques such as packet aggregation is unaffected by RtQ-RAC.

RtQ-RAC, SWAN, and SoftMAC are at a disadvantage with respect to parameter tuning. However, once properly done, tuning is only an issue when the WMN topology changes drastically, which we assume to be very rare. We can enhance RtQ-RAC with additional self-tuning functionality to mitigate possible performance effects. In RtQ-RAC's admission control, there is a waiting time during slow-rise probing before deciding admittance. However, waiting (*i.e.*, probing-time) can be adjusted to very low values as discussed.

4.5 Summary

Compared to previous work RtQ-RAC utilizes a simple real-time queue class and packet marking to enforce rate and admission control. We claim novelty in using average real-time queue size to enforce local and remote rate control on elastic traffic. Our slow-rise probing approach incrementally test network state to decide admittance, based on the dynamic ratio of marked and total real-time packets.

5 EVALUATION

In this chapter, we assess the efficacy of RtQ-RAC in achieving soft real-time guarantees and interactivity in WMNs. We implement RtQ-RAC and assess its performance by simulations, using the Network Simulator, ns-2 [1]. For the most part, soft real-time performance is quantified with the three metrics: average end-to-end delay, jitter, and packet loss ratio. Further, we evaluate the efficiency of RtQ-RAC in regards to elastic traffic usage of residual bandwidth, determinining whether elastic traffic achieves good throughput.

Next, we describe the experiment design, where we detail the simulation configuration, topologies, traffic models and measurement techniques used in evaluating RtQ-RAC. After, we report the results of the experiments that show that RtQ-RAC is effective.

5.1 EXPERIMENT DESIGN

We now detail the experiment setup as follows: ns-2 configuration, topologies, traffic models and measurement techniques which is summarized in Table 5.1.

5.1.1 NS-2 CONFIGURATION

The network parameters are configured for an IEEE 802.11b DCF operating at 11 Mbps physical rate with RTS/CTS disabled. The transmission and interference range is 250 m and 550 m, respectively. A default distribution of ns-2 implements three radio propagation models: free space, two-ray ground and shadowing models. A radio propagation model generates a received signal to noise ratio at the receiver that determines whether a packet is correctly decoded or successfully received. We use the two-ray ground model which is adequately accurate as it models direct and ground-reflection path effects.When simulating RtQ-RAC, we utilize our own queueing management scheme. Real-time queues are limited to 50 packets; elastic queues are also limited to 50 packets but are divided equally between the TBF queues (that are designated for rescheduling) and the interface queues. Otherwise, when simulating only the default DCF, the queue is limited to 50 packets and adheres to the default drop-tail discipline. The destination-sequenced

Parameter	**Setting**
Physical rate	11 Mbps
RTS/CTS	Off
Transmission Range	250 m
Carrier-sense Range	550 m
Radio Propagation	Two-ray Ground
Area	1000 × 1000 m^2
Default Queue Type	Queue/Drop-Tail
Default Queue Size	50 packets
Routing protocol	DSDV
Traffic Type	UDP, Infinite TCP(of 1500-byte packets)
Interactive real-time traffic	48 Kbps/60-byte packets per 10 ms; Q3 model

Table 5.1: Experiment Baseline Design Parameters

distance vector (DSDV) routing protocol is used to automatically assign routes to the mesh gateway. Across all experiments, we allow a 1000 second warm-up time for routing convergence, during which no data is collected. We do this to mitigate the effects of routing and miscellaneous packets, such as address resolution protocol (ARP) packets.

5.1.2 TOPOLOGIES

In our simulations, we evaluate performance across variations of: chain, grid and random topology mesh networks. We use single chain topologies for simple analysis. Increasing the complexity of the analysis, we utilize grid topologies. Grid topologies sufficiently emulate realistic WMN topologies in office-type environments and communities, where office, roads and houses are arranged in a grid-type fashion. The distance between adjacent nodes are the same at 200 m. Finally, we generate random topologies by using special scripts. Each random topology, created from a different seed is located within a 1000 × 1000 m^2 area. In the grid and random topologies that we create, flows originate at green nodes and terminate at red nodes while a grey line connecting two nodes indicates they are within interference range.

5.1.3 TRAFFIC MODELS

Interactive real-time traffic is bidirectional, denoted as $r(i, j)$, representing the flow between node i and node j, where j is the gateway node. For example, $r(3, 5)$ represents the flows $3 \rightarrow 5$ and $5 \rightarrow 3$. Elastic (TCP) flows are similar represented as $e(i, j)$ and is considered unidirectional, ignoring ACKs. It is important to note that while certain interactive application, such as VoIP, generate equal load in either direction, FPS traffic loads are very different. FPS client traffic loads are usually higher and more dynamic with varying bit-rate and packet sizes than the server. In most instances, FPS server and VoIP traffic is adequately modelled as constant bit-rate (CBR) traffic with an upper-bound on the load. In our evaluation, we show the separate component of each multimedia flow. For example, $r(4, 2)$ is shown separately as $2 \rightarrow 4$ and $4 \rightarrow 2$.

Internet low bit-rate codec (iLBC), G.723, and G.729 are popular VoIP codecs. iLBC is used in Skype [15]. It has a codec bit-rate of 15.2 Kbps if packetizing at 20 ms and 13.3 Kbps at 30 ms intervals, which is respect to 38-byte and 50-byte payload. VoIP traffic is mostly made up of UDP/RTP packets. Factoring in the RTP application header, the total 40-byte header overhead consists of RTP (12 bytes), UDP (8 bytes), and IP (20 bytes). Therefore, with its payload, 20 ms-iLBC is be seen as UDP CBR traffic of 78-byte packets at a rate of 31.2 Kbps. However, we model interactive real-time flows as constant bit-rate (CBR) traffic of 60-byte packets every 10 ms, which has a rate of 48 kbps. In addition to simplifying analysis, we believe this model is adequate because the differences in actual air-times between small packets are negligible. We are also unaware of smaller packet intervals than 10 ms; FPS server packet intervals are significantly larger, at 50 ms to 70 ms, for example. Therefore, we model worst-case traffic for VoIP and FPS applications. For completeness, we also implement and run simulations with the Quake III model described in [63]. Elastic traffic is TCP traffic modelled as infinite FTP flows of 1500-byte packets and can originate or terminate at the gateway node.

5.1.4 EVALUATION CRITERIA

For VoIP, the International Telecommunication Union (ITU) recommends a one-way end-to-end delay no greater than 150 ms for good voice quality calls; the network delay budget is about 80 ms, while the packet loss rate should be less than 10% [44]. There is a limit of 400 ms for acceptable voice calls [44]. A WMN is the first or last mile access network. We thus employ stricter requirements of 65 ms network delay budget and a less than 5% packet loss to consider the additional delay, jitter and loss from the Internet and from the last-mile access network to end-hosts.

FPS games have stricter requirements, requiring a round-trip time (RTT) that is no greater than 150 ms. The best experience is attained at a RTT no greater than 75 ms. There is no study we are aware of to indicate at what point loss rates impact perceived quality. Subjective experiments within our lab suggest that it can be quite large (10% or higher) for FPS games. Using similar reasoning as in the above, we employ stricter requirements with a RTT no greater than 120 ms, 60 ms RTT for best experience, and a loss rate no greater than 5%.

5.2 RtQ-RC Performance

In this section, we evaluate the performance of the rate control mechanism, RtQ-RC. First, we look at the results when RtQ-RC is operational in the example scenarios from Section 2.3.6. Second, we report the results when it is applied to an example grid topology. Finally, to evaluate its overall performance, we run RtQ-RC in over 30 randomly generated 25-node topologies, with randomly generated flows. The simulations run for 100 seconds.

5.2.1 Performance in Example Scenarios

Tables 5.2 show the significant improvements in real-time QoS when RtQ-RC is applied to the scenarios in Figure 2.5. Compared to RtRED, $e(4,5)$ has lower throughput, but $r(0,5)$ receives desirable real-time QoS. Our explicit rate control is faster reacting than that of the TCP congestion control mechanism, enabling better real-time QoS. Similarly, TCP's behaviour that is caused packet loss (see Section 2.3.6) are mitigated resulting in slightly lower but reasonable throughput than that of RtRED. Moreover, the use of the default AIMD discipline has signficant effect on reducing TCP throughput because of its mutiplicative decrease factor (see Sections 5.4.1 and 5.4.2). We also simulate RtQ-RC across a 3×6 grid shown in Figure 5.1. We limit the grid topology to 3×6 as larger grids causes significant intra-flow (self) contention and delay because of the large hop-count when placed at the far exterior of the grid. Flows $e(0,5)$, $r(6,5)$, and $e(12,5)$ compete for media in the grid. Figure 5.2 shows results for the end-to-end delay. RtQ-RC is very effective in slowing TCP traffic, improving real-time traffic QoS from one-way delays as high as 400 ms to lower than 20 ms. Jitter and loss also improve as shown in Table 5.3.

(a) QoS for Scenario A

$S \rightarrow D$	Throughput (kbps)	Delay (ms)	Jitter (ms)	Loss Rate (%)
$4 \rightarrow 5$	839.7	273	202	0.2
$0 \rightarrow 5$	47.98	12	10	0
$5 \rightarrow 0$	47.99	12	10	0

(b) QoS for Scenario B

$S \rightarrow D$	Throughput (kbps)	Delay (ms)	Jitter (ms)	Loss Rate (%)
$6 \rightarrow 5$	866.2	274	202	0
$0 \rightarrow 5$	47.96	10	10	0
$5 \rightarrow 0$	48.0	10	10	0

Table 5.2: The achieved real-time QoS with RtQ-RC

Configuration	Delay (ms)	Jitter (ms)	Loss Rate (%)
DCF with TCP	790	463	19.8
RtQ-RC with TCP	51	69	0.1

Table 5.3: Aggregate real-time QoS Improvements with RtQ-RC in 3×6 grid topology

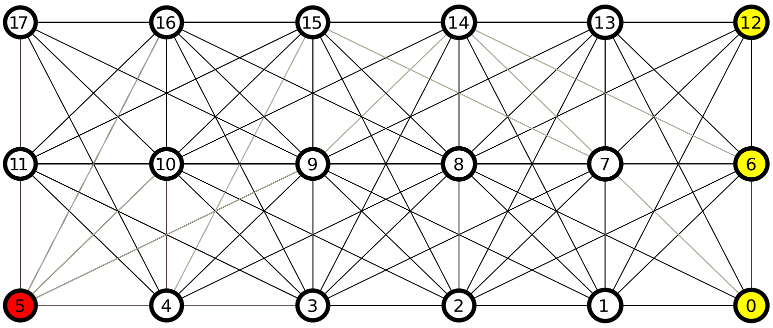


Figure 5.1: 3×6 grid topology

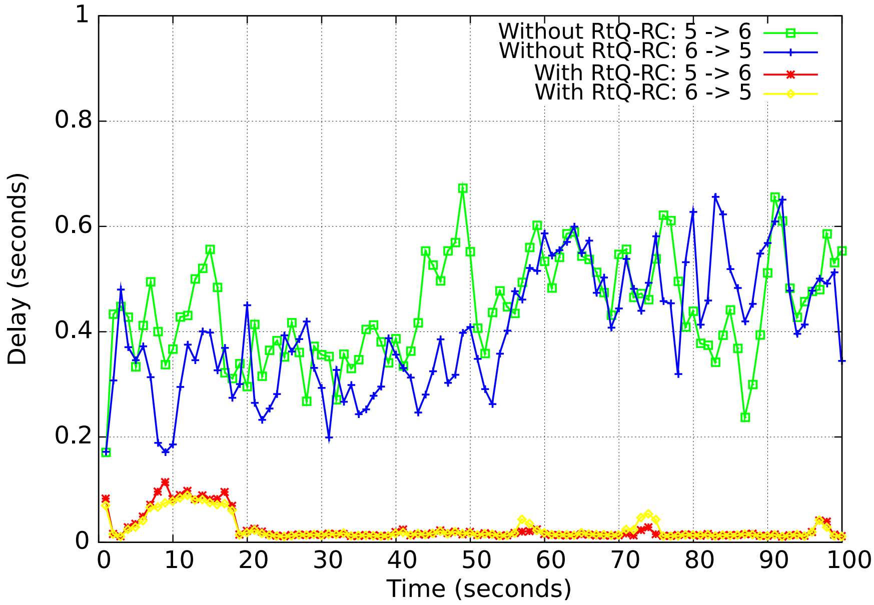


Figure 5.2: End-to-end delay comparison with RtQ-RC in 3×6 grid topology

5.2.2 General Performance

We evaluate RtQ-RC with over 30 randomly generated 25-node topologies. A 25-node topology represents a fairly dense mesh, which is challenging, considering that real meshes would have more than one gateway to the Internet. First, we simulate real-time flows with only the default 802.11 DCF and without TCP traffic. We examine the results to ensure that real-time flows in these simulations are adequately supported, meeting their required QoS. We repeat these experiments, first with TCP traffic using only the default 802.11 DCF, then with the same TCP traffic using RtQ-RC.

Figure 5.3 shows the one-way packet delay cumulative distribution functions (CDFs) for all random topologies. Using RtQ-RC, over 95% of the one-way real-time packet delays are kept below 40 ms. The default DCF without any additional mechanism does not achieve interactive real-time QoS; over 50% of the real-time packets have delays greater than 100 ms. In comparison to when there is no TCP traffic, real-time one-way delay is slightly larger when TCP traffic exist and RtQ-RC is operating. The difference in end-to-end delay is caused by an increase in media contention from TCP traffic and is inherent by design since RtQ-RC is feedback-based.

Figure 5.4 shows each CDF (per experiment) when TCP traffic interacts with real-

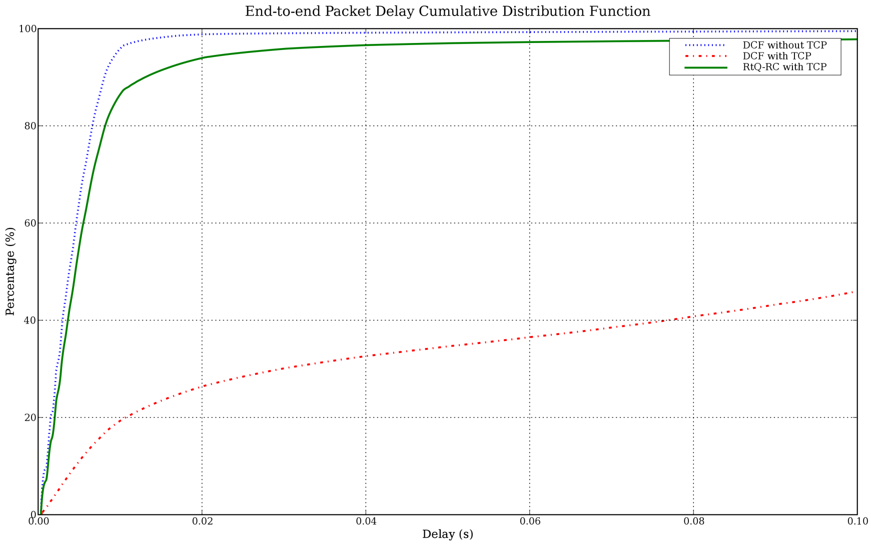


Figure 5.3: End-to-end Packet Delay Comparison for all 25-node Random Topologies

time traffic, using the default 802.11 DCF. Similar to results and reasoning from Section 2.3.6, TCP traffic degrades real-time traffic QoS in most experiments. Figure 5.5(a) shows each CDF when there is only real-time traffic using the default DCF. Similarly, Figure 5.5(b) show the results when TCP and real-time traffic are in the mesh with RtQ-RC enabled. In each experiment, using RtQ-RC resulted in minimal real-time packet end-to-end delays with at least 90% of the delays less than 60 ms. Experiments 38 and 39 are notable exceptions. Figure 5.6 shows their topologies. In the case when no TCP traffic exist, flow $r(13,0)$ in experiment 38, and flows $r(14,0)$ and $r(5,0)$ in experiment 39 packet delays are higher because of structural unfairness effects [67, 68]. It is expected that because the TCP flows (*i.e.* $e(15,0)$, $e(7,0)$, $e(17,0)$, $e(8,0)$, and $e(16,0)$ in experiment 38 and $e(23,0)$, $e(20,0)$, and $e(12,0)$ in experiment 39) are closer to the gateways, packet delay would increase, however, RtQ-RC maintains similar packet delays as before.

Table 5.4 reports the percentage of end-to-end one-way packet delays less than δ. We use these particular δ thresholds to effectively gauge the performance of RtQ-RC, with respect to VoIP and FPS application requirements as discussed in Section 5.1.4 [1].

[1] Although interactive real-time applications do not require one-way delays less than 10 ms, a δ at this

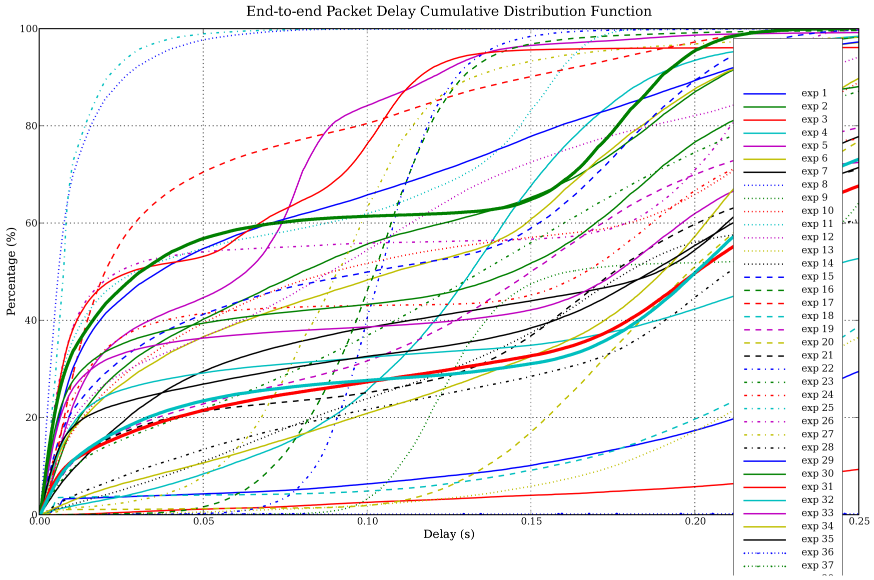
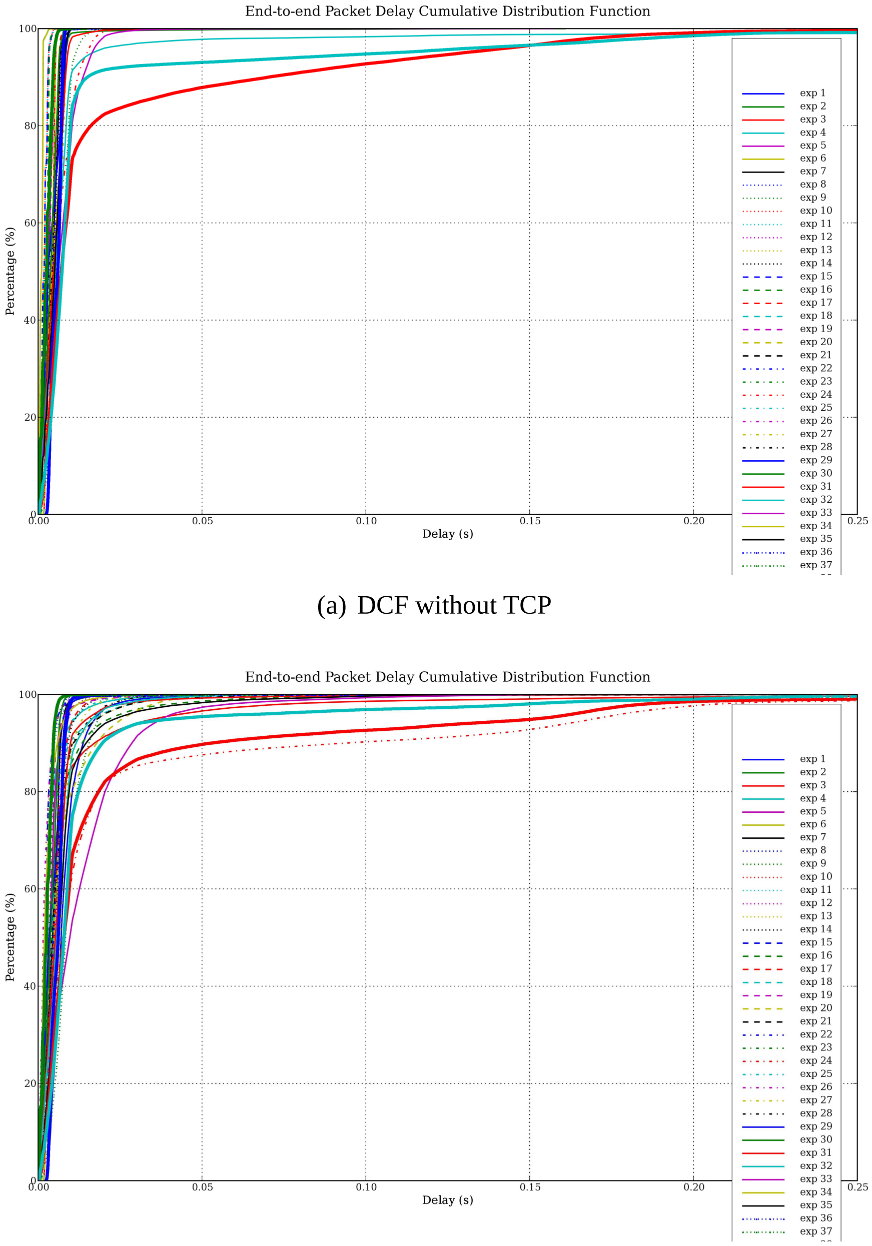


Figure 5.4: Packet Delay per Experiment with TCP, using 802.11 DCF

Approximately 97% to 98% of one-way packet delays are kept below 30 ms, 60 ms, and 65 ms when RtQ-RC is used. In comparison, when only the default DCF is used, approximately 46% of one-way packet delays are greater than 150 ms. Overall, RtQ-RC is able to achieve the required end-to-end delay for VoIP and FPS. In addition, on average, RtQ-RC achieves real-time end-to-end delay, jitter (*i.e.*, standard deviation of delay) and packet loss requirements as shown in Figure 5.7.

Our results indicate that using RtQ-RC with its default AIMD increase-decrease algorithm achieves the desired QoS for interactive real-time services. In all cases, once a signal of real-time congestion is perceived for any real-time flow, rate control is enforced on TCP flows transiently slowing them down until that condition no longer exists for all real-time flows that a node sees. This effectively reduces the channel contention and network congestion caused by TCP, increasing the bandwidth or air-time allocation for the real-time flows which reduces their packet delay, jitter and loss.

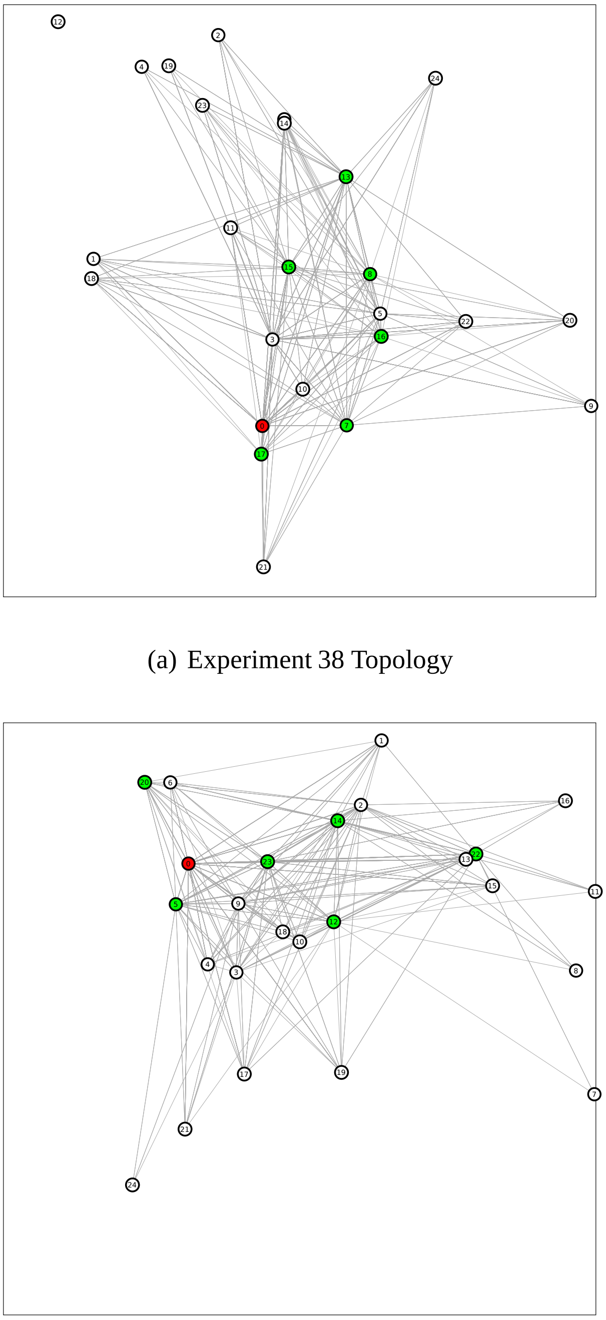
value serves to assess RtQ-RC performance with respect to very low delays, which are easier to achieve within a 100 Mbps Ethernet LAN, for example.



(a) DCF without TCP

(b) RtQ-RC with TCP

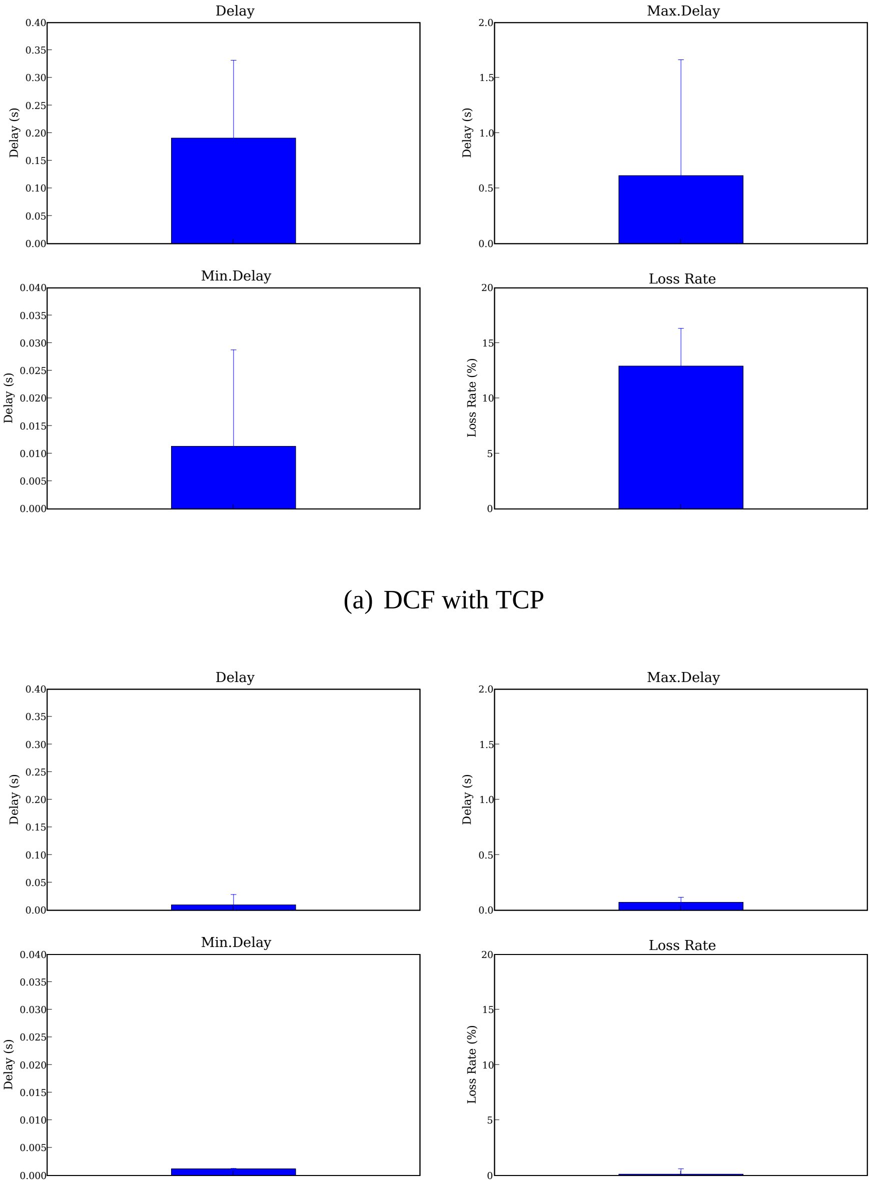
Figure 5.5: Packet Delay Comparison for each 25-node Random Topology



(a) Experiment 38 Topology

(b) Experiment 39 Topology

Figure 5.6: Experiment Topologies



(a) DCF with TCP

(b) RtQ-RC with TCP

Figure 5.7: The Overall Average Performance within 25-node Random Topologies

δ (ms)	DCF with TCP		RtQ-RC with TCP	
	% delay	% standard dev.	% delay	% standard dev.
10	16.7	17.1	88.1	16.1
30	25.8	23.7	96.7	8.5
60	31.4	25.3	98	6.6
65	32.2	25.4	98	6.5
120	46.6	28.3	98.7	4.7
150	53.6	29.3	99	3.4

Table 5.4: The Percentage of End-to-end One-way Packet Delay $\leq \delta$

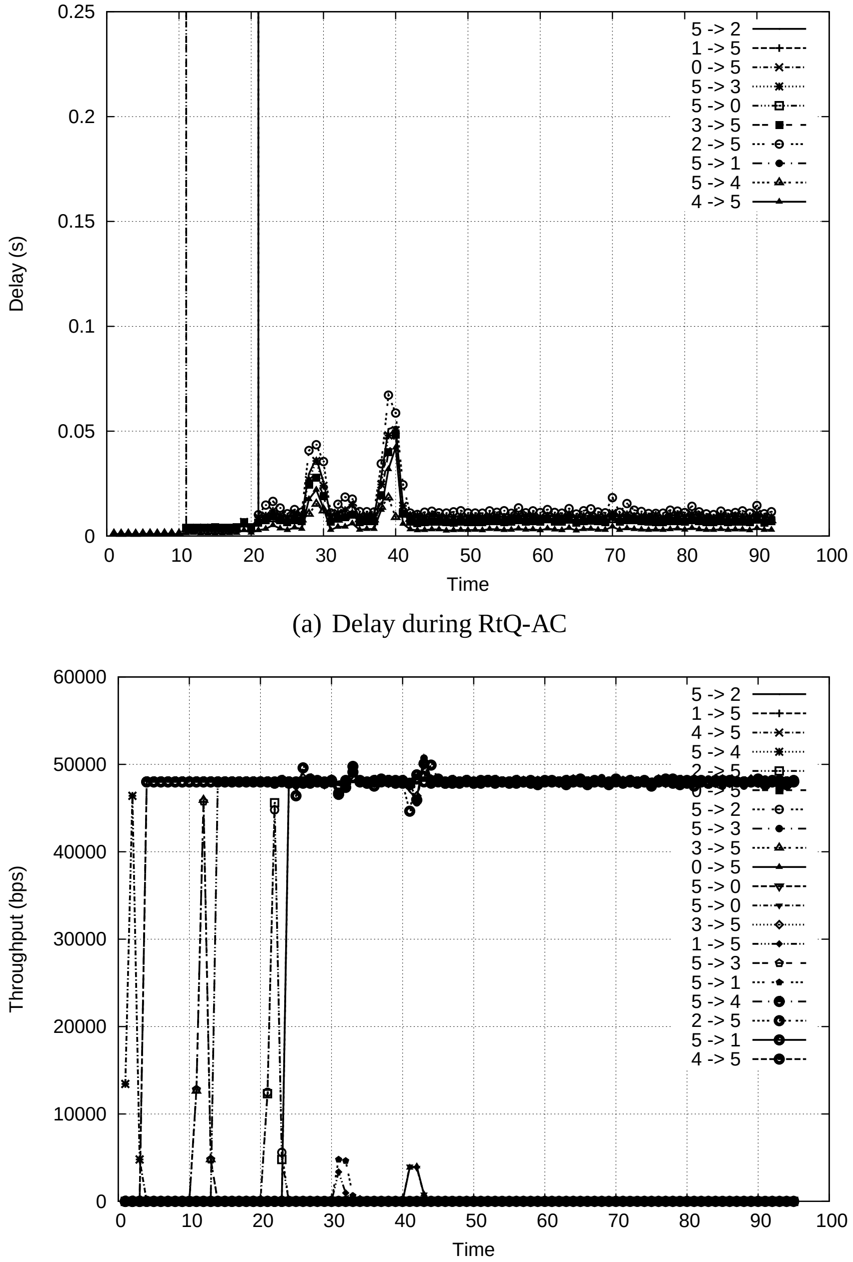
5.3 RtQ-AC Performance

In this section, we evaluate the performance of the admission control (AC) mechanism, RtQ-AC. First, we look at the results when RtQ-AC is operational in a 6-node chain scenario. Second, we report the results when it is applied to the example grid topology (Figure 5.1). Finally, to evaluate its overall performance, we run RtQ-AC in over 20 randomly generated 25-node topologies, with randomly generated real-time flows. For the random topologies, the simulations run for 150 seconds, injecting real-time flows every 10 seconds. We simulate RtQ-AC with an admittance threshold, α of 0.25 and a slow-rise probe time of 2 seconds.

5.3.1 Performance in Example Scenarios

We evaluate RtQ-AC effectiveness across the 6-node chain and 3×6 grid. For the chain, we start injecting flows from $r(4, 5)$ until $r(0, 5)$. In the grid, we use the top-outer nodes, 17 to 13, and start with $r(17, 5)$ until $r(13, 5)$.

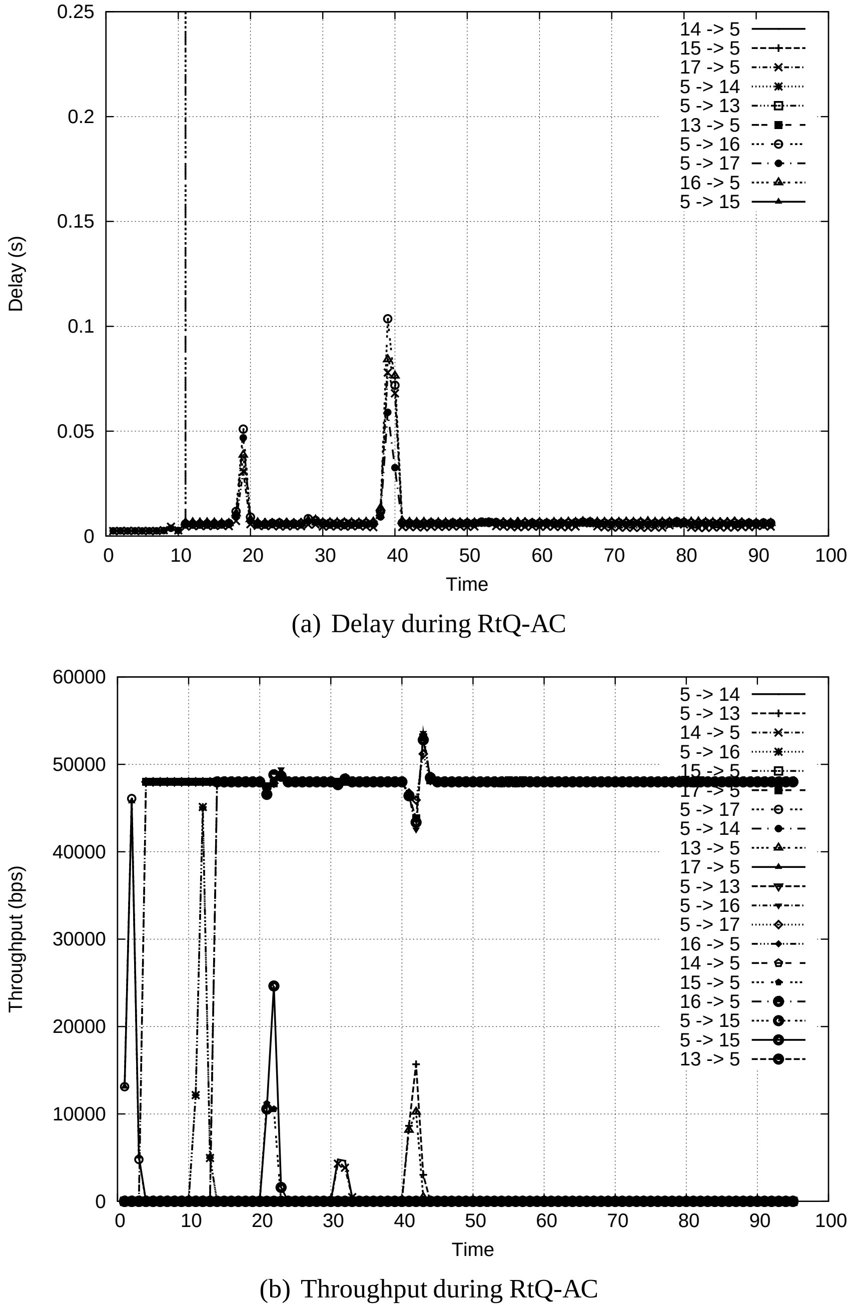
Figure 5.8 illustrates RtQ-AC in operation, admitting and denying flows for the chain. The periodic spikes up to 45 seconds of simulation time in (Figure 5.8(b)) are the slow-rise probes. To compare, we manually add flows from node 4 to 0. When $r(1, 5)$ and or $r(0, 5)$ are added, the chain becomes congested and QoS drastically degrades. RtQ-RAC had correctly denied $r(1, 5)$ and $r(0, 5)$. The admitted flows are adequately supported, operating at their tspec. Figure 5.9 show the QoS during RtQ-AC in the grid topology. The slow-rise probing effects and denied flows are evident, but more importantly, average delay is below 10 ms.



(a) Delay during RtQ-AC

(b) Throughput during RtQ-AC

Figure 5.8: RtQ-AC operating on 6-node chain topology



(a) Delay during RtQ-AC

(b) Throughput during RtQ-AC

Figure 5.9: RtQ-AC operating on 3×6 grid topology

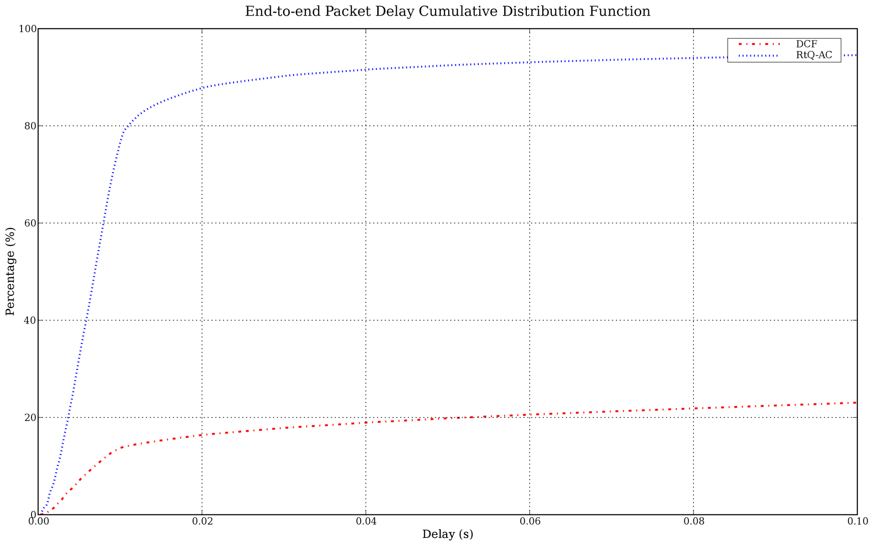


Figure 5.10: Packet Delay Comparison for 14 Random Topologies

5.3.2 General Performance

We evaluate RtQ-AC with 23 randomly generated 25-node topologies, with randomly generated real-time flow. Fourteen of the 23 have real-time flows such that when they are all admitted the network is congested and some or all flows do not get their desired QoS. The remaining 9 have real-time flows such that all could be admitted without congesting the network and achieves their desired QoS.

First, we simulate the real-time flows using RtQ-AC. If a real-time flow is denied, we then simulate using only the default DCF and the previously admitted flows and try to add the previously denied flow. We examine the result to determine whether the QoS requirements of all (*i.e.*, previously admitted and newly admitted) real-time flows are still being achieved. If any real-time flow's QoS requirement is now not achieved, then the decision to deny that flow was correct.

For most topologies, RtQ-AC made correct admittance decisions, maintaining real-time end-to-end delay requirements as shown in Figure 5.10. It is also evident that without admission control, the resulting end-to-end delays are not within desired real-time QoS requirements because too many real-time flows are within the WMN. Table 5.5 reports the percentage of end-to-end delays less than δ while Figure 5.11 shows the overall

δ (ms)	DCF		RtQ-AC	
	% delay	% standard dev.	% delay	% standard dev.
10	12.1	8.4	78.8	19.3
30	16.5	13.2	91.4	14.1
60	19.3	16.9	93.7	12.1
65	19.6	17.3	93.9	12
120	22.4	19.1	95.4	10.4
150	24.7	20	96.3	9.4

Table 5.5: The Percentage of One-way Packet Delay $\leq \delta$

results when RtQ-AC is used. The real-time QoS during RtQ-AC is adequate. Compared to the results from Section 5.2.2, the attained real-time QoS with admission control is desirable but reduced. It is caused by the slow-rise probing operation. Slow-rise probing transiently disrupts the QoS of the network when it tests for bandwidth because the probe rate additively increases until $\gamma > \alpha$ or when the tspec is reached. If $\gamma > \alpha$, then the amount of marked packets from the real-time queues have increased as they perceive momentary congestion which is increasing the delay of some real-time packets. Further, compared to the simulation time, the injection of new requesting real-time flows is frequent. Therefore, the slow-rise probing effect appears more significant because the simulation time is short.

Slow-rise probing does not significantly affect the real-time QoS when flows are admissible as is evident from Figure 5.12. In such situations, the slow-rise probe rates rise to the tspec rate without increasing γ. Two out of the 14 experiments had real-time flows that did not achieve their desired throughput and real-time QoS. Both experiment's incorrectly admitted flows because of structural unfairness effects. The real-time flows' throughputs during simulation are shown in Figure 5.13(b). In this particular topology (Figure 5.13(a)), flow $r(24, 0)$ is correctly admitted. Subsequently, $r(5, 0)$ gains admittance because of structural unfairness. During slow-rise probing, $r(5, 0)$ captured the medium more often because of its relatively short wireless hop and path length compared to that of $r(24, 0)$. Therefore, $r(5, 0)$ gained more sending opportunities that caused its probe rate to quickly rise to the tspec, acheiving admittance. In addition, because $r(24, 0)$ and its intermediate nodes are in an unfair situation, their marked packets are delayed in reaching the gateway and hence $r(5, 0)$ probes experience no multiplicative decrease. Figure 5.15 shows the results for the other experiment (Figure 5.14). After flow $r(16, 0)$

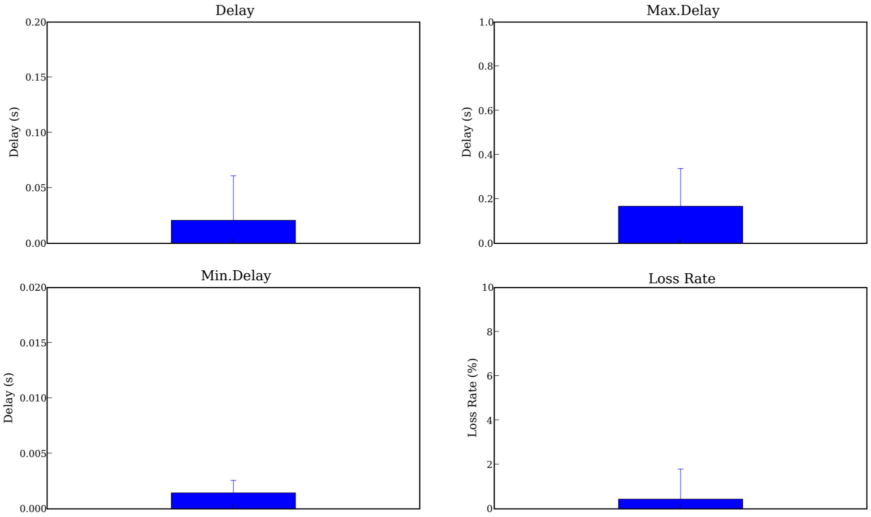


Figure 5.11: The Overall Average Performance within 14 Random Topologies

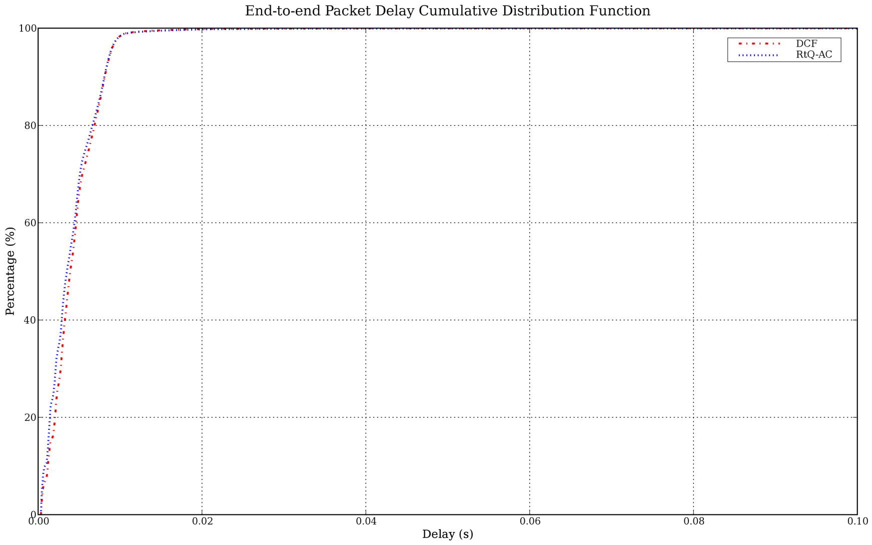
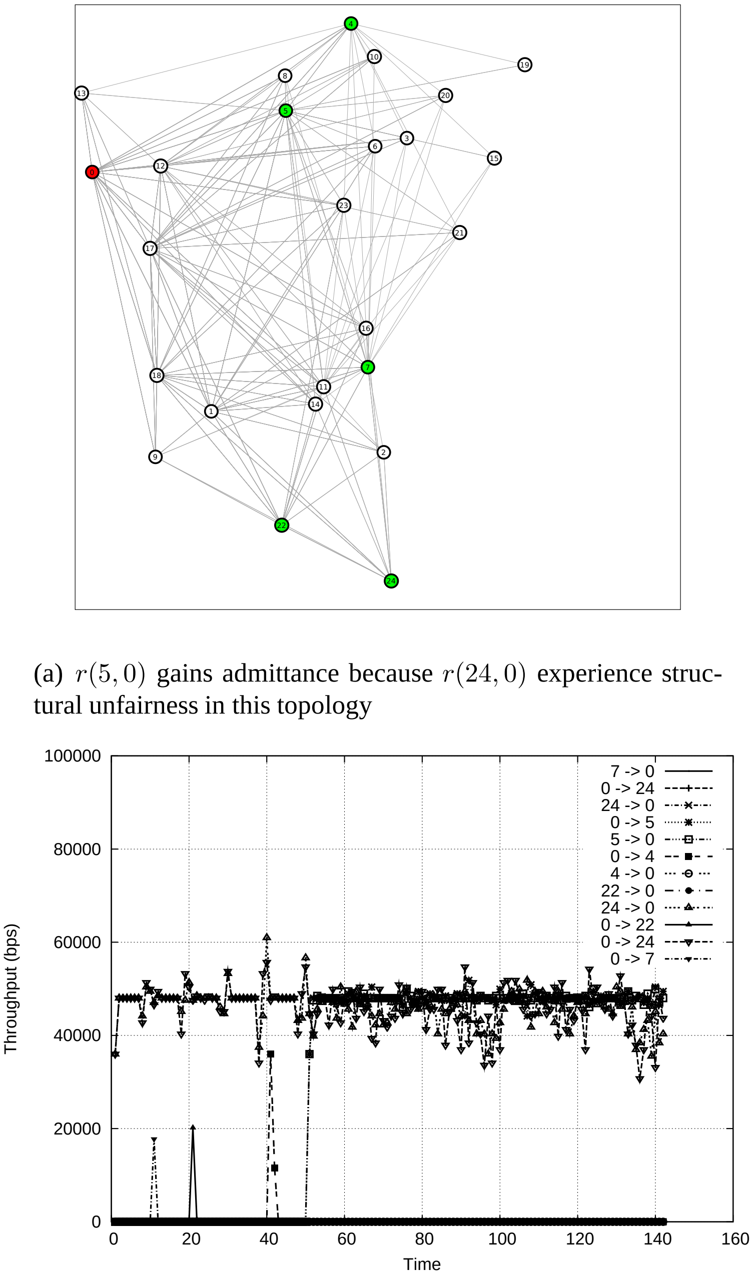


Figure 5.12: End-to-end Packet Delay Comparison for 9 Random Topologies



(a) $r(5,0)$ gains admittance because $r(24,0)$ experience structural unfairness in this topology

(b) Throughput during simulation run

Figure 5.13: Structural Unfairness During Admission Control

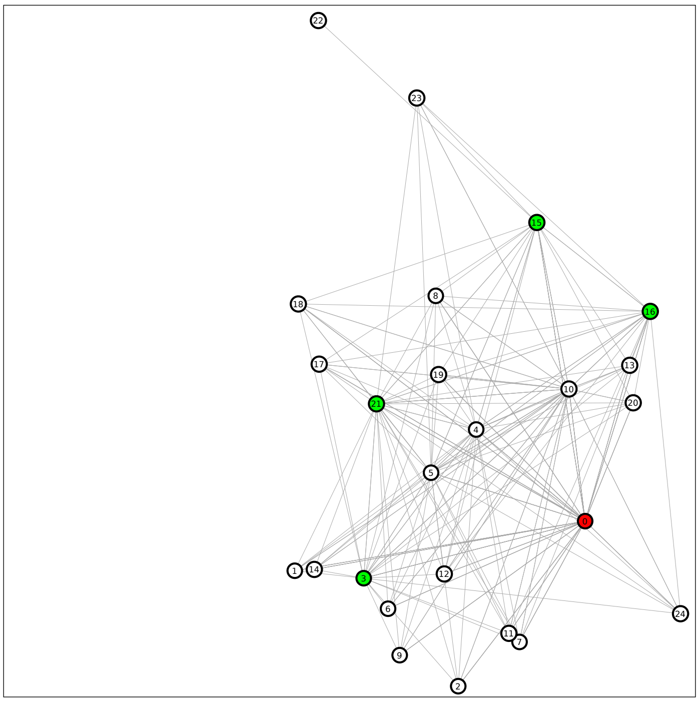
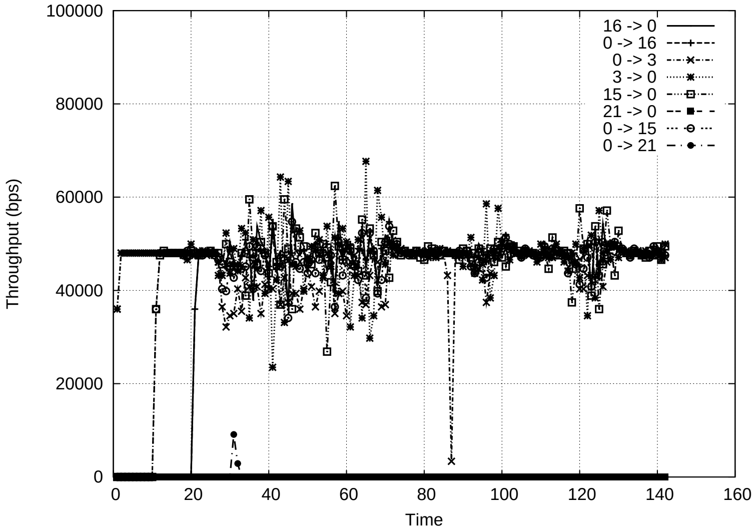


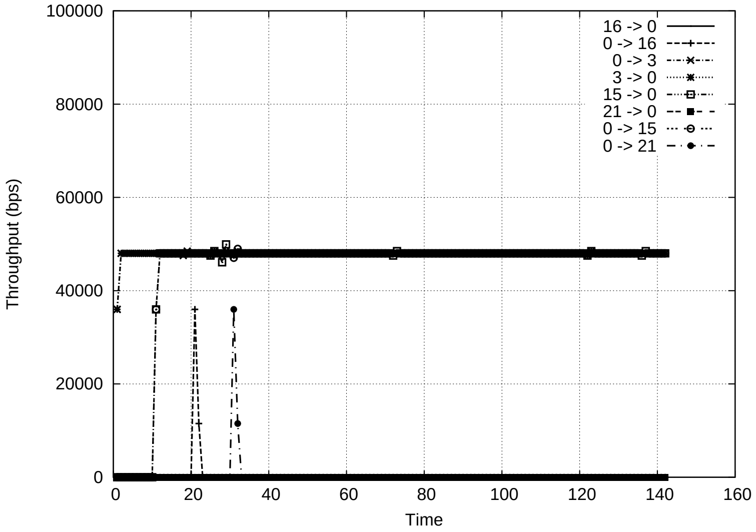
Figure 5.14: $r(16, 0)$ gains admittance because the other flows, $r(3, 0)$ and $r(15, 0)$ experiences structural unfairness in this topology

is admitted, smooth throughput cannot be maintained as it should have been denied. Similar to the first experiment, $r(16, 0)$ was able to gain admittance because of structural unfairness to the other flows ($r(3, 0)$ and $r(15, 0)$).We repeat this experiment with different admittance thresholds. Figure 5.15(b) shows a run, using a more stringent α of 0.1. The real-time QoS is improved as $r(16, 0)$ is now correctly denied.

Our results indicate that using slow-rise probing, RtQ-AC is able to adequately infer available bandwidth to determine if there are enough resources to support new requests and ensures that admitted flows are protected. Its efficacy is affected by structural unfairness, however, these effects are adequately mitigated with a strigent α. A possible optimization is to dynamically adjust α with respect to the number of admitted real-time flows and give marked packets higher queueing priority than unmarked packets.



(a) With an α of 0.25, $r(16, 0)$ distrupts the desired QoS. It has gained admitted since $r(3, 0)$ and $r(15, 0)$ experience structural unfairness



(b) Desired throughput and QoS is achieved as $r(16, 0)$ is correctly denied with α of 0.1

Figure 5.15: Mitigating Structural Unfairness with a Stricter α during Admission Control

Delay(s)	Max.Delay (s)	Min.Delay(s)	Loss Rate (%)
0.0026 ± 0.0188	0.04228 ± 0.1144	8.727e-07 ± 0.000143	0.1233 ± 1.379

Table 5.6: Difference in Performance between AIMD and AIAD

5.4 Increase-Decrease Algorithm Performance

In this section, we quantify the relative performance of the AIMD and AIAD in satisfying real-time QoS requirements. In addition, we investigate the algorithms efficiency in TCP traffic usage of the residual bandwidth.

5.4.1 Comparative Performance

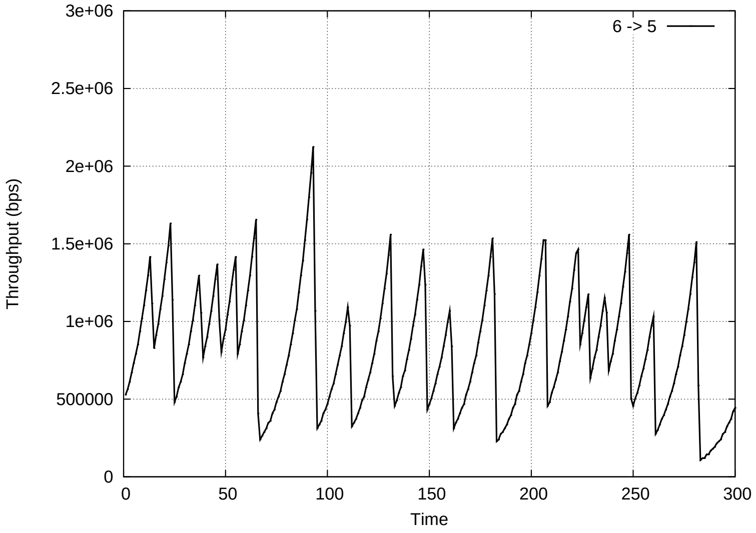
We now compare the relative performance of AIMD and AIAD. Using a particular descipline, we simulate RtQ-RC, ensuring that the randomly generated real-time flows are properly supported.

There is an increase in average delay and loss rate as seen in Table 5.6 when RtQ-RC uses AIAD. The reduction in real-time QoS performance is expected. AIAD is less aggressive on reducing TCP rate which increases the time of recovering real-time QoS after transient and incipient congestion as is evident in Figure 5.16 that shows the TCP throughput during a simulation run for RtQ-RC using AIMD and AIAD. AIAD is less aggressive on decrease than AIMD and results in relatively smoother and higher throughput for TCP. However, maximum delay is significantly larger in AIAD and is caused by an inherent slower response to congestion because TCP rate is slowly decreased with subtraction instead of a multiplicative factor.

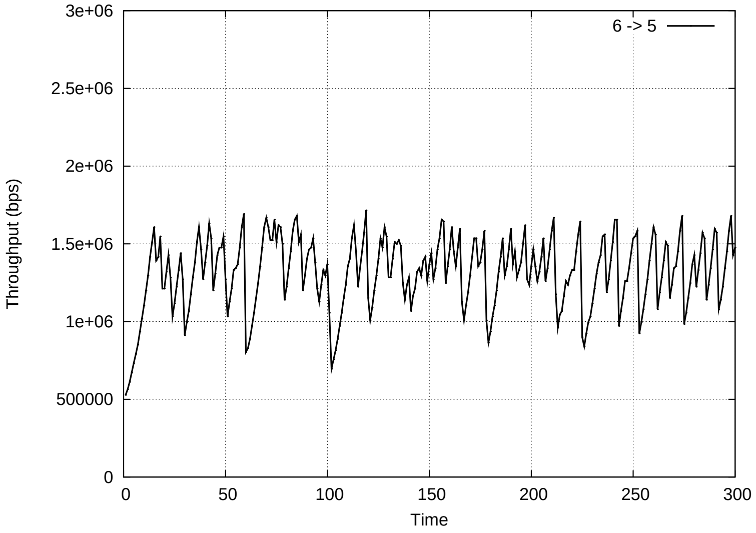
5.4.2 Bandwidth Usage Efficiency

We now evaluate our rate control efficiency in using the residual bandwidth for TCP traffic. It is very important to not excessively penalize elastic traffic while achieving real-time QoS. RtQ-RC should set an adequate rate for TCP traffic such that it uses most or the residual bandwidth.

Using a particular descipline, we simulate RtQ-RC, ensuring that the randomly generated real-time flows are properly supported. We randomly generate one TCP flow to mitigate TCP unfairness effects. We record the TCP throughput and the average delay of the real-time flows. We repeat the experiment using just the default 802.11 DCF.



(a) TCP Throughput during RtQ-RC using AIMD



(b) TCP Throughput during RtQ-RC using AIAD

Figure 5.16: Example TCP Throughput during RtQ-RC using AIMD and AIAD

AIMD		AIAD	
% Efficiency	% standard dev.	% Efficiency	% standard dev.
88.8	30.2	90.3	22.8

Table 5.7: Increase-Decrease Algorithm Efficiency

We set the rate of ns-2's token bucket filter (TBF) to the recorded TCP throughput. We continously increase the rate of TCP flow via the TBF until the average delay of the real-time flows is greater than or equal to the recorded average delay. We compare the TCP throughput at this point with that achieved using RtQ-RC to determine the bandwidth usage efficiency. We use this methodology for over 30 randomly generated 25-node networks.

AIMD and AIAD are quite efficient as shown in Table 5.7. AIAD has better TCP usage of residual bandwidth during rate control because of its less aggressive rate decrease. AIMD results are also desirable. TCP is not unfairly reduced to an unncessary low rate because the AIMD and AIAD controlled rates are close to the residual bandwidth. RtQ-RC can be considered TCP friendly.

5.5 PERFORMANCE WITH QUAKE III

In this section, we complete our evaluation by simulating the Quake III (Q3) traffic model. The following simplifications where made to the Q3 model [63]:

- For the server, the packet length is only dependent on the number on players. In reality it is also dependent on the map played but that a weaker dependency and has been ignored. The packet interval for update packets is fixed to 50 ms. The real distribution is a gamma distribution, peaking at 50 ms. However, this distribution would increases the simulation time without significantly affecting the simulated traffic quality.

- For the client, the packet length is modelled by a normal distribution, which is imprecise. The actual distribution has not been found and is assumed to be too complex to be valid for a simulation model. The packet intervals are dependent on the graphics cards and is modelled based on two different modern 32 MB graphic cards discussed in [63].

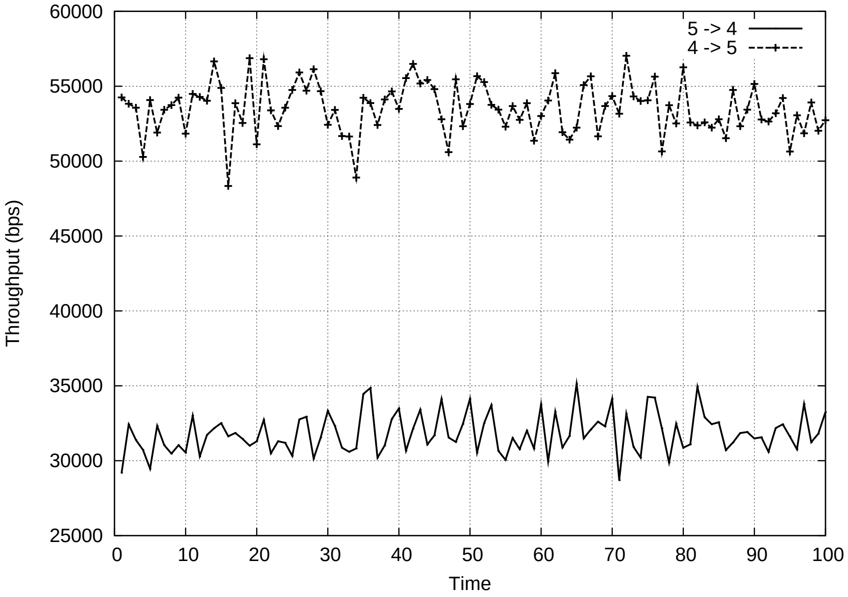


Figure 5.17: Example Throughput of Quake III with 10 players

We use the methodology from Section 5.2.2 to evaluate RtQ-RC performance (using AIMD) with this traffic model. However, in contrast, TCP traffic now originates from the gateway. We simulate a game of 10 players on 35 random 25-node networks. A 10-player Quake III game generates client and server traffic of about 54 kbps and 32 kbps, respectively as shown in Figure 5.17, for example.

Using this traffic model, RtQ-RC remains effective as is seen in Figures 5.18 and 5.19 and Table 5.8, illustrating its efficacy with variable bit-rate traffic. As discussed in Section 4.4, RtQ-RAC is able to effectively support variable real-time traffic because of Little's result and by setting the average real-time queue threshold, rtq_{Th} to 1.

5.6 Summary

The results from this chapter demonstrated the general performance of RtQ-RAC. We evaluated RtQ-RAC across example chains and grid and large random topologies. From these evaluations, it was shown that though simple, RtQ-RAC is very effective in acheiving soft real-time QoS requirements, while allowing TCP traffic to efficiently use residual bandwidth. Assuming, all traffic traverses the gateway, communicating between a mesh node and an Internet host, RtQ-RAC reduces interfering TCP traffic guaranteeing desir-

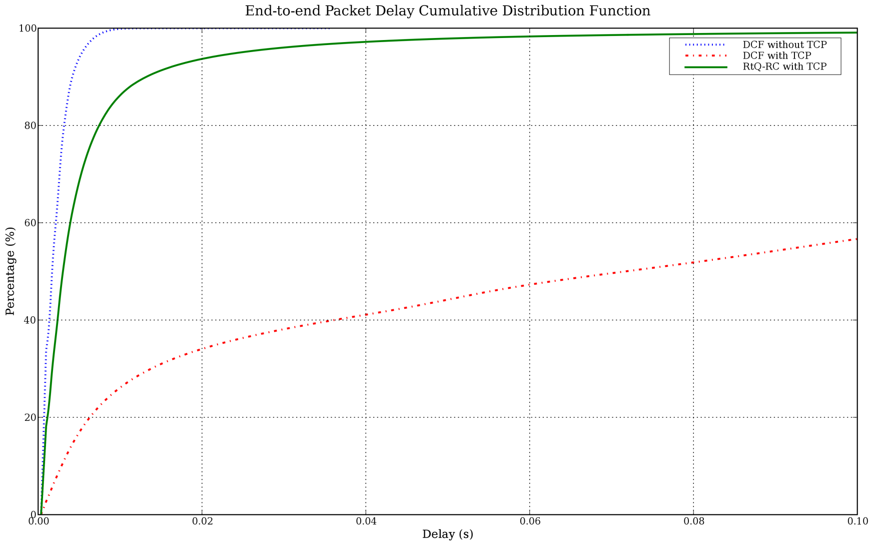


Figure 5.18: End-to-end Packet Delay Comparison using Quake III traffic model

δ (ms)	DCF		RtQ-RC	
	% delay	% standard dev.	% delay	% standard dev.
10	24	21.2	85.3	11.3
30	34.5	26	95.5	6.2
60	42.9	27.6	98.2	3.9
65	44	27.8	98.4	3.7
120	56.3	29.6	99.4	2.1
150	62.9	30.2	99.6	1.6

Table 5.8: The Percentage of End-to-end One-way Packet Delay $\leq \delta$ using Quake III Traffic Model

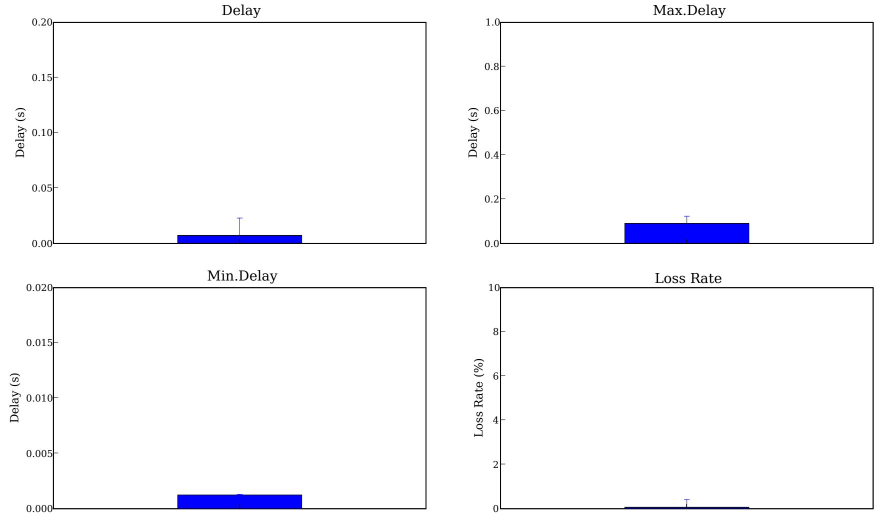


Figure 5.19: The Overall Average Performance within 25-node Random Topologies using Quake III Traffic Model

able real-time QoS, unlike RtRED. Further, when there is no TCP traffic, RtQ-RAC has negligible overhead. Overhead in such cases is caused by our decision to give priority (in our per-class queue management) to control traffic which is assumed to be relatively minimal.

6 Conclusion and Future Work

We proposed a simple, novel and efficient scheme, called Real-time Queue-based Rate and Admission Control, RtQ-RAC that achieves soft real-time QoS for interactive multimedia applications. RtQ-RAC utilizes a single real-time queue-class to enforce local and distributed rate-limiting of elastic traffic. RtQ-RAC is an improved solution derived from the evaluation of our initial RtRED scheme. If we consider the additional delay, jitter, and loss from the Internet to the end-host, RtRED fails to provide desirable real-time QoS for interactive applications. Further, RtRED cannot achieve real-time QoS when TCP traffic do not intersect at the same nodes as real-time traffic, but whose transmission is within interference range of real-time traffic. Therefore, RtQ-RAC uses packet marking to enforce remote rate-limiting of any interfering TCP traffic. Packet marking combined with slow-rise probing is also used for admission control, keeping the load of real-time traffic below network capacity, ensuring required QoS. Using ns-2, we have demonstrated RtQ-RAC's efficacy. With RtQ-RC, 97% of one-way real-time packet delays are below 30 ms. We achieve at least 89% elastic usage efficiency; TCP traffic is not unfairly rate-limited. Further, RtQ-AC demonstrated good performance although real-time QoS transiently degrades during slow-rise probing.

Prior to presenting our RtRED and RtQ-RAC schemes, we demonstrated the adverse effects of TCP traffic on interactive real-time traffic. We showed that while traditional fair queueing, priority queueing, and AQM are effective in wired networks, they fail to provide guaranteed QoS for real-time applications in wireless mesh networks. These service disciplines fail because they operate irrespective of the interference effects from elastic TCP traffic.

RtQ-RAC and this work can be further extended. First, is in a more comprehensive investigation of admittance sensitivity with its threshold. This is useful in improving admission control via dynamic variation of the admittance threshold with real-time traffic load. Second, is to quantify the improvements (*e.g.*, faster rate-control response) gained from enhancing RtQ-RAC for 802.11 promiscuous mode. Next, would be to investigate the state clean-up mechanisms described. Finally, RtQ-RAC should be implemented and evaluated in an experimental testbed to fully determine its efficacy.

REFERENCES

[1] The network simulator. ns-2. http://www.isi.edu/nsnam/ns/.

[2] G.S. Ahn, AT Campbell, A. Veres, and L.H. Sun. Supporting service differentiation for real-time and best-effort traffic in stateless wireless ad hoc networks (SWAN). *IEEE Transactions on Mobile Computing*, 1(3):192–207, 2002.

[3] I.F. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: a survey. *Computer Networks*, 47(4):445–487, 2005.

[4] F.M. Anjum and L. Tassiulas. *Balanced-RED: An Algorithm to Achieve Fairness in the Internet*. Defense Technical Information Center, 1999.

[5] M. Barry, AT Campbell, and A. Veres. Distributed control algorithms for service differentiation in wireless packet networks. *Proc. of IEEE INFOCOM*, 1, 2001.

[6] T. Beigbeder, R. Coughlan, C. Lusher, J. Plunkett, E. Agu, and M. Claypool. The effects of loss and latency on user performance in unreal tournament 2003®. *Proc. of ACM SIGCOMM 2004 workshops on NetGames*, pages 144–151, 2004.

[7] J.C.R. Bennett and H. Zhang. WF 2 Q: worst-case fair weighted fair queueing. *Proc. of IEEE INFOCOM*, 1, 1996.

[8] Y.W. Bernier. Latency Compensating Methods in Client/Server In-game Protocol Design and Optimization. *Proc. of Game Developers Conference*, 2001.

[9] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. MACAW: a media access protocol for wireless LAN's. *Proc. of ACM SIGCOMM*, pages 212–225, 1994.

[10] V. Bharghavan, S. Lu, and T. Nandagopal. Fair queuing in wireless networks: issues and approaches. *IEEE Personal Communications [see also IEEE Wireless Communications]*, 6(1):44–53, 1999.

[11] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. RFC2475: An Architecture for Differentiated Service. *Internet RFCs*, 1998.

[12] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP)-Version 1 Functional Specification. *Request for Comments*, 2205, 1997.

[13] L. Breslau, E.W. Knightly, S. Shenker, I. Stoica, and H. Zhang. Endpoint admission control: architectural issues and performance. *ACM SIGCOMM Computer Communication Review*, 30(4):57–69, 2000.

[14] Report Buyer. Fibre to the Home Broadband Subscribers in Japan Estimated to Reach 10 Million, from http://www.przoom.com/news/26687/, 2007.

[15] L. Cai, X. Shen, J.W. Mark, and Y. Xiao. Voice Capacity Analysis of WLAN With Unbalanced Traffic. *IEEE Transactions on Vehicular Technology*, 55(3):752–761, 2006.

[16] C. Cetinkaya, V. Kanodia, and EW Knightly. Scalable services via egress admission control. *IEEE Transactions on Multimedia*, 3(1):69–81, 2001.

[17] I.D. Chakeres and E.M. Belding-Royer. PAC: Perceptive Admission Control for Mobile Wireless Networks. *Proc. of Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine)*, 2006.

[18] S. Chen and K. Nahrstedt. Distributed quality-of-service routing in ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1488–1505, 1999.

[19] D.M. Chiu and R. Jain. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *Computer Networks and ISDN Systems*, 17(1):1–14, 1989.

[20] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol (OLSR). *Request for Comments*, 3626, 2003.

[21] E.J. Coffman. Not Your Father's PBX?, from http://www.acmqueue.org. 2004.

[22] M. Conti and S. Giordano. Multihop Ad Hoc Networking: The Reality. *IEEE Communications Magazine*, 45(4):88–95, 2007.

[23] FTTH Council. Fiber to the home council, from http://www.ftthcouncil.org, 2006.

[24] A. Dailianas and A. Bovopoulos. Real-time admission control algorithms with delay and lossguarantees in ATM networks. *Proc. of IEEE INFOCOM*, pages 1065–1072, 1994.

[25] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. In *Proc. of SIGCOMM*, pages 1–12, 1989.

[26] Z. Fang and B. Bensaou. Fair bandwidth sharing algorithms based on game theory frameworks for wireless ad-hoc networks. *Proc. of IEEE INFOCOM*, 2, 2004.

[27] H. Fattah and C. Leung. An overview of scheduling algorithms in wireless multimedia networks. *IEEE Wireless Communications [see also IEEE Personal Communications]*, 9(5):76–83, 2002.

[28] W. Feng, KG Shin, DD Kandlur, and D. Saha. The BLUE active queue management algorithms. *IEEE/ACM Transactions on Networking (TON)*, 10(4):513–528, 2002.

[29] W.C. Feng, DD Kandlur, D. Saha, and KG Shin. A self-configuring RED gateway. *Proc. of IEEE INFOCOM*, 3, 1999.

[30] D. Ferrari. Client requirements for real-time communication services. *IEEE Communications Magazine*, 28(11):65–72, 1990.

[31] D. Ferrari and DC Verma. A scheme for real-time channel establishment in wide-area networks. *IEEE Journal on Selected Areas in Communications*, 8(3):368–379, 1990.

[32] S. Floyd. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking (TON)*, 1(4):397–413, 1993.

[33] S. Floyd. TCP and Explicit Congestion Notification. *ACM SIGCOMM Computer Communication Review*, 24(5):8–23, 1994.

[34] Z. Fu, H. Luo, P. Zerfos, S. Lu, L. Zhang, and M. Gerla. The impact of multihop wireless channel on TCP performance. *IEEE Transactions on Mobile Computing*, 4(2):209–221, 2005.

[35] V. Gambiroza, B. Sadeghi, and E.W. Knightly. End-to-end performance and fairness in multihop wireless backhaul networks. *Proc. of ACM SIGCOMM*, pages 287–301, 2004.

[36] Michele Garetto, Theodoros Salonidis, and Edward W.Knightly. Modeling per-flow throughput and capturing starvation in csma multi-hop wireless networks. In *Proc. of IEEE INFOCOM*, 2006.

[37] M Gast. *802.11 Wireless Networks : the Definitive Guide*. O'Reilly, Sebastopol, CA, 2005.

[38] S.J. GoIestani. A Stop-and-Go Queueing Framework for Congestion Management. *Proc. of SIGCOMM*, 1:8–18, 1990.

[39] S.J. Golestani. A self-clocked fair queueing scheme for broadband applications. *Proc. of IEEE INFOCOM*, pages 636–646, 1994.

[40] M. Grossglauser and D. Tse. A framework for robust measurement-based admission control. *ACM SIGCOMM Computer Communication Review*, 27(4):237–248, 1997.

[41] Z.J. Haas, M.R. Pearlman, et al. The Zone Routing Protocol (ZRP) for Ad Hoc Networks. *TERNET DRAFT-Mobile Ad hoc Networking (MANET) Working Group of the bternet Engineering Task Force (ETF), November*, 1997.

[42] M. Heusse, F. Rousseau, R. Guillier, and A. Duda. Idle sense: an optimal access method for high throughput and fairness in rate diverse wireless LANs. *Proc. of ACM SIGCOMM*, pages 121–132, 2005.

[43] Ellacoya Networks Inc. Ellacoya Data Shows Web Traffic Overtakes Peer-to-Peer (P2P). 2007. http://www.ellacoya.com/news/pdf/2007/NXTcommEllacoyaMediaAlert.pdf.

[44] G. ITU. 114; One-way transmission time. *International Telecommunication Union, Geneva, Switzerland*, 2, 2003.

[45] A. Jain, M. Gruteser, M. Neufeld, and D. Grunwald. *Benefits of packet aggregation in ad-hoc wireless network.* PhD thesis, Tech. Rep. CU-CS-960-03, University of Colorado at Boulder, 2003, 2003.

[46] K. Jain, J. Padhye, V.N. Padmanabhan, and L. Qiu. Impact of Interference on Multi-Hop Wireless Network Performance. *Wireless Networks*, 11(4):471–487, 2005.

[47] S. Jamin, P.B. Danzig, S.J. Shenker, and L. Zhang. A measurement-based admission control algorithm for integrated service packet networks. *IEEE/ACM Transactions on Networking (TON),* 5(1):56–70, 1997.

[48] K. Jamshaid and P.A.S. Ward. Experiences using Gateway-Enforced Rate-Limiting Techniques in Wireless Mesh Networks. *IEEE Wireless Communications and Networking Conference (WCNC),* pages 3725–3730, 2007.

[49] D.B. Johnson and D.A. Maltz. Dynamic source routing in ad hoc wireless networks. *Mobile Computing*, 353:153–181, 1996.

[50] J. Jun, P. Peddabachagari, and M. Sichitiu. Theoretical maximum throughput of IEEE 802.11 and its applications. *IEEE Conference on Network Computing and Applications (NCA),* pages 249–256, 2003.

[51] J. Jun and ML Sichitiu. Fairness and QoS in multihop wireless networks. *IEEE Vehicular Technology Conference (VTC),* 5, 2003.

[52] J. Jun and M.L. Sichitiu. The Nominal Capacity of Wireless Mesh Networks. *IEEE Wireless Communications*, page 9, 2003.

[53] C.R. Kalmanek, H. Kanakia, and S. Keshav. Rate controlled servers for very high-speed networks. *IEEE Global Telecommunications Conference (GLOBECOM),* pages 12–20, 1990.

[54] D.D. Kandhlur, K.G. Shin, and D. Ferrari. Real-time communication in multihop networks. *IEEE Transactions on Parallel and Distributed Systems*, 5(10):1044–1056, 1994.

[55] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly. Distributed multihop scheduling and medium access with delay and throughput constraints. In *Proc. of ACM MOBICOM*, pages 200–209, 2001.

[56] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly. Distributed Priority Scheduling and Medium Access in Ad Hoc Networks. *Wireless Networks*, 8(5):455–466, 2002.

[57] M. Karsten and J. Schmitt. Admission Control Based on Packet Marking and Feedback Signalling—Mechanisms, Implementation and Experiments. *TU-Darmstadt Technical Report TR-KOM-2002-03, URL: http://www. kom. e-technik. tudarmstadt. de/publications/abstracts/KS02-5. html (May, 2002)*.

[58] M. Karsten and J. Schmitt. Packet marking for integrated load control. *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 499–512, 2005.

[59] FP Kelly, PB Key, and S. Zachary. Distributed admission control. *IEEE Journal on Selected Areas in Communications*, 18(12):2617–2628, 2000.

[60] Y. Kim, S. Choi, K. Jang, and H. Hwang. Throughput enhancement of IEEE 802.11 WLAN via frame aggregation. *IEEE Vehicular Technology Conference (VTC)*, 4, 2004.

[61] L. Kleinrock. Queueing Systems, Volume I: Theory. *A Wiley-Interscience Publication*, 1975.

[62] S.S. Kunniyur and R. Srikant. An adaptive virtual queue (AVQ) algorithm for active queue management. *IEEE/ACM Transactions on Networking (TON)*, 12(2):286–299, 2004.

[63] T. Lang, P. Branch, and G. Armitage. A synthetic traffic model for Quake3. *Proc. of 2004 ACM SIGCHI*, pages 233–238, 2004.

[64] A. Lee. Achieving Fairness in 802.11-based Multi-channel Wireless Mesh Networks. Master's thesis, University of Waterloo, 2006.

[65] S. Lee, S. Banerjee, and B. Bhattacharjee. The case for a multi-hop wireless local area network. *Proc. of IEEE INFOCOM*, 2004.

[66] 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.

[67] L. Li. The Efficacy of Source Rate Control in Achieving Fairness in Wireless Mesh Networks. Master's thesis, University of Waterloo, 2007.

[68] L. Li and P.A.S. Ward. Structural Unfairness in 802.11-basedWireless Mesh Networks. *Proc. of IEEE Conference on Communication Networks and Services Research (CNSR)*, pages 213–220, 2007.

[69] D.C. Libertelli. Emerging Internet Voice Services: The Year Ahead, from http://www.acmqueue.org. 2004.

[70] J. Liebeherr, D.E. Wrege, and D. Ferrari. Exact admission control for networks with a bounded delay service. *IEEE/ACM Transactions on Networking (TON)*, 4(6):885–901, 1996.

[71] A.Y.M. Lin and J. Silvester. Priority queueing strategies and buffer allocation protocols for traffic control at an ATM integrated broadband switching system. *IEEE Journal on Selected Areas in Communications*, 9(9):1524–1536, 1991.

[72] D. Lin and R. Morris. Dynamics of random early detection. *ACM SIGCOMM Computer Communication Review*, 27(4):127–137, 1997.

[73] H. Luo, S. Lu, and V. Bharghavan. A new model for packet scheduling in multihop wireless networks. *Proc. of ACM MOBICOM*, pages 76–86, 2000.

[74] H. Luo, S. Lu, V. Bharghavan, J. Cheng, and G. Zhong. A Packet Scheduling Approach to QoS Support in Multihop Wireless Networks. *Mobile Networks and Applications*, 9(3):193–206, 2004.

[75] S. Mangold, S. Choi, GR Hiertz, O. Klein, B. Walke, P. Res, and G. Aachen. Analysis of IEEE 802.11 e for QoS support in wireless LANs. *IEEE Wireless Communications [see also IEEE Personal Communications]*, 10(6):40–50, 2003.

[76] S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz, and L. Stibor. IEEE 802.11 e Wireless LAN for Quality of Service. *Proc. European Wireless*, 18:32–39, 2002.

[77] Meraki. Meraki Networks Inc., from http://www.meraki.com. 2006.

[78] D. Niculescu, S. Ganguly, K. Kim, and R. Izmailov. Performance of VoIP in a 802.11 Wireless Mesh Network. *Proc. of IEEE INFOCOM*, pages 1–11, 2006.

[79] W. Noureddine and F. Tobagi. The transmission control protocol. *URL citeseer. ist. psu. edu/noureddine02transmission. html*.

[80] L. Pantel and L.C. Wolf. On the impact of delay on real-time multiplayer games. *Proc. of International Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 23–29, 2002.

[81] A.K. Parekh and R.G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: the single-node case. *IEEE/ACM Transactions on Networking (TON)*, 1(3):344–357, 1993.

[82] C.E. Perkins and P. Bhagwat. Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers. *Proceedings of the Conference on Communications Architectures, Protocols and Applications*, pages 234–244, 1994.

[83] J. Qiu and E.W. Knightly. Measurement-based admission control with aggregate traffic envelopes. *IEEE/ACM Transactions on Networking (TON)*, 9(2):199–210, 2001.

[84] P. Release. Atheros Communications, Atheros Ships Combo Rolling Three WLAN Standards into a Single Solution. *retrieved from www.atheros.com/news/combo. html (Mar. 11, 2002) pp*, pages 1–3.

[85] S. Shenker et al. Integrated Services in the Internet Architecture: an Overview. *IETF Request for Comments (RFC)*, 1633, 1994.

[86] J. Shin, D.C. Lee, and C.C.J. Kuo. *Quality of Service for Internet Multimedia*. Prentice Hall, 2004.

[87] M. Shreedhar and G. Varghese. Efficient fair queueing using deficit round-robin. *IEEE/ACM Transactions on Networking (TON)*, 4(3):375–385, 1996.

[88] V.A. Siris, C. Courcoubetis, and G. Margetis. Service Differentiation in ECN Networks Using Weighted Window-Based Congestion Control for Various Packet Marking Algorithms. *Proceedings of Second International Workshop on Quality of Future Internet Services, Coimbra, Portugal, 2001*, pages 190–206.

[89] R. Sivakumar, P. Sinha, and V. Bharghavan. CEDAR: a core-extraction distributed ad hoc routing algorithm. *IEEE Journal on Selected Areas in Communications*, 17(8):1454–1465, 1999.

[90] S. Sivavakeesar and G. Pavlou. Quality of service aware MAC based on IEEE 802.11 for multihop ad-hoc networks. *Wireless Communications and Networking Conference (WCNC)*, 3, 2004.

[91] J.L. Sobrinho and A.S. Krishnakumar. Quality-of-service in ad hoc carrier sense multiple access wireless networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1353–1368, 1999.

[92] Incorporated Sunrise Telecom. Working with t3 and t1:technology, troubleshooting and fault location, from http://www.sunrisetelecom.com/technotes/t1-t3_book.pdf.

[93] T. Szigeti and C. Hattingh. Quality of Service Design Overview. *Cisco Press*, 2004. http://www.ciscopress.com.

[94] A.S. Tanenbaum. *Computer Networks*. Prentice Hall PTR, 2002.

[95] Skype Technologies. Skype, from http://www.skype.com, 2005.

[96] Skype Technologies. Skype hits 100 million users: countdown, a my picture and obscure statistics, from http://share.skype.com/sites/en/2006/04/skype_hits_100_million_users_c.html, 2006.

[97] F. Tobagi and L. Kleinrock. Packet Switching in Radio Channels: Part II–The Hidden Terminal Problem in Carrier Sense Multiple-Access and the Busy-Tone Solution. *IEEE Transactions on Communications*, 23(12):1417–1433, 1975.

[98] A. Veres, AT Campbell, and M. Barry. Supporting service differentiation in wireless packet networks using distributed control. *IEEE Journal on Selected Areas in Communications*, 19(10):2081–2093, 2001.

[99] D. Verma, H. Zhang, and D. Ferrari. Guaranteeing delay jitter bounds in packet switching networks. *Proceedings of Tricomm*, 91:35–46, 1991.

[100] B. Wachtel. All You Wanted to Know About T1 But Were Afraid to Ask, from http://www.dcbnet.com/notes/9611t1.html.

[101] Y.F. WEN and Y.S.F. LIN. Fair Bandwidth Allocation and End-to-End Delay Routing Algorithms for Wireless Mesh Networks. *IEICE Transactions on Communications*, 90(5):1042, 2007.

[102] Wikipedia. T-carrier — wikipedia, the free encyclopedia, 2007. [Online; http://en.wikipedia.org/w/index.php?title=T-carrier&oldid=175491321].

[103] H. Wu, Y. Liu, Q. Zhang, and Z.L. Zhang. SoftMAC: Layer 2.5 Collaborative MAC for Multimedia Support in Multihop Wireless Networks. *IEEE Transactions on Mobile Computing*, 6(1):12–25, 2007.

[104] K. Wu and J. Harms. QoS Support in Mobile Ad Hoc Networks. *Crossing Boundaries-the GSA Journal of University of Alberta*, 1(1):92–106, 2001.

[105] H. Xiao, W.K.G Seah, A. Lo, and K.C. Chua. A flexible quality of service model for mobile ad-hoc networks. *IEEE Vehicular Technology Conference (VTC)*, 1, 2000.

[106] Y. Xiao. IEEE 802.11n: enhancements for higher throughput in wireless LANs. *IEEE Wireless Communications [see also IEEE Personal Communications]*, 12(6):82–91, 2005.

[107] K. Xu, M. Gerla, L. Qi, and Y. Shu. Enhancing TCP fairness in ad hoc wireless networks using neighborhood RED. *Proc. of ACM MOBICOM, San Diego, CA, USA*, page 1628, 2003.

[108] 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.

[109] Y. Yang and R. Kravets. Distributed QoS guarantees for realtime traffic in ad hoc networks. *IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, pages 118–127, 2004.

[110] Y. Yang and R. Kravets. Throughput guarantees for multi-priority traffic in ad hoc networks. *IEEE Conference on Mobile Ad-hoc and Sensor Systems*, pages 379–388, 2004.

[111] Y. Yang and R. Kravets. Contention-aware admission control for ad hoc networks. *IEEE Transactions on Mobile Computing*, 4(4):363–377, 2005.

[112] Y. Yang and R. Kravets. Achieving Delay Guarantees in Ad Hoc Networks Through Dynamic Contention Window Adaptation. *Proc. of IEEE INFOCOM*, pages 1–12, 2006.

[113] S. Yin, Y. Xiong, Q. Zhang, and X. Lin. Traffic-aware Routing for Real Time Communications in Wireless Multi-hop Networks. *Proc. of IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC)*, pages 568–569, 2006.

[114] H. Zhang. Service disciplines for guaranteed performance service in packet-switching networks. *Proceedings of the IEEE*, 83(10):1374–1396, 1995.

[115] H. Zhang and D. Ferrari. Rate-controlled static-priority queueing. *Proc. of IEEE INFOCOM*, pages 227–236, 1993.

[116] H. Zhang and D. Ferrari. Rate-controlled service disciplines. *Journal of High Speed Networks*, 3(4):389–412, 1994.

[117] L. Zhang. Virtual clock: a new traffic control algorithm for packet switching networks. *ACM SIGCOMM Computer Communication Review*, 20(4):19–29, 1990.

[118] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala. RSVP: a new resource reservation protocol. *IEEE Communications Magazine*, 40(5):116–127, 2002.