Throughput Guarantees for Multi-priority Traffic in Ad Hoc Networks

Yaling Yang
Department of Computer Science
University of Illinois at Urbana-Champaign
Email: yyang8@uiuc.edu

Robin Kravets
Department of Computer Science
University of Illinois at Urbana-Champaign
Email: rhk@cs.uiuc.edu

***Abstract*— In this paper, we present MPARC (Multi-Priority Admission and Rate Control), a novel joint admission control and rate policing protocol for multi-priority ad hoc networks. MPARC is based on our novel bandwidth allocation model and guarantees that the throughput of admitted realtime flows will not decrease due to later arriving realtime flows with equal or lower priorities or due to best effort flows. MPARC achieves this goal by performing accurate admission control on every newly arriving realtime flow and appropriate rate policing on all best effort traffic. Through simulation, we demonstrate that MPARC has better performance than existing approaches.**

I. Introduction

The fast spread of small wireless computers has enabled the design and deployment of wireless ad hoc networks. Typical applications proposed for such networks include both realtime and non realtime applications. While realtime applications, such as conversational audio/video conferencing or on-demand multimedia retrieval, require quality of service (QoS) guarantees for effective communication, best effort applications, such as file transfer, are more tolerant to the changes of bandwidth and delay and generally always has backlogged packets for transmission. Supporting both types of applications in an ad hoc network is challenging due to the shared nature of the underlying wireless communication channel. The goal of our research is to provide QoS guarantees, especially throughput guarantees, for realtime traffic in the presence of best effort traffic and at the same time achieve efficient network utilization.

Providing QoS support in ad hoc networks requires support from the MAC layer to regulate access to the wireless channel. Given this tight coupling, most QoS schemes are designed for a specific MAC layer scheme. In this paper, we focus on networks based on IEEE 802.11 [17] types of MAC protocols. While IEEE 802.11 is often proposed for ad hoc networks due to its wide availability and simple and robust contention-based access mechanism, IEEE 802.11 does not provide any assurance or service differentiation for the throughput of flows. Recently, it has been proposed to extend IEEE 802.11 to support service differentiation by dividing traffic into different classes that use different contention related parameters (e.g., minimum contention window size, maximum MAC frame size, etc.) [1], [10]. However, these extensions still do not provide any guarantees for the throughput of realtime flows. As the wireless channel becomes overloaded and the number of competing flows increases, the bandwidth share of each flow may decrease. Our focus is to support throughput guarantees in ad hoc networks that use IEEE 802.11 or its MAC layer extensions for service differentiation.

QoS support for realtime flows in ad hoc networks requires three main components. First, admission control must be used to prevent new realtime flows from consuming too many resources and disrupting the guarantees made to existing realtime flows. Second, rate policing must be used to control the sending rate of best effort traffic to prevent it from degrading the QoS of existing realtime flows. Essentially, best effort traffic is given a lower priority than realtime traffic. Finally, considering that ad hoc networks are proposed for in search and rescue environments, it is important to classify and prioritize realtime traffic so that an important flow will not be blocked due to existing lower priority flows.

Based on the above requirements, the goal of our research is to provide an effective multi-priority based admission control protocol for realtime traffic and a rate policing protocol for best effort traffic for wireless ad hoc networks based on IEEE 802.11 and its extensions to service differentiation (e.g., IEEE 802.11e [10] and [1]). Our joint admission control and rate policing protocol, MPARC (**M**ulti-**P**riority **A**dmission and **R**ate **C**ontrol), guarantees that the throughput of an admitted realtime flow can be maintained and will not be disrupted by newly arriving realtime flows with equal or lower priorities or by best effort flows. Our admission control protocol may admit a higher priority realtime flow even if this higher priority flow degrades the QoS of some existing lower priority realtime flows and best effort flows. Our rate policing protocol for best effort traffic ensures that best effort traffic does not hurt any existing realtime flows while it is allowed to fill the bandwidth that is not used by realtime traffic.

Admission control for realtime traffic and rate policing for best effort traffic are essentially a problem of determining available bandwidth. For admission control, the available bandwidth of a new realtime flow is defined as the maximum amount of bandwidth that the new flow can consume without degrading the throughput of existing equal or higher priority flows. If this available bandwidth is smaller than the required bandwidth of the new flow, admission fails. For rate policing, the available bandwidth for all best effort traffic is defined as the maximum bandwidth that best effort traffic can consume without degrading the throughput of any existing realtime flows. The sharing of the available bandwidth of best effort traffic between best effort flows is determined by transport protocols such as TCP. Through competition at the transport layer, a new best effort flow is allowed to reduce the throughput of existing best effort flows. However, rate policing controls the total sending rate of all best effort flows so that their bandwidth consumption is no larger than the available bandwidth to best effort traffic.

In current wired networks, such admission control and rate policing are mainly performed at routers, which have centralized control and global knowledge of the allocations of their link bandwidth. A multihop realtime flow can simply find its available bandwidth at any of the nodes along the route to determine its end-to-end available bandwidth and then make admission decisions. A router can simply police the rate of best effort traffic by dropping packets or scheduling realtime traffic before best effort traffic.

However, due to the differences between wireless networks and wired networks, providing the same admission control and rate policing in ad hoc networks is more challenging than in wired networks. This challenge is due to the difficulties of providing accurate available bandwidth estimation. In wireless networks, since the channel is shared, there is no centralized control on how bandwidth is allocated between flows located at different nodes. Therefore, it is non-trivial to estimate the maximum amount of bandwidth that a new flow is able to get by contending with existing flows. Additionally, since nodes that are contending for the channel have no knowledge of the priorities of the flows on other nodes, there is no centralized scheduler to guarantee that a higher priority packet is sent before a lower priority packet. Therefore, a new flow can potentially affect the throughput of all existing flows in all priority levels. Hence, the impact of a newly added flow on the throughput of existing flows is not easy to quantify.

Current admission control algorithms for wireless networks take one of three approaches to estimate available bandwidth. The first approach, such as VMAC [3], uses free channel bandwidth as an estimate for available bandwidth. This approach does not support priorities between flows. A best effort flow of a file transfer can occupy all of the channel bandwidth and prevent the admission of any realtime traffic. This is not desirable since a best effort flow is designed to adapt to changes in throughput. To improve the performance of the network, a realtime flow should be allowed to “push” best effort flows to get its desired bandwidth. In the second approach [8], [15], [16], a node uses the channel access time of its current traffic to calculate the available bandwidth of a new flow. This approach has two drawbacks. First, it does not consider the impact of admitting a new flow on other existing flows, hence it can not prevent the newly admitted flow from degrading the QoS of existing flows. Second, it does not consider the fact that as a new flow is added into the network, the competition for bandwidth intensifies and the channel access time increases. Therefore, the bandwidth estimation before the new flow starts is often larger than the actual bandwidth allocation to the new flow when it actually starts. The third approach [2], [11], [14], estimates available bandwidth under a very conservative assumption that every active nodes in the network is saturated (i.e., every node always has backlogged packets). This assumption is based on an extreme state of the network where all active nodes are overloaded, which is not likely to always be true and which should be avoided to support throughput guarantees to realtime traffic. Therefore, this approach is overly pessimistic and may reduce the capacity of the network for realtime flows. A more detailed analysis of these three existing approaches can be found in our prior work in [18].

Due to the drawbacks of existing approaches, we propose MPARC, a joint admission control and rate policing protocol, which is based on our novel model of bandwidth allocation that captures bandwidth sharing between competing traffic classes in all possible network states [18]. Using this model for accurate estimation of available bandwidth, MPARC identifies the effects of adding a new realtime flow and identifies the amount of best effort traffic that can be supported. MPARC makes priority-based admission control decisions about realtime traffic and controls the rate of best effort traffic so that throughput guarantees for realtime traffic are maintained.

In Section II, we briefly review IEEE 802.11 and its extensions for service differentiation. In Section III, we briefly introduce our novel bandwidth allocation model for a single hop network. In Section IV, we address the extensions of the single hop model to a multihop environment. Section V discusses how MPARC performs admission control and rate policing based on this model. Section VI evaluates the performance of MPARC and compares it with the free, delay model and saturation models. Section VII concludes our work.

II. IEEE 802.11 Protocol and Its Extensions

The IEEE 802.11 standard provides two functions in the MAC sublayer: the distributed coordination function (DCF) and the point coordination function (PCF). PCF provides contention-free frame transfer. Since PCF requires a Point Coordinator in the Access Point, it is not appropriate for a multihop wireless network. Hence, we only examine admission control for DCF and the extensions to DCF.

A. IEEE 802.11 DCF Mode

IEEE 802.11 DCF provides automatic medium sharing between nodes through the use of CSMA/CA and a random backoff time following a busy medium. Prior to transfer of data packets, a node invokes the carrier-sense mechanism to determine the busy/idle state of the medium. If the medium is idle, the node defers for a constant period of time, called *DCF interframe space* (DIFS), which is determined by the physical layer. If the medium stays idle during this DIFS period, the node may transmit its packet. If the medium is busy, the node waits until the medium is observed to be idle. The length of this idle period depends on the success or failure of the previous frame. If the last frame was received correctly, the node waits DIFS time units. If the last frame was not received correctly, the node waits *extended interframe space* (EIFS) time units. After this DIFS or EIFS idle time, the node selects a random backoff period for deferring before transmitting an RTS. If the backoff timer already contains a non-zero value, the selection of a random number is not needed. The backoff period is calculated as *Backoff Time = Random()* × *aSlotTime*, where $Random()$ is a pseudo-random integer drawn from a uniform distribution over the interval [0,CW]. CW, called the *contention window*, is an integer within the range of *minimum contention window* (CW^{min}) and *maximum contention window* (CW^{max}) (i.e., $CW^{min} \leq CW \leq CW^{max}$).

For the first transmission attempt of each packet, CW is set to CW^{min}. After each unsuccessful transmission, the value of CW is doubled (*binary exponential backoff*), up to the maximum value, CW^{max}. The backoff time is decremented by $aSlotTime$ period if the channel is idle during this period and stopped when a transmission is detected on the channel. $aSlotTime$ is a constant value determined by the physical layer of the network. The backoff timer is reactivated when the channel is sensed idle again for more than DIFS time. The node transmits when the backoff timer reaches zero. At the end of every successful transmission, the CW value reverts to CW^{min} and a backoff procedure is performed immediately, even if no additional transmissions are currently queued.

B. Service Differentiation Extensions of DCF Mode

In recent years, several approaches have been proposed to provide service differentiation in IEEE 802.11 by adjusting contention related parameters [1], [10]. In these approaches, packets from different classes are put into different queues in a node. Each queue acts like a virtual node that observes the channel and contends for the channel independently (e.g. IEEE 802.11e [10]). Therefore, in the rest of the paper, we assume that each node (which may be a virtual node) only carries traffic of a single class.

Depending on the contention related parameters that are adjusted, current approaches can be separated into four categories [1], [10]. First, different classes of traffic are assigned different CW^{min}. Second, different classes are assigned different packet sizes. Third, different exponential backoff schemes are used to adjust contention windows after a collision. Fourth, the DIFS is different from class to class (called AIFS in IEEE 802.11e). In [1], it shows that the service differentiation effect of the third category is less obvious and less stable than the first two categories since it only takes effect when collisions happen, which are rare events compared to ordinary packet transmission. Therefore, the differentiation schemes in the third category is not the focus of this paper. The schemes in the fourth category may suffer from inefficient channel usage since even if the majority of the traffic is from the class with the larger DIFS, they all must wait a very long period of time before they can compete for the channel. Due to this drawback, the differentiation schemes in the fourth category are again not the focus of this paper. Instead, we focus on the first and second types of methods where service differentiation is realized through different CW^{min}'s and frame sizes.

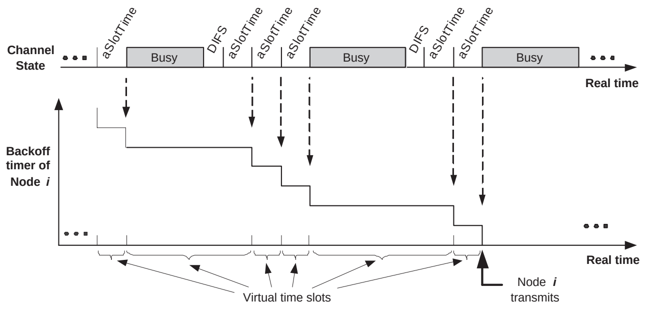


Fig. 1. Virtual time slots

III. Bandwidth Allocation Model

In this section, we briefly introduce our novel model of bandwidth allocation in a single hop network (detailed analysis and proofs can be found in [18]). In our model, a discrete Markov process model of wireless channel is used to examine the behavior of saturated and non-saturated nodes in three network states, saturated, non-saturated and semi-saturated. Our model enables accurate estimation of bandwidth allocation for nodes in these three network states. The extension of this model to a multihop network is discussed in Section IV and Section V shows how this model can be used for the admission control and rate policing in MPARC.

A. Channel Model

In the single hop model, there is a fixed set $\mathcal{N} = \{1, 2 \ldots, n\}$ of transmitting nodes and every node can hear each other's transmissions. Using the method derived in [4], real time can be divided into *virtual time slots*, where a node decrements its backoff timer once per virtual time slot. Consider the example shown in Figure 1. Node i's virtual time slots come in two types. First, a virtual time slot equals $aSlotTime$ when the channel is idle (e.g., Node i's first virtual time slot). However, Node i's second virtual time slot extends from the beginning of the busy period until the end of the $aSlotTime$ period, since the backoff timer is not decremented until after the channel becomes idle for a DIFS period. There can be at most one packet sent in a virtual time slot. If multiple nodes attempt to send a packet in the same virtual slot, a collision happens. By dividing real time into virtual time slots, the backoff process of a node can be modeled as a discrete Markov process (for details see [4]).

B. States of Nodes

To perform admission control, it is necessary to understand the bandwidth allocation in the network, which depends on the states of the nodes. A node in a wireless network can be in two states: saturated and non-saturated. A saturated node always has backlogged packets while a non-saturated node often has an empty queue. This section briefly presents the relationship between bandwidth allocation and node states and shows that the bandwidth share of a node depends on the states of all competing nodes in the network.

Let S_i be the amount of bandwidth allocated to a node $i \in \mathcal{N}$ and P_i be the probability that the node successfully transmits a packet in a virtual slot. Subscript sat and $\overline{sat}$ are used to indicate saturated and non-saturated nodes respectively. For example, $S_{i,\overline{sat}}$ represents Node i's bandwidth when Node i is a non-saturated node. W_i and L_i denote the minimum contention window size and frame size for Node i respectively, allowing our model to support service differentiation.

The bandwidth allocated to a Node i is related to the collision probability of its packets, ϕ_i, the probability that it transmits in a randomly chosen virtual time slot, τ_i, and its load in terms of packets per second, R_i. For a saturated node, such relationship is captured in the following theorem.

Theorem 1: For a saturated Node i,

1)

$$P_{i,sat} = \frac{\tau_{i,sat}}{1-\tau_{i,sat}} \prod_{j=1}^{n} (1-\tau_j), \tag{1}$$

$$\tau_{i,sat} = \frac{2(1-2\phi_i)}{(1-2\phi_i)(W_i+2)+\phi_i(W_i+1)(1-(2\phi_i)^{m_i})}, \tag{2}$$

where m_i is the number of collisions that are needed for the contention window size to reach CW^{max}.

2) $S_{i,sat}$ is the maximum bandwidth allocation of Node i and

$$S_{i,sat} = \frac{P_{i,sat} L_i \sum_{j=1}^{n} S_j}{\sum_{j=1}^{n} P_j L_j}. \tag{3}$$

3) Node i is a saturated node if and only if the total amount of traffic that Node i needs to send is larger than its maximum bandwidth allocation.

$$S_{i,sat} < R_i L_i. \tag{4}$$

Theorem 1 shows that the maximum bandwidth allocation to a saturated node is constrained by its W_i and ϕ_i. For a non-saturated node, since its queue often is empty, the limiting factor of its bandwidth allocation is actually its load R_i.

Theorem 2: For any non-saturated Node i,

$$S_{i,\overline{sat}} = R_i L_i, \tag{5}$$

$$S_{i,\overline{sat}} \leq S_{i,sat}, \tag{6}$$

$$P_{i,\overline{sat}} = \frac{R_i \sum_{j=1}^{n} P_j L_j}{\sum_{j=1}^{n} S_j}, \tag{7}$$

$$P_{i,\overline{sat}} < P_{i,sat}, \tag{8}$$

As can be seen from Equations (3) and (5), the bandwidth allocation to a saturated node depends on both the node's own state and the bandwidth allocations of the other nodes, which in turn is related to the state of the other nodes. Essentially, the bandwidth allocation to a node is related to the congestion level of the whole network.

C. States of Networks

In this section, we classify the congestion level of a network into three states and illustrate the relationship between the bandwidth allocations and these three states. The formulation of this relationship is presented in Section III-D.

Depending on the traffic types and load, an IEEE 802.11 network can be in one of three states: saturated, non-saturated or semi-saturated. A network is in a *saturated state* when every node always has backlogged packets. In a *non-saturated network*, every node is non-saturated, indicating a lightly loaded network. A *semi-saturated* network is between the saturated and non-saturated state, where some of the nodes are saturated while other nodes are non-saturated.

To better illustrate the relationship between bandwidth allocation and network state, we present a simple NS2 [6] simulation using the topology shown in Figure 2. The channel capacity of the network is 2Mbps. The queue size in each node is 50 packets. The packet size is 512 bytes. The simulation runs for 150 seconds. There are four nodes in the network with Nodes 1 and 2 transmitting to Nodes 3 and 4, respectively.

Figures 3 and 4 depict the queue length and the throughput of Nodes 1 and 3. From time 5s to time 50s, Nodes 1 and 3 each carry a realtime flow that generates 50 packets per second. The queues in both nodes are often empty during this period, indicating a non-saturated network. Both flows can achieve throughput that matches their packet generation rates. At time 50s, the traffic type of Node 1 changes to a file transfer. The queue in Node 1 becomes full while the queue in Node 3 is still often empty, indicating a semi-saturated network. During this period, even though Node 1 tries to send more packets, it is not able to "push down" Node 3's bandwidth allocation. From time 100s to time 150s, the realtime traffic in Node 3 increases its generating rate to 300 packets per second. Both queues in Node 1 and Node 3 become constantly full, indicating a saturated network. During this period, Nodes 1 and 3 share the channel bandwidth equally and the realtime traffic in Node 3 is unable to achieve its desired bandwidth.

This example shows that bandwidth allocations are related to the state of the network. Depending on the traffic load and type, a practical network can be in any of the three states. Therefore, an effective admission control protocol must capture the bandwidth allocation in all network states.

D. Bandwidth Allocation for Different Networks States

In this section, we briefly present the analytical results for bandwidth allocation in saturated, non-saturated or semi-saturated networks. In Section V, these results are used by MPARC to perform admission control and rate policing.

1) Semi-saturated Network: Consider a semi-saturated network, where the set of saturated nodes is N_1, the set of non-saturated nodes is N_2 and $N_1 \cup N_2 = \mathcal{N}$. Since the saturated nodes in the network always have packets to transmit and hence fill up the network bandwidth,

$$\sum_{i=1}^{n} S_i \approx C, \tag{9}$$

where C is the maximum throughput of the channel. To solve the S_i for any Node i, it is necessary to determine the state of Node i. Theorems 1 and 2 show that Node i's bandwidth allocation S_i has an upper bound determined by $S_{i,sat}$ and $R_i L_i$. If $S_{i,sat}$ is larger than its load $R_i L_i$, Node i is non-saturated and its bandwidth allocation equals $R_i L_i$. If $S_{i,sat}$ is smaller than $R_i L_i$, Node i is saturated and its

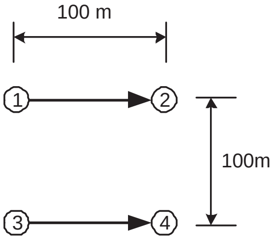


Fig. 2. Topology

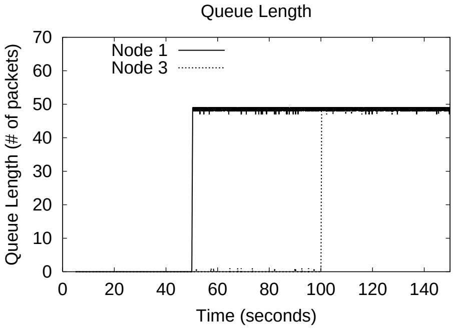


Fig. 3. Queue length of Nodes 1 and 3

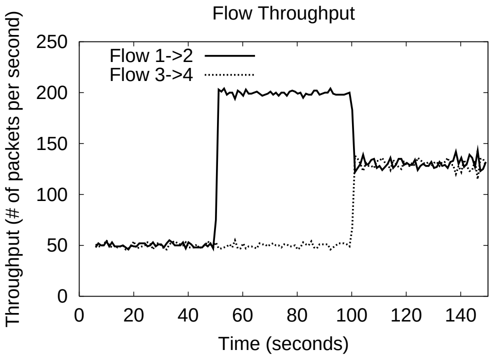


Fig. 4. Throughput of Nodes 1 and 3

bandwidth allocation becomes $S_{i,sat}$. Therefore, as long as $S_{i,sat}$ is known, the bandwidth allocation of Node i can be easily determined according to the offered load on Node i. Based on Theorems 1 and 2, $S_{i,sat}$ in a semi-saturated network can be expressed as:

$$S_{i,sat} = \frac{L_i C}{\eta W_i}, \tag{10}$$

where $\eta = \frac{\sum_{i \in N_1} \frac{L_i}{W_i}}{1 - \sum_{i \in N_2} \frac{R_i L_i}{C}}$ and represents the congestion level of the network.

Equation (10) shows that the maximum bandwidth allocation to Node i, which equals $S_{i,sat}$, is determined by the η of the whole network as well as Node i's own parameters W_i and L_i. The larger the η, the smaller the $S_{i,sat}$. According to Theorem 1, when $R_i L_i > S_{i,sat}$, Node i becomes saturated. Therefore, as η increases, $S_{i,sat}$ decreases so that more and more nodes in the network become saturated. When Node i is at the edge of turning from non-saturated to saturated, $R_i L_i = S_{i,sat}$. Combined with Equation (10), the threshold value of η at this turning point, η_i^*, can be expressed as:

$$\eta_i^* = \frac{C}{R_i W_i}. \tag{11}$$

Sorting the nodes according to their η_i^* in ascending order results in a sequence of nodes $(x_1, x_2, \ldots, x_n)$ where $\eta_{x_i}^* \leq \eta_{x_j}^*$ if $i < j$. If $\eta_{x_k}^* < \eta < \eta_{x_{k+1}}^*$, nodes $x_1, \ldots, x_k$ are saturated and nodes $x_{k+1}, \ldots, x_n$ are non-saturated. Therefore,

$$\eta \quad = \quad \eta(k) = \frac{\sum_{i=1}^{k} \frac{L_{x_i}}{W_{x_i}}}{1 - \sum_{i=k+1}^{n} \frac{R_{x_i} L_{x_i}}{C}}, \tag{12}$$

$$\eta_{x_k}^* \quad \leq \quad \eta(k) < \eta_{x_{k+1}}^*. \tag{13}$$

Since the range of k is the number of competing neighboring nodes, which is generally not large, we can calculate the value of η corresponding to each value of k using Equation (12). The value of η that satisfies the inequality constraint (13) is a valid solution to η and determines the value of k. With the value of η and k, the state of the nodes can be decided, where the saturated nodes are $N_1 = \{x_1, x_2, ..., x_k\}$ and and the non-saturated nodes are $N_2 = \{x_{k+1}, x_{k+2}, ..., x_n\}$. The bandwidth allocation to every node can be determined as:

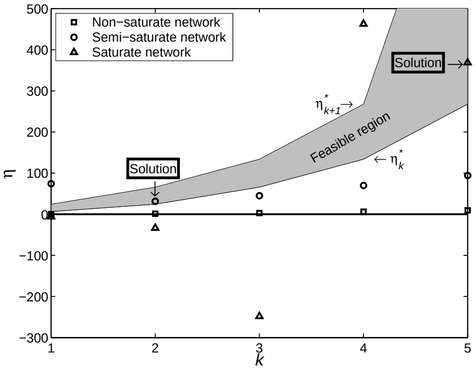


Fig. 5. Example of η, η^* and the corresponding solution

$$S_i = \begin{cases} \frac{L_i C}{\eta W_i}, & i \in N_1, \\ R_i L_i, & i \in N_2. \end{cases} \tag{14}$$

2) Saturated and Non-saturated Networks: Note that in deriving Equation (14), we assume that the network is semi-saturated, meaning that both N_1 and N_2 are non-empty. However, it is also possible that the network is saturated or unsaturated. By setting $\eta_{x_{n+1}}^* = \infty$, the solution of η for a saturated network is obtained at $k = n$, where $\eta_{x_{n+1}}^* > \eta(n) = \sum_{i=1}^{n} \frac{L_{x_i}}{W_{x_i}} > \eta_{x_n}^*$. In this case, N_2 is empty and it is easy to check that Equation (14) is still valid for calculating S_i, although only the part corresponding to N_1 is used. When none of the nodes in the network are saturated, there is no solution to η since $0 < \eta(k) < \eta_{x_k}^*$ holds for all $1 \leq k \leq n$. In this case, N_1 is empty and it is easy to check that Equation (14) is still valid for calculating S_i, although only the part corresponding to N_2 is used.

Figure 5 shows an example of the $\eta(k)$ in a five node network in saturated, non-saturated and semi-saturated states, respectively. The points in the figure represent the values of η corresponding to k calculated using Equation (12). The inequality constraint (13) is represented by the shaded area. When a point for η is located in the shaded area, the point represents a valid solution for η. In Figure 5, the solution for a saturated network is achieved when $k = 5$, the solution for

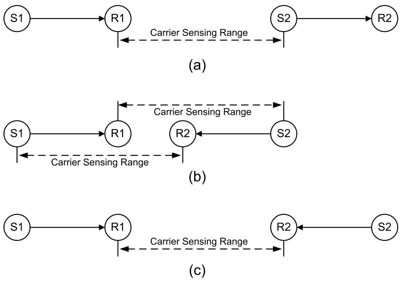


Fig. 6. Topologies with Hidden Terminals. S1 and S2 are sending nodes. R1 and R2 are receiving nodes.

a semi-saturated network is achieved when $k = 2$, and the non-saturated network has no solution for $1 \leq k \leq 5$.

IV. Multihop Extensions

To extend our model to multihop ad hoc networks, we must address two assumptions that hold in a single hop network may not be true for multihop networks. First, unlike a single hop network where active nodes can hear each other, in a multihop network, two active nodes may not hear each other but can still affect each other's throughput due to the hidden terminal problem. Second, in a single hop network, every flow is only one hop, hence the rate of the flow is the bandwidth consumption of the flow. However, in a multihop network, a flow may travel through multiple nodes and each of the nodes on its route requires a bandwidth allocation that equals the rate of the flow. In this section, we discuss the impact of these two differences on the accuracy of our bandwidth allocation model and extend the model to multihop ad hoc networks.

A. *Effects of Hidden Terminals*

In this section, we examine how hidden terminals affect bandwidth allocation in multihop networks. The hidden terminal problem happens when the receiving node contends with nodes that the sending node cannot detect. Figure 6 shows typical topologies for the hidden terminal problem. In all three topologies, Nodes S1 and R1 are in transmission range and Nodes S2 and R2 are in transmission range.

In Figure 6(a), Node S2 is in carrier-sensing range of Node R1, but outside carrier-sensing range of S1. Since S2 can only detect but not decode the transmission from R1, S2 does not know the duration of the transmission between S1 and R1. If S2 starts sensing the channel while R1 is sending the CTS to S1, S2 waits until R1 finishes sending, waits a period of EIFS and then tries to access the channel again. At this time, even though S1 is busy sending R1 the DATA packet, S2 is not able to detect it. Therefore, S2 transmits its RTS to R2, which may corrupt R1's reception of the DATA packet depending on the ratio of received signal strength at R1. Furthermore, since S2 is transmitting to R2, R1 detects S2's activity and does not respond to S1's RTS. However, S1 does not know when S2's transmission ends, and therefore, S1's retransmission attempts have a high chance to collide with S2's transmission activity again. After six failed retransmissions, S1 decides that the link between S1 and R1 is broken. Therefore, when S2 transmits, it gets all the bandwidth, while S1 gets none, causing long term unfairness between S1 and S2.

In Figure 6(b), S1 is using the channel to communicate with R1 when S2 gets a packet to transmit. The only transmission activities that S2 can detect (but can not understand) are the short CTS and ACK packets from R1. Therefore, as S2 sends out its RTS, chances are great that the packet collides with the DATA packet from S1 at R2. Since R2 can sense the DATA packet from S1, R2 does not respond to the RTS from S2. Therefore, after six retransmission attempts, S2 gives up and the MAC layer in S2 reports a broken link. If by chance S2 successfully gets the channel, S1 will have a difficult time to compete with S2. In brief, the node that gets the channel once has a high probability to win the channel in its subsequent channel access attempts. Therefore, S1 and S2 alternate accessing the channel for a long periods. The throughput of S1 and S2 has large variations and shows short term unfairness, although long term allocations are fair.

In Figure 6(c), R1 and R2 can only detect each other's CTS and ACK packets. These packets are relatively short, so that both R2 and R1 have a chance to respond to RTS packets from S2 and S1 respectively. Depending on the received signal strength at R2, R1's activity may or may not corrupt the packets that R2 tries to receive. If no corruption happens due to the capture effects, both S1 and S2 can send their packets independently except when R1 and R2 detect each other's CTS or ACK packets. Therefore, the bandwidth allocation to S1 (or S2) is the full channel bandwidth minus the fraction of bandwidth consumed by the CTS and ACK from R2 (or R1). Therefore, in this case, S1 and S2 share the bandwidth fairly, although a high collision rate is expected.

These examples show that with hidden terminals, a node's bandwidth allocation is related to the receiver's contention environment and the location of competing flows. A node's bandwidth allocation may also vary dramatically if it has the second hidden terminal problem. To predict which hidden terminal problem a flow may suffer from requires precise knowledge of the node's neighborhood and hence is not practical to implement in real networks. Although none of the existing admission control protocols consider hidden terminals, we expect that this unfairness caused by location can be alleviated in a multi-flow environment. For example, in the first example that exhibits the strongest unfairness, if Node S1 and Node S2 have a common neighbor, Node S3, that is transmitting, both S1 and S2 sense it. As soon as S3 finishes transmitting, S1 and S2 start contending simultaneously and, therefore, contend fairly. Although we expect that our model is also not precise in the presence of hidden terminals, its performance is accurate enough to have practical usage.

B. *Multihop Flows*

For a multihop flow in an ad hoc network, the bandwidth consumption of the flow at a node on the route is not equal to the rate of the flow. This is because the other nodes on the route of the flow also contend for the bandwidth. For example,

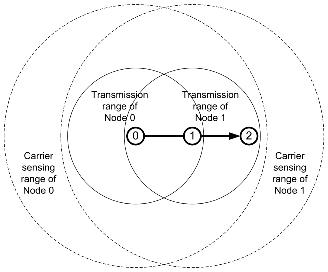

Transmission range of Node 0
Transmission range of Node 1
0
1
2
Carrier sensing range of Node 0
Carrier sensing range of Node 1

Fig. 7. Multihop Flows

in Figure 7, a flow goes through route $0 \rightarrow 1 \rightarrow 2$. Since Node 0 and Node 1 are in each other's carrier-sensing range, only one node can transmit at a time. Therefore, Node 0 must share its bandwidth with Node 1. The bandwidth consumption of the flow, B_f, can be expressed as $B_f = 2R_fL_f$, where R_f is the rate of the flow and L_f is the frame size of the flow. To generalize, for a Node V, if there are α nodes (which may include Node V itself) on the route of the flow that are also in the carrier-sensing range of Node V, the bandwidth consumption of the flow at Node V is $\alpha R_f L_f$. Therefore, admission control of a multihop flow must consider the value of α to determine the bandwidth consumption of the flow.

V. Admission Control and Rate Policing

In Sections III and IV, we introduced our bandwidth allocation model that is the basis for our admission control and rate policing protocol MPARC. In this section, we discuss the design of MPARC.

A. Collection of Neighbor Information

The analysis in Section III shows that a node's bandwidth allocation is related to the loads and traffic classes of its competing neighbors. Therefore, to ensure that a newly added flow can obtain its desired QoS without degrading the bandwidth allocation of existing flows, it is necessary to collect traffic information at a node's competing neighbors, which includes reservations and classes of realtime traffic and the average packet arrival rate and size of best effort traffic. Since a node contends for bandwidth not only with its neighbors in its transmission range, but also with its near-neighbors in carrier-sensing range, the node must collect multihop neighbors' traffic information. In our experiments, we use three hops as the collection range. This is purely a heuristic and does not guarantee to involve all contending nodes and may involve non-contending nodes. More elaborate methods, such as using the locations of nodes to decide contention relationships, may be used to improve the accuracy of finding contending nodes.

In MPARC, every node periodically broadcasts its traffic information in its one-hop neighborhood. The broadcast message also carries traffic information of its two hop neighbors, which it has gathered through listening to other nodes' broadcasts. Using this method, every node learns the traffic for competing nodes in its three-hop neighborhood. Besides periodic updates, a triggered update can also be performed when a new reservation is made. The packet overhead of update messages can be reduced by piggybacking load information on control and data packets, adding minimal overhead to heavily loaded networks.

B. Admission Control

In this section, we discuss the admission control part of MPARC in terms of the *signaling process* and the *bandwidth prediction function*, which is a function that is stored at every node and is used to identify whether a new flow can achieve its desired rate and at the same time not decrease the throughput of existing flows with equal or higher priorities.

1) Signaling Process: We assume that before admission control is performed, some ad hoc routing protocol (e.g., DSR [7], DSDV [13] or AODV [12]) has been used to find the route for a new flow. Then QoS signaling protocols, such INSIGNIA [9] or RSVP [5], can be used to setup admission control and resource reservation at each node along the route. In brief, a reservation request message, which carries the flow route, packet length, traffic class and flow rate information, is sent along the route of the new flow. Each node that receives this message performs admission control using its bandwidth prediction function. If admission control succeeds, a soft bandwidth reservation is made and the reservation request message is forwarded to the next hop. If admission control succeeds at every node, this route has enough bandwidth for the new flow and the new flow can start. If admission control fails at some node, the flow is rejected and the reservation is torn down using explicit messages or timeouts.

2) Building the Bandwidth Prediction Function: To build its bandwidth prediction function, Node V learns the traffic loads for its n competing neighbors through the periodic exchange of traffic load information. For a new flow through Node V, there are α nodes (including Node V itself) along this route that are also in Node V's three-hop neighborhood. The frame size of the new flow's traffic class is L_{new} and the rate is R_{new}. If the new flow is admitted and achieves its desired rate, its flow rate will be aggregated with Node V's existing realtime traffic, which is of the same class as the new flow since Node V carries only one class of traffic (See Section II-B). The load that the new flow will impose on the network, U_{new}, can be expressed as:

$$U_{new} = \alpha R_{new} L_{new}. \tag{15}$$

Similar to Section III-D, the competing nodes are sorted according to their saturation threshold η_i^* in ascending order to get a sequence of nodes $(x_1, x_2, \ldots, x_n)$ where $\eta_{x_i}^* \leq \eta_{x_j}^*$ if $i < j$. Based on Equation (12), if the new flow is admitted, the new η at node V can be expressed as:

$$\eta = \frac{\sum_{i=1}^{k} \frac{L_{x_i}}{W_{x_i}}}{1 - \sum_{i=k+1}^{n} \frac{R_{x_i} L_{x_i}}{C} - \frac{U_{new}}{C}}, \tag{16}$$

where $\eta_{x_k}^* < \eta < \eta_{x_{k+1}}^*$. Solving for U_{new}, we get the

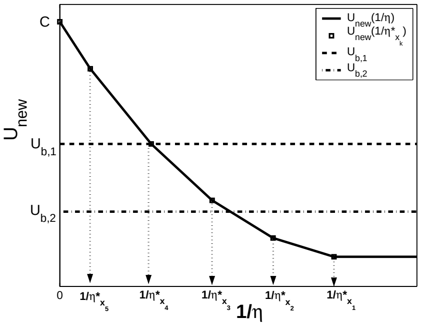


Fig. 8. Piecewise linear function of U_{new} and $1/\eta$

bandwidth prediction function:

$$U_{new} = C \times \begin{cases} 1 - \sum_{i=k+1}^{n} \frac{R_{x_i} L_{x_i}}{C} - \frac{1}{\eta} \sum_{i=1}^{k} \frac{L_{x_i}}{W_{x_i}}, \\ \qquad\qquad\qquad \text{for } \eta^*_{x_k} \leq \eta < \eta^*_{x_{k+1}}, \\ 1 - \sum_{i=1}^{n} \frac{R_{x_i} L_{x_i}}{C}, \quad \text{for } 0 \leq \eta < \eta^*_{x_1}, \\ 1 - \frac{1}{\eta} \sum_{i=1}^{n} \frac{L_{x_i}}{W_{x_i}}, \quad \text{for } \eta^*_{x_n} \leq \eta. \end{cases} \tag{17}$$

Note that the bandwidth prediction function is a piece-wise linear function of U_{new} and $1/\eta$, which can be pre-calculated and stored in a node. An example of the bandwidth prediction function is shown in Figure 8, where there are five competing nodes. The bandwidth prediction function consists of six line segments. The five end points of these line segments correspond to the $1/\eta^*_i$'s of the five competing nodes, which can be easily calculated based the traffic load information and Equation (11). It can be seen that a larger U_{new} corresponds to smaller $1/\eta$ in the bandwidth prediction function. As $1/\eta$ becomes smaller than the reciprocal of a Node i's saturation threshold $1/\eta^*_i$, Node i is pushed to its saturated state by the new flow and the throughput of Node i's flows decreases.

3) Using the Bandwidth Prediction Function: When Node V receives a reservation request message, it can easily calculate U_{new} using Equation (15), where α is obtained by comparing the route information in the reservation request message to the identities of Node V three-hop neighbors. The bandwidth prediction function can be used to calculate two upper bounds, denoted $U_{b,1}$ and $U_{b,2}$, of U_{new}, which determine the maximum value of U_{new}. The two upper bounds are related to the priorities of existing flows and Node V's own saturation threshold. If a new flow requires a U_{new} that is larger than either upper bound, the flow must be rejected due to lack of bandwidth.

The first upper bound, $U_{b,1}$, is defined by the priorities of existing flows, since the new flow should not degrade the throughput of any existing flows with equal or higher priorities. Therefore, using the prediction function in Equation (17), $U_{b,1}$ can be expressed as:

$$U_{b,1} = C \left(1 - \sum_{i=\gamma}^{n} \frac{R_{x_i} L_{x_i}}{C} - \frac{1}{\eta^*_\gamma} \sum_{i=1}^{\gamma-1} \frac{L_{x_i}}{W_{x_i}} \right), \tag{18}$$

where Node x_γ is the first node starting from Node x_1 that carries traffic with equal or higher priority than the new flow.

The second upper bound is defined by the saturation threshold η^*_V of Node V itself. Based on Equation (11), if the new flow is admitted, η^*_V becomes:

$$\eta^*_V = \frac{C}{R_V W_V} = \frac{C}{(R_{new} + R_{V,old}) W_V}, \tag{19}$$

where $R_{V,old}$ is existing traffic in Node V before the new flow starts. Equation (19) shows that when R_{new} increases, η^*_V decreases. Additionally, when R_{new} increases, U_{new} increases (See Equation (15)), therefore $1/\eta$ decreases (See Equation (17)). Hence, the η of the network may first reach Node V's saturation threshold η^*_V before U_{new} hits $U_{b,1}$. After this, the new flow will not be able to achieve any larger sending rate since Node V is saturated. Therefore, the second upper bound on U_{new}, denoted as $U_{b,2}$ can be expressed as:

$$U_{b,2} = C \left(1 - \sum_{i=v}^{n} \frac{R_{x_i} L_{x_i}}{C} - \frac{1}{\eta^*_V} \sum_{i=1}^{v-1} \frac{L_{x_i}}{W_{x_i}} \right), \tag{20}$$

where $x_v = V$ and $\eta^*_{x_{v-1}} < \eta^*_V < \eta^*_{x_{v+1}}$. Figure 8 shows the case when $R_{b,1}$ is larger than $R_{b,2}$.

The two upper bounds determine whether a new flow should be admitted. When Node V needs to perform admission control on a new flow, based on the rate of the new flow R_{new} and its priority, Node V can use the bandwidth prediction function in Equation (17) to calculate $U_{b,1}$ and $U_{b,2}$. If U_{new}, which is calculated according to Equation (15), is larger than $U_{b,1}$, the new flow can decrease the throughput of existing equal or higher priority flows. If U_{new} is larger than $U_{b,2}$, the new flow can not obtain its desired throughput by competing with existing flows. In both cases, the new flow is not admitted. Only when U_{new} is smaller than both $U_{b,1}$ and $U_{b,2}$, does admission succeed. If every node on the route of the new flow admits this flow, which shows that the new flow has enough end-to-end bandwidth, then the new flow can start.

C. Rate Policing

Because of the contention-based nature of IEEE 802.11, it is necessary to control the sending rate of best effort traffic so that it does not affect the QoS of existing realtime flows. To calculate the sending rate of best effort traffic at a Node V, it is necessary to identify the available bandwidth to best effort traffic at Node V, which is defined as the amount of bandwidth that best effort traffic can use without degrading the QoS of existing realtime flows. The available bandwidth to best effort traffic can be estimated using the same bandwidth prediction function introduced in the previous section. The only difference is that unlike realtime traffic, where a new realtime flow is not allowed to decrease the throughput of existing realtime flows, a new best effort flow is allowed to push existing best effort flows since these best effort flows can adapt to bandwidth and delay changes. Therefore, to guarantee that no realtime traffic is affected by best effort traffic, the upper bound on the amount of best effort traffic that Node V can impose on the network is:

$$U_{b,1} = C \left(1 - \sum_{i=\gamma}^{n} \frac{R_{x_i} L_{x_i}}{C} - \frac{1}{\eta^*_\gamma} \sum_{i=1}^{\gamma-1} \frac{L_{x_i}}{W_{x_i}} \right), \tag{21}$$

where Node x_γ is the first node that carries realtime traffic. Note that we do not need to calculate $U_{b,2}$ since we do not care what rate a best effort flow can achieve. By using a rate control mechanism, such as leaky bucket, Node V is able to control the amount of its best effort traffic $R_V L_V$ below $U_{b,1}$ and hence protects the throughput of realtime traffic.

VI. Evaluation

In this section, we evaluate the performance of MPARC using NS2 [6]. The evaluation focuses on MPARC's accuracy in admission control and rate policing and its ability to support multipriority-based admission control. The performance of MPARC is compared with other admission control protocols based on the free, delay and saturation models.

The first set of simulations demonstrates MPARC's ability to maintain the throughput of admitted realtime flows. Five randomly generated topologies are used, each is $1000m \times 1000m$ square with 50 randomly positioned nodes. The simulations run 100 seconds. TCP is used for best effort traffic and UDP is used for realtime traffic. Every five seconds for the first 50 seconds of the simulation, a new realtime CBR flow with 512 Byte packets and randomly selected rates between [10, 50] packets per second performs admission control. After the 50 seconds, every 5 seconds, a new best effort FTP flow starts. The sources and destinations of all flows are randomly selected. Due to the similarities of the simulations with different topologies, we only present the results from one representative simulation. Figure 9 shows the total violation of throughput guarantees, which is defined as the total throughput of all CBR flows minus the total desired rate of all CBR flows. The delay model starts to show throughput violations at 30 seconds, indicating that it admits too many realtime flows. At 55 seconds, the free model starts to show violations because it does not have the rate policing mechanism of best effort traffic to protect the throughput of realtime flows. However, MPARC and the saturation model can effectively keep the throughput guarantees to realtime flows. Figure 10 shows the total throughput of all the network flows. Before 50 seconds, the total throughput of saturation model is much less than the total throughput of all the other models, which means that it rejects more realtime flows than the other models. These unnecessary rejections reduce network utilization and limit the number of realtime flows that the network is able to carry. After 50 seconds, MPARC achieves comparable total throughput even though it has rate policing for best effort traffic, demonstrating that the rate policing in MPARC is efficient and does not penalize best effort traffic unnecessarily. These results demonstrates that MPARC maintains its guarantees to admitted realtime flows, does not reject realtime flows unnecessarily and achieves high network utilization. None of the other approaches achieves all of these three goals.

The second set of simulations demonstrates MPARC's ability to support admission control when there are multiple priorities of realtime flows. In the first simulation, 5 CBR realtime flows with increasing priority start consecutively. The rate of the flows are all 200 packets per second and the packet sizes are all 512Bytes. The rate of the flows are deliberately set larger than half of the network capacity so that no two flows can achieve their desired rates simultaneously. Figure 11 shows the violation of throughput guarantees to each admitted flow. As a higher priority flow arrives, if this flow can achieve its desired bandwidth by competing with existing flows, MPARC and the saturation model admit the flow even if the new flow may degrade the throughput of existing lower priority flows. The throughput of the highest priority flow is always maintained in MPARC and the saturation model. The delay model, however, admits all newly arrived flows even if the new flow cannot achieve its desired rate, resulting in the violation of throughput guarantees to every flow. Since the free model does not recognize priority, it only admits the first flow and rejects all later flows even if the later flows have higher priorities. The second simulation is the same as the first except that the priorities of the five CBR flows are decreasing. Since the first admitted flow has the highest priority, the later lower priority flows should be rejected to protect the throughput of the first flow. Figure 12 depicts the violation of throughput guarantees to admitted flows. Since the saturation model, free model and MPARC all only admit the first flow, they show no violation of throughput guarantees to the first admitted flow. The delay model, however, admits the first two flows and shows violation of throughput guarantees to both admitted flows. In conclusion, both MPARC and the saturation model can achieve priority based admission control. However, as shown in the first set of simulations, the saturation model may falsely reject realtime flows even if the network has enough bandwidth. Hence, among all the four protocols, MPARC is the only protocol that can achieve accurate priority-based admission control and rate policing.

VII. Conclusion

In this paper, we use our novel bandwidth allocation model to design a joint admission control and rate policing protocol, MPARC. Through simulation, we show that MPARC achieves accurate admission control of realtime traffic and rate policing of best effort traffic, which ensures that throughput guarantees for realtime flows are maintained and at the same time the network utilization is efficient. In the future, we plan to extend the bandwidth allocation model to express packet delays so that delay-based admission control can be used.

References

[1] Imad Aad and Claude Castelluccia. Differentiation mechanisms for IEEE 802.11. In *Proceedings of INFOCOM*, 2001.

[2] Albert Banchs, Xavier Perez-Costa, and Daji Qiao. Providing Throughput Guarantees in IEEE 802.11e Wireless LANs. In *Proceeding of the 18th International Teletraffic Congress(ITC-18)*, 2003.

[3] Michael G. Barry, Andrew T. Campbell, and Andras Veres. Distributed Control Algorithms for Service Differentiation in Wireless Packet Networks. In *Proceedings of Infocom*, 2001.

[4] Giuseppe Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3), 2000.

[5] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification. RFC 2205, September 1997.

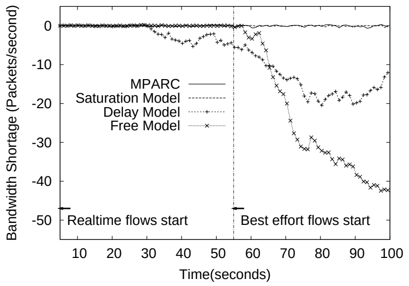


Fig. 9. Total Violation of Throughput Guarantee

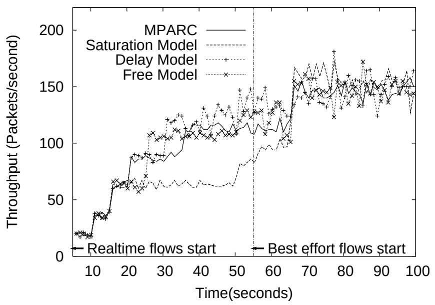
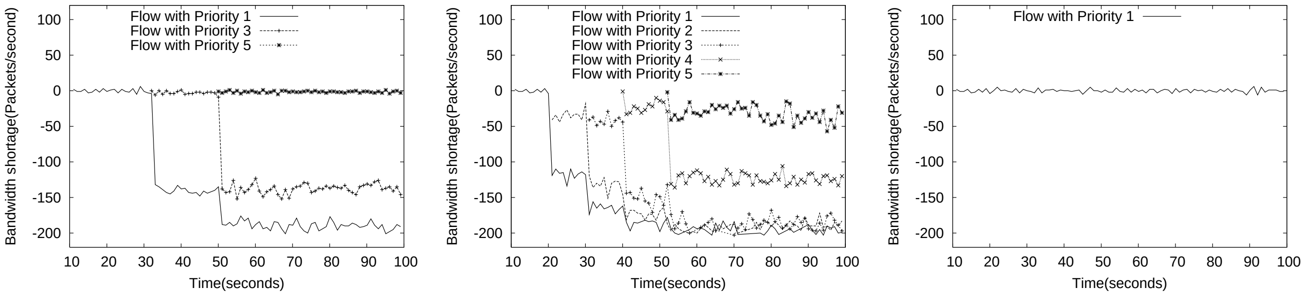
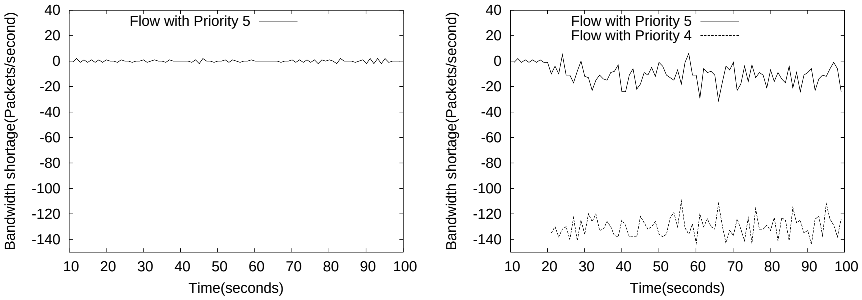


Fig. 10. Network Utilization



(a) MPARC and saturation model (b) Delay model (c) Free model

Fig. 11. Per Flow Violation of Throughput Guarantee (Increasing Priority Flows)



(a) MPARC, saturation and free Model (b) Delay model

Fig. 12. Per Flow Violation of Throughput Guarantee (Decreasing Priority Flows)

[6] Kevin Fall and Kannan Varadhan. NS notes and documentation. In *The VINT Project, UC Berkely, LBL, USC/ISI, and Xerox PARC*, 1997.

[7] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[8] Manthos Kazantzidis, Mario Gerla, and Sung-Ju Lee. Permissible Throughput Network Feedback for Adaptive Multimedia in AODV MANETs. In *IEEE International Conference of Communications (ICC)*, 2001.

[9] Seoung-Bum Lee, Gahng-Seop Ahn, Xiaowei Zhang, and Andrew Campbell. INSIGNIA: An IP-Based Quality of Service Framework for Mobile Ad Hoc Networks. *Journal of Parallel and Distributed computing, Special issue on Wireless and Mobile Computing and Communications*, 60:374–406, 2000.

[10] Stefan Mangold, Sunghyun Choi, Peter May, Ole Klein, Guido Hiertz, and Lothar Stibor. IEEE 802.11e Wireless LAN for Quality of Service. In *Proceedings of European Wireless*, 2002.

[11] P.Chatzimisios, V. Vitsas, and A. C. Boucouvalas. Throughput and Delay analysis of IEEE 802.11 protocol. In *Proceedings of the 5th IEEE International Workshop on Networked Appliances*, 2002.

[12] Charles Perkins. Ad-hoc on-demand distance vector routing. In *MILCOM '97 panel on Ad Hoc Networks*, 1997.

[13] Charles Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications*, 1994.

[14] Dennis Pong and Tim Moors. Call Admission Control for IEEE 802.11 Contention Access Mechanism. In *Procceedings of IEEE Globecom*, 2003.

[15] Samarth H. Shah, Kai Chen, and Klara Nahrstedt. Available Bandwidth Estimation in IEEE 802.11-based Wireless Networks. In *The 1st Bandwidth Estimation Workshop (BEst 2003)*, 2003.

[16] Samarth H. Shah, Kai Chen, and Klara Nahrstedt. Dynamic Bandwidth Management for Single-hop Ad Hoc Wireless Networks. In *Proceedings of IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2003.

[17] IEEE Computer Society. 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

[18] Yaling Yang, Jun Wang, and Robin Kravets. Achievable bandwidth prediction in multihop wireless networks. Technical Report UIUCDCS-R-2003-2367 and under submission to Mobicom 2004, December 2003.