A Priority-based Distributed Call Admission Protocol for Multi-hop Wireless Ad hoc Networks

Yuan Sun Elizabeth M. Belding-Royer
Department of Computer Science
University of California, Santa Barbara
{suny, ebelding}@cs.ucsb.edu

Xia Gao James Kempf
DoCoMo Communications Laboratories USA
{gao, kempf}@docomolabs-usa.com

Abstract

Ad hoc networks have been proposed for a variety of applications where support for real time, multimedia services may be necessary. This requires that the network is able to offer service differentiation and quality of service (QoS) appropriate for the latency and jitter bounds needed to meet the real time constraint. This paper describes a design for realistic QoS support using a system approach that involves co-ordinated changes at the MAC and IP layers. At the MAC layer, we propose a priority-based scheduling mechanism to provide service differentiation based on current channel status. We develop a priority-based delay model for the adaptive backoff scheme. The delay model allows each node to make local admission decisions. At the IP layer, the network resource availability distribution and flow admission in multi-hop ad hoc networks is achieved through a proposed call admission protocol, so that each node has the correct view of the shared channel usage, and the correct flow admission decision is made based on the estimated flow quality (accumulated delay of the path). Analytical and simulation results show that our approach can provide bounded latency and low jitter for real-time traffic, such as VoIP. The results also demonstrate that the aggregated network throughput is significantly improved given the quality requirements.

1 Introduction

Wireless networking and multimedia content are two rapidly emerging technological trends. Among types of wireless networks, multi-hop wireless ad hoc networks provide a flexible means of communication when there is little or no infrastructure, or the existing infrastructure is inconvenient or expensive to use. With the development of ad hoc networks, we can anticipate that multimedia applications will be popular in personal networks or other collaborative scenarios.

An important requirement for providing multimedia services in multi-hop ad hoc networks is that certain quality of service (QoS) metrics can be satisfied. There has been significant research on providing QoS in wired networks. For instance, Intserv [29] and Diffserv [13, 25] are two well-known approaches. These approaches rely on the availability of precise resource utilization information of wired links. However, because of the shared nature of wireless communication channels and node movement, these techniques cannot be directly applied to wireless networks. For infrastructured wireless networks, the base station can act as a central coordination point, thereby enabling the use of centralized quality of service approaches. For example, the base station can simply deny the admission request of a new flow if the traffic load in the network is already saturated. An approach like the IEEE 802.11 Point Coordination Function (PCF) can be used by the base station to give priority to delay sensitive traffic. In ad hoc networks, however, there is no centralized point that can provide resource coordination for the network; every node is responsible for its own traffic and is unaware of other traffic flows in the network. Furthermore, a flow must often traverse multiple hops to reach the destination; multiple nodes must coordinate to route traffic. Hence, an approach that provides QoS must support multi-hop communication.

Wireless networks generally have limited resources in terms of both device capabilities and available network bandwidth. Consequently, it is beneficial to have call admission to prevent unprovisioned traffic from being injected into the network beyond the saturation point. If a flow has rigid QoS requirements, an admission mechanism will prevent the waste of resources of both the source node itself and the whole network, if the network cannot support the flow. Furthermore, wireless communication channels are shared by all nodes within transmission range; consequently, all nodes within a transmission area contend for the limited channel bandwidth. In a multi-hop scenario, an admitted flow at a source node not only consumes the source's bandwidth, but the bandwidth of all the neighboring nodes along the data propagation path, thereby affecting ongoing flows of other nodes. Hence, it is essential to perform admission control along the entire path.

Service differentiation is another important aspect of providing QoS. In many ad hoc network applications, such as disaster rescue, communication terminals may have different priority ranks. For example, the messages sent by the commander should supersede traffic sent out by other rescue team members so that urgent information can be delivered. Many applications that are deployable in ad hoc networks, such as multimedia applications, may have different delivery requirements, i.e., low delay and jitter, and high throughput. For instance, a typical Voice over IP (VoIP) traffic session has the requirement of very low transmission delay. While multimedia streaming traffic is more tolerant to latency than VoIP traffic, it requires more bandwidth. We can therefore label

different traffic classes with different priority levels and provide service differentiation among traffic flows.

The essential problem of providing QoS in multi-hop ad hoc networks is trying to admit as many traffic flows as possible in order to achieve high efficiency of the channel usage, while at the same time providing service quality guarantees according to traffic priority. Recent studies [11, 22] indicate that advanced techniques such as directional antennas and multiple channels can significantly improve network efficiency. However, the capacity limit still remains when more flows try to join the network. In this paper we limit our study to single channel usage only. Specifically, we focus on IEEE 802.11 based wireless networks.

The contribution of this paper is three-fold. First we propose a priority-based scheduling mechanism to provide service differentiation based on current network status. Specifically, the collision rate is considered in the backoff scheme for different priority flows. Second, we present an analytical model for the adaptive backoff scheme and derive a priority-based delay model. Third, we propose an admission control protocol in multi-hop ad hoc networks so that each node has the correct view of its shared channel usage, and correct admission decisions are made based on the estimated quality (delay) of a flow calculated using the delay model.

The remainder of the paper is organized as follows. Section 2 first describes the operation of the IEEE 802.11 protocol, and then presents related work in the area of QoS provisioning in ad hoc networks. Section 3 presents our proposed adaptive priority scheduling scheme and a derived delay model. We then extend the scheme to multi-hop networks and explain how admission control is achieved in multi-hop ad hoc networks in section 4. The performance of our proposed approach is evaluated in section 5. Section 6 discusses our observations and finally section 7 concludes the paper.

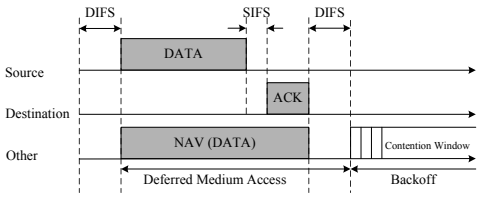
2 Background

2.1 IEEE 802.11

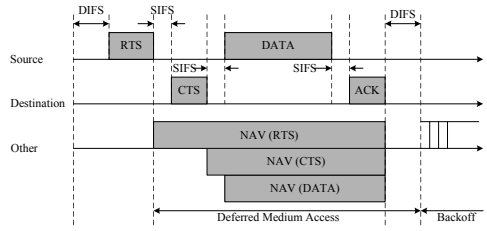
In this section, we briefly describe the operation of both the IEEE 802.11 standard and the 802.11e QoS extension. The standard includes specifications for medium access control (MAC) and physical layer (PHY) [31]. It supports two access mechanisms: Distributed Coordination Function (DCF), which uses a basic scheme of Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), and Point Coordination Function (PCF), which provides centralized control of medium usage through polling of the mobile stations by the access point. Because DCF is the only supported mode in ad hoc networks, we limit our investigation to DCF in this paper.

2.1.1 Distributed Coordination Function (DCF)

The operation for DCF in wireless networks is based on the use of CSMA/CA. A node with a packet ready for transmission waits until the channel is sensed idle for a specified time



(a) Basic IEEE 802.11 DCF Access Scheme.



(b) DCF Combined with RTS/CTS Access Scheme.

Figure 1: Operation of IEEE 802.11 Medium Access.

duration, called the Distributed Inter Frame Spacing (DIFS). If the channel is sensed busy, the node defers the transmission until it senses the channel to be idle for a period of DIFS. The time following the DIFS is slotted for efficiency, with the slot size equal to the time needed for any node to detect the transmission of a packet from any other station. A station attempts transmission only at the beginning of each slot time. The random backoff timer is uniformly chosen in a range of $[0, CW-1]$ slots, where CW is the contention window size. The backoff timer is decremented as long as the channel is sensed idle, stopped when a transmission is detected, and reactivated when the channel is sensed idle again for more than a DIFS duration. DCF uses a binary exponential backoff scheme, where CW is initially set to CW_{min} at the first transmission attempt, and is doubled, up to CW_{max}, upon each unsuccessful transmission. CW is reset to CW_{min} after each successful transmission. After the backoff time reaches zero, the station transmits. Upon a successful reception of the packet, the receiver transmits an ACK frame after a Short Inter Frame Spacing (SIFS) interval; SIFS is shorter than DIFS. The ACK enables the sending stations to detect a collision. If the transmitting station does not receive an ACK, a collision is presumed to have occurred and after an Extended Inter Frame Spacing (EIFS), the frame is retransmitted. The frame is dropped after a maximum number of unsuccessful retransmissions. Figure 1(a) depicts the operation and the timing of the DCF basic access mechanism.

To solve the hidden terminal problem, an optional Request To Send/Clear To Send (RTS/CTS) mechanism is adopted as shown in figure 1(b). After the channel is sensed idle, instead of transmitting the data frame immediately, the station transmits a 20 Byte RTS frame. Upon the reception of RTS, the receiver responds with a 14 Byte CTS frame after a SIFS interval. The source station can only transmit the data frame upon successful reception of a CTS packet.

2.1.2 Enhanced DCF (EDCF)

As an enhancement to the basic DCF function, the IEEE 802.11 Working Group is working on providing QoS support to the 802.11 MAC protocol, called 802.11e. Enhanced DCF (EDCF) [17] is introduced to provide service differentiation for channel access. The basic approach of EDCF includes two distinctions from DCF: (1) assignment of different CW_{min} values to different priority classes, resulting in high priority traffic with smaller CW_{min} values; (2) assignment of Arbitration IFS (AIFS), instead of DIFS, to different traffic classes, resulting in high priority classes with smaller AIFS values.

2.2 Related Work

The existing related work can be categorized into two groups: QoS routing for ad hoc networks, and MAC protocol enhancement to provide QoS.

Many routing schemes/frameworks have been proposed to provide QoS support for ad hoc networks [2, 9, 10, 21, 30]. Among them, INSIGNIA [21] uses an in-band signaling protocol for distribution of QoS information. The information is included in the IP headers of the data packets, and the available resources are calculated at each station the packet traverses so that a QoS decision can be made. SWAN [2] improves INSIGNIA by introducing an Additive Increase Multiplicative Decrease (AIMD)-based rate control algorithm. Specifically, Explicit Congestion Notification (ECN) is used to dynamically regulate admitted real-time sessions. Both [9] and [10] utilize a distance-vector protocol to collect end-to-end QoS information via either flooding or hop-by-hop propagation. Once collected, the receiver selects the path that can satisfy the QoS requirement. CEDAR [30] proposes a core-extraction distributed routing algorithm that maintains a self-organizing routing infrastructure, called the "core". The core nodes establish a route that satisfies the QoS constraints on behalf of other nodes. None of these approaches significantly diverge from QoS approaches for wired networks, and they do not significantly address the differences between wired and wireless networks.

Recently, there has been other work that proposes to improve the performance of MAC protocols and to provide service differentiation. Many of these approaches specifically target IEEE 802.11. For example, studies in [1, 8, 16, 20] propose to tune the contention windows sizes or the interframe spacing values to improve network throughput. Among these solutions, MFS [20] proposes estimation techniques for the current network status and each node determines an extra scheduling delay so as to improve the network utilization. Recent studies in [23, 24, 33] investigate the problem of achieving fairness at MAC layer. Studies in [1, 3, 19, 28, 34], on the other hand, propose priority-based scheduling to provide service differentiation. Most of these studies utilize different backoff mechanisms, different DIFS lengths, or different maximum frame lengths, based on the priority of the traffic/node. A different approach [35] proposes a scheme using two narrow-band busy tone signals to ensure medium access for high priority source stations. In addition to these studies, several analytical models are proposed in [3, 5, 20, 34]. Specifically, [5] models the IEEE 802.11 binary exponential backoff behavior of a tagged station as a discrete Markov chain model and captures all the MAC protocol details. Based on this work, MFS [20] proposes an analytical model including the consideration of EIFS when collision occurs. Veres et. al. [34] develop a delay model based on the channel utilization, and propose to tune the contention window size so that service differentiation can be provided.

Among the discussed solutions, our approach is most closely related to the work in [19], which uses piggybacked information on control and data packets to know neighbor nodes' head-of-line packets. This information allows nodes to determine their relative priority. Subsequently, priority-based scheduling can be achieved. The solution utilizes multi-hop coordination so that a next-hop node can increase a packet's relative priority in order to meet the delay guarantee, thereby achieving the quality requirement along a multi-hop path. Our work is similar to [19] in that we also utilize priority scheduling by varying the backoff behavior of different priority flows. We utilize multi-hop coordination along the data delivery path to accomplish a call setup. However, there are also significant differences between the approaches. First, our work uses a traffic-class based priority, and differentiation is based on per-flow traffic, while [19] provides relative priority on a per-packet basis. Second, our priority scheduling takes the current network status into consideration so that we can adapt to varying network conditions, while [19] uses static adjustment of the contention window. Third, our work does not rely on MAC protocol control packets to collect QoS information. Instead, we utilize the on-demand routing protocols to disseminate a node's load information to its neighbors. Piggybacking information on data and control packets on a per-frame basis, as recommended in [19], adds extra overhead, consequently reducing the goodput of the channel. For example, given a 120 Byte VoIP packet, the overhead will be 48 Bytes (20 Bytes for the RTS, plus 14 Bytes each for the CTS and ACK), and the extra overhead for piggybacking priority information is 24 Bytes according to the algorithm described in [19]. Furthermore, RTS/CTS is optional for the IEEE 802.11 MAC protocol, especially when small packet sizes are used (such as for a VoIP packet). Hence the approach will result in less channel efficiency. Finally, [19] does not provide an admission control mechanism, resulting in performance degradation as the traffic load increases.

3 Adaptive Priority Scheduling

In this section, we describe our proposed priority scheduling solution and derive an analytical model of the backoff operation, as well as a delay model with the priority scheduling.

3.1 Priority Based Scheduling

As stated in section 1, service differentiation is needed for different applications. The differentiation can be achieved by assigning different priorities to the traffic flows and scheduling packet transmission based on the priority of the associated traffic class. For instance, VoIP traffic can be given a higher priority so that it has a greater probability of obtaining channel access and subsequently meeting its end-to-end delay/jitter requirements. On the other hand, non-real-time traffic can be given lower priority. To enable this, each node can have a separate queue for each priority, and traffic within a node can compete based on the priority policy. Another way of supporting priorities is for different traffic classes to share the same priority queue, with the head-of-line packet being the packet with the highest priority. In our approach, following the IEEE 802.11e standard, we consider the former case, where each traffic class acts as a virtual station; each queue contends for medium access independently and has its own backoff policy.

In the context of 802.11e, service differentiation at the MAC layer can be achieved by different schemes [1]. Possibilities include scaling the contention window according to the priority of each flow, assigning different inter-frame spacings, and using different maximum frame sizes. Here we primarily focus on the adaptive backoff schemes because typically the frame sizes cannot be controlled by the MAC layer. Specifically, by assigning a different set of CW_{min} and CW_{max} values to different traffic classes, we can achieve an initial service differentiation. As stated in the basic DCF mechanism, the backoff function f of a flow with priority denoted as pri is decided by:

$$f_{pri} = Rand[0, 2^r CW[pri]_{min}] \times T_{slot} \quad 0 \le r \le m \quad (1)$$

where r and m denote the number of retransmissions and the maximum allowed number of retransmissions, respectively. However, predefined static CW_{min} and CW_{max} values may not achieve optimal performance given different real traffic composition. To achieve better service differentiation, one approach is to change the backoff rate, i.e., choose a different constant α for different priority as indicated in Eq. (2),

$$f_{pri} = Rand[0, \alpha^r CW_{min}] \times T_{slot} \quad 0 \le r \le m \quad (2)$$

For instance, a larger α for low priority traffic indicates a larger backoff range, resulting in a smaller chance of successful capture of the channel. This can improve the service differentiation; however, it also brings less stability to the whole system, as indicated in [1]. In addition, the faster backoff rate will result in channel waste since the channel is idle for a longer time while all the stations backoff, especially when all the traffic has low priority.

Hence, here we consider a constant α for all traffic. Specifically, we adopt binary exponential backoff, where α is equal to 2. However, better service differentiation as the network conditions change is accomplished by an adaptive backoff scheme, as we describe in next section.

3.2 Adaptive Backoff Scheme

As stated in the IEEE 802.11e standard, different traffic class priorities are assigned different CW values. Typically, these values are predefined and hence do not adapt to the network state. However, because the state of ad hoc networks can vary greatly due to mobility and channel interference, it is advantageous to adjust the values according to the current channel condition. Specifically, mechanisms for avoiding collisions can be considered. Given a high traffic load in the network, the number of retransmissions significantly affects the throughput and subsequently packet delivery latency [20]. Hence, it is beneficial to consider the collision rate in the backoff scheme.

To achieve service differentiation, as well as to adapt to the current network condition, we combine the collision rate with the backoff mechanism, and we have:

$$f_{pri} = Rand[0, (2^r + R_{col} \times pri) \times CW_{min}] \times T_{slot} \quad 0 \le r \le m \quad (3)$$

where, R_{col} denotes the collision rate between a station's two successful frame transmissions, and pri is a variable associated with the priority level of the traffic. By applying Eq. (3), traffic with different priority levels will have different backoff behavior when collisions occur. Specifically, after a collision occurs, low priority traffic will backoff for longer, and subsequently high priority traffic will have a better chance of accessing the channel. Additionally, after a successful transmission, the reset of the contention window also takes the collision possibility into account because the next packet transmission has a greater chance of suffering a collision, given a high collision rate in the past.

3.3 Analytical Model for Backoff Schemes

We now develop an analytical model for our priority based adaptive backoff scheme with consideration of the collision rate, as indicated in Eq. (3). Our assumptions are the same as in other previous work [20]: the channel attempt rate is exponentially distributed with average rate λ_c, and the collision rate, given in an empty slot, is constant and only relates to the current traffic load.

Let $A = \{a_1, a_2, ..., a_s\}$ be the set of flows with different priorities in the network, where s denotes the total number of priority classes supported by the system, and $\forall a_i \in A$, a_i is the number of flows of priority class i. Flows with the same priority level have the same average packet length F_i and the same average backoff window size $L(a_i)$. The current channel attempt rate, λ_c, can then be represented by

$$\lambda_c = \sum_{i=1}^{s} \sum_{j=1}^{a_i} \frac{1}{b_{i,j}} = \sum_{i=1}^{s} \frac{a_i}{\bar{b_i}} = \sum_{i=1}^{s} \frac{a_i}{L(a_i)} \quad (4)$$

where $b_{i,j}$ denotes the backoff window size of flow j with priority class i and $\bar{b_i} = E_j[b_{i,j}] = L(a_i)$.

Given the channel attempt rate λ_c, the competing traffic attempt rate of a node with priority i is

$$\lambda_i = \lambda_c - \frac{1}{L(a_i)} \tag{5}$$

The collision probability of the node, p_i, is

$$p_i = 1 - e^{-\lambda_i} \tag{6}$$

and for each priority backoff function $f_{a_i}(p_i)$, $CW_{next} = f_{a_i}(p_i, CW, i)$.

Let $m = \lceil \log_2 \frac{CW_{max}}{CW_{min}} \rceil$, then the probability that $CW = 2^j CW_{min}$ is

$$c_j = \begin{cases} c_0 \cdot p_i^j & 1 \le j \le m-1, \\ c_0 \cdot \sum_{k=m}^{\infty} p_i^k = \frac{c_0 p_i^m}{1-p_i} & j = m \end{cases} \tag{7}$$

where $\sum_{k=0}^{m} c_k = 1$, and we also have

$$c_0 \cdot (1 + p_i + p_i^2 + \cdots + \frac{p_i^m}{1-p_i}) = 1 \Longrightarrow c_0 = 1 - p_i \tag{8}$$

Then for any priority class i, the average backoff window size during collisions is

$$\begin{aligned} L(a_i|\text{backoff}) &= b_0 c_0 + b_1 c_1 + \cdots + b_m c_m \\ &= \sum_{j=0}^{m-1} \frac{CW_j - 1}{2} p_i^j (1-p_i) + \frac{CW_m - 1}{2} p_i^m \end{aligned} \tag{9}$$

For our proposed backoff scheme as indicated in Eq. (3), $CW_j = [2^j + p(t)\alpha_i]CW_{min} \simeq [2^j + p_i\alpha_i]CW_{min} (\forall j \in [0, m])$, and we have

$$\begin{aligned} L(a_i \mid \text{backoff}) &= \sum_{j=0}^{m-1} \frac{(2^j + p_i\alpha_i)CW_{min,i} - 1}{2} p_i^j (1-p_i) \\ &+ \frac{(2^m + p_i\alpha_i)CW_{min,i} - 1}{2} p_i^m \\ &= \sum_{j=0}^{m-1} \frac{2^j CW_{min,i} - 1}{2} p_i^j (1-p_i) + \frac{2^m CW_{min,i} - 1}{2} p_i^m \\ &+ [\sum_{j=0}^{m-1} \frac{\alpha_i}{2} p_i^{j+1}(1-p_i) + \frac{\alpha^i}{2} p_i^m] \cdot CW_{min,i} \\ &= \frac{CW_{min,i}}{2(1-2p_i)}[1 - p_i - p_i(2p_i)^m] + \frac{\alpha_i}{2} p_i \cdot CW_{min,i} - \frac{1}{2} \end{aligned} \tag{10}$$

where $CW_{min,i}$ is the minimum contention window size for priority i, and α_i can be a constant associated with the flow's priority class i. Hence, besides the different CW_{min} value, by adjusting α_i we can adjust the sensitivity of the difference of different priority classes with respect to the collision rate p_i. Furthermore, it is envisioned that more flexible $g(\alpha_i)$ could be used to provide stronger differentiation among the priority classes.

Continuing the $L(a_i)$ calculation, Eq. (10) gives the average backoff window size under the condition that the channel is sensed busy. IEEE 802.11 specifies that the node transmits immediately without backoff if the channel is sensed idle for the DIFS period. Let P_{free} denote the probability of free channel when the node attempts a transmission, and P_{busy} denote the probability of busy channel, we have

$$\begin{aligned} P_{free} &= \frac{\frac{1}{\lambda_i}}{\bar{F} + \frac{1}{\lambda_i}} \\ P_{busy} &= \frac{\bar{F}}{\bar{F} + \frac{1}{\lambda_i}} \end{aligned} \tag{11}$$

where $\bar{F}$ is the average packet transmission time. Then the average backoff window size $L(a_i)$ is

$$\begin{aligned} L(a_i) = P_{free}(1 - e^{-\lambda_i}) L(a_i \mid \text{backoff}) \\ + P_{busy} L(a_i \mid \text{backoff}) \end{aligned} \tag{12}$$

Hence, we have derived the expression of the current channel attempt rate λ_c as a function of average backoff window size $L(a_i)$ in Eq. (4), contending traffic rate λ_i for a node with priority i as a function of λ_c and $L(a_i)$ in Eq. (5), the expression of collision possibility p_i as a function of λ_i in Eq. (6), and finally, the expression of average backoff window size $L(a_i)$ as a function of the collision possibility p_i in Eq. (12).

For any given flow set $a_1, a_2, ..., a_s$, we can get a derived $\lambda_i^\star$, $p_i^\star$ and $L(a_i)^\star$ by using the same iteration algorithm as used in [20]. Specifically, as proved in [15] by Goodman et. al., the expected number of collisions in a binary backoff algorithm grows asymptotically with $O(\log M)$, where M is the number of active stations in the network. Hence, the initial backoff window size $L(a_i)^{(0)}$, in our scheme, is bounded by following

$$L(a_i)^{(0)} < CW_{min} \cdot 2^{K \cdot \log(\sum_{i=1}^{s} a_i)} \tag{13}$$

where $K(> 0)$ is some arbitrary constant.

Given $L(a_i)^{(0)}$, which represents the largest backoff window size, we can calculate $\lambda_i^{(0)}$ and $p_i^{(0)}$ using Eq. (5) and (6), respectively. Then, by applying Eq. (12), we can calculate $L(a_i)^{(1)}$. The iteration repeats until the difference of two consecutive iteration values satisfies $|L(a_i)^{(j+1)} - L(a_i)^{(j)}| < \epsilon$, where ϵ denotes some pre-defined small value. Here, we make a small adjustment when calculating $L(a_i)$. Instead of using Eq. (12) to calculate $L(a_i)^{(j+1)}$, we set this result to $L(a_i)^{(j)'}$, which represents the smallest backoff window size. $L(a_i)^{(j+1)}$ can then be obtained as the arithmetic mean of $L(a_i)^{(j)}$ and $L(a_i)^{(j)'}$. We find this can achieve a much faster convergence speed than the original iteration algorithm in [20], where the arithmetic mean is only used in the first iteration. The iterative algorithm always converges as proved in Theorem 1 in [20].

Figure 2 shows a comparison between the analytical model and simulation results for the throughput versus the number

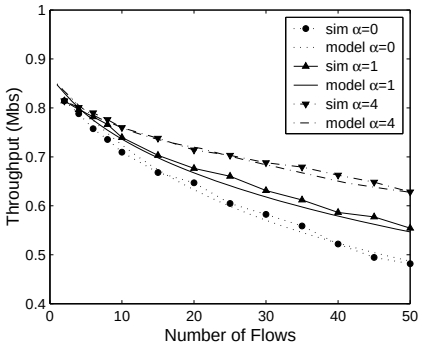


Figure 2: Comparison between Analytical and Simulation Results (CW_{min}=32).

of flows in a single broadcast region. CW_{min} value is set to 32 and the packet size is set to 500 bytes. The numerical result is calculated using Eq. (37) in [20] with inputs λ and p calculated with our derived model. We can see that the simulation results closely match the analysis, thereby verifying our model.

3.4 Delay Analysis of Adaptive Backoff Scheme

Given the current traffic rate λ_i, collision possibility p_i, and the average backoff window size $L(a_i)$, as calculated in section 3.3, we now derive the delay model for priority i.

Following the same analysis in [34], let $d_j(a_i)$ denote the total deferred time during the jth backoff for priority i. Because the backoff timer only decreases when the channel is idle, we have

$$d_j(a_i) = \begin{cases} \bar{F}' + k_j\bar{F} + b_j & j = 1, \\ k_j\bar{F} + b_j + \bar{F} & j > 1 \end{cases} \tag{14}$$

where b_j is the backoff time of the jth collision and k_j is a Poisson random variable with average of $\lambda_i \cdot b_j$ and denotes the number of packets that are sent during the jth collision. $\bar{F}$ is the average packet length of the traffic and

$$\bar{F} = \frac{\sum_{i=1}^{s}\sum_{j=1}^{a_i} F_{ij}}{\sum_{i=1}^{s} a_i} = \frac{\sum_{i=1}^{s} a_i \cdot F_i}{\sum_{i=1}^{s} a_i} \tag{15}$$

where F_i is the average frame size of flows for priority i. $\bar{F}'$ is the residual packet length that caused the collision on the first try and $\bar{F}' = \frac{\bar{F}}{2}$.

Hence, given the current attempt rate λ_i and the collision possibility p_i calculated using Eq. (5) and (6), the average value of the total accumulated deferred time for priority i, denoted as d_l, can be estimated as

$$\begin{aligned} d_l &= E[\sum d_j] \\ &= \sum_{l=0}^{\infty} E\sum_{j=0}^{l}[d_j|l\_\text{backoffs}](1-p_i)p_i^l \\ &= \sum_{l=1}^{\infty}(\sum_{j=1}^{l} E[(k_j+1)\bar{F} + b_j|l\_\text{backoffs}] \\ &\quad + E[(k_1+\frac{1}{2})\bar{F} + b_1|l\_\text{backoffs}])(1-p_i)p_i^l \\ &\quad + E[(k_1+\frac{1}{2})\bar{F} + b_1|l\_\text{backoffs}](1-p_i) \\ &= \sum_{l=0}^{\infty}\sum_{j=0}^{l} E[(\lambda\bar{F}+1)b_j + \bar{F}](1-p_i)p_i^l \\ &\quad - \frac{\bar{F}}{2}\sum_{l=0}^{\infty}(1-p_i)p_i^l \end{aligned} \tag{16}$$

For the basic DCF backoff scheme, where

$$E[b_j] = \frac{2^j \cdot CW_{min} - 1}{2} \tag{17}$$

we have

$$\begin{aligned} d_l(basic) = \frac{\lambda_i F + 1}{2} CW_{min}[&\frac{2^m p_i^{m+1}}{1-p_i} \\ &+ \frac{1 + p_i^m - (2^{m+1}+3)p_i^{m+1} - 2p_i^{m+2}}{1-2p_i}] \\ &+ [F - \frac{\lambda F + 1}{2}]\frac{1}{1-p_i} - \frac{\bar{F}}{2} \end{aligned} \tag{18}$$

For our proposed adaptive backoff scheme,

$$E[b_j] = \frac{(2^j + p_i\alpha_i)CW_{min} - 1}{2} \tag{19}$$

Hence, we have

$$\begin{aligned} d_l &= d_l(basic) + [\sum_{l=m+1}^{\infty}\sum_{j=0}^{m} p_i\alpha_i + \sum_{l=0}^{m}\sum_{j=0}^{l} p_i\alpha_i] \\ &\times (\frac{\lambda_i\bar{F}+1}{2})(1-p_i)p_i^l \\ &= d_l(basic) + p_i\alpha_i\frac{\lambda_i\bar{F}+1}{2}\frac{p_i(1-p_i^m)}{1-p_i}\cdot CW_{min} \end{aligned} \tag{20}$$

$\forall$ priority i,

$$\Delta = p_i\alpha_i\frac{\lambda_i\bar{F}+1}{2}\frac{p_i(1-p_i^m)}{1-p_i}\cdot CW_{min} \tag{21}$$

is the differentiation item. All other items in Eq. (20) are the same for all priorities. Hence, the difference of delay between different priority classes is linear with respect to α_i. By adjusting α_i, we can adjust the sensitivity of the difference with respect to the collision rate p_i.

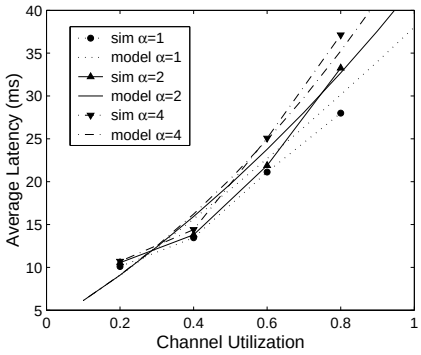


Figure 3: Comparison between Analytical and Simulation Results (CW_{min}=32).

Continuing the delay derivation, Eq. (20) gives the average of the total accumulated deferred time for a packet transmission when collisions occur. Similar to Eq. (12), the average deferred time is

$$d_{defer} = P_{free}(1 - e^{-\lambda_i})(\bar{F} + d_l) + P_{busy} d_l \qquad (22)$$

Let $d_{transmission}$ denote transmission time of the packet, then the average service delay is

$$d_{service} = d_{defer} + d_{transmission} \qquad (23)$$

Eq. (23) is the result for head-of-line packets. When queuing delay is considered, the total delay can be obtained by utilizing the delay results of an M/M/1 queue [4]. Specifically, suppose the traffic arrival rate is Poisson distributed with average rate of $R_{arrival}$, and the service rate is also Poisson distributed with average rate of $R_{service} = 1/\, d_{service}$. Then the total delay is

$$d = 1/\,(R_{service} - R_{arrival}) \qquad (24)$$

Eq. (24) is the result needed to check whether the delay bound for the priority flow is satisfied. Based on this, admission control can be achieved.

Figure 3 shows the comparison between the analytical model and the simulation results for the average packet service latency (excluding queuing delay) when the traffic load increases. Three α values were given to different priority traffic and each priority level has the same number of flows during the simulation. The packet size is set to 500 bytes and CW_{min} is set to 32. We can see that the simulation results and the numerical results are close to each other, thereby verifying our analysis: with different α values, different priority traffic experiences different latency, and the differentiation increases when the channel utilization increases.

4 Multi-hop Call Setup

In this section, we propose to utilize the route set up and maintenance process in ad hoc routing protocols to perform call admission and resource management. Many of the current routing protocols in ad hoc networks can be divided into two general categories: proactive and reactive routing protocols [27]. We consider the utilization of reactive routing protocols in this paper, in which routing activities are initiated in an "on demand" basis, and hence have the advantage of reduced routing load given low bandwith wireless links, as described in [6]. Specifically, during call setup, the source node disseminates the flow's priority information along with the *Route Discovery* process of the routing protocol. Each node on the path decides whether the flow can be admitted based on its local information, i.e., its active neighbors and their associated load. The goal of admission is to admit as many flows as the channel permits, while not causing significant performance degradation to ongoing high priority traffic. This can be accomplished using the analysis and prediction described in section 3.4. Section 4.1 elaborates the details of the protocol. Once a call finishes, the flow termination information propagates so that active stations that were affected by the flow become aware of the change and can subsequently make correct decisions for future call admissions. This propagation occurs along with the *Route Maintenance* process of reactive routing protocols. We describe the operation of this process in section 4.2.

4.1 Call Setup

When a new flow is issued, the call setup process determines whether the flow can be admitted with the needed service level while the requirements of current sessions are still satisfied. We apply the proposed mechanism and the analysis as described in sections 3.2 and 3.4 in a multi-hop ad hoc network and combine it with a reactive ad hoc routing protocol to provide call admission control.

Before a flow can be admitted, a *Route Discovery* process is needed to setup a route from the source to the destination. This is typically accomplished through multi-hop forwarding. Route discovery works by flooding the network with a route request (RREQ) packet. Upon reception of a RREQ, each node rebroadcasts it to its neighbors, unless it is the destination or has a route to the destination. Such a node replies to the RREQ with a route reply (RREP) packet. The RREP is propagated hop-by-hop back to the source node. Once received by the source, data packets can be routed to the destination. Details of two well-known on-demand routing protocols can be found in [12].

Admission control is needed during the call setup. This is because a node's transmission consumes not only its own channel resources, but the resources of all its neighbor nodes. Additionally, a flow typically traverses multiple hops to reach the destination. This affects all the flows in the neighborhood of the nodes on the transmission path. During the call admission, the impact information of the new flow, i.e., increased traffic load on the nodes, is collected along with route setup, so that each node has the correct view of its shared channel state. Each intermediate node uses this information to calculate an estimated local transmission delay. Finally, the source node uses the collected load information, as well as the delay information obtained through routing control messages, to make the admission decision.

Call Setup Request

Call setup is integrated with route discovery to find paths that can satisfy the QoS requirement. Specifically, through the request process, routes from the source node to the destination are obtained, and every node has the correct view of the traffic load in the shared channel.

In addition to the routing table locally stored at each station, each node also keeps a set of neighbors, called a *neighbor set*. The neighbor set maintains information about the node's neighbors, i.e., nodes that are within its transmission range. Here we assume bi-directional link connectivity. Each record in the neighbor set contains the neighbor node's address, as well as its load information, in terms of the current number of service flows and their respective priority level. Load information has an associated state, *confirmed* or *pending*, or *unknown*, indicating whether the load has been admitted or is in the process of call admission. An unknown state indicates an inactive neighbor of a node.

When a route request packet is broadcast for a new flow, the priority of the traffic class, as well as the required quality (delay threshold), is included in the RREQ packet. The accumulated delay through the traversed path is also included in the RREQ packet. Upon the reception of a RREQ from node a, node b first adds a pending record for node a with the requested flow information into its neighbor set. If the packet is not destined to b, node b decides whether to rebroadcast the RREQ. Node b first updates its own potential load in the neighbor set. Then, based on this updated set, it uses Eqs. (5), (6) and (10) in section 3.3 to calculate an estimation of the future traffic rate λ and collision rate p. By applying Eq. (23) in section 3.4, a predicted delay d' for the high priority flows can be calculated. The flow can only be admitted if $d' \leq D$, where D is a predefined delay threshold, and the sum of d' and the accumulated delay of previous hops included in the RREQ packet meets the required quality. If the flow is admissible, node b then rebroadcasts the RREQ message. It also sends a neighbor reply (NREP) packet back to node a, indicating its updated neighbor set information. The NREP packet contains a record of flow information for each neighbor node. Otherwise, if the delay criteria cannot be satisfied, node b drops the RREQ packet without any rebroadcast or reply. It also deletes the potential load cost of itself from its neighbor set. However, at this point, it still has the pending record for node a with the requested flow. The record will expire after some expiration time value (for instance $2 \times$ RREQ_TIMEOUT), if no further action is taken. If b is the destination, it replies to a with a RREP packet. In all cases, if node a was previously unknown to node b, b sends a copy of its neighbor set to a through a NREP packet.

The algorithmic description of this process is presented in figure 4. Note in all algorithmic descriptions, we do not include the actions of the routing protocol, such as routing table look up, insert, or update, nor do we include the data packet transmission.

Upon the reception of a NREP packet, a node updates the load information of its neighbor nodes, i.e., nodes that are in the intersection of the neighbor sets of itself and the received NREP. The reason for this intersection is that the neighbor set of node a's neighbors contains nodes that are more than two hops away from a. These nodes are not of interest to a when a is performing admission control. Note the change of neighbor set information and the transmission of an NREP message only occurs "on-demand", i.e., only when a new neighbor is discovered, or the load associated with the node changes.

Hence, during the journey of a call setup request along with a route request, each node along the propagation path makes a decision based on its current load condition, as well as the load of the nodes in its neighbor set. The nodes become aware of the load of neighbor nodes that will be affected by the new flow through the transmission of NREP packets.

Algorithm 4.1: RECV_RREQ(P_{rreq}, a, pri)

Input: P_{rreq}: the received RREQ packet.
Input: a: the sender address of the RREQ.
Input: pri: the priority of the requested flow.

// Check if a is a new node.
if $a \notin S$
  then do
    $insert(a, S, unknown)$;
    $flag = true$;
  enddo;

// Check if this RREQ was received before.
if P_{rreq} is duplicate **then goto STEP1;**

// Otherwise, fresh RREQ.
$update(a, pri, S, pending)$;
if I am the destination
  then $unicast\_RREP(a, pri, P_{rreq})$;
  else do
    $update(myself, pri, S, pending)$;
    $d' \leftarrow predict\_delay(S)$;
    if $d' \leq D_{threshold}$
      then do
        $rebroadcast\_RREQ(P_{rreq})$;
        $flag = true$;
      enddo;
      else do
        $delete(myself, pri, S)$;
        $drop(P_{rreq})$;
      enddo;
  enddo;

STEP1: if $flag$ **then** $unicast\_NREP(a, S)$;

Figure 4: The node process upon reception of a RREQ packet.

Call Setup Reply

As described above, when a destination node receives a RREQ destined to itself, it unicasts a RREP packet along the reverse path. Note, RREP generation by intermediate nodes,

Algorithm 4.2: RECV_RREP(P_{rrep}, a, pri)

Input: P_{rrep}: the received RREP packet.
Input: a: the address of the RREP sender.
Input: pri: the priority of the requested flow.

if $a \neq$ Destination
  then do
    if $a \notin S$
      then $insert(a, pri, S, pending)$;
      else $update(a, pri, S, pending)$;

// Determine the potential delay adding the requested flow.
$d' \leftarrow predict\_delay(S)$;
if $d' \leq D_{threshold}$
  then do
    if I am the source
    then $update(myself, pri, S, confirmed)$;
    else $unicast\_RREP(a, pri, P_{rrep})$;
  enddo
  else do
    $drop(P_{rrep})$;
    $delete(myself, pri, S)$;
  enddo

Figure 5: The node process upon reception of a RREP packet.

while utilized by many routing protocols [18, 26], is disabled here because the intermediate node may not have correct load information for the succeeding nodes along the path. Subsequently, it cannot make an admission decision. Upon the reception of the RREP packet, each intermediate node adds a record for the sender of the RREP (the previous hop) if there is no existing entry for that node in its neighbor set (the destination node should be excluded because it does not transmit packets for this session). Then, the node recalculates the delay d', as in the request phase, based on its updated neighbor set information. A node forwards the RREP along the reverse path as long as the criteria can be satisfied; otherwise, the RREP is dropped. The forwarding node also updates the accumulated delay in the forwarded RREP. Finally, the RREP reaches the source node. After a re-examination of its neighbor set, the source node decides whether to admit the flow, and it uses the path indicated in RREP if it is admissible. After a successful call setup, the source node updates its neighbor set and sets the pending state of itself, as well as the next hop node, as confirmed. The source node also deletes the pending record for this flow associated with the nodes that are not on the selected next hop path. Note that to make an admission decision, the source node compares the sum of the estimated delay to a total threshold value. If a source node receives multiple RREPs, it chooses the route that best meets the service requirements. An algorithmic description of the process of RREP reception is given in figure 5.

4.2 Resource Management

As described in the call setup request process, each node that rebroadcasts the RREQ creates a pending record for the to-be-admitted flow from the RREQ source. Because the RREP is unicast, nodes in a broadcast region that are not on the path do not know whether the flow is admitted. Subsequently the new load of the nodes along the path is unknown to other nodes in the broadcast region. Hence, there must be a mechanism to pass this information on to these off-path nodes. We achieve this objective by marking the very first packet of the flow. Each node along the communication path knows the flow is admitted through this marked packet, and as a result, updates its local load. It confirms the pending information of itself and its downstream neighbors. Then, each node along the path broadcasts a NREP packet to inform its neighbors of the change. Upon reception of the NREP, the nodes that are not on the delivery path of the flow update the load change for their neighbor nodes that are on the path and delete the pending record about this service load from the neighbor set. If the pending record expires and a node has not received a NREP message about the update, the node deletes the pending record. An algorithmic description of the process of data packets is given in figure 6.

Algorithm 4.3: RECV_DATA(P_{data}, a, pri)

Input: P_{rrep}: the received DATA packet.
Input: a: the last-hop address of the DATA.
Input: pri: the priority of the data traffic.

if P_{data} is marked as the first packet
  then do
    // Update its own load information.
    $update(myself, pri, S, confirmed)$;

    // Update the load information in its neighbor set S.
    $update(a, pri, S, confirmed)$;

    // Inform neighbors about the change.
    $broadcast\_NREP(S)$;
  enddo

Figure 6: The node process upon reception of a data packet.

When a flow ends, the load of all the nodes along the path is updated to reflect the released bandwidth. This information

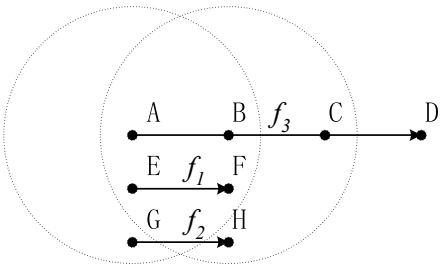


Figure 7: An example topology.

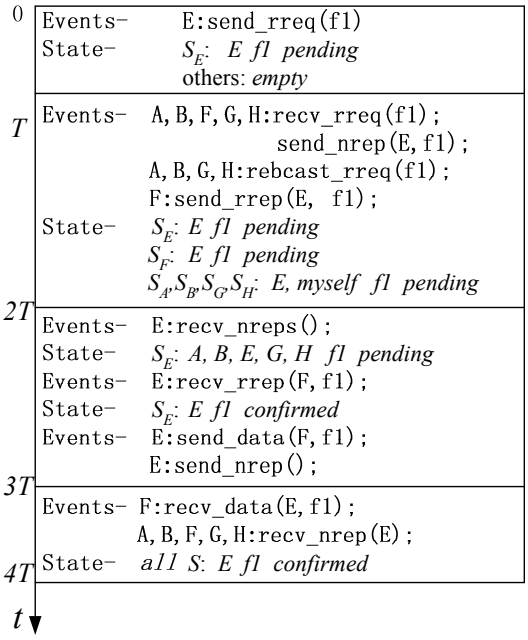


Figure 8: The events and states change of nodes during call setup of f_1.

should also be propagated into the node's broadcast region so that other nodes learn of the available resources. This process is performed through the *Route Maintenance* phase of reactive routing protocols. Specifically, a node detects the termination of a flow by either the route expiration, or by the enqueue rate measured for each priority queue at each node. After the detection, the node broadcasts a NREP packet to inform its neighbors of the change in its current load.

Consider an example scenario as shown in figure 7, where node A's neighbor set is $S_A = \{B, E, F, G\}$. Similarly, $S_B = \{A, C, E, F, H\}$, $S_E = \{A, B, F, G, H\}$, and $S_C = \{B, D\}$. There are 3 flows: $f_1 : E \rightarrow F$, $f_2 : G \rightarrow H$ and $f_3 : A \rightarrow D$. The starting time of the flows are $f_1 < f_2 < f_3$.

Figure 8 shows the communication events and neighbor set changes that occur during call admission when node E sends a request for flow f_1. Note we only show the neighbor set of a node if there is a change triggered by a packet reception, due to the space limitation.

4.3 Optimization

In this section, we discuss optimizations for the protocol by considering message loss and interference of nodes in the carrier sensing range.

Message Loss: The correct admission decision is based on an accurate view of a node's neighbor set. As described in section 4.1, the update of the neighbor set is triggered by a message reception. Message loss due to collisions and node movement can be frequent in wireless networks. If a RREQ packet from node a is not received at a neighbor node b, b will not update a's potential load. This is likely to impact b's future admission decisions. However, if a's load is confirmed, b will receive a NREP packet from a indicating the change. Node b can also learn of this change transitively from other neighbors in its broadcast region. Similarly, if a specific broadcast NREP packet is lost, the affected node will learn of the change of load through its neighbors unicast NREP packet exchanges. The sender of a unicast NREP and RREP packet will receive an ACK message from the next hop, thereby enabling the sender to detect packet loss. To improve the robustness of the protocol to message loss, we can introduce periodic hello message exchanges between neighbors, as utilized in many proactive routing protocols. The neighbor set information can be included in the hello messages so that each node has an updated view of its neighbor information.

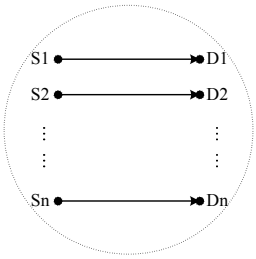


Figure 9: Network topology for the first simulation set.

Interference from Carrier Sensing Range: Our described protocol does not explicitly consider the interference from nodes within the carrier sensing range but outside of transmission range. Because the measured collision rate used in our delay analysis already takes the interference of carrier sensing into consideration, the problem is mitigated. However, to improve the accuracy of neighbor information, we can utilize current power control techniques so that control packets are transmitted at a higher power. This enables all neighbors within the carrier sensing range to be reached.

5 Performance Evaluation

The performance of our adaptive priority based scheduling algorithm, as well as the call setup protocol, is evaluated in the following simulations. Our approach is implemented in the NS-2 [14] simulator with the Monarch mobility extensions [6].

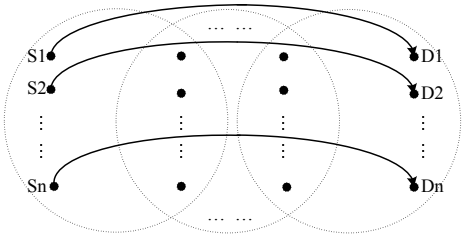
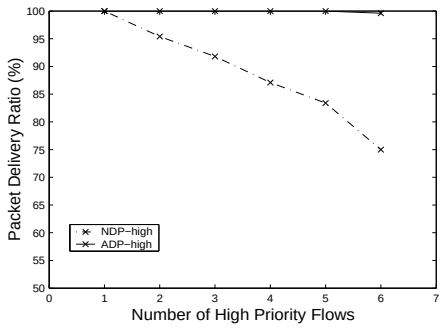
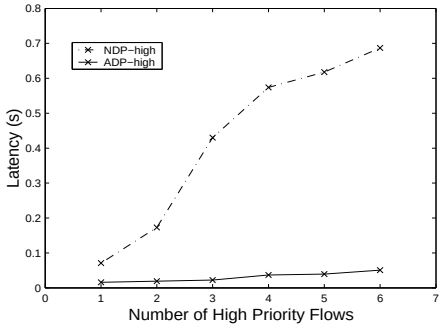


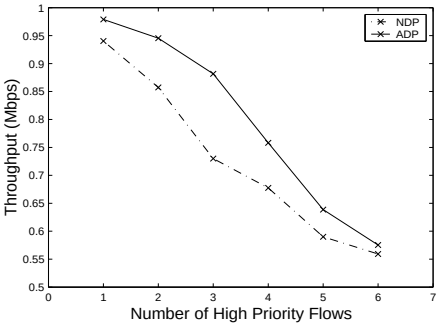
Figure 10: Network topology for the second simulation set.



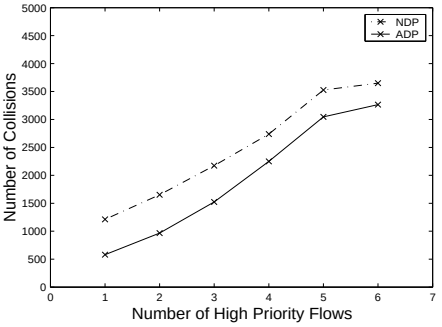
(a) Packet Delivery Ratio of High Priority Flows.



(b) Packet Delivery Latency of High Priority Flows.



(c) Aggregated Data Throughput of the Network.



(d) Total Collisions of the Network.

Figure 11: Performance Comparison between Adaptive Backoff and Non-adaptive Backoff without RTS/CTS.

5.1 Experimental Setup

The first set of simulations explores the performance of our proposed adaptive backoff algorithm described in section 3. The simulated network topology is shown in figure 9. A group of node pairs, S_i, D_i are all within the same broadcast region. The link bit rate is 1 Mbps. The traffic flows are of three different priority classes; the parameters for the classes are shown in table 1. The background traffic includes three flows each of medium and low priority, where the source and destination pairs are randomly chosen. We vary the number of flows with high priority and evaluate the performance of our adaptive backoff scheme. Specifically, the performance metrics for evaluating the backoff algorithm include:

- **Packet delivery fraction**: The number of data packets received by the destination compared with the number of data packets generated by the source for each priority class.
- **End-to-end packet delivery latency**: The average delivery delay of the data packets from the source to the destination.

Table 1: Priority Traffic Parameters

Priority Class	Packet Size (bytes)	Data Rate (Kbps)
High (G.711 VoIP)	160	64
Medium	1000	320
Low	500	80

- **Aggregated throughput**: The sum of the throughput for active flows in the network, including flows of different priority classes.
- **Number of collisions**: The total number of collisions that occur in the network during the simulation.

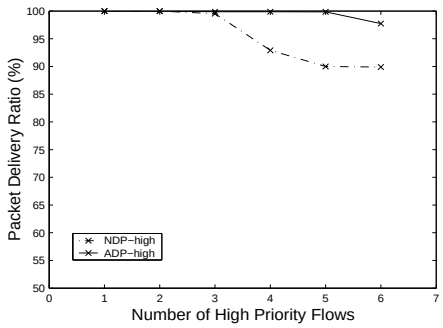
The second set of simulations studies the performance impact of our multi-hop call admission control. The simulation topology is shown in figure 10; all the flows need to traverse an average of 2 hops from S_i to D_i. The background traffic includes 2 flows each of medium and low priority. We increase the number of high priority flows by one, at each 10 second interval. In addition to the performance metrics described for the first set of experiments, we also consider the following metric:

- **Control overhead**: The number of control packets transmitted during the call setup.

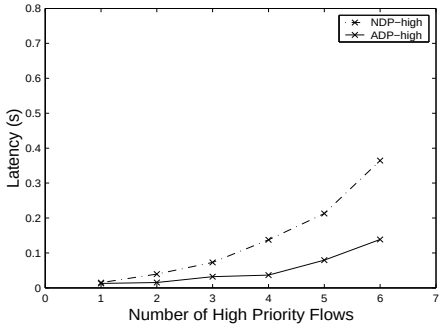
5.2 Results

5.2.1 Adaptive Backoff Scheme

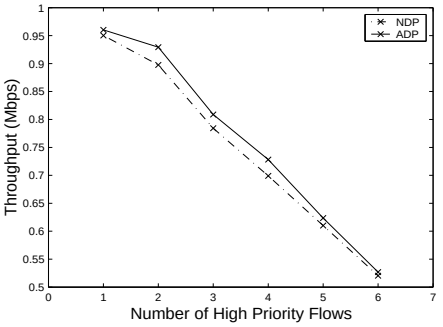
Figure 11 shows the effect of using our adaptive backoff mechanism (denoted as ADP) with and without RTS/CTS control packets. For comparison, we also show the results of a non-adaptive scheme (denoted as NDP), which does not take the collision rate into consideration and only varies the CW_{min} values of different priorities. Figures 11(a) and (b)



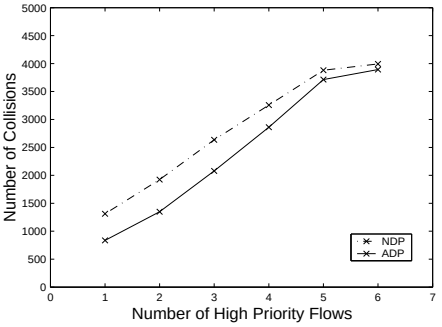
(a) Packet Delivery Ratio of High Priority Flows.



(b) Packet Delivery Latency of High Priority Flows.



(c) Aggregated Data Throughput of the Network.



(d) Total Collisions of the Network.

Figure 12: Performance Comparison between Adaptive Backoff and Non-adaptive Backoff with RTS/CTS.

show that adaptive backoff provides better service differentiation than the NDP scheme in that the packet delivery ratio of high priority flows does not suffer a significant decrease when the traffic load increases, and the end-to-end delivery latency of high priority flows is kept very low so that the service requirements can be met. Further, the aggregated throughput of the network using our scheme outperforms the NDP scheme as shown in figure 11(c). By including the collision rate in the next CW calculation, we reduce the possibility of future collisions, as shown in figure 11(d), thereby improving the throughput.

Figure 12 shows the simulation results of the same setup with the RTS/CTS mechanism enabled. Again, the adaptive backoff algorithm achieves better service differentiation, as well as reduces the number of collisions in the network.

We also observe that the performance gained by using the adaptive backoff scheme is more significant when RTS/CTS is not employed. This is because by using RTS/CTS, most collisions occur with the small RTS/CTS packets, not the data packets. Hence the stations detect the collisions more quickly and enter the backoff state sooner. This results in a smaller average end-to-end delay for the non-adaptive backoff, as shown in figure 12(b), than without using RTS/CTS, as shown in figure 11(b). On the other hand, for the adaptive backoff scheme, the delay without RTS/CTS is smaller, because it does not include the extra latency of the RTS/CTS packet exchange. Furthermore, with heavy traffic load, the number of collisions in the network when RTS/CTS is employed (shown in figure 12(d)) is higher than without RTS/CTS (figure 11(d)). As the nodes enter the backoff state earlier, there is less delay before they attempt the next transmission. This causes an increase in collisions. The aggregated network throughput of the adaptive scheme with RTS/CTS enabled is less than without RTS/CTS, as shown in the solid lines in figures 12(c) and 11(c), due to the RTS/CTS overhead. On the other hand, the aggregated network throughput of the non-adaptive scheme with RTS/CTS enabled is larger than without RTS/CTS, because most collisions occur among RTS/CTS packets, instead of data packets, thereby improving throughput.

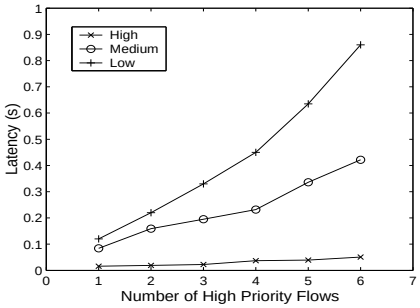
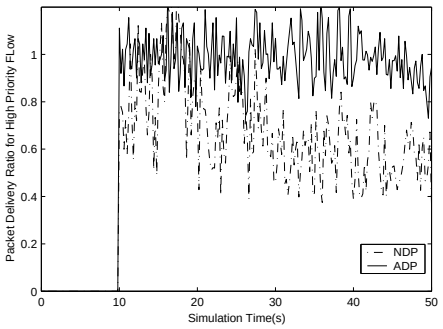
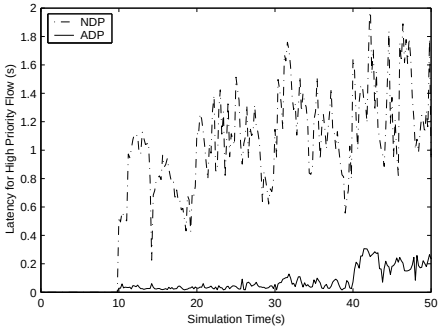


Figure 13: Delay Differentiation between Priority Flows.

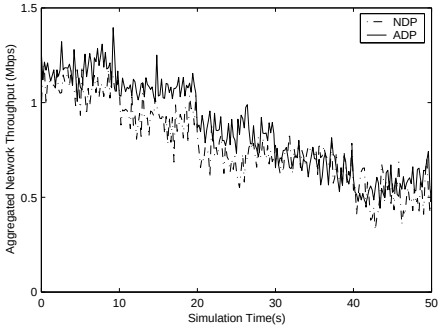
Figure 13 shows the average latency of each priority flow level as traffic increases when RTS/CTS is not applied. The



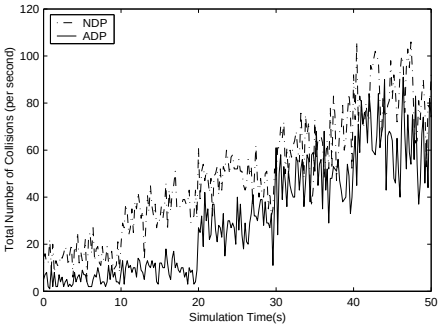
(a) Packet Delivery Ratio For High Priority Flows.



(b) Packet Delivery Latency of High Priority Flows.



(c) Aggregated Data Throughput of the Network.



(d) Total Collisions of the Network.

Figure 14: Performance Comparison between Adaptive Backoff and Non-adaptive Backoff without RTS/CTS.

figure indicates that service differentiation is achieved by our adaptive backoff scheme.

In summary, having an adaptive backoff mechanism has several advantages. First, it can adapt to network congestion (collisions) by increasing the backoff time to reduce the possibility of collision, thereby improving the aggregate network throughput. Further, it provides better service differentiation, resulting in a greater chance of high priority traffic meeting its real-time constraints.

5.2.2 Multi-hop Call Admission

Figure 14 shows the protocol performance in a multi-hop scenario when RTS/CTS is not employed. The solid lines represent the results for our adaptive backoff scheme. The non-adaptive backoff scheme is shown in dotted lines for comparison. Given our traffic model and network topology, our admission control process can admit up to four multi-hop high priority flows. For the admitted high priority flows, the average packet delivery ratio is higher than 90% and average latency is less than 200ms, as shown in figures 14(a) and 14(b). Here we use 200ms for the latency threshold. In figure 14(a), the packet delivery ratio above one indicates that the number of received packets exceeds the number of sent packets in the measured interval, due to the transmission latency. We also observe that the variation of transmission delay of high flows is comparatively small, indicating low inter-transmission delay, and thereby reducing transmission jitter of the flows. While not shown, when the fifth high priority flow starts, the estimated delay at the source node reaches 439ms, which is above the threshold. Hence this request is rejected. On the other hand, if the non-adaptive backoff scheme is used, significant performance degradation for high priority flows can be seen due to the lack of sufficient service differentiation. Figures 14(c) and 14(d) show the aggregated network throughput and the total number of collisions in the network. The aggregated throughput decreases as the number of high priority flows increases. This is because the high priority flows always supersede the other flows, obtaining a greater chance of channel access. Since they use a small packet size, the channel utilization decreases accordingly. Figure 14(d) shows that as the number of flows increases, the number of collisions also increases. We observe that most collisions occur among high priority flows when the traffic load is heavy. This is because high priority flows have a smaller CW value, as well as a smaller α value. Hence, their average backoff time is much smaller than other flows when collisions occur. In addition, we use static α_{pri} values in our simulation. The usage of a more flexible $g(\alpha_{pri})$ is envisioned to reduce the collisions further. In all cases, our adaptive scheme outperforms the non-adaptive backoff mechanism.

Figure 15 shows the average latency of high priority flows when call admission is not enabled. We only show the interval after the fifth flow (which is rejected by our call ad-

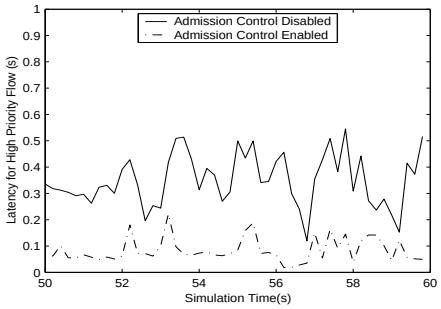


Figure 15: Average latency of High Priority Flows.

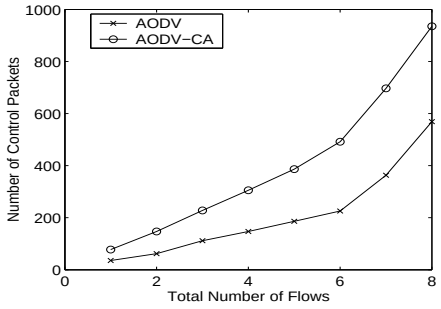


Figure 16: Control Overhead.

mission protocol) started. For comparison, we also show the results when call admission is enabled. As shown in figure 15, the average latency of all high priority flows increases significantly, and is above the 200ms threshold once the fifth flow is admitted. This indicates that our admission control protocol functions correctly to provide ensured quality of service to existing flows.

To evaluate the additional overhead of call admission, we integrate our call admission protocol into the AODV routing protocol [26]. We compare the number of control packets during call setup of our protocol and of unmodified AODV routing protocol. Figure 16 shows the number of control packets when the number of flows increases. Compared to unmodified AODV, our routing protocol with call admission included (noted as AODV-CA) has nearly double the control overhead. This is due to the NREP transmissions.

In summary, through the usage of multi-hop call admission, the service quality of existing high priority flows is maintained when new flows are requested. At the same time, by using the adaptive backoff scheme, the aggregated network throughput is increased so that as many flows as possible are admitted, while service differentiation is still provided.

In all simulations, we do not introduce node mobility. In a highly mobile ad hoc network, it is difficult to maintain service quality due to broken paths. Further, neighbor nodes change frequently due to mobility, resulting in inaccurate information of a node's neighbors. This will also result in a significant increase of control overhead. Evaluation of the performance of our scheme in mobile networks is future work.

6 Optimization and Discussion

Our model provides a statistically "soft" quality assurance, where the average quality of a class of traffic flows is guaranteed. Other schemes, such as [19], aim to provide hard guarantees. The techniques in [19] can further be applied to our approach to provide a more fine-grained quality guarantee.

A recent study [11] on the usage of directional antenna and multiple channels shows that these advanced techniques can improve the space division multiple access (SDMA) efficiency. We foresee that by combining these techniques with multipath routing, we can achieve significant gains on system performance. Specifically, by utilizing disjoint paths between the source and destination, data packets can traverse multiple routes, thereby improving the network access efficiency. On the other hand, the improvement of channel usage is at the expense of potentially increased packet delivery jitter and out-of-order of packets. Hence, this may not be directly applicable to real-time traffic such as VoIP. However, the problem can be alleviated through the application of techniques, such as digital fountain [7], that deliver content over multiple unreliable transport channels to provide support for multimedia systems. Another possible improvement for better service differentiation is to apply TCP Friendly Rate Control (TFRC) to low priority traffic. Specifically, when the network load increases and the traffic cannot reach its expected data rate, the feedback mechanism can notify the source node to adjust its sending rate, so that it can better adapt to the current channel state.

Our admission control mechanism suffers from security vulnerabilities. Security is a general problem in study in ad hoc networks, in that selfish/malicious nodes can send falsified information about the route, or the current traffic load. Many security enhancements to current routing protocols are likely to prove beneficial to our work. Specifically, the ADMIX technique [32], which conceals the true destination of packets from intermediate nodes along the path, is a likely candidate to force a node to participate or risk dropping packets destined for itself, thereby facilitating anonymization and secure communication between nodes.

7 Conclusion

This paper proposes an adaptive priority-based scheduling mechanism to provide better service differentiation. An analytical model of the mechanism is given, based on which we derive a delay model to predict average traffic latency given the current network load. Multi-hop coordination for admission control, integrated with reactive routing protocols, was studied. Specifically, during a call setup, each node along the propagation path estimates delay for the traffic using the derived delay model and uses this information to make an admission decision. Analytical and simulation results show that our approach provides service differentiation and quality of service support through the adaptive scheduling scheme and the admission control process. This is beneficial to the de-

ployment of multi-hop ad hoc networks where a variety of applications, such as multimedia and VoIP, will be utilized, and the admission of as many fbws as possible is desired as long as the needed service requirements are still met.

References

[1] I. Ada and C. Castelluccia. Differentiation Mechanisms for IEEE 802.11. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, Anchorage, Alaska, April 2001.

[2] G.-S. Ahn, A. Campbell, A. Veres, and L.-H. Sun. Supporting Service Differentiation for Real-Time and Best-Effort Traffi c in Stateless Wireless Ad Hoc Networks (SWAN). *IEEE Transactions on Mobile Computing*, 1(3):192–207, July-Septemeber 2002.

[3] A. Banchs, X. Perez-Costa, and D. Qiao. Providing Throughput Guarantees in IEEE 802.11e Wireless LANs. In *Proceedings of the 18^{th} International Teletraffic Congress (ITC-18)*, Berlin, Germany, September 2003.

[4] O. Bertsekas and R. Gallager. . In *Data Networks, 2nd Edition*. Prentice -Hall, 1992.

[5] G. Bianchi. Performance Analysis of the IEEE 802.11 Distributed Coordination Function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, March 2000.

[6] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A Performance Comparison of Multihop Wireless Ad Hoc Network Routing Protocols. In *Proceedings of the 4^{th} ACM/IEEE International Conference on Mobile Computing and Networking (MobiCOM'98)*, pages 85–97, Dallas, TX, October 1998.

[7] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege. Digital Fountain Approach to Reliable Distribution of Bulk Data. In *ACM SIGCOMM Conference on Communications Architectures, Protocols and Applications*, pages 56–67, Vancouver, September 1998.

[8] F. Calí, M. Conti, and E. Gregori. Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit. *IEEE/ACM Transactions on Networking*, 8(6), December 2000.

[9] S. Chen and K. Nahrstedt. Distributed Quality-of-Service Routing in Ad-Hoc Networks. *IEEE Journal of Selected Areas in Communications*, 17(8), August 1999.

[10] T. Chen, M. Gerla, and J. Tsai. QoS Routing Performance in a Multihop, Wireless Network. In *Proceedings of the IEEE ICUPC'97*, 1997.

[11] R. R. Choudhury, X. Yang, R. Ramanathan, and N. Vaidya. Using Directional Antennas for Medium Access Control in Ad Hoc Networks. In *Proceedings The Eighth Annual International Conference on Mobile Computing and Networking (MobiCOM'02)*, Atlanta, GA, Septemper 2002.

[12] S. R. Das, C. E. Perkins, and E. M. Royer. Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, pages 3–12, Tel-Aviv, Isarel, March 2000.

[13] D.Grossman. New Terminology and Clarifi cations for Diffserv. Request For Comments (Draft Stardard) 3260, Internet Engineering Task Force, April 2002.

[14] K. Fall and K. Varadhan. ns Manual. *http://www.isi.edu/nsnam/ns/doc/*. The VINT Project.

[15] J. Goodman and A. Greenberg. Stability of Binary Exponential Backoff. *Journal of the ACM*, 35(3), March 1998.

[16] T. S. Ho and K. C. Chen. Performance Evaluation and Enhancement of CSMA/CA MAC Protocol for 802.11 Wireless LANs. In *Proceedings of the IEEE PIMRC*, October 1996.

[17] IEEE. 802.11e Draft 3.1, May 2002.

[18] D. B. Johnson and D. A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In T. Imielinski and H. Korth, editors, *Mobile Computing*, pages 153–181. Kluwer Academic Publishers, 1996.

[19] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly. Distributed Multi-Hop Scheduling and Medium Access with Delay and Throughput Constraints. In *Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking (MobiCOM'01)*, Rome, Italy, July 2001.

[20] H. Kim and J. C. Hou. Improving Protocol Capacity with Model-based Frame Scheduling in IEEE 802.11-operated WLANs. In *Proceedings of the Ninth Annual International Conference on Mobile Computing and Networking (MobiCOM'03)*, pages 190–204, San Diego, CA, September 2003.

[21] S. 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, Special issue on Wireless and Mobile Computing and Communications*, 60:374–406, 2000.

[22] J. Li, Z. Haas, M. Sheng, and Y. Chen. Performance Evaluation of Modifi ed IEEE 802.11 MAC for Multi-channel Multi-hop Ad Hoc Networks. *Journal of Interconnection Networks*, 4(3), 2003.

[23] H. Luo, S. Lu, and V. Bharghavan. A New Model for Packet Scheduling in Multihop Wireless Networks. In *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCOM'00)*, Boston, MA, August 2000.

[24] T. Nandagopal, T. Kim, X. Gao, and V. Bharghavan. Achieving MAC Layer Fairness in Wireless Packet Networks. In *Proceedings of the Sisth Annual International Conference on Mobile Computing and Networking (MobiCOM'00)*, Boston, MA, August 2000.

[25] K. Nichols and B. Carpenter. Defi nition of Differentiated Services Per Domain Behaviors and Rules for Their Specifi cation. Request For Comments (Draft Stardard) 3086, Internet Engineering Task Force, April 2001.

[26] C. E. Perkins and E. M. Royer. Ad-hoc On-Demand Distance Vector Routing. In *Proceedings of the 2^{nd} IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, New Orleans, LA, February 1999.

[27] E. M. Royer and C.-K. Toh. A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks. *IEEE Personal Communications Magazine*, 6(2):46–55, April 1999.

[28] R. Rozovsky and P. Kumar. "SEEDEX: A MAC Protocol for Ad Hoc Networks. In *Proceedings of the 2^{nd} ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'01)*, Long Beach, CA, October 2001.

[29] S. Shenker, C.Partridge, and R. Guerin. Specifi cation of Guaranteed Quality of Service. Request For Comments (Draft Stardard) 2212, Internet Engineering Task Force, September 1997.

[30] P. Sinha, R. Sivakumar, and V. Bharghavan. CEDAR: A Core-Extraction Distributed Ad Hoc Routing Algorithm. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, pages 202–209, 1999.

[31] I. C. Society. IEEE Standard for Wireless LAN-Medium Access Control and Physical Layer Specifi cation, Novermber 1999.

[32] S. Sundaramurthy and E. M. Belding-Royer. The AD-MIX Protocol for Encouraging Participation in Mobile Ad hoc Networks. In *Proceedings of the International Conference on Network Protocols (ICNP)*, Atlanta, GA, November 2003.

[33] N. Vaidya, P. Bahl, and S. Gupta. Distributed Fair Scheduling in a Wireless LAN. In *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCOM'00)*, Boston, MA, August 2000.

[34] A. Veres, A. T. Campbell, M. Barry, and L.-H. Sun. Supporting Service Differentiation in Wireless Packet Networks Using Distributed Control. *IEEE Journal of Selected Areas in Communications*, 19(10), October 2001.

[35] X. Yang and N. Vaidya. Priority Scheduling in Wireless Ad Hoc Networks. In *Proceedings of the 3^{rd} ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02)*, Lausanne, Switzerland, June 2002.