Distributed Multi-Hop Scheduling and Medium Access with Delay and Throughput Constraints

V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly

Department of Electrical and Computer Engineering
Rice University

{kanodia,chengzhi,ashu,bahar,knightly}@ece.rice.edu

ABSTRACT

Providing quality of service in random access multi-hop wireless networks requires support from both medium access and packet scheduling algorithms. However, due to the distributed nature of ad hoc networks, nodes may not be able to determine the next packet that would be transmitted in a (hypothetical) centralized and ideal dynamic priority scheduler. In this paper, we develop two mechanisms for QoS communication in multi-hop wireless networks. First, we devise *distributed priority scheduling*, a technique that piggybacks the priority tag of a node's head-of-line packet onto handshake and data packets; *e.g.*, RTS/DATA packets in IEEE 802.11. By monitoring transmitted packets, each node maintains a scheduling table which is used to assess the node's priority level relative to other nodes. We then incorporate this scheduling table into existing IEEE 802.11 priority back-off schemes to approximate the idealized schedule. Second, we observe that congestion, link errors, and the random nature of medium access prohibit an exact realization of the ideal schedule. Consequently, we devise a scheduling scheme termed *multi-hop coordination* so that downstream nodes can increase a packet's relative priority to make up for excessive delays incurred upstream. We next develop a simple analytical model to quantitatively explore these two mechanisms. In the former case, we study the impact of the probability of overhearing another packet's priority index on the scheme's ability to achieve the ideal schedule. In the latter case, we explore the role of multi-hop coordination in increasing the probability that a packet satisfies its end-to-end QoS target. Finally, we perform a set of ns-2 simulations to study the scheme's performance under more realistic conditions.

1. INTRODUCTION

Supporting real-time flows with delay and throughput constraints is an important challenge for future wireless networks. Indeed, providing differentiated quality-of-service levels increases a system's total utility when applications have diverse performance requirements, e.g., some preferring low delay, others high throughput, and others merely best effort service [17]. Consequently, both medium access control and network-layer scheduling algorithms must select and transmit packets in accordance with their QoS requirements.

In wireless networks with base stations, the base station acts as a centralization point for arbitration of such QoS demands. For example, suppose the goal is to support delay-sensitive traffic using the Earliest Deadline First (EDF) service discipline. In this case, each packet has a priority index given by its arrival time plus its delay bound. Consequently, the base station can simply select the packet with the smallest priority index for transmission on the down-link, subject to its channel being sufficiently error-free. In this way, an "ideal" EDF schedule could be approximated to the largest extent possible allowed by the error-prone wireless link.

However, in networks without base stations, there is no centralized controller which can assess the relative priorities of packets contending for the medium. Consequently, the node actually possessing the highest priority packet is unaware that this is the case; nor are other nodes with lower priority packets aware that they should defer access. Moreover, in *multi-hop* (or ad hoc) networks in which packets are forwarded across multiple broadcast regions, it becomes increasingly challenging to satisfy a flow's end-to-end QoS target.

In this paper, we introduce a new framework for dynamic priority packet transmission in multi-hop wireless networks. Our key insight is that the broadcast nature of the wireless medium together with the store-and-forward nature of multi-hop networks provide opportunities to communicate and coordinate priority information among nodes. Our goal is to exploit these system attributes and develop integrated medium access and scheduling algorithms that satisfy a high fraction of QoS targets using fully distributed mechanisms.

Our contribution is two fold. First, within a broadcast region, we devise a mechanism termed *distributed priority scheduling* in which each node locally constructs a scheduling table based on overheard information, and incorporates its estimate of its relative priority into medium access control. In particular, each packet has an associated priority index which can be computed with purely local information (e.g., a deadline). When a node issues a Request To Send (RTS) in IEEE 802.11 [7, 15], it piggybacks the priority index of its current packet. Nodes that overhear this RTS will insert an entry into a local scheduling table. If the node is granted a CTS, it includes the priority index of its head-of-line (higher priority) packet in the DATA packet, which is also inserted in the local table by overhearing nodes. Each node can then assess the priority of its own head-of-line packet in relation to its (necessarily partial) list of other head-of-line packets. We show that this information can

be exploited via a minor modification of existing 802.11 prioritized back-off schemes to closely approximate a "global" dynamic priority schedule in a distributed way.

In practice, all nodes are not assured to hear all RTSs due to a number of factors including node mobility, location dependent errors, partially overlapping broadcast regions, and collisions. Thus, each node's scheduling table will be incomplete. To address this issue, we devise a simple analytical model to explore the relationship between the probability, q, that a head-of-line packet is in a node's scheduling table and the system's ability to satisfy its QoS targets. The model indicates and simulations corroborate that even with moderate values of q, the scheme can achieve significant improvements over 802.11 and closely approximate the ideal case of $q = 1$ (corresponding to all RTSs overheard and perfect scheduling tables). For example, in ns-2 simulations with 38 nodes transmitting and 74% load, we found that with $q = 0.60$, the scheme reduces the mean delay from 2.86 secs (for 802.11) to 0.6 secs.

Our second contribution is *coordinated multi-hop scheduling*, a mechanism for modifying downstream priorities based on a packet's upstream service in order to better satisfy end-to-end QoS targets across multiple nodes of ad hoc networks. In particular, with a distributed random access protocol and bursty traffic arrivals, not every packet will satisfy its local QoS target, even if $q = 1$. We show that by recursively computing a packet's priority index based on its previous (upstream) index, downstream nodes can help packets catch up if they are excessively delayed upstream, whereas packets arriving early can have their priority reduced to allow more urgent packets to pass through quickly.

We then describe several multi-node policies within this framework. For example, we describe delay and rate-based policies in which flows can target a maximum delay or minimum service rate respectively. To quantify the performance impact of multi-hop coordination, we extend the aforementioned analytical model to include multiple broadcast regions and flows forwarded over multiple hops. Moreover, we study its performance gains via simulations and find for example, that under a simple policy of a single per-hop local delay target and 90% load, coordination decreases the average delay by 60% as compared to 802.11 and by 25% as compared to distributed priority scheduling without coordination.

Thus, together, distributed priority scheduling and multi-hop coordination provide a framework for distributed medium access control and scheduling designed to satisfy end-to-end QoS targets. Our contribution is to introduce these mechanisms, develop an analytical model to characterize their effect, devise simple policies to illustrate their application, and perform simulation experiments to quantify their performance in more realistic environments.

The remainder of this paper is organized as follows. In Section 2 we present distributed priority scheduling. In Section 3 we describe multi-hop coordination. Finally, in Section 4 we review related work and in Section 5 we conclude.

2. DISTRIBUTED PRIORITY SCHEDULING
2.1 Preliminaries
In this section, we devise a scheme for approximating a dynamic priority scheduler within a broadcast region (a region in which all nodes are within radio range of all other nodes) controlled by a CSMA/CA scheme. Our technique applies to the class of schedulers in which packets are serviced in increasing order of a priority

index, where the index can be computed using only flow and node information, i.e., state available at the node or carried in the packet, and not state of other flows. This class includes Earliest Deadline First and Virtual Clock (VC) [21], the two schedulers that we focus on throughout this paper. In EDF, a packet arriving at time t and having (class) delay bound d has deadline (priority index) $t + d$. In virtual clock, a packet with size L of a flow with service rate r has a priority index of L/r plus the maximum of the current time t and the priority index of the flow's previous packet.

Observe that this class of schedulers does *not* include Weighted Fair Queueing [16], as computation of a packet's priority index in WFQ requires knowledge of whether or not other flows are backlogged, information that we will see is problematic to obtain in a distributed environment.

For a given set of packets in a broadcast region and a given packet service discipline such as EDF or VC, an ideal system would service packets exactly in order of their priority indexes. We refer to such a hypothetical schedule as the *ideal* or *correct* schedule and seek to design distributed algorithms to closely approximate this service order. Finally, we refer to a node's head-of-line (HOL) packet as the packet with the highest priority (lowest index) that is queued locally. Thus, each node has a unique HOL packet (if any).

2.2 A Mechanism for Distributed Approximation of Priority Schedules
As described in the Introduction, a centralized scheduler with knowledge of all packet priority indexes can in principle schedule packets in exact order of the ideal schedule. However, due to the distributed nature of *ad hoc* wireless networks, each node is equipped with its own buffer state (local information), and at best partial information about other nodes. Thus, it is immediate that if the scheduler is distributed, with incomplete system information, the ideal schedule cannot be met exactly.

To better approximate the ideal schedule, we propose to exploit the broadcast nature of the medium and piggyback priority indexes of the current and HOL packets. The proposed piggyback mechanism allows for efficient exchange of information while imposing minimal overhead. Each node then maintains a local scheduling table and on hearing newly announced priority indices, adds them to the local table. This local table is then adaptively used to control the channel access policy used by the node. We emphasize that all policies are completely distributed with information exchange relying solely on existing broadcasts by each node. Using analytical and simulation based studies, we show that the piggybacked information can yield significant gains in the probability of transmitting packets in order of the ideal schedule, and corresponding reductions in packet delay. Moreover, we show that these gains can be achieved while maintaining high levels of throughput.

Working in the framework of IEEE 802.11, service differentiation in the MAC protocol can be obtained by varying the backoff timer distribution, the defer time (DIFS), and the size of the packets [1]. Assuming that packet lengths cannot be controlled by the MAC layer for real-time traffic, we focus our attention on the first two parameters and next present our proposed mechanism for distributed priority scheduling and adaptive backoff for IEEE 802.11.

2.3 Proposed Algorithm

In this section, we first briefly review the IEEE 802.11 distributed coordinated function; for more details, readers are referred to [7]. Next, the proposed information exchange mechanism using piggybacked priority tags is presented. Finally, we introduce adaptive backoff policies for IEEE 802.11 that exploits this additional information.

2.3.1 IEEE 802.11 Distributed Cooordination Function

In IEEE 802.11, there are two common modes of packet transmission: a basic access mechanism with a two-way handshake and a four-way handshake mechanism with short request packets before the actual transmission. In this paper, we focus on the four-way handshake depicted in Figure 1. A node which intends to transmit a packet waits until the channel is sensed idle for a time period equal to Distributed InterFrame Spacing (DIFS). If the channel is sensed idle for a duration of DIFS, the node generates a random backoff interval before transmitting (this is the collision avoidance feature of the protocol). In addition, to avoid channel capture, a node must wait a random backoff time between two consecutive new packet transmissions, even if the medium is sensed idle in the DIFS time.

SIFS SIFS SIFS DIFS

Source: RTS | DATA
Destination: CTS | ACK
Others: NAV (RTS) | NAV (CTS)
Delayed Medium Access | Channel Access with Backoff
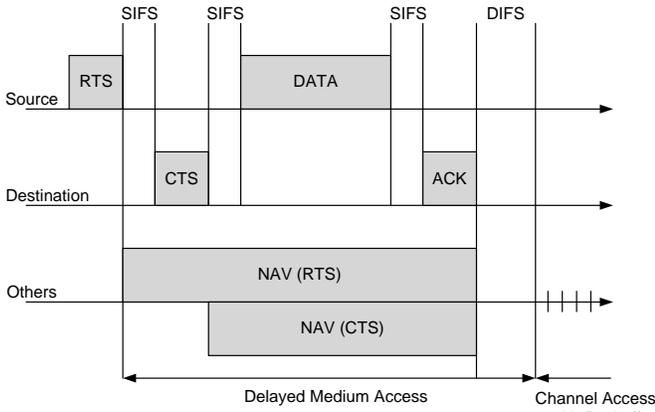
Figure 1: IEEE 802.11 four-way handshake.

A discrete backoff timer is used for reasons of efficiency, and the time following an idle DIFS is slotted. A node is allowed to transmit only at the beginning of each slot time. Further, DCF uses a binary exponential backoff scheme. At each packet transmission, the backoff timer is chosen uniformly from the range $[0, w - 1]$, where w is called the contention window. At the first transmission attempt, w is set to $\mathrm{CW}_{\min}$ which is labeled *minimum contention window*. After each unsuccessful transmission, the value of w is doubled, upto the maximum value $\mathrm{CW}_{\max} = 2^m \mathrm{CW}_{\min}$.

The backoff timer is decremented as long as the channel is sensed idle, and stopped when a transmission is detected on the channel. The backoff timer is reactivated when the channel is sensed idle again for more than a DIFS amount of time. The node transmits when the backoff timer reaches zero. The first transmission is a short request to send (RTS) message. When the receiving node detects an RTS, it responds after a time period equal to the Short InterFrame Spacing (SIFS) with a clear to send (CTS) packet. The transmitting node is allowed to transmit its actual data packet only if the CTS packet is correctly received.

The RTS and CTS packets have information regarding the destination node and the length of the data packet to be transmitted. Any other node which hears either the RTS or CTS packet can use the data packet length information to update its network allocation vector (NAV) containing the information of the period for which the channel will remain busy. Thus, any hidden node can defer its transmission suitably to avoid collision.

2.3.2 Priority Broadcast

To distribute information about the current and HOL packets at other nodes, we propose to piggyback current packet information in the RTS/CTS frames and the HOL packet information in DATA/ACK frames (see Figure 2). The piggybacked information includes the packet priority tag and source node ID for CTS, and only the packet priority tag for RTS frame. Source/destination IDs require four bytes in IPv4 and sixteen bytes in IPv6 and priority tags can be represented using one byte. If the RTS suffers no collisions, then all nodes in the broadcast region hear the RTS (node 9 in Figure 2) and add an entry in their local scheduling table. When the receiving node grants a CTS, it also appends the priority in the CTS frame. This allows the hidden nodes (node 7 in Figure 2), which are unable to hear the RTS, to add an entry in their scheduling tables upon hearing the CTS. Upon the successful completion of the packet transmission, which is marked by the ACK frame, each node removes the current packet from their scheduling table. If either the CTS is not granted by the receiving node or the ACK frame is not received, the current packet information is not removed from the scheduling tables.

CURRENT PACKET INFO   HOL PACKET INFO

SIFS SIFS SIFS DIFS

Source (node1): RTS | DATA
Destination (node 3): CTS | ACK
Others: NAV (RTS) | NAV (CTS)

(node 9)

2	4	P6
3	5	P11

2	4	P6
3	5	P11
1	3	P20

2	4	P9
3	5	P17
1	3	P20
1	4	P7

2	4	P5
3	5	P12
1	4	P7

NODE 9 SCHEDULING TABLE UPDATES

(node 7)

7	8	P10
8	9	P19

7	8	P10
8	9	P19
1	3	P20

7	8	P8
8	9	P1
1	4	P7

NODE 7 SCHEDULING TABLE UPDATES
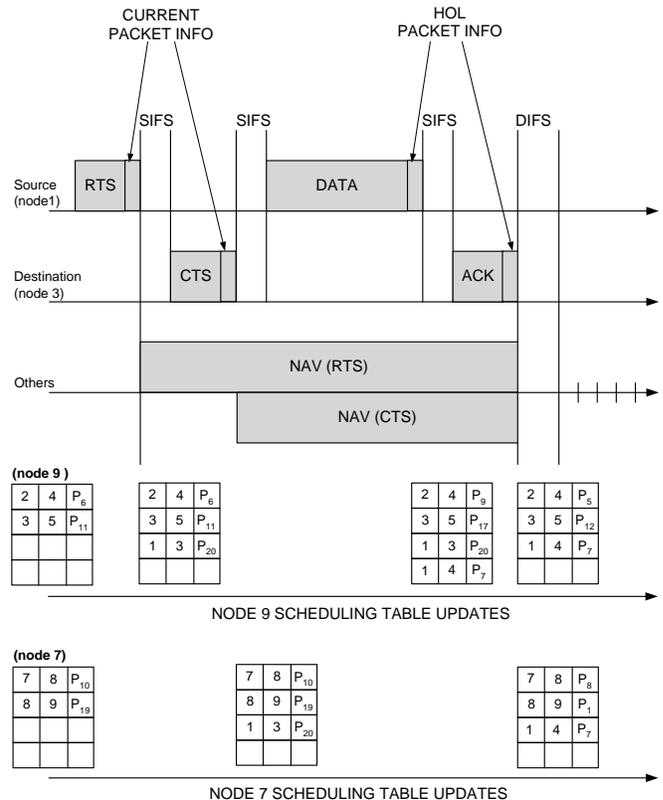
Figure 2: Piggybacking on IEEE 802.11 four-way handshake, and the updating of scheduling tables.

When transmitting the DATA packet, each node also piggybacks its

HOL packet information, which includes the destination and source ID along with its priority tag, a total of nine bytes for IPv4 and thirty three bytes for IPv6; this information is also copied in the ACK frame to allow hidden terminals to hear the HOL packet information. Each node, after hearing the data packet, adds another entry in its scheduling table. Thus, if the ACK is not heard, each waiting node's scheduling table would grow by two entries; otherwise it would have one additional entry. Thus, in the event of a successful transmission, each overhearing node has the same additional entry in its table. On the other hand, if the transmission was not successful, each overhearing node has either one or two additional entries in its tables.

In any case, our design philosophy considers that the common case will be for nodes to have incomplete scheduling tables, and our goal is to closely approximate the ideal schedule even under more adverse and realistic conditions.

2.3.3 Modified Backoff Policies
Here, we describe how the overheard information in each local scheduling table can be mapped to a backoff scheme, thereby using partial knowledge of other nodes' HOL packets to closely approximate the ideal transmission schedule.

Let n denote the number of nodes in the broadcast region. The scheduling table of node j, $\mathbf{S}_j$ is a list of three tuples, (s_i, d_i, P_i), where s_i is the source node ID, d_i is the destination node ID and $P_{(i)} \in \{P_{\min}, P_{\min} + 1, \ldots, P_{\max}\}$ is the priority index of the packet. Thus,

$$\mathbf{S}_j = \left\{ (s_i, d_i, P_{(i)}) : s_i \neq d_i; s_i, d_i \in \{1, \ldots, n\}; 1 \leq i \leq t_j \right\},$$
(1)

where t_j is the size of the scheduling table $\mathbf{S}_j$. If node j is backlogged, then its scheduling table consists of an entry with $s_i = j$. Without loss of generality, we assume that the scheduling table entries are sorted such that $P_{(1)} \leq P_{(2)} \leq \cdots \leq P_{(t_j)}$.

In the context of IEEE 802.11, a collision resolution policy involves selecting a backoff timer distribution. In other words, given the scheduling table $\mathbf{S}_j$, the channel access policy computes $f(\mathbf{S}_j)$, which is the backoff timer distribution. We limit our attention to the following class of distributions,

$$f_l(\mathbf{S}_j) = W_l(\mathbf{S}_j) + \text{Uniform}[0, 2^l g_l(\mathbf{S}_j) - 1], \qquad (2)$$

where $\text{Uniform}[a, b]$ represents the discrete uniform distribution on the range from a to b. The function g_l maps a scheduling table, $\mathbf{S}$ to a real number greater than 1. The index $0 \leq l \leq (m - 1)$ represents the number of retransmission attempts. The constants $W_l(\cdot)$ denotes the additional waiting time beyond DIFS, and allows for the possibility of *contention reduction* (explained below). In IEEE 802.11, $W_l(\cdot) \equiv 0$ and $g_l(\cdot) \equiv \text{CW}_{\min}$. Note that only the backoff distributions have been changed and the remaining components of collision resolution/avoidance are identical to those of in IEEE 802.11.

Let r_j be the rank of node j's packet in its own scheduling table $\mathbf{S}_j$. Then the proposed backoff policy, characterized by distributions $f_l(\cdot)$, is:

$$f_l(\mathbf{S}_j) = \begin{cases} \text{Unif}[0, 2^l \text{CW}_{\min} - 1], & r_j = 1, l < m \\ \alpha \text{CW}_{\min} + \text{Unif}[0, \gamma \text{CW}_{\min} - 1], & r_j > 1, l = 0 \\ \text{Unif}[0, 2^l \gamma \text{CW}_{\min} - 1], & r_j > 1, l \geq 1 \end{cases}$$
(3)

Here $g_l(\cdot)$ is a two-part function $g_l(\cdot) = \text{CW}_{\min}$ if rank of the node $r_j = 1$ else $g_l(\cdot) = \gamma \text{CW}_{\min}$ if $r_j > 1$. The policy uses a combination of contention reduction and collision resolution. The contention reduction is achieved by deterministically deferring the transmission beyond DIFS for some of the nodes, thereby reducing the contention in first $\alpha \text{CW}_{\min}$ time slots. The constant α controls the extent of contention reduction. If $\alpha = 1$, then all nodes which are not ranked one in their scheduling table do not contend for the first $\text{CW}_{\min}$ slots, thereby reducing the contention in first attempt for top ranked nodes (recall rank is determined from the local scheduling table). For q close to one, $\alpha = 1$ would imply that the highest rank node will capture the channel successfully with high probability in the first attempt. The constant γ controls the total contention in the second attempt for highest rank nodes. For small γ, the contention after $\text{CW}_{\min}$ increases significantly, since all waiting nodes contend. This also means increased collisions and potential throughput loss. A larger γ allows for reduced probability of collision and provides a better chance of successful channel capture; we use $\gamma = 2$ to allow equal contention for all nodes after the first $\text{CW}_{\min}$ slots.

Finally note that the policy is independent of the size of the network and the number of overheard HOL indexes in the scheduling tables. Regardless, the performance of the above policy improves when the scheduling table contains a higher fraction of the backlogged nodes' HOL indexes: however, below we show that even with tables that are quite incomplete, the performance gain of a perfect table can be closely approximated.

2.4 Analytical Model
In practice, a number of factors preclude nodes from having complete scheduling tables with an entry for every HOL packet in the broadcast region. These factors include node mobility, location dependent errors, partially overlapping broadcast regions, and collisions. In this section, we develop a simple analytical model to explore the relationship between the completeness of the scheduling table and ability to approximate the ideal schedule. Using results from [5], we compute the probability of correct scheduling as a function of available information in local scheduling tables. With the aid of simplifying assumptions, our results provide insights into the basic role of information sharing (communicating priority indexes) in distributed scheduling.

In [5], an analytical model is developed to compute the 802.11 DCF throughput under ideal channel conditions with an assumption that each node always has a packet available to transmit. A Markov model for each node is obtained in [5], by assuming that probability of collision, p, in any slot is independent of transmission history of nodes. Note that the assumption of time-invariant p is a good approximation for large n and $\text{CW}_{\min}$. The Markov model is then used to compute τ, which is the probability that a node transmits in a randomly chosen time slot. The probability τ with n mobile stations, is given as [5]

$$\tau(p) = \frac{2}{1 + CW_{\min} + pCW_{\min} \sum_{i=0}^{m-1} (2p)^i}, \qquad (4)$$

where m is the maximum back-off stage and p is the conditional collision probability given by

$$p = 1 - (1 - \tau)^{n-1}. \qquad (5)$$

Equations (4) and (5) represent a nonlinear system in the two unknowns τ and p, which is shown to have a unique solution [5].

Here, we generalize Equations (4) and (5) to dynamic priority scheduling in which the size of the contention window is a function of the priority of the packet. However for simplicity no exponential backoff is considered, i.e. $m = 0$. Let P_k denote the priority index of the HOL packet of node k, $k = 1, 2, \ldots, n$. Further consider discrete priority indexes, uniformly distributed between $P_{\min}$ and $P_{\max}$. Based on the proposed policy in Section 2.3, the contention window of each node depends on the rank of its HOL packet in its local scheduling table. Let $[0, W_h - 1]$ and $[\overline{W}, W_l - 1]$ be the contention window for the node with the highest rank and all others, respectively, where $\overline{W} = \alpha CW_{\min}$ is the fixed waiting period and $W_h = CW_{\min}$ and $W_l = \beta CW_{\min}$. The probability of transmission for the node(s) with highest priority packet(s) in slots 1 to W_h, is then given by

$$\tau_h = \frac{2}{1 + W_h}. \tag{6}$$

The same probability will be zero for all other nodes in the aforementioned slots; however the probability of transmission for all nodes for slots greater than $\overline{W}$ is given by

$$\tau_l = \frac{2}{1 + W_l - \overline{W}}. \tag{7}$$

Assuming that n_h nodes contend in the first window, the conditional collision probability will be

$$p = 1 - (1 - \tau_h)^{n_h - 1}, \tag{8}$$

for slots 1 to W_h, and

$$p = 1 - \frac{1}{n} \left[n_h \tau_h (1 - \tau_h)^{n_h - 1} (1 - \tau_l)^{n - n_h} + (n - n_h) \tau_l (1 - \tau_l)^{n - n_h - 1} (1 - \tau_h)^{n_h} \right], \tag{9}$$

for slots greater than $\overline{W}$.

With perfect information the number of nodes contending in the first window is limited to those having packets with the highest priority in their queues. And if there is only one such a node, the probability of capturing the channel by the highest priority packet will be equal to one. Since the ties are broken at random, the number of contending nodes in the first window can be greater than one, thereby improving the approximation for τ_h even for perfect information case.

However, as described above, perfect information on all nodes' HOL packets will not be available in practice. Thus, we model the imperfect information by assuming that each node has a scheduling table entry of the HOL packet of any other node (in the broadcast) region with probability q.

With $q < 1$, nodes can mistakenly infer that the highest priority packet in the entire broadcast region (i.e., the packet that would be selected by the ideal scheduler) is queued locally, when in fact it is queued at an alternate node. The probability that node i's own HOL packet is the highest priority packet in its table (i.e., node i believes that it possesses the region's highest priority packet), denoted by q_h, is given by

$$q_h = \sum_{l=P_{\min}}^{P_{\max}} P(P_i = l) \prod_{j \neq i}^{n} [P(P_j \geq l)]. \tag{10}$$

Noticing that the priorities of different packets are statistically in-

dependent, Equation (10) can be further simplified as

$$q_h = \sum_{l=P_{\min}}^{P_{\max}} \frac{1}{P_{\max} - P_{\min} + 1} \left[\frac{P_{\max} - l + 1}{P_{\max} - P_{\min} + 1} q + (1 - q) \right]^{n-1}. \tag{11}$$

Hence, during the primary contention window $[0, \overline{W}]$, approximately $q_h \cdot n$ nodes contend for the channel, whereas after that all n nodes may take part in contention.

Denote the probability of successful transmission of the true highest priority packet before any other packet in the broadcast region by $P(T_h)$. This probability for the primary contention window can be written as

$$P_{W_h}(T_h) = \sum_{i=1}^{W_h} (1 - \tau_h)^{q_h n (i-1)} \tau_h (1 - \tau_h)^{q_h n - 1} \tag{12}$$

which is the probability of having i idle slot times, followed by transmission of highest priority packet only. After $\overline{W}$ slots the successful transmission of the highest priority packet will be possible if all other $n - 1$ nodes defer. The probability of correct scheduling for this case is

$$P_{\geq \overline{W}}(T_h) \approx \sum_{i=\overline{W}}^{\infty} \left\{ \left[(1 - \tau_h)^{q_h n} (1 - \tau_l)^{(1 - q_h)n} \right]^{i-1} \right. \left. \tau_h (1 - \tau_h)^{q_h n - 1} (1 - \tau_l)^{(1 - q_h)n} \right\} \tag{13}$$

Finally the general form for $P(T_h)$ is given by

$$P(T_h) = P_{< \overline{W}}(T_h) + P_{\geq \overline{W}}(T_h) \tag{14}$$

We now present numerical investigations applying the analysis above. Figure 3 depicts the probability of correct scheduling vs. number of nodes for different values of q. Results are shown for $P_{\min} = 1$, $P_{\max} = 20$, $W_h = \overline{W} = 31$, $W_l = 63$, and $m = 0$ (maximum back-off stage equal to zero).
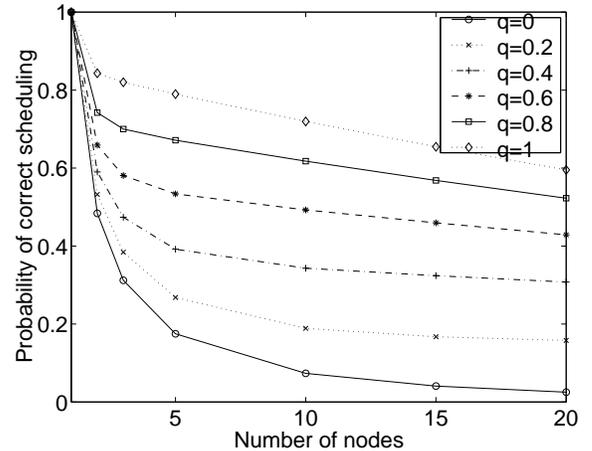
Figure 3: Probability of correct scheduling vs. number of nodes for different values of q.

Observe that as the number of nodes increases, the probability of correct scheduling (transmission of the region-wide highest priority packet first) decreases, reflecting the fundamental challenge of distributed scheduling. Moreover this probability is a function of q,

and indeed, a large value of q mitigates the effect of a large number of nodes. The curve labeled $q = 0$ shows the performance of the system when nodes independently set their contention windows without any knowledge of other packets' priority indexes. This behavior is identical to IEEE 802.11, and the probability of correct scheduling in this case is inversely proportional to the number of nodes in the broadcast region.

On the other hand, with $q = 1$ all nodes have complete knowledge of the priorities of HOL packets in the broadcast region, and hence can adjust their contention window respectively. In this case, a gain of over 40% in probability of correct scheduling is observed for $n = 20$ compared to $q = 0$. Note that even with $q = 1$ the probability of correct scheduling is less than 1. The reason for this is that since the priorities are chosen to be discrete, the probability of two packets having the same priority is non zero. It is clear that as the difference between $P_{\max}$ and $P_{\min}$ increases this probability decreases, and hence this line will approach $P(T_h) = 1$.

Thus, the model and example illustrate the significant benefits of having $q > 0$, i.e., of communicating priority indexes via piggybacking. Moreover, the results indicate that perfect communication of HOL indexes is not required and that moderate values of q have a significant impact. Indeed, in the simulation study below, we find the effect of non-zero q to be even more pronounced.

2.5 Simulation Experiments

Here, we present a set of simulations to explore the performance of distributed priority scheduling and the IEEE 802.11 protocol under realistic scenarios. The simulator was implemented within the *ns-2* (version 2.1b7a).

We consider a single broadcast region with an available link capacity of 2 Mb/sec with an effective data rate of approximately 1.6 Mb/sec (results with multiple broadcast regions and flows traversing multiple hops are presented in Section 3.5). Each node generates variable-rate traffic according to the exponential on-off traffic model with an on-rate of 78 kb/sec, and equal mean on and off times of 500 msec each. The data packet size is set to 1000 bytes. All other parameters (including 802.11 physical layer parameters) were set to the default values as recommended in [7] (also the default values set in ns-2).

In practice, the value of q is affected by a number of factors described previously. In our simulations, nodes update their scheduling tables as follows. Upon receiving a piggybacked RTS, a node enters the priority index into its local scheduling table with probability q, otherwise it ignores the priority information, as would be the case if there were link errors, nodes temporarily moving out of range, etc. In this way, we incorporate a number of effects in a single way and isolate the performance impact of q.

Since our goal with distributed priority scheduling is to approximate an ideal dynamic priority schedule, we chose the performance metric for our simulations to be end-to-end delay. Figure 4(a) depicts the mean delay versus the fraction of available information about other nodes, q. The number of flows is 38 resulting in a mean offered load of 74% (ignoring the overhead of RTS/CTS mechanism for calculation of the load). Figure 4(a) depicts the average values and 95% confidence intervals of end-to-end delay for 100 independent simulation runs. Note the point corresponding to zero available information is the delay under the standard IEEE 802.11 scheme, as our priority scheme degenerates to this standard when

the scheduling table is empty.

We observe that as q increases, distributed priority scheduling results in a significantly lower delay than IEEE 802.11. Also, note that even for a moderate fraction of available information (between 0.6 and 0.8), distributed priority scheduling is able to reduce delay from about 2.9 sec to about 0.4 sec, closely approximating the case of perfect HOL information distribution and $q = 1$. The reduction in delay is due to the fact that distributed priority scheduling achieves a closer approximation to an ideal deadline based schedule than 802.11 so that contention is dramatically reduced. This is further illustrated by Figure 4(b) which shows that distributed priority scheduling leads to a decrease in the total number of collisions. As the number of collisions in the system are reduced, nodes backoff less often resulting in lower delays.
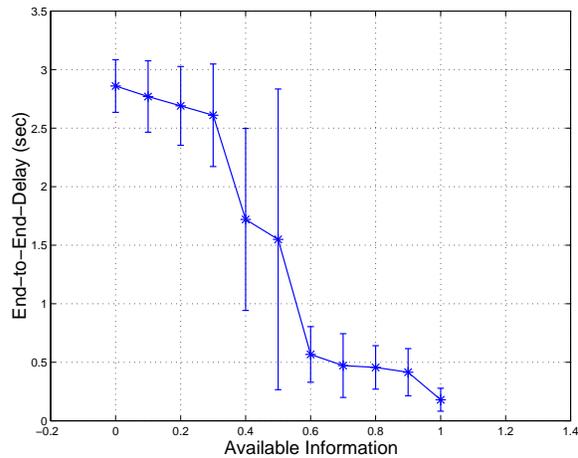
(a) Delay vs. available information
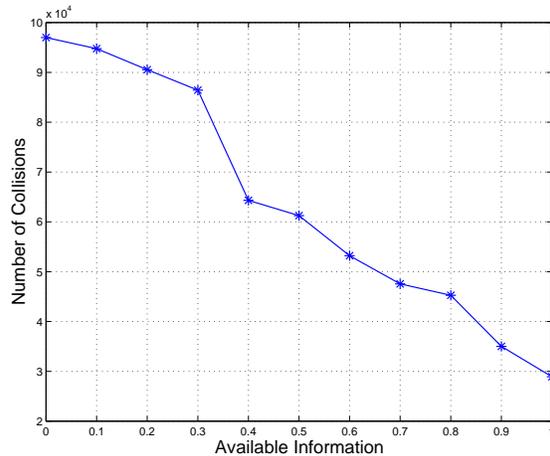
(b) Number of collisions vs. available information

Figure 4: Performance of distributed priority scheduling for a single broadcast region.

3. MULTI-HOP COORDINATION

3.1 A Mechanism for Multi-hop Priority Scheduling

In the previous section we showed how distributed priority scheduling can be used within a broadcast region to approximate the transmission order of an ideal dynamic priority scheduler. However, this transmission order is necessarily imperfect due to factors such

as link errors, the random-access nature of the medium, and the burstiness of the traffic demands. As packets traverse multiple hops, these effects can be compounded, and severely limit a flow's ability to satisfy its end-to-end QoS targets.

Our key observation is that downstream nodes can adjust the priority level of packets based on their performance upstream. In particular, we develop multi-hop coordination as a technique that enables packets to "catch up" downstream if they endure excessive delays upstream, due to events such as collisions, queuing from other bursts of traffic, or mobility of intermediate hops. In this way, we increase the opportunity for a packet to meet its end-to-end QoS target.

3.2 Definition

In [6], the FIFO+ scheduling algorithm is defined (for wired networks) as follows. At the first network node, a packet's priority index is simply its arrival time. Hence, packets are served in FIFO order. However, at downstream node j, an offset of $\bar{d}_j - \widehat{d}_{j-1}^k$ is accumulated into the packet's original priority index, where $\bar{d}_j$ is the mean queueing delay at node j and $\widehat{d}_{j-1}^k$ is the actual delay of packet k at the immediately upstream node. Consequently, if a packet is late relative to others its priority is increased downstream.

Here, we utilize this concept of coordination, and building on the definition in [9], generalize the technique to multi-hop wireless networks as follows. When a node receives a packet, it also receives its priority index in the RTS piggyback. If the node is an intermediate hop and the packet is to be forwarded further, the node will compute the new priority index *recursively* based on the received index. Indeed, we will show that simple coordination functions can have significant impact on end-to-end performance.

More precisely, let $d_{i,j}^k$ denote the priority index assigned to the kth packet of flow-i with size l_i^k at its jth hop. Moreover, let t_i^k denote the time when the kth packet of flow i arrives at its *first* hop. Finally, let $\delta_{i,j}^k$ denote the increment of the priority index of the kth packet of flow i at its jth hop. We consider the class of coordinated multi-hop schedulers such that the priority index can be expressed as

$$d_{i,j}^k = \begin{cases} t_i^k + \delta_{i,1}^k, & j = 1 \\ d_{i,j-1}^k + \delta_{i,j}^k, & j > 1 \end{cases} \quad (15)$$

where $\delta_{i,j}^k$ is a non-negative function of $i, j, l_i^k, t_{i,1}^k$.

With priority recursively expressed in this form, the index of each packet at a downstream node depends on its priority index at upstream nodes, so that all nodes in the *ad hoc* network cooperate to provide the end-to-end service. For example, if a packet violates a local deadline at an upstream node, downstream nodes will increase the packet's priority thereby increasing the likelihood that the packet will meet its end-to-end delay bound. Similarly, if a packet arrives "early" due to a lack of contention upstream, downstream nodes will reduce the priority of the packet.

3.3 Index Assignment Schemes

Priority indexes can be assigned according to whether the flow or class targets an end-to-end delay or rate. In the former case, different per-node allocation schemes can simplify coordination as described below.

3.3.1 Deadline Targets

If $\delta_{i,j}^k$ represents a delay parameter of the kth packet of flow i at its jth hop, then the nodes coordinate to attempt to meet the end-to-end delay target $\sum_j \delta i, j$ to the maximal extent possible under the network load and channel conditions. For such deadline targets, we consider three index assignment schemes.

Time To Live (TTL) Allocation: In this scheme, a packet inserts its desired end-to-end delay bound as its priority index. Each node decrements the TTL by the time taken to access the channel and transmit the packet. In particular, the priority indexes are given by $\delta_{i,1}^k = TTL_i$ and $\delta_{i,j}^k = 0$ for $j > 1$. We refer to this scheme as the *TTL* scheme since TTL_i can be interpreted as the time duration for each packet of flow i to be allowed to stay in network. We make the following observations. First, flows can be differentiated by assigning different TTL's as flows with smaller TTL's will obtain better service than flows with larger TTL's. Second, this scheme gives preference to packets that have traveled several hops and implicitly assumes that the priority of a packet increases with the time it has spent in the network.

Fixed Per-Node Allocation: In this scheme, each node m is assigned a constant G_m, and $\delta_{i,j}^k = G_m$ if node m is the jth hop of the packet. Since the G_m's are independent of packets, different nodes may have different constants, but all packets that are transmitted by the same node have the same increment of priority indexes at that node. While the scheme could be generalized to support different classes, we observe that while the technique is quite simple to implement, a fixed per-node allocation scheme does discriminate against packets with long paths.

Uniform Delay Budget (UDB) Allocation: In this scheme, for a flow i with an end-to-end delay target D, the increments of the priority index of the flow's packets are assigned as $\delta_{i,j}^k = \frac{D}{K}$ for $j = 1, 2, \cdots, K$, where K is the length of the flow's path and can be obtained from routing table under source routing (e.g., as in Dynamic Source Routing [8]). Thus, whereas the TTL scheme allocates the entire delay budget to the first node (but then uses coordination to ensure that packets have sufficient priority downstream), the UDB scheme allocates the delay budget uniformly among nodes.

3.3.2 Rate Targets

In addition to targeting maximum delays, flows can target minimum service rates. In particular, if $\delta_{i,j}^k$ is a function of l_i^k and r_i, where l_i^k is the size of the kth packet of flow i and r_i is the targeted bandwidth for flow i, then a coordinated virtual clock scheduler can be attained by assigning indexes as:[1]

$$d_{i,j}^k = \begin{cases} \max\{t_{i,i}^k, d_{i,1}^k\} + \frac{l_i^k}{r_i}, & j = 1 \\ d_{i,j-1}^k + \frac{l_i^k}{r_i}, & j > 1 \end{cases} \quad (16)$$

We observe that *coordinated* virtual clock attempts to allocate different rates to flows on an end-to-end basis, just as virtual clock [21] allocates rates on a per-hop basis. Note however, that despite the targets of minimum rates, this scheduling algorithm is not targeting max-min fairness for reasons described in Section 4.

---

[1] We note that such allocation is similar to core-stateless jitter virtual clock [18] which uses a related index assignment scheme to achieve scalability.

3.4 Analytical Model

In this section, we devise a simple analytical model of multi-hop coordination and compare the probability of meeting an end-to-end delay bound over a multi-hop path with and without coordination.

Our aim is to characterize transmission and medium access delays, and hence we neglect queuing delay by assuming that there is always one and only one packet in each node's queue. With this assumption, the delay that each packet suffers at any node is completely determined by the time required to successfully reserve the channel in the presence of competing nodes. The total delay consists of time slots which were idle, slots with collisions and slots with successful packet transmission by other nodes. Let δ denote a slot time, and assume that a successful transmission and a collision need respectively T_s and T_c slots; then the delay D can be expressed as

$$D = (n_i + T_s \cdot n_s + T_c \cdot n_c)\delta,$$

where n_i, n_s, and n_c represent the number of idle slots, slots used for successful transmissions and slots with collisions, respectively. Assuming that the probability of transmission is high, and that $T_s \gg T_c$ and $T_s \gg 1$, then the delay can be approximated by $D \approx T_s \cdot n_s \cdot \delta$. In other words, in a single broadcast region the delay that a packet suffers before its successful transmission is approximately equal to the number of successful transmissions of other packets during its defer interval. Hence, to compute the probability that a given packet meets its single-hop delay bound, it suffices to find the distribution of the number of successful transmissions prior to the aforementioned packet.

To simplify the analysis, we consider only the case with high q. This assumption implies that with high probability, the highest priority node will capture the channel in the first contention window. Furthermore, since the priorities are time based, we assume that the probability that two packets have exactly the same priority tag is negligible. Combined with the above two assumptions, it immediately follows that if a packet has the highest priority in a broadcast region, it will capture the channel in the first transmission attempt. Then the probability of capturing the channel in any given contention window is equal to the probability that a node is ranked one for a given priority tag. For delay priority indexes, the priority tags can be set based on TTL. If we assume that $TTLs$ are distributed uniformly from 0 to T_d (the end-to-end delay), then the probability that a node has rank r among n nodes in a broadcast region is

$$P(\text{ rank } = r|TTL) = \binom{n-1}{r-1}\left[\frac{Td - TTL}{Td}\right]^{n-r}\left[\frac{TTL}{Td}\right]^{r-1} \tag{17}$$

Since a node can transmit if and only if it is ranked 1 (based on the assumptions above), it follows that the probability of delay equal to $kT_s\delta$ slots for node 1 with arrival TTL as TTL_0 is

$$P(kT_s\delta|TTL_0) = \prod_{z=0}^{(k-2)} (1 - P(r_1 = 1|T_1))\, P(r_1 = 1|T_2) \tag{18}$$

where

$$T_1 = TTL_0 - z\delta T_s \quad \text{and} \quad T_2 = TTL_0 - (k-1)\delta T_s.$$

Using the above delay distribution, the probability of meeting deadlines can be computed for multi-hop networks with arbitrary configuration. For illustration, we present the results for a two-hop flow with an end-to-end delay target of T_d; Coordinated multi-hop scheduling, UDB without coordination and IEEE 802.11 are an-
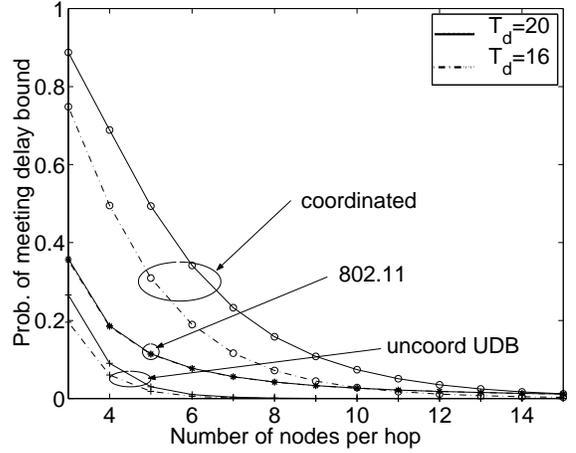
Figure 5: Probability of satisfying end-to-end delay target under different priority schemes .

alyzed. For the sake of simplicity, we assume that the number of competing nodes per hop is fixed to be n. Then the probability of meeting the end-to-end deadline for coordinated multi-hop scheduling, Π_{coord} is given by

$$\Pi_{\text{coord}} = \sum_{k=1}^{K-1} \sum_{l=1}^{K-k} P(kT_s\delta|T_d)P(lT_s\delta|T_d - kT_s\delta), \tag{19}$$

where $K = \lfloor \frac{Td}{Ts} \rfloor$. For UDB with no coordination (under the assumption that a packet is dropped if it fails to meet its local deadline of $T_d/2$), the probability of meeting the deadline Π_{uncoord} is given by

$$\Pi_{\text{uncoord}} = \sum_{k=1}^{\lfloor K/2 \rfloor} \sum_{l=1}^{\min(K-k,\lfloor K/2 \rfloor)} P(kT_s\delta|T_d)P(lT_s\delta|T_d - kT_s\delta). \tag{20}$$

It follows immediately from (19) and (20) that $\Pi_{\text{coord}} \geq \Pi_{\text{uncoord}}$. Finally, for IEEE 802.11 (which has no coordination), the probability of meeting the deadlines is given by

$$\Pi_{802.11} = \sum_{k=1}^{K-1} \sum_{l=0}^{K-k-1} \frac{1}{n^2}\left[\frac{(n-1)}{n}\left(\frac{n-1}{n}\right)^{n-2}\right]^{m+k-1}. \tag{21}$$

In Figure 5, sample numerical results are shown for $T_d = 20$ msec and $T_d = 16$ msec, with $T_s = 2$ msec. From Figure 5, it is clear that coordination can significantly improve the probability of meeting end-to-end deadlines. However, we also note that under heavier loads, depicted by an increased number of nodes per hop, all schemes are degraded significantly. For example for the smaller deadline of $T_d = 16$ msec, IEEE 802.11 is superior to coordination with a heavy load corresponding to more than 11 nodes. In this case, the delay target of 16 msec is unrealistic in the current load, and many packets are dropped. A higher delay target would be required to efficiently coordinate priorities in this regime.

3.5 Simulation Experiments

Here, we experimentally study the performance impact of internode coordination by comparing the coordinated multi-hop scheduling with uncoordinated scheduling as well as unmodified IEEE 802.11. We consider a scenario extending that of Section 2.5 to the
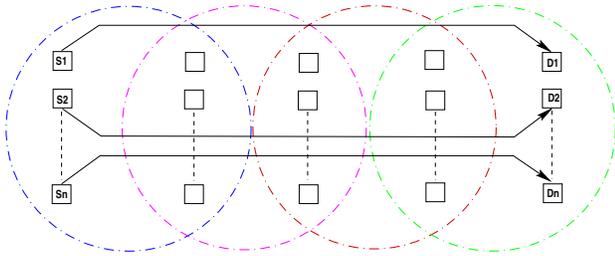
Figure 6: Simulation topology

topology depicted in Figure 6. We consider n exponential on-off traffic flows that traverse at least 2 hops from source S_k to destination D_k, $k = 1, 2, \cdots, n$. We use the UDB (uniform delay budget) scheme with an end-to-end delay target of 240 msec for each flow. We use the average end-to-end delay as our performance metric and explore the variation in mean delay with varying offered load. To compute the offered load with multiple overlapping broadcast regions, we calculate the normalized average rates of all contending traffic flows in a single broadcast domain.

We also note that while nodes remain in a fixed position throughout the simulation, we use an ad-hoc routing protocol since the routes are not setup at start time. For our simulations we use the Dynamic Source Routing (DSR) protocol [8].
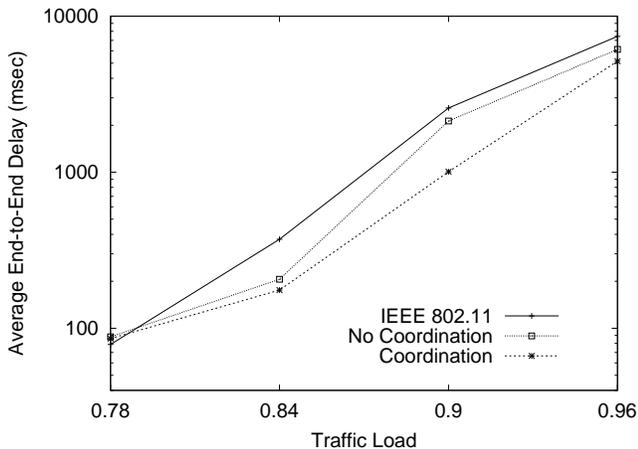
Figure 7: Simulated delay performance of multi-hop coordination.

The simulation results presented in Figure 7 depict the end-to-end delay performance for the three schemes. We make two observations regarding this figure. First, when the traffic load is above 80%, the coordinated service discipline and the uncoordinated service discipline always outperform the IEEE 802.11 DCF MAC protocol. This is a consequence of using distributed priority scheduling within each broadcast region. Thus the prioritized medium access reduces the packet collision and avoids repeated back-off intervals.

Second, notice that for very high traffic load (90%), the coordinated scheme outperforms uncoordinated by more than 50%. This is due to the fact that the coordinated multi-hop scheduling explicitly targets satisfaction of a flow's end-to-end delay target, yielding opportunities to mitigate the effects of poor service received upstream.

4. RELATED WORK

In wireless networks with base stations, recent results in scheduling have shown how to best achieve fairness and weighted fairness in the presence of link errors, e.g., [4, 10, 14]. As described in the Introduction, new issues arise in the case of ad hoc networks without base stations. For example, how to achieve fairness accounting for the distributed nature of the nodes that contend for the same medium, the limits of information exchange between the nodes, the fact that packets of a multi-hop flow contend with each other in successive hops, and the need for spatial re-use are topics of intense recent study and progress [3, 11, 12, 13, 19, 20].

In contrast, our goal of achieving delay or rate QoS targets can yield significantly different schedules than those to achieve fairness or even weighted fairness. For example, for satisfying delay constraints, EDF is easily shown to outperform WFQ. Furthermore, in our problem formulation, if a flow endures location dependent errors, no attempt is made to increase service later for the sake of achieving fairness. Indeed increasing service to such a "lagging flow" could be wasteful if the packets' deadlines have passed. Regardless, our use of multi-hop coordination would increase a packet's priority downstream if it is delayed upstream (for whatever reason), yet the goal is to satisfy the delay or rate constraint rather than to achieve system wide fairness.[2] Regardless, techniques developed for fairness could also be incorporated into our scheme. For example, the ideal schedule could be modified to satisfy QoS targets subject to limits on unfairness or a minimum level of spatial reuse.

The coordination mechanism has been studied previously to improve multi-node performance properties [2, 6, 9]. For example coordinated EDF was studied in [2, 9] as a mechanism for minimizing end-to-end delays in networks of work-conserving schedulers. Likewise, FIFO+ was proposed in [6] as an alternative to both FIFO and fair queueing for delay-sensitive traffic: in FIFO+, downstream nodes adjust a packet's priority index based in its upstream queueing delay. In our work, we generalize the technique for application to ad hoc networks, consider both delay- and rate-based coordination, and integrate coordination with MAC-layer mechanisms.

In [3], a distributed scheduling algorithm was proposed to approximate first come first serve in *ad hoc* networks, also using piggybacked information regarding the HOL packets. Our results are more general as we consider non-perfect information exchange (the delay and throughput analysis in [3] implicitly assumes perfect information about the other nodes' packets, equivalent to the case of $q = 1$), a general class of dynamic priority schedulers, the medium access algorithm, and multi-hop scenarios.

In [1], the authors propose modifications to the IEEE 802.11 protocol to achieve performance differentiation. In particular, the authors explore a number of differentiation mechanisms and conclude that the most superior scheme is to use a DIFS-based approach in which each class has a different value of DIFS: since stations must wait at least DIFS before attempting to access the medium, flows in classes with the smallest value of DIFS receive the best performance. Meanwhile, back-off schemes are left unmodified to retain the desirable stability properties of 802.11. In our work, our goal is to satisfy a dynamic priority scheduler rather than a static priority

---

[2]Observe that virtual clock (a scheduler that we also use as an ideal baseline) possesses the *isolation* property which, similar to WFQ, can be used to provide minimum service rates. However, unlike WFQ, virtual clock does not assure max-min fairness.

schedule. Consequently, nodes use distributed priority scheduling to assess their relative priority before adjusting their value of DIFS. Regardless, techniques and lessons learned in [1] are also applicable in our scheme of distributed priority scheduling and multi-hop coordination.

5. CONCLUSIONS

This paper addresses three issues fundamental to quality-of-service scheduling in ad hoc networks: distributed priority scheduling, priority-based medium access and multi-hop priority management. We introduced a distributed scheduling scheme in which the priority index of a head-of-line packet is piggy backed onto existing messages so that other nodes can better assess the relative priority of their own head-of-line packet. We devised a simple mechanism to incorporate this priority information into the IEEE 802.11 protocol and achieve most of the gains of an ideal schedule with only a moderate fraction of piggybacked messages overheard. We devised a multi-node scheduling algorithm such that downstream nodes can make up for excessive latencies incurred upstream via multi-hop coordination: given the random nature of many aspects of wireless ad hoc networks, we showed how coordination is an important ingredient for targeting end-to-end QoS objectives. Finally, we used analytical models and simulation experiments to quantify the performance impact of the scheme.

6. ACKNOWLEDGEMENTS

7. REFERENCES

[1] I. Aad and C. Castelluccia. Differentiation mechanisms for IEEE 802.11. In *Proceedings of IEEE INFOCOM'01*, Anchorage, Alaska, April 2001.

[2] M. Andrews and L. Zhang. Minimizing end-to-end delay in high-speed networks with a simple coordinated schedule. In *Proceedings of IEEE INFOCOM '99*, New York, NY, March 1999.

[3] C. Barrack and K-Y. Siu. A distributed scheduling algorithm for quality of service support in multiaccess networks. In *Proceedings of IEEE ICNP '99*, Toronto, Canada, October 1999.

[4] P. Bhagwat, P. Bhattacharya, A. Krishna, and S. Tripathi. Enhancing throughput over wireless LANs using channel state dependent packet scheduling. In *Proceedings of IEEE INFOCOM'96*, San Francisco, CA, March 1996.

[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] D. Clark, S. Shenker, and L. Zhang. Supporting real-time applications in an integrated services packet network: Architecture and mechanism. In *Proceedings of ACM SIGCOMM '92*, pages 14–26, Baltimore, Maryland, August 1992.

[7] IEEE. IEEE standard 802.11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, 1997.

[8] D. Johnson and D. Maltz. *Mobile Computing*, chapter Dynamic source routing in Ad Hoc wireless networks. Kluwer Academic, 1996.

[9] C. Li and E. Knightly. Coordinated network scheduling: A framework for end-to-end services. In *Proceedings of IEEE ICNP '00*, Osaka, Japan, November 2000.

[10] S. Lu, V. Bharghavan, and R. Srikant. Fair scheduling in wireless packet networks. *IEEE/ACM Transactions on Networking*, 7(4):473–489, August 1999.

[11] H. Luo and S. Lu. A topology independent fair queueing model in Ad Hoc wireless networks. In *Proceedings of IEEE ICNP'00*, Osaka, Japan, August 2000.

[12] H. Luo, S. Lu, and V. Bharghavan. A new model for packet scheduling in multihop wireless networks. In *Proceedings of ACM MOBICOM'00*, Boston, MA, August 2000.

[13] T. Nandagopal, T. Kim, X. Gao, and V. Bharghavan. Achieving MAC layer fairness in wireless packet networks. In *Proceedings of ACM MOBICOM'00*, Boston, MA, August 2000.

[14] T. Ng, I. Stoica, and H. Zhang. Packet fair queueing algorithms for wireless networks with location dependent errors. In *Proceedings of IEEE INFOCOM'98*, San Francisco, CA, May 1998.

[15] B. O'Hara and A. Petrick. *IEEE 802.11 Handbook, A Designer's Companion*. IEEE Press, 1999.

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

[17] S. Shenker. Fundamental design issues for the future Internet. *IEEE Journal on Selected Areas in Communications*, 13(7):1176–1188, September 1995.

[18] I. Stoica and H. Zhang. Providing guaranteed services without per flow management. In *Proceedings of ACM SIGCOMM '99*, Cambridge, MA, August 1999.

[19] N. Vaidya and P. Bahl. Fair scheduling in broadcast environments, August 1999. Microsoft Research Tech. Rep. MSR-TR-99-61.

[20] N. Vaidya, P. Bahl, and S. Gupta. Distributed fair scheduling in a wireless LAN. In *Proceedings of ACM MOBICOM'00*, Boston, MA, August 2000.

[21] L. Zhang. *A New Architecture for Packet Switched Network Protocols*. Ph.D. dissertation, Massachusetts Institute of Technology, July 1989.