New Adaptive Protocols for Fine-Level End-to-End Rate Control in Wireless Networks

Ying Jian† Shigang Chen† Liang Zhang† Yuguang Fang‡
†Department of Computer & Information Science & Engineering, University of Florida
‡Department of Electrical and Computer Engineering, University of Florida
{yjian, sgchen, lzhang}@cise.ufl.edu, fang@ece.ufl.edu

***Abstract*—Fine-level rate control, particularly meeting rate requirements and differentiating various types of end-to-end traffic, remains an open problem for multihop wireless networks. Traditionally, rate assurance in wired networks is achieved through resource reservation and admission control, which can be efficiently implemented since the bandwidth capacity of each communication link is known and the sender of a link has the information of all flows that compete for the bandwidth of the link. In a wireless network, however, the capacity of each wireless link can change unpredictably over time due to contention from nearby links and dynamic channel conditions. An end-to-end flow consumes available bandwidth not only at links on its route but also at all nearby contending links, which makes resource reservation extremely complicated. We believe fundamental differences require a fundamentally different paradigm shift in solutions. Is there a simpler alternative to resource reservation and admission control that is better suited for wireless network dynamics? In this paper, we propose a new adaptive rate control function based on two novel protocols, called *dynamic weight adaptation with floor and ceiling* and *proportional packet scheduling*, which together implement prioritized rate assurance and sophisticated bandwidth differentiation among all end-to-end flows in a multihop wireless network without resource reservation and admission control. The adaptive function achieves global rate control objectives in a fully distributed way using only localized operations.**

I. INTRODUCTION

Following the enormous success of WLAN, multihop wireless networks, including mesh networks, ad-hoc networks, sensor networks, are expected to lead in the next wave of deployment. To improve their applicability in practice, not only must these networks provide a robust and efficient communication infrastructure, but also they should provide flexible tools for traffic engineering in order to support diverse user applications. Rate control is an important network function for meeting rate requirements and differentiating various types of end-to-end data flows.

Fine-level rate control remains an open problem in multihop wireless networks, particularly those based on the popular CSMA/CA protocols. The large body of literature for wired networks cannot be applied to wireless networks due to their fundamental differences. Meeting rate requirements is traditionally implemented through resource reservation and admission control in wired networks, which can be efficiently done since the bandwidth capacity of each communication link is known and the sender of a link has the information of all flows that compete for the bandwidth of the link. However, in a wireless network based on CSMA/CA, the capacity of a wireless link is undefined and may change drastically over time, depending on the load of the contending links, the relative positions of the links, and the channel conditions. Even the channel capacity varies from place to place and from time to time due to environmental noise, obstacles, multipath fading, and multi-rate links. For instance, when IEEE 802.11b links select different rates (11Mbps, 5Mbps, 2Mbps and 1Mbps) based on their levels of signal strength, the channel capacity is a variable dependent upon how much time each link occupies the channel. Other unpredictable factors can also come into play. For example, when two wireless networks with overlapping channels are deployed in the same area, the channel capacity perceived by one network will depend on the activities of the other. Admission control cannot be performed if the link/channel capacity is dynamic.

Resource reservation in multihop wireless networks also has problems. An end-to-end flow consumes bandwidth not only at links on its route but also at all nearby contending links, which makes resource reservation extremely complicated. With spatial channel reuse, the local channel perceived by each wireless link is different because each link has a different set of contending links. (Two contending links will consume bandwidth in each other's perceived channel.) Consider a new flow whose rate requirement is r and routing path is $a \rightarrow b \rightarrow c \rightarrow d$. In order to support the flow, the channel perceived by link (a, b) should have $3r$ residual (unused) bandwidth because (a, b), (b, c) and (c, d) mutually contend and they will each consume r bandwidth in the same channel when carrying the flow (assuming IEEE 802.11 DCF). Similarly, the channels perceived by other links on the routing path also need more than r residual bandwidth for the flow. Even links outside of the path need residual bandwidth to support the flow. Consider a nearby link (x, y) that contends with (a, b). Suppose its perceived channel is already *saturated* due to heavy traffic on some other contending links. Now if we add the new flow, as the rate on (a, b) is increased, the rate on (x, y) will be driven down, causing the violation of the previous resource reservation made on (x, y). Determining how much bandwidth (x, y) needs in order to support the new flow is not an easy task. It depends on how much channel spatial reuse can be done between (a, b) and other links contending with (x, y). Therefore, resource reservation requires coordination among links on the route and all other links that contend with them.

Facing the above challenges, the past research has followed three directions. The first direction is to restrict the study on

1-4244-2507-5/08/$20.00 ©2008 IEEE

wireless LANs [1], [2], [3], [4]. When every link sees the same channel with the same set of contending links, many of the above problems are either avoided or much simplified. A different restriction can be assuming each node transmits at a different frequency [5]. The second direction is to work on coarse-level service differentiation that does not provide rate assurance [6], [7], [8], [9]. For example, different backoff policies [7] or different contention window sizes [6] are assigned to packets of different classes to provide qualitative scheduling preferences. IEEE 802.11e belongs to this category when it is applied in a multihop wireless network. The third direction is to design heuristics to address the hard problems in resource reservation and admission control. Most work focuses on establishing a heuristic approach for each node to estimate its channel's residual bandwidth, which will be used to guide admission control. The bandwidth estimation is made based on channel idle time [6], [10], average packet transmission delay [11], [8], [2], or channel-access probabilistic models [12], [13], [14], [15]. As detailed analysis in [15] points out, none of them considers the impact of hidden terminals in multihop wireless networks, and each will perform poorly under certain scenarios. Moreover, the residual bandwidth measured may continuously change due to dynamic channel conditions, and estimating bandwidth does not solve the complicated resource reservation problem discussed previously.

Instead of taking a head-on approach to address the difficult problems of resource reservation and admission control, we want to take a step back and ask whether resource reservation and admission control, legacy from wired networks, are suitable for multihop wireless networks. We want to find an alternative solution for wireless networks that can not only solve the problems but also have a much simpler design. In this paper, we propose to replace admission control and resource reservation with a simple yet effective adaptive rate control function suited for handling network/traffic dynamics. It automatically adapts the bandwidth distribution to satisfy the rate requirements of as many flows as possible *in the order of their priorities*. This *global* objective for all *end-to-end* flows in the network should be implemented solely based on *localized* operations. The adaptive rate control should not require the exchange of any topological or per-flow information among contending nodes, nor should it rely on the accurate measurement of link or channel capacities.

We classify end-to-end traffic into two categories: best-effort flows and QoS flows with minimum rate requirements. The QoS flows are assigned to service classes of different priorities. We have the following three objectives for rate control. The *rate assurance objective* requires QoS flows to be supported in the order of priorities. A higher-priority flow can preempt the bandwidth of a lower-priority flow. Following the priority order, the network should support as many QoS flows as possible. Beyond meeting the minimum rate requirements, the *bandwidth differentiation objective* requires the remaining bandwidth to be allocated to end-to-end flows based on their priorities as well as bandwidth demand. A flow with a higher minimum rate requirement and a higher priority should receive a larger amount of extra bandwidth. The *no-starvation/maximum-utilization objective* requires that no flow is starved and all network bandwidth is utilized when possible.

To achieve the three objectives, we design our adaptive rate control function based on two novel protocols. First, working on top of a MAC layer that supports weighted bandwidth allocation, we propose a new rate control protocol, called *dynamic weight adaptation with floor and ceiling* (DWA), which allows each MAC (one-hop) flow to independently adapt its weight based on local information and acquire an appropriate fraction of channel bandwidth. We show that, when the weights of the MAC flows are adapted between certain upper bounds (ceilings) and lower bound (floors), the three objectives can be met. Adaptation at the MAC layer is common, but such adaptation designed for end-to-end objectives is not. DWA demonstrates great flexibility in bandwidth distribution, yet it is simple to implement, which is important for practical wireless systems. Second, to support DWA, we enhance CSMA/CA protocols for weighted bandwidth allocation through a new *proportional packet scheduling* protocol (PPS), which distributes channel bandwidth among MAC flows in proportion to their weights. Comparing with the existing schemes, PPS is much simpler yet reduces radio collision. It is also the first fully localized solution that achieves provable weighted maxmin fairness in CSMA/CA networks with dynamic flow set and dynamic channel conditions, making it particularly suited for supporting DWA.

The rest of the paper is organized as follows. Section II proposes DWA, the protocol of dynamic weight adaptation with floor and ceiling. Section III proposes PPS, the protocol of proportional packet scheduling. Section IV evaluates our new protocols through simulations. Section V draws the conclusion.

II. Adaptive Rate Control

Our adaptive rate control protocol is called *dynamic weight adaption with floor and ceiling* (DWA), which provides prioritized rate assurance and sophisticated bandwidth differentiation among a dynamic set of end-to-end flows in multihop wireless networks with dynamic channel conditions.

A. Objectives

We classify end-to-end traffic in a multihop wireless network into two broad categories: *best-effort flows* and *QoS flows*. Each QoS flow has a minimum rate requirement, but it may send data at a higher rate if extra bandwidth is available. The rate requirement may be *soft* or *hard*. With a soft requirement, we assume the application is able to adapt to live with a lower-than-expected rate, for example, by compressing data before sending. With a hard requirement, we assume the application will terminate the flow when the rate is too low, and it may attempt to re-establish the flow after a timeout period, which may be doubled for each failed attempt until the application gives up. If the network cannot satisfy a flow's minimum rate requirement, it will continue serving the flow to the best it can. It is up to the application to decide whether adaptation should be performed or the flow should be terminated. We assume each flow has a routing path established by a routing protocol; the subject of optimal routing is beyond the scope of this paper.

The network provides a number of service classes, each having a different priority. Best-effort flows are assigned to the *best-effort service class* that has the lowest priority. QoS flows are assigned to other classes. The priority of a QoS flow is equal to the priority of the service class to which the flow is assigned. When the minimum rate requirement of a QoS flow is satisfied, we say the network *supports* the flow. We have three objectives.

- *Rate Assurance Objective*: QoS flows in the network are supported in the order of their priorities. When the bandwidth available in the network cannot support all end-to-end QoS flows, we first try to support the flows with the highest priority, then try to support the flows with the second highest priority, and this policy repeats until there is no longer enough bandwidth to support flows of a certain priority. In other words, when two end-to-end flows contend for bandwidth at a common bottleneck location, the lower-priority flow will be supported only after the higher-priority is supported.
- *Bandwidth Differentiation Objective*: After the minimum rate requirements of all QoS flows are satisfied, if there is extra bandwidth left, the remaining bandwidth should be distributed to contending flows in proportion to their minimum rate requirements.[1] However, if there is not enough bandwidth to support all QoS flows, after high-priority classes are supported, all remaining bandwidth (which is not enough to meet the rate requirements of low-priority classes) should be distributed to unsupported contending flows in proportion to the product of *minimum rate requirement* and *differentiating factor*. Each service class is pre-assigned a differentiating factor, which increases with the priority of the class. Hence, the above design not only considers each flow's bandwidth demand but also takes the priority into consideration.
- *No Starvation and Maximum Utilization Objective*: While QoS flows are supported in the order of their priorities, low-priority flows (including best-effort flows) should not be starved. Hence, the bandwidth consumed by each service class should be limited to a certain fraction of the available bandwidth in the network. This fraction is proportional to both the differentiating factor of the class and the combined rate requirement of all flows in the class.[2] Consequently, the fraction of bandwidth available to a service class is not only dynamic (due to flow join/departure) but also easily configurable (by changing the differentiating factor). No bandwidth should be wasted; bandwidth assigned to but not used by any flow should be automatically picked up by other flows based on the distribution policies specified in the previous objectives.

[1] Since the minimum rate requirements have already been met, bandwidth demand (instead of priorities) is used as the criterion for distributing the remaining bandwidth. For example, it is reasonable that a video stream is assigned more *extra* bandwidth than an audio stream.

[2] In other words, for each unit of rate requirement, a flow in a higher-priority class will be entitled to a larger share of bandwidth because of its larger differentiating factor.

It should be noted that there are other ways of defining the objectives. For example, in the second objective above, instead of distributing the remaining bandwidth among end-to-end flows based on minimum rate requirement and differentiating factor, one may set a different goal to maximize the aggregate throughput. Optimizing the aggregate throughput will naturally prefer short flows over long flows and disregard the flows' priorities. Proportional fairness [16] can be used to address the problem of short-flow preference. However, integrating proportional fairness with multiple prioritized service classes will complicate our system design. Focusing on rate assurance and bandwidth differentiation in this paper, we shall leave other design choices to future work.

B. Network Model

We consider static multihop wireless networks using CSMA/CA MAC protocols. Each node has a single radio that operates at a single or multiple rates. A wireless link forms between two neighboring nodes that are able to communicate with each other with acceptable reliability.

We model a multihop wireless network as a set of MAC (one-hop) flows. Each wireless link carries one MAC flow for each service class. Two MAC flows contend with each other if they belong to the same link or two contending links. An end-to-end flow of a given priority is mapped to a sequence of MAC flows of the same priority along its routing path. A MAC flow of a given priority carries all end-to-end flows of the same priority that pass the link. The rate requirement of a MAC flow is the summation of the rate requirements of the end-to-end flows that are carried by the MAC flow.

For now, we assume there exists a MAC-layer scheduling protocol that is able to perform weighted bandwidth allocation among contending MAC flows. After each MAC flow is artificially assigned a weight, the scheduling protocol will make sure that the bandwidth shares acquired by contending flows in the same bottleneck channel are roughly proportional to their weights. *We will discuss such protocols in the next section.*

C. Design Overview

There are two levels of bandwidth distribution. At the first level, we perform weighted bandwidth allocation that distributes the channel capacity among contending MAC flows. Whenever possible, each MAC flow should acquire enough bandwidth to support the end-to-end flows it carries, but not too much bandwidth that causes shortage for other MAC flows to support end-to-end flows they carry. At the second level, we distribute the bandwidth acquired by each MAC flow to the end-to-end flows (that it carries) by weighted fair queueing. Both levels use weights, but they are independent of each other. The challenge is on the first level, because once enough bandwidth is acquired by a MAC flow, at the second level, we can simply use the end-to-end flows' rate requirements as weights and perform any classical weighted fair queueing algorithm [17] to assure every end-to-end flow's rate requirement is met (locally at this wireless link). Therefore, we will focus on the first level in the rest of this section.

While the above two-level bandwidth distribution architecture may appear to be a routine design, our novelty is in solving the problem of how to assign appropriate weights to MAC flows (at the first level) such that the objectives in Section II-A can be achieved in a dynamic, fully-distributed environment, where both flows and channel conditions may change and there is not an entity that has global network/traffic information.

To solve this problem, we propose a new rate control protocol called *dynamic weight adaptation with floor and ceiling*. The weight of each MAC flow adapts between a lower bound (called *floor*) and an upper bound (called *ceiling*). The floor is proportional to the rate requirement of the MAC flow, and the ceiling is proportional to the product of the rate requirement and the differentiating factor (which is larger for a MAC flow of higher priority, giving such a flow a higher ceiling). Because the rate requirement of a MAC flow may change as end-to-end flows come and go, the floor and the ceiling may change, too. Each MAC flow periodically adapts its weight between the floor and the ceiling as follow: The sender of the MAC flow measures its rate over each weight-adaptation period. If the measured rate is below the requirement of the MAC flow, the sender increases the weight of the flow at the end of each period until the ceiling is reached or the requirement is satisfied. If the measured rate is above the requirement, the sender decreases the weight at the end of each period until the floor is reached or the rate is equal to or lower than the requirement. By setting the ceiling and the floor appropriately, we can show that, when the weights of all MAC flows stabilize, the three objectives will be met.

D. DWA: Dynamic Weight Adaptation with Floor and Ceiling

We first define some notations. Let $f^k_{i,j}$ be the MAC flow on link (i, j) that carries the service class of priority k. For the best-effort service class, $k = 0$. For the QoS service classes, $k = 1, 2, 3, ...$ Let $w^k_{i,j}$ be the weight of flow $f^k_{i,j}$. Let $L^k_{i,j}$ and $H^k_{i,j}$ be the floor and the ceiling for $w^k_{i,j}$, respectively. Let $q^k_{i,j}$ be the rate requirement of the MAC flow $f^k_{i,j}$, and $r^k_{i,j}$ be the measured data rate of $f^k_{i,j}$. Let d_k be the differentiating factor for priority k. We require $d_k > d_{k'}$ for $k > k'$.

To compute $q^k_{i,j}$, we need to examine two cases. First, if an end-to-end flow carried by $f^k_{i,j}$ has a backlogged queue at i, we should allocate sufficient bandwidth for $f^k_{i,j}$ to support the minimum rate requirement. Second, if the end-to-end flow does not have a backlogged queue and the arrival rate to the queue is smaller than its rate requirement, the flow must have an upstream bottleneck and we only need to allocate enough bandwidth to cover the arrival rate. The *effective rate requirement* of an end-to-end flow is equal to the minimum rate requirement in the first case and the arrival rate in the second case. We define $q^k_{i,j}$ as the summation of the effective rate requirements of all end-to-end flows carried by $f^k_{i,j}$.

We define the floor and the ceiling for the weight $w^k_{i,j}$ of a MAC flow $f^k_{i,j}$ as follows. When $k = 0$, the flow is best-effort and we set the floor and the ceiling to be a fixed small value. Namely, the weight of a best-effort MAC flow is not adaptable. When $k > 0$, its floor and ceiling are

$$L^k_{i,j} = a \times q^k_{i,j}$$

$$H^k_{i,j} = a \times d_k \times q^k_{i,j}$$

where a is a scaling coefficient whose value can be set arbitrarily without changing the network's behavior. For example, if $q^k_{i,j}$ is 10 kbps, then $L^k_{i,j} = 10$ if the scaling coefficient is chosen to be $a = \frac{1}{1kbps}$.

Weight $w^k_{i,j}$ is initialized to be $L^k_{i,j}$ and iteratively adjusted. At the end of each weight-adaptation period, if $r^k_{i,j} < q^k_{i,j}$, we increase $w^k_{i,j}$ by a percentage of β if it has not reached the ceiling yet.

$$w^k_{i,j} \leftarrow \min\{w^k_{i,j} \times (1 + \beta), H^k_{i,j}\}$$

If $r^k_{i,j} > q^k_{i,j}$, we decrease $w^k_{i,j}$ by a percentage of β if it has not reached the floor yet.

$$w^k_{i,j} \leftarrow \max\{w^k_{i,j} \times (1 - \beta), L^k_{i,j}\}$$

It is possible to adapt the value of β based on the gap between $r^k_{i,j}$ and $q^k_{i,j}$. But we found a constant value for β, such as 10%, already worked very well in our simulations.

Below we show that the above design of dynamic weight adaptation is able to satisfy the three objectives. Recall that we assume a MAC-layer scheduling protocol that allocates channel bandwidth to contending MAC flows in proportion to their weights. To facilitate the discussion, we introduce the concept of *normalized weight* for a MAC flow $f^k_{i,j}$, which is defined as

$$\overline{w}^k_{i,j} = w^k_{i,j}/q^k_{i,j}$$

It is the average weight per unit of rate requirement. Because $\frac{L^k_{i,j}}{q^k_{i,j}} \leq \overline{w}^k_{i,j} \leq \frac{H^k_{i,j}}{q^k_{i,j}}$, the lowest normalized rate for any MAC flow is a, and the highest normalized rate is $a \times d_k$, which varies for different priorities. If a MAC flow has a higher normalized weight than another flow under the same contention condition, it will acquire a larger amount of bandwidth for each unit of its rate requirement and therefore has a better chance to satisfy its requirement.

First, consider two end-to-end flows contend for bandwidth at a common bottleneck. Suppose they pass two contending MAC flows, $f^k_{i,j}$ and $f^{k'}_{i',j'}$, in the same bottleneck channel, and $k > k'$. If the rates of the MAC flows are below the minimum requirements, they will both increase weights unless the ceilings are reached. Their ceilings are different. The largest normalized weight for $f^k_{i,j}$ (which is $a \times d_k$) is larger than the largest normalized weight for $f^{k'}_{i',j'}$ (which is $a \times d_{k'}$). Hence, $f^k_{i,j}$ is able to increase its weight further to acquire more bandwidth per unit of rate requirement than $f^{k'}_{i',j'}$. Consequently, if $f^{k'}_{i',j'}$ is supported, $f^k_{i,j}$ must also be supported, but if $f^k_{i,j}$ is supported, $f^{k'}_{i',j'}$ may or may not be supported (due to the constraint of lower ceiling). The rate assurance objective is met. Now, suppose neither $f^k_{i,j}$

nor $f^{k'}_{i',j'}$ can be supported. Their weights will both reach the ceilings, and the bandwidth allocation between them will be proportional to the product of the rate requirement and the differentiating factor. Due to the two-level bandwidth distribution architecture, the ratio of bandwidth allocations among MAC flows will be inherited by the end-to-end flows that the MAC flows carry. Hence, the second half of the bandwidth differentiation objective is satisfied.

Next, we consider the case that the network has enough bandwidth to support the minimum rate requirements of all end-to-end flows. Consider contending MAC flows at an arbitrary location in the network. We first show that all flows will be supported. Any flow whose rate is larger than its minimum requirement will decrease its weight, giving up some bandwidth. If no flow uses more bandwidth than its minimum requirement, certainly every one will be supported. What happens if a flow has a higher rate than the minimum requirement even when its weight is reduced to the floor? Since the flow's normalized weight, now a, is the lowest among all, other flows must be receiving the same or more bandwidth for each unit of rate requirement. Hence, their rate requirements must have been satisfied as well. Now, if there is still extra bandwidth left, because all flows will reduce their weights to the floors in an effort to avoid taking more-than-minimally-required bandwidth, the extra bandwidth will be distributed based on the ratio of the floors, which are proportional to the rate requirements. Therefore, the first half of the bandwidth differentiation objective is met.

Finally, because the weight of each MAC flow has a ceiling ($a \times d_k \times q^k_{i,j}$), it cannot indefinitely increase the fraction of channel bandwidth that it consumes. In other words, the bandwidth consumed by end-to-end flows in a certain priority class is limited at any location in the network; the maximum fraction of bandwidth they can consume is proportional to the differentiating factor d_k, which is configurable. Therefore, the no-starvation objective is also met. To maximally utilize the available bandwidth, the underlying MAC-layer scheduling protocol that implements weighted bandwidth allocation must be work-conserving, i.e., it must allow MAC flows to consume bandwidth left unused by other MAC flows.

E. Avoiding Packet Drops

An end-to-end flow may receive different amount of bandwidth from the links on its path. Let (i, j) be the bottleneck link and (k, i) be the link preceding the bottleneck. Because there is more bandwidth available upstream, i will receive more packets from k than it can forward to j. Its queue for the flow will be filled up and eventually overflowed, causing packet drops. We adopt the congestion avoidance scheme in [18], which allows the upstream node k to send a packet to i only when i has enough free space in the queue to hold the packet. Suppose the buffer space for the queue is slotted with each slot storing one packet. The residual buffer at node i changes when i receives or sends a packet. To keep the upstream node updated with i's buffer state, whenever i transmits a packet (RTS/CTS/DATA/ACK), it piggybacks its current buffer state in the frame header, for example, using one bit to indicate whether there is at least one free buffer slot. When the upstream node k overhears a packet from i, it caches the buffer state of i. If i's buffer is full, k will hold its packets and wait until overhearing new buffer state from i. Readers are referred to [18] for discussion on various issues such as failed overhearing.

F. Flow Dynamics and Channel Dynamics

The adaptive nature of the proposed bandwidth distribution scheme makes it suitable for a dynamic environment with a changing set of flows and evolving channel conditions. We illustrate the adaptation process by the following example. Consider a network with two QoS service classes and all end-to-end QoS flows having the same rate requirement. Suppose at one time each class has one end-to-end flow and the network is able to support both flows. The weights of all MAC flows are at their floors.

First, let another high-priority end-to-end flow join in the network. The flow source signals along the routing path to add its rate requirement to the high-priority MAC flows. The ceilings of the high-priority MAC flows are increased accordingly. Suppose at one location x the channel capacity is not sufficient to support all QoS flows. With their weights at the floors, the MAC flows of two priorities both find that they are not getting enough bandwidth. They will increase their weights. The low-priority flow has a lower ceiling and will stop increasing first. The high-priority flow will be able to get more bandwidth to satisfy the requirement.

Second, let the newly-joined end-to-end flow depart from the network. At location x the bandwidth for the departed flow will be inherited by the other high-priority end-to-end flow sharing the same MAC flow. Since it acquires more bandwidth than the rate requirement, the high-priority MAC flow will reduce its weight to the floor, giving away bandwidth to the low-priority flow.

Third, suppose the channel capacity at location x is decreased due to environmental noise, causing the actual data rates of both MAC flows to decrease below the rate requirements. Similar to the first scenario, their weights will adapt individually and independently, but because the high-priority flow has a higher ceiling, it will receive more bandwidth to meet its rate requirement.

G. Intra-flow Contention and Inter-flow Contention

One may question why we have not discussed intra-flow and inter-flow contentions [19] in bandwidth distribution among end-to-end flows. Consider an end-to-end flow that follows a path $k \to i \to j$ where the intra-flow contention between sub-flow $k \to i$ and sub-flow $i \to j$ may lead to one sub-flow grabbing more bandwidth than the other, while they should each have the same bandwidth. Consider another end-to-end flow that follows the same path. The inter-flow contention may assign the same bandwidth share to each end-to-end flow, while they should be assigned shares based on their rate requirements and priorities. Intra-flow and inter-flow contentions are a problem for random-access wireless networks. However, when we have a MAC-layer scheduling protocol (in the next section) that achieves weighted bandwidth allocation and a congestion avoidance scheme [18] that prevents packet drops, these contentions can be solved by assigning appropriate weights to MAC flows, as we did in this section.

III. Weighted Bandwidth Allocation among MAC Flows

Dynamic weight adaptation proposed in the previous section needs to work on top of a MAC-layer protocol that supports weighted bandwidth allocation. In this section, we design such a protocol.

A. State of the Art and Our Contribution

The MAC-layer packet scheduling protocol to be proposed achieves weighted bandwidth allocation and is particularly suitable for dynamic weight adaptation. First, its operations are fully localized; a node only uses its local information and the information it currently overhears. It does not maintain the state information of its contending flows (which can cause problem if the information becomes stale). Second, it achieves *provable* weighted maxmin fairness in CSMA/CA networks with a bounded error that can be made arbitrarily small. We believe this is a strong result. Third, it does not assume a fixed channel capacity, and does not assume a static set of flows. It can work in a wireless environment where channel capacity evolves spatially/temporally and flows join and depart.

While a number of MAC protocols in the literature [20], [21], [22], [23], [24], [25] were designed to achieve fairness among contending MAC flows, to the best of our knowledge, our protocol is the first to achieve all three properties discussed above. In comparison, EMLM-FQ [23] requires each node to keep track of certain state information of contending flows and does not guarantee a tight bound on its approximate fairness. The max-min fairness achieved in [26] uses a multi-channel contention model that is different from the CSMA/CA model used in this paper. The work in [22] requires each node to compute fair bandwidth shares for its own links and the nearby contending links, which in turn relies on the knowledge of neighborhood topology. To calculate fair shares, a node must know the set of contending flows that are currently backlogged, from which the local contention cliques can be computed. In addition, it assumes all cliques have equal, fixed capacity. These requirements make the protocol less suitable for a dynamic wireless environment where the set of backlogged flows, as well as the channel capacity, may constantly change.

The new scheduling protocol also has other advantages: It is much simpler, easy to implement and analyze, and reduces radio collision. It does not require modifying the backoff algorithm of the existing channel access protocols.

B. Proportional Packet Scheduling

Our objective is to design an enhanced CSMA/CA protocol that allocates bandwidth to MAC flows in proportion to their weights. One solution is to modify the backoff algorithm [22], [24], [27] such that a MAC flow with a larger weight will have a smaller backoff window and thus acquire more bandwidth. We take a different approach called *proportional packet scheduling* (PPS), which does not require changing the backoff algorithm and thus makes the integration with existing MAC protocols easier. To simplify our description, we consider one MAC flow for each wireless link. The extension to multiple MAC flows per link is trivial.

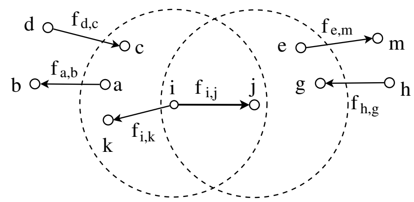

Fig. 1: Five types of contending flows of $f_{i,j}$.

Consider a MAC flow $f_{i,j}$ over a wireless link (i, j), where i is the sender and j is the receiver. Let $w_{i,j}$ be the weight of the flow. The *mean rate* of $f_{i,j}$ is defined as the flow's rate divided by its weight. The goal of PPS is to equalize the mean rates of contending flows, which is equivalent to achieving weighted bandwidth allocation.

Assume the clocks at all nodes are loosely synchronized. Time is divided into periods of length T (called *PPS periods*). The sender i of flow $f_{i,j}$ keeps a counter, denoted as $c_{i,j}$, which is initialized to zero at the beginning of each period. Within a period, the counter is increased by one for every $w_{i,j} \times l$ bits of data transmitted by flow $f_{i,j}$ over link (i, j), where l is a system-wide parameter whose impact will be discussed later. Let Δt be the time passed since the beginning of the current period. The rate of flow $f_{i,j}$ is about $\frac{c_{i,j} \times w_{i,j} \times l}{\Delta t}$. The mean rate is $\frac{c_{i,j} \times w_{i,j} \times l}{\Delta t \times w_{i,j}} = c_{i,j} \frac{l}{\Delta t}$. Hence, $c_{i,j}$ can serve as a discrete measurement of mean rate in units of $\frac{l}{\Delta t}$ (which becomes $\frac{l}{T}$ at the end of each PPS period). Therefore, if the counter values of contending flows are equalized at the end of each period, the mean rates of the flows are also equalized.

The solution for equalizing the mean rates of contending flows is simple: We always let the flow with the smallest mean rate (i.e., smallest counter value) to transmit, while letting other contending flows to wait until their mean rates become the smallest. However, in order for a MAC flow to determine whether it has the smallest counter value, the sender/receiver have to keep track of the current counter values for all contending flows. To avoid such overhead, PPS adopts an alternative solution: A MAC flow competes for media access if its sender/receiver *do not overhear* on-going transmission from another flow with a smaller or equal counter; a MAC flow refrains from accessing media if its sender/receiver *overhear* on-going transmission from another flow with a smaller or equal counter. Because flows with higher counter values refrain from accessing media, radio collision is reduced, leading to higher throughput. Below we give detailed description of PPS.

Consider an arbitrary MAC flow $f_{i,j}$, whose five types of contending flows are shown in Fig. 1, where flow $f_{a,b}$ includes the case where b is also within the transmission range of i, and flow $f_{e,m}$ includes the case where m is also within the transmission range of j. Contention in CSMA/CA occurs between two MAC flows when either the sender or the receiver of one flow is within the transmission range of either the sender or the receiver of the other flow.

Let $n_{i,j}$ be the number of bits yet to be transmitted over (i, j) before $c_{i,j}$ is increased by one. When a data packet is sent from i to j, RTS/CTS/DATA/ACK will piggyback both $c_{i,j}$ and $n_{i,j}$. Neighbors of either i or j, such as a, c, e and g, will learn this information through overhearing. Similarly, i will learn the counter value of flow $f_{a,b}$ (or $f_{d,c}$) when the flow is transmitting, and j will learn the counter value of $f_{e,m}$ (or $f_{h,g}$), as well as the number of bits yet to be transmitted before each counter is increased by one. Note that the sender i can overhear only some contending flows, and the receiver j can overhear other contending flows.

The operations of PPS are described as follows. When i is the sender of multiple flows such as $f_{i,j}$ and $f_{i,k}$, i always schedules the flow with the minimum counter. Without losing generality, let this flow be $f_{i,j}$. If i overhears that a contending flow with a smaller or equal counter is transmitting, it will refrain from accessing media. Otherwise, it will attempt to access media with RTS. After RTS is successfully delivered to the receiver j through a CSMA/CA protocol, there are two possible cases.

Case 1: Node j responds with CTS if it does not overhear transmission of another flow with a smaller or equal counter right before receiving RTS. After that, i and j exchange DATA/ACK. Node i will continue to send j a sequence of data packets (called *transmission burst*) until $c_{i,j}$ is increased by one.

Case 2: By overhearing, if j knows that a contending flow ($f_{e,m}$ or $f_{h,g}$) with a smaller or equal counter is currently in a transmission burst, j will not reply CTS and, *optionally*, it may send i a control message REJ, carrying the contending flow's counter and the number of bits to be transmitted before that counter will be increased by one. Note that REJ should be sent after j's current NAV expires in order to avoid interfering with concurrent transmissions. Based on the information received in REJ, i sets an appropriate timer and will re-attempt transmission after timeout. If REJ is not transmitted, i will perform exponential backoff.

C. Properties

When the sender of a flow is not transmitting, the flow is said to be *inactive*. When the sender of a flow is transmitting data packets, the flow is said to be *active* for a transmission burst. The transmission burst is said to be *preempted* if a contending flow stops the burst and starts its own transmission of data packets. Preemption changes the right of transmission from one flow to another. It does not mean that the preempted burst has wasted its effort; those packets in burst prior to preemption have been delivered. The length of each transmission burst is controlled by the system parameter l.

We prove two properties of PPS, base on which it can be shown that PPS achieves weighted bandwidth allocation (more precisely, weighted maxmin fairness) among MAC flows with an error that can be made arbitrarily small. Here we define a *channel* as a maximum set of mutually-contending backlogged flows. Since it is possible for a flow to form different maximum mutually-contending sets with its nearby flows, it may belong to multiple channels, among which the one that sets the tightest limit on the flow's rate is called its bottleneck channel.

Property 1: An active flow will be preempted if a backlogged contending flow in the same bottleneck channel has a smaller counter value.

Proof: By the design of PPS, the sender/receiver of a contending flow with a smaller counter will attempt to access media and perform RTS/CTS/DATA/ACK exchange when they learn that the current transmission piggybacks a larger counter. Once the current flow is preempted, it will not be able to transmit until the preempting flow's counter is increased. That is because its sender/receiver will overhear transmissions piggybacking the smaller counter of the preempting flow, which prevents it from completing RTS/CTS exchange. □

Property 2: The channel will not be left idle if there exists a backlogged flow (with a non-empty packet queue).

Proof: When the channel becomes idle, the sender of the flow with a non-empty queue certainly does not overhear any contending flow with a smaller or equal counter is transmitting. By the design of PPS, the sender will attempt to access the media and the channel will not be left idle. □

Among several equivalent definitions of weighted maxmin fairness, one states that each flow has a *bottleneck* (saturated local channel) that constrained its rate and all flows *constrained by* the same bottleneck should have the same mean rate (which means those flows' rates are proportional to their weights). We say a packet scheduling protocol achieves *ϵ-approximation of weighted maxmin fairness* if the mean rates of any two MAC flows that are constrained by the same bottleneck channel differ by no more than ϵ.

Theorem 1: PPS achieves ϵ-approximation of weighted maxmin fairness, for $\epsilon = \frac{l}{T}$.

Proof: Consider backlogged flows. Property 2 ensures that the flows will fully utilize channel capacity and their saturated local channels become bottleneck, limiting the rates of the flows from further increase. Property 1 ensures that the counter values of all MAC flows in the same bottleneck channel will not differ by more than one. The reason is that, once a flow f_1's counter is greater than another flow f_2's counter by one, f_2 will preempt f_1's transmission and prevent f_1's counter from further increasing until f_2's counter is increased. We know that the counter value of a flow represents the flow's mean rate in units of $\frac{l}{T}$ at the end of each PPS period. Because the maximum difference between the counters of contending flows is bounded by one, the maximum difference between the mean rates of the flows must be bounded by $\frac{l}{T}$. □

Note that $\frac{l}{T}$ can be made arbitrarily small if we increase the length of the PPS period T.

IV. Performance Evaluation

In this section, we evaluate the performance of the proposed protocols through extensive simulations on ns2 [28].

A. Simulation Setup

We implemented PPS (proportional packet scheduling) and DWA (dynamic weight adaptation) on ns2 v2.32. They are implemented on top of IEEE 802.11 DCF provided by ns2. For comparison, we also implemented an influential packet scheduling protocol [22] (referred as AdjConWin) and IEEE 802.11e EDCA. AdjConWin achieves fairness among MAC flows by dynamically adjusting their minimum contention windows. IEEE 802.11e EDCA has four access categories: background, best effort, video, and voice.

We first evaluate DWA and show how well it achieves rate assurance and bandwidth differentiation. We then evaluate the performance of PPS and compare it with other protocols. If not specified otherwise, the default simulation parameters are given as follows: The transmission rate is set to be 11Mbps based on IEEE 802.11b, and each packet is 1000 bytes long. The PPS period is 2 seconds. The parameter l is set to be the length of five packets. The weight-adaptation period is 2 seconds. There are two QoS service classes for the lower priority 1 and the higher priority 2. Their differentiating factors are $d_1 = 2$ and $d_2 = 4$, respectively. β is 10%. Besides what have been stated above, other parameters (such that those for 802.11 DCF) use the default values set by ns2 according to the protocol standards.

B. Prioritized Rate Assurance and Bandwidth Differentiation

We evaluate DWA (which is implemented on top of PPS) using the network topology in Fig. 2 with a dynamic set of flows, whose rate requirements and priorities change over time. The network consists of 30 nodes that are randomly deployed in an area. The wireless links formed between nodes are shown in the figure, where the average degree of a node is 4.4. The diameter of the network is 7 hops. There are 24 multihop end-to-end flows, which cannot be shown in the figure. The source/destination nodes of each flow are randomly chosen. We perform simulations under four different settings. The results are shown in Fig. 3-9.

Setting A: *Flows 0-7 are assigned to the best-effort service class, flows 8-15 to the QoS service class of priority 1, and flows 16-23 to the QoS service class of priority 2. All QoS flows have the same rate requirement of 10 pps (packets per second).*

We first turn off DWA. The flow rates under IEEE 802.11 DCF are shown in Fig. 3. Flow rate is measured in the number of packets successfully transmitted per second (pps). The contention levels experienced by the randomly-generated flows are vastly different. Without additional mechanisms to compensate such difference, the flow rates achieved under 802.11 are quite unpredictable with some much higher than others.

We then turn on DWA. The simulation result is shown in Fig. 4. The network is able to satisfy the rate requirements of all QoS flows. Flows 4, 6 and 16 have much higher rates than others because their routing paths happen to have less contention with other flows. The rates of all QoS flows vary from one to another also because the flows experience different levels of contention on their paths. The variation among the rates of best-effort flows are due to the same reason.

Next we create 8 new flows in each service class. For new QoS flows, their rate requirements are still 10 pps. The result is shown in Fig. 5. For easy comparison, we reassigned flow ids such that the rates of best-effort flows are shown under ids 0-15, the rates of priority-1 flows are shown under 16-31, and the rates of priority-2 flows are shown under 32-47. After doubling the number of flows, the rate requirements of priority-2 flows can still be met, but those of priority-1 flows can no longer be met. The rates of priority-1 flows are not identical because their routing paths contend with different sets of other flows.

Finally, we let the new flows depart from the network, and the flow rates go back to Fig. 4.

Setting B: *It is the same as Setting A except that we increase the rate requirements of all QoS flows to 20 pps, so that not all of them can be satisfied.*

When DWA is turned on, the simulation result is shown in Fig. 6. The rate requirements of flows 16-23 (priority 2) are satisfied or nearly satisfied. Some of them receive slightly lower rates due to more intense contentions. The network cannot support the rate requirements of flows 8-15 (priority 1) even when their weights are adapted to the ceilings. Due to lower priority, their ceilings are lower than those of flows 16-23. But they have higher rates with DWA than without it (Fig. 3). The best-effort flows receive the remaining network bandwidth. By design, no best-effort flow is starved.

For the purpose of comparison, we perform the same simulation under IEEE 802.11e EDCA, with priority-2 flows assigned to the video access category, priority-1 flows assigned to the best-effort access category, and prority-0 flows assigned to the background category. The results are shown in Fig. 7, which shows that flows 16-23 (priority 2) takes network bandwidth aggressively, starving the other flows. EDCA gives fixed preference to higher-priority flows and lacks a fine-level rate control mechanism that not only allocates bandwidth based on priorities but also balance the minimum needs among all QoS flows.

The next two settings are designed to demonstrate the great flexibility of DWA in controlling the bandwidth distribution among end-to-end flows.

Setting C: *Keeping the rate requirements to be 20 pps, we now reassign the flow priorities. Let flows 8-15 be priority 2, flows 16-23 be priority 1, and flows 0-7 be best-effort.* The simulation result is shown in Fig. 8. Now the rate requirements of flows 8-15 (priority 2) are satisfied.

Setting D: *We again reassign the flow priorities. Let flows*

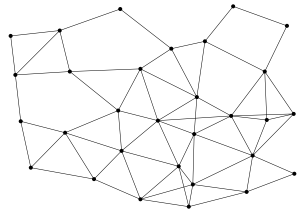

Fig. 2: Network topology

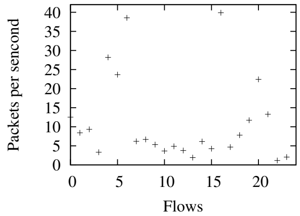

Fig. 3: Setting A, IEEE 802.11 DCF

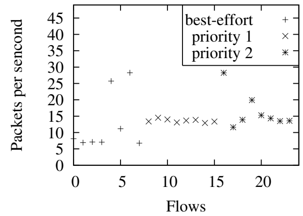

Fig. 4: Setting A, DWA

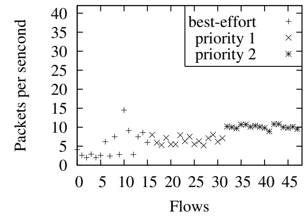

Fig. 5: Setting A, DWA, doubling the number of flows

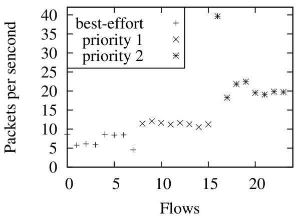

Fig. 6: Setting B, DWA

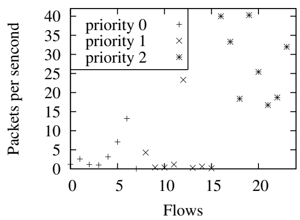

Fig. 7: Setting B, EDCA

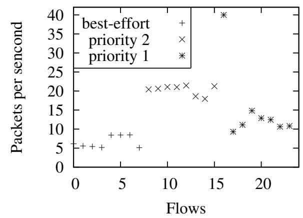

Fig. 8: Setting C, DWA

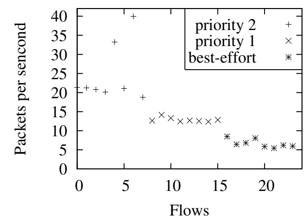

Fig. 9: Setting D, DWA

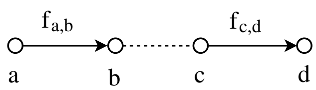

Fig. 10: Two-flow topology

TABLE I: Flow rates (in packets per second) on the two-flow topology

	$f_{a,b}$	$f_{c,d}$
802.11 DCF	64.6	381.0
802.11e EDCA	347.8	193.1
AdjConWin	232.0	229.5
PPS	227.9	228.8

0-7 be priority 2, flows 8-15 be priority 1, and flows 16-23 be best-effort. The simulation result is shown in Fig. 9. The rate requirements of the new priority-2 flows, whose ids are 0-7, are satisfied.

C. Performance of PPS

We give further evaluation on the performance of PPS at the MAC layer, in comparison with IEEE 802.11 DCF, IEEE 802.11e EDCA, and AdjConWin. Since AdjConWin is designed to achieve fairness, in order to make the simulation results comparable, we set the weights of all flows in PPS to be one. We first perform simulations on the simple topology in Fig. 10, where a is not in the transmission range of c, but b is. Suppose two MAC flows, $f_{a,b}$ and $f_{c,d}$, are both backlogged. Under IEEE 802.11 DCF, it is well known that severe unfairness can happen in this scenario because $f_{a,b}$ can hardly acquire the channel [22], [25]. We run simulations for fifty seconds under DCF, EDCA, AdjConWin, and PPS, respectively. The flows' rates are still measured in packets per second. The results are shown in Table I. Clearly, DCF causes severe unfairness, with $f_{c,d}$ acquiring most of the channel capacity and $f_{a,b}$ receiving little. For EDCA, we assign $f_{a,b}$ to the video access category and $f_{c,d}$ to the best-effort category in order to give $f_{a,b}$ more bandwidth, which however causes unfairness the other way around. Other category assignments will also lead to unfairness, which is understandable because EDCA is not designed for fairness or weighted bandwidth allocation. Both AdjConWin and PPS can achieve fairness between the two flows.

There are some important differences between AdjConWin and PPS, which are elaborated in Section III-A. Because of its fully localized operations, PPS is equally effective under dynamic setting where the set of MAC flows, as well as the weights of the flows, change over time. This is critical for DWA because a MAC flow for a service class on a wireless link will be inactive when its queue is empty and become active again when its queue is backlogged, which happens when end-to-end flows join or depart from the network. Unlike PPS, AdjConWin is less effective with a dynamic set of MAC flows because it requires each node to centrally compute the local contention cliques and the fair bandwidth shares for its own flows as well as nearby contending flows, under the assumption that each clique has an equal, fixed capacity. The senders of contending flows can be three hops away. In a dynamic environment, the overhead will be very high if all nodes constantly exchange their current flow information in order to update the correct values for fair bandwidth shares. If such update is not done, the network performance suffers. We perform simulations on the five-flow topology in Fig. 11, where each dashed ellipse contains nodes can hear each other's transmission. Suppose flow 1's queue is empty from time 10 to 20, flow 2's queue is empty from time 20 to 30, flow 3's queue is empty from time 30 to 40, flow 5's queue is empty from time 40 to 50, and all queues are otherwise backlogged. Fig. 12 and 13 show the flow rates under AdjConWin and PPS, respectively. The average flow rates under PPS are much higher because AdjConWin requires close coordination among contending nodes and such coordination breaks down with

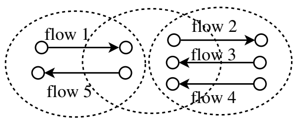

Fig. 11: Five-flow topology

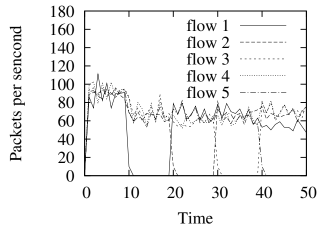

Fig. 12: AdjConWin

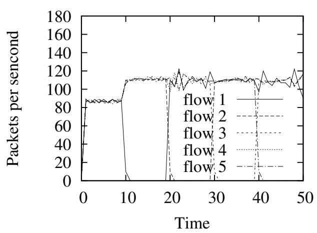

Fig. 13: PPS

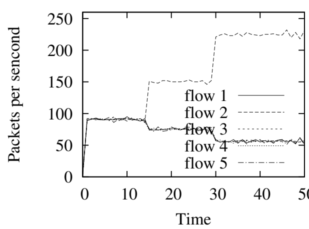

Fig. 14: PPS supports dynamic weight

a dynamic set of flows, while PPS relies on fully localized operations.

To support DWA, each MAC flow must be allowed to *independently and locally* change its weight. Yet, without explicit coordination among contending flows, the bandwidth allocation must follow such changes to maintain the invariant that the flow rates are proportional to their current weights. We perform a simulation on the topology of Fig. 11, where all flows begin with weight one and flow 1 changes its weight to 2 at time 15 and then to 4 at time 30. The result in Fig. 14 shows that PPS maintains weighted bandwidth allocation under dynamic weight. AdjConWin does not consider flow weights, let alone dynamic weight (which requires fully localized operations).

V. Conclusion

This paper presents a new adaptive rate control function based on two novel protocols, dynamic weight adaptation and proportional packet scheduling, which together enable prioritized rate assurance and sophisticated bandwidth differentiation among end-to-end flows in multihop wireless networks. The new adaptive function represents a paradigm change in fine-level bandwidth management without resource reservation and admission control.

VI. Acknowledgments

This work was supported in part by the US National Science Foundation under grant CNS-0644033 and grant CNS-0721731.

References

[1] A. Banchs and X. Perez, “Providing Throughput Guarantees in IEEE 802.11 Wireless LANs,” *Proc. IEEE WCNC*, 2002.

[2] S. Shah, K. Chen, and K. Nahrstedt, “Dynamic Bandwidth Management in Single-Hop Ad Hoc Wireless Networks,” *Proc. IEEE PERCOM*, 2003.

[3] C.-T. Chou and K. G. Shin, “Analysis of Adaptive Bandwidth Allocation in Wireless Networks with Multilevel Degradable Quality of Service,” *IEEE Trans. on M. Comp.*, vol. 3, no. 1, 2004.

[4] P. Gupta, Y. Sankarasubramaniam, and A. L. Stolyar, “Random-Access Scheduling with Service Differentiation in Wireless Networks,” *Proc. IEEE INFOCOM*, 2005.

[5] S. Sarkar and L. Tassiulas, “End-to-end Bandwidth Guarantees Through Fair Local Spectrum Share in Wireless Ad-hoc Networks,” *Proc. IEEE Conference on Decision and Control*, 2003.

[6] A. Veres, A. Campbell, M. Barry, and L. Sun, “Supporting Service Differentiation in Wireless Packet Networks Using Distributed Control,” *IEEE JSAC, Vol. 19, No. 10*, 2001.

[7] S. Kang and M. Mutka, “Provisioning Service Differentiation in Ad Hoc Networks by The Modification of Backoff Algorithm,” *Proc. IEEE ICCCN*, 2001.

[8] G. Ahn, A. Campbell, A. Veres, and L. Sun, “SWAN: Service Differentiation in Stateless Wireless Ad Hoc Networks,” *Proc. IEEE INFOCOM*, 2002.

[9] K. Karenos, V. Kalogeraki, and S. V. Krishnamurthy, “A Rate Control Framework for Supporting Multiple Classes of Traffic in Sensor Networks,” *Proc. IEEE Real-Time Systems Symposium*, 2005.

[10] Y. Yang and R. Kravets, “Contention-Aware Admission Control for Ad Hoc Networks,” *IEEE Transactions on Mobile Computing*, vol. 4, no. 4, 2005.

[11] M. Kazantzidis, M. Gerla, and S.-J. Lee, “Permissible Throughput Network Feedback for Adaptive Multimedia in AODV MANETs,” *Proc. of IEEE ICC*, 2001.

[12] G. Bianchi, “Performance Analysis of the IEEE 802.11 Distributed Coordination Function,” *IEEE JSAC*, vol. 18, no. 3, 2000.

[13] F. Cali, M. Conti, and E. Gregori, “Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit,” *IEEE/ACM Transactions on Networking*, vol. 8, no. 6, 2000.

[14] B. Li and R. Battiti, “Performance Analysis of An Enhanced IEEE 802.11 Distributed Coordination Function Supporting Service Differentiation,” *Int’l Workshop on Quality of Future Internet Service*, 2003.

[15] Y. Yang and R. Kravets, “Throughput Guarantees for Multi-priority Traffic in Ad Hoc Networks,” *Elsevier Ad Hoc Networks Journal*, vol. 5, no. 2, 2007.

[16] F. Kelly, A. Maulloo, and D. Tan, “Rate control in communication networks: shadow prices, proportional fairness and stability,” *Journal of the Operational Research*, vol. 49, 1998.

[17] H. Zhang, “Service Disciplines for Guaranteed Performance Service Inpacket-Switching Networks,” *Proc. of the IEEE*, vol. 83, no. 10, 1995.

[18] S. Chen and N. Yang, “Congestion Avoidance based on Light-Weight Buffer Management in Sensor Networks,” *IEEE Transactions on Parallel and Distributed Systems, Special Issue on Localized Communication and Topology Protocols for Ad Hoc Networks*, vol. 17, no. 9, Sep 2006.

[19] B. Li, “End-to-End Fair Bandwidth Allocation in Multi-Hop Wireless Ad Hoc Networks,” *Proc. IEEE ICDCS*, 2005.

[20] H. Luo, S. Lu, and V. Bhurghawn., “A New Model for Packet Scheduling in Multihop Wireless Networks,” *Proc. ACM MOBICOM*, 2000.

[21] B. Bensaou, Y. Wang, and C. Ko, “Fair Medium Access in 802.11 Based Wireless Ad Hoc Networks,” *Proc. ACM MOBIHOC*, 2000.

[22] X. Huang and B. Bensaou, “On Max-Min Fairness and Scheduling in Wireless Ad Hoc Networks: Analytical Framework and Implementation,” *Proc. ACM MOBIHOC*, 2001.

[23] H. Luo, J. Cheng, and S. Lu, “Self-Coordinating Localized Fair Queueing in Wireless Ad Hoc Networks,” *IEEE Trans. on M. Comp.*, vol. 3, no. 1, 2004.

[24] S. Chen and Z. Zhang, “Localized Algorithm for Aggregate Fairness in Wireless Sensor Networks,” *Proc. ACM Mobicom*, 2006.

[25] A. Rao and I. Stoica, “An Overlay Mac Layer for 802.11 Networks,” *Proc. ACM MOBISYS*, 2005.

[26] L. Tassiulas and S. Sarkar, “Maxmin Fair Scheduling in Wireless Networks,” *Proc. IEEE INFOCOM*, 2002.

[27] N. H. Vaidya, P. Bahl, and S. Gupta, “Distributed fair scheduling in a wireless LAN,” *Proc. ACM MOBICOM*, 2000.

[28] “The Network Simulator - ns-2,” *http://www.isi.edu/nsnam/ns/*.

1-4244-2507-5/08/$20.00 ©2008 IEEE