

Experimental Evaluation of MAC Protocols for Fairness and QoS Support in Wireless Networks

Ajit Warriar, Injong Rhee
Dept of Computer Science
North Carolina State University
acwarrie,rhee@ncsu.edu

Jae H. Kim
Boeing Phantom Works
jae.h.kim@boeing.com

Abstract—Existing wireless MACs are known to have fairness and QoS problems¹. In response, the wireless community have come up with numerous “point solutions”, with one-to-one comparison of their solution with existing MACs for the specific fairness model they target. However, with some tradeoff of flexibility and efficiency, many of these MACs can be made relatively free of a specific fairness model. This decoupling of the MAC from the fairness model is desirable because it allows objective testing of the MAC under different fairness models and also allows independent evolution of fairness policies and research into medium access issues. However, there is no cross-model evaluation of such MACs especially in the context of real multihop wireless testbeds. In this paper we address this issue by testing three MACs – 802.11e, EY-NPMA and DWOP, (which represent different ways of enabling fairness in the MAC) across a number of fairness and QoS models – temporal/rate and proportional fairness, and EDF and static priority scheduling. In the process we also extend EY-NPMA significantly (which we call Siren) for multihop wireless environments. We perform these comparisons on a 30 node multihop wireless sensor testbed comprising of MicaZ devices equipped with the CC2420 radio.

I. INTRODUCTION

One of the frequently cited problems of current wireless networks is related to lack of fairness and QoS. The reported problems of MAC include, to name a few, starvation [26], [9], [31], priority inversion [32], inequitable allocation of bandwidth [31], lack of QoS support [16] and multi-rate LAN unfairness [12]. The research community has come up with several novel solutions [26], [32], [16] to such problems. The common thread among such solutions is that they often require altering the MAC protocol significantly to achieve the authors’ notion of “fairness”, in effect embedding a fairness policy into MAC. Clearly, this makes them “point solutions”, wherein the proposed changes to MAC proposed for one solution are incompatible with those for other solutions. To address this issue, Nandagopal et al. [23] propose a general analytical framework to allow automatic generation of MAC layer algorithms tailored to specific utility functions. In addition, extensive prior work [8], [20] on cross-layer optimization optimizes MAC functions to support a specific predefined utility function. However, note that each specific instance of a

¹We consider QoS to be a form of a user-defined notion of fairness under heterogeneous environments e.g. a particular policy may enforce the rule that VoIP flows should always have higher priority compared to best effort flows like TCP, so even though at the flow level this policy is inherently “unfair”, from the user’s perspective it is “fair”. We will henceforth use fairness and QoS interchangeably in the paper.

MAC algorithm is still specific to a fairness policy (or a utility function) used to generate it, and may not be used to support other fairness policies.

We perceive fairness to be an application-defined property governed strictly by the utility of the application. In an ideal wireless infrastructure, MAC simply does what it does best – medium contention, while exporting enough control “knobs” to the application so that it may enforce its own fairness policy. This *application-centric* notion of fairness follows directly from the well-known system design principle to separate mechanisms from policies. This decoupling between the fairness policy and fairness mechanism is highly desirable since (1) it allows independent evolution of fairness policies and research into medium contention mechanisms and (2) it allows objective testing of MAC under different fairness models. In fact, many of existing MACs do exhibit such decoupling, albeit with varying degrees of flexibility and efficiency. This brings up an important question – how do these different MAC protocols measure up in the support of different fairness policies, and in the process how much do we lose in terms of flexibility and efficiency? To the best of our knowledge, this question has remained unanswered so far especially in the context of a real-world multihop wireless networks.

One main reason behind this lack of a comprehensive evaluation study is that most of the cited solutions [26], [32], [16], [17] require significant changes to MAC layer which were not possible to be implemented on current commercial radios. This is because efficiency considerations make most of the MAC layer functions in current radios be performed in firmware which is difficult to modify. This limitation forces even the one-to-one comparison of most cited solutions with existing work for a specific fairness policy to be limited to simulation. In the process significant real-world issues such as system latency, ambient channel noise, time synchronization issues, wireless channel errors, specific radio limitations such as minimum/maximum packet sizes etc. are simply glossed over. To some extent, this problem has been mitigated by the development of Software Defined Radios (SDRs) [2], which allow all components of the radio (including PHY modulation) to be specified in software. However the prohibitive per-unit costs of SDR radios make it difficult to be used for large testbed deployments. In this paper, we address this issue by conducting an extensive experimental evaluation of existing MACs which provide fairness support in a real multihop

wireless testbed environment.

Our methodology is as follows. We first survey existing work on fairness issues in single/multihop wireless environments and their corresponding proposed solutions. A common technique to construct mechanisms for implementing various fairness policies is to (1) assign priorities to data packets, (2) manipulate these priorities dynamically to implement a specific policy dictated by applications, and (3) ensure channel access to be ordered in terms of the priorities. Several schemes [17], [16], [27], [29] follow this approach where fairness mechanisms are supported by *priority resolution* (PR) – the mechanisms to implement the prioritized access of packets. We classify such MACs supporting PR into three categories based on the way they perform PR, and identify three representative schemes for each of the three categories.

- 1) Differentiated backoff based; represented by 802.11e [3], [18].
- 2) Beacon-based; represented by EY-NPMA [28], [32], [6].
- 3) Scheduling-based; represented by DWOP [16], [17].

This categorization significantly reduces the number of combinations that need to be evaluated without compromising the breadth of our study. We implement these three schemes on the ZigBee-based CC2420 radio on the MicaZ sensor platform. This radio is different from contemporary 802.11 radios in that it exposes a considerable amount of functionality (e.g. clear channel assessment (CCA)) to software while still maintaining a raw bitrate of 250Kbps. The implementation of the above MACs on such a radio for multihop environments is a significant engineering and design challenge since it is important to make sure that our design decisions do not affect the experimental results and that the conclusions are valid not just for the CC2420 radio but also for other CSMA-based radios. The implementation of EY-NPMA for multi-hop environments is particularly difficult as it is designed only for wireless LAN environments and thus does not provide any mechanisms for hidden terminals. We extend EY-NPMA for multihop environments and call our extension *Siren*. In the next step, we identify and implement a list of commonly used fairness policies including static priority, temporal fairness, proportional fairness and earliest deadline first (EDF) on top of these three MAC protocols. Finally we compare the performance of these MAC protocols in enforcing these fairness schemes on our 30 node multihop wireless testbed.

The rest of the paper is as follows. In Section 2 we describe the three MAC protocols we implement, while in Section 3 we present implementation details on the CC2420 radio. In Section 4 we describe the fairness schemes we use for comparison. Section 5 contains experimental results. Section 6 presents related work and we conclude in Section 7.

II. MAC MECHANISMS FOR FAIRNESS AND QOS

We classify MAC schemes into three categories based on the manner in which they perform PR – (a) differentiated backoff-based schemes, (b) beacon-based schemes, and (c) scheduling based schemes. In this section, we describe these categories and their representative schemes used for our evaluation.

A. Differentiated backoff-based schemes

In such schemes, nodes access the channel with a backoff-period which is a function of the priority of their head-of-line (HOL) packet, e.g. a node *A* with higher priority² will access the channel with a shorter backoff compared to neighboring node *B* with lower priority. This ensures that *A* captures the channel with high probability compared to *B*, with the exact probability of channel capture being a function of the relative backoff periods of *A* and *B*. IEEE 802.11e and its numerous variants [3], [18] fall in this category. Note that the original standard defines 4 priority levels with varying but overlapping backoff periods for each priority. This allowed lower priority nodes to sometimes capture the channel in the presence of high priority nodes, or priority inversion [32]. In order to achieve *absolute prioritization*, where the higher priority nodes always capture the channel earlier than lower priority nodes, Aad et al. [7] propose having non-overlapping backoff periods for each priority, i.e. $DIFS_i = DIFS_{i-1} + CW_{i-1}$, where CW_i is the backoff window for priority level *i*. We use this version of 802.11e for our evaluation.

B. Beacon-based schemes

In beacon-based schemes, nodes rely on bursts of energy to inform their neighbors of their priority. The actual priority is encoded in the length of the beacon (e.g. black burst [28], prioMAC [25]) or the time (e.g. EY-NPMA [6]) at which they are sent. A subset of such schemes also rely on an additional channel to send beacons, (e.g. BTPS [32]). We focus our evaluation on EY-NPMA which uses short beacons sent at pre-defined moments in time to encode priority information. The reasoning for choosing EY-NPMA is twofold. First, encoding the priority in the length of the beacon wastes energy. Second, using a separate channel for beacon transmissions is not possible in contemporary single transceiver radios.

In EY-NPMA, each node goes through a PR phase followed by a contention resolution (CR) phase. The PR phase is as follows. As soon as the channel becomes idle, a node with a back-logged packet with priority *i* waits for *i beacon periods* which is defined to be the time to fully transmit a short *beacon*. At the end of *i beacon periods*, if the channel is idle, the node transmits the beacon and enters contention for the medium. However, if it hears any beacon transmission prior to the *i beacon periods*, then it recognizes that a higher priority transmission is pending and defers its own transmission. If there exist more than one node with the same highest priority, they will all transmit beacons simultaneously and will enter the CR phase together. At the end of the PR phase, one or more nodes with the highest priority are left in contention, and they contend for the medium in the CR phase.

C. Scheduling-based schemes

In scheduling-based schemes, nodes disseminate priority information well before the actual packet transmissions and

²For brevity, we will abuse terminology and refer to "nodes with HOL packet priority *i*" as "nodes with priority *i*".

then co-ordinate their transmissions in such a way that the order of packet transmissions matches the advertised priorities. Distributed priority scheduling [16] and DWOP [17] follow this scheme. In both schemes, nodes piggyback their packet priorities onto RTS, CTS, DATA and ACK messages, which are overheard by neighbors. Thus each node builds a database of priorities of all its neighbors within a two-hop communication range. RTS and CTS messages of a node inform its immediate neighbors and potential two-hop interferers respectively about its priority. DATA and ACK messages inform the neighbors that the packet has been successfully transmitted so that they may update their database. In addition, DATA and ACK messages also inform neighbors about the priority of the next packet in its queue.

In distributed priority scheduling [16], the backoff period of a node is calculated based on its priority relative to other priorities in the priority database. DWOP [17] is similar, except that it is stricter in its enforcement of packet priorities where nodes are not allowed to send an RTS message unless the maximum priority level in their cache corresponds to their HOL packet. We choose DWOP as our representative scheme due to this priority resolution mechanism as it provides the absolute prioritization property seen in our version of 802.11e (Section II-A) and EY-NPMA. As we shall see later in Section IV, absolute prioritization allows a simplified implementation of various fairness policies.

III. MAC MODIFICATIONS FOR ZIGBEE AND MULTIHOP ENVIRONMENTS

We modify 802.11e, EY-NPMA and DWOP MACs to use them for multihop wireless environments under the CC2420 radio. In this section we describe how each protocol is modified. Although the details are to some extent CC2420-specific, we believe that similar issues appear for other CSMA-based radios like commercial off-the-shelf 802.11 radios. Before going into specific details for each MAC, we first give a brief description of the CC2420 transceiver.

A. CC2420 Transceiver – Overview

The MicaZ [5] sensor node contains the CC2420 [1] transceiver which is a ZigBee [4] compatible radio. It has a link speed of 250Kbps, resulting in a per-byte transmission time of 32 μ s. It supports 8 different transmission power levels ranging from a minimum of -25 dBm ($< 10\mu$ W) to a maximum of 0 dBm (1 mW). It contains two buffers of size 128 bytes each for outgoing and incoming packets respectively. Packets placed in the outgoing buffer need to strictly adhere to the ZigBee packet format, else they are dropped.

The ZigBee [4] standard specifies the minimum PHY layer header of 5 bytes (4 bytes preamble and 1 byte synchronization), a MAC layer header of 9 bytes and 2 bytes of CRC. This implies that the minimum packet size with no payload in CC2420 is 16 bytes. Also, the standard specifies that the maximum packet size (including PHY, MAC headers, CRC and payload) is restricted to 128 bytes.

B. Large Packet Size Emulation

The maximum packet size of CC2420 is 128 bytes which is highly restrictive compared to commercial 802.11 radios with MTU of 1500 bytes. To overcome this limitation, we maintain a *virtual packet* at the MAC layer in which the transmission of larger packets is emulated. The size of a virtual packet is set to X times 128 bytes. For every data packet of size 128 bytes received from the upper layers, the MAC driver retransmits the same packet X number of times. The receiver counts the number of CRC-valid copies of a packet it receives. If it receives X copies of the same packet, it will send ACK. Otherwise it will silently discard all copies of the packet. The sender, after sending X copies, waits for ACK. If it does not receive ACK, it will signal a failure to the upper layer, and try again. This emulates closely the behavior of a packet of size $128 \times X$ bytes. In our experiment, we set X to 3 for an emulated packet size of 384 bytes.

C. 802.11e

IEEE 802.11e pauses backoff timers when the channel becomes busy and resumes them again when the channel becomes idle. This is difficult to implement in CC2420. When transmitting a packet, the MAC driver first loads the packet into the CC240 transmit buffer. Once the buffer is loaded, transmission is initiated by an explicit command (STXONCCA command) to the hardware. After that, the radio checks the noise level on the channel (CCA or Clear Channel Assessment) for 4 byte-times or 128 μ s. If the CCA detects that the channel is idle, the packet is put on the channel byte by byte. If not, the radio reports an error and takes an appropriate error handling action. Hence the radio does not perform the CCA continuously in the transmit mode. This makes it difficult to pause and resume backoff timers based on channel activity.

To get over this limitation, we do not follow the 802.11e standard of pausing backoff timers. In our implementation, which is called PMAC, when a node has a packet to transmit, it takes a random backoff. During the backoff period it does not check the channel. At the end of the backoff period, it will issue the STXONCCA command to the PHY. If it receives an error because the channel is busy, the node will take another random backoff. Otherwise the PHY will begin transmission.

PMAC uses non-overlapping backoff windows to support absolute prioritization. Despite this, it is still possible to have priority inversions if the backoff window is paused. To see why, consider two nodes A and B , with A having a higher priority than B . A will naturally use a shorter backoff and will access the channel multiple times. However, during each channel access, there is some small channel idle time during A 's backoff period. During this period, B 's backoff timer keeps counting down since the timer resumes during channel idle times and pauses during channel busy times. Hence eventually B will access the channel with a shorter backoff than A causing a priority inversion. Thus by not pausing backoff timers and forcing B to take a new backoff every time, PMAC completely avoids priority inversions.

A second issue with 802.11e is the use of exponentially increasing backoff timers. In the standard, if a node's backoff timer expires, it checks the medium and if it is still busy, it doubles the backoff window and takes another random backoff within this window. However, PMAC needs to support several priority levels and supporting multiple exponential increases in the backoff window leads to large backoffs for each priority level. This induces high overhead for each priority level. As we shall see in Section III-F, even while using a fixed window without exponential backoffs, PMAC still incurs high overhead since every priority level requires one backoff window. Hence we choose to use fixed backoff windows for PMAC.

D. EY-NPMA

EY-NPMA was intended for a single-hop WLAN environment where every node can sense the transmissions of its neighbors, and hence can enter the PR and CR phases in synchrony. To enable this behavior for multi-hop environments we require time-synchronization among neighboring nodes. We refer to this modified protocol as *Siren* to differentiate it from EY-NPMA for the rest of this section. In *Siren*, time is slotted.

Note that the overhead incurred in the PR phase of EY-NPMA is the time taken for a beacon transmission times the number of priority levels supported. Clearly, a smaller beacon size incurs less overhead. Early radios (e.g. CC1000 [1]) support bit streaming which can be used to construct extremely small beacons. However the new generation of packetizing radios (e.g. CC2420 [1], most commercial 802.11 radios) does not support bit streaming and allows only valid packets with standard compatible headers to be transmitted. This limitation significantly increases the overhead of the priority resolution phase if the minimum packet size is large. *Siren* overcomes this limitation by the use of *beacon sensing*. The basic idea is that a node does not need to completely receive a beacon to sense a higher priority node in its vicinity; it only needs to detect the presence of the beacon on the channel through CSMA. We now discuss this concept in more detail.

1) *Beacon Sensing*: A node with a priority i waits for i CCA times which is the minimum time to sense a (beacon) transmission from a neighbor. At the end of the i -th CCA time, if the channel is idle, the node transmits the beacon and enters the CR phase. However, if it senses any beacon before the i CCA times, then it defers its transmission to the next time slot. If there exist multiple nodes with the same priority, they all transmit beacons simultaneously and enter the CR phase together. As illustrated in Figure 1 (A), the overhead for the PR phase in *Siren* is the CCA time multiplied by the number of priority levels plus one beacon transmission time. Since the CCA time for most radios is much smaller than the minimum valid packet transmission time, beacon sensing results in a significant overhead saving. The PR and CR phases of *Siren* in each timeslot is shown in Figure 1 (B).

The minimum packet size limitation of 16 bytes corresponds to a beacon transmission time of $16 \times 32 = 512\mu\text{s}$. By means of experiments, we found out that CC2420 requires sensing

the medium for 10 byte-times = $320\mu\text{s}$ to sense any activity on the channel. Hence we use a beacon sensing time of $320\mu\text{s}$ for our implementation. Unless otherwise specified, we use 6 priority levels for *Siren* incurring an overhead of $320 \times 6 + 512\mu\text{s} = 2.4\text{ms}$ per packet transmission for priority resolution.

2) *Beacon Transmission Power*: In *Siren*, a node uses beacons to inform its neighbors about its priority. However, hidden terminals located at two hops away from the node, and at one hop away from its intended receiver may not sense the beacon and may interfere with packet reception at the receiver [26], [9], as illustrated in Figure 1 (C). This problem is not handled in EY-NPMA.

To fix this problem, we adopt a higher transmission power for beacon transmissions compared to data transmissions to preserve the correctness of PR across two hops. The transmission power for beacons should be enough to reach all potential interfering nodes for A's intended receiver. This approach has been used in [15], [30] as well. In our study, we set the beacon transmission power of all nodes to be roughly three times that of their data transmission power for a simplified implementation. This is decided based on the following reasoning. Assuming free-space propagation, where the transmitted signal drops by square root of distance, quadrupling transmission power for beacons corresponds to doubling distance, hence ensuring that the beacon is received by all two-hop interferers. However, since beacon packets need to be only sensed, not received, we find through experiment that simply using three times the data transmission power works for most cases.

Using the same beacon transmission power for all nodes in the network is a conservative approach which may cause the beacon transmissions of some nodes to reach farther away from interfering nodes and shut down non-interfering nodes from transmission leading to loss of network capacity. This problem is called the *exposed terminal* problem. An example is shown in Figure 1 (C). Node H will be shut down by A's beacon, even though its transmission will not interfere with A's transmission to B . Recent work has established that it is possible to detect the interference relations between nodes on a run-time basis [34]. In addition, transmission power can also be tuned so that transmissions reach only a set of interfering nodes [19]. Hence it is possible to tune the beacon transmission power on a per-node basis, so that it reaches the set of interfering nodes and no farther. This will not eliminate all cases of exposed terminals, but will definitely result in improvement in network capacity. However, we do not explore this option in this paper, and leave it for future work. Our experimental results presented in Section V are obtained without this tuning. We use -5 dBm ($316\mu\text{W}$) transmit power for data packets, and 0 dBm (1 mW) for beacons.

3) *Time Synchronization*: *Siren* does not require global clock synchronization. The protocol works as long as the sender is synchronized with all potential interferers (both one and two-hop nodes). Hence we use a conventional distributed hop-by-hop clock synchronization algorithm – FTSP [21].

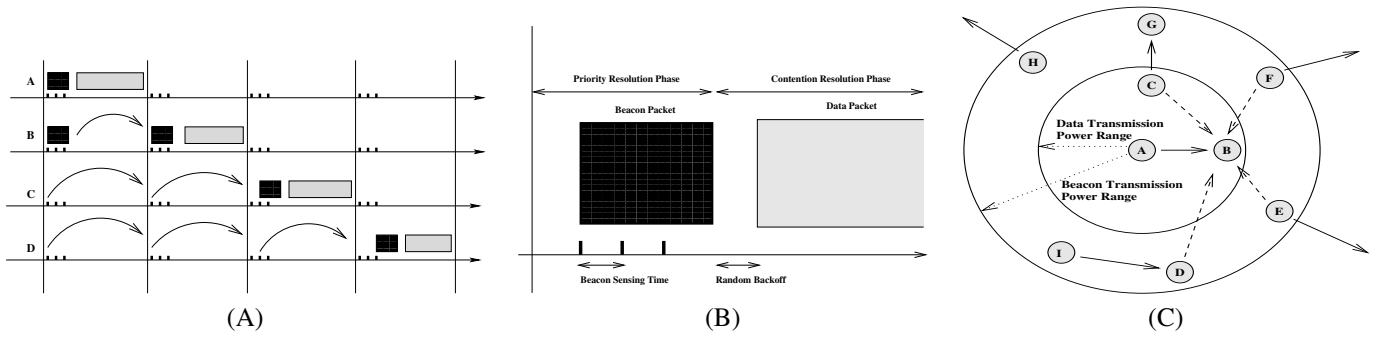


Fig. 1. (A) shows the working of Siren for nodes A , B , C and D with priorities 1,1,2 and 3 respectively. Both A and B send beacons after one CCA time. C and D sense the beacons after two CCA times, and hence they postpone their transmissions to the next time slot. Both A and B enter the CR phase and contend for the channel. A picks a shorter backoff and successfully transmits a packet while B senses this packet and defers its transmission to the next slot. In the third time slot, C transmits a beacon after two CCA times which is sensed by D after three CCA times. Hence D defers its transmission while C transmits its packet. Finally D transmits its beacon and packet in the fourth time slot. (B) shows the expanded view of one time slot which clearly shows the PR and CR phases. (C) shows how node A uses a higher beacon transmission power to block out potential interfering nodes for receiver B . C is a one-hop node whose transmission to G would interfere with both A 's transmission and B 's reception. F and E are two-hop nodes whose transmission would affect B 's reception. I 's transmission to D would not affect A or B , however D 's acknowledgment to I would interfere with B 's reception. Node H 's transmission would not affect A or B , but it will get blocked out. This is an instance of the exposed terminal due to Siren's beacon transmission.

4) *Contention Resolution in Siren:* The nodes that survive the PR phase compete on an equal basis for channel accesses in the CR phase. Siren uses CSMA/CA with fixed window backoffs for CR. Nodes take a random backoff and sense the channel at the end of the backoff period. If a node senses a carrier, then it simply goes to the next slot where it runs the PR phase again. This scheme provides equal chances to all competing nodes in the CR phase. We choose this CR mechanism because it is simple to implement.

E. DWOP

In DWOP, nodes exchange priority information by overhearing RTS/CTS/DATA/ACK exchanges onto which the packet priorities are piggybacked. The default MAC in the CC2420 does not support RTS/CTS. We added support for RTS/CTS in the CC2420 MAC. RTS and CTS are sent as minimum size packets (16 bytes) plus 2 bytes for the packet priority.

One issue with DWOP is that in multihop wireless environments, it is possible that priority information often become stale or lost. This happens due to several reasons. First, packet priorities may change dynamically even at the time-scale of medium access (a few hundred microseconds to a few milliseconds), requiring nodes to perform an RTS/CTS exchange every time the packet priority changes (the CTS is necessary to inform nodes two hops away). Second, priority information could be lost due to collisions and interference. Stale or lost priority information may lead to deadlock. To deal with the deadlock caused by lost or stale information, we introduce a simple timeout feature which works as follows. A node with a packet to transmit checks the priority table. If its priority is lower than the highest priority in the table, it takes a backoff for a random period of time and initializes a counter. At the end of the backoff period, it will check the table again. This cycle is repeated N times, incrementing the counter at every check. If the counter reaches N with the highest priority remaining unchanged in the table, the node is allowed to go

ahead and transmit the packet. We verified that $N = 20$ works the best in our experiments.

F. Relative Overheads

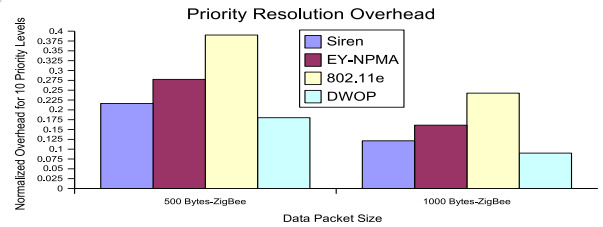


Fig. 2. Normalized priority resolution overhead for Siren, EY-NPMA and 802.11e with non-overlapped backoff periods with different data packet sizes.

In this section we quantify the overhead for each MAC by comparing the time taken for PR normalized by the total time for transmission of one data packet (with different data packet sizes and 10 priority levels). We plot the result in Figure 2. ZigBee radios have a link speed of 250Kbps (resulting in a per-byte transmission time or “byte-time” of $32\mu s$, CCA time of $320\mu s$ and minimum packet transmission time of $512\mu s$). For PMAC (802.11e), the overhead for priority resolution is the backoff period times the number of priority levels. We assume the conventional backoff period of 32 byte-times; 1.024ms for ZigBee. Siren's beacon sensing saves about 20% overhead compared to a naive implementation of beaconing (marked as EY-NPMA in the figure) in ZigBee radios. PMAC incurs significantly more overhead compared to Siren. This is because in PMAC, the PR and CR schemes are interlocked with each other, and hence each additional priority level incurs the overhead of one additional backoff period while in Siren, adding an additional priority level requires only an additional CCA time. An important attribute of DWOP is that it supports infinite priority levels – the overhead is always the time taken for the RTS and CTS exchange.

IV. IMPLEMENTING FAIRNESS POLICIES

In this section, we describe how different fairness policies are implemented on top of PMAC, Siren and DWOP. Each implementation is mostly agnostic to the kind of MAC used, except in a few cases which we highlight in this section. The only requirement is that the MAC should support absolute prioritization, which is supported by all of our MAC implementations discussed in Section III. We implement the following fairness and QoS policies: static priority, EDF, proportional rate allocation, and proportional fairness. Below, we present our implementation; since the implementation of static priority is straightforward, we omit its description.

A. Proportional Rate Allocation and Time Fairness

Consider the problem of proportional throughput allocation in a WLAN [10]. Let there be N nodes in a WLAN which require service differentiation to the effect that their obtained throughput values are in the ratio $r_1 : r_2 : \dots : r_N$. A related problem is maintaining *temporal fairness* (where all nodes receive equal channel time) in multi-rate WLANs. Suppose N stations access the channel with different transmission rates R_1, R_2, \dots, R_N (due to varying channel quality) in a WLAN where all nodes can hear each other. IEEE 802.11b has been shown to be highly “unfair” in such an environment [12]. A solution is to assign equal time-shares among competing nodes on the channel. Existing work [13], [11], [29], [33] implements this solution by directly modifying MAC. The temporal fairness problem can also be framed as a proportional rate allocation problem in the following manner: if the N nodes access the channel so that their obtained throughputs are proportional to their transmission rates, i.e. $R_1 : R_2 : \dots : R_N$, then the time spent by each node on the channel is equalized.

We propose the following algorithm for achieving *approximate proportional rate allocation*. Let there be P priority levels available. A station i , $1 \leq i \leq N$, sets the priority p_i of its outgoing packets by executing a maximum of $P - 1$ Bernoulli trials with probability $\phi(i) = \frac{r_i}{\sum_{i=1}^N r_i}$. If it wins in trial m , $1 \leq m \leq (P - 1)$, it sets $p_i = m$. If it runs $P - 1$ trials without success, then it sets $p_i = P$.

We omit formal proof to save space. Instead, we provide an informal discussion. Consider a single time slot. If only a single node wins in the first Bernoulli trial, then it sends a beacon with priority 1, and grabs the channel and transmit its data packet. This occurs with probability $\sum_{i=1}^N \phi(i)(1 - \phi(i))^{N-1}$. Now, if more than one node wins the first Bernoulli trial, then they simultaneously transmit beacons and enter the CR phase. This gives an equal opportunity to such nodes (as all are in one hop), so the medium access probabilities among such nodes do not follow the required ratios. However, note that the probabilities $\phi(i)$ are normalized to 1. Hence, on average, the number of nodes winning the first (or any) Bernoulli trial is close to 1. Hence the probability of medium access for a node i approximates the Bernoulli probability $\phi(i)$. Since the Bernoulli probabilities are normalized based on the required proportional rate allocation, the medium access probability

of nodes winning the first Bernoulli trial approximates the required ratios.

Now consider the case that no node wins the first Bernoulli trial – this happens with probability $\prod_{i=1}^N (1 - \phi(i))$. Such nodes try the Bernoulli trial again and by the same logic as above their access pattern also approximates the required ratios. This process is repeated for $P - 1$ trials. All nodes which do not win in any of the $P - 1$ Bernoulli trials transmit beacons with priority P . Clearly the medium access pattern between this set of nodes will not follow the required ratio, but will be equally distributed. However, the probability of all nodes not winning their $P - 1$ Bernoulli trials is given by $(\prod_{i=1}^N (1 - \phi(i)))^{P-1}$. This probability quickly falls down to 0 with increase in P . Hence the more priority levels becomes available to the system, the closer the approximation to the desired ratios gets.

Implementation Specifics: PMAC and Siren are implemented with 6 priority levels ($P = 6$). However, DWOP supports infinite priority levels. This has an implication as to how close PMAC and Siren can approximate the desired rate allocation ratios when compared to DWOP. While DWOP will always show closer approximation to the desired rate allocation, we shall see in Section V-E that 6 priority levels is enough for PMAC and 802.11e to have reasonable accuracy.

B. EDF Scheduling in Multi-hop Networks

EDF scheduling can be implemented as described by Kandia et al. [16]. Every packet at its source is stamped with a TTL (time-to-live) which is its desired end-to-end delay bound. At every hop, just before the transmission of this packet, its TTL is decremented by the amount of time spent in the queue. The TTL value is directly mapped to a priority level at each hop. This allows nodes with backlogged packets with a very low TTL to access the channel quickly and flush such packets. More precisely, we assign the priority p_i of an outgoing packet i as follows. Let D_{max} be the maximum deadline value that a real time packet can have and P be the number of priority levels. Suppose that a packet i has a deadline d_i . Then p_i is set to d_i/Δ where $\Delta = D_{max}/P$.

Implementation Specifics: In our EDF experiments in Section V-B, all realtime flows are stamped with an initial TTL of $D_{max} = 500$ ms. For Siren and PMAC implemented with 6 priority levels, this corresponds to $\Delta = 83.33$. Since DWOP supports infinite priority levels, the TTL value is itself used as the priority and nodes order packet transmissions based on the TTL of their interfering neighbors.

C. Proportional Fairness in Multi-hop Networks

Chen et al. [8] propose a joint design of congestion control, routing, and scheduling to maximize aggregate utility in a wireless network. The algorithm is implemented as follows.

Each node maintains a per-flow queue. A node decides which per-flow queue it services next and where to route a packet from this queue based on a *price* which is computed by taking a queue size difference between that node and its neighbors. More precisely, let l_i^k and l_j^k be the per-flow queue

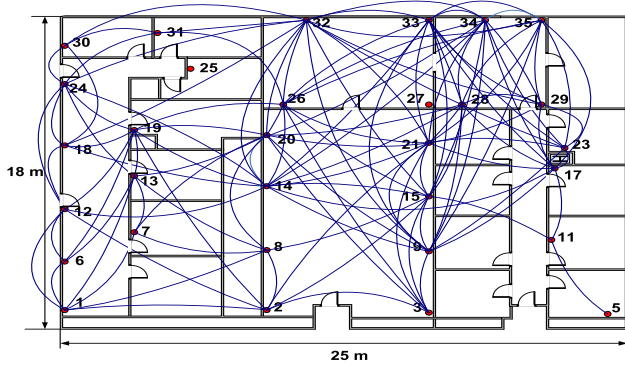


Fig. 3. Above figure shows the 35 node mote testbed distributed across one floor of the MRC building. Lines between nodes indicate connectivity achieved with data transmission power of -5 dBm.

length for flow k at nodes i and j . For each flow k , node i computes price C_{ij}^k between itself and each of its neighbors j as follows: $C_{ij}^k \leftarrow l_i^k - l_j^k$. C_{ij}^k indicates whether node i should route packets belonging to flow k over node j . Intuitively, the largest value of C_{ij}^k hints that node i should route packets for flow k over the neighbor j with the smallest queue length l_j^k . If multiple per-destination queues at node i are backlogged, node i first schedules a packet of flow k and forwards this packet to its neighbor j so that the differential price C_{ij}^k is maximized for all flows k and all neighbors j . The rationale behind this is to give to congested nodes a prioritized access to the channel so that they can drain their queues faster (a large queue may cause a large price). Note that with a queue size of Δ packets per flow, the value of C_{ij}^k may vary between $-(\Delta - 1)$ and Δ , hence requiring 2Δ priority levels for a one-to-one mapping. The packet is then transmitted by the MAC using this priority level. This approximation does not guarantee proportional fairness, but it provides a practical implementation that still achieves a high aggregate utility of the system, which is defined to be $\sum \log(x)$ where x is a per-flow throughput. This is shown in Section V-D.

Implementation Specifics: In our experiments we use a per-flow queue size of 3 packets, $\Delta = 3$. This allows us to use a one-to-one linear mapping of the queue differential $[-2, 3]$ to the packet priority $[0, 5]$ without any quantization error. Hence all three MACs – Siren, PMAC and DWOP use the mapped packet priority directly for priority resolution.

V. TESTBED RESULTS

A. Experimental Methodology

Our sensor network testbed consists of 30 MicaZ sensor nodes distributed in labs and student offices in our computer science building, as shown in Figure 3. The lines connecting the nodes indicate wireless links. We implement Siren, PMAC and DWOP as described in Section III. As the base case of a MAC protocol which does not support priority resolution, we consider B-MAC [24], which is the default MAC protocol for MicaZ sensor nodes under the TinyOS environment. Medium access in B-MAC is similar to that in 802.11, with the

exception that it does not perform exponential backoffs. Just before packet transmission, a node performs a backoff over a fixed window. At the end of the backoff, the node transmits the packet if the channel is idle, else it takes a backoff again. We will refer to this MAC as *CSMA-DATA/ACK* for the rest of this section. For a fair comparison, we also compare with B-MAC with RTS/CTS as an additional performance baseline. We denote it as *CSMA-RTS/CTS*. We now look at how each MAC performs for different fairness policies.

B. EDF Scheduling

We implement EDF scheduling as described in Section IV-B. Our experiment is as follows. We generate 10 concurrent flows on the testbed, with sources and destinations for the flows generated randomly. The source node of each flow stamps each outgoing packet with a TTL of 500ms. All flows are transmitted at 3 Kbps. The experiment is repeated 15 times. We report the distribution of the average per-flow delay for all runs in Figure 4 (A). Figure 4 (B) shows the fraction of the runs for each MAC in which the average per-flow delay is less than the EDF deadline of 500ms.

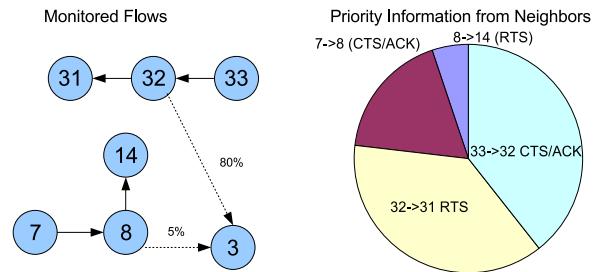


Fig. 5. Above figure illustrates the DWOP problem in lossy networks. Flows $33 \rightarrow 32 \rightarrow 31$ and $7 \rightarrow 8 \rightarrow 14$ are real-time flows from one of the runs in the EDF scheduling experiment. Node 3 is the source of a flow which was observed to be starved. We monitored node 3 and found that it was continuously deferring its own transmission due to perceived high priority packets in its neighborhood. The figure on the right represents the normalized fraction of time spent in deferring attributed to the constituent flow segments. About 75% of the time 3 defers due to the RTS heard from node 32 ($32 \rightarrow 31$) or the CTS heard from 32 ($33 \rightarrow 32$).

Siren is able to maintain a delay less than the desired TTL for 80% of the runs, while in the case of DWOP and PMAC, only 30% and 15% of the runs fulfill the EDF deadline. The poor performance of DWOP is due to inconsistent cache information causing frequent deadlocks. This problem comes up quite frequently on our testbed due to long, lossy links common in wireless networks. To investigate this problem further, we look at one of the scenarios closely in Figure 5. In the figure, node 3 is the source of a realtime flow getting starved. This was because 3 kept deferring its own transmissions due to RTS and CTS messages from node 32, which is quite some distance away. The small size of RTS and CTS messages (18 bytes) allowed these packets to be received intermittently at 3. However, the consequent DATA packet signaling the end of the high priority transmission is frequently lost at node 3, due to its large size (384 bytes). This leaves 3 deadlocked. The priority information from neighboring node 8 reaches

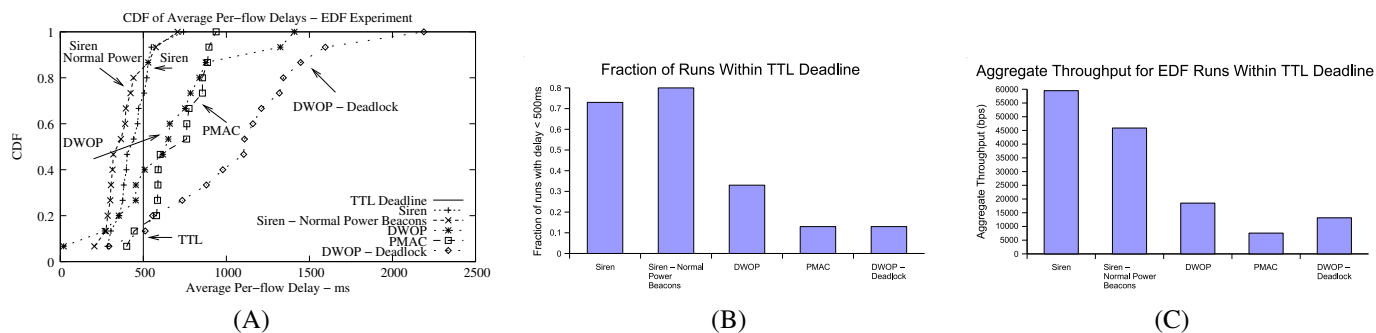


Fig. 4. (A) shows the distribution of delays in the EDF scheduling experiment. (B) shows the fraction of runs which have average per-flow delay to be less than the EDF deadline of 500ms. (C) shows the aggregate throughput of all such runs for each MAC.

3 without much problem since the loss rate from 8 to 3 is 5%. Our modification to DWOP to deal with this issue (Section III-E) does not help much as it simply reduces the duration of this deadlock. We also run the original DWOP protocol, referred to in Figure 4 as “DWOP - Deadlock”. The long delays for “DWOP - Deadlock” show the degree by which the original DWOP gets deadlocked without a deadlock timeout feature.

An interesting observation here is that a beacon based scheme like Siren does not suffer from the same problem in the presence of lossy channel. This is because HOL priorities are advertised through beacons which are short packets, and hence do not incur high loss rate. In addition, it is not necessary for a node to receive the beacon fully and correctly to determine that a neighbor has a high priority packet pending. As long as it detects a busy channel, it will defer access.

At this time, we tested an important component of Siren – the high power beacons (Section III-D2). We ran the same experiment with Siren with the beacon transmission power set to be the same as data transmission power, denoted by *Siren - Normal Power Beacons*. Although this seems to slightly reduce the average per-flow delay compared to Siren with high power beacons, the aggregate throughput per run is also reduced. This is because the normal power beacons cannot preserve priority across two hops, causing hidden terminal collisions, reducing the throughput at the destination. PMAC’s differentiated backoffs do not protect against hidden terminals, which causes to its poor performance.

Siren with normal power beacons results in about 10% less throughput for both high and low priority flows. This indicates that high power beacons are effective in preserving priorities across two hops. However, Siren’s high power beacons may not reach all potential interfering nodes. As a result, Siren’s throughput for high priority flows is less than that for DWOP (Figure 6 (A)), which is able to prevent all two hop interference through the use of RTS/CTS messages.

C. Static Priority Scheduling

We consider an environment where two classes of applications coexist: high priority (HP) and low priority (LP). Our fairness policy is to allow the HP flows access to the channel as much as possible, deferring accesses of LP flows as long as

HP flows are present. This is achieved by setting the priority of all HP flows to be higher (set to 2) than that of all the LP flows (set to 3). Our experiment consists of 5 HP flows which transmit at a constant bit rate of 3 Kbps, and 5 LP flows transmit at a constant bit rate of 5 Kbps. The sources and destinations of these 10 flows are selected randomly. This experiment is repeated 15 times with different source and destination pairs. We report the throughput of each class normalized with the source sending rate in Figures 6A and B.

Figure 6 (A) shows that both DWOP and Siren are able to provide full access to HP flows with DWOP performing better. However, DWOP does this at the expense of throughput for LP flows. From Figure 6 (B), we can see that LP flows in DWOP get about 50% of the throughput achieved in Siren. This is again due to the inconsistent priority database information caused by lossy links. Note that for DWOP, the HP flows do not affect each other, since they are all set to the same high priority level, and hence the HP flows do not get into deadlock. However, the LP flows may experience deadlock if there is a high priority flow within two-hops. The deadlock happens when a HP packet blocking LP packets has actually finished transmission, but the information about finishing the transmission is lost while the channel is free. Since the DATA packet is much larger than RTS and CTS packets and is subject to higher packet loss, its transmission may not register among neighboring nodes, which leaves them deadlocked, as seen in Figure 5 earlier. Figure 6 (C) shows the fraction of LP flows in all the runs that get starved (0 throughput).

D. Proportional Fairness in Multi-hop Networks

The optimization framework described in Section IV-C consisted of three components, source rate control, per-flow queuing and MAC scheduling. Instead of directly comparing the performance of the framework under each of DWOP, Siren and PMAC, we take the approach of incremental performance evaluation of each component.

We start with the baseline – a framework without any congestion control or scheduling. We select 8 sources randomly which transmit packets as fast as possible toward 8 randomly selected destinations. We refer to this as *CSMA-DATA/ACK-Full-Rate*. We then add the source rate control and per-flow queuing components to the framework, but leave

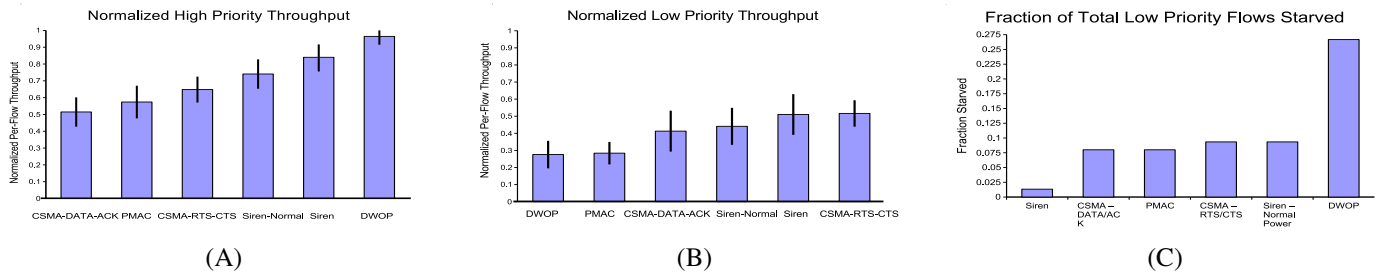


Fig. 6. (A) and (B) show the normalized per-flow throughput of high and low priority flows respectively, in increasing order. Throughput is normalized w.r.t source sending rate (3Kbps for high priority and 5Kbps for low priority). (C) shows the fraction of low priority flows getting starved (0 throughput).

out the MAC scheduling component. We refer to this result as *CLO-CSMA-XX* – the suffix *XX* denoting whether the underlying MAC uses RTS/CTS or just DATA/ACK. Finally, we test the full framework under Siren, DWOP and PMAC, denoted respectively as *CLO-Siren*, *CLO-DWOP*, and *CLO-PMAC*. The performance metric we report is the aggregate utility ($\sum \log(x)$), where x is the throughput obtained by a flow. We report the distribution of this metric in Figure 7 (A).

Note that a low value of the aggregate utility indicates severe starvation for some flows, although the aggregate throughput may be high as seen in Figure 7 (B). The main causes of flow starvation in conventional wireless networks are the inherent MAC-layer unfairness observed in CSMA-based MAC protocols [26] and the lack of effective congestion control which causes queue overflows at intermediate nodes [14], [31]. Both these factors cause the low aggregate utility in the case of (*CSMA-DATA-ACK-Full Rate*). By adding source rate control and per-flow queuing starvation due to queue overflows is mitigated to some extent as can be seen in the case of *CLO-CSMA-DATA/ACK* and *CLO-CSMA-RTS/CTS*. The large improvement seen in *CLO-CSMA-RTS/CTS* indicates that a significant factor of the flow starvation is due to hidden terminal scenarios on our testbed. The disappointingly low utility in *CLO-PMAC* compared to *CLO-CSMA-RTS/CTS* is due to the vulnerability of PMAC to hidden terminals.

Finally *CLO-Siren* and *CLO-DWOP* both show the maximum value of the aggregate utility among all the tested cases. This shows that even after congestion control and per-flow queuing, there is still exists some scope of improvement by tuning the MAC scheduling component. It is interesting that both Siren and DWOP provide more or less equivalent performance for the proportional fairness framework. An interesting question here is why *CLO-DWOP* does not suffer from the deadlock situations seen in the earlier experiments. This is because in CLO, a deadlocked flow immediately begins to be backlogged, thus increasing its queue differential. Eventually, its priority becomes equal to the highest priority, and it is able to flush out its packets. This prevents complete starvation of any particular flow. However in the case of static priority scheduling, the deadlocked flow’s priority never changes, and in EDF scheduling, a deadlocked flow’s priority (TTL) changes very slowly before its priority becomes large enough to transmit. Clearly, if used in wireless multihop environments

where deadlock situations abound, with a fairness policy where the priority of a flow is does not change fast enough in response to deadlock situations, DWOP will perform poorly.

E. Proportional Rate Allocation

For proportional rate allocation, we run the following test. We pick 4 and 6 nodes of our testbed in sequence in such a way that they are all within radio range of each other. These nodes then transmit packets to a common sink node by the proportional rate allocation algorithm described in IV-A. The sink node records the ratio of the throughputs obtained from each node which we report in Figure 7 (C). We use Siren with 6 priority levels for this experiment.

DWOP represents the faithful implementation of our rate allocation algorithm, since it is not constrained by number of priority levels. As a result, its rate allocation is closest to the desired ratios. Siren with just 6 priority levels shows performance close to that of DWOP. The performance of PMAC was similar to that of Siren for this experiment because of the absence of hidden terminals.

VI. RELATED WORK

Substantial research exists on the individual components of a fairness control framework i.e. fairness policy, PR, and medium access. However we will focus only on the PR scheme in this section. Priority resolution has been achieved in the MAC layer by various means. Broadly, they can be classified into three approaches – beacon based [28], [6], [32] or backoff based [3] or scheduling based [17], [16]. This bifurcation can be traced back to the original two competing standards for the wireless MAC – 802.11/802.11e [3], which uses a differentiated backoff strategy, and HIPERLAN/1 [6] which uses beacons (the scheduling-based algorithms came later).

Vaidya et al. propose a PR scheme, BTPS [32] (Busy Tone Priority Scheduling) supporting two priority levels. In this scheme, nodes are required to listen on three channels during idle periods – a data channel and two narrow-band busy-tone channels. A node with a backlogged high priority packet transmits a busy tone signal (BT1) every M slots. Neighbors that hear BT1 will forward this signal to the node’s two-hop neighbors on a different band (BT2). The node’s two-hop neighbors with backlogged low priority data that hear BT2 will defer their transmissions. This ensures the transmission of high priority packets. BTPS solves the hidden terminal problem

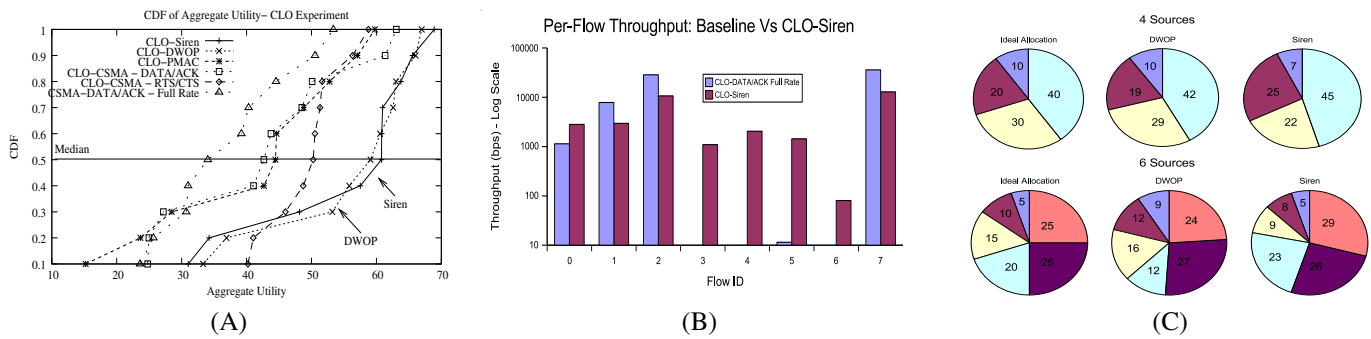


Fig. 7. (A) shows the distribution of aggregate utility $\sum \log(x)$ for CLO experiment. (B) shows per-flow throughput of baseline CSMA and CLO-Siren for one specific run for the CLO experiment. Note that CSMA's aggregate throughput is higher, but CLO-Siren gets better utility since no flow is starved. (C) shows the proportional rate allocations for DWOP and Siren compared with the ideal rate allocations for 4 and 6 sources.

effectively, but requires nodes to listen on multiple channels during idle periods. It is also constrained to two priority levels.

In black burst contention [28], [25], nodes with backlogged packets first wait for a short time period and then transmit small pulses of energy or *black bursts* which serve to “jam” the channel. The length of a black burst is an increasing function of the priority of the backlogged packet and/or how long a node has been waiting for access to the channel. After transmitting the black burst, a node observes the channel for a short period of time to check if any other node is transmitting a longer black burst. If not it will transmit the data packet.

VII. CONCLUSION

We evaluate MAC protocols for their effectiveness in implementing different commonly cited fairness policies. Differential backoff-based schemes are easy to implement (do not need special constructs such as RTS/CTS or time synchronization), but are vulnerable to hidden terminals under multihop wireless environments. Beacon-based schemes are robust to hidden terminals and lossy wireless environments, but are difficult to implement due to the requirement of time synchronization. Scheduling-based schemes provide infinite priority levels with low overhead, but are vulnerable to loss of priority information, often leading to deadlocks.

REFERENCES

- [1] CC1000/CC2420 Low Power FSK transceiver, Chipcon Corporation.
- [2] E. Inc. Universal Software Radio Peripheral. <http://ettus.com>.
- [3] IEEE 802.11 standard.
- [4] IEEE 802.15.4, ZigBee standard, <http://www.zigbee.org/>.
- [5] XBOW MicaZ Mote Specifications, <http://www.xbow.com/>.
- [6] Radio Equipment and Systems (RES); High Performance Radio Local Area Network (HIPERLAN); Functional Specifications, ETSI, 1995.
- [7] I. Aad and C. Castelluccia. Differentiation mechanisms for IEEE 802.11. In *INFOCOM '01*.
- [8] L. Chen, S. Low, M. Chiang, and J. Doyle. Optimal cross-layer congestion control, routing and scheduling design in ad hoc wireless networks. In *INFOCOM '06*.
- [9] M. Garetto, J. Shi, and E. W. Knightly. Modeling media access in embedded two-flow topologies of multi-hop wireless networks. In *MobiCom '05*.
- [10] Y. Ge, J. Hou, and S. Choi. An analytic study of tuning system parameters in ieee 802.11e enhanced distributed channel access. *Computer Networks*, 2005.
- [11] D. Hadaller, S. Keshav, and T. Brecht. Mv-max: improving wireless infrastructure access for multi-vehicular communication. In *CHANTS '06*.

- [12] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda. Performance anomaly of 802.11b. In *INFOCOM '03*.
- [13] M. Heusse, F. Rousseau, R. Guillier, and A. Duda. Idle sense: an optimal access method for high throughput and fairness in rate diverse wireless lans. In *SIGCOMM '05*.
- [14] B. Hull, K. Jamieson, and H. Balakrishnan. Mitigating congestion in wireless sensor networks. In *Sensys '04*.
- [15] E.-S. Jung and N. H. Vaidya. A power control mac protocol for ad hoc networks. In *MobiCom '02*.
- [16] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly. Distributed multi-hop scheduling and medium access with delay and throughput constraints. In *MobiCom '01*.
- [17] V. Kanodia, A. Sabharwal, B. Sadeghi, and E. Knightly. Ordered packet scheduling in wireless ad hoc networks: mechanisms and performance analysis. In *MobiHoc '02*.
- [18] D. J. Leith and P. Clifford. Using the 802.11e edcf to achieve tcp upload fairness over wlan links. In *WIOPT '05*.
- [19] S. Lin, J. Zhang, G. Zhou, L. Gu, J. A. Stankovic, and T. He. Atpc: adaptive transmission power control for wireless sensor networks. In *Sensys '06*.
- [20] X. Lin and N. B. Shroff. The impact of imperfect scheduling on cross-layer congestion control in wireless networks. *IEEE/ACM Trans. Netw.*, 14(2):302–315, 2006.
- [21] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi. The flooding time synchronization protocol. In *Sensys '04*.
- [22] T. Nandagopal, T.-E. Kim, X. Gao, and V. Bharghavan. Achieving mac layer fairness in wireless packet networks. In *MobiCom '00*.
- [23] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *Sensys '04*.
- [24] J.-P. Sheu, C.-H. Liu, S.-L. Wu, and Y.-C. Tseng. A priority mac protocol to support real-time traffic in ad hoc networks. *Wirel. Netw.*, 10(1):61–69, 2004.
- [25] J. Shi, T. Salonidis, and E. W. Knightly. Starvation mitigation through multi-channel coordination in csma multi-hop wireless networks. In *MobiHoc '06*.
- [26] V. A. Siris and G. Stamatakis. Optimal cwmin selection for achieving proportional fairness in multi-rate 802.11e w lans: test-bed implementation and evaluation. In *WINTECH '06*.
- [27] J. L. Sobrinho and A. S. Krishnakumar. Quality-of-service in ad hoc carrier sense multiple access wireless networks. In *IEEE JSAC Vol.17, No.8, pp. 1353-1368, 1999*.
- [28] I. Tinnirello and S. Choi. Temporal fairness provisioning in multi-rate contention-based 802.11e w lans. In *WOWMOM '05*.
- [29] S. Wu, Y. Tseng, and J. Sheu. Intelligent medium access for mobile ad hoc networks with busy tones and power control. *IEEE Journal on Selected Area in Communications*, 18(9):1647–57, 2000.
- [30] K. Xu, M. Gerla, L. Qi, and Y. Shu. Enhancing tcp fairness in ad hoc wireless networks using neighborhood red. In *MobiCom '03*.
- [31] X. Yang and N. H. Vaidya. Priority scheduling in wireless ad hoc networks. In *MobiHoc '02*.
- [32] S. Yoo, J. Choi, J. Hwang, and C. Yoo. Eliminating the performance anomaly of 802.11b. volume 3421, pages 1055–1062, 2005.
- [33] G. Zhou, T. He, J. Stankovic, and T. Abdelzaher. Rid: radio interference detection in wireless sensor networks. In *INFOCOM '05*.