

Rate Adaptation in Congested Wireless Networks through Real-Time Measurements

Prashanth A.K. Acharya[†], Ashish Sharma[†], Elizabeth M. Belding[†],
Kevin C. Almeroth[†], Konstantina Papagiannaki[‡]

[†]Department of Computer Science, University of California, Santa Barbara CA 93106

[‡]Intel Research, Pittsburgh PA 15213

{acharya, asharma, ebelding, almeroth}@cs.ucsb.edu, dina.papagiannaki@intel.com

Abstract—Rate adaptation is a critical component that impacts the performance of IEEE 802.11 wireless networks. In congested networks, traditional rate adaptation algorithms have been shown to choose lower data-rates for packet transmissions, leading to reduced total network throughput and capacity. A primary reason for this behavior is the lack of real-time congestion measurement techniques that can assist in the identification of congestion related packet losses in a wireless network. In this work, we first propose two real-time congestion measurement techniques, namely an active probe-based method called Channel Access Delay, and a passive method called Channel Busy Time. We evaluate the two techniques in a testbed network and a large WLAN connected to the Internet. We then present the design and evaluation of Wireless cOngestion Optimized Fallback (WOOF), a rate adaptation scheme that uses congestion measurement to identify congestion related packet losses. Through simulation and testbed implementation we show that, compared to other well-known rate adaptation algorithms, WOOF achieves up to 300% throughput improvement in congested networks.

Index Terms—Wireless communication, Access schemes, Algorithm/protocol design and analysis.



1 INTRODUCTION

The proliferation of IEEE 802.11 networks in recent years demonstrates a dramatic shift in the primary mechanism for Internet access. According to a survey conducted by the Pew Internet Project in February 2007, about one-third of the population of Internet users in the USA connect via wireless networks [1]. IEEE 802.11 networks, in the form of WLANs or city-wide multihop mesh networks, are now expected to support the connectivity requirements of hundreds to thousands of users simultaneously.

The increased usage of 802.11 networks and devices, however, exposes many problems in current networks. IEEE 802.11 is a CSMA/CA based medium access scheme. All the users in the vicinity of each other share the medium as a common resource. A large number of users in a network can lead to excessive load or congestion in the network. Jardosh et al. present an example case study of a large congested WLAN and describe the adverse effects of such congestion [2]. In this network, more than 1000 clients attempted to use the network simultaneously. The network could not sustain this high load: users obtained unacceptably low throughput, and many users were unable to even maintain association with the access points (APs). Eventually the network broke down, causing frustration among the users.

Congestion has an adverse impact on current *rate adaptation* algorithms, an important aspect of the IEEE 802.11 MAC protocol that determines the network throughput. In a multi-rate 802.11 network, rate adaptation is the operation of selecting the best transmission rate, and dynamically adapting this selection to the channel quality variations. The data rates offered by 802.11a/b/g networks vary from a low of 1 Mbps

to the high rate of 54 Mbps. This wide range in the choice of data rates makes the behavior of the rate adaptation algorithm critical to the throughput performance, especially in congested scenarios. Current rate adaptation solutions are typically designed for operation in uncongested networks, where packet loss is more likely to correlate with poor link quality rather than congestion. These solutions have been shown to exhibit inferior performance in congested networks [3], [4]. These solutions do not distinguish congestion-related packet losses from those caused by poor link quality, and react to all packet losses by switching to a lower transmission rate. This rate switch, in turn, increases the channel occupancy time of packet transmissions and adds to the already existing congestion.

In this work, our goal is to design a rate adaptation scheme that provides high network performance in congested networks as well as lightly-loaded networks. In order to design such a rate adaptation scheme, however, our approach is to develop mechanisms that can identify and measure the network congestion level in real-time. Traditional metrics, such as network throughput, do not accurately characterize congestion in a wireless network because of the locally shared channel and the use of multiple transmission rates. Current congestion metrics proposed for wireless networks are processor intensive, and, therefore, are not suitable for real-time operation. Hence, there is a need for lightweight congestion measurement solutions that can identify congestion in a wireless network in real-time. These mechanisms enable the rate adaptation scheme to respond to the network congestion levels and make intelligent decisions about the choice of transmission rate. In summary, we require congestion measurement solutions to assist in the development of a congestion-aware rate adaptation scheme.

To this end, we present a measurement-driven approach to the characterization of congestion in wireless networks and to the design of a congestion-aware rate adaptation scheme. Our two main contributions are as follows. First, we develop two measurement techniques that can identify wireless network congestion in real-time. The first technique is active and measures the *channel access delay*, the minimum time delay for a packet transmission in the network. The second technique is passive and measures the *channel busy time*, the fraction of time for which the medium was utilized during some time interval. We evaluate and compare the performance of these techniques in a testbed as well as a large WLAN with active users connected to the Internet. We show that the channel busy time can accurately measure network congestion in real-time.

Second, we present the design and implementation of a new rate adaptation scheme called Wireless cOngestion Optimized Fallback (WOOF). This scheme uses the channel busy time metric in real-time to probabilistically differentiate between packet losses due to congestion and those due to poor link quality. Our testbed evaluations in congested wireless network scenarios show that WOOF obtains significantly higher throughput (up to a three fold improvement) compared to current solutions. Simulations further show that WOOF is able to offer significant performance improvements in large WLANs with hundreds of users.

In a prior version of this work, we presented the design and evaluation of the channel busy time metric [5]. Further, we presented the design and initial results from the evaluation of the WOOF rate adaptation algorithm. In this work, we extend our exploration of congestion measurement techniques, and also perform comprehensive performance evaluations to understand the robustness and scalability of the WOOF algorithm. In particular, we present Channel Access Delay, an alternate technique for real-time identification of congestion in wireless networks. In addition, we compare the performance of WOOF against that of Collision-Aware Rate Adaptation [6], an algorithm designed with goals similar to ours. We demonstrate the utility of incremental adoption of WOOF. Further, we present results from simulation-based performance evaluations of WOOF in large scale networks.

The remainder of the paper is organized as follows. Section 2 surveys the literature on rate adaptation algorithms for IEEE 802.11 networks. Section 3 describes the different congestion measurement methods. We evaluate the performance of these methods in Section 4. Sections 5 and 6 describe the design and evaluation of the WOOF scheme. We conclude the paper in Section 7. Throughout the paper, we use the term data rate to refer to the rate of transmissions in the wireless network as governed by the physical layer signal modulation scheme.

2 STATE-OF-THE-ART IN RATE ADAPTATION

Rate adaptation in a multi-rate IEEE 802.11 network is the technique of choosing the best data rate for packet transmission under the current channel conditions. The IEEE 802.11 standard does not specify the details of the rate adaptation algorithm to be used. Thus IEEE 802.11 card vendors and

researchers have proposed and implemented a variety of rate adaptation algorithms.

The probability of successful transmission of a packet for a given data rate can be modeled as a function of the Signal-to-Noise Ratio (SNR) of the packet at the receiver [7]. A packet can be transmitted at a high data rate if the SNR at the receiver is high and the packet can be received without errors. On the other hand, if the SNR is not high, a lower data rate helps achieve more robust communication. Therefore, one of the ideal metrics to base the choice of transmission data rate is the SNR of a packet at the receiver. However, under current IEEE 802.11 implementations, it is not trivial for the transmitter to accurately estimate the SNR at the receiver because signal strength exhibits significant variations on a per-packet basis. This has led to the development of various solutions that attempt to estimate link quality through other metrics.

Receiver-Based Auto Rate (RBAR) [8] is a rate adaptation scheme that proposes use of the RTS-CTS handshake by a receiver node to communicate the signal strength of received frames. The receiver measures the signal strength of the RTS message and uses this information to select an appropriate data rate for transmission of the data frame. The transmitter is informed of the selected data rate through the CTS message. A drawback of this scheme is that it cannot be used in modern 802.11 networks where the RTS-CTS messaging is generally disabled. Additionally, RBAR requires modification to the format of the CTS message, which in many cases necessitates modification of hardware and is thus infeasible.

A recent work by Judd et. al. uses the property of channel reciprocity to estimate the signal strength at the receiver, based on local measurements of received signal strength [9]. This approach requires exchange of information such as noise floor, transmit power among the nodes in the network, similar to the RTS-CTS messaging of RBAR.

At the transmitter node, the most commonly used information to help in choosing a data rate is the packet loss information (i.e., when an ACK is not received). Auto-Rate Fallback (ARF) was among the first rate adaptation schemes that was practically implemented [10]. ARF interprets patterns of packet loss (e.g., four consecutive losses) as triggers to change the data rate. Several other rate adaptation schemes, such as AARF [11], also use packet loss patterns for rate adaptation decisions. Most current 802.11 devices implement ARF or variations of ARF [6]. Recent work such as SampleRate [12] shows that ARF and AARF perform poorly for links that are not always 100% reliable. Therefore SampleRate uses a statistical view of packet loss rates over a period of time (e.g. 2s in [12]) to choose the rate with the least expected transmission time. We describe SampleRate in detail in Section 5.3.

A common feature among all the above described rate adaptation schemes is that they consider all packet losses to be due to poor link quality. They do not distinguish between packet losses caused by channel quality and packet losses caused by either hidden terminal transmission or congestion. Ideally, the rate adaptation algorithm should only consider the packet losses due to poor channel conditions, multipath effects, fading, etc. Packet losses due to hidden terminals or congestion

should not affect the rate adaptation algorithm. On observing packet loss, a rate adaptation scheme that does not distinguish the cause of the packet loss reduces the transmission data rate. In the case of packet loss due to congestion or hidden terminals, such a reduction of data rate is unnecessary. Even worse, the lower data rate increases the duration of packet transmission, thereby *increasing* congestion and the probability of a packet collision. Additional collisions result in packet loss, which leads to further reduction in data rate.

The challenge for a rate adaptation algorithm is to be able to identify the cause of a packet loss, i.e., whether a packet was lost because of a bad link, hidden terminal or congestion. In the absence of such a distinction, rate adaptation algorithms may actually compound network congestion [4]. In our work, we attempt to probabilistically identify congestion-related packet losses and minimize their impact on rate adaptation.

Two rate adaptation algorithms, namely Robust Rate Adaptation Algorithm (RRAA) [13] and Collision-Aware Rate Adaptation (CARA) [6], are designed to minimize the impact of packet losses that are not due to channel errors. RRAA selectively uses RTS-CTS handshaking to avoid hidden terminal collisions. RRAA was not designed to explicitly handle congestion-related losses in the network. On the other hand, CARA builds upon ARF [10] and suggests the use of an adaptive RTS-CTS mechanism to prevent losses due to contention. However, CARA requires turning on the RTS-CTS mechanism for the first retransmission of a packet, i.e., upon failure of the first transmission attempt. Most current hardware does not support this facility and thus may require modification. In contrast, our solution is implemented purely in software. Moreover, CARA is built upon ARF and thus inherits the problems of ARF, where it uses patterns of packet loss for adaptation decisions. This has been shown to lead to incorrect rate selection [13].

An orthogonal approach to address the problem is to modify the contention resolution mechanism of IEEE 802.11 and minimize the congestion-related losses. The Idle Sense protocol [14] adjusts the contention-window parameters of a node to reduce packet collisions. This method enables a node to estimate collision rate, from which it can estimate the frame error rate due to poor channel conditions. Idle Sense, however, requires each node to measure the number of idle slots between transmissions - this requires a firmware update, and is not possible on many hardware platforms. Further, Idle Sense requires modification to the 802.11 DCF mechanism; its interaction with other existing 802.11 devices is not clear. A comparison of our solution with that of Idle Sense is beyond the scope of this work.

Based on the above discussion, we note that while metrics such as SNR and idle slots provide valuable input for a rate adaptation algorithm, the complexity of implementation and the associated overhead makes it difficult to develop a practical solution. On the other hand, we show that the network utilization metric can measure congestion locally, in real-time, and with low overhead. Therefore, it serves as a suitable metric that can be used in the design of a congestion-aware rate adaptation algorithm. Our scheme, Wireless cOngestion Optimized Fallback (WOOF) is implemented on existing hardware,

and we show that WOOF can coexist with current 802.11 implementations.

We next discuss two techniques to measure congestion levels in a wireless network in real-time. Later, in Section 5, we describe the design of a rate adaptation scheme that uses these measurement techniques to adapt to congestion.

3 CONGESTION MEASUREMENT

Congestion on the wired Internet is caused when the offered load on a link approaches the capacity of the network link. Similarly, congestion in IEEE 802.11 wireless networks may be defined as a state where the shared wireless medium is close to being fully utilized by the nodes, because of given channel conditions and/or external interference, while operating within the constraints of the 802.11 protocol [4].

Identification of congestion in wireless networks presents new challenges as compared to wired networks. The shared nature of the wireless medium causes a node to share the transmission channel not just with other nodes in the network, but also with external sources of interference. Unlike wired networks, where throughput degradation on a network link is indicative of congestion, throughput degradation in wireless networks can occur due to a lossy channel, increased packet collisions during congestion or external interference. In addition, throughput of a wireless link is also directly influenced by the rate adaptation algorithm through its choice of transmission data rate. Clearly, if a lower data rate is in use, the throughput for a given time interval will be lower than with a high data rate. Traditional rate adaptation schemes for 802.11 networks fail to identify congestion related packet losses from poor channel quality and resort to the use of lower data rates. In the case when the medium is heavily utilized by a large number of users, packet losses occur primarily due to congestion. The use of a lower data rate increases the transmission time for the same packet size, further degrading network performance [3], [15].

For the above reasons, the time available to a node for transmission, governed by the current medium utilization level, characterizes congestion in a wireless network better than the observed throughput. Several studies have proposed the use of medium utilization as a measure of congestion in the wireless medium [4], [16], [17]. Jardosh et al. show that medium utilization can be used to classify network state as *uncongested*, *moderately congested* and *highly congested* [4]. Hu and Johnson suggest the use of MAC layer utilization information as one of the metrics for route selection in a multi-hop wireless network [16]. They also suggest use of the utilization metric to trigger the Explicit Congestion Notification (ECN) feature of TCP for better throughput in congested wireless networks. AQOR is an admission control scheme for multihop wireless networks that uses medium utilization information for flow admission decisions [17].

There are two possible approaches to measuring medium utilization in real time: *active probing* and *passive measurement*. While an active approach relies on sending probe packets to determine the state of the network, a passive approach monitors a system variable, and then uses that to determine the current network state.

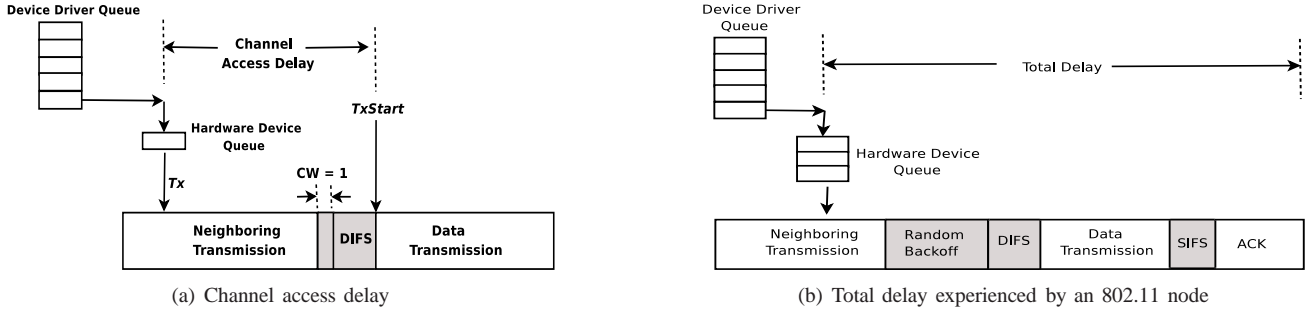


Fig. 1. Per packet channel access delay vs total delay for an 802.11 node.

In this paper, we implement and evaluate two real-time congestion measurement techniques for wireless networks. The first is an active technique that measures the *channel access delay*, the minimum time delay for a packet transmission in the network at any instant. The second technique is passive in nature and measures *channel busy time*, the fraction of time for which the medium was utilized, during some time interval.

3.1 Channel Access Delay: An Active Approach

Channel Access Delay (CAD) refers to the minimum delay between the time a packet is delivered to the 802.11 hardware by the device driver and the time when the medium is first detected to be idle for transmission. Intuitively, if the medium is heavily utilized, the probability that the probing node experiences a higher channel access delay will be higher as compared to a scenario when the medium utilization is low. Thus, CAD values in a given time period can provide useful insight into the cause of packet loss experienced by a node and may be used in network debugging, rate adaptation and congestion measurement. We evaluate the utility of channel access delay to obtain an estimate of the current congestion level in the vicinity of the node by monitoring the CAD values for transmitted probe packets sent at regular intervals.

As shown in Figure 1, the definition of the channel access delay is different from the total delay an 802.11 node experiences to transmit a packet successfully. The latter value includes the time spent by the node in the random backoff phase and the delays experienced by the packet in the device driver and hardware queues. Thus it is necessary to isolate the individual backoff and queuing delay values before calculating channel access delay.

To this end, we developed a tool to accurately compute the channel access delay, based on the framework provided by MadMAC [18], an extension to the open source MadWifi driver for Atheros chipset-based 802.11 devices. Using MadMAC, we control the random backoff by setting the CWmin and CWmax parameters to one (minimum allowed for data queues) and disable retransmission of packets. Queuing delay at the hardware queues is avoided by limiting the queue size to one. This is achieved by controlling the rate at which the device driver delivers the packets to the hardware for transmission. We measure the channel access delay by timestamping two network events for the transmission of each probe packet:

- 1) The local time (T_x) at a node when the device driver delivers a packet to the 802.11 card for transmission.
- 2) The local time (T_xStart) at a node when an interrupt is received from the 802.11 device indicating successful initiation of transmission of the probe packet by the hardware.

The channel access delay can then be computed as:

$$CAD = TxStart - T_x \quad (1)$$

We use fixed-size broadcast packets (to prevent retransmissions) for probing that are transmitted at a fixed bit-rate. The absolute value of CAD also depends on the Distributed Inter Frame Spacing (DIFS) interval and the slot time which may differ based on the 802.11a/b/g mode of operation.

It is important to note that channel access delay for any probe packet is dependent on the instantaneous network activity in the wireless medium. For instance, if a packet is delivered to the hardware for transmission during an ongoing neighboring transmission, the channel access delay will depend on the time it takes for the neighboring transmission to finish. Thus individual values are susceptible to high variability and are unlikely to accurately reflect current medium utilization levels. However, the distribution of a number of CAD values measured within a short time interval enables us to estimate the current congestion level of the network. While on one hand, the distribution of CAD values obtained from a large number of samples yields a more representative statistical view of the current channel conditions, this also adds to the overhead due to probe packets. Clearly, such an active probing technique has an inherent tradeoff between the estimation accuracy and the overhead of probe packets in a time interval further adding to congestion.

We use the *Baumgartner-Weiß-Schindler (BWS)* statistical test [19] to estimate if the medium utilization is above a given threshold. This is achieved by comparing the empirical distribution of CAD values obtained during a live experiment with a known distribution for different medium utilization levels and data rates at which the packets are sent, obtained during the training session on our testbed. The BWS test is a well known nonparametric statistical technique used in the field of biometrics to determine the probability that two individually collected sets of empirical data belong to the same underlying distribution. This nonparametric test uses the difference between empirical distribution functions, and this quantity is weighed by its variance. Such

a test avoids any assumptions on the distribution underlying the observed data. It also performs well even with small sample sizes in complex systems where there is no *a priori* information available about the distribution from which the measured data originate. Section 4.3.1 describes our methodology and the performance of the BWS test in detail.

3.2 Channel Busy Time: A Passive Approach

Channel Busy Time (CBT) refers to the fraction of time for which the wireless channel is busy within a given interval. As measured at a wireless device, it includes the time for transmission of packets from the device, reception of packets, packet transmissions from neighbors, the delays that precede the transmission of data and control frames called Inter-Frame Spacings, and environmental noise.

Jardosh et al. outline a method to calculate medium utilization by adding the transmission duration of *all* data, management, and control frames recorded by a sniffer [4]. However, one drawback of this approach is that it involves significant processing overhead for each received packet, as it requires sniffing the network in *monitor* mode and accounting for transmission delays of data and ACK packets, and the SIFS and DIFS intervals that precede frame transmissions. These complexities make it unsuitable for congestion identification in real-time.

In this paper, we present a practical light-weight implementation of the CBT metric for 802.11 networks using a feature provided in Atheros chipset-based wireless devices, and compare its performance with the technique proposed by Jardosh et al. [4].

To measure the channel busy time, we use the reverse-engineered Open HAL¹ implementation of the MadWifi driver for Atheros AR5212 chipset radios. Atheros maintains 32-bit register counters to track “medium busy time” and “cycle time”. The cycle time counter is incremented at every clock tick of the radio, and the medium busy counter represents the number of clock ticks for which the medium was sensed busy. The medium is considered busy if the measured signal strength is greater than the Clear Channel Assessment (CCA). For Atheros radios, the CCA has been found to be -81dBm [20].

The ratio of the “medium busy time” and the “cycle time” counters gives the fraction of time during which the channel was busy. We found that the counters were reset (to a random value) about once every minute. In our implementation we expose an interface in the `/proc` filesystem to read the counter values from the registers periodically at an interval of one second.

Our implementation of channel busy time measurement is based on the Atheros chipset. The CBT functionality is now supported for all Atheros chipsets via the open-source `ath5k` Linux driver [21]. Based on a study of open-source code and SNMP MIB specifications, we believe that chipsets from other vendors such as Prism and Cisco support CBT-like functionality [22], [21]. Further, the 802.11h Radio Resource Management extension recommends APs to support

measurement of *ChannelLoad*, a metric similar to channel utilization [23]. Therefore we expect the CBT functionality to be supported by a large number of hardware vendors. As we show later in this paper, the CBT metric can provide very useful information for network protocol designers. We believe that other hardware vendors should also expose a similar interface and facilitate cross-layered wireless protocol designs that maximize network performance.

4 EVALUATION OF CONGESTION METRICS

In Section 3, we proposed two techniques to measure congestion in a wireless network in real time. While channel access delay is an active technique that requires the node to actively transmit data packets in the network, channel busy time involves passive measurements without actually requiring data transmission.

To evaluate the performance of the two techniques, we use as a benchmark the medium utilization as seen by a sniffer operating in *monitor* mode. In order to calculate medium utilization, we use the methodology proposed by Jardosh et al. to account for the transmission duration of *all* management, control and data frames, along with the SIFS and DIFS durations preceding each transmission [4]. This helps determine the accuracy of our low overhead implementations of channel access delay and channel busy time by comparing against a fairly comprehensive but high overhead mechanism.

We first describe the experimental setup used to measure medium utilization using the two proposed techniques as well as the benchmark technique, which relies on analysis of packets captured by a sniffer. We then describe in detail the two test environments where we conduct our experiments. Next, we present the performance results of the two techniques in each of the test environments. Finally, we discuss the relative merits and limitations of the two techniques.

4.1 Experimental Setup

In our experiments, we use four Linux laptops equipped with Atheros chipset IEEE 802.11 a/b/g cards, and an access point to evaluate both the active (CAD) and passive (CBT) congestion measurement techniques as described below.

Sniffer: One laptop acts as the sniffer and is placed close to the AP to perform *vicinity sniffing* [24]. As part of vicinity sniffing, the radio on the sniffer laptop operates in *monitor* mode and captures all packet transmissions using the `tethereal` utility. This technique allows us to study the wireless network activity in the vicinity of the AP. The traffic trace from the sniffer is used for the offline calculation of medium utilization values during the experiment. The calculated value of utilization is then used to compare against the CAD and CBT values during the corresponding time interval of the experiment.

We calculate the medium utilization value using the methodology proposed by Jardosh et al. [4]. In the interest of space, we briefly summarize the technique as follows. The medium utilization for a given time interval is the sum of the time required for all data, management, and control frames

1. <http://madwifi.org/wiki/OpenHal> (Dec 2006)

Delay Component	Duration ($\mu\text{sec.}$)
DIFS	50
SIFS	10
Preamble (short)	96
Frame	Preamble + $\frac{8 \cdot \text{frame size}}{\text{rate}}$

TABLE 1

Delay parameters for calculation of medium utilization.

transmitted in the interval and the necessary MAC delay components for each frame. The time required for a frame transmission is determined by the data rate and the size of the frame in addition to the fixed duration preamble. The delay components include the Inter-Frame Spacings such as SIFS and DIFS. Table 1 lists the parameters used for our calculation of medium utilization. We use the short preamble delay of $96\mu\text{s}$ to estimate the minimum such delay in a network with a mix of devices that use a short preamble of $96\mu\text{s}$ and devices that use a long preamble of $192\mu\text{s}$.

Channel Access Delay: To accurately measure the channel access delay, two laptops run our CAD measurement tool using MadMAC [18] as their driver. Both nodes broadcast fixed size probe packets (98 bytes each) at a fixed bit-rate (54 Mbps) and measure the channel access delay for each probe. These nodes are not connected to the AP and hence are not part of the wireless network under test. We fix the contention parameters to a minimum ($CW_{min} = CW_{max} = 1$).

Channel Busy Time: A fourth laptop, also placed close to the AP, continuously measures and records the channel busy time as described in Section 3.2.

In order to compare CAD and CBT values with medium utilization values during the corresponding time intervals, the laptops are time synchronized to a millisecond granularity using NTP. Note that both laptops are tuned to the same channel as the AP.

We next describe the two test environments where the above described experimental setup is used for the performance evaluation.

4.2 Testing Scenarios

We evaluate the CAD and CBT congestion measurement techniques in two different environments. The first is a controlled testbed involving eight client laptops connected to an access point. The other is a real world large scale deployment of a wireless network providing connectivity to more than 1000 clients. We choose the two environments because of their vastly different characteristics. The controlled environment of a testbed allows us the flexibility to vary network load to generate a range of medium utilization values and limit external sources of interference. A real world deployment, on the other hand, serves to verify the performance of our tools in an environment characterized by live Internet traffic, a large number of heterogeneous wireless devices, dynamic user behavior and other environmental factors.

4.2.1 Testbed

We conduct two phases of experiments on an indoor wireless testbed of eight client laptops connected to an access point.

Each client initiates a bidirectional UDP traffic flow with the AP. The rate of data traffic is controlled at each client to generate a range of medium utilization levels.

In the first phase we generate the training data set for the BWS test, based on the CAD values observed for different medium utilization levels, as described in Sections 3.1 and 4.3. This training data is then used to estimate the medium utilization level in the second phase of experiments on the testbed as described in Section 4.3.1, as well as the IETF experiments as described below.

We use UDP traffic as opposed to TCP in our testbed experiments because TCP's congestion control and backoff mechanisms prevent us from controlling the rate at which data is injected in the network. Each client exchanges UDP data with the access point bidirectionally. This creates both incoming and outgoing traffic from the AP and provides us with a mechanism to create a range of medium utilization and congestion levels in the testbed.

4.2.2 IETF Wireless LAN

To verify the performance of the two congestion estimation techniques in a real world scenario with live Internet traffic, we conducted experiments at the 67th IETF meeting held in San Diego in November 2006. The network at the IETF meeting consisted of a large WLAN connected to the Internet with 38 physical AP devices that provided connectivity to more than 1000 clients. The APs were dual-radio devices with one radio tuned to the 802.11a spectrum and the other to the 802.11b/g spectrum. The APs were tuned to orthogonal channels to enable spatial reuse. We chose to perform our experiments with 802.11b/g as there were approximately three times as many users on the 2.4GHz spectrum as the 5GHz spectrum of 802.11a. The APs advertised the following as accepted data rates (Mbps): 11, 12, 18, 24, 36, 48 and 54. This restriction on acceptable data rates enables limiting the cell-size of each AP.

We conducted experiments during several sessions at the IETF, each characterized by a different number of clients connected to the AP. For example, a working group meeting is typically held in a small room and is attended by about 50-100 people on average. On the other hand, a plenary session is attended by approximately 1000 people. The room for the plenary session at the 67th IETF was serviced by eight dual radio physical AP devices. The 2.4GHz APs were tuned to the three non-overlapping channels of the 802.11b/g spectrum. For the evaluation of our congestion measurement techniques, we focused on Day 3 of the meeting, a day that included a plenary session.

4.3 Congestion Estimation Results

We now present performance results for both the congestion measurement techniques in each of the two test environments. There are four sets of results, corresponding to each combination of the two measurement techniques, CAD and CBT, paired with the two test environments, testbed and IETF.

The active probing technique of calculating channel access delays requires sampling of a set of values within a short time interval, following which this set is compared with a

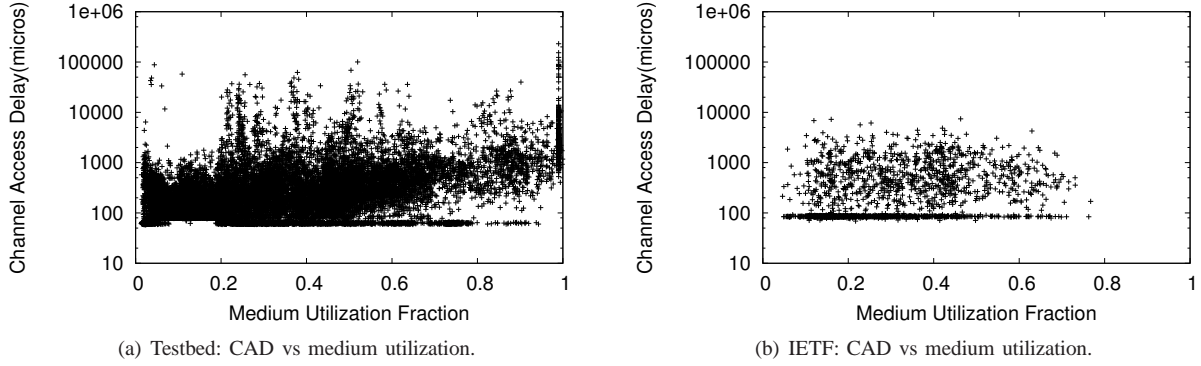


Fig. 2. Correlation between CAD (active probe technique) and medium utilization.

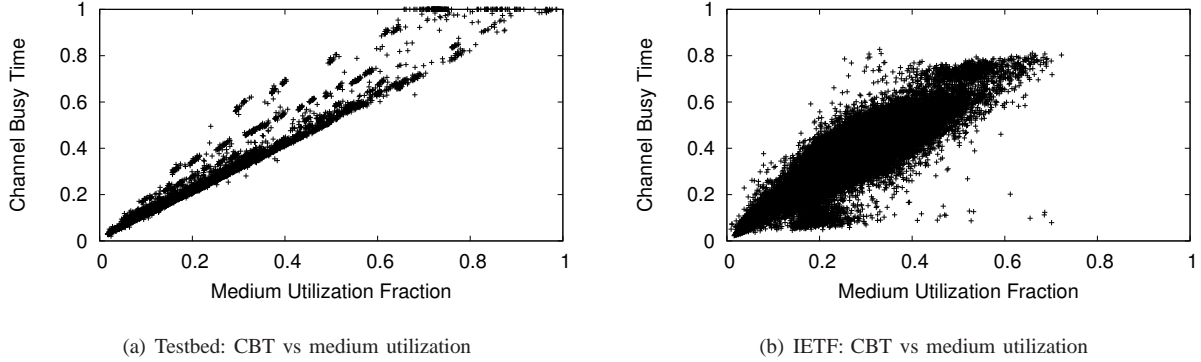


Fig. 3. Correlation between CBT (passive measurement technique) and medium utilization.

known distribution, to determine whether the current medium utilization is above or below a specified threshold value. Channel busy time measured during an interval bears a direct correlation with the medium utilization, and predicts a range for the current medium utilization level.

Due to the difference in the nature of results obtained from each of these techniques, we do not compare the two quantitatively. We first present the results for CAD in both test environments followed by those for CBT.

4.3.1 Channel Access Delay

As explained in Section 3, the channel access delay for a packet depends on the instantaneous state of the network when the measurement was made. For example, if the device driver delivers a packet to the hardware for transmission during an ongoing packet transmission in the channel, then the CAD value depends on the time required for the ongoing transmission to finish. As can be seen in Figures 2(a) and (b), for a given medium utilization level, individual CAD values observed show no obvious trends. The exception is the lower bound on the measured CAD values ($\approx 80 \mu\text{s}$), which corresponds to the minimum channel access delay observed if the medium is idle at the instant when the probe packet is delivered to the hardware for transmission. Figures 2(a) and (b) show average CAD values over one second intervals for four probe packets (98 bytes each) sent at a data rate of 54 Mbps.

While individual CAD values are susceptible to noisy estimates, the BWS technique allows us to estimate the channel conditions based on a distribution of samples taken during an

interval. The BWS test compares two distribution samples and assigns a probability measure (p -value) to the event that the two samples originate from the same underlying distribution.

We first train our prediction system during a training phase, in which we obtain an expected distribution for each 10% bin of medium utilization values ranging from 0 to 100% ($bin_{(0,10)}, bin_{(10,20)}, \dots, bin_{(90,100)}$). In the real-time experiment, we obtain a distribution d of the CAD values from the active probe packets and use the nonparametric BWS test to obtain a p -value for the event that d and $bin_{(i,j)}$ have the same underlying distribution. Next, we choose the bin $bin_{(a,b)}$ with the highest p -value and determine whether the range (a,b) is above or below the specified threshold (T_c) for medium utilization that defines congestion. If the range (a,b) lies above the threshold T_c , we declare the medium to be congested and un-congested otherwise. We verify the accuracy of our threshold-based congestion estimation by determining whether the value of medium utilization obtained from the sniffer during post-analysis was also observed to be above or below the medium utilization threshold T_c .

Table 2 shows the accuracy of the real-time made by the the CAD congestion estimation tool, in both test scenarios, for varying CAD values collected over one second intervals. The accuracy of the BWS test predictions was slightly higher in the testbed environment as compared to the IETF. This is because the number of CAD samples collected in the testbed was higher (10 packets/second) than the IETF experiment (4 packets/second).

In conclusion, the accuracy of the BWS test results varies

Medium Utilization Threshold (%)	BWS accuracy (%)	
	Testbed	IETF
10	64.69	67.63
20	70.39	65.89
30	76.09	63.58
40	77.50	57.51
50	83.22	57.22
60	88.11	69.79
70	92.78	81.21
80	94.23	85.84
90	95.65	94.50
100	100	100

TABLE 2
BWS test prediction accuracy with varying medium utilization threshold values.

depending on the number of CAD samples available during an interval. In a general setting, we expect a node to calculate the channel access delay for a majority of its transmitted packets, which will yield a sufficiently large number of CAD values within a short interval. However, in our experiments at the IETF meeting, we limited the number of packets sent by the test nodes to a maximum of 4 packets/second to limit the impact of our experiment on the network.

4.3.2 Channel Busy Time

In Figures 3(a) and (b), we plot the CBT metric against the medium utilization calculated based on sniffer data for each second, for experiments conducted on the testbed and at the IETF meeting, respectively. Every point in the graph represents the measured CBT value compared to the calculated medium utilization value during the corresponding time interval. Both Figures 3(a) and 3(b) show a strong linear correlation between CBT and medium utilization, with a linear correlation coefficient of 0.97 for the testbed network and 0.925 for the IETF network. This high degree of correlation indicates that channel busy time estimates the medium utilization with high accuracy.

From the graphs, we observe that the CBT metric sometimes indicates a higher value than medium utilization. This behavior is because CBT accounts for the time during which the medium was busy, but a packet was not necessarily received (e.g., channel noise, packet collisions). Therefore CBT represents a more accurate picture of the channel in such scenarios. Also, it can be seen from Figure 3(b) that the CBT metric sometimes under-estimates the channel utilization value. The specification for the Atheros chipset quotes the radio sensitivity for some data rates (e.g., -95dBm for 1Mbps) to be lower than the CCA threshold. Thus, some low data rate packets are received correctly at the sniffer at a signal strength that is below the CCA threshold.

4.4 Discussion

The results in the previous section indicate that channel busy time is an effective technique to determine channel utilization at a low overhead. Channel access delays experienced by a node can be used to estimate whether medium utilization is high or low depending on a specified threshold value. While

the results of the CAD technique do not provide us with the exact value of medium utilization, the decision on whether the medium utilization is above or below any specified threshold is sufficient for most applications involving rate adaptation, admission control and network debugging. On the other hand, the CBT metric provides a medium utilization estimation with high accuracy, using a feature exported by the Atheros-based 802.11 devices.

For its ease of use and low overhead, we use the CBT metric in the rest of the paper to design a novel congestion-aware rate adaptation scheme for wireless networks. However, in scenarios where the CBT metric functionality is not available in the 802.11 cards, the scheme could be easily modified to use the channel access delay metric.

5 WIRELESS CONGESTION OPTIMIZED FALLBACK (WOOF)

We now demonstrate the utility of real-time congestion metrics in improving the performance of wireless networks in congested scenarios. Our focus is on rate adaptation in wireless networks. In the following sections, we analyze the performance of rate adaptation schemes in a large WLAN connected to the Internet. Based on this analysis, we then describe the design of our congestion-aware rate adaptation scheme.

5.1 Rate Adaptation during Congestion

We now analyze the behavior of current rate adaptation schemes in a congested network. Our focus is on the packet loss rates in such networks and their impact on rate adaptation. In addition, we explore the relationship between packet loss and congestion levels in the network. The traffic traces from the 67th IETF are used for this analysis.

We focus on the Wednesday plenary session of the IETF meeting. This session had more than 1000 attendees in one large room with 16 APs. We choose this session in order to study the packet loss behavior in a network with a high number of users and a high load on the network. We assume the original transmission of a packet to be lost if, in the trace, we observe a packet transmission with the retry flag set. This technique, however, does not account for retransmitted packets that were not captured by the sniffer. Thus the estimate is a lower bound for the number of packet losses. The fraction of lost packets is calculated as the ratio of the number of retransmitted packets to the sum of the number of packets transmitted and the number of packets lost.

Figure 4 plots the medium utilization levels and the fraction of data frames that were lost during the Wednesday plenary session. The medium utilization fraction is calculated with the same technique as used in Section 4.1. During periods of high utilization, the number of packet losses also increases. This can be attributed to the losses caused by contention for the medium (i.e., when the backoff counters of two or more nodes expire at the same time.) Alarming, the percentage of lost packets is as high as 30%. With such a high number of packet losses, any rate adaptation scheme that relies on packet loss as a link quality metric is highly likely to lower the data rate, often to the minimum possible transmission rate.

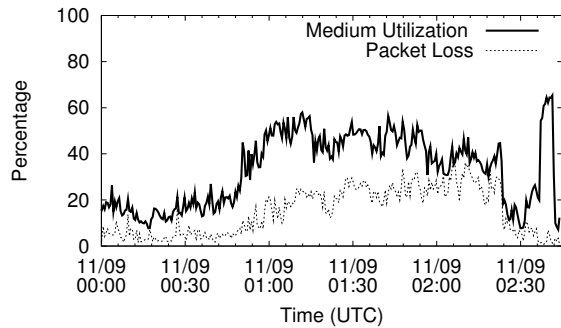


Fig. 4. Medium utilization and packet loss rate in a congested 802.11 network.

To analyze the impact of such high packet loss rates on rate adaptation schemes, we study the distribution of data rates used for transmissions. The access points at the IETF meeting advertised only the following data rates (in Mbps) as supported: 11, 12, 18, 24, 36, 48, and 54. A client that supports IEEE 802.11b only is limited to use the 11 Mbps data rate alone and thus cannot perform rate adaptation. To study the distribution of data rates, we consider only the data packets sent to/received from clients that support IEEE 802.11g. We consider a client to be 802.11g-enabled if a) it specifies an 802.11g data rate in the association message, or b) in the entire traffic trace, we observe at least one packet to/from the client using an IEEE 802.11g data rate. Table 3 shows the distribution of data rates for only the 802.11g clients observed during the session. We see that a majority of the transmissions (73%) used the lowest possible data rate². This behavior can be attributed to the rate adaptation schemes used by the wireless devices in the network. The high rate of packet loss forces the rate adaptation scheme to consider the link to be of poor quality and, thus, use lower data rates. A study of the SNRs shows that during this period 67% of the 11Mbps transmissions had higher SNR than the average SNR of a 54Mbps transmissions. This shows that higher data rates could be used in this scenario.

Previous work has also observed a similar effect of congestion on rate adaptation [3], [24]. In a congested network, a majority of the 802.11 transmissions occur at the lowest possible rate. Such transmissions also consume a large fraction of the medium time, since the packets take longer to be transmitted. Switching to a lower rate as a result of contention losses is not only unnecessary but also increases the medium utilization. The packet transmissions take longer to complete and are more susceptible to collisions (e.g. from hidden terminals). The above problem of rate adaptation is similar to the behavior of TCP reducing its congestion window in response to all types of packet losses, which leads to reduced throughput even though the losses are not related to congestion [3]. Thus, it is important to understand the cause of a packet loss, and respond

2. An 802.11g capable client may have been incorrectly classified as an 802.11b client if it used only the 11 Mbps data rate and the association message was not captured by the sniffer. Accurate classification of such clients would increase the fraction of data packets at 11 Mbps.

Rate (Mbps)	Percentage of Data Packets
1	0%
2	0%
5.5	0%
6	0%
9	0%
11	72.94%
12	3.95%
18	1.53%
24	2.76%
36	3.90%
48	3.59%
54	11.51%

TABLE 3

Data rate distribution for 802.11g clients during the Wednesday plenary session.

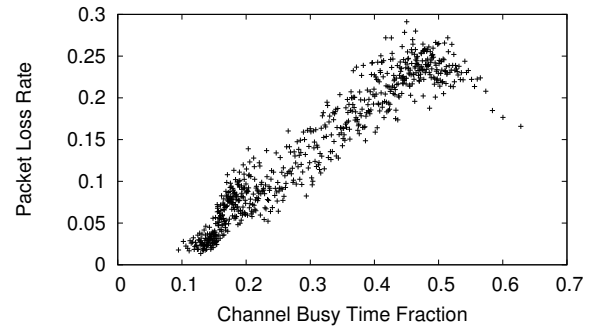


Fig. 5. Relationship between channel busy time and packet loss rate during the Wednesday plenary session.

appropriately in the rate adaptation algorithm.

Based on the above discussion, we conclude that rate adaptation schemes must identify the cause of a packet loss and account only for packet losses that are not congestion-related. To this end, we now discuss the design and implementation of Wireless cOngestion Optimized Fallback (WOOF), a rate adaptation scheme that identifies the cause of packet losses. Packet losses related to congestion are omitted in the determination of an appropriate transmission data rate. Thus the decision relies only on losses due to poor link quality.

5.2 Identification of Congestion-related Packet Loss

In Section 4 we noted that channel busy time was a good predictor of network congestion levels. We now explore the relationship between the channel busy time metric and packet loss rate.

Figure 5 plots a graph of the packet loss rate as a function of the Channel Busy Time during the corresponding time interval of the Wednesday Plenary session. The plotted rates are averaged over 30s time windows. In other words, a point (x,y) represents a 30s window wherein the with x Channel Busy Time and y packet loss rate. We observe a strong linear correlation with the packet loss rate and the observed channel busy time values. In other words, as the channel busy time increases, the probability of a packet loss due to congestion also increases.

Unfortunately, a similar study of packet loss versus channel busy time values for other sessions in the 67th IETF did not

exhibit such strong correlation. However, we note that the average packet loss rate was higher during periods of high utilization in these sessions. These observations lead us to conclude that the channel busy time information can be used as a good indicator of packet loss caused by the congestion level in the network. However, the exact relationship of channel busy time (and therefore medium utilization) may vary depending on the environmental factors in the wireless network. A rate-adaptation scheme that uses channel busy time as a heuristic to identify congestion-related packet losses must therefore be dynamic and capable of adapting to changes in the wireless network environment. In the design of our rate adaptation scheme WOOOF, we initiate our prediction heuristic with the initial setting of a linear relationship between packet loss and observed utilization level. We then dynamically adapt the weight of this relationship based on the observed network performance to model the current environment in the wireless network.

The channel busy time metric only helps in identifying the cause of packet loss, i.e., whether it was congestion-related. The rate adaptation scheme must continue to deal with packet losses caused by other factors such as poor link quality. Thus we claim that Channel Busy Time provides supplementary information that a rate adaptation scheme can use in addition to packet loss information. We, therefore, borrow the basic framework of the design of SampleRate [12] scheme in order to handle the packet loss information in WOOOF. WOOOF builds on SampleRate through the incorporation of channel busy time and its relationship with congestion related packet loss. We now outline the operation of SampleRate, and then discuss the design of WOOOF.

5.3 SampleRate

SampleRate is a rate adaptation scheme that accounts for the time required for successful transmission of a packet [12]. The underlying idea of SampleRate is to choose the data rate that is expected to require the least time for transmission, i.e., the data rate with maximum throughput. Note that this rate need not always be the highest possible rate (i.e., 54 Mbps) because of poor link SNR and variable link quality. SampleRate uses frequent probing of different data rates in addition to the currently used data rate to calculate the Expected Transmission Count (ETX) [25] for each data rate. The ETX represents the average number of transmission attempts required for successful reception of a packet. A link has $ETX=1$ if a packet can be successfully received on the first transmission attempt. On the other hand, if the packet is lost and subsequent retransmissions are required for successful packet delivery, then $ETX>1$. The ETX is calculated using either a sliding-window time average or using EWMA. The Expected Transmission Time (ETT) is calculated using ETX information at a given data rate and accounts for the backoff times when the ETX metric predicts that a retransmission is required (i.e., $ETX>1$). SampleRate then chooses to transmit data packets using the data rate with the lowest expected transmission time.

While SampleRate is able to successfully adapt the data rate in the presence of link variability, it does not respond

appropriately when congestion occurs. In particular, it does not distinguish the cause of packet loss; all packet losses contribute towards the calculation of ETX. Previous research has observed this phenomenon of SampleRate’s data rate reduction [26]. Congestion losses impact SampleRate’s estimation of ETX at the different data rates and lead to a sub-optimal choice of transmission rate.

5.4 Design of WOOOF

We base the design of the WOOOF scheme on the design of SampleRate. In particular, we build on SampleRate’s framework of calculation of Expected Transmission Time and use this information to choose an appropriate data rate for transmission. In addition, we incorporate the ability to discern the cause of packet loss, in order to enable operation in congested networks.

In Section 5.1 we observed that channel busy time can be used as a metric to predict congestion-related packet loss. We incorporate this insight into the design of WOOOF with the following enhancement to SampleRate. We use *effective packet loss* instead of the observed packet loss for calculation of ETX and the resulting calculation of ETT. Whenever we observe a packet loss, we associate a probability P_{CL} that the loss was due to congestion. We then account for the fraction of packet loss that was not due to congestion in the calculation of ETX. In other words, we weigh every packet loss proportionally to the probability that it was not a congestion-related loss.

$$EffectiveLoss = ObservedLoss \cdot (1 - P_{CL})$$

For the calculation of P_{CL} , we use the following equation to capture the relationship between Channel Busy Time and packet loss:

$$P_{CL} = \beta \cdot CBT$$

where CBT represents Channel Busy Time fraction and β represents the confidence factor, $0 \leq \beta \leq 1$. The Channel Busy Time values are measured over intervals of time of size W seconds.

The confidence factor β is a measure of the degree of correlation between $CBTF$ and congestion-related packet loss. The confidence factor is adaptively varied based on the observed network performance. The value of β is calculated as follows. At the end of each measurement interval, W , we compare the performance of rate adaptation in the current interval to that during the previous interval. The metric for performance comparison is the transmission time consumed during the interval. To enable comparison of transmissions using a diverse set of data rates, we normalize the measured transmission time with respect to the corresponding time using a fixed data rate on a reliable channel, e.g., 54 Mbps. In other words, the metric is analogous to the transmission time required per byte of successfully transferred data. If the metric indicates an improvement in performance in comparison with the previous interval of measurement, the value of β is increased in steps of 0.05. This increase in β models the increased confidence in using $CBTF$ to distinguish packet losses due to congestion. Similarly, when the metric indicates a drop in network performance, β is decreased in steps of 0.05. The confidence factor

β enables WOOF to adapt to different network environments. In particular, this enables WOOF to ensure good performance (at least as good as SampleRate) in situations of low SNR links and high congestion. In Section 6.5, we examine the impact of the measurement window, W , and its effect on the convergence time for β values. In Section 6.3, we evaluate the performance of WOOF under different combinations of link SNR and congestion

5.5 Implementation

We implemented WOOF as a rate adaptation module for the MadWifi driver v0.9.2 for Atheros chipsets on Linux. We choose $W = 1$ s as the window of observation and recalibration. A large value of W reduces the responsiveness of WOOF to changes in the environment utilization. Smaller values of W increase the load on the driver due to the need for frequent recalibration. We set the initial value of β to 0.5. At each interval of W seconds, the driver reads the Atheros registers described in Section 3.2 to calculate the Channel Busy Fraction. In addition, the normalized network performance is calculated as described in Section 5.4. The β values are also updated at each interval. In the following section we use our implementation of WOOF to study the benefit of WOOF in a congested wireless network.

6 EVALUATION

We evaluate the performance of WOOF in two testbed networks as well as through simulation. The testbed networks represent two scenarios, a WLAN and a multihop mesh network. These testbeds help us to evaluate WOOF on real 802.11 devices and networks. The simulations enable us to scale the performance evaluation to networks larger than the testbed networks. We first present results from the testbed experiments, followed by the simulation-based experiments.

Among the two testbeds, we first use the WLAN scenario since it allows us to control the experiment parameters and the environment. The WLAN consists of one laptop acting as an AP and eight laptops as client devices. Each laptop is equipped with an IEEE 802.11b/g radio based on the Atheros chipset. The laptops use Linux (kernel version 2.6) as their OS. The wireless radio is controlled by the MadWifi driver v0.9.2 along with the WOOF rate adaptation module.

We compare the performance of WOOF against that of SampleRate. Previous work has shown that SampleRate performs better than ARF and AARF in most network scenarios [12], [13]. Thus we expect WOOF to provide better performance than ARF and AARF in all cases where WOOF performs better than SampleRate³. We also compare the performance of WOOF with that of CARA [6]. As described in Section 2, CARA is built upon ARF, and uses RTS-CTS to combat collision losses. We implement CARA for Madwifi and use it for our comparison. In addition, for the WLAN scenario,

3. Implementation of RRAA [13] requires a specialized programmable AP platform. Therefore, we are unable to compare WOOF against RRAA. However, we note that RRAA was designed for better performance in hidden terminal scenarios and not specifically for congested networks.

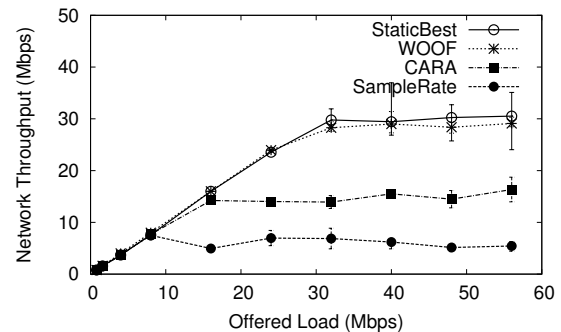


Fig. 6. Impact of network load.

we also compare performance against a scenario wherein the data rate of the client-AP link is fixed at the best possible rate. This scenario, called the StaticBest scenario, gives us an estimate of the upper-bound on the network performance. The best static rate is determined by running a simple performance test at each data rate immediately prior to the corresponding tests with SampleRate, CARA, and WOOF.

6.1 Impact of Network Load

In the following set of experiments, we examine the impact of network load on the rate adaptation schemes. The clients implement either SampleRate, CARA, WOOF or use the fixed data rate (StaticBest). The load on each of the eight clients is varied from 100 Kbps to 7 Mbps to vary the overall load on the network from 800 Kbps to 56 Mbps. The AP operates using 802.11b/g and thus the maximum theoretical raw bandwidth of the network is 54 Mbps. However the mandatory MAC and PHY layer overheads limit the achievable network throughput to lower values. The network performance for each offered load is measured using the `iperf` utility and UDP traffic with 1500 byte packets for 5 minutes. For each trial of the experiment, the drivers on the AP and clients are reset. This is followed by an initial warm-up period of 60 seconds for each client during which clients transmit low-rate traffic (10Kbps) to the AP.

Figure 6 graphs the total network throughput as a function of the offered load. Each data-point is an average based on five trials of the experiment. The error-bars indicate the minimum and the maximum throughput values over different experiment trials. We observe that the network throughput for StaticBest saturates at about 32 Mbps and for Sample-Rate at 7 Mbps. The throughput for WOOF is around 29 Mbps, close to that of Static-Best. From the graph, we observe that for non-congested scenarios (offered load < 8 Mbps), all four schemes are able to sustain the offered load. In other words, WOOF matches the performance of the other schemes in low congestion environments. With the increase of congestion (offered load > 8 Mbps), SampleRate is affected by the congestion-related packet losses and, thus, begins to use lower data rates. WOOF correctly identifies these packet losses as congestion-related and continues to use high data rates, resulting in better throughput. CARA provides higher throughput than SampleRate, but less than that of WOOF. CARA identifies congestion-related losses, uses RTS-CTS

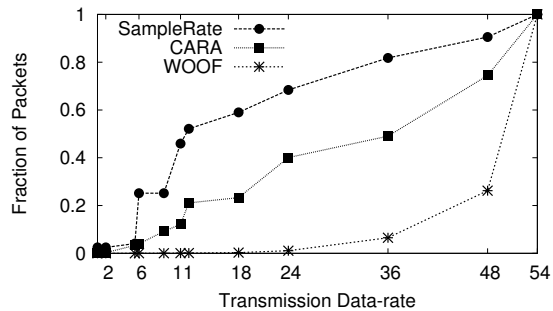


Fig. 7. Distribution (CDF) of transmission data rates in a representative trial of the experiment.

to protect transmissions at higher data rates, and obtains more throughput than SampleRate. However, the additional overhead of the RTS-CTS handshake restricts the network throughput to less than that of WOOF.

Figure 7 plots a CDF of the data rates used in a representative trial of the experiment with an offered load of 40 Mbps. The graph shows that a majority of the packet transmissions with WOOF use high data rates of 48 Mbps and 54 Mbps. On the other hand, SampleRate transmits about 50% of the packets using 11 Mbps or lower data rates. We note that although CARA uses higher data rates for transmissions, the overall throughput is less than that of WOOF. This, again, points to the overhead of the RTS-CTS handshake at the 1 Mbps data rate to avoid the collision of a data packet at a higher data rate.

6.2 Impact of the Number of Clients

We now examine the impact of contention in the network and study the network performance as the number of clients increases. The experimental configuration is similar to the one described in the previous section. In this case, however, we incrementally increase the number of clients associated with the AP from one to eight. Each client offers a load of 10 Mbps UDP traffic.

Figure 8 plots a graph of the total network throughput versus the number of clients in the network. At low contention levels (<4 clients), we observe that the throughput of SampleRate increases almost linearly to reach a maximum of about 24 Mbps. Once the network starts to become congested (≥ 4 clients), however, the average throughput for SampleRate starts to drop. With eight clients, the throughput for SampleRate is 7 Mbps. This drastic reduction in network throughput (about 70%) is because, with increased contention, SampleRate reduces the data rate and adds to the congestion. In contrast, the drop in throughput for WOOF is from 33 Mbps to 30 Mbps, i.e., only a 10% reduction. We observe that the throughput reduction for StaticBest is also about 10%. Therefore, we conclude that the reduction in throughput is primarily due to actual packet losses. WOOF is successful in identifying congestion-related packet losses and omitting them from the ETX calculations. On the other hand, SampleRate does not attempt to identify these losses as congestion-related and accounts for them in its ETX calculations, thereby lowering the transmission data rates and actually increasing the amount of congestion. On the other

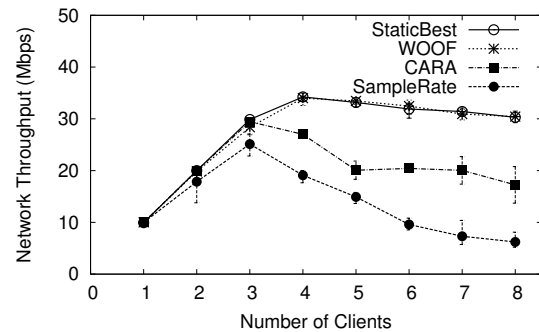


Fig. 8. Impact of number of clients.

hand, the use of RTS-CTS limits the throughput improvement for CARA.

6.3 Performance in Poor Link Conditions

We now conduct experiments to understand the performance of WOOF under different network conditions. In particular, we are interested in the scenarios wherein the links are weak i.e., the SNR of received packets is low. We conduct experiments similar to that in Section 6.2. We consider four different combinations of link SNR and congestion levels. The good SNR link scenario has all client links with sufficient SNR to operate at 48 and 54 Mbps. The low SNR scenario is achieved by increasing the physical distance between the clients and the AP, and decreasing the transmit power of all the radios. The StaticBest rates for the clients in this scenario range between 2 Mbps and 18 Mbps. We chose two congestion levels: low congestion corresponds to two clients with an offered load of 5 Mbps each; high congestion corresponds to eight clients with offered load of 5 Mbps each.

Table 4 lists the network throughput in each of the scenarios for both SampleRate and WOOF. We see that the performance of WOOF under low congestion is comparable to that of SampleRate. During high congestion, we observe that WOOF improves the network throughput for both SNR scenarios. Therefore, we conclude that WOOF provides performance gains in congested networks while having minimal impact in uncongested networks. Further, WOOF responds appropriately when the link quality is poor by decreasing the data rate to a rate more suitable to the poor link quality.

	Low SNR	High SNR
Low Congestion	SampleRate: 0.79 WOOF: 0.73	SampleRate: 7.67 WOOF: 7.45
High Congestion	SampleRate: 0.55 WOOF: 0.79	SampleRate: 10.63 WOOF: 23.04

TABLE 4
Network throughput (in Mbps) under different combinations of SNR and congestion levels.

6.4 Performance in a Mixed Network

In this next experiment, we evaluate the gains obtained through incremental deployment of WOOF. The experimental configuration is similar to the one described in Section 6.1. We

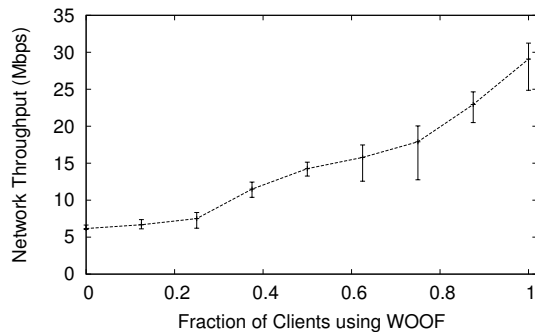


Fig. 9. Network throughput with a mix of WOOF and SampleRate clients.

hold the number of active clients constant at eight and we vary the number of these clients that use WOOF. The non-WOOF clients in the network use SampleRate. Each client has a fixed offered load of 10 Mbps, and therefore, the overall load exceeds network capacity. Figure 9 plots the network throughput as a function of the fraction of clients that use WOOF. The left-most point on the curve (zero WOOF clients) represents the scenario where all the clients use SampleRate.

We observe that the overall network throughput improves as the fraction of WOOF clients increases, i.e., the incremental use of WOOF provides network performance gains. We also note that the change in throughput of the individual WOOF clients (not shown in the figure) does not always account for the increase in overall network throughput. In a few cases, the SampleRate clients obtained more throughput than the WOOF clients. This behavior is due to the medium contention mechanism in IEEE 802.11. Nodes in a 802.11 network contend for the medium on a per-packet basis, irrespective of the data rate or size of the packet. A WOOF client that transmits at a higher data rate consumes less medium time for a packet transmission. The extra time available enables contention resolution for more packets in the network, for both WOOF clients and non-WOOF clients. Thus we see an increase in the overall throughput of the network.

6.5 Choice of Parameter W

We now explore the impact of using different values for W , the interval of recalibration for WOOF. We use the same experimental configuration as in Section 6.1. Each of the eight clients has an offered load of 10 Mbps for a five minute duration. Table 5 shows the average network throughput for different W values. We observe that for low W values, between 0.25s and 2s, the network throughput remains high and fairly stable. For $W > 2$ s, we see that the throughput values decrease. At high values of W , the throughput is comparable to that obtained by SampleRate. A low value of W enables WOOF to adapt to network conditions quickly and obtain better performance. However, a low value of W also increases the processing load due to the rate adaptation algorithm. On the other hand, a high value of W makes WOOF less responsive to the environment. Based on these tradeoffs, we recommend a value of $W = 1$ s.

Closely related to the choice of value of W is the number of recalibration cycles required for the β value to stabilize

W (seconds)	Throughput (Mbps)
0.05	17.68
0.10	21.43
0.25	28.77
0.5	27.63
1	28.85
2	27.72
4	21.98
8	16.44
16	14.92
32	10.30

TABLE 5

Impact of measurement interval W .

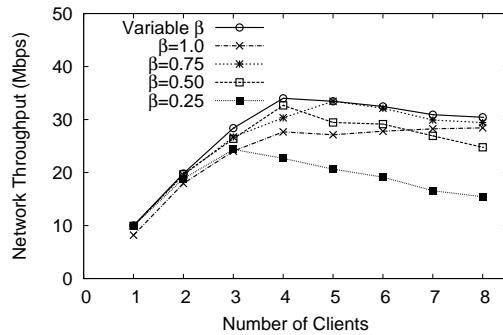


Fig. 10. Impact of β parameter.

in response to a change in the environment. In our WLAN testbed we found that the median number of cycles for β to stabilize is six. Similarly, in the MeshNet environment that we describe in the next section, the median number of cycles was five. Together with W , the number of cycles for β to stabilize impacts the time delay for WOOF to respond to a change in the environment (e.g. arrival of a new node in the network).

6.6 Impact of parameter β

We now demonstrate the importance of the confidence factor β in adapting to different network conditions. We use the experiment setup of Section 6.2. We increase the number of clients associated with the AP, and each client offers a load of 10Mbps. We repeat the experiment with fixed values of β as well as adaptive β . Figure 10 shows the results of these experiments. We observe that the throughput of each β value peaks with different number of clients. On the other hand, adaptive β is able to provide the best throughput with the different number of clients. Therefore, we conclude that the relationship between CBT and congestion-related packet losses, as captured by the factor β , varies with the network scenario. Further, the results highlight the importance of varying β based on observed network performance.

6.7 Performance in a Mesh Network

Having obtained insight into the different performance aspects of WOOF in the WLAN environment, we conduct a set of experiments in an uncontrolled mesh network. The purpose of the experiments is to understand the performance of WOOF in real multi-hop network deployments. We conduct our experiments on the UCSB MeshNet testbed [27]. The MeshNet

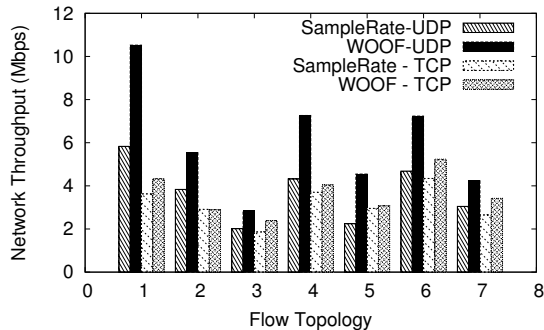


Fig. 11. Network throughput with UDP and TCP for different flow topologies in the UCSB MeshNet.

is an indoor multihop IEEE 802.11 network with 25 dual-radio devices. For our experiments, we use a subset of these nodes connected to a single gateway node. We use only one radio of each node operating in the 802.11b/g mode. SRCR [28] is used as the routing protocol. The physical distance between the nodes and the presence of barriers in the form of walls and doors result in a majority of the links operating at low data rates, even in the absence of competing traffic. The median number of neighbors for MeshNet nodes is three.

We study the performance of the network by measuring the sum of throughputs achieved by the individual nodes in the network. To model the flow behavior in a mesh network, all the flows originate from the gateway node. The number of flows and the destination node for each flow is chosen randomly, but we ensure that there are a minimum of three flows in the network at all times. A combination of the selected number of flows and the corresponding destination nodes constitutes a flow topology. The experiment is conducted for seven different flow topologies, and for both SampleRate and WOOF. We repeat the experiment for both TCP and 10 Mbps UDP flows.

Figure 11 compares the throughput of SampleRate and WOOF for these experiments. From the graph we see that WOOF provides higher network throughput for both UDP and TCP as compared to SampleRate. The median increase in throughput for UDP is 54.49%. The throughput gains for TCP, however, are less pronounced, with a median improvement of 20.52%. This behavior can be attributed to the dynamics of TCP congestion control mechanisms and its sensitivity to packet loss.

6.8 Simulation-based Evaluation

To better understand the performance of WOOF in a wider variety of networks, we use the Qualnet simulator [29]. In particular, we are interested in the performance of WOOF in scenarios similar to those found in the IETF network, e.g., the plenary session with hundreds of clients connected to a single AP.

Our implementation of WOOF for Qualnet consists of three main components. First, we extend the 802.11 MAC implementation to consult a rate adaptation module to select a data rate for packet transmissions. We implement SampleRate as the base rate adaptation algorithm. Second, we implement the Channel Busy Time metric by tracking the durations of packet transmissions, packet receptions, and busy channel scenarios

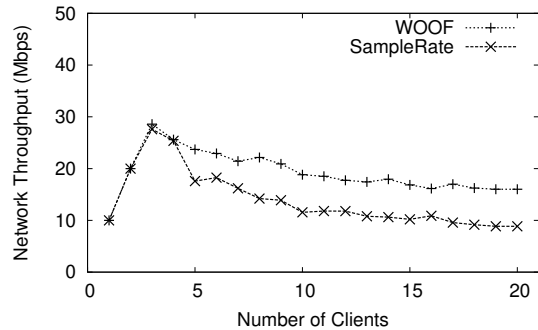


Fig. 12. Simulation-based evaluation of network performance with increasing number of clients.

at each node. Third, we implement WOOF by extending the base SampleRate module.

We first validate our Qualnet implementation of SampleRate and WOOF by simulating a scenario similar to our experimental setup in Section 6.2. A key difference in the simulation setup is that Qualnet supports only pure 802.11b or pure 802.11g networks. In other words, the 802.11b/g mixed mode operation of the Atheros radios cannot be fully captured by the simulator. Therefore, we choose to perform rate selection among the eight data rates of 802.11g (6 Mbps to 54 Mbps) rather than the 12 data rates of 802.11b/g (1 Mbps to 54 Mbps). We use the default parameters provided by Qualnet for all the 802.11g nodes in the simulation, as listed in Table 6. We disable the use of RTS-CTS to mimic our testbed network. Similar to the experiment in Section 6.2, we simulate a WLAN environment with one AP and an increasing number of clients, each with 10 Mbps offered load.

Parameter Name	Value
PHY	IEEE 802.11g
DIFS	40 μ s
SIFS	16 μ s
Slot Time	9 μ s
Data Rates	6, 9, 12, 18, 24, 36, 48, 54 (Mbps)
Transmit Power @6Mbps	20 dBm
Receiver Sensitivity @6Mbps	-85 dBm

TABLE 6
Simulation Parameters.

Figure 12 plots the average network throughput for 10 trials of these experiments. From the graph, we observe that the overall trends obtained from the simulation are similar to that of the testbed. WOOF consistently provides higher network throughput, even in the presence of 20 contending clients. For example, WOOF provides about 6.2 Mbps more throughput than SampleRate. We note that the drop in throughput for SampleRate is not as steep as observed in the testbed experiments. This is because the lowest possible data rate in the simulation is 6 Mbps compared to 1 Mbps in the testbed. In the testbed, the use of lower data rates decreases the effective network capacity, and results in reduced throughput.

Next, we evaluate the scalability and performance of WOOF in a large WLAN with hundreds of clients. In this experiment, we characterize the gains obtained with the use of WOOF in

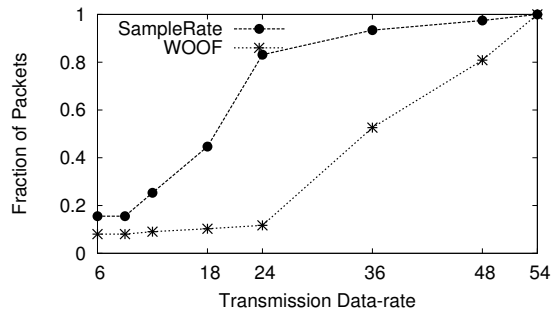


Fig. 13. Distribution of data rates used in the simulation of the IETF meeting Plenary session.

terms of the reduction in channel utilization. For this purpose, we refer to the Wednesday plenary session of the 67th IETF meeting described earlier. We consider the traffic on one particular channel (channel 6), and use it as a traffic trace to input to the simulator. In other words, for every packet found in the trace we schedule an equivalent transmission in the simulation. However, the traffic trace was captured by a single sniffer from actual transmissions on the channel. The trace therefore is the result of contention-resolution algorithms used by the devices in the network and therefore represents a perfect collision-free transmission schedule. In order to create contention among the packets in the trace, we perturb the packet generation time to be a random value within a time window of 5ms before the actual time found in the trace. We choose a representative one hour of the meeting for simulation. Each MAC address in the trace (except broadcast and multicast addresses) is represented by a node in the simulation. There were 592 unique MAC addresses in the chosen trace. The location of the nodes is chosen randomly. However, we ensure that all the nodes are in communication range of each other, at least when communicating using the lowest rate of 6 Mbps. We conduct the experiment with both SampleRate and WOOF as the rate adaptation algorithms.

We observe the data rates used by each algorithm. We also record the total time used for transmissions, i.e., the medium utilization of each algorithm. Figure 13 plots the CDF of the data rates used by SampleRate and WOOF. We observe that WOOF uses higher data rates more often than SampleRate. This is because WOOF is able to incorporate the CBT information in decision-making and avoid switching to lower data rates during congested periods. The medium utilization for WOOF was 82% of that for SampleRate. We conclude that WOOF provides savings in network resource consumption, and therefore reduces congestion.

7 CONCLUSION

Congestion in an IEEE 802.11 wireless network causes drastic reduction in network performance. Critical to tackling this problem is the ability to identify and measure congestion. In this paper we presented two techniques, an active technique (CAD) and a passive technique (CBT), that measure the utilization of the wireless medium in real-time. We then used the CBT measurement technique to develop a rate adaptation scheme, WOOF, for IEEE 802.11. Performance evaluation

show up to a three-fold gain in throughput in a congested network. Simulations demonstrated the utility of using WOOF in a large WLAN. In addition to our congestion-aware rate adaptation algorithm, we believe that the measurement techniques proposed in this paper can be used to design new protocols or solutions that perform well under congested scenarios. For example, the CBT metric can be used for bandwidth estimation to facilitate effective flow admission control in wireless networks.

ACKNOWLEDGMENTS

This work is supported in part by NSF Wireless Networks award CNS-07220275 and a grant from Intel Corporation.

REFERENCES

- [1] J. Horrigan. (2007, Feb) Memo on Wireless Internet Access. Pew Internet & American Life Project. [Online]. Available: http://www.pewinternet.org/pdfs/PIP_Wireless.Use.pdf
- [2] A. P. Jardosh, K. Mittal, K. N. Ramachandran, E. M. Belding, and K. C. Almeroth, "IQU: Practical Queue-Based User Association Management for WLANs," in *Proc. of MobiCom*, Los Angeles, CA, Sep 2006.
- [3] M. Rodrig, C. Reis, R. Mahajan, D. Wetherall, and J. Zahorjan, "Measurement-based Characterization of 802.11 in a Hotspot Setting," in *Proc. of Workshop on Experimental Approaches to Wireless Network Design and Analysis (EWIND)*, Philadelphia, PA, Aug 2005.
- [4] A. Jardosh, K. Ramachandran, K. Almeroth, and E. Belding-Royer, "Understanding Congestion in IEEE 802.11b Wireless Networks," in *Proc. of Internet Measurement Conference*, Berkeley, CA, Oct 2005.
- [5] P. Acharya, A. Sharma, E. Belding, K. Almeroth, and K. Papagianaki, "Congestion-Aware Rate Adaptation in Wireless Networks: A Measurement-Driven Approach," in *Proc. of SECON*, San Francisco, CA, Jun 2008.
- [6] J. Kim, S. Kim, S. Choi, and D. Qiao, "CARA: Collision-aware Rate Adaptation for IEEE 802.11 WLANs," in *Proc. of INFOCOM*, Barcelona, Spain, Apr 2006.
- [7] X. Yang and N. Vaidya, "On the Physical Carrier Sense in Wireless Ad Hoc Networks," in *Proc. of INFOCOM*, Miami, FL, Mar 2005.
- [8] G. Holland, N. Vaidya, and P. Bahl, "A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks," in *Proc. of MobiCom*, Rome, Italy, Jul 2001.
- [9] G. Judd, X. Wang, and P. Steenkiste, "Efficient Channel-aware Rate Adaptation in Dynamic Environments," in *Proc. of MobiSys*, Breckenridge, CO, Jun. 2008.
- [10] A. Kamerman and L. Monteban, "WaveLAN II: A High-Performance Wireless LAN for the Unlicensed Band," *Bell Labs Technical Journal*, vol. 2, no. 3, pp. 118–133, Aug 1997.
- [11] M. Lacage, M. Manshaei, and T. Turletti, "IEEE 802.11 Rate Adaptation: A Practical Approach," in *Proc. of ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, Venice, Italy, Oct 2004.
- [12] J. Bicket, "Bit-rate Selection in Wireless Networks," Master's thesis, Massachusetts Institute of Technology, 2005.
- [13] S. H. Y. Wong, S. Lu, H. Yang, and V. Bharghavan, "Robust Rate Adaptation for 802.11 Wireless Networks," in *Proc. of MobiCom*, Los Angeles, CA, Sep 2006.
- [14] 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 *Proc. of SIGCOMM*, Philadelphia, PA, Aug. 2005.
- [15] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance Anomaly of 802.11b," in *Proc. of INFOCOM*, San Francisco, CA, Apr 2003.
- [16] Y. Hu and D. Johnson, "Exploiting Congestion Information in Network and Higher Layer Protocols in Multihop Wireless Ad Hoc Networks," in *Proc. of International Conference on Distributed Computing Systems*, Tokyo, Japan, Mar 2004.
- [17] Q. Xue and A. Ganz, "Ad hoc QoS On-Demand Routing (AQOR) in Mobile Ad Hoc Networks," *Journal of Parallel and Distributed Computing*, vol. 63, no. 2, pp. 154–165, 2003.

- [18] A. Sharma, M. Tiwari, and H. Zheng, "MadMAC: Building a Reconfigurable Radio Testbed Using Commodity 802.11 Hardware," in *Proc. of Workshop on Networking Technologies for Software Defined Radio Networks*, Reston, VA, Sep 2006.
- [19] W. Baumgartner, P. Weiß, and H. Schindler, "A Nonparametric Test for the General Two-Sample Problem," *Biometrics*, vol. 54, no. 3, pp. 1129–1135, 1998.
- [20] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan, "Measurement-Based Models of Delivery and Interference in Static Wireless Networks," in *Proc. of SIGCOMM*, Pisa, Italy, Sep 2006.
- [21] (2009, Jun.) Linux Wireless. [Online]. Available: <http://www.linuxwireless.org/>
- [22] (2009, Jun.) Netdisco - Network Discovery and Management. [Online]. Available: <http://www.netdisco.org/>
- [23] "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 1: Radio Resource Management of Wireless LANs," IEEE Std. 802.11k-2008, Jun. 2008.
- [24] A. Jardosh, K. Ramachandran, K. Almeroth, and E. Belding-Royer, "Understanding Link-Layer Behavior in Highly Congested IEEE 802.11b Wireless Networks," in *Proc. of Workshop on Experimental Approaches to Wireless Network Design and Analysis (EWIND)*, Philadelphia, PA, Aug 2005.
- [25] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A High-Throughput Path Metric for Multi-hop Wireless Routing," in *Proc. of MobiCom*, San Diego, CA, Oct 2003.
- [26] K. Ramachandran, H. Kremo, M. Gruteser, P. Spasojevic, and I. Seskar, "Experimental Scalability Analysis of Rate Adaptation Techniques in Congested 802.11 Networks," in *Proc. of World of Wireless Mobile and Multimedia Networks (WoWMoM)*, Helsinki, Finland, Jun 2007.
- [27] H. Lundgren, K. Ramachandran, E. Belding-Royer, K. Almeroth, M. Benny, A. Hewatt, A. Touma, and A. Jardosh, "Experiences from the Design, Deployment, and Usage of the UCSB MeshNet Testbed," *IEEE Wireless Communications*, vol. 13, no. 2, pp. 18–29, Apr 2006.
- [28] J. Bicket, D. Aguayo, S. Biswas, and R. Morris, "Architecture and Evaluation of an Unplanned 802.11b Mesh Network," in *Proc. of MobiCom*, Cologne, Germany, Aug 2005.
- [29] (2008) Qualnet Network Simulator, Version 4.0. [Online]. Available: <http://www.scalable-networks.com>



Elizabeth M. Belding is a Professor in the Department of Computer Science at the University of California, Santa Barbara. Elizabeth's research focuses on mobile networking, specifically mesh networks, multimedia, monitoring, and solutions for networking in under-developed regions. She is the founder of the Mobility Management and Networking (MOMENT) Laboratory (<http://moment.cs.ucsb.edu>) at UCSB. Elizabeth is the author of over 80 papers related to mobile networking and has served on over 50 program committees for networking conferences. Elizabeth served as the TPC Co-Chair of ACM MobiCom 2005 and IEEE SECON 2005, and the TPC Co-Chair of ACM MobiHoc 2007. She also served on the editorial board for the IEEE Transactions on Mobile Computing. Elizabeth is the recipient of an NSF CAREER award, and a 2002 Technology Review 100 award, awarded to the world's top young investigators. See <http://www.cs.ucsb.edu/~ebelding> for further details.



Kevin C. Almeroth is currently a Professor in the Department of Computer Science at the University of California in Santa Barbara where his main research interests include computer networks and protocols, wireless networking, multicast communication, large-scale multimedia systems, and mobile applications. He has published extensively with more than 150 journal and conference papers. He is also heavily engaged in stewardship activities for a variety of research outlets including journal editorial boards, conference steering committees, new workshops, and the IETF. He is a Member of the ACM and a Senior Member of the IEEE.



Prashanth A.K. Acharya received his Ph.D. in 2009 from the Department of Computer Science at the University of California, Santa Barbara. He received his B.Eng degree at the National Institute of Technology, Karnataka, India in 2002. His research interests include mobile and wireless networks, wireless multihop and mesh networks, Quality of Service, and multimedia networks. He is currently with Amazon Web Services.



Ashish Sharma is a Ph.D. candidate in the Department of Computer Science at University of California, Santa Barbara. He received his B.Tech. in Computer Science and Engineering at the Indian Institute of Technology, Guwahati in 2005. His research interests are in Wireless Systems and Networking.



Konstantina (Dina) Papagiannaki has been a researcher at Intel Labs since January 2004; from 2004 until the end of 2006 in Cambridge and since 2007 in Pittsburgh. From the beginning of 2000 until the end of 2003 she was a member of the IP Group at the Sprint Advanced Technology Labs. She got awarded her PhD from the Computer Science Department of University College London (UCL) in March 2003, receiving the Distinguished Dissertations Award 2003. She got her first degree in Electrical and Computer Engineering at the National Technical University of Athens (NTUA) in October 1998. She currently holds an adjunct faculty position in the Computer Science Department at Carnegie Mellon University and in 2008 she received the ACM Sigcomm rising star award.