Congestion-Aware Rate Adaptation in Wireless Networks: A Measurement-Driven Approach

Prashanth Aravinda Kumar 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*—Traditional rate adaptation solutions for IEEE 802.11 wireless networks perform poorly in congested networks. Measurement studies show that congestion in a wireless network leads to the use of lower transmission data rates and thus reduces overall network throughput and capacity. The lack of techniques to reliably identify and characterize congestion in wireless networks has prevented development of rate adaptation solutions that incorporate congestion information in their decision framework. To this end, our main contributions in this paper are two-fold. First, we present a technique that identifies and measures congestion in an 802.11 network in real time. Second, we design Wireless cOngestion Optimized Fallback (WOOF), a measurement-driven rate adaptation scheme for 802.11 devices that uses the congestion measurement to identify congestion related packet losses. Through experimental evaluation, we show that WOOF achieves up to 300% higher throughput in congested networks, compared to other well-known adaptation algorithms.**

I. Introduction

The use of IEEE 802.11 wireless networks is on the rise and an increasing number of people depend on a wireless connection for their Internet access. A recent survey indicated that about one-third of the Internet users in the USA obtain connectivity through wireless networks[1]. The proliferation of 802.11 networks and users, however, brings forth its own set of problems. IEEE 802.11 is a CSMA/CA based medium access scheme wherein all the users share the medium as a common resource.

A large number of users in a network can lead to excessive load or congestion and impact network performance. A case study of a large WLAN by Jardosh et al. presented an example of the adverse effects of such congestion [1]. 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 APs. Eventually the network broke down, causing frustration among the users.

One of the causes of the network meltdown was IEEE 802.11's *rate adaptation* scheme, an important aspect of the protocol that affects 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 variations in channel quality. Measurement studies have shown that current rate adaptation solutions do not perform well in congested networks [2], [3]. These solutions, not designed for operation in congested scenarios, unnecessarily switch to a lower transmission rate. This rate switch increases the channel occupation time, thereby compounding the congestion.

Our goal is to design a rate adaptation scheme that provides high network performance in both congested networks and lightly-loaded networks. A preliminary step required before we can develop a new rate adaptation scheme is to identify and measure network congestion levels in real-time. Traditional metrics, such as network throughput, fail to characterize congestion in a wireless network because of the locally shared nature of the medium and the use of multiple transmission rates. Thus, there is a need for a lightweight measurement solution that can identify congestion in a wireless network in real-time. This solution in turn provides information to the rate adaptation scheme about the current congestion level and enables an intelligent decision of what data rate to use for transmission.

In this paper, we present a measurement-driven approach to the characterization of congestion in wireless networks and incorporate this information in design of a congestion-aware rate adaptation scheme. Our two main contributions are as follows. First, we develop a congestion measurement technique for wireless networks to identify congestion in real-time. We passively measure the *channel busy time*, the fraction of time for which the medium is utilized in some time interval. We evaluate the performance of the technique in a large WLAN with active users connected to the Internet. Second, we employ the channel busy time congestion metric in the design and implementation of a new rate-adaptation scheme called Wireless cOngestion Optimized Fallback (WOOF). The use of a congestion metric enables the rate-adaptation algorithm to differentiate between packet losses due to congestion and those due to poor link quality. Through experimental

[1] http://www.pewinternet.org/pdfs/PIP_Wireless.Use.pdf (Feb 2007).

evaluation in a congested wireless network, we show that WOOF obtains significantly higher throughput (up to a three fold improvement) than current solutions.

The remainder of the paper is organized as follows. Section II describes the congestion measurement method and its performance evaluation. We survey existing rate adaptation schemes in Section III. Section IV presents a performance analysis of rate adaptation schemes in congested WLANs. Sections V and VI describe the design and evaluation of the WOOF scheme. We conclude the paper in Section VII. 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.

II. Congestion Measurement

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 [3]. 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 interference sources. Unlike wired networks, where throughput degradation on a network link is indicative of congestion, in wireless networks throughput degradation 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.

For these 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 [3], [4]. Jardosh et al. show that medium utilization can be used to classify network state as *uncongested*, *moderately congested*, and *highly congested*.

In this paper, we implement and evaluate a real-time congestion measurement technique for wireless networks. The technique is passive in nature and measures *channel busy time*, the fraction of time for which the medium was utilized, during some time interval.

A. 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, 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 [3]. 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 this metric for 802.11 networks using a feature provided in Atheros chipset-based devices.

To measure the channel busy time, we use the reverse-engineered Open HAL [5] implementation of the Mad-Wifi 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 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 [6]. The ratio of the “medium busy time” and the “cycle time” counters gives the fraction of time during which the channel was busy. 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. As we show later in this paper, this 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.

B. Evaluation of Congestion Metric

To evaluate the performance of the proposed technique, we use as a benchmark the medium utilization as seen by a sniffer operating in *monitor* mode. 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 [3]. This helps determinine the accuracy of our low overhead implementation of CBT by comparing against a fairly comprehensive but high overhead mechanism.

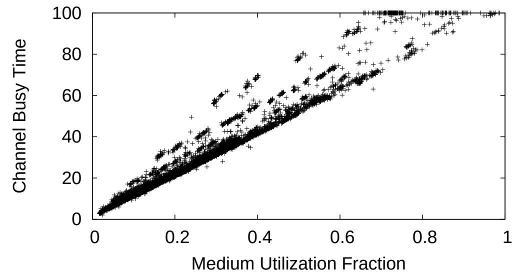
Experimental Setup: In our experiments, we use two Linux laptops equipped with Atheros chipset IEEE 802.11a/b/g cards and an access point to evaluate the passive Channel Busy Time congestion measurement technique. One laptop acts as a wireless sniffer and is placed close to the AP to perform *vicinity sniffing* [7]. As part of vicinity sniffing, the radio on the sniffer laptop operates in monitor mode and captures all packet transmissions using the `tethreal` 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 utilization value is then used to compare against CBT values during the corresponding time interval of the experiment.

We calculate the medium utilization value using the methodology proposed by Jardosh et al. [3]. 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 transmitted in the interval and the necessary MAC delay components for each frame. The second laptop, also placed close to the AP, continuously measures and records the channel busy time as described in Section II-A. In order to compare CBT values with medium utilization values during corresponding time intervals, the laptops are time synchronized to milliseconds granularity using NTP. Note that both laptops are tuned to the same channel as the AP.

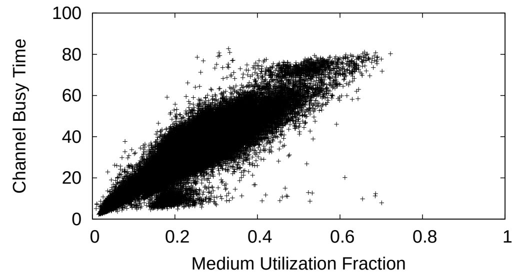
Testing Scenarios: We evaluate the congestion measurement technique in two different environments. We choose the two environments because of their vastly different characteristics.

Testbed: We conduct the first set of experiments in an indoor 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. The controlled environment of the testbed gives us the flexibility to vary network load to generate a range of medium utilization values and limit external interference. We use UDP traffic as opposed to TCP because TCP's congestion control and backoff mechanisms prevent us from controlling the rate at which data is injected into the network.

IETF Wireless LAN: To verify the performance of our technique in a real world network characterized by live Internet traffic, a large number of heterogeneous devices, dynamic user behavior, and other external factors, 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 advertized the



(a) Testbed: CBT vs medium utilization



(b) IETF: CBT vs medium utilization

Fig. 1. Correlation between CBT and medium utilization.

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 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.

Results: In Figures 1(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 1(a) and (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 over-estimates the medium utilization. This behavior is because CBT accounts for the time during which the medium was busy, but no packet was received, e.g., channel noise, packet collisions. Also, it can be seen from Figure 1(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.

We now demonstrate the utility of real-time congestion metrics in improving the performance of congested wireless networks. Our focus is on rate adaptation in wireless networks. In the following sections, we first survey existing rate adaptation algorithms. We then analyze the performance of rate adaptation schemes in a large WLAN.

III. Related Work 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 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 Signal-to-Noise Ratio (SNR) of the packet at the receiver [8]. 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) [9] is a 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.

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 message is not received). Auto-Rate Fallback interprets patterns of packet loss (e.g., four consecutive losses) as triggers to change the data rate [10]. 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 [12]. Recent work such as SampleRate [13] show 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 [13]) to choose the rate. We describe SampleRate in detail in Section V-B.

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 [3]. 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) [14] and Collision-Aware Rate Adaptation (CARA) [12], 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 handle congestion 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 adaptation decisions [14].

IV. Rate Adaptation during Congestion

We now analyze the behavior of current rate adaptation schemes in a congested network. We focus 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 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 the packet transmission with the retry flag set. 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 2 plots the medium utilization levels and the fraction of data frames that were lost during the Wednesday plenary session. During periods of high utilization, the number of packet losses also increases. This can be attributed to the losses caused by medium contention (i.e., when the backoff counters of two or more nodes expire at the same time.) Alarmingly, 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.

To analyze the impact of such high packet loss rates on rate adaptation schemes, we study the distribution of data rates. The access points at the IETF meeting advertized only the following data rates (in Mbps) as supported: 11, 12, 18, 24, 36, 48, and 54. A client that supports 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 802.11g [2]. Table I shows the distribution of data rates for only the 802.11g clients observed during the session. We see that 73% of the packets used the lowest possible data rate. This behavior can be attributed to the rate adaptation schemes. 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.

Previous work has observed a similar effect of congestion on rate adaptation [2], [7]. In a congested network, a majority of the 802.11 packets use the lowest possible rate. Such packets also consume a large fraction of the medium time, since they take a longer time for transmission. These packets are more susceptible to collisions (e.g., by hidden terminals). Switching to a lower rate as a result of contention losses is not only unnecessary but also increases the medium busy time. Thus it is important to understand the cause of a packet loss and respond appropriately in the rate adaptation algorithm.

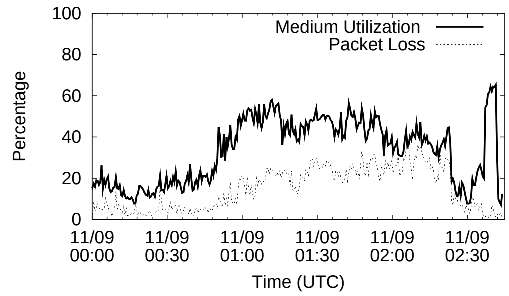


Fig. 2. Medium utilization and packet loss rate in a congested 802.11 network.

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

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

V. Wireless cOngestion Optimized Fallback (WOOF)

The discussion in the previous section leads us to 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.

A. Identification of Congestion-Related Packet Loss

In Section II-B we noted that Channel Busy Time (CBT) was a good predictor of network congestion levels. We now explore the relationship between the CBT metric and packet loss rate.

Figure 3 plots 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 averages over 30 second time windows. We observe a strong linear correlation with the packet loss rate and the observed CBT values; as the CBT increases, the probability of a packet loss due to congestion also increases.

Unfortunately, our analysis of packet loss versus CBT values for other sessions in the 67th IETF did not exhibit such strong correlations. However, we note that the average packet loss rate was higher during periods

[2]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 802.11g data rate.

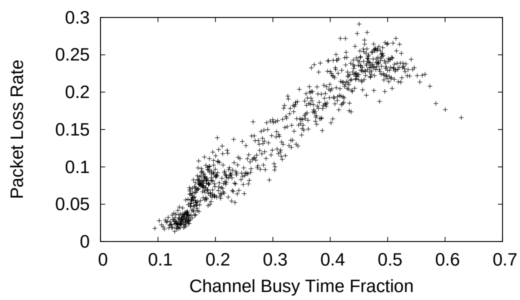


Fig. 3. Relationship between channel busy time and packet loss rate during the Wednesday plenary session.

of high utilization in these sessions. These observations lead us to conclude that the CBT information can be used as a good indicator of packet loss caused by the congestion levels in the network. The exact relationship of CBT may vary depending on environmental factors. A rate-adaptation scheme that uses CBT as a heuristic to identify congestion-related packet losses must therefore be dynamic and capable of adapting to changes in the environment. In the design of WOOF, we initiate our prediction heuristic with the initial setting of a linear relationship between packet loss and CBT. We then dynamically adapt the weight of this relationship based on the observed network performance to model the current environment.

The CBT metric only helps in identifying the cause of a 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 supplemental information that a rate adaptation scheme should use in addition to packet loss information. To do so, we borrow the basic framework of the design of SampleRate [13] scheme in order to handle the packet loss information in WOOF. We now outline the operation of SampleRate, and then discuss the design of WOOF.

B. *SampleRate*

SampleRate is a rate adaptation scheme that accounts for the time required for successful transmission of a packet [13]. The basic idea of SampleRate is to choose the data rate that is expected to consume the least medium time, 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) [15] for each data rate. The ETX represents the average number of transmission attempts required for successful reception of a packet. 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 [16]. Congestion losses impact SampleRate's estimation of ETX at the different data rates and lead to a sub-optimal choice of transmission rate.

C. *Design of WOOF*

We base the design of the WOOF solution on that of SampleRate. In particular, we build on SampleRate's framework of calculation of ETT and use this information to choose the best data rate. In addition, we incorporate the ability to discern the cause of packet loss, in order to enable operation in congested networks.

In Section IV 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 WOOF 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 weight every packet loss proportionally to the probability that it was not a congestion-related loss.

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

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

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

where $CBTF$ represents Channel Busy Time Fraction and β represents the confidence factor, $0 \leq \beta \leq 1$. The CBT 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, e.g., 54 Mbps. 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 congestion-related packet losses. Similarly, when the metric indicates a drop in performance, β is decreased by 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 VI-D, we examine the impact of the measurement window, W, and its effect on the convergence time for β values. In Section VI-C, we evaluate the performance of WOOF under different combinations of link SNR and congestion levels.

D. Implementation

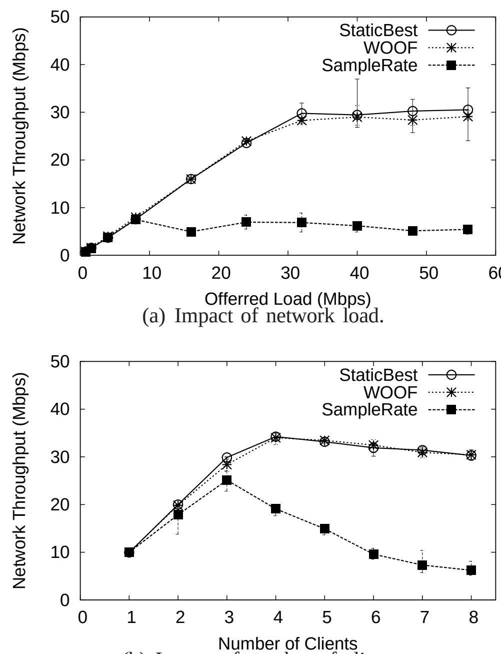
We implemented WOOF as a rate adaptation module for the MadWifi driver v0.9.2. We choose W=1s as the window of observation and recalibration. A large value of W reduces the responsiveness to changes in the environment utilization. Smaller values of W increase the processing load due to frequent recalibrations. We set the initial value of β to 0.5. At each interval W, the driver calculates the Channel Busy Fraction. In addition, the normalized network performance, as described in Section V-C, and the value of β are updated at each interval.

VI. EVALUATION

We evaluate the performance of WOOF in two network scenarios: a WLAN and a multihop mesh network. We first describe the experiments in the WLAN environment.

The WLAN scenario 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 and MadWifi as the driver.

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 [13], [14]. Thus we expect WOOF to provide better performance than ARF and AARF in all cases where WOOF performs better than SampleRate. In addition, for the WLAN we also compare the 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 and WOOF.



(a) Impact of network load.

(b) Impact of number of clients.

Fig. 4. Performance comparison between WOOF and SampleRate.

A. 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, 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 network performance for each offered load is measured using the `iperf` utility and UDP traffic 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 4(a) 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 three 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.

B. 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 4(b) plots a graph of the total network throughput versus the number of clients in the network. At low contention levels ($<$4 clients), we observe that the throughput of SampleRate increases almost linearly to reach a maximum of about 24 Mbps. Once the network starts to become congested ($\geq$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, adding 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.

C. 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 the experiments similar to that in Section VI-B. We consider four different combinations of link SNR and congestion levels. The good SNR link scenario has all of the client links with sufficient SNR to operate at 48 and 54Mbps. The low SNR scenario is acheived 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 ranges between 2Mbps and 18Mbps. We chose two congestion levels: low congestion corresponds to two clients with an offered load of 5Mbps each; high congestion corresponds to eight clients with offered load of 5Mbps each.

	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 II
NETWORK THROUGHPUT (IN MBPS) UNDER DIFFERENT COMBINATIONS OF SNR AND CONGESTION LEVELS.

Table II lists the network throughput in each of the scenarios for both SampleRate and WOOF. We see that the performance of WOOF under low congestion scenarios are comparable to that of SampleRate. During high congestion, we observe that WOOF improves the network throughput for both the 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.

D. 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 VI-A. Each of the eight clients has an offered load of 10 Mbps. Table III shows the average network throughput for different W values. We observe that for low W values, below 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 those 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 on the rate adaptation algorithm and the device driver. A high value of W makes WOOF less responsive to the environment. Based on these tradeoffs, we recommend a value of $W = 1$s.

W (seconds)	Throughput (Mbps)
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 III
IMPACT OF MEASUREMENT INTERVAL W.

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

E. 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

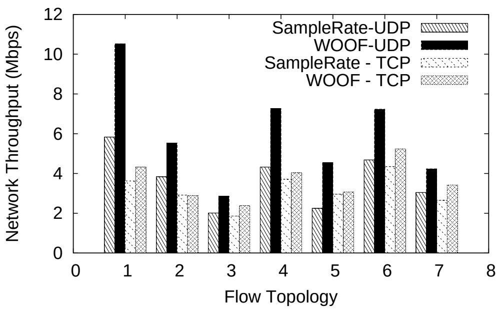


Fig. 5. Network throughput with UDP and TCP for different flow topologies in the UCSB MeshNet.

the performance of WOOF in real network deployments. We conduct our experiments on the UCSB MeshNet testbed [17]. The 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 [18] 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.

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 5 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.

VII. Conclusion

Congestion in an IEEE 802.11 network causes a drastic reduction in network performance. Critical to tackling this problem is the ability to identify and measure congestion. In this paper we have presented a passive technique that measures the utilization of the wireless medium in real-time. We then used the congestion measurement technique to develop a rate adaptation scheme called WOOF. Performance evaluation showed up to a three-fold gain in throughput with the use of WOOF in a congested network. In addition to WOOF, we believe that our congestion measurement technique can be used to design new solutions that perform well under congestion scenarios. For example, the CBT metric can be used for bandwidth estimation to facilitate effective flow admission control.

References

[1] 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.

[2] M. Rodrig, C. Reis, R. Mahajan, D. Wetherall, and J. Zahorjan, "Measurement-based Characterization of 802.11 in a Hotspot Setting," in *Proc. of EWIND*, Philadelphia, PA, Aug 2005.

[3] A. Jardosh, K. Ramachandran, K. Almeroth, and E. Belding-Royer, "Understanding Congestion in IEEE 802.11b Wireless Networks," in *Proc. of IMC*, Berkeley, CA, Oct 2005.

[4] Y. Hu and D. Johnson, "Exploiting Congestion Information in Network and Higher Layer Protocols in Multihop Wireless Ad Hoc Networks," in *Proc. of ICDCS*, Tokyo, Japan, Mar 2004.

[5] (2006, Dec) OpenHAL - MadWifi. [Online]. Available: http://madwifi.org/wiki/openHAL

[6] 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.

[7] 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 EWIND*, Philadelphia, PA, Aug 2005.

[8] X. Yang and N. Vaidya, "On the Physical Carrier Sense in Wireless Ad Hoc Networks," in *Proc. of INFOCOM*, Miami, FL, Mar 2005.

[9] 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.

[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 MSWiM*, Venice, Italy, Oct 2004.

[12] 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.

[13] J. Bicket, "Bit-rate Selection in Wireless Networks," Master's thesis, Massachusetts Institute of Technology, 2005.

[14] 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.

[15] 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.

[16] 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 WoWMoM*, Helsinki, Finland, Jun 2007.

[17] 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.

[18] 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.