Experimentation and Performance Evaluation of Rate Adaptation Algorithms in Wireless Mesh Networks*

Emilio Ancillotti, Raffaele Bruno, Marco Conti
Institute for Informatics and Telematics
Italian National Research Council
Via G. Moruzzi 1, 56124 Pisa Italy
{e.ancillotti,r.bruno,m.conti}@iit.cnr.it

ABSTRACT

In this paper we present an experimental study conducted in 802.11-based mesh networks of three existing rate adaptation algorithms. The aim of this study is twofold. On the one hand, we explore the ability of these algorithms to cope with moderate to high medium contention levels. On the other hand, we investigate their performance on medium-distance 802.11 links. Our study indicates that, in congested networks, the network throughput can degrade up to ten times with respect to the best performance if the rate decision process is based solely on frame loss rates, without differentiating between the various causes of losses (i.e., channel errors or collisions). In addition, we have shown that these rate adaptation strategies perform reasonably well when the time correlation between channel errors is at least of the order of the sampling period used to estimate the channel dynamics. We believe that this study can be useful to derive correct guidelines for the design of new optimized rate adaptation algorithms taking into consideration the above factors.

Categories and Subject Descriptors

C.2.5 [**Computer-Communication Networks**]: Local and Wide-Area Networks—*Access schemes*; C.4 [**Performance of Systems**]: Measurement techniques.

General Terms

Algorithms, Measurement, Performance.

Keywords

802.11 technology, rate adaptation algorithms, wireless mesh networks, experimental evaluation.

1. INTRODUCTION

*This work was supported by the European Commission through project EU-MESH (Enhanced, Ubiquitous, and Dependable Broadband Access using MESH Networks), FP7 ICT-215320

802.11-based wireless mesh networks have experienced an enormous growth over the last few years. Within a mesh backbone a few mesh nodes may have a high-speed Internet connection, but Internet access is shared among all the mesh nodes in the network by exploiting the mesh connectivity and the multi-hop routing capabilities of other mesh nodes [6]. Thus, mesh networks have the potential to provide easy-to-deploy and low-cost ubiquitous Internet access for metro-scale areas. A sign of the interest in mesh networking is the number of start-ups and big companies offering mesh solutions (e.g., Tropos Networks, MeshDynamics, Firetide, etc.), as well as the considerable number of grassroots groups and research initiatives that use this technology to share Internet connectivity across neighborhoods, campuses and city blocks (e.g., MIT Roofnet [4], UCSB MeshNet [18], TAP Project at Rice University [7], Heraklion MESH in Crete [10], etc.).

Several obstacles still prevent the wide deployment of broadband metro-scale wireless mesh networks, including low capacity, limited system performance, and the inefficient usage of the wireless resources. Thus, a lot of work have been done recently on understanding the impact of various factors, such as multi-hop communications and interference, on the mesh network performance [1, 7, 10]. In parallel, there have been several studies to develop new mechanisms and algorithms to take advantage of the multi-channel and multi-rate capabilities of 802.11 devices, and to exploit the channel and path diversity [2, 4, 7, 11, 12]. All these studies have pointed out that the transmission rate used by the wireless interfaces is one of the most important factors in influencing the network performance.

In principle, a wireless card with multi-rate capabilities should select the run-time transmission rate based on the wireless channel dynamics with the objective of achieving the best link performance (e.g., in terms of maximum throughput or minimum packet delay). To this end, several rate adaptation algorithms have been proposed in the literature. These proposals primarily differ in the metrics used to estimate the link quality, and in the rate-selection decision process. A brief overview of prior work is presented in Section 2. A subset of the proposed rate adaptation schemes, when compliant with current 802.11 technology constraints, have been also implemented in commodity hardware [3, 17, 19, 27] and used in real products. Thus, experimental studies are now appearing in the literature, which investigate how effectively these adaptive rate adaptation algorithms perform in practical settings, and in the presence of various environmental dynamics. However, these studies have mainly focused on indoor wireless networks, considering the impact of channel dynamics due to rapid fluctuations of the receive signal strength [21], random channel errors, mobility-induced channel variations, and contention from hidden stations [27]. Moreover, these experimental studies have been conducted mostly in small wireless networks consisting of an AP and a few clients. The authors in [4] have conducted performance tests of the SampleRate [3] algorithm in an outdoor mesh networks. However, their focus is on the ad hoc routing protocol, and they do not analyze the rate adaptation problem in isolation.

In this paper, we conduct an experimental study comparing the performance of three rate adaptation algorithms implemented in the open-source MadWifi driver, namely AMRR [17], ONOE [19], and SampleRate [3]. These algorithms are representative of the autorate schemes that utilize an estimate of the number of frame retransmissions to select the best transmission rate. AMRR and ONOE solutions implement a simple threshold-based algorithm to adjust the transmission rate, while SampleRate is a more sophisticated scheme that tries to predict which is the bit rate that provides the maximum link-layer throughput. A more detailed description of these rate adaptation algorithms and their implementation in the MadWifi driver is reported in Section 2.

The goal of our investigations is twofold. Firstly, we assess the performance of these rate adaptation algorithms in situations of moderate to high medium contention levels. Secondly, we explore the ability of these schemes to provide the best link performance in an outdoor mesh network consisting of medium-distance 802.11 links. Our experimental results show the following: *i*) the use of the frame loss rate to adjust the transmission rate used by the wireless interface may trigger unnecessary rate decreases when the medium is congested, resulting into considerable throughout degradations. For instance, with eleven saturated stations the throughput achieved with frame loss-rate threshold-based schemes (namely, AMRR and ONOE) can be up to ten times lower than the best throughput obtained with a fixed transmission rate; *ii*) these rate adaptation strategies perform reasonably well in an outdoor mesh network if the time correlation between channel errors is at least of the order of the sampling period used to estimate channel dynamics. Based on our measurements we argue that SampleRate is probably the best rate adaptation algorithm for static settings, and relatively stable links. We believe that our experimental study contributes to reveal some challenges in the design of rate adaptation strategies suitable for urban mesh networks (i.e., multi-hop wireless networks made of medium-distance, highly-loaded 802.11 links), whose impact has been underestimated in previous papers. In addition, our results can be useful to derive correct guidelines for the design of new optimized rate adaptation algorithms that take the above factors into consideration.

The remaining of this paper is organized as follows. In Section 2, we review the related work. Section 3 describes the design and features of the mesh network testbeds used in this study. Section 4 reports the results of our experiments and discusses the performance of existing rate adaptation algorithms. Finally, Section 5 concludes the paper.

2. BACKGROUND

The aim of this section is to briefly discuss the advantages and weaknesses of most common approaches adopted for rate adaptation in 802.11-based wireless networks. Then, we describe in detail the rate adaptation algorithms implemented in the MadWifi driver, which are the focus of this study.

The IEEE 802.11 standard mandates the support of multiple transmission rates at each physical layer [14]. Specifically, the 802.11*b* PHY supports four transmission rates (1∼11 Mbps), the 802.11*a* PHY offers eight rates (6∼54 Mbps), and the 802.11*g* PHY supports both 802.11*b* and 802.11*a* bit rates. However, the 802.11 standards left unspecified the rate adaptation algorithm. Thus, several strategies have been proposed recently to support intelligent rate decisions. Broadly speaking, two main approaches can be identified in the design of rate adaptation schemes for 802.11 wireless networks: *i*) signal-strength-based algorithms, and *ii*) statistics-based algorithms. In the former case, the rate adaptation algorithm relies on wireless signal measurements, such as Signal-to-Noise Ratio (SNR) or Received Signal Strength Indicator (RSSI) to infer the transmission rate that would provide the best performance, usually in terms of maximum link-layer throughput [24]. For instance, both RBAR (Receiver-Based AutoRate) [13] and OAR (Opportunistic Auto-Rate) [25] employ RTS frames to estimate the SNR at the receiver, and piggyback this information to the sender on subsequent CTS frames, so that the sender can adjust the rate accordingly. Since SNR estimates can be unreliable or difficult to obtain with available hardware, in [23] it is proposed to directly measure the RSSI value at the sender and to assume channel symmetry. The main drawback of these schemes is that they require an accurate channel model to map the signal strength measurements to the corresponding link performance, which may be impractical for highly variable environments.

The rate adaptation schemes that collect information on frame transmissions (e.g., number of retries, number of consecutive frame successes and failures, etc.) to guide the rate decision process, fall within the category of statistics-based algorithms. The first example of this type of rate adaptation algorithms is ARF (Automatic Rate Fallback) [15]. Basically, using the ARF scheme a *probe packet* is sent after either ten consecutive transmission successes or a timeout. Each probe packet is transmitted at a bit rate higher than the current one in use. If the probe packet succeeds, ARF increases the transmission rate. On the contrary, ARF reduces the transmission rate upon two consecutive transmission failures. ARF's simple heuristic has inspired several subsequent schemes, such as AARF (Adaptive ARF) [17], AMRR (Adaptive Multi Rate Retry) [17] and ONOE [19], which tried to reduce the ARF's probing overhead, and to make the rate adaptation process less vulnerable to short-term channel fluctuations. Although very popular, the ARF-like rate adaptation schemes suffer from a number of drawbacks. First of all, not always there is a strong correlation between frame loss rates and future channel conditions [3]. In addition, it may be inefficient to use only a few probe packets to estimate the channel quality [27]. Finally, rate adaptation algorithms based on the number of retries are negatively affected by medium contention because packet collisions generate unnecessary rate decreases. To address this last limitation, other algorithms have attempted to use loss differentiation techniques, such as in LD-ARF (Loss-differentiating ARF) [22] or CARA (Collision-aware Rate Adaptation) [16], or to directly estimate the contention level, such as in SampleRate [3] or BEWARE (Background Traffic-Aware Rate Adaptation) [26]. However, the first category of proposals may require changes to the 802.11 MAC specification, making them unsuitable for ready deployment on available radio interfaces, while the latter category of proposals usually requires a significant number of samples for building reliable contention level estimates, making them unsuitable for fast-varying environments.

2.1 Rate Adaptation Algorithms in the MadWifi Driver

2.1.1 AMRR

AMRR algorithm employs the multiple rate retry capabilities of the MadWifi driver [20]. Specifically, the MadWifi driver allows the network interface to transmit at different data rates the retransmissions of a given frame. According to the default MadWifi policy, four rates (r_0, r_1, r_2, r_3) and transmission counts (c_0, c_1, c_2, c_3) are associated to each frame. The driver starts transmitting using bit rate r_0 and continues using this rate for the first $c_0 - 1$ retries. If the transmission keeps on failing, the driver tries the rate r_1 for c_1 times, then the rate r_2 for c_2 times and, finally, the rate r_3 for c_3 times before discarding the frame.

AMRR set $c_0 = c_1 = c_2 = c_3 = 1$, namely, each rate is tried just once. Then, r_3 is always set as the lowest bit rate (i.e., 1 Mbps in $802.11b/g$, and 6 Mbps in $802.11a$), while r_1 is the rate immediately lower than r_0, and r_2 is the rate immediately lower than r_1. To select r_0, the AMRR algorithm employs the following simple heuristic: if less than 10% of the packet transmissions failed during the last observation period (and total frame transmissions are at least 10), then increase the transmission rate; otherwise, if more than 33% of the packet transmissions failed during the last period (and total frame transmissions are at least 10), then decrease the transmission rate. By default, an observation period is one second in AMRR.

2.1.2 ONOE

ONOE algorithm is a variant of the AMRR scheme. Specifically, ONOE uses larger retransmission counts than AMRR (i.e., $c_0 = 4$ and $c_1 = c_2 = c_3 = 2$), while it sets r_1, r_2, r_3 rates as AMRR. The major difference between the two schemes is that the ONOE algorithm associates a number of *credits* to the current rate r_0. More precisely, if less than 10% of the frame transmissions failed during the last period (and total frame transmissions are at least 10), then the credits are incremented by one, otherwise the credits are reduced by one. If the total credits at the current transmission rate are above a threshold (default is 10), then rate r_0 is increased. If more than 50% of the frame transmissions failed during the last period (and total frame transmissions are at least 10), then the transmission rate is immediately decreased. Whenever the rate is changed, the credit counter and the rate statistics are reset. Note that ONOE is less sensitive to single packet failures than AMRR. However, it is also more conservative, and it takes several seconds to increase the transmission rate.

2.1.3 SampleRate

SampleRate algorithm implicitly estimates the medium contention level by evaluating the expected transmission time, say *tx_time*, for a frame at different data rates. Specifically, the sender measures for each destination the number r of retransmissions needed to successfully transmit a frame of size L (in bytes) with bit rate b. Then, the expected transmission time is approximated as follows

$$tx\_time(L,r,b) = backoff(r+1) + (r+1) \cdot (\Delta + L * 8/b) \,, \quad (1)$$

where $backoff(r+1)$ expresses the average backoff delay introduced after r retries, and Δ accounts for the fixed MAC overheads (e.g., interframe spaces, acknowledgment frames, etc.). Normally, SampleRate algorithm transmits each frame at the bit rate that is characterized by the shortest expected transmission time. In addition, to update the *tx_time* statistics, SampleRate sends probe packets every ten frames. However, the sender does not probe all the available rates, but only the ones that have a minimum packet transmission time (i.e., with $r = 0$) lower than the average transmission time of the current bit rate.

Note that the SampleRate algorithm implemented in the MadWifi driver is slightly different from the above description because it does not perform per-frame rate adaptation as specified in [3], but the best rate is only changed every 2 seconds or upon four consecutive losses. In addition, the MadWifi implementation of SampleRate adopts a multiple rate retry strategy similar to ONOE. However, this variant of the SampleRate algorithm showed degraded performance in our experiments, and we modified the MadWifi implementation on this last feature to be compliant with the original specification [3].

3. MESH PLATFORM

The experimental data reported in this paper are the results of measurements we have taken from two mesh testbeds, one deployed indoors, and the other one deployed outdoors in the CNR's campus area. In the following we separately present the features of these two experimental networks.

3.1 Indoor Mesh Testbed

The indoor mesh network is a 12-node network deployed on one floor of a fairly typical office building, where uncoordinated APs are also deployed, operating on non-orthogonal channels. The nodes of our testbed are all IBM Thinkpad model R50E laptops. Each of these machines has a 1.5GHz Intel Pentium processor with 512MB of memory. They all run Linux Debian with kernel version 2.6.22. Each node has one NetGear WPN511 card, which is a multi-band radio interfaces supporting $802.11b/g$ transmissions, managed with the MadWifi driver version 0.9.4. The nodes run OLSR daemon version 0.4.10 (by olsr.org) as the ad hoc routing protocol [9]. All our experiments were conducted over IPv4 using statically assigned addresses.

3.2 Outdoor Mesh Testbed

The outdoor mesh network is composed of five mesh routers that we have designed and assembled using commodity hardware. These mesh routers are deployed on the rooftops of three office buildings located in the CNR's campus area. Note that the CNR's campus area is situated on the outskirts of Pisa, but it is surrounded by several residential buildings. We have verified that a large number (up to 40) of APs are visible, on average, from each mesh node. In addition, other uncoordinated wireless networks are also deployed inside the CNR's office buildings. Thus, we believe that our experimental testbed, although deployed inside a research campus, is well representing the typical interference conditions of urban mesh networks.

Each mesh node consists of a waterproof plastic box containing a Soekris net4801 communication board (266 GHz CPU and 256MB SDRAM) with a 100GB 2.5" HDD. Each board is equipped with two Atheros AR5414 mini PCI cards, which support $802.11a/b/g$ transmissions. Similarly to the indoor mesh tested, each mesh router runs Linux Debian (kernel version 2.6.22), with the MadWifi driver version 0.9.4, and OLSR daemon version 0.4.10 (by olsr.org). Figure 1 illustrates the scaled network topology and connectivity graph of our testbed. As shown in the diagram, all mesh routers except node A are equipped with omni-directional antennas of various gains (8 dBi for nodes B and D, being link B-D the shortest one in our network, and 15 dBi for nodes C and E). Node A is equipped with a 15 dBi Yagi directional antenna pointing to node C, and a 19 dBi Grid directional antenna pointing to node B, while nodes B and C are equipped also with a 19 dBi Grid directional antenna and a 15 dBi Yagi directional antenna, respectively, both pointing to node A. Note that the differences in link distances and antenna characteristics ensure a reasonable variability of link qualities, which is useful for achieving more realistic results. Finally, each network card is connected to the external antenna with a coaxial cable that introduces 3 dB of attenuation.

Our mesh router design supports additional features to permit remote and automatic system recovery, battery-powered operations and run-time system monitoring. Specifically, each mesh node contains an intelligent remote power switch (Dataprobe iBoot) to sup-

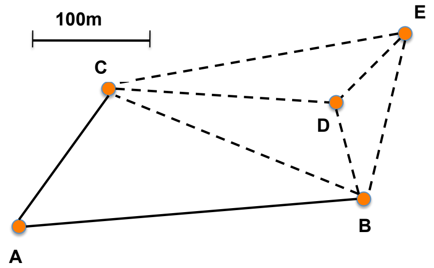


Figure 1: Connectivity graph of our outdoor mesh network. Solid lines are directional links, while dashed lines are omni-directional links. Scaling is used to represent the real distances between nodes.

port automatic reboots of the Soekris board based on the results of direct pinging of this device. In addition, each node is equipped with a rechargeable 12V battery to permit continuous operations, even in situations of power outages. Finally, to support run-time remote management and monitoring of the mesh router even when its wireless interfaces are down, we connected each node to the wired network infrastructure, which provides a parallel and independent monitoring network.

4. EXPERIMENTAL EVALUATION

In this section we show experimental results comparing the performance of the three existing rate adaptation schemes implemented in the MadWifi driver, i.e., AMRR, ONOE and SampleRate. Specifically, we present two main sets of experiments. With the first set of measurements we aim at assessing the ability of these algorithms to cope with moderate to high medium contention levels. To this end, we used our indoor mesh network and we varied the number of simultaneous traffic flows. The goal of the second set of experiments is to evaluate the performance of these rate adaptation algorithms when used on medium-distance 802.11 links established between static mesh nodes.

4.1 Experimental Methodology

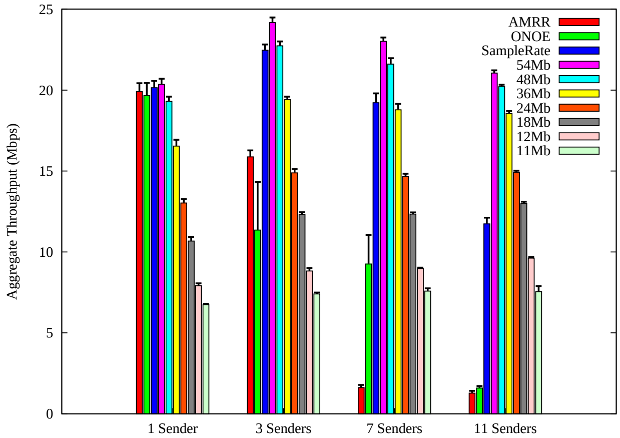
All experiments are performed using 802.11*g* transmission mode with channel 11 in order to have a broader rate variability. As previously pointed out, our indoor environment is a typical office setting, where uncoordinated wireless networks are also present. Thus, unpredictable and uncontrolled interference can be originated by nearby APs and wireless clients. However, we believe that the randomness due to the external interference is well representing the characteristics of real radio environments, and it is useful to attain more realistic results. Additionally, we repeated each trial ten times and we computed the 95% confidence intervals in addition to the average values to verify the statistical validity of our measurements. When more controlled experiments where needed (e.g., to investigate the influence of medium contention levels on the performance of rate adaptation algorithms) we conducted our measurement campaigns during nightly hours to minimize external factors. More specifically, our objective was to reduce the impact of unpredictable background traffic from other wireless networks, and to have comparable measurements from experiments conducted in a period of several hours. We believe that conducting our tests in nightly hours does not limit the significance of our experimental results, but it simply allowed us to better control background traffic variability during the tests.

All the experiments with unicast traffic are performed using UDP sources with a sending rate that saturates the channel bandwidth. If not otherwise stated, the packet size is 1500 bytes. During the unicast experiments we collected per-station statistics every 2 seconds. The metrics we consider during these tests are the distribution of transmission rates, the distribution of transmission attempts, and the application-level throughput. Note that we extended the MadWifi driver to obtain these link-layer statistics. We also performed experiments with broadcast traffic to evaluate the frame loss characteristics of the 802.11 links in our outdoor mesh network. In these experiments, we generated 1500-byte long broadcast frames with a sending rate that saturates the channel bandwidth. We conducted our broadcast tests using fixed transmission rates and we collected statistics on the packet error rates.

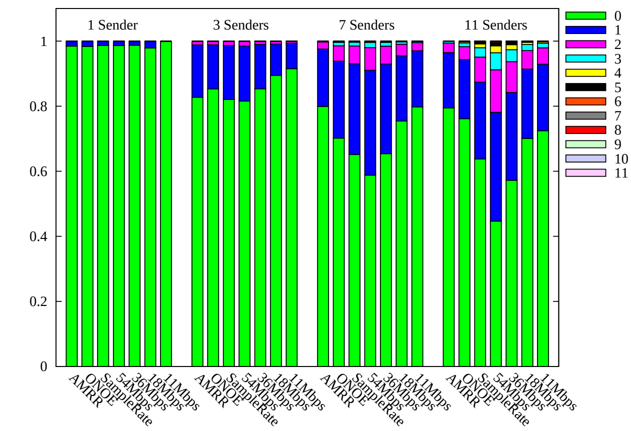
4.2 Indoor Experiments

The goal of this set of experiments is to investigate how the rate adaptation algorithms implemented in the MadWifi driver adjust the transmission rate when the frame losses are the result of collisions on the wireless medium rather than channel errors. To this end, we utilize our indoor mesh testbed and we position the twelve stationary mesh clients described in Section 3.1 in a single large room to form a single-hop wireless network. The purpose of using a small-scale mesh network is to ensure similar channel conditions between all mesh nodes. Then, we varied the number of UDP senders (the destination is the same for all the senders), and for each setting we tested, in turn, the AMRR, ONOE and SampleRate algorithms and the twelve fixed transmission rates supported by 802.11*g* PHY(from the highest to the lowest). Each experiment run lasts three minutes, thus the entire trial requires almost one hour to be completed. The same trial is repeated ten times for confidence of the results. The results of the performance tests, aggregating the statistics of all the senders, are reported in Figure 2.

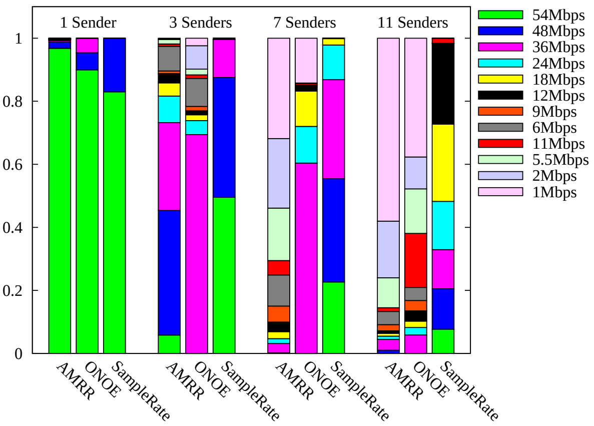
As shown in Figure 2(a), with a single sender the maximum throughput is achieved when the node's wireless interface adopts a fixed transmission rate equal to 54 Mbps. Moreover, SampleRate and AMRR algorithms approach very close the maximum throughput (i.e., with less than 2% of degradation), and ONOE scheme is the worst one with about 11% reduction in link performance. To explain these results, we can observe in Figure 2(b) that more than 98% of frames are successfully transmitted at the first attempt using 54 Mbps as fixed transmission rate, indicating that the link is of good quality. When we reduce the transmission rate the aggregate throughput decreases because the efficiency of the 802.11 MAC protocol decreases without a comparable reduction of frame loss rates. As shown in Figure 2(c), SampleRate transmits about 83% of frames at the highest transmission rate and almost all the remaining frames at 48 Mbps. The explanation of this behavior is that SampleRate estimates the transmission time by sending every ten regular frames a *probe* frame at a different bit rate. The differences between the throughput achieved with SampleRate and the maximum throughput is negligible because the MAC efficiency when transmitting frames at either 54 Mbps or 48 Mbps is similar (see Figure 2(a)). AMRR's rate distribution in Figure 2(c) shows that AMRR sends most of the frames at 54 Mbps. Indeed, AMRR reduces the transmission rate of the first transmission attempt only if the frame loss rate is greater than 33%. Thus, AMRR keeps using 54 Mbps as most preferred transmission rate. However, AMRR adopts a multiple rate retry strategy with small retry limits (all retry limits are set to one). This induces a higher rate variability than SampleRate, which negatively affects the overall throughput performance, because AMRR occasionally tries very low transmission rates. ONOE performs similarly to AMRR although it sends 89% of frames at 54 Mbps, while AMRR sends 96% of frames



(a) Aggregate throughput



(b) Frame retransmission distribution



(c) Rate distribution

Figure 2: Effect of medium contention on different rate adaptation strategies.

at 54 Mbps. However, ONOE employs larger retry counters than AMRR, thus it rarely uses very low transmission rates.

Let us now consider a situation of high medium contention with eleven senders transmitting packets concurrently. As shown in Figure 2(a), a fixed transmission rate of 54 Mbps is still the best strategy yielding the maximum link throughput. However, only 34% of frames are received successfully at the first transmission attempt due to collisions, and the average frame loss rate is about 47% (see Figure 2(b)). SampleRate evaluates the expected transmission time of a frame at a given rate using formula (1), which does not account for channel occupations due to other stations' transmissions. As a consequence, SampleRate overestimates the maximum throughput achievable at the different transmission rates, leading to a conservative rate selection. Thus, SampleRate sends only 7% of frames at 54 Mbps with a throughput reduction of 55% over the best link performance. Both AMRR and ONOE performs remarkably worse than SampleRate, obtaining a throughput that is about ten times lower than SampleRate's throughput. The explanation of this phenomenon is that the rate downgrade policies in both AMRR and ONOE operate to keep the frame loss rate below pre-determined and fixed thresholds (namely, 33% for AMRR and 50% for ONE). However, in the 802.11 MAC protocol collisions are needed to increase the backoff window size, thus achieving a time spreading of the transmission attempts that is adequate for the current congestion level. As proven in [5], an optimal collision rate exists, depending on the number of competing stations, that maximizes the channel capacity. On the contrary, both AMRR and ONOE try to enforce arbitrary collision rates that can lead to operating conditions far from the optimum. Note that AMRR shows the worst performance because it permits only four attempts per each frame.

In case of low-mid medium contention SampleRate still provides a throughput gain over the other rate adaptation algorithms based solely on frame loss-rate characteristics. Indeed, SampleRate makes more intelligent rate decisions than AMRR and ONE because it tries to explicitly find the transmission rate that yields the highest expected throughput rather than minimizing the frame loss rate. Note that AMRR shows better throughput performance than ONOE with three senders, while ONOE achieves better throughout than AMRR with seven senders. This inversion with a steep degradation of the AMRR's performance is caused by the fact that AMRR uses a lower frame loss-rate threshold to trigger the transmission rate downgrade than ONOE. Indeed, ONOE sustains a rate if it suffers less than 50% frame losses, while AMRR sustains a rate if it suffers less than 33% frame losses.

In conclusion, our experimental results indicate that all the considered rate adaptation algorithms reduce the throughput even when the wireless channel is moderately congested because the collision events trigger unnecessary rate downgrades. However, SampleRate is a more robust algorithm against medium contention than frame loss-rate threshold-based schemes, such as AMRR and ONOE.

4.3 Outdoor Experiments

The aim of this section is to investigate how the rate adaptation algorithms implemented in the MadWifi driver perform with lossy links, as the ones observed in our outdoor mesh testbed. The focus now is to understand how the channel conditions of medium-distance 802.11 links established between static mesh nodes may impact the rate decision process.

There have been detailed performance studies of 802.11 links in community-based mesh networks [1, 7]. These studies have shown that frame error rates on outdoor 802.11 links can be highly variable due to multi-path fading and external interference. It is evident that the performance of a rate adaptation algorithm is closely

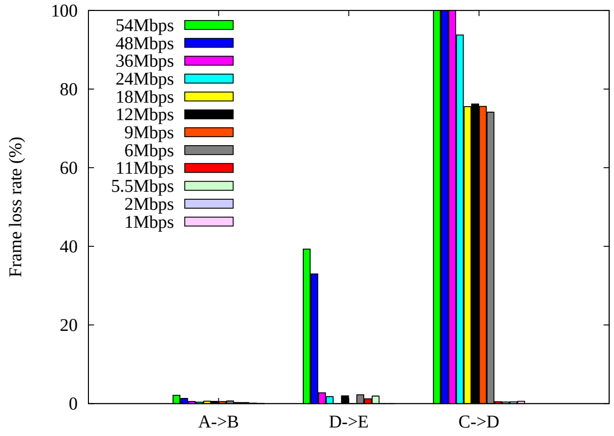


Figure 3: Average frame loss rates for representative links at various transmission rates.

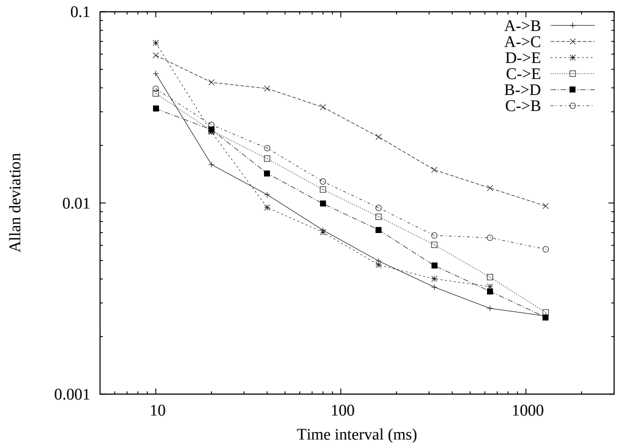


Figure 4: Allan deviation of various links for the most robust bit rate.

connected to the link-level loss characteristics affecting each transmission rate. Therefore, we have conducted a preliminary set of experiments to better understand the relation between the frame error rate due to channel errors and the various transmission rates in our mesh testbed. Specifically, we have instructed each mesh router to transmit broadcast frames tagged with a unique sequence number at different transmission rates (each bit rate is tested for three minutes), and the other mesh routers to record the sequence numbers of the received frames. Then, the collected traces are used to analyze the statistical properties of the frame loss rate.

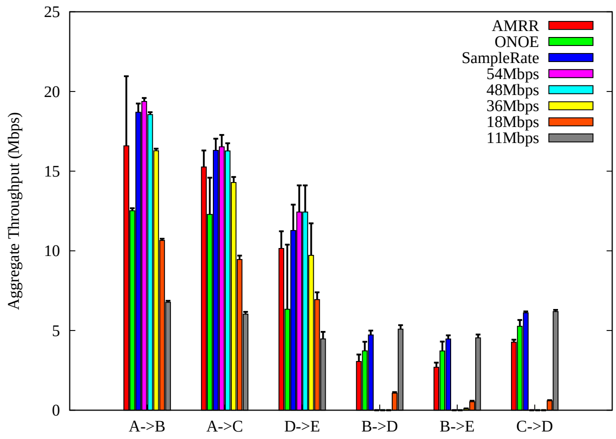
The first metric of interest is the average frame loss rate because this value drives the rate decision process of AMRR and ONOE algorithms. In Figure 3 we report the frame loss rates of some representative links of our mesh network, varying the transmission rate. First of all, let us consider link $A \rightarrow B$, which is a directional link established between two 19dBi grid antennas. The shown results indicate that this link is of very good quality, with negligible frame loss rates at all the bit rates. This is a quite expected result because high-gain directional antennas are aimed at providing high-quality radio links. The other two links reported in Figure 3 are omni-directional links. From the shown results, we can observe that the frame loss rates are not uniform over all transmission rates, but we can groups the bit rates into three categories. Generally, there is a group of transmission rates almost not affected by channel errors, which experience negligible frame loss rates. Then, we may have a group of rates that do not work at all, because most of the sent frames are lost. Finally, we may have a group of transmission rates with intermediate frame loss rates, which are probably due to fading or interference rather than simple channel attenuation. A common observation is that the most robust transmission rates are the set of 802.11*b* rates. On the contrary, the highest 802.11*a* rates are the most vulnerable to the channel conditions. However, the extent of this vulnerability is highly variable. For instance, on link $D \rightarrow E$ 54 Mbps and 48 Mbps rates show moderate frame loss rates, while on link $C \rightarrow D$ they do not work altogether. Note that these behaviors are consistent with the results of other experimental studies conducted on urban mesh networks [1, 7].

To clearly understand the dynamics of the rate adaptation algorithms, the average frame loss rate is not enough. It is also important to investigate the fluctuations of the channel quality and the time correlation of frame errors. To this end, we have also explored how much the frame loss rate of broadcast traffic fluctuates at different time scales. To measure the time correlation of frame errors a common metric used in prior work is the *Allan deviation* [1, 8]. Specifically, given a sequence of n values x_i, the Allan deviation is generally defined as:

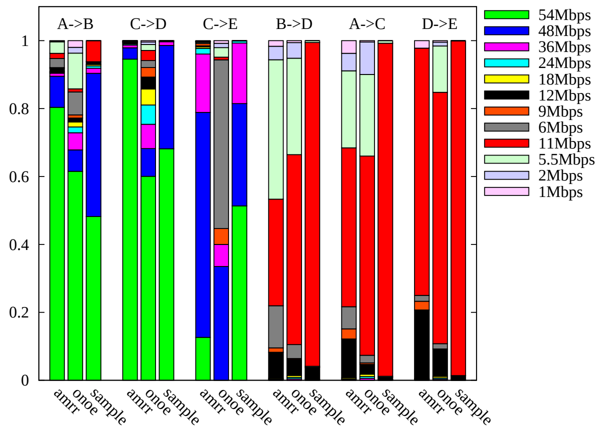
$$dev = \sqrt{\frac{\sum_{i=2}^{n} (x_i - x_{i-1})^2}{2n}} \,. \qquad (2)$$

In our case the values x_i are the frame error rates computed over a time interval T. If the Allan deviation is high for a given T value, this means that there is a loose correlation between adjacent intervals. In this case, T would represent an estimate of the characteristic error-burst length. In Figure 4, we have reported the Allan deviation for a few representative links of our mesh network for increasing averaging interval T. The plotted curves show that the Allan deviation is maximal for small intervals. Then, the deviation decreases almost linearly with T, and it is practically negligible for time intervals greater than one second, which is the sampling period used by both AMRR and ONOE to estimate the long-term frame loss rate. This behavior is typical of links affected by almost independent channel errors. Note that these trends are consistent with the results reported in [1], where the authors noted that non-bursty links may be predominant in urban 802.11-based mesh networks.

The implications of the above observations are the following. In our mesh network, the are transmission rates with negligible frame loss rate. These links are also quite stable, and observation periods of one second give reliable estimates of the long-term average frame loss rate. In other words, these links vary in loss rate by only a few percent from one second to the next. Consequently, we can expect that rate adaptation schemes based on loss rate thresholds will perform reasonable well on these links. To confirm this intuition, in Figure 5(a) we report the average throughput and the 95% confidence level, measured using the existing rate adaptation algorithms and fixed transmission rates for six representative links. From the shown results, we can observe that directional links (namely, $A \rightarrow B$ and $A \rightarrow C$) in general perform better than omni-directional links. It also useful to comment on the relationship between transmission rates and throughput of individual links as shown in Figure 5(a). Specifically, the two directional links $A \rightarrow B$ and $A \rightarrow C$, and the omni-directional link $D \rightarrow E$ show a graceful throughput degradation when decreasing the bit rate. On the contrary the other omni-directional links reported in Figure 5(a) (i.e.,



(a) Aggregate throughput



(b) Rate distribution (%)

Figure 5: Effect of outdoor link characteristics on different rate adaptation strategies.

$B \rightarrow D$, $B \rightarrow E$ and $C \rightarrow D$ links) show a drastic throughput drop for transmission rates higher than 11 Mbps, when they almost stop to work altogether. Note that the first type of links is usually classified as a *gradual* link, while the second type of links is classified as a *steep* link according to the link taxonomy proposed in [3]. These two categories of links are well representing typical channel conditions of medium-distance 802.11 links in urban mesh networks [1,7]. Thus, they are useful benchmarks for the performance of rate adaptation schemes.

As shown in Figure 5(a), SampleRate outperforms both AMRR and ONOE, and it closely approximates the best link performance. This is also in line with the results reported in [27], where the authors argue that SampleRate is the best algorithm for static settings, although the conclusions in [27] are based on results obtained from an indoor wireless network, while here we are considering outdoor mesh networks. AMRR works reasonably well in both gradual and steep links, but it is worse than SampleRate. Two main reasons can be identified. Firstly, AMRR permits only four consecutive retries for each frame and this induces a higher packet loss. In addition, AMRR occasionally tries very low transmission rates due to its particular setting of the MadWifi multiple rate retry strategy, which permits only one transmission attempt at the best transmission rate. Regarding ONOE, it works reasonably well for steep links (even better than AMRR), but is the worst for gradual links. This is due to the credit-based rate upgrade policy implemented in ONOE, which makes this algorithm very slow in increasing the rate. Thus, when ONOE selects an intermediate transmission rate affected by a moderate frame loss rate, which is more common for gradual links than steep links, it keeps using that bit rate for several seconds, even if the channel conditions improve.

5. CONCLUSIONS

In this paper we conducted a comparative study of the rate adaption algorithms implemented in the MadWifi driver. We conducted our tests in two experimental mesh testbeds, and we examined the performance of these algorithms in practical settings, including moderate to high medium contention levels and medium-distance 802.11 links with variable loss patterns. Based on our measurements we concluded that medium contention negatively affect the performance of existing rate adaptation algorithms, and frame loss-rate threshold-based strategies (i.e., AMRR and ONOE) stops working properly with high medium contention levels. In addition, we observed that if the time correlation between channel errors is at least of the order of the sampling period used to estimate the long-term loss rate characteristics, rate adaptation algorithms perform reasonably well. Therefore, there is not a clear performance gain in adjusting the transmission rates following the short-term variations of channel conditions, at least for static settings.

The results of this paper provide a better understanding of the challenges posed by the design of rate adaptation strategies suitable for urban mesh networks. Basing on the experimental evidence we can derive useful guidelines for the design of rate adaptation algorithms, and to propose new optimized strategies suitable for medium-distance, highly loaded 802.11 links. Thus, as future work we plan to investigate more efficient techniques to correctly estimate the medium contention level, in order to eliminate unnecessary rate downgrades. In addition, another direction we want to explore is how to make the sampling period used to estimate the long-term frame loss rates adaptive to the channel conditions, and to the variability of channel fluctuations. Note that not only link performance will benefit from a more efficient rate adaptation algorithm, but also routing performance can obtain an important advantage from more accurate rate decisions [4].

6. REFERENCES

[1] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Link-level Measurements from an 802.11b Mesh Network. *SIGCOMM Comput. Commun. Rev.*, 34(4):121–132, 2004.

[2] M. Alicherry, R. Bhatia, and L. Li. Joint Channel Assignment and Routing for Throughput Optimization in Multi-radio Wireless Mesh Networks. In *Proc. of ACM MobiCom*, pages 58–72, Cologne, Germany, Aug. 28–Sept. 2 2005.

[3] J. Bicket. Bit-rate Selection in Wireless Networks, February, 2005. Massachusetts Intitute of Technology, M.S. Thesis.

[4] J. Bicket, D. Aguayo, S. Biswas, and R. Morris. Architecture and Evaluation of an Unplanned 802.11b Mesh Network. In *Proc. of ACM MobiCom*, pages 31–42, Cologne, Germany, Aug. 28–Sept. 2 2005.

[5] L. Bononi, M. Conti, and E. Gregori. Runtime Optimization of IEEE 802.11 Wireless LAN Performance. *IEEE Trans. Parallel Distrib. Syst.*, 15(1):66–80, Jan. 2004.

[6] R. Bruno, M. Conti, and E. Gregori. Mesh Networks: Commodity Multihop Ad Hoc Networks. *IEEE Commun. Mag.*, 43(3):123–131, March 2005.

[7] J. Camp, J. Robinson, C. Steger, and E. Knightly. Measurement Driven Deployment of a Two-Tier Urban Mesh Access Network. In *Proc. of ACM MobiSys*, Uppsala, Sweden, June, 19–22 2006.

[8] K. Chebrolu, B. Raman, and S. Sen. Long-Distance 802.11b Links: Performance Measurements and Experience. In *Proc. of ACM MobiCom*, pages 74–84, Los Angeles, CA, USA, September 23–26 2006.

[9] T. Clausen and P. Jaquet. Optimized Link State Routing Protocol (OLSR). RFC 3626, October 2003.

[10] M. Delakis, K. Mathioudakis, N. Petroulakis, and V. Siris. Experiences and Investigations with Heraklion MESH: An Experimental Metropolitan Multi-Radio Mesh Network. In *Proc. of Tridentcom 2008*, Innsbruck, Austria, March, 18–20 2008.

[11] R. Draves, J. Padhye, and B. Zill. Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks. In *Proc. of ACM MobiCom*, pages 114–128, Philadelphia, PE, USA, Sept. 26–Oct. 1 2004.

[12] M. Genetzakis and V. Siris. A Contention-Aware Routing Metric for Multi-rate Multi-Radio Mesh Networks. In *Proc. of SECON 2008*, 2008.

[13] G. Holland, N. Vaidya, and P. Bahl. A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks. In *Proc. Mobicom 2001*, pages 236–251, Rome, Italy, July 2001.

[14] IEEE WG 802.11. IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, June 12 2007.

[15] A. Kamerman and L. Monteban. WaveLAN-II: A High-performance wireless LAN for the unlicensed band. *Bella Lab Technical Journal*, pages 118–133, Summer, 1997.

[16] J. Kim, S. Kim, S. Choi, and D. Qiao. CARA: Collision-Aware Rate Adaptation for IEEE 802.11 WLANs. In *Proc. INFOCOM 2006*, pages 1–11, Barcelona, Spain, April 23–29, 2006.

[17] M. Lacage, M. H. Manshaei, and T. Turletti. IEEE 802.11 Rate Adaptation: A Practical Approach. In *Proc. MSWiM '04*, pages 126–134, Venice, Italy, 2004.

[18] 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 Commun. Mag.*, 13(2):18–29, April 2006.

[19] MadWifi driver documentation. Onoe Rate Control. http://madwifi.org/wiki/UserDocs/RateControl.

[20] MADWIFI Driver Specification. Multiband Atheros Driver For WIFI. http://madwifi.org/.

[21] S. Pal, S. Kundu, K. Basu, and S. Das. IEEE 802.11 Rate Control Algorithms: Experimentation and Performance Evaluation in Infrastructure Mode. In *Proc. Passive and Active Measurement Conference (PAM) 2006*, Adelaide, Australia, March 30–31, 2006.

[22] Q. Pang, V. Leung, and S. Liew. A Rate Adaptation Algorithm for IEEE 802.11 WLANs Based on Mac-Layer Loss Differentiation. In *Proc. BROADNETS 2005*, pages 709–717, Boston, Massachusetts, USA, October 3–5, 2005.

[23] J. Pavon and S. Choi. Link Adaptation Strategy for IEEE 802.11 WLAN via Received Signal Strength Measurement. In *Proc. ICC'03*, volume 2, pages 1108–1113, Seattle, WA, USA, May 20–30, 2003.

[24] D. Qiao, S. Choi, and K. Shin. Goodput Analysis and Link Adaptation for IEEE 802.11a Wireless LANs. *IEEE Transactions on Mobile Computing*, 1(4):278–292, 2002.

[25] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly. Opportunistic Media Access for Multirate Ad Hoc Networks. In *Proc. MobiCom '02*, pages 24–35, Atlanta, Georgia, USA, September 23–28, 2002.

[26] S. Wang and A. Helmy. BEWARE: Background Traffic-Aware Rate Adaptation for IEEE 802.11 MAC. In *Proc. WoWMoM 2008*, New Port Beach, CA, USA, June 23–27, 2008.

[27] S. Wong, H. Yang, S. Lu, and V. Bharghavan. Robust Rate Adaptation for 802.11 Wireless Networks. In *Proc. Mobicom 2006*, pages 146–157, Los Angeles, CA, USA, September 24–29, 2006.