Evaluations of MadWifi MAC Layer Rate Control Mechanisms

Wei Yin*†, Konstanty Bialkowski*†, Jadwiga Indulska*† and Peizhao Hu†
*The University of Queensland,
School of Information Technology and Electrical Engineering
†National ICT Australia (NICTA)
Email: {Firstname.Lastname}@nicta.com.au

***Abstract*—The 802.11 standards specify several transmission rates that can be used at the MAC layer protocol to adapt the transmission rate to channel conditions. Such dynamic adaptations can improve per-hop performance in Wireless Networks and therefore can have impact on the Quality of Service provided for communicating applications. In this paper we present a comprehensive evaluation of the performance of four rate control mechanisms used by the MadWifi driver in Linux: Onoe, AMRR, SampleRate and minstrel. The evaluation of these four rate control mechanisms was carried out in our platform for controllable and repeatable experiments.**

I. INTRODUCTION

Each protocol layer has some impact on the communication Quality of Service (QoS) delivered to the applications. In this paper we discuss the performance of the rate control algorithms used in the MAC layer in Wireless Networks. Wireless channels are subject to various types of interference. Wireless networks are often deployed in community/neighbourhood networks, such as MIT's Roofnet[1] and CalRADIO-I in Cal-(IT)[2]. In these networks, people and vehicles are moving around and this generates unpredictable interference. In addition to this general interference, there is also the hidden terminal problem, i.e. interference created by overlapping transmissions of wireless routers due to the shared wireless medium and the carrier sense limitations. Both types of interference introduce variability in the channel quality - adaptive selection of transmission rates according to dynamic channel quality can improve per-hop performance by utilising the variety of transmission rates available in the 802.11 standards.

A number of algorithms have been proposed to adaptively select transmission rates according to various metrics, such as packet loss ratio, signal to noise ratio (SNR), and bit error ratio (BER). In this paper we evaluate the performance of four rate control algorithms implemented in the open-source MadWifi driver[3] which is a driver commonly used in Linux operating systems. These rate control algorithms are: Onoe [1], AMRR [2], [3], SampleRate [4] and minstrel [5]. Onoe adapts rate control based on the packet loss ratio. SampleRate periodically sends sample data packets to probe possible optimal rates. AMRR is based on a retry chain, whereas minstrel takes advantage of the retry chain and modifies the sampling approach used in SampleRate. These algorithms differ in the metrics that they use to make a decision on rate adaptation and therefore may differ in the capability to accurately capture the channel quality and select the appropriate transmission rate.

In our evaluation we measure the impact of the Received Signal Strength (RSS) and both types of interference on the performance of rate control mechanisms. The RSS is directly related to the SNR of the communication channel. Many existing rate control mechanisms use SNR as a metric, e.g. RBAR [6], CHARM [7] and FRAR [8]. However in most commercial network cards, the RSS or SNR measurements do not have enough granularity to be used as a metric directly. Hence, we evaluate the throughput for SampleRate, Onoe, AMRR and minstrel under various RSS values. Low throughput indicates the inability to select an appropriate transmission rate for a specific channel condition. Thus we also estimate the rate selection accuracy based on the percentage of packets that are transmitted out of the optimal rate. For the general interference we analyse throughput while varying three parameters: the duration, the interval and the strength of the interference. The interference duration is the time period which the interference lasts for, while the interference interval is the time period of adjacent interferences. For hidden terminal interference we investigate the impact of the hidden terminal traffic load and the hidden terminal link quality.

Our comprehensive performance evaluation of the four rate control mechanisms has the following features:

- The experiments are carried out in a platform that can produce controllable and repeatable environments and therefore the rate control mechanisms are compared in the same scenarios (the same offered load, parameters, etc). Most of the existing evaluations are either based on simulations or on experiments in networks. It is well known that simulations cannot appropriately account for the impact of interference as the simulators use very simple noise models. On the other hand it is difficult to achieve repeatable environments in ad-hoc experimental settings.
- These four rate control mechanisms have not been compared against each other and there is no evaluation of minstrel in the literature. minstrel is used in the MadWifi driver but it is also the default rate control algorithm in the new mac80211 framework in Linux Wireless that is used

---

[1] http://pdos.lcs.mit.edu/roofnet/
[2] http://www.calit2.net
[3] http://madwifi-project.org/

by a number of wireless network card drivers, including ath5k — a replacement of the MadWifi driver.

The remainder of the paper is organized as follows. In section II we briefly describe the four rate control mechanisms. The platform for controllable and repeatable experiments is described in section III, followed by the performance evaluations in section IV. The Related Work is described in section V and paper concludes in section VI.

II. RATE CONTROL MECHANISMS ON MADWIFI

In this section we briefly describe the rate control mechanisms used by the Madwifi driver.

Onoe calculates the credit for the current transmission rate based on the packet loss ratio. Onoe sets the initial rate to 24 Mbps in both IEEE 802.11g and IEEE 802.11a while it sets 11 Mbps in IEEE 802.11b. The credit for the initial rate is 0. Onoe observes received ACKs and increases the credit by 1 when less than 10% of frames in a time window of 1 second need retransmission (and the total frame transmissions are at least 10), while decreasing it by 1 otherwise. When the credit for the current rate reaches 10, Onoe increases the rate to the next higher level, whereas it decreases it to the next lower level if 10 or more frames have been sent and more than 50% of the frame transmissions failed during the last period. When the rate is changed the credit is reset to 0. Onoe is conservative, because it does not increase the current transmission rate when it detects good channel quality, but waits until the credit value reaches the threshold.

SampleRate periodically sends a number of data packets as sample packets at a certain rate other than the current rate to gather statistics and make rate selection decisions. For each transmission rate, SampleRate calculates the average transmission time (ATT) every 10 seconds based on the transmission results. The rate which has the smallest ATT is selected for normal packets in the next control period (10s). The sample packets are transmitted every tenth packet, for which the transmission rate is randomly selected from a set of available rates that have smaller ATT compared to that of the current transmission rate for normal packets. SampleRate takes the highest rate when it starts and stops using a rate when it experiences four successive failures.

AMRR uses a set of four pairs of rate and transmission counts (r0/c0, r1/c1, r2/c2, r3/c3)for each frame to be sent. If the first transmission of the frame at the rate r0 fails, AMRR keeps sending the frame at the rate r0 for c0-1 times. If these retransmissions are not successful, AMRR selects the rate r1 and tries c1 times. When the frame has been transmitted for c0+c1+c2+c3 times, AMRR abandons this frame and the transmission status is updated. This mechanism is called multi rate retry and is used to handle short-term channel variations. In AMRR rate r3 is always set to the minimum transmission rate, while r1 and r2 are set to the immediate lower available rates of r0 and r1, respectively. The determination of r0 is based on the previous value of r0 and the transmission results of the elapsed period. If more than 10 packets were transmitted in the previous period (default is 1 second) and less than 10% of the transmissions failed AMRR increases the retry chain values to the next higher level. The retry chain fails if more than 33% of the packet transmissions failed during the previous period and in this case the transmission rate r0 is decreased to the next lower level. In AMRR, the long-term channel variation is handled by a binary exponential back-off mechanism which adapts the length of the period to change the value of the four rate-count pairs.

minstrel uses a multi-rate retry chain with four rate-count pairs (r0/c0, r1/c1, r2/c2, r3/c3) as in AMRR. The determination of the values for the four rates in the retry chain is based on the measured throughput and the probability of success for each rate. The throughput and the probability of success for each rate are calculated every 100 ms. minstrel uses 10 percent of frames to randomly try other rates to collect statistics therefore transmitted frames are classified into normal and sample frames. For the sample frames, when the randomly selected rate is slower than the rate which has the best throughput, the rate which has the best throughput is chosen as r0 and r1 is set to the randomly selected rate. r2 is set to the rate which achieves the best probability of success. r3 is set to the lowest base rate. When the randomly selected rate is higher than the rate which has the best throughput, r0 is set to the randomly selected rate, r1 is set to the rate which has the best throughput, r2 is set to the rate which achieves the best probability of success and r3 is set to the lowest base rate. For a normal frame, the rate which has the best throughput is chosen as r0, the rate that has the next best throughput is selected as r1, r2 is set to the rate that has the best probability of success and r3 is set to the lowest base rate.

III. CONTROLLABLE EVALUATION PLATFORM

We carried out evaluation studies of the performance and robustness of the MadWifi rate control mechanisms under three different scenarios: (A) performance in fixed channel conditions, (B) presence of general interference (that is, variation in link quality/channel conditions) and, (C) the hidden terminal problem. In this section, we present our evaluation platform and experiment setup. Within the evaluation platform, wireless signals are carried by co-axial cables rather than being transmitted over the air by antennas.

Fig. 1(a) shows the experiment setup that we designed to emulate the scenarios (A) and (B), while we use the setup shown in Fig. 1(b) to emulate the scenario (C).

As shown in Fig. 1(a) we connect a traffic source to a traffic sink using co-axial cables and vary the link quality/channel conditions using a variable attenuator (Vaunix LabBrick LDA-602). This topology is used for both scenarios (A) and (B). In scenario (A) the variable attenuator is remains fixed in value during each measurement, whereas for scenario (B), the attenuation (that is, the path loss) is varied during the measurement period. As shown in Fig. 1(b), we form two links with two traffic sources A, B and a traffic sink. This is used in scenario (C). Wireless signals from each traffic source are combined by an 8-way signal splitter/combiner. The variable attenuator is placed between traffic source A and the traffic sink; hence allowing the the degree of interference seen by the traffic source B to be varied.

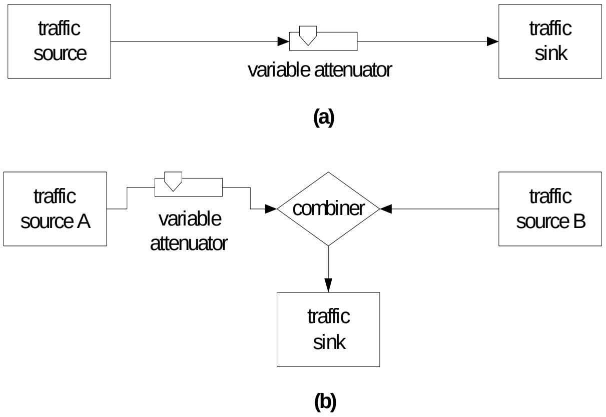


Fig. 1. Experiment topologies used in this paper.

Each traffic source and sink runs on a single-board computer equipped with a Wistron CM9 Atheros wireless card[4]. The computer nodes run Linux operating system (kernel 2.6.28) and MadWifi driver (version 0.9.4 svn 4029). The evaluation results are measured using iperf[5] on each node (that is, iperf clients as the traffic sources and iperf server as the traffic sink).

To ensure absolute minimal interference from the environment, all experiments are performed using IEEE 802.11a. A 20dB fixed attenuator is attached to each traffic node in each scenario to avoid damage to the radio and to ensure a fully disconnected link by adjusting the variable attenuator (operation range 0–63dB). In the scenario (C), we also adjust the transmission power to 13dBm. These configurations make sure all links can achieve maximum throughput when the variable attenuator's value is 0 and the corresponding topologies can be formed by varying the variable attenuator's value. Each scenario setup is verified and confirmed multiple times manually by various tools. The combination of the variable attenuator, the fixed attenuator, cables and splitters forms the total path loss in between the source and the sink. For all results the path loss value will be used to describe the channel conditions in this environment.

The measurements are captured using a packet sniffing tool (tcpdump). Each measurement contains information regarding the achievable throughput and the number of packets that have been sent at each rate. A parser library (Banjax[6]) is used to analyse the measurement files. In the experiments, we set the UDP packet size to 1000-byte, corresponding to a maximum throughput of 28 Mbps in a good channel. Each measurement is performed for 60 seconds, and the results reported in this paper are the average of 10 different runs of each experiment.

IV. EVALUATION AND DISCUSSION

Using the evaluation platform described in the previous section, the performance of four rate control mechanisms are evaluated. This is done in three scenarios. The first evaluates performance under fixed channel conditions with different average signal to noise ratio (SNR). The noise experienced by this system is fixed, therefore high and low SNR conditions are achieved by varying the path loss over the channel between source and sink. The second scenario compares performance under a general interference environment. Here, the impact of interference duration, interference interval and interference strength are evaluated. In the last scenario, the effect of hidden terminal interference is evaluated using varying amounts of hidden terminal traffic and heterogeneous links.

A. Fixed channel conditions

The evaluation of rate control mechanisms under fixed channel conditions or varying signal to noise ratio (SNR) is done by varying the path loss between the source and destination between 56 dB and 86 dB. For each value of path loss UDP throughput is measured at the receiver. For this scenario the transmitter power is configured to 8 dBm and the iperf traffic load is set to 35 Mbps. To compare the rate control mechansisms to the optimum transmission rate, all fixed transmission rates are also evaluated in the same channel environments.

Fig. 2(a) shows the throughput achieved by the four rate control mechanisms as a function of the path loss. Higher path loss means lower received signal strength. In general, the throughput remains similar for SampleRate, minstrel and AMRR when the received signal strength is above a threshold (path loss below 61 dB). Furthermore, the throughput for all mechanisms begin to decrease when the path loss exceeds 61 dB. In addition, above 75 dB path loss they all converge and the link is broken at 86 dB path loss. In this scenario *minstrel* achieves the best throughput across all channels, except 66 dB path loss. Onoe has the lowest throughput for channels below 66 dB path loss.

In Fig. 2(b), the percentage of packets which are transmitted at the optimum rate is shown. The optimum transmission rate is the fixed transmission rate which is able to achieve the highest throughput. For minstrel, only around 5% of the tested channels are not transmitted at the optimum rate. Additionally, when the path loss is below 61 dB, the percentage is almost zero. Comparing minstrel and SampleRate [4], minstrel has 10% less packets transmitted at non optimum transmission rates for most of the values of path loss. This is because minstrel transmits packets first at the rate which has the best throughput. This mechanism allows minstrel to be able to select the optimum transmission rate, thereby achieving higher throughput overall. However, minstrel fails to identify the best optimal rate at 66 dB. The optimal rate for 66 dB is 48 Mbps, however for minstrel 44% packets are sent at 54 Mbps.

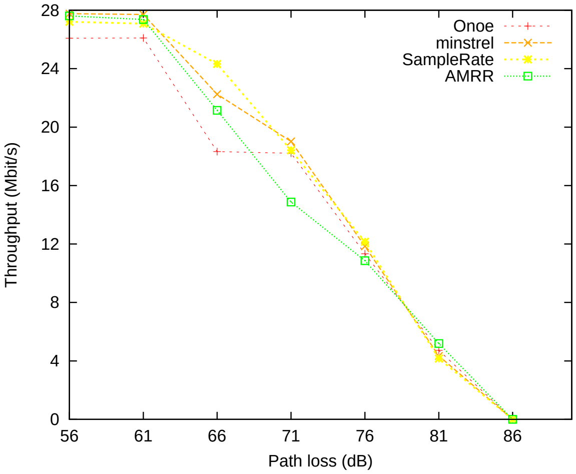
For SampleRate, the optimum transmission rate at 81 dB path loss is unable to be identified. Almost 100% of packets are transmitted at 6 Mbps, and the optimal rate is 9 Mbps. Thus it achieves the lowest throughput. At this path loss value, packets are sampled at 12 Mbps, for all of the 10 experiment runs. The reason that SampleRate has the lowest throughput for this channel is due to the fact that no packet is sampled at the optimal rate, hence the statistics necessary for selecting this rate are not available. Therefore, SampleRate has the problem for the mechanism which decides which rate is to be sampled for packets.

---

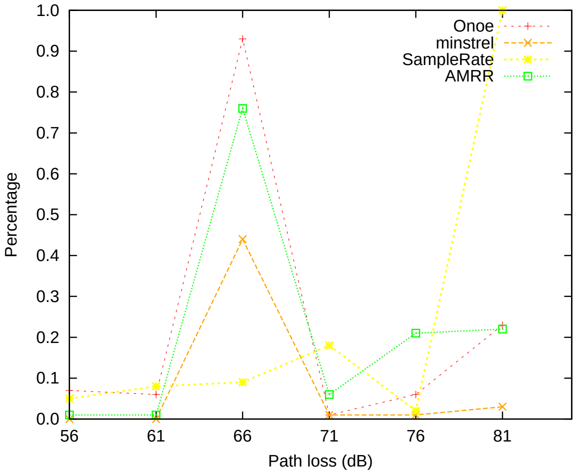
[4] http://www.wneweb.com/Networking/mini-PCI.htm

[5] http://sourceforge.net/projects/iperf/

[6] http://code.google.com/p/banjax/



(a) Throughput



(b) Percentage of packets at non-optimum TX rate

Fig. 2. Auto rate performance vs path loss (received signal strength).

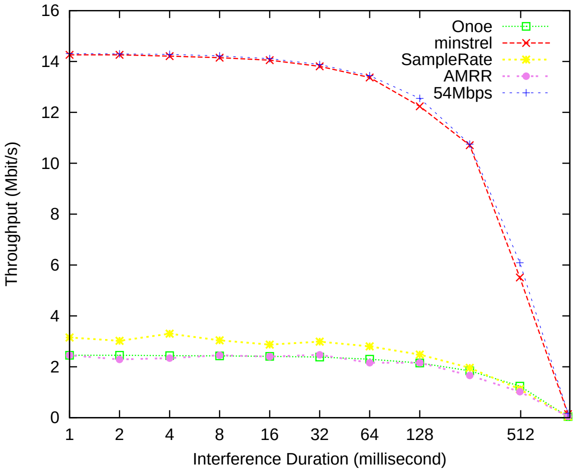
Using both Fig. 2(b) and Fig. 2(a), it can be seen that the more packets that are not transmitted at the optimum rate, the lower the throughput will be. For example, Onoe at 56 dB and 66 dB, AMRR at 76 dB, and SampleRate at 81 dB. Thus, calculation of the percentage of packets that are transmitted at or not at the optimal rate is necessary to identify the reason for different performance.

B. Impact of Interference

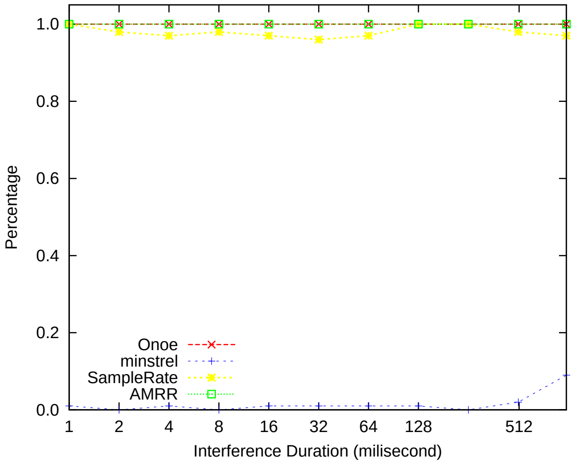
Channel quality in a real environment is subject to randomly moving people and vehicles, which generates unpredictable interference. Thus in this scenario, the performance of the four rate control mechanisms is evaluated under general interference by varying the duration, interval and strength of interference.

1) Interference Duration: Firstly, the impact of interference duration on rate control mechanisms is evaluated. The interference duration is varied exponentially from 1 ms to 999 ms and set the path loss during interference to 86 dB. This strength of interference (86 dB) is high enough to disable the connection, as shown in the previous experiment in Fig. 2(a). The interval between interference is set to 32 ms. Path loss is set to 46 dB during the period between interference, which is able to support 54 Mbps transmission rate. In addition to testing each of the rate control mechanisms, each fixed transmission rate is tested under these conditions to provide information on what the optimum transmission rate would be for each of the scenarios.

In Fig. 3(a) and Fig. 4, the throughput of each rate control mechanism and fixed transmission rate is shown as a function of interference duration. In each case, throughput drops as the interference duration increases. This is especially visible when the duration increases above the interference interval. When the interference duration is above 999 ms, the connection between transmitter and receiver is disconnected. When the duration is 999 ms, all rate control mechanisms and fixed transmission rates have almost zero throughput.



(a) Throughput



(b) Percentage of packets at non-optimum TX rate

Fig. 3. Auto rate performance vs interference duration.

In general, minstrel achieves 4 times greater throughput than the other three mechanisms. The throughput achieved approaches the throughput of the highest and optimum fixed rate (54 Mbps). In Fig. 3(b), the percentage of packets not transmitted at the optimal rate is shown. For minstrel, 99.99% of packets are transmitted at the optimum rate of 54 Mbps.

From Fig. 3(b), the reason why AMRR, SampleRate and Onoe achieve very low throughput is also seen. Here the

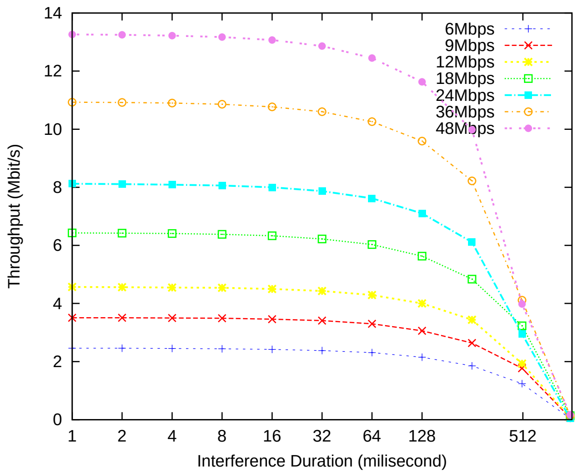


Fig. 4. Throughput versus interference duration (fixed transmission rate).

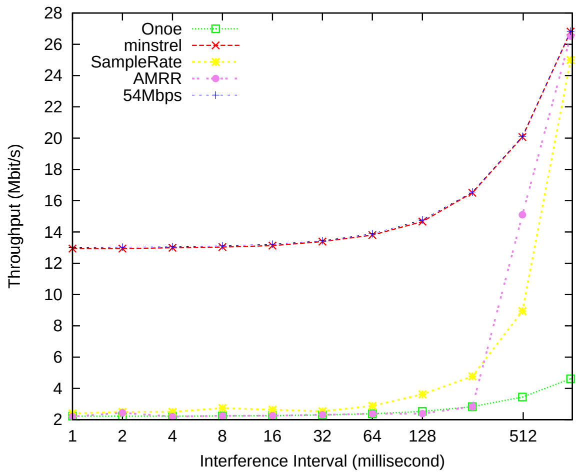
percentage of packets that are transmitted at the non-optimal rate is more than 95%, hence they all achieve low throughput. For Onoe, all packets are transmitted at 6 Mbps, due to the credit metric in Onoe not achieving the threshold required to increase transmission rates. For SampleRate, the metric average transmission time (ATT), does not work well when there is strong interference. The statistics for SampleRate are shown in Table I. In this table it can be seen that SampleRate does sample packets at 54 Mbps and other rates. However, it is unable to identify 54 Mbps as the optimal rate. From Fig. 4, it can be seen that any transmission rate above 6 Mbps achieves better throughput than SampleRate. However, SampleRate does not select these higher rates due to the metric ATT not being able to identify these as a more optimum rate.

From this experiment it can be seen that the duration of the interference has a significant impact on SampleRate, Onoe and AMRR, because they are unable to select the optimal rate to increase throughput. Whereas for minstrel the duration of interference does not have significant impact. This is because it is capable to select the rate which has the best throughput across all tested interference duration values.

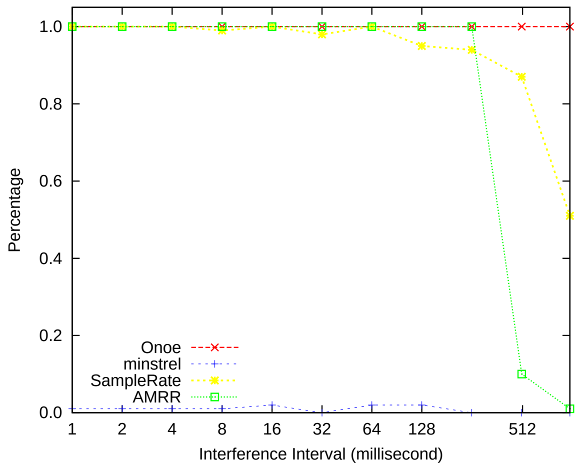
2) Interference Interval: Next, the impact of the interval between interference on rate control mechanisms is evaluated. We exponentially vary the interval between interference from 1 ms to 999 ms. The interferences duration is fixed to 64 ms. The interference strength is set to 86 dB as before. During the period between interference, we set the attenuator to 46 dB.

Fig. 5(a) shows the throughput of each rate control mechanism as the function of interference interval. For all rate control mechanisms, throughput increases as the interference interval is increased, and when the interval is above 64 ms (the interference duration) the increase is more visible. When the interval becomes 999 ms, all rate control mechanisms other than Onoe achieve high throughput. However, Onoe remains at quite low throughput.

Generally, minstrel shows more than 5 times throughput compared to other mechanisms. Again, the throughput is similar to that achieved by 54 Mbps fixed rate. In Fig. 5(b), only 0.01% packets for minstrel are sent at a non-optimal rate, hence the reason for such good throughput. Therefore,



(a) Throughput



(b) Percentage of packets at non-optimum TX rate

Fig. 5. Auto rate performance vs interference interval.

the interval between adjacent interference has no significant impact on minstrel, because minstrel is capable of selecting the optimal rate for varying interference intervals.

Another finding is that the impact of interference interval on SampleRate and AMRR depends on the ratio of interference interval to interference duration. When the ratio is less than one, they are achieve very low throughput, whereas it begins to have less impact on AMRR and SampleRate when the ratio is greater than 1. The phenomenon is shown in Fig. 5(a). However, even if the ratio is very high, Onoe is still stuck in low throughput.

3) Interference Strength: In the last experiment regarding interference, the impact of interference strength on rate control mechanism is evaluated. This is done by varying the path loss during interference between 63 dB to 84 dB. A high quality channel is used in between periods of interference, with 46 dB path loss. Both the interference duration and interference interval are set to 100 ms.

Fig. 6 shows the throughput of each rate control mechanism as the function of interference strength. In general, the impact of interference strength has more impact on Onoe, AMRR and SampleRate than minstrel. The throughput for Onoe, AMRR and SampleRate degrades as the interference strength

TABLE I
NUMBER OF PACKETS TRANSMITTED AT EACH RATE FOR SAMPLERATE.

Duration	6Mbps	9Mbps	12Mbps	18Mbps	24Mbps	36Mbps	48Mbps	54Mbps
1	16145	227	2341	1689	1848	647	1786	108
2	16671	65	2295	1157	1587	548	990	396
4	15681	156	2870	1622	1816	1299	1682	842
8	16310	90	2600	1095	1662	870	902	399
16	17101	84	1724	632	673	930	648	756
32	15587	247	2982	1509	1097	607	665	826
64	15967	150	1725	498	1674	587	712	687
128	15238	130	1554	465	972	763	376	3
256	13583	62	568	133	203	398	482	20
512	8344	1	101	0	11	162	162	174
999	208	6	0	13	0	38	38	8

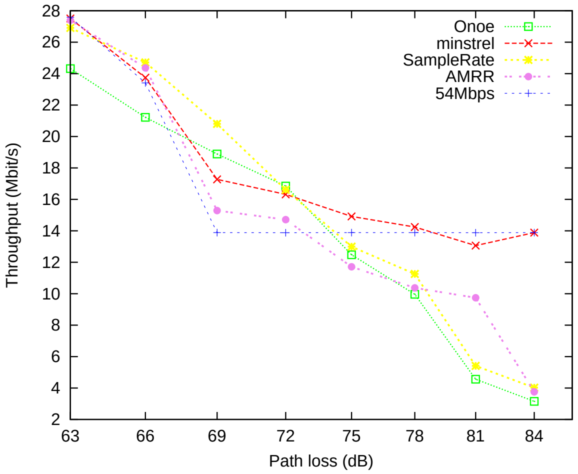


Fig. 6. Throughput vs interference strength.

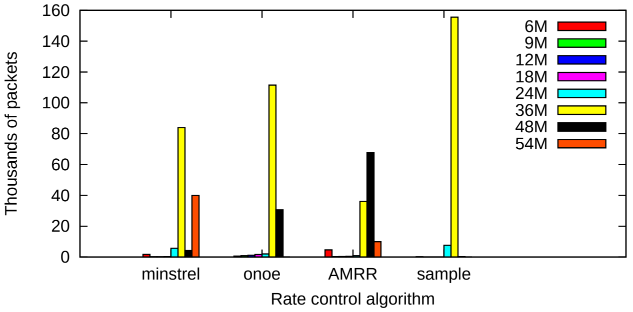


Fig. 7. Number of packets sent at each rate for 69 dB.

increases. However, when the interference strength is higher than 78 dB, the interference strength has less impact on minstrel.

When the path loss is 69 dB, the four rate control mechanisms achieve a different order of throughputs. To analyse this, the distribution of transmission rates used was plotted in Fig. 7. For this situation, the optimum transmission rate at each of the channel conditions 46 dB and 69 dB path loss is 54 Mbps and 36 Mbps respectively. We find in Fig. 7 that minstrel correctly identifies the two rates, but is unable to switch to each rate fast enough. From the minstrel documentation we note that rate adaptation occurs every 100 ms. In our scenario, this time is used for both the interference duration and the interference interval. It is possible that when minstrel devices to change transmission rate, the channel conditions have already changed. For SampleRate, the overall throughput is highest, and all packets are sent at 36 Mbps. SampleRate uses a time window of 10 seconds for ATT calculations and rate selections, hence it cannot detect 100 ms variations. However, the throughput achieved by SampleRate is not the achievable highest throughput for 69 dB. Hence it is important to note that reaction time for channel change in addition to ability of capturing channel change are significant factors for rate control mechanisms.

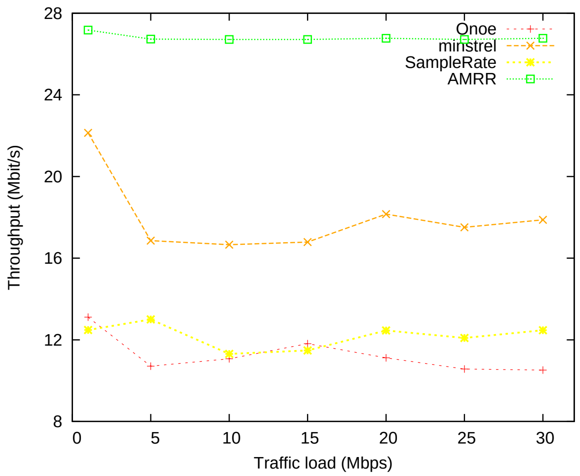
C. Hidden terminal Interference

Unlike the previous scenario, this one looks at interference caused by other 802.11 wireless devices. Specifically, this is concerned with hidden node interference with two variables. The first variable concerning the load of the hidden terminal traffic, the second concerns the link quality of the link under test. For this scenario, topology (b) in Fig. 1 is used, and each of the three computers is configured to use the same rate control mechanism. Before running the experiments it was confirmed that full throughput can be achieved on both of the single links independently.

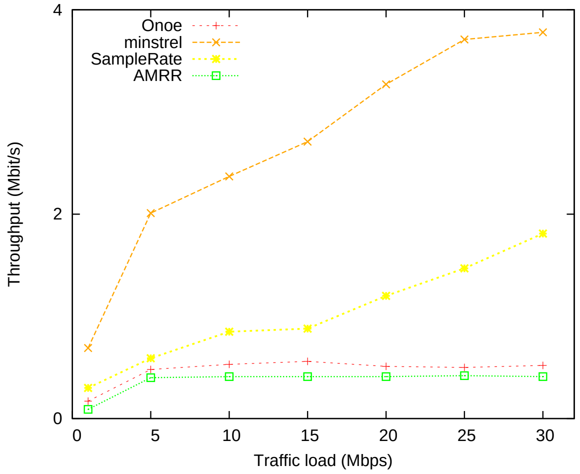
1) Heterogeneous Traffic Load: To evaluate the effect of hidden terminal traffic load, an iperf client on traffic source B is used to send packets with an offered load of 35 Mbps, creating a high load link. On traffic source A, the offered load is varied between 1 Mbps to 30 Mbps, creating a low traffic link.

Fig. 8(a) and Fig. 8(b) show the throughput of the high load (link B) and low load (link A) links as a function of the traffic load on the low load (link B) link respectively. Generally, the same rate control mechanism shows higher performance in high load link than low load link. Furthermore, minstrel shows the best performance. This is because it achieves the second highest throughput on the high load link while it performs the best on the low load link. Although AMRR has the best throughput on the high load link, it achieves the worst on the low load link. As minstrel achieves the best fairness in the hidden terminal scenario with heterogeneous traffic load, it has the best overall performance.

2) Heterogeneous Link Quality: The impact of heterogeneous link quality between is evaluated using the same topol-

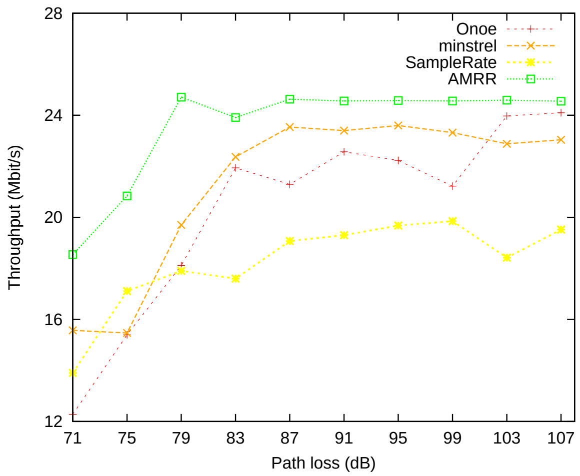


(a) Throughput on high load link.

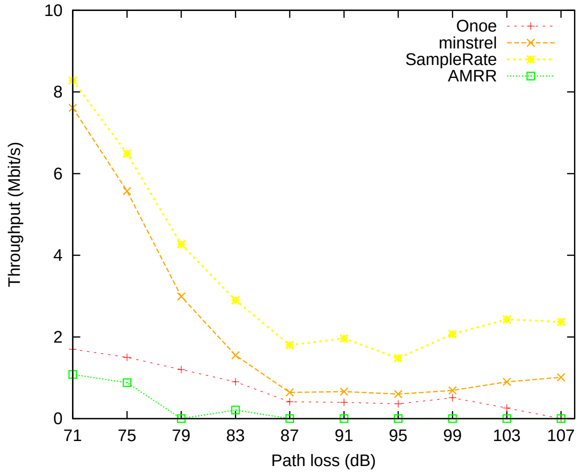


(b) Throughput on low load link.

Fig. 8. Auto rate performance vs hidden terminal traffic load.



(a) Throughput on high quality link.



(b) Throughput on low quality link.

Fig. 9. Auto rate performance vs hidden terminal link quality.

ogy as for the heterogeneous traffic load. In this experiment, the path loss between traffic source B and the sink is fixed to 71 dB (a good link) and the path loss between traffic source A and the sink is varied between 71 dB and 108 dB thereby making a bad link. Both links are configured to use an offered load of 35 Mbps.

Fig. 9(b) and Fig. 9(a) respectively show the throughput as a function of link B's path loss, for the good link (link A) and the bad link (link B). As the path loss in the bad link increases, a larger difference in link quality is seen between the links. SampleRate performs the best on the bad link, but achieves the lowest throughput on the good link. On the other hand, AMRR achieves the highest performance on the good link, but exhibits starvation on the bad link. Onoe also exhibits starvation on the bad link. The best overall performance is achieved by minstrel. When both links are active, it does not behave too aggressively on the good link, nor does starvation occur on the bad link.

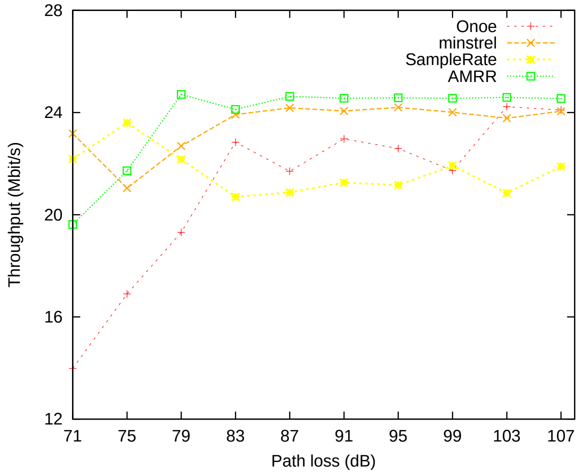


Fig. 10. Aggregate throughput for heterogeneous links.

The aggregate throughput achieved over both links when link quality of link B is varied is shown in Fig. 10. It shows that AMRR achieves the highest aggregate throughput. However, the throughput gain is achieved with the sacrifice of the bad link. minstrel achieves the second highest aggregate throughput, however provides more fair sharing of the channel in terms of air time. Minstrel is able to achieve more fairness with relatively high aggregate throughput, and hence achieves good performance in the hidden terminal environment.

V. RELATED WORK

There exist a number of rate control mechanisms and these mechanisms can be classified into three groups:

ACK Based Mechanisms. The ACK based mechanisms adapt rates according to transmission results estimated based on received acknowledgements (ACKs). The four rate control mechanisms that we evaluate in this paper are ACK based mechanisms. Other algorithms in this category are RRAA, ARF and AARF. Similar to Onoe and AMRR, RRAA [9] calculates the packet loss ratio. In addition, it uses an adaptive RTS (A-RTS) mechanism to eliminate communication overhead, in which only packets in a window are sent with RTS packets. Instead of calculating packet loss ratio, ARF [10] and AARF [3] measure the number of consecutive successful transmissions to conduct rate control.

SNR Based Mechanisms. Different from ACK based mechanisms, SNR based mechanisms measure the SNR to select transmission rates. These mechanisms include RBAR [6], CHARM [7] and FARA [8].

BER Based Mechanisms. BER (bit error ratio) is a more fine-grained metric than packet loss ratio or SNR. To estimate the BER metric, SoftRate [11] calculates the average BER for the current transmission rate over all bits in frame. However, there is limitations in calculating the BER using conventional network cards — no access to mine data from the error correction algorithms.

The existing evaluations of rate control algorithms have are mostly based on simulations, on experiments in real life deployment or in controllable platforms. The advantage of simulation is that the results are repeatable, while the disadvantage is that it cannot accurately simulate the real environment dynamics, such as movement of obstacles (i.e., moving vehicles) and presence of various types of interference (i.e., microwave). Experiments carried out in real life deployment often fail to take into account environmental metrics/parameters that are responsible for the throughput change. Moreover it is extremely difficult to recreate the same experiment environment for each algorithm under test. The controllable platform we use in our evaluation resolves the above issues.

As the paper aims to evaluate the four rate control algorithms currently supported by the Madwifi driver, we present a survey of existing work in this area. It should be noted that only three mechanisms (Onoe, AMRR and SampleRate) have been evaluated in the literature; the performance of minstrel has not been comprehensively studied yet.

A. Simulation Based Evaluation

Mythili, et al. [11] evaluate the performance of SoftRate, RRAA, SampleRate, an RBAR-like mechanism and a CHARM like mechanism using ns3 simulations. The RBAR-like mechanism enables the Acknowledgements to carry the SNR feedback and the RTS/CTS mechanism is disabled, while the CHARM-like mechanism uses the average SNR over multiple frames. The topology in the experiment is a N-clients to 1-AP (Access Point) scenario and the traffic is modelled as TCP flows with a packet size of 1400 bytes. The slow fading channel is simulated by the sender moving away from the receiver at a walking speed while the fast fading channel is simulated by the speed of a vehicule. The interference is simulated by the imperfect carrier sense and the carrier sense probability is set to between 0 and 1. In the slow fading scenario, SoftRate shows 4 times higher throughput than SampleRate and the authors show that SampleRate cannot adapt fast enough to channel slow fading. The slow fading scenario is similar to our interference interval scenario (Fig. 5(a)). Our finding confirms that SampleRate achieves very low throughput. However, minstrel achieves more than 4 times the throughput of SampleRate in spite of being an ACK based mechanism. It is important to note that SoftRate cannot be implemented in the current network cards as it requires information from the error correction algorithms. In addition, the authors show that SampleRate is resilient to interference [11]. However, in our scenario, we find that if the interference is strong (so that link outages occur), SampleRate performs badly. In this case its performance depends on the ratio of interference interval to interference duration. Specifically, when the ratio is higher than the threshold (256/64), SampleRate's performance improves and ultimately meets the performance of the optimal fixed rate and minstrel (Fig. 5(a)). When the ratio is smaller than the threshold, SampleRate achieves almost 0 throughput.

B. Real Environment Evaluation

ARF, AARF, Onoe and SampleRate are evaluated in [4] on a 45-node indoor test-bed and a 38-node outdoor test-bed. Packets in [4] are 1500 byte UDP packets. The main finding is that Onoe and SampleRate perform very close to the performance of the best fixed rate. However, we find that Onoe fails to select the best static/fixed bit-rate at 66 dB as shown in Fig. 2(b) and more than 90% packets are transmitted out of the optimal rate. Another finding in [4] is that the only links on which Onoe performs poorly are the low quality links in the 802.11a indoor networks and outdoor 802.11b networks. However, in our experiments Onoe achieves the lowest throughput of the four evaluated rate control mechanisms in good quality links as shown in (Fig. 2(a) and Fig. 2(b)). This is because of its conservative reaction to good quality channels.

Wong, et al. [9] evaluate RRAA, SampleRate, ARF, AARF and Onoe in a realistic IEEE 802.11a/b networks with various settings, such as static/mobile clients, with/without hidden stations. In the non-mobile client scenario, UDP based results for four clients in the IEEE 802.11a network show that RRAA has the best performance and SampleRate performs differently for differently positioned groups of clients. SampleRate shows better performance than ARF and AARF in IEEE 802.11b networks. In a mobile-client scenario with UDP packets, SampleRate has lower throughput than RRAA, ARF and AARF. In the hidden terminal scenario, SampleRate shows lower throughput than RRAA. The experiments in [9] are conducted in real environment with limited nodes and the result is subject to various client positions. It is not clear how the hidden terminal scenario is created in the real environment and it is obvious that it will be difficult to achieve repeatable evaluation environment. The impact of general interference is not evaluated.

SampleRate, AMRR and Onoe are evaluated in [12] in both indoor and outdoor environment. The indoor environment has 12 nodes on one floor, while the outdoor environment has five wireless routers deployed on the rooftops of three buildings. Experiments are conducted in the 802.11g mode using UDP traffic. In good quality channels in the indoor environment where the highest throughput is achieved by 54Mbps, AMRR and SampleRate show better performance than Onoe. This is confirmed by our experiment in scenario (A). A high contention channel condition is formed in the indoor environment by enabling eleven senders to transmit packets concurrently. In the scenario, SampleRate sends only 7% of frames at the optimal rate. AMRR and Onoe perform even worse than SampleRate. In low-mid medium contention chennel condition, SampleRate also shows better performance than AMRR and Onoe. However, it is not clear how to determine the degree of contention in the environment. In the outdoor environment, SampleRate also shows better performance than AMRR and Onoe. However, the impact of interference duration, interference interval, interference strength and the hidden terminal interference on rate control mechanisms are not evaluated.

C. Controllable Environment

Our evaluation is much closer to the evaluation carried out by Camp et.al [13] on the WARP platform. Three rate control mechanisms plus two of their extensions are implemented and evaluated - no rate control mechanisms from the MadWifi driver are evaluated. The evaluated mechanisms include ARF, RRAA, RRAA with the A-RTS extension, RBAR, and RBAR with the OAR extension. OAR [14] is an extension which enables rate adaption mechanisms to send back to back packets. The authors use a Spirent Communication Channel Emulator (SR5500) to simulate various channel conditions. They vary the coherence time from 100 μs to 100 ms on a single channel with a high average RSS of -40 dBm to evaluate the throughput. The coherence time refers to the time interval over which the channel condition is sufficiently constant to decode the received symbols. The coherence time in the experiment is similar to the interference interval scenario in our experiment. However the impact of hidden terminal is not evaluated. CHARM, Onoe, SampleRate and AMRR are also evaluated in [7] in a controllable environment. The authors show that Onoe and AMRR perform poorly in the hidden terminal scenario. We found it is true in low quality or low load links. However, we also found that AMRR can perform the best on the good quality and high load links. The impact of general interference is not evaluated in [7].

VI. CONCLUSION

In this paper we presented the performance evaluation of four rate control mechanisms implemented in the MadWifi driver: minstrel, SampleRate, Onoe and AMRR. These rate control mechanisms are ACK based, i.e. they use information on packet loss to make decisions on transmission rate adaptations. The evaluation was carried out in a platform that provides a controllable and repeatable environment. While three of these four rate control mechanisms have been already evaluated in the past their evaluation was not as as comprehensive as the one presented in this paper. We compared the rate control mechanisms in the same environment (the same offered load and parameters) and evaluted the impact that the three different scenarios: (i) fixed channel conditions with different average signal to noise ratio, (ii) external interference, and (iii) hidden terminal interference, have on the rate control performance. The important finding of this evaluation is that the minstrel's performance is not only far superior to the performance of SampleRate, Onoe and AMRR, but also that it has a very good performance compared to rate control mechanisms that are SNR or BER based.

ACKNOWLEDGEMENT

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program; and the Queensland Government.

REFERENCES

[1] "Onoe specification," http://sourceforge.net/project/madwifi.

[2] M. Lacage, M. H. Manshaei, and T. Turletti, "Ieee 802.11 rate adaptation: a practical approach," *INRIA Research report number 5208*, 2004.

[3] ——, "IEEE 802.11 rate adaptation: a practical approach," in *MSWiM '04: Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*. Venice, Italy: ACM, 2004, pp. 126–134.

[4] J. C. Bicket, "Bit-rate selection in wireless networks," Ph.D. dissertation, MIT Master's Thesis, 2005.

[5] "minstrel specification," http://madwifi-project.org/browser/madwifi/trunk/ath_rate/minstrel/minstrel.txt.

[6] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive mac protocol for multi-hop wireless networks," in *Proc. of ACM MobiCom*, 2001.

[7] G. Judd, X. Wang, and P. Steenkiste, "Low-overhead channel-aware rate adaptation," in *MobiCom '07: Proceedings of the 13th annual ACM international conference on Mobile computing and networking*. Montreal, Quebec, Canada: ACM, 2007, pp. 354–357.

[8] H. Rahul, F. Edalat, D. Katabi, and C. G. Sodini, "Frequency-aware rate adaptation and mac protocols," in *MobiCom '09: Proceedings of the 15th annual international conference on Mobile computing and networking*. Beijing, China: ACM, 2009, pp. 193–204.

[9] S. H. Y. Wong, H. Yang, S. Lu, and V. Bharghavan, "Robust rate adaptation for 802.11 wireless networks," in *MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking*. Los Angeles, California, USA: ACM, 2006, pp. 146–157.

[10] A. Karmerman and L. Monteban, "Wavelan-ii: A high-performance wireless lan for the unlicensed band," *Bell Labs Technical Journal*, 1997.

[11] M. Vutukuru, H. Balakrishnan, and K. Jamieson, "Cross-layer wireless bit rate adaptation," in *SIGCOMM '09: Proceedings of the ACM SIGCOMM 2009 conference on Data communication*. Spain, Barcelona: ACM, 2009, pp. 3–14.

[12] E. Ancillotti, R. Bruno, and M. Conti, "Experimentation and performance evaluation of rate adaptation algorithms in wireless mesh networks," in *PE-WASUN '08: Proceedings of the 5th ACM symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*. Vancouver, BC, Canada: ACM, 2008, pp. 7–14.

[13] J. Camp and E. Knightly, "Modulation rate adaptation in urban and vehicular environments: cross-layer implementation and experimental evaluation," in *MobiCom '08: Proceedings of the 14th ACM international conference on Mobile computing and networking*. San Francisco, California, USA: ACM, 2008, pp. 315–326.

[14] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, "Opportunistic media access for multirate ad hoc networks," in *MobiCom '02: Proceedings of the 8th annual international conference on Mobile computing and networking*. Atlanta, Georgia, USA: ACM, 2002, pp. 24–35.