Performance of mac80211 Rate Control Mechanisms

Wei Yin [1 2 3], Peizhao Hu [2], Jadwiga Indulska [1 2], and Konstanty Bialkowski [1]
[1] The University of Queensland, School of ITEE, Australia
[2] National ICT Australia (NICTA)
[3] National University of Defence Technology, China
{Firstname.Lastname}@nicta.com.au

ABSTRACT

The 802.11 wireless networking standard has several transmission rates which can be adaptively selected by the MAC layer to cater for various channel conditions. Such dynamic adaptations can improve per-hop performance in wireless networks by enhancing the data throughput, reducing the channel usage time and decreasing the number of failed transmissions. This in turn has an impact on the quality of service (QoS) provided for communicating applications. The new mac80211 framework, which is now part of the Linux operating system, defines functionalities and interfaces that can be used by wireless network drivers to access physical-layer information which has impact on high-level protocols/algorithms. In this paper we present a comprehensive evaluation of the per-hop performance of two rate control mechanisms used by the mac80211 framework in Linux: Minstrel and PID. The evaluation results show that Minstrel outperforms PID. We discuss the PID's performance problems and propose an enhancement to PID to address them. The evaluation of the enhancement confirms substantial improvement of PID's performance in a range of conducted and over-the-air communication scenarios.

Categories and Subject Descriptors

D.2.8 [**Software Engineering**]: Metrics—*performance measures*

General Terms

Performance

1. INTRODUCTION

Wireless channels are subject to various types of interference which introduces variability to the wireless channel quality — adaptive selection of transmission rates according to the dynamic channel quality can improve per-hop performance by utilising the variety of transmission rates available in the IEEE 802.11 standards. A number of algorithms have been proposed to adaptively select transmission rates according to various metrics. Based on the type of metrics used, rate control algorithms can be classified into three groups: ACK (Acknowledgement), SNR (Signal to Noise Ratio) and BER (Bit Error Rate) based mechanisms. ACK based mechanisms adapt rates according to the results of successful transmissions (e.g., packet loss ratio and average transmission time), which are estimated based on received acknowledgements. For example, RRAA [1], AMRR [2], SampleRate [3], Onoe [4], Minstrel [5] and PID [6] are ACK based. SNR based mechanisms measure the SNR to select transmission rates. These mechanisms include RBAR [7], CHARM [8] and FARA [9]. BER based mechanisms use a more fine-grained metric, which uses the average number of bit errors in each frame. An example of such a mechanism is SoftRate [10]. It is difficult to accurately measure the SNR metric and there are limitations in calculating the BER using conventional network cards therefore the ACK based mechanisms are commonly used in the current wireless networks.

In earlier Linux kernels, four ACK based mechanisms, including Onoe, AMRR, SampleRate and Minstrel, were implemented on the Madwifi driver. In [11], we reported on evaluation of these rate control mechanisms and showed that Minstrel has the best performance and it is also not worse than the SNR or BER based mechanisms. The recently defined mac80211 framework introduced functionalities and interfaces that can be used by drivers to access physical-layer information, which was previously provided by a proprietary implementation of HAL (Hardware Abstraction Layer) in Madwifi driver. As a result the functionality of rate control is now part of the framework rather than being dependent on individual drivers. The mac80211 framework is used by a number of new wireless network card drivers, including ath5k — a replacement of the MadWifi driver.

In this paper we present a comprehensive evaluation of the per-hop performance of two rate control mechanisms used by the mac80211 framework in Linux: Minstrel and PID, and propose an enhancement to PID that improves its performance significantly. In addition, as packet/frame loss ratio is used as a metric in many rate adaptation algorithms we analyse the pros and cons of using this metric. Although we have evaluated Minstrel in Madwifi previously, this paper presents the first comprehensive evaluation of Minstrel on the mac80211 framework.

The evaluation is carried out in a range of conducted and over-the-air experiment scenarios. We first measure the performance of these two rate control mechanisms in static channel conditions and follow it with a performance investigation for various dynamic channel conditions caused by

MSWiM'11, October 31–November 4, 2011, Miami, Florida, USA.

interference and mobility (repeating pattern, fading channel, or progressive increase/decrease of channel link quality). For the interference scenario, we analyse throughput while varying three parameters: the duration, the interval and the strength of the interference. For the fading channel, a well-known Rayleigh model is used to generate the channel conditions, whereas for the progressive increase/decrease of link quality we model the channel quality changes of two communication stations moving towards or away from each other. The responsiveness of rate control mechanisms to channel changes is also studied. Furthermore, there are over-the-air experiments that confirm our observations in the conducted testbed.

The remainder of the paper is organized as follows. In section 2 we briefly describe the two rate control mechanisms. The performance comparison of Minstrel and PID carried out in a conducted testbed for repeatable experiments is described in section 3, followed by the validation of the comparison results by over-the-air experiments in section 4. A detailed discussion of PID shortcomings, a description of a proposed PID enhancement and performance evaluation of the enhanced PID (PIDE) are presented in section 5. The lesson learned about using packet/frame loss ratio as a metric for rate control mechanisms is presented in section 6. The paper concludes in section 7.

2. RATE CONTROL ON MAC80211

In this section we briefly describe the rate control mechanisms used by the mac80211 framework.

2.1 Minstrel

The core of the Minstrel [5] rate control algorithm is a retry chain, which proposes four candidate rates (i.e., $r0$, $r1$, $r2$ and $r3$) to attempt in case re-transmission is necessary. Each of these candidate rates has a corresponding counter (i.e., $c0$, $c1$, $c2$, and $c3$), which specifies the maximum number of retry a candidate rate is allowed before giving up. For example, the algorithm will first attempt to use rate $r0$ for the transmission; if the transmission fails $c0$ times, then rate $r1$ is attempted for the re-transmission; in the case when rate $r1$ also fails $c1$ times, then rate $r2$ and $r3$ are attempted in the same manner until it reaches the base rate. Minstrel uses this retry chain to define the retry preferences (shown in Table 1).

Table 1: Retry preferences

Attempt	Lookaround rate		Normal
	$RR < BTR$	$RR > BTR$	rate
$r0$	BTR	RR	BTR
$r1$	RR	BTR	NBTR
$r2$	BPR	BPR	BPR
$r3$	BR	BR	BR

As an attempt to determine the optimal rate for a given channel condition, Minstrel dedicates 10% of its data traffic to probe the performance statistics of other rates by randomly selecting a rate (as *Lookaround rate*) that are not currently in use. For this 10% data traffic, the retry preferences are the best throughput rate (BTR), the random rate (RR), the best probability rate (BPR) and the base rate (BR) if the randomly selected rate (RR) is lower than the current best throughput rate (BTR); otherwise they are the random rate, the best throughput rate, the best probability rate, and the base rate. For other 90% of data traffic (considered as *Normal rate*), the retry preferences are the best throughput rate, next best throughput rate (NBTR), the best probability rate and the base rate.

To obtain the statistics for the retry chain, for every rate adaptation period of 100 ms Minstrel calculates the measured throughput and the probability of success in transmissions for each rate. The calculation of the probability of success is based on an Exponential Weighted Moving Average (EWMA), which controls the balance of influence of both the old and new packet delivery statistics, as shown in Eqn (1). The throughput of each rate is then computed based on the weighted probability of success P_{new} and the maximum number of packets that can be sent.

$$P_{new} = (1 - \alpha) * P_{this\_interval} + \alpha * P_{previous} \quad (1)$$

- P_{new} is the weighted probability of success for this interval, which will be used by the rate selection process.
- α is a smoothing factor (or the *scaling value*) in the EWMA mechanism.
- $P_{this\_interval}$ is the probability of success of this interval before the rate selection, and it is calculated as the ratio of the number of packets sent successfully to the number of packets sent.
- $P_{previous}$ is the weighted probability of success for the last interval used to select last transmission rate.

In Eqn (1), the smoothing factor α can be manually configured at Minstrel's instantiation time, and defaults to 75%. The α value controls the influence of current and historical measurements on the next rate selection. A value of 0% means using only the latest measurements and a value of 99% means using the old results, with a tiny influence from the new measurements.

2.2 PID

The PID [6] rate control algorithm conducts rate control based on a proportional integral derivative (PID) controller. In essence, the controller is a control loop feedback mechanism that tries to minimise the difference (i.e., *error* in control system's term) of the current and target frame loss ratio (i.e., $FLR_{current}$ and FLR_{target} respectively) as a result of switching to a new transmission rate. By default, the FLR_{target} is set to 14% for all rates.

To determine the appropriate transmission rate, the controller computes an adjustment value, adj, as follow.

$$\begin{aligned} adj = \gamma * (1 + sharpening) * (e_{current} - e_{last}) \\ + \alpha * e_{current} + \beta * e_{avg} \end{aligned} \quad (2)$$

where $e_{current}$ is the current error, and its value is calculated as $FLR_{target} - FLR_{current}$. e_{avg} is the average of recent errors, while e_{last} is the last error. In addition, there are four tuneable parameters. *sharpening* is a smoothing factor (non-zero when fast response is needed), whereas α, β and γ are the corresponding *proportional*, *integral* and *derivative* coefficients.

Using Eqn. 2 PID computes the adjustment value, adj, at the end of each rate adaptation period and decides on whether to switch to a new transmission rate. When adj is positive, the new rate R_{new} is set to the highest rate, in the range of $R_{current} \leq R_{new} \leq (R_{current} + adj)$, and its error (i.e., the difference between the target and respective frame loss ratio) is no more than the error of the current rate, $R_{current}$. When adj is negative, the new rate R_{new} is set to the lowest rate, in the range of $(R_{current} + adj) \leq R_{new} \leq R_{current}$, and its error is no more than the error of the current rate. No rate adaptation is required, if adj is equal to zero.

3. CONDUCTED EXPERIMENTS

To fairly compare the performance of Minstrel and PID, a fully controlled testbed is used and the results are also validated using over-the-air experiments (discuss in the next section). In this section, we describe the controllable evaluation platform and the results of performance evaluation carried out in this platform for five selected scenarios.

3.1 Controllable Platform

Fig. 1 shows the experiment setup. As shown 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). The wireless signals are transmitted along the co-axial cables [12], rather than being sent over the air by antennas. Therefore, the testbed can produce repeatable evaluation experiments, i.e., the rate control mechanisms are compared for the same parameters (e.g., path loss) and offered load. During the experiments, the attenuation value varies according to the specifications defined in each evaluation scenario.

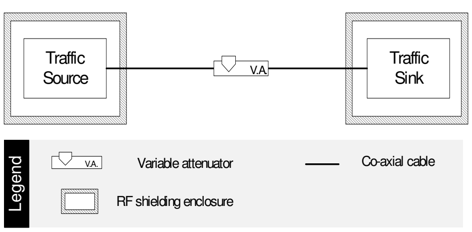


Figure 1: Experiment setup.

Each traffic source and traffic sink runs on a single board computer. Each computer is equipped with a Wistron CM9 Atheros wireless card. The computers run the Linux operating system (with kernel version 2.6.35 and the corresponding mac80211 framework). The debugfs system for Minstrel and PID are modified to allow access to necessary information. In addition to the programmable attenuator, a 25 dB fixed attenuator is attached to each traffic node, and each node's transmission power is set to 16 dBm. This prevents damage of the radios and ensures that within the operation range of the variable attenuator (0–63 dB), a full range of link qualities is possible (i.e., minimal attenuation corresponds to full throughput and maximum attenuation corresponds to a disconnected link).

To ensure consistent results, experiment nodes are enclosed within RF shielding boxes (JRE 4400), which ensures 85 dB of isolation to external sources. Further to this, all experiments are performed using IEEE 802.11a. This has two benefits, the first being that the 5 GHz frequency band is currently much less used than 2.4 GHz, and secondly it means that all transmission rates use the same family of modulation and coding methods. After configuration, each setup is verified manually using various tools.

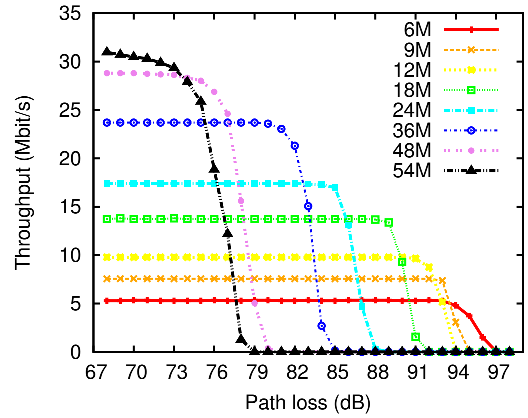


Figure 2: Fixed rates performance.

All transmitted data packets are captured at the receiver using a packet sniffing tool (tcpdump). Each measurement contains information regarding the achievable throughput and the number of successful packets at each rate. A parser library (Banjax) is used to analyse the measurement files. In the experiments, we use UDP traffic; the UDP packet size is set to the iperf default size. Experiments using TCP packets are not discussed in this paper, as congestion control in TCP makes it difficult to determine whether the poor performance is due to the congestion control mechanism or rate control at the MAC layer.

3.2 Evaluation Scenarios and Results

In this section we discuss the results of our extensive evaluation of the Minstrel and PID performance. The evaluation was carried out in five representative communication scenarios, such as static channel, dynamic channel, fading channel, progressive increase/decrease of channel quality and a sudden channel quality change.

3.2.1 Static channel conditions

Under static channel conditions, the rate control mechanisms should converge at an optimal rate and minimise hopping between different rates. To evaluate this scenario, we set a constant effective path loss between the two traffic nodes for a duration and observe the throughput and rate selection in relation to the given effective path loss. During the experiment the transmission power of the traffic source is set to 16 dBm, while iperf offered load is set to saturated.

To compare the rate control mechanisms to the optimum transmission rate, all fixed transmission rates are also evaluated in the same channel conditions as shown in Fig. 2. This experiment is performed to investigate the highest throughput rate (optimum rate) for each path loss, e.g., the optimum rate for 70 dB is 54 Mbit/s, because it achieves the highest throughput.

Fig. 3 shows the throughput achieved by the two rate control mechanisms as a function of the path loss. Higher path loss means lower received signal strength. In general, Minstrel shows better throughput compared with PID, except for very good or bad channel conditions. The number of successful packet arrivals at the receiver for each rate is plotted in Fig. 4. It shows that Minstrel almost selects the highest throughput rate and achieves throughput similar to the optimum rate for each scenario, except 79 dB. For

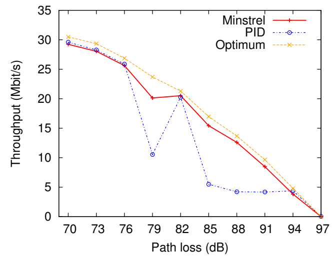
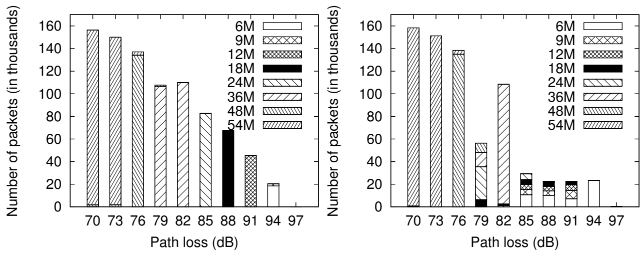


Figure 3: Auto rate performance.



(a) Minstrel (b) PID

Figure 4: # of packets succeeded at each rate.

this path loss, a small number of packets are transmitted at 48 Mbit/s, which may fail and increase the retransmission count.

As shown in Fig.3, PID performs well in extreme channel conditions in which either the highest rate of 54 Mbit/s or the base rate achieves the best throughput, which matches the optimum rate. However, in other channel conditions (e.g., 79, 85, 88 and 91 dB) PID performs poorly. This is because PID fails to select a rate that can achieve the highest throughput as shown in Fig. 4(b). To further investigate this problem, we show in Fig. 5 the rate selection history of PID for the static channel conditions scenario with path loss value at 85 dB. For clarity we show only the first 30 seconds of a 60 seconds experiment of the repeating pattern. From this repeating pattern, we can see that PID tends to select a higher rate than the channel is capable to support; this results in higher frame loss ratio (FLR) and PID then falls back to the base rate and gradually increases the rate again as shown. By analysing Figs. 2 and 5, we discover that this instability (repeating pattern in rate selection) is the result of the FLR of a selected rate falls below the target FLR defined in PID; thus PID tends to increase rate. The instability effect occurs whenever a rate reaches the path loss regions (e.g., between 82-84dB for 36 Mbit/s), as shown in Fig. 2, where the rate's throughput experiences sharp fall. In these path loss regions, it is likely that the FLR is above the defined target FLR in PID, which causes PID to drop rate. This is a problem in rate control mechanisms if they do not stop switching to a higher rate, even if the current channel conditions cannot support it.

3.2.2 *Impact of Interference*

In this section, we study the impact of interference on rate control mechanisms. In wireless networks, one of the significant effects of interference is the drop of channel quality (i.e., decrease in the signal to noise ratio). In our controlled platform, we emulate the presence of interference by increas-

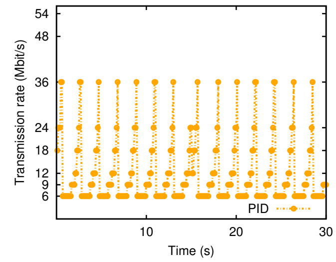


Figure 5: Repeating pattern in rate selection in PID.

ing in the effective path loss, which in effect also decreases the signal to noise ratio.

To this end, we define an evaluation scenario in which link quality changes periodically in a repeating pattern (i.e., to emulate the presence of interference), and we measure how well a rate control algorithm performs when the duration, interval, and strength of interference vary.

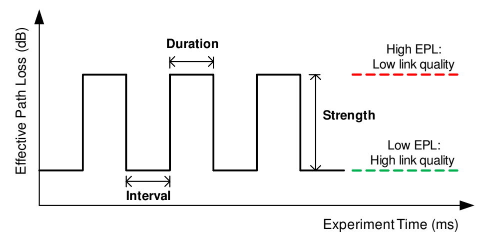


Figure 6: Emulation model for channel changes.

As shown in Fig. 6, the changes in link quality are emulated by adjusting the effective path loss (EPL); that is, low EPL means high/good link quality in which data transmission at 54 Mbit/s is possible, while high EPL means low/bad link quality allowing data transmission at rates lower than 54 Mbit/s. By varying the interval, duration and strength of the effective path loss, we can emulate various changes of link quality.

Fig. 7 shows the results when varying different interference parameters. In the figures, we also include the optimum rate at each interference configurations that is calculated using Fig. 2. Details of each experiment are discussed below.

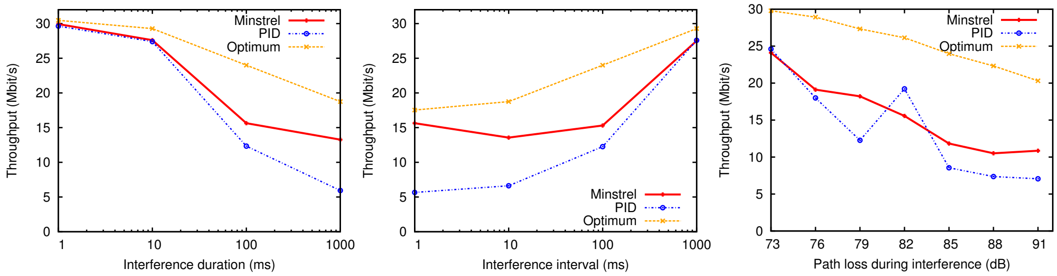
Duration

In this experiment, we vary the duration exponentially from 1 to 1000 ms. The path loss during interference is set to 85 dB (for which the highest throughput rate is 24 Mbit/s, as shown in Fig. 2). Between the two interference there is a clear channel in which a 54 Mbit/s traffic is supported. The interval between interference is set to 100 ms.

As shown in Fig. 7(a), the throughput drops when the interference duration increases; that is, when interference occurs only lower rates are supported, therefore the longer the duration the lower the throughput can be achieved. One observation is that Minstrel always outperforms PID and achieves almost the maximum throughput when duration is less than 100 ms. We argue the poor performance of PID is due to the instability of rate selection.

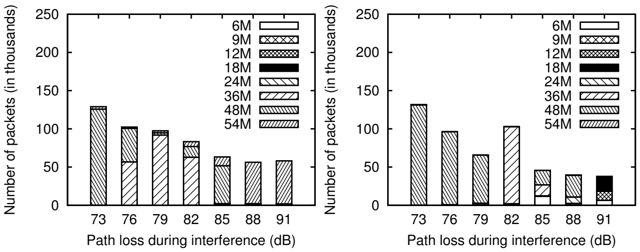
Interval

This is similar to the previous experiment, but we vary the interval between interference from 1 to 1000 ms and fix the



(a) Duration (b) Interval (c) Strength

Figure 7: Auto rates performance for various interference parameters.



(a) Minstrel (b) PID

Figure 8: # of packets succeeded at each rate.

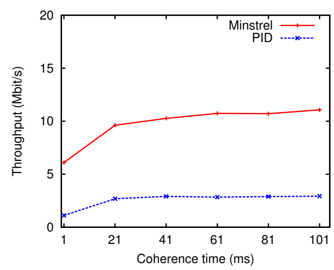


Figure 9: Performance in Rayleigh model.

interference duration to 100 ms. The path loss during interference is set to 85 dB, while there is clear channel between interference. Fig. 7(b) shows the results for this experiment. The throughput increases when we increase the interval between interference. In this experiment, Minstrel again constantly outperforms PID. Similar to the previous scenario, instability of rate selection in PID could be the cause.

Strength

In this experiment we vary the path loss from 73 to 91 dB to emulate different strength of interference. A clear channel (capable of supporting 54 Mbit/s) is used in between the interference. The channel condition varies between two path losses (very good channel and lower quality channel) and each lasts for 15 ms, for every 15 ms (i.e., the typical minimum coherent time in residential urban area with vehicles passing by [13]).

From Fig. 7(c) we can see that for Minstrel the throughput decreases as the emulated interference becomes stronger. In general PID follows similar behaviour, except for one occasion when path loss is at 82 dB (correlate to Fig. 3). Fig. 8(b) shows that PID selects 36 Mbit/s for all its transmissions at path loss of 82 dB. The reason being Minstrel and PID adapt rates based on the performance of each rate over the previous rate adaptation periods. Within each rate adaptation period, half of the time when there is interference, using optimal rate will gain full throughput, while using any higher rate will only achieve half of the throughput. In this case, rate selection is depending on whether full throughput of the optimal rate is better than half of the throughput of higher rates. From Fig. 2 we see that at 82 dB, 36 Mbit/s with full throughput is clearly better.

3.2.3 *Fading Channel*

Rayleigh fading [14] is one of best-known models which capture the effect of signal propagation in wireless environments. It represents the statistical time-varying nature of the received signal envelope of a flat fading signal when there are objects in the environment that scatter the signal before it reaches the receiver. The probability density function of the received signal power is described:

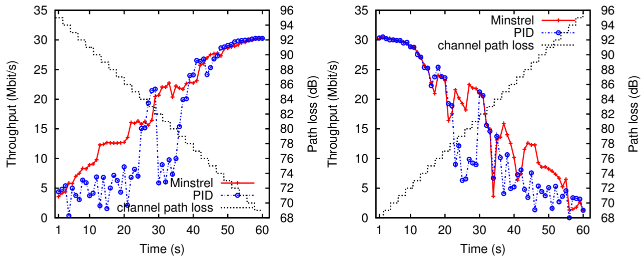
$$p(\gamma) = \frac{\gamma}{\sigma} e^{-\frac{\gamma^2}{2\sigma^2}} (0 \leq \gamma \leq \infty) \qquad (3)$$

where $\frac{\gamma^2}{2}$ is the instantaneous power and σ^2 is the average power of the received signal. The Rayleigh distribution is controlled by σ, the root mean square value of the received signal. We can generate various sets of Rayleigh distribution random numbers with different σ by using the inverse transform function with the uniform distribution.

By using this Rayleigh fading model, we investigate how a rate control algorithm behaves under channel fading in a fully controlled environment. Using Eqn. 3, we generate 60000 samples of attenuation value which are to be played-back on our programmable attenuator. The average path loss of the model is set to 85 dB. To study how the two mechanisms cope under different mobility pattern, we vary the coherence time, which refers to the time period during which the channel remains constant. In [13], Camp and Knightly discuss that typical coherence time in residential urban area caused by passing vehicles is roughly between 15 and 100 ms. As our attenuators can get up to 1 ms granularity we perform experiments with coherence time varying from 1 to 101 ms to include cases of faster fading patterns. Fig. 9 shows the throughput achieved by both mechanisms in this scenario. Minstrel performs much better than PID.

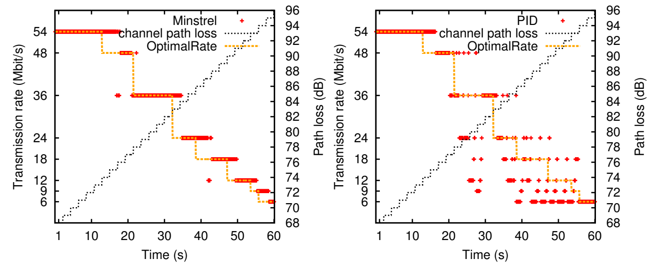
3.2.4 *Channel quality progressive increase/decrease*

Node mobility is common in wireless networks. As a wireless node moves toward/away from another node, there are changes in link quality. To compare how the rate control



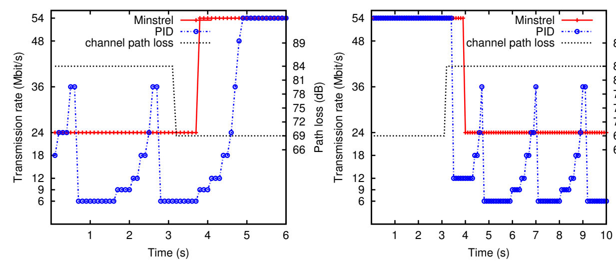
(a) Quality linearly increase (b) Quality linearly decrease

Figure 10: Throughput when channel quality changes linearly.



(a) Minstrel rate selection (b) PID rate selection

Figure 11: History of rate selection when channel quality linearly decrease.



(a) Decrease attenuation (b) Increase attenuation

Figure 12: Rate selection when channel quality sudden changes.

mechanisms perform under these scenarios, we designed a scenario in which channel quality progressively increase or decrease.

In our conducted platform, we emulate these two scenarios by gradually decreasing/increasing the attenuation (corresponding to the increase/decrease of channel quality) between two nodes. In the experiment we vary the attenuation between 68 and 95 dB over a period of 60 s. During the experiment, we record, in every second, the achieved throughput and the history of rate selection.

Fig. 10 shows the results of these two experiments. There are several important observations: (i) in both cases Minstrel outperforms PID whenever rate adaptation is possible and necessary; the instability in rate selection of PID being the reason for such poor performance; (ii) Minstrel performs relatively well in the case when channel quality is gradually increased, but it experiences problems when channel quality falls. There are seven signs of sharp throughput drops, as shown in Fig. 10(b). After a detailed analysis, we discover that the path loss values of these throughput drops correlate exactly with the path loss regions where the throughput of a fixed rate drops rapidly, as shown in Fig. 2. We call these regions the *rate transition regions*. As an example the drop between the time 32 to 34 s, the path loss value increases from 82 to 84 dB. At these path loss, there should be a transition in rate selection, from 36 to 24 Mbit/s. However, for Minstrel a great portion of the outgoing packets are still sent at 36 Mbit/s, rather than using the optimal rate, as shown in the history of rate selection when channel quality decreases (in Fig. 11(a)); some of them will fail and retry using lower rates, hence results in poor throughput. One reason for why Minstrel still selects 36 Mbit/s after the channel quality changed is due to the way Minstrel calculates its metric. As discussed in Section 2, the smoothing factor α (in Eqn. 1) defaults to 75% to increase the weight on history measurements in its calculation. Before the channel changes, the rate of 36 Mbit/s achieves highest throughput. It will take time for the 36 Mbit/s measurement to have no impact on the rate selection. A comprehensive study on the impact of α value on the Minstrel performance (on the Madwifi driver) is presented in [15]. Another reason is that Minstrel wrongly assumes that a lost frame takes the same time as a successfully delivered frame to transmit, thus it overestimates the link capacity. Due to the combination of these reasons, Minstrel uses 36 Mbit/s even within the rate transition regions. The other throughput drops can be explained in the same manner.

3.2.5 *Responsiveness due to sudden channel quality changes*

Responsiveness is one of key aspects of a rate control mechanism; that is, how fast a rate control mechanism can select the optimal rate for the given channel conditions. To create such a scenario, we start the experiment with one path loss value; after a period of time, we either increase or decrease the path loss value to emulate sudden channel quality changes. From the point when attenuation is changed, we measure the time it takes for rate control mechanisms to converge on an appropriate rate for the new channel conditions.

Fig. 12 shows traces for these experiments. In the case when the channel quality suddenly increases (i.e., decrease in path loss) as shown in Fig. 12(a), we notice that Minstrel will take approximately 500 ms to stabilise at 54 Mbit/s for the new channel conditions. For PID, we again observe the repeating pattern as before. Obviously PID does not converge at a rate. While it may look that PID converges at 54 Mbit/s at around 5 s, 54 Mbit/s is in fact already the highest rate supported by the IEEE 802.11a standard; otherwise, the repeating pattern will continue. This is a fundamental issue of the PID rate selection mechanism — it fails to verify whether the higher rate should be used. As expected, we see a similar repeating pattern for PID in the case when channel quality decreases, as shown in Fig. 12(b). On the other hand, Minstrel takes around 600 ms to converge at 24 Mbit/s after the channel quality changes.

4. OVER-THE-AIR EXPERIMENTS

In the last section, we discussed the evaluation of both rate control mechanisms in a conducted platform in which all RF signals are sent over coaxial cables rather than over-

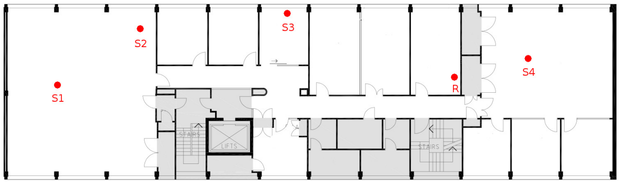


Figure 13: Experimental floor plan.

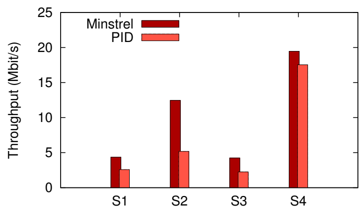


Figure 14: Performance of semi-controlled experiments.

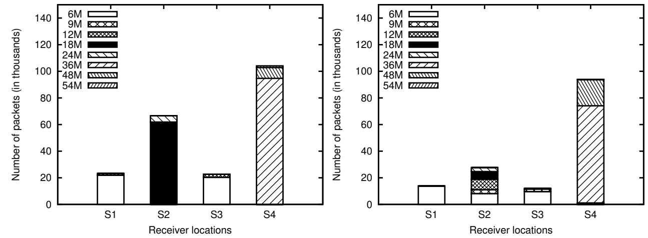
the-air. The advantage is that we have a full control of the experiments — the experiments are repeatable; hence we can analyse the cause of a particular observation in the measurements. To validate these conducted evaluation results, we put together an over-the-air setup (i.e., RF signals are sent via antenna) and present our findings from two experiment configurations.

Fig. 13 shows the random placement of four senders (i.e., S1, S2, S3, and S4), which send saturated traffic (one at a time) to the receiver (i.e., R) in our office environment. With this five nodes setup, we perform two experiments: *Semi-controlled* configurations in which we select a wireless channel and experiment time that have minimum level of external interference; and *In-the-wild* configurations in which we perform our experiments using a wireless channel that is shared with wireless access points around the office and at random office hours. The latter experiments will potentially test a rate control mechanism's ability to adapt to uncontrolled (but realistic) channel quality changes.

4.1 Semi-controlled Configurations

The aim of this over-the-air experiment is to recreate the static channel scenario we have in the conducted experiment setup. We want to validate our findings in a more realistic environment; that is, RF signals are sent via the antenna and from senders at different random locations (hence different propagation delays and different path loss to the receiver). To ease our analysis, we first run our experiments at midnight over the weekends and use a wireless channel that is not shared with other wireless access points in the vicinity. Each experiment runs for one hour, and the measurements are averaged.

Fig. 14 presents the throughput results for both mechanisms measured from different locations (refer to Fig. 13) and shows that Minstrel clearly outperforms PID at all locations. Since these experiments are performed at off-peak time and using channel that is not shared, the channel conditions are relatively stable. We investigate how both rate control mechanisms determine what rate to use under the given channel conditions. Fig. 15 shows the number of packets sent and succeeded at each rate for these experiments. We can see Minstrel is very stable in terms of picking on the



(a) Minstrel (b) PID

Figure 15: # of packets succeeded at each rate.

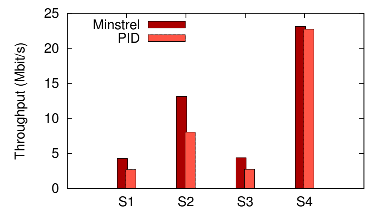


Figure 16: Performance of *in-the-wild* experiments.

optimum rate for the given channel conditions. In contrast, these experiments confirm the instability of rate selection in PID.

4.2 In-the-wild Configurations

Having the experiment results in the semi-controlled configurations in which we compared the two mechanisms, our next goal is to investigate how the mechanisms perform in a completely uncontrolled settings; i.e., the *In-the-wild* experiments. In these experiments, we perform random tests during random office hour; therefore there will be people moving around in the office (or even using the microwave at a location between these nodes). In addition, we use a wireless channel that is shared with four university's wireless access points around the office; in terms of traffic pattern, they are completely out of our control. We run each experiment for one hour and average the measurements.

The results of these experiments are presented in Fig. 16. Although difference in the performance is not very significant at some locations, overall we still see that Minstrel outperforms PID. Due to the complexity involved in modelling the exact channel changes, we cannot provide the most definitive reason for the performance results. However, we strongly believe that the instability in rate selection will be one of the significant drawbacks, which causes poor performance for PID.

5. WAYS TO IMPROVE PID

Based on the previous experiments we can conclude that Minstrel outperforms PID. In this section, we discuss the potential reasons for the PID's poor performance, present a proposal for improving PID, and show that this PID enhancement significantly improves its throughput.

5.1 Issues in PID

As we have emphasised a number of times, the most significant problem with PID is its instability in rate selection, as first shown in Fig. 5 and demonstrated in subsequent experiments.

As discussed in Section 2, the idea of PID is based on control system's principle of a feedback loop. That is, defining a target frame loss ratio (FLR) — e.g., the target FLR in PID is fixed at 14% — let the rate control mechanism to adapt its rate to meet that target by minimising the difference between the current FLR and the target FLR. It is very straight forward; such that PID drops the sending rate when FLR is greater than 14%, while increases the sending rate when FLR is less than the target FLR.

During the adaptation process PID will increase the rate whenever the current FLR is below the target threshold, regardless the current channel conditions which may not be sufficient to support the higher rate. The consequence of this is the FLR of the higher rate is higher than the target threshold and this causes PID to drop the rate again. Hence, this results in oscillation in rate selection as we can see in Fig. 5.

One solution to this problem is to implement a verification mechanism that probes the achieved throughput of the proposed rate compared to the current throughput. The goal is to maximise throughput therefore if the proposed rate achieves higher throughput than the current sending rate the algorithm should select the proposed rate. Otherwise, the rate adaptation requests should be ignored. The same rule applies for requests either to increase rate or to decrease rate.

5.2 The Making of PIDE

As an enhancement of PID (PIDE[1]) we implemented the proposed verification mechanism. The mechanism uses information passed by the PID to detect requests on rate adaptation at the end of each rate adaptation period. When a new rate is proposed (at the end of the last adaptation period) and the proposed rate has not been verified, PIDE then sends n frames (e.g., current implementation uses three frames) using the proposed rate in the current rate adaptation period. Other frames within the adaptation period will continue to use the current rate. Each frame is associated with a status that recordes a number of statistics regarding the transmission (e.g., retry count). PIDE collects these statistics and maintains a table of performance information for each rate. At the end of the current adaptation period, PIDE compares the achieved throughput to decide whether to use the proposed rate in the next rate adaptation period. The achieved throughput tp is calculated as

$$tp = (1 - FLR) * (1s\ /T) \tag{4}$$

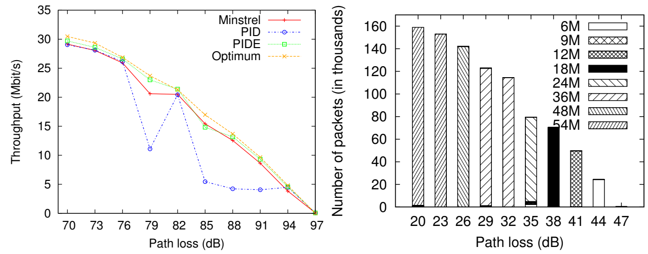
where FLR is the frame loss ratio, $1s$ is one second, and T is calculated as $DIFS + T_{DATA} + SIFS + T_{ACK}$ [16]; and T_{DATA} and T_{ACK} are the respective transmission times of the DATA and ACK frames.

In addition to the verification mechanism, we also modify the way frame loss ratio is calculated in PID. When calculating the FLR, PID counts multiple failures of frame transmission attempts as one failed frame; this potentially underestimates the frame loss ratio.

5.3 Evaluation of PIDE

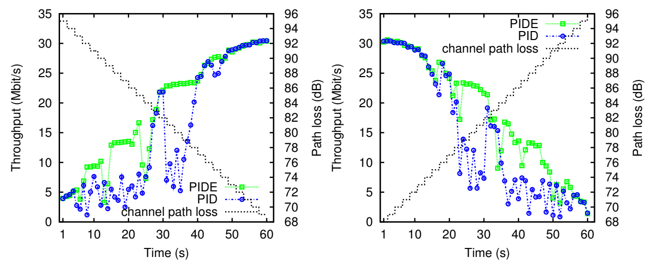
To show the effectiveness of our proposed enhancement, we revisit the same set of conducted and over-the-air experiments, as discussed in Sections 3 and 4. Due to the space limitation, we only discuss some of them in the paper.

[1]patch available at `http://itee.uq.edu.au/~uqwyin/publications/pid-3.3.0.patch`



(a) Performance under different path loss (b) # of packets succeeded at each rate for PIDE

Figure 17: Performance in static channel.



(a) Quality linearly increase (b) Quality linearly decrease

Figure 18: Performance when channel quality linearly changes

5.3.1 *Static channel conditions*

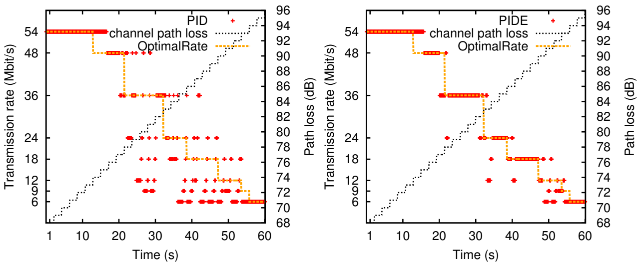
Fig. 17(a) shows that PIDE achieves much better performance than PID when the channel conditions are relatively static; on some occasions, PIDE even outperforms Minstrel and achieves throughput that is very close to the optimum. When comparing the history of rate selection as in Figs. 4(b) and 17(b), it is very obvious that PIDE solves the instability problem in PID. Rather than hoping around between different rates, as in Fig. 4(b), PIDE is able to quickly converge at rates that can achieve high throughput.

5.3.2 *Channel quality progressive increase/decrease*

Fig. 18 shows the achieved throughput for PID and PIDE as the link quality linearly increases (i.e., decrease in attenuation)/decreases (i.e., increase in attenuation). Although PIDE does not solve the significant throughput drops in both cases, it is still very obvious that PIDE by far performs better than PID in these scenarios. The reason being PIDE selects the optimal rate quicker and stays at the optimal rate most of the time, as shown in the history of rate selection when channel quality decreases (in Fig. 19(b)); in contrast PID has fewer time slots in which it uses the optimal rate, as shown in Fig. 19(a). Another observation is that in terms of the optimal rate usage PIDE and Minstrel are very similar, as shown in Figs. 11(a) and 19(b). Fundamentally, this is because the verification mechanism in PIDE solves the problem of rate hoping in PID.

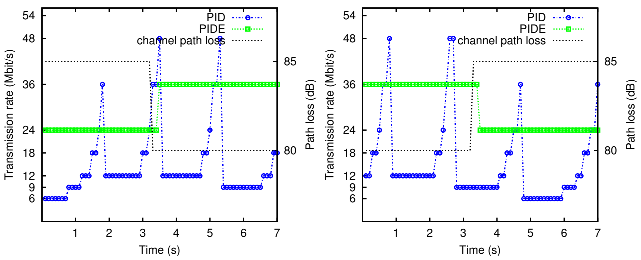
5.3.3 *Responsiveness to sudden channel quality changes*

In the experiments to test the responsiveness of PIDE due to sudden channel quality changes, PIDE shows significant improvement over PID, as it converges at an optimal rate in the matter of around 200 ms in both cases when changes of channel conditions happen. Another observation is that in



(a) Rate selection for PID. (b) Rate selection for PIDE.

Figure 19: History of rate selection channel quality decreases.



(a) Rate selection of PIDE. (b) Rate selection of PID.

Figure 20: Rate selection when channel quality suddenly changes.

these experiment scenarios PIDE shows better responsiveness than Minstrel, which is shown when comparing Figs. 12 and 20.

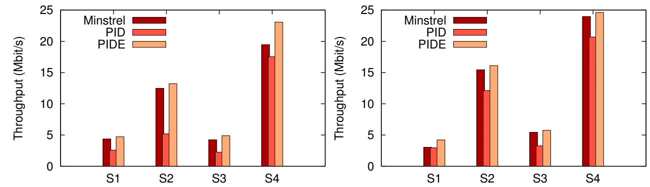
5.3.4 *Over-the-air experiments*

To validate results of our conducted experiments, we also revisit the two over-the-air experiment configurations. Fig. 21 shows the performance results in both configurations. As shown in the figures, the results confirm our findings from the conducted experiments, showing that PIDE performs much better than PID and outperforms Minstrel. The absolute throughput difference for the same pair of nodes can be explained by the frequency-selective fading, as described in [9] and [17]. Figs. 15(a) (refers to Minstrel) and 22 (refers to PID and PIDE) show the number of packets sent and succeeded at each rate (for the semi-controlled experiments), which again confirm that PIDE with the verification mechanism is much better in selecting the optimal rate and staying at this rate. Similar pattern in rate selection is also observed for the *in-the-wild* experiments.

6. DISCUSSION

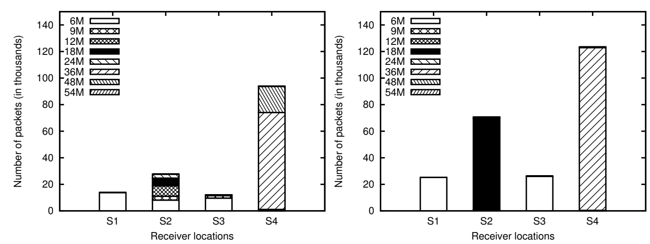
Based on the experience we gained from this exercise of evaluating and improving the PID rate control mechanism, we can summarise three lessons learned in terms of using frame loss ratio (FLR), or packet loss ratio (PLR), as a metric in designing rate control mechanisms.

First, usually when the FLR of a rate is below a threshold, the mechanism should increase the rate. But one condition is that the proposed rate should be verified to check whether the new rate provides better throughput (or is still below the target FLR, depending on the goal of the network) before switching rate; at a minimum whether the current channel



(a) Semi-controlled (b) In-the-wild

Figure 21: Over-the-air performance.



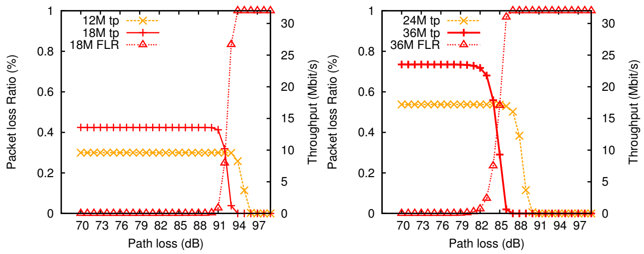
(a) PID (b) PIDE

Figure 22: # of packets succeeded at each rate.

conditions can support such a rate increase. This will prevent oscillation in rate selections, as we have seen in PID.

Second, rate control mechanisms usually decrease the rate when the FLR of a rate is above the target threshold (e.g., in PID this target FLR is set to 14%). This is based on the assumption that a rate with some degree of FLR achieves lower throughput than its adjacent lower rate. But this might not be the case in some scenarios. Fig. 23 shows two examples of the relation between throughput and FLR of two adjacent fixed rates at different path loss values; that is, 12 and 18 Mbit/s in Fig. 23(a), and 24 and 36 Mbit/s in Fig. 23(b). These figures show two evidences of how PID using 14% as the target FLR for all rates failed under the above assumption. At the path loss of 92 dB in Fig. 23(a) and 84 dB in Fig. 23(b), we can see that the FLRs at these path loss values are both above 20% for the rate of 18 and 36 Mbit/s, respectively. Therefore, PID will drop the rate under the above assumption. However, if we look closely at the throughput that each respective fixed rate is achieving, we will notice that even with FLR above 20% they still outperform their adjacent lower rates. Hence, we conclude that the above assumption does not hold true in some cases.

Third, assuming a unique FLR threshold for all rates is dangerous. If the goal is to achieve maximum throughput, from Fig. 2 we should see that the absolute maximum throughput for a given path loss is on the envelop that connects the maximum throughput points of all the fixed rates. Hence the optimal maximum FLR of each fixed rate are the points when a higher rate switches to a lower rate to maintain highest throughput. When we compared the FLR of these throughput points in Fig. 2, we found that they are all different values. Therefore, PID using a fixed target FLR of 14% for all rates fails to achieve maximum throughput. In literature, authors of RRAA [1] also notice that there is no single optimal FLR threshold for all the rates. In addition, in [11] we have shown that other similar mechanisms (e.g., AMRR [2], Onoe [4]) perform poorly when compared to Minstrel, which uses the achieved throughput as the metric for rate adaptation.



(a) Evidence 1 (b) Evidence 2

Figure 23: Relation between throughput and FLR.

In summary, we are questioning whether FLR (or PLR) is a good metric to be used in rate control mechanisms. It can be seen from our analysis that directly measuring the achieved throughput and using it as the metric for rate adaptation leads to higher throughput.

7. CONCLUSION

In this paper we presented the performance evaluation of two MAC layer rate control mechanisms implemented on the mac80211 framework: Minstrel and PID. These rate control mechanisms are used in millions of computers running the Linux operating system. We showed that PID has poor performance compared to Minstrel; we then proposed and evaluated a PID enhancement (PIDE) that improves the performance. In addition, we discussed a lesson learned in this research, i.e., frame loss ratio is not the best metric for rate adaptation.

We compared the rate control mechanisms in a conducted testbed that can provide the same experiment environment (i.e., the same effective path loss) for each experiment. We evaluated the mechanisms in five communication scenarios: (i) static channel conditions, (ii) interference with repeating pattern, (iii) Rayleigh model fading channel, (iv) progressive increase/decrease of channel quality, and (v) sudden change of channel quality. The results from the conducted testbed were also validated by over-the-air experiments.

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.

8. REFERENCES

[1] S. H. Y. Wong, H. Yang, S. Lu, and V. Bharghavan, "Robust rate adaptation for 802.11 wireless networks," in *Proceedings of MobiCom 2006*. Los Angeles, California, USA: ACM, 2006, pp. 146–157.

[2] M. Lacage, M. H. Manshaei, and T. Turletti, "Ieee 802.11 rate adaptation: a practical approach," in *Proceedings of MSWiM 2004*. Venice, Italy: ACM, 2004, pp. 126–134.

[3] J. C. Bicket, "Bit-rate selection in wireless networks," MIT Master Thesis, 2005.

[4] "Onoe specification," Madwifi Project.

[5] "Minstrel specification," Madwifi Project.

[6] "Pid specification," LinuxWireless Project.

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

[8] G. Judd, X. Wang, and P. Steenkiste, "Low-overhead channel-aware rate adaptation," in *Proceedings of MobiCom '07*. Montreal, Quebec, Canada: ACM, 2007, pp. 354–357.

[9] H. Rahul, F. Edalat, D. Katabi, and C. G. Sodini, "Frequency-aware rate adaptation and mac protocols," in *Proceedings of MobiCom 2009*. Beijing, China: ACM, 2009, pp. 193–204.

[10] M. Vutukuru, H. Balakrishnan, and K. Jamieson, "Cross-layer wireless bit rate adaptation," in *Proceedings of ACM SIGCOMM 2009*. Spain, Barcelona: ACM, 2009, pp. 3–14.

[11] W. Yin, K. Bialkowski, J. Indulska, and P. Hu, "Evaluation of madwifi mac layer rate control mechanisms," in *Preceedings of IWQoS 2010*, Beijing, China, June 2010.

[12] K. Bialkowski and M. Portmann, "Design of test-bed for wireless mesh networks," in *IEEE Antennas and Propagation International Symposium*, Toronto, Canada, July 2010.

[13] J. Camp and E. Knightly, "Modulation rate adaptation in urban and vehicular environments: cross-layer implementation and experimental evaluation," *IEEE/ACM Trans. Netw.*, vol. 18, pp. 1949–1962, December 2010.

[14] V. K. Garg, *Wireless Communications and Networking*. ELSEVIER, June 2007.

[15] W. Yin, P. Hu, J. Indulska, and K. Bialkowski, "A method to improve adaptability of the minstrel mac rate control algorithm," in *Proceedings of UIC 2010*, Xi'an, China, October 2010.

[16] "Ieee standard 802.11-2007," IEEE Computer Society, p. 625, 2007.

[17] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Predictable 802.11 packet delivery from wireless channel measurements," in *Proceedings of SIGCOMM 2010*, New Delhi, India, 2010, pp. 159–170.