This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2008 proceedings.

A Practical SNR-Guided Rate Adaptation

Jiansong Zhang Kun Tan Jun Zhao Haitao Wu Yongguang Zhang
Microsoft Research Asia, Beijing, China
{jiazhang, kuntan, junzhao, hwu, ygz}@microsoft.com

***Abstract*—Rate adaptation is critical to the system performance of wireless networks. Typically, rate adaptation is considered as a MAC layer mechanism in IEEE 802.11. Most previous work relies only on frame losses to infer channel quality, but performs poorly if frame losses are mainly caused by interference. Recently SNR-based rate adaptation schemes have been proposed, but most of them have not been studied in a real environment. In this paper, we first conduct a systematic measurement-based study to confirm that in general SNR is a good prediction tool for channel quality, and identify two key challenges for this to be used in practice: (1) The SNR measures in hardware are often uncalibrated, and thus the SNR thresholds are hardware dependent. (2) The direct prediction from SNR to frame delivery ratio (FDR) is often over optimistic under interference conditions. Based on these observations, we present a novel practical SNR-Guided Rate Adaptation (SGRA) scheme. We implement and evaluate SGRA in a real test-bed and compare it with other three algorithms: ARF, RRAA and HRC. Our results show that SGRA outperforms the other three algorithms in all cases we have tested.**

Keywords- Rate adaptation, 802.11, SNR

I. INTRODUCTION

Rate adaptation provides a critical mechanism for wireless systems to trade between physical layer data rate and robustness to maximize the performance. Traditionally, rate adaptation is considered as a MAC layer mechanism and many algorithms have been studied, most of which exploit only the MAC layer information, i.e. making rate selection decision based on the frame losses. One important assumption of loss-based approaches is that if the frame loss rate increases, it means the channel quality is deteriorated, and it should reduce the physical data rate by using a more robust modulation or coding scheme. However, this assumption does not hold if the frame losses are due to the activities of interfering senders instead of channel degradation. Thereby, adopting a low rate may not mitigate the frame losses. In contrast, it may cause an even higher loss rate as the lower rate will prolong the transmission time of a frame.

In fact, it is very difficult (if not impossible) to distinguish losses due to collision from losses due to channel error. One approach proposed by CARA [5] and RRAA [1] is to use RTS/CTS handshake. However, RTS/CTS cause significant overhead for high-speed 802.11 networks (37% and 29% for 11b and 11g, respectively [8]). Even though, RTS/CTS cannot effectively detect interference losses in practice, as the interference range is generally larger than the reliable transmission range of a frame. Lastly, RTS/CTS are specific to 802.11 MAC. Therefore, it cannot handle cases if the interference comes from a non-802.11 source.

It is expected that exploiting PHY layer information that directly characterizes the channel quality would gives a better guideline for rate adaptation. In literature, many schemes have proposed to use Signal-to-Noise Ratio (SNR) to assist the rate adaptation, e.g. RBAR [2], OAR [4], and RAF [7]. However, these designs are seldom studied in real environments, and little has been discussed on how SNR measures can be mapped an optimal rate in practice.

To understand the practical challenges of using SNR information, we first conduct a systematic measurement-based study on state-of-the-art commercial-off-the-shelf IEEE 802.11 hardware in several indoor/outdoor environments. Our results confirm that in general SNR is a good prediction tool for channel quality. We also identify several practical challenges. Our major findings are:

1. There is a 5~7dB *transition band* in SNR where the frame delivery ratio (FDR) goes from zero to one. Therefore, the traditional assumption that FDR rapidly changes to one right after the SNR threshold does not accurately capture this transition band in practice.
2. The SNR thresholds are hardware dependent. This may be mainly due to the fact that most hardware is not calibrated very well. Therefore, the previous assumption that one can obtain these values from theory calculations or single calibration is unrealistic.
3. External interference would distort the SNR-FDR relationship by introducing additional frame losses even if the SNR measures are high. When interference exists, directly predict FDR from SNR measure is over-optimistic.

To address the aforementioned challenges, we design a novel practical SNR-Guided Rate Adaptation (SGRA) for IEEE 802.11 networks. Unlike previous works, SGRA develops a novel automatic on-line calibration technique that rapidly and reliably builds the SNR-FDR relationship on per node-pair basis through real time measurements. SGRA classifies the channel into two states: *interfered state* and *interference-free state*. It detects interference and adopts two different adaptation strategies in different channel states. If the channel is *interference-free*, SGRA aggressively exploits the SNR value to predict the optimal rate; while in the *interfered state*, SGRA uses SNR information as a guideline to select a set of candidate rates, but relies on probing to obtain the optimal selection.

We use commodity (Atheros) chipset to prototype our design. We evaluate SGRA's performance using thorough experiments. Our results show that it consistently outperforms three state-of-the-art algorithms of ARF [3], RRAA [1] and HRC [6], in all scenarios with 802.11 a/g channels. The throughput improvement of SGRA over these algorithms can be 30% in interference free case, 120% with 802.11 hidden terminals and up to 900% with non-802.11 interference sources.

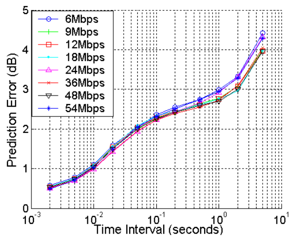


Figure 1. Average Prediction Error of SNR measurement in different time scales with different transmission rates. The walking speed is 1m/s.

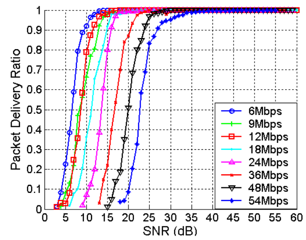


Figure 2. Plot of FDR as a function of SNR with different transmission rate. The size of UDP packet is 50bytes.

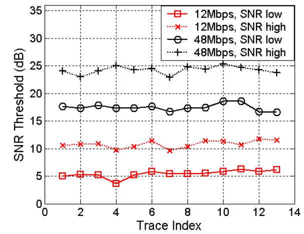


Figure 3. The SNR threshold for two transmission rates for all traces in the indoor office environment.

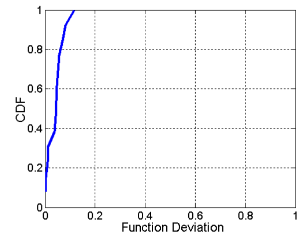


Figure 4. CDF of fuction deivation of SNR-FDR relation between each trace and the avereaged function from all indoor traces.

II. CHARACTERIZE SNR FOR CHANNEL ESTIMATION

A. Experimental setup and methodology

For most of our experiments, we use Netgear WAG511 802.11a/b/g cards with Atheros 5212 chipset. We also use some Linksys Wireless A+G PCBus cards and Cisco Aironet 802.11a/b/g Wireless PCI adapters for comparison. We customize a driver that disables the automatic rate-adaptation and provide an API for user to specify a fixed transmission rate. We conduct our experiments in four typical different environments, including indoor office, indoor classroom, outdoor business center and outdoor open field. Our measurements are conducted using 802.11a radio on channel 161. We confirm there is little interference in this channel except for ones that we explicitly set up.

In most of our experiments, we let the sender broadcast back-to-back UDP packets. Each packet is marked with a unique sequence number. The sender transmits packets at different rates (6~54Mbps) each for two minutes, and receivers (laptops) are moved along some pre-defined paths with different speeds. The device moving paths are selected to satisfy that FDR drops from 100% to 0%. All received packets are recorded with received signal strength and noise level. We use the term *trace* to refer to one dataset in our experiment that contains all recorded packets with all eight rates for a specific sender/receiver pair along a path.

B. Stability of measured SNR

It is critical to confirm if the transmitter's signal can be measured reliably. More specifically, our goal is to verify whether currently measured SNR is feasible to predict the upcoming SNR when the devices is moving, and if so what would be the time scale to make the SNR reasonable for further rate adaptation. We define prediction error as the difference between the average SNR measured in a time window and the SNR measured right after this time window. Figure 1 shows the average prediction error of a trace recorded in the indoor office environment. We observe that the predication error becomes larger with longer time interval. This observation is reasonable since during such long time interval, the distance between the sender and the receiver has significantly changed and thus signal strength is also changed. However, during a short-time scale (10~20ms), the SNR prediction error is only around 1dB. Actually, this time scale accords with the channel coherent time of human walking speed. Thus we conclude that it is feasible to predict SNR and 10~20ms is the reasonable time scale.

C. Estimating Channel quality with SNR

Now, we study how well SNR measure can be used to estimate the channel quality. In Figure 2, we show the plot of FDR in different transmission rates versus SNR. It is clear that different transmission rates have different SNR thresholds. Generally, higher rate requires higher SNR to sustain. We further observe that FDR drops greatly when SNR decays to some value. Our measurement results show a strong correlation between SNR and FDR. There is a "cut-off" SNR threshold for each rate, below which, frames are hardly received; while above that for a few dB, frames can be received with high probability. There is a "transition band" in SNR that FDR goes from zero to 100%. This transition band can be 5~7 dB wide. In this work, we use two thresholds to explicitly capture the existence of this transition band. We define two thresholds SNR_{low} and SNR_{high} as the SNR values that are corresponding to FDR of 10% and 90%, respectively.

We further evaluate the impact of locations and environments on SNR-FDR relationship. In Figure 3, we plot SNR_{low} and SNR_{high} observed for two selected rates for different traces. We see these thresholds are consistent in all traces within 1~2dB, which suggests that the SNR thresholds remains stable regardless of locations of the devices involved. Furthermore, for environment impact, we express FDR as a function of SNR for each trace, and obtain a "standard" function by averaging all indoor traces. Then, we calculate the Mean Square Deviation (MSD) between any trace we obtained in indoor/outdoor environments. MSD is a common parameter to characterize the similarity of two functions. We plot the CDF of MSD in Figure 4. We see all MSD is less than 11%. This concludes the SNR-FDR relationship is stable with respect to location changes in different environments we have tested.

D. Impact of Different Hardware

We study whether the SNR-FDR relationship is affected by different hardware. We replace the receiver hardware used in previous experiments with a Linksys card. We collect two traces for every receiver along each path. In Figure 5, it shows that the SNR-FDR relationship of these two receivers differ significantly. There are about 4dB shift in the SNR threshold in each rate. Note that we use the same receiver and we exactly follow the same path for the two traces. Similar experiments in different environments are performed and this phenomenon is consistent. Therefore, this difference is consistent with the same hardware while independent of experimental environment. This observation suggests that mapping from SNR to optimal rate is

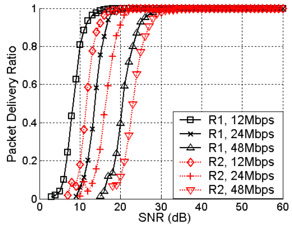


Figure 5. The SNR-FDR relationship for two different receivers. R1 is a NetGear card and R2 is a LinkSys card. The traces are collected along the same path.

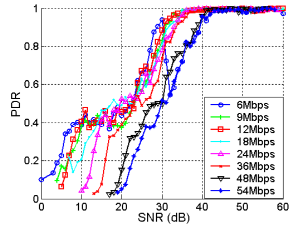


Figure 6. The impact of interference on the SNR-FDR relationship in the indoor office environment, where we explicitly add an interference source.

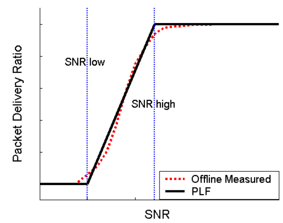


Figure 7. SGRA use two SNR thresholds and PLF to approximate the SNR-FDR relationship. We can see that this model characterizes the relationship well.

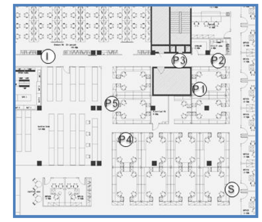


Figure 10. Experiment setting, I is a hidden node, S is the sender, and P1~P5 are 5 receiver locations.

very difficult in practice. Different hardware can turns out with different SNR thresholds. As a consequence, calibration must be done at least per card basis.

E. Impact of Interference

If interference exists in the environment, the SNR-FDR relationship may be distorted. As shown in Figure 6, where we explicitly add an interference source, we can see that the interference source will cause many frame losses. The transition bands are stretched and the SNR-FDR curves become irregular.

Table 1. FDR of two nodes in different locations with and without the interference source. The first two rows present the results in Location 1, and the last two rows preent the results in Location 2.

Rate(Mbps)	6	9	12	18	24
L1 w/o I	0.98	0.98	0.99	0.98	0.98
L1 w I	0.06	0.13	0.26	0.34	0.37
L2 w/o I	0.98	0.99	0.98	0.98	0.99
L2 w I	0.95	0.95	0.92	0.42	0.36

To understand better how interference impacts rate selection, in Table 1, we summarize FDR of different rates for two nodes in two different locations receiving broadcast data from the same sender with and without an interference source. It is clear to see that without interference, both nodes can achieve very high FDR because there are high SNR values in both nodes. However, when the interference source is active, the behaviors of two nodes are much different. At location 1 (L1), with the decrease of the transmission rate, FDR drops from 37% to 6%; while at location 2 (L2), FDR increases with the decrease of the transmission rate. It can be explained as it is actually Signal-to-Interference-plus-Noise Ratio (SINR) that determines the delivery rate of a frame. L1 is near the interference source, so that even the lowest rate cannot pass through. As lower rate prolongs the transmission time, it increases the frame loss rate. In this case, the optimal rate is 24Mbps. L2 is far away from the interference. Therefore, the interference signal has been much attenuated. With a lower rate, e.g. 12Mbps, many frames can be delivered. In this case, the optimal rate is 12Mbps. However, if one always uses SNR to predict rate, it will always select 24Mbps rate, which turns to be the second worst rate in L2. Therefore, when interference exists, using SNR can overestimate the channel quality.

III. SNR-Guided Rate Adaptation

In this section, we present our design of a practical SNR-Guided Rate Adaptation (SGRA).

A. On-line Calibration

In Section II.C, we propose to use two parameters (SNR_{low} and SNR_{high}) to characterize this SNR-FDR relationship, as we observe there is a transition band in SNR. Moreover, we use a Piecewise Linear Function (PLF) to approximate the SNR-FDR relationship, as shown in Figure 7. Based on this model, we develop an on-line calibration scheme to reliably find out the corresponding SNR_{low} and SNR_{high} for each rate from on-line measurements. We sample channel and get a measure of SNR and FDR in a time window of 20ms, during which the channel is stable according to our observation in Section II.B.

The fundamental difficulty here is how we can handle interference. As we show before, interference could distort this SNR-FDR relationship by introducing additional frame-losses (Figure 6). Fortunately, as interference would always shift the curve to the right-hand, it would be possible to filter out interference-polluted samples. To achieve this, we calibrate SNR_{low} as the lowest SNR value that at least gives 10% FDR, and similarly SNR_{high} is the lowest SNR value that at least gives 90% FDR. Here, we implicitly assume we could opportunistically get samples when the channel is temporarily clean. For robustness, we use the 10 percentile instead of the lowest value in our implementation.

There is yet another subtle interaction between on-line calibration and the rate control algorithm (described later in next section). Let's consider the following scenario. Assume the on-line calibration has not converged. A node may transmit in a high rate, but at the same time it encounters an interfering time period. As a result, it gets a SNR_{low} sample for the high rate and switch to a lower rate. However, this SNR_{low} threshold may be too high. Now, the channel is slightly degraded, but the interference is gone. So the optimal action should be switching back to high rate. However, SGRA would prevent so as current SNR is lower than SNR_{low} of the high rate, and this estimation will not be updated since no frames will be sent with that rate. So it becomes a sort of chicken-egg problem.

We break this problem by adding a procedure of *forced probe*. A *forced probe* is triggered if these conditions are satisfied:

1. Current SNR measure is less than SNR_{low} of next higher rate;
2. In last τ seconds, a *forced probe* is not triggered; and
3. The sender stays at a stable rate for at least τ seconds before next *forced probe*.

The *forced probe* sends a packet with the next higher rate. If the packet is passed (maybe through retransmission), the node will continue send one more probe packet. If it again gets passed, the node will update $\mathrm{SNR_{low}}$ for the next higher rate. This may cause this higher rate be selected and thus continue the on-line calibration process.

The pseudo-code for the on-line calibration process is outlined in Figure 8. The function *OnlineCalibration* is called each time a sample on the channel is made. Note that the last function *monotonicity* maintains simple relations between all calibrated SNR thresholds that satisfies: 1) $\mathrm{SNR_{low}(R)} \leq \mathrm{SNR_{high}(R)}$; 2) $\mathrm{SNR_{high}(R)} \leq \mathrm{SNR_{low}(R)} + \delta$; and 3) $\mathrm{SNR_{low}(R)} \leq \mathrm{SNR_{low}(R+1)}$. Note that rule 2 tries to upper bound the wide of transition band in practice. In our implementation, we choose $\tau = 1\mathrm{s}$ and $\delta = 7\mathrm{dB}$.

```
OnlineCalibration( SNR, FDR, R_cur) {
    if (SNR < SNR_low[R_cur] and FDR > 0.1) )
        SNR_low[R_cur] = SNR;
    else if (SNR < SNR_high[R_cur] and FDR >0.9)
        SNR_high[R_cur] = SNR;
    if ( is_time_forced_probe () )
        do_forced_probe ();
    monotonicity ();
}
```

Figure 8. The online calibration algorithm

B. Rate adaptation

SGRA maintains a dynamic estimation of $\mathrm{FDR_R}$ on each possible transmission rate *R*. The basic rate selection routine is to iterate all possible rates and find out the one that has the maximal expected throughput, i.e. $\mathrm{FDR_R} \cdot \mathrm{R}$.

1) Detect interfered channel: We define FDR_R^* as the predicted FDR for rate R which is gotten from the currently measured SNR and the calibrated SNR-FDR relationship. SGRA detects interfered channel by comparing the measured FDR to the FDR_R^*. More specifically, SGRA declares the channel is in interfered state if following two conditions satisfied: 1) current SNR is larger than SNR_{high} of current rate; and 2) FDR is less than FDR^* by a threshold θ.

2) Rate adaptation in interference-free state: If no interference has been detected, since predicting FDR from SNR measure is fair accurate and therefore SGRA aggressively use this prediction to select rate. More specifically, when a SNR sample is measured, SGRA uses this FDR_R^* to updates FDR_R of all unused rates.

3) Rate adaptation in interfered state: If the channel state is interfered, then directly update each FDR_R with the predicted value, FDR_R^*, would over-estimate the delivery ratio as some frames may be lost due to collision. In SGRA, FDR_R^* is still used to update FDR estimation of rate R, but with cautions. As line 14 in Figure 9, we proportionally count in the packet loss caused by interference. Furthermore, SGRA use probing to detect whether the strength of interference changes. The rate to probe is selected in different cases. If the case is as at L1 in Table 1, we choose to probe the lower rate that can achieve better throughput if FDR in that rate is about 100%. Otherwise as at L2, we choose to probe the next higher rate to see if the strength of interference decrease. The probing is scheduled every 1s and lasts for 20ms or at least 20 frames.

```
1   Variable:
2       State : interference_free or interfered
3   Rate_Adaptation ( FDR, SNR, R) {
4       if ( detect_interferece )
5           State = interfered;
6       else
7           State = interference_free;
8       foreach Rate i do
9           if (i == R) FDR[R] =FDR;
10          else if (State == interference_free)
11              FDR[i] =FDR*(SNR, R);
12          else
13              FDR[i] = FDR[i] + (FDR*_R(SNR)−FDR*_R(SNR_last)) / (FDR*_R(SNR_last)/FDR[i])
14  }
15  Find_Rate () {
16      if (! is_probling () )
17          Optimal_rate = argmax_R FDR[R] * R;
18      else
19          Optimal_rate = Get_probe_rate ();
20      Return Optimal_rate;
21  }
```

Figure 9. The SGRA algorithm

The pseudo-code of SGRA rate adaptation is outlined in Figure 9. Function Rate_Adaptation is called each time a sample comes. Function Find_Rate is called each time a packet is about to be sent out and returns the rate for next transmission. The is_probing function tests if the packet should be used to probe addition rate, as discussed above. The Get_probe_rate returns the rate that needs be probed.

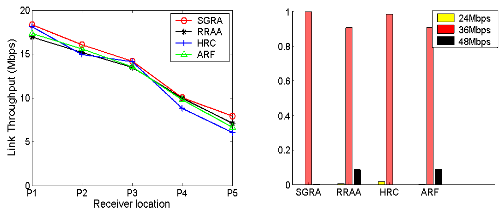
IV. Implementation and Performance Evaluation

We have implemented SGRA as well as RRAA, ARF and HRC in Windows platform based on Atheros ar5212 Chipset. Since the content of ACK is not able to be modified to carry back the SNR, we measure the RSSI of ACK instead. To select parameters, we choose $\mathrm{P_{MTL}}$ and $\mathrm{P_{ORI}}$ of RRAA for 802.11g based on the method described in the paper. Probe interval and rate decision window of HRC are set to 40ms and 1 second. Timeout of ARF is set to 500ms.

We evaluate the performance of SGRA in our test-bed as shown in Figure 10, where *S* denotes a sender, *I* denotes a hidden node, and *P1~P5* represent 5 different locations to place receiver. There's an 802.11g corporation network deployed for the entire floor, but we did not detect any 802.11a devices. For each run of experiment, we generate 2mins back-to-back UDP traffic. The experiment is repeated for 3 runs and the results presented are the average. The packet size is 1500 Bytes.

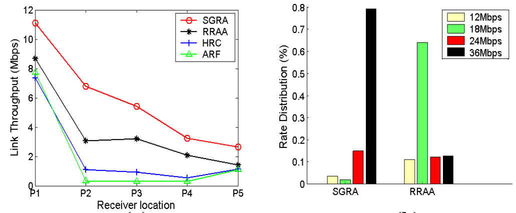
A. Online calibration

The performance metrics of online calibration algorithm are long term accuracy and convergence time. To evaluate the accuracy, we run SGRA and walk around to automatically characterize the SNR-FDR relationship. We compare these online calibrated SNR thresholds with the measured values as described in section II. The difference is consistently no more than 1 dB. We further enable the interference source, the calibration results do not change. These results show that the



(a) (b)

Figure 11. a) UDP throughput in 802.11a. b) Rate distribution for each algorithm at P1. SGRA avoids probing the higher rate and thus outperforms other algorithms.



(a) (b)

Figure 12. a) UDP throughput in 802.11a with a hidden node. b) Rate distribution for SGRA and RRAA at P1. In this typical interference setting, SGRA shows significant performance gain.

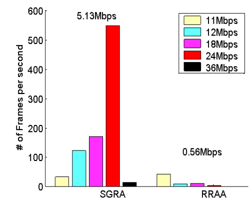


Figure 13. Experiment with bluetooth interference. SGRA obtains 900% throughput as RRAA.

online calibration algorithm of SGRA is accurate and robust. The convergence time of the all SNR-FDR curves relies on how the SNR varies. Here we only care about the time SGRA obtains the optimal rate. We change the position of the nodes and then start SGRA. At all the positions, SGRA always get to the optimal rate at the first second.

B. Interference free

We test SGRA using 802.11a on a clean channel. Figure 11a shows throughput of the four algorithms at the five locations. SGRA outperforms the other three algorithms at all the locations, with the achieved throughput gain ranging from 7% to 30%. Figure 11b shows the rate distribution of four algorithms at P1. We can see that SGRA has less probing overhead than all the other algorithms.

C. 802.11 Hidden Terminal

In this section, we let the Node *I* broadcast at a constant speed of 2.5Mbps, which results in a 802.11 'hidden terminal' for Sender S and induce 60% frame loss at most of the locations. The throughput of all the algorithms is shown in Figure 12a. ARF and HRC leads to starvation. RRAA does not starve. SGRA outperforms RRAA and yields a throughput gain from 30%~120%. The performance loss of RRAA is mainly because of two reasons: 1) RTS/CTS cannot wholly prevent collisions from hidden terminals; and 2) the RTS/CTS handshake introduces considerable overhead. In Figure 12b, rate distribution at P1 shows that RRAA doesn't obtain the optimal rate whereas SGRA does.

D. With External Interference

IEEE 802.11 based systems are easily interfered by other wireless devices which shares the same unlicensed frequency band e.g. Bluetooth, Microwave oven and cordless phones. In this scenario, we compare the performance of SGRA and RRAA. At location P1, we put two laptops near the receiver and transmit a file over Bluetooth between them. The sender S sends UDP back-to-back packets to the receiver on 802.11g channel. The number of frames transmitted by SGRA and RRAA at different rate is shown in Figure 13. We can see that RRAA stays at the lowest rate whereas SGRA stays at a high rate which yields 5.13Mbps throughput, almost 900% compared to the 0.56Mbps throughput of RRAA in the same condition.

V. RELATED WORKS

Rate adaptation algorithms can be divided to loss based and SNR based. Loss based algorithms have been discussed in section I. All most all the previous SNR based algorithms are not studied in real environment except HRC [6]. HRC proposed a parameterized method on-line correcting the SNR-to-Rate mapping. However, this method is not practical since the parameters are environment dependent and therefore impossible to obtain. During our experiments, we find the default value of these parameters make the SNR-to-Rate mapping very unstable and drive HRC quickly into starvation. The fundamental problem of HRC is that it does not consider external interference, and thus doesn't perform well in real environment.

VI. CONCLUSIONS

In this paper, we make two key contributions. First, we perform a systematical measurement-based study on the capability of SNR to characterize the channel quality using commodity NICs. We confirm that SNR is a good prediction tool for channel quality, but also identify several practical challenges. Second, we design, implement and evaluate a novel SNR-Guided Rate Adaptation that addresses all identified challenges. SGRA is fully compliant with 802.11 standards. Thorough evaluation showed that compared to existing work, SGRA achieves 7-30% throughput gain for interference-free scenario, 120% gain when there exists 802.11 hidden terminal and boosts network throughput up to 900% when non-802.11 interference exists.

REFERENCES

[1] S. Wong, H. Yang, S. Lu and V. Bharghavan, "Robust Rate Adaptation for 802.11 Wireless Networks," in *Proceedings of ACM MOBICOM,* 2006.

[2] G. Holland, N. Vaidya and P. Bahl, "A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks," in *Proceedings of ACM MOBICOM,* 2001.

[3] A. Kamerman and L. Monteban, "WaveLAN-II: a high-performance wireless LAN for the unlicensed band," *Bell Labs Technical Journal*, vol.2, no. 3, pp. 118-133, August 1997.

[4] B. Sadeghi, V. Kanodia, A. Sabharwal and E. Knightly, "Opportunistic Media Access for Multirate ad hoc Networks," in *Proceedings of ACM MobiCom*, 2002.

[5] J. Kim, S. Kim, S. Choi, and D. Qiao, "CARA: Collision-Aware Rate Adaptation for IEEE 802.11 WLANs," in *Proceedings of IEEE INFOCOM*, 2006.

[6] I. Haratcherev, K. Langendoen, R. Lagendijk, and H. Sips, "Hybrid Rate Control for IEEE 802.11," in *Proceedings of ACM MobiWac*, 2004.

[7] C. Chen, H. Luo, E. Seo, N. H. Vaidya and X. Wang, "Rate-Adaptive Framing for Interfered Wireless Networks," in *Proceedings of IEEE INFOCOM*, 2007.

[8] C. Chen and H. Luo, "The Case for Heterogeneous Wireless MACs," in *Proceedings of HotNets*, 2005.