Methods for Restoring MAC Layer Fairness in IEEE 802.11 Networks with Physical Layer Capture

Sachin Ganu, Kishore Ramachandran, Marco Gruteser, Ivan Seskar and Jing Deng[*]
WINLAB, ECE Department
Rutgers, The State University of New Jersey
671 Route 1 South, New Brunswick, NJ 08902
{sachin, kishore, gruteser, seskar}@winlab.rutgers.edu, jing@cs.uno.edu

ABSTRACT

In this paper, we experimentally investigate the physical layer capture effect in off-the-shelf 802.11 network cards and confirm that it reduces throughput fairness of traffic flows. We then study the feasibility of using the following PHY and MAC layer approaches to mitigate the disproportionate allocation of throughput in capture dominated scenarios: transmit power control, retransmission limits, CWmin adjustment, TXOP adjustment, and AIFS control. The results obtained on the ORBIT indoor wireless testbed[1] show that the 802.11e EDCF parameters provide the most fine-grained control of fairness.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Measurement techniques; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design– wireless communication

General Terms

Design, Experimentation

Keywords

Wireless Networks, Experimental evaluation, capture effect, fairness, EDCF

ABSTRACT

In this paper, we experimentally investigate the physical layer capture effect in off-the-shelf 802.11 network cards and confirm that it reduces throughput fairness of traffic flows. We then study the feasibility of using the following PHY and MAC layer approaches to mitigate the uneven allocation of throughput in capture dominated scenarios: transmit power control, retransmission limits, RTS/CTS, CWmin adjustment, TXOP adjustment, and AIFS control. The results obtained on the ORBIT indoor wireless testbed1 show that the 802.11e EDCF parameters provide the most fine-grained control of fairness. ; C.4 [**Performance of Systems**]: Measurement techniques; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design–wireless communication

General Terms

Design, Experimentation

Keywords

Wireless Networks, Experimental evaluation, capture effect, fairness, EDCF

1. INTRODUCTION

Estimating and controlling the share of bandwidth available to a communication stream over a mobile ad hoc network requires understanding and controlling MAC layer fairness. Moreover, in a multi-hop ad-hoc network, as the number of hops increases, the overall throughput performance deteriorates due to self-interference of transmissions along the forwarding path. Thus, these networks can easily reach a congested state with several simultaneous flows. Under these conditions, the share of channel capacity for each flow is governed by the throughput fairness properties of the system.

In this paper we study per-node throughput fairness for a single bit-rate network using an experimental methodology that can reconstruct a global per-packet timeline of the transmission from several senders. While other notions of fairness, such as per-flow fairness or time-based fairness are often the goal in ad hoc networks, these are difficult to achieve without control over the basic per-node fairness properties of the underlying MAC layer. Experimental measurements show that the physical layer capture effect significantly reduces per-node throughput fairness. Physical layer capture is a phenomenon where, in the event of a collision between two frames at a receiver, the hardware is capable of detecting and decoding the packet with a stronger signal strength. This effect has been observed with multiple wireless NICs based on different chipsets (Atheros and Prism), occurring even in small setups (about 10m separation) with line-of-sight communications and is not usually modeled correctly in existing simulation tools (as

---

[*] J. Deng is with the Department of Computer Science, University of New Orleans, 2000 Lakeshore Dr., New Orleans, LA 70148.

[1] This work was supported by NSF ORBIT Testbed Project, NSF NRT Grant ANI0335244

[*] J. Deng is with the Department of Computer Science, University of New Orleans, 2000 Lakeshore Dr., New Orleans, LA 70148.

REALMAN'06, May 26, 2006, Florence, Italy.

shown in [1]). We then measure the effectiveness of several mechanisms to restore fairness, including transmission power control and backoff adjustments through the Wireless Multimedia Extensions derived from the IEEE 802.11e standards. Two mechanisms in particular, TXOP and AIFS control, are most promising. These could form the backbone of a distributed algorithm to monitor and control fairness in ad hoc networks.

The organization of the paper is as follows: Section 2 describes related work. In Section 3 describes the experimental setup used to detect capture using an approach based on wireless sniffers and packet level analysis. In Section 4, unfairness in flow throughputs caused by the capture phenomenon is evaluated. We explore various physical and MAC layer options to restore throughput fairness and summarize the effectiveness of each of these in Section 5. A heuristics based approach to restore fairness for a multiple flow network is also proposed and evaluated. Section 6 presents our conclusions and also motivates future work.

2. RELATED WORK

The existence of physical layer capture (PLC) [2] effect in 802.11 networks has been studied analytically and using simulations in [3]. A general description of the PLC effect is as follows: if two MAC frames collide at the receiver, the frame with the stronger signal strength will still be correctly decoded. In [1], the authors present an empirical study of PLC and provide evidence to show that in the event of collision between frames, the stronger frame is decoded irrespective of its arrival time relative to the other frames involved in the collision (provided it is within 128μs from the start of reception of the first received frame [4, pp. 202-203]). The implications of this effect are that the traditional view of a collision that assumes the loss of all involved packets or frames does not apply. Moreover, if this effect happens consistently and frequently, it can be a source of significant unfairness between throughputs of stronger senders that are captured by the receiver, and those of weaker senders that experience multiple retransmissions and backoff. An experimental study in [5] presents the unfairness caused by PLC for TCP flows in hidden terminal scenarios. This phenomenon is shown to occur despite the use of RTS/CTS frames with SNR differentials as low as 5dB. Our contributions include confirming the existence of this effect as well as an experimental evaluation of various PHY and MAC layer parameters to restore fairness in environments where PLC is present.

Previous work has also looked at unfairness problems arising in 802.11 networks due to contention between upstream data towards the Access Point (AP) and downstream acknowledgements from the AP towards the clients. In [6, 7], Leith et al. use the 802.11e [8] Enhanced Distributed Co-ordination Function (EDCF) parameters such as CWmin, TXOP and AIFS interval to alleviate unfairness between multiple contending TCP flows in 802.11 infrastructure networks. They utilize these parameters to prioritize downstream traffic from the Access Point (AP). However, these studies were carried out using a topology that minimized PLC - all stations were positioned in a manner such that they had a similar radio link to the AP. Other related work considers reliable transport protocol fairness over WLANs but propose solutions that either require changes to the 802.11 MAC protocol [9] or modifications to TCP acknowledgements [10]. We differ from these works in that our solutions do not require changing the underlying protocol.

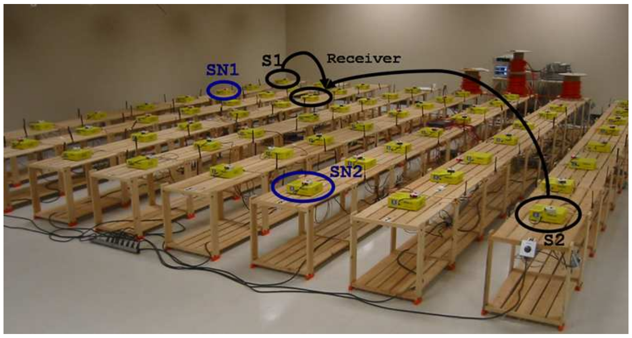


Figure 1: 8x8 radio grid testbed

3. EXPERIMENTAL SETUP

3.1 The Testbed

All our experiments were conducted on the ORBIT testbed comprising 64 wireless nodes arranged in an 8x8 grid [11, 12] as shown in Figure 1. Each node has two 802.11 a/b/g cards. We used 802.11b channel 1 for all our experiments. There is an equal distribution of nodes with Intel IPW 2915 chipset based cards and Atheros AR5212 chipset based cards.

For all our experiments, we have used the nodes with Atheros cards since they allow software control over various parameters such as CWmin selection, disabling retries etc. The open source Madwifi [13] driver for the Atheros chipset based cards implements a majority of MAC protocol features in the driver rather than in hardware, thereby allowing a variety of modifications at the software level. We have also developed a supporting software library that allows us to extract useful information such as RSSI, PHY rate, hardware timestamp (1μsecond granularity) from the device driver for each successfully received packet. Note that there are no hidden nodes in our testbed and each node is within transmission range of every other node. There is no external interference from other 802.11 wireless devices in all our experiments. This was verified by using the *iwlist (interface) scan* utility that detects infrastructure or ad-hoc networks in the vicinity.

3.2 Analysis of the capture phenomenon

To experimentally detect the physical layer capture phenomenon, we adapted the technique of using per-sender sniffers and constructing a global timeline of all transmission and reception events in each of our experiments, as described in [1].

3.2.1 Methodology

In these experiments, we use two transmitters S1 and S2 that send packets to a common receiver. We chose one sniffer near each sender (as shown in Figure 2) such that the signal strength or RSSI of packets received from this sender is higher than that of frames received from any other sender. The reasoning behind this placement is that a sniffer is also a regular radio receiver susceptible to the capture phenomenon. The primary difference between our technique and the one proposed in [1] is the use of a feature provided by Atheros cards - a station can perform ”live monitoring”[1] and observe WLAN traffic while still being synchronized with the rest of

---

[1] The driver provides a separate virtual network interface, called ath0raw, which can be used to send/ receive frames directly to/from the card from user-space (bypassing the driver state machine). This interface can be enabled using the commands: *sysctl -w dev.ath0.rawdev=1; ifconfig ath0raw up;*

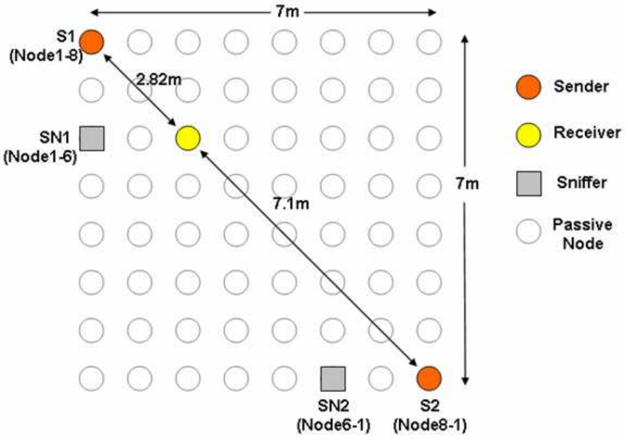


Figure 2: Experiment setup to study capture effect

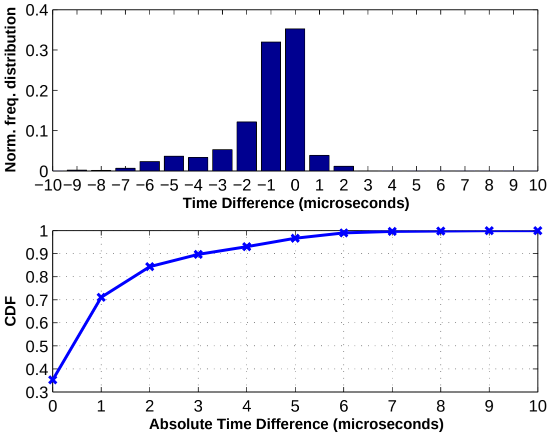


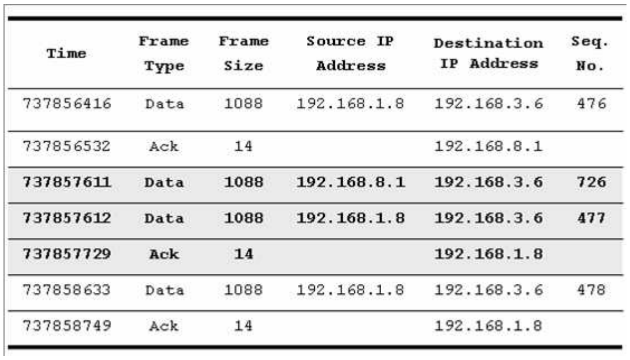
Figure 3: Histogram and Cumulative density function of the difference between the hardware timestamps at each sniffer for the same packet sent by one sender.

the stations in the network. This implies that the logs from each of the sniffers do not have to be explicitly ”synchronized”; they can be merged directly based on the hardware timestamp of each received frame. We used tcpdump [14] on the sniffers and processed the collected information using *awk* scripts.

3.2.2 *Detecting packet capture using traces*

To measure the synchronization accuracy between sniffers, we calculated the difference between the hardware timestamps for each frame received by both our sniffers. Figure 3 shows the cumulative distribution function (CDF) and histogram of these differences from one of our experiments.

As seen in the Figure, the absolute difference does not exceed 9μs and, for more than 95% of frames, the difference is less than 5μs[2]. Given that the transmission time of an 802.11b frame is at least 120μs (using short PLCP header), we believe this accuracy to be sufficient.



Time	Frame Type	Frame Size	Source IP Address	Destination IP Address	Seq. No.
737856416	Data	1088	192.168.1.8	192.168.3.6	476
737856532	Ack	14		192.168.8.1	
737857611	**Data**	**1088**	**192.168.8.1**	**192.168.3.6**	**726**
737857612	**Data**	**1088**	**192.168.1.8**	**192.168.3.6**	**477**
737857729	**Ack**	**14**		**192.168.1.8**	
737858633	Data	1088	192.168.1.8	192.168.3.6	478
737858749	Ack	14		192.168.1.8	

Figure 4: Collision detection - the highlighted rows represent collision and subsequent capture. The two frames are received 1μs apart but an acknowledgement is sent to the stronger sender.

Table 1: Interframe delay

	Strong sender	**Weaker sender**
Average delay	1.443ms	3.957ms
Maximum delay	22.72ms	207.71ms
Variance	1.884ms	82.4168ms

Figure 4 shows a snapshot from one of our traces that demonstrates the capture phenomenon. From these merged traces, we can see that frames collided because they picked the same time slot for transmission and an 802.11 acknowledgement was sent back for one of the senders implying that the stronger frame was correctly decoded. Thus, the stronger sender is able to transmit the next frame while the weaker sender doubles its contention window and backs off. Table 1 shows the average and maximum delay between two successful transmissions and the variance of this delay.

This shows that, on average, the weaker sender has to wait much longer before its next opportunity to send a packet. This results in a disproportionate share of throughput for the flow that experiences multiple retransmissions due to capture. In the next section, we quantify the observed unfairness due to the capture phenomenon in terms of UDP throughput.

4. CAPTURE EFFECT AND FAIRNESS

Using the same experimental settings as described earlier, we measured the throughput unfairness caused by PLC. We used the Iperf traffic generator [15] to generate UDP traffic at each transmitter. Each sender uses an offered load of 8 Mbps during the course of the experiment.The goal was to observe the flow throughputs for different packet sizes. We used packet sizes of 256, 512 and 1024 bytes for this experiment. For each test, both senders used the same CWmin (default set to 31).

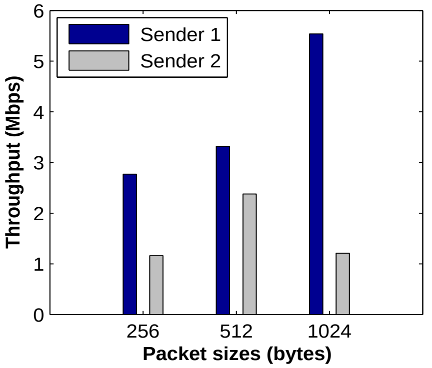
As seen in Fig 5(a), there is significant unfairness in the throughputs of sender S1 and S2 at the receiver. Unfairness is higher for the larger packet sizes (1024 bytes). The observed RSSIs of each sender plotted in Fig 5(b) show that S1 is received almost 20 RSSI units stronger than S2.

5. TECHNIQUES FOR RESTORING FAIRNESS

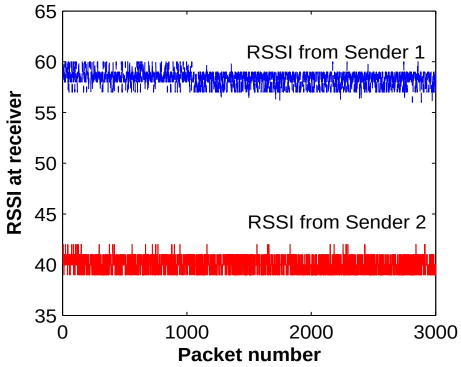
In order to restore fairness caused by PLC, we experimentally evaluate various approaches that span both PHY layer as well as

---

[2] In [1], the maximum absolute difference does not exceed 4μs. We attribute the relative difference in our observations to the differences in underlying hardware (chipset) and the technique used to timestamp each incoming flow



(a) Individual flow throughputs for different packet sizes



(b) RSSI at receiver for each sender

Figure 5: Throughput unfairness due to PLC

MAC layer adjustments. In particular, we look at the following knobs and their effectiveness in restoring fairness.

- Transmission power control (Physical Layer)
- Retransmissions (MAC)
- 802.11e QoS Parameters
  - CWmin (MAC) (default = 31)
  - TxOP (MAC) (default = 1 packet per attempt)
  - AIFS (MAC) (default = DIFS)

In addition, the advantages and limitations of each approach are described. All the experiments were conducted with three different packet sizes (256 bytes, 512 bytes and 1024 bytes). Each experiment lasted 60 seconds. In each run, a set of sniffers receives and reports every transmission during the course of the experiment. Independently, we used the *athstats* tool (provided with the Madwifi driver) to record successfully transmitted packets and the retries at each node throughout the experiment duration.

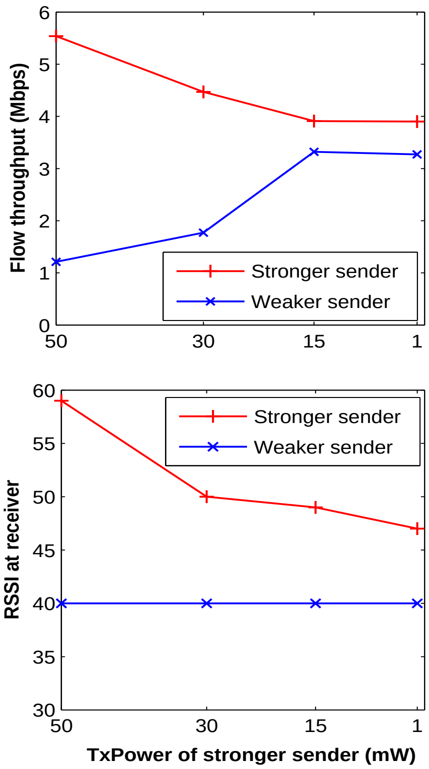


Figure 6: Throughput distribution and RSSI at the receiver with transmission power control at the stronger sender

5.1 Transmission Power Control at the Stronger Sender

The first approach to mitigate unfairness is to reduce the transmit power of the sender whose signal strength is stronger at the receiver. We configure [3] the transmit power of the stronger sender from 60mW (18dBm) down to 1 mW (0 dBm) with two intermediate power levels of 30 mW (14.7 dBm) and 15 mW (11.7 dBm).

As seen in Figure 6, transmission power control at the stronger sender reduces the gap between the two flow throughputs as well as the signal strength difference at the receiver from the two senders. A possible explanation is that since the difference in RSSI (and hence SNR) at the receiver from the two senders is lower, the probability of capture of the stronger sender is reduced. This results in an improvement in throughput for the weaker sender. However, using transmit power control alone, we were unable to restore fairness between the flows because of the limited dynamic range of allowable power level settings. Typically, most of the current hardware devices available off the shelf do not allow power levels below 1 mW or 0 dBm. Additionally, there is no hardware support for per-packet transmission power adaptation and only certain discrete power levels are allowed, thereby limiting the granularity of control.

5.2 Adjusting MAC retry limit

---

[3] For the MadWifi driver, we write to a file in the /proc directory (*/proc/sys/dev/ath0/txpowlimit;*) using the *echo* command. The values are from 1 to 100 in milliwatts which translate to 0 to 18dBm (clamped at 18 dBm).

Due to PLC, the weaker station has to retry packets that collided and were dropped by the receiver. According to the 802.11 standard, this station doubles its contention window for each unsuccessful attempt and defers until the CW counts down to zero. This significantly reduces the amount of data traffic that the station can send.

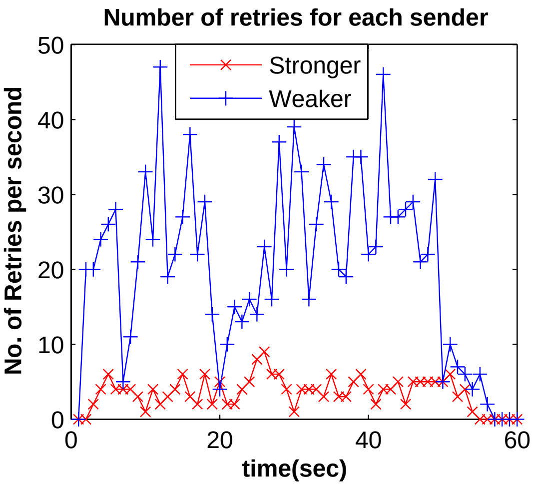


Figure 7: Number of retransmission attempts per second at each sender during the experiment duration

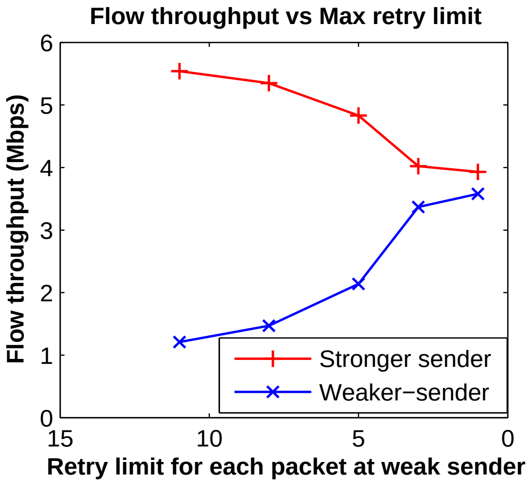


Figure 8: Flow throughputs as a function of per packet Tx attempts limit for the weaker sender

In our experiments, we measured the cumulative number of retries by each sender (reported per second) over the entire experiment duration. As seen in Figure 7, the weaker sender encounters 4x more retransmissions than the stronger sender, on average. In our experiment, we varied the maximum number of transmission attempts per packet at the weaker sender from the default setting of eleven to one (no retries). As seen in Figure 8, as the retry limit is decreased, the weaker sender spends lesser time in backoff before attempting to transmit the next packet. This results in a higher UDP throughput. This trend is seen for all the packet sizes that we studied. Figure 9 shows the flow throughputs after disabling retransmissions at the weaker sender for each packet size. Thus, disabling retransmissions may be used as an option by applications that are tolerant to packet losses (since the packets are dropped from the buffer if they a re unacknowledged and MAC retries are disabled). However, it would have an impact on the performance of the applications that use TCP as the underlying transport layer.

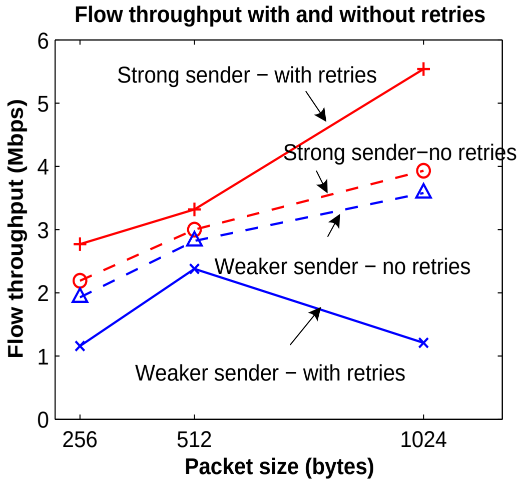


Figure 9: Flow throughputs for different packet size with and without retransmissions at the weaker sender

5.3 Tuning EDCF QoS Parameters

As per the latest 802.11e [8] standard, each station supports up to four queues for traffic. Each queue is associated with a specific access category (AC) and contends for the channel independent of the others. Different levels of service are provided to each AC through a combination of three service differentiation mechanisms as follows:

- CWmin for each AC
- Transmit opportunity (TXOP);
- Arbitration Inter-frame space (AIFS)

The Madwifi driver for Atheros chipset based cards exposes most of these settings, with a hardware abstraction layer (HAL) controlling the actual interface to the hardware.

5.3.1 Adjusting minimum contention window size

The basic idea behind adapting the minimum contention window is to increase the likelihood of channel access for the weaker sender (based on the probabilistic assumption that the weaker sender will, on average, select earlier slots than the stronger one). We tried to set arbitrary values for the CWmin values that were not powers of two, however our observation was that the HAL rounds it off to the next higher power of two, thereby restricting our adjustment choices.

In Figure 10, the numbers in the brackets represent the tuple (CWminSS, CWminWS) where SS and WS imply the strong sender and the weak sender respectively. For each packet size, reducing the CWmin of the weaker sender increases its share of throughput. This is seen for the setting (31,15) in each case. However, reducing CWmin further tends to overcorrect the unfairness as seen in the (31,7) case for each packet size. We also increased the CWmin for the stronger sender to 63 while keeping the default CWmin for the weaker sender. This is represented by the (63, 31) column for each packet size. Even though the flow throughputs are more proportionate for this setting, we see that it results in a reduction in the

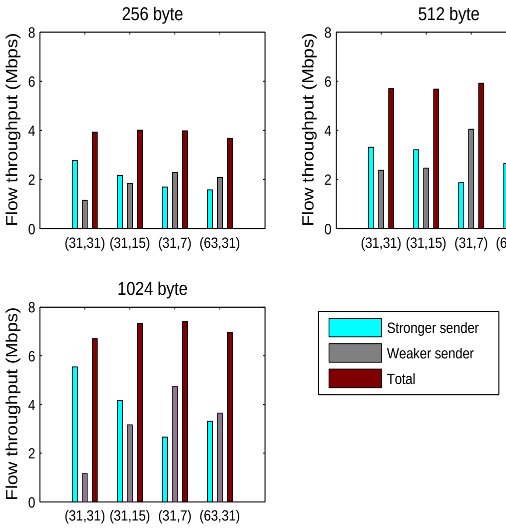


Figure 10: Flow throughputs for different packet sizes with different CWmin combinations

overall system throughput because of inefficient use of the channel.

5.3.2 *Adjusting TXOP*

IEEE 802.11e provides TXOP (Transmission Opportunity in units of μseconds) for each class of service. This allows stations to send more than one packet separated by SIFS during their channel accesses instead of having to contend for the medium for every packet. By default, the transmit opportunity is set to one packet per channel access. Under ideal conditions, the two flows should contend equally for the channel and gain equal amounts of time to transmit data. However, in the event of collisions, capture and retransmissions, this time share on the channel is disproportionate. In order to rectify the problem, we varied the TXOP parameter for the weaker sender roughly in units of time required to transmit one packet of the given size. We only present the results for the 1024 byte packets due to space limitations.

The total transmission time for a 1024 byte packet (with additional 28 byte MAC header + 8 byte SAP/SNAP header + 20 byte IP header + 8 byte UDP header) using the short preamble option is around 911 μseconds. Also, the station has to wait for DIFS interval and an additional deferral time before it can send the first packet. In our experiments, we used normalized TXOP of 2 and 3 packets per channel access for the weaker sender (corresponding to 2 ms and 3 ms respectively).

In [6], the authors have reported a linear relationship between throughput and TXOP. However, in our capture dominated environment, we found that the throughput increases much slower beyond TXOP = 2. As shown in Figure 11, by setting TXOP = 3 packets per channel access for the weaker sender, we restored throughput fairness. To gain further accuracy, we propose that the proportion of time spent by each flow on the channel should be measured and the TXOP of the weaker sender should be appropriately chosen to balance this ratio.

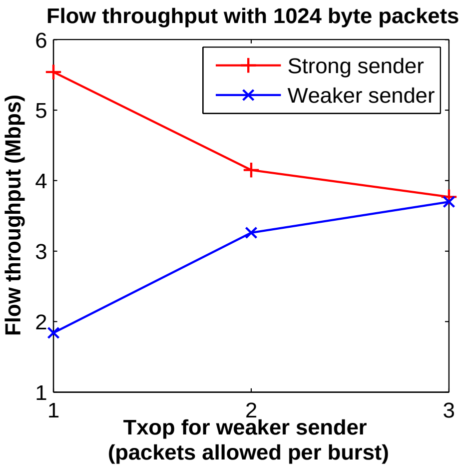


Figure 11: Flow throughputs as a function of TXOP for weaker sender

5.3.3 *Adjusting AIFS*

AIFS (Arbitration inter-frame spacing) is equivalent to DIFS in the 802.11b standard and represents the minimum mandatory spacing between two frames in addition to the deferral time. Prior to sending a packet, each station waits a fixed interval plus an additional randomly chosen interval from (0, CW). By decreasing the AIFS for the weaker sender, we prioritize its transmissions over those of the stronger sender and thus reduce the number of collisions.In our experiments, we varied the AIFS for the strong sender as shown in the Figure 12. For the 1024 and 512 byte packet sizes, AIFS values around DIFS +12 slot times for the stronger sender resulted in fair throughput allocation. For the smaller packet size (256 bytes), this balance was achieved further away at AIFS values of around DIFS+17 slot times for the strong sender.

5.3.4 *Summary of observations and comparison of each approach*

We summarize our observations for each adaptation mechanism and also compare throughput fairness achieved by each approach. Our findings suggest that

- Reducing the transmission power of the strong sender may achieve fairness; however the adjustment is limited by the discrete power levels allowed by the underlying hardware device.
- Reducing the number of retransmissions of the weaker sender helps; this may be useful for applications that are tolerant to packet loss.
- Increasing CWmin of stronger sender may be better than reducing CWmin of weaker sender due to reduced number of collisions in the former case. However, CWmin control is restricted to 10 settings (strictly in powers of 2) and hence we cannot achieve fine grained control.
- Increasing TXOP for the weaker sender allows increased number of packet transmissions per channel access. Also, it allows finer granularity of control as compared to the previous approaches.
- Increasing AIFS for the stronger sender achieves the desired throughput fairness due to reduced number of collisions.

Table 2 summarizes the flow throughput before and after each adjustment.

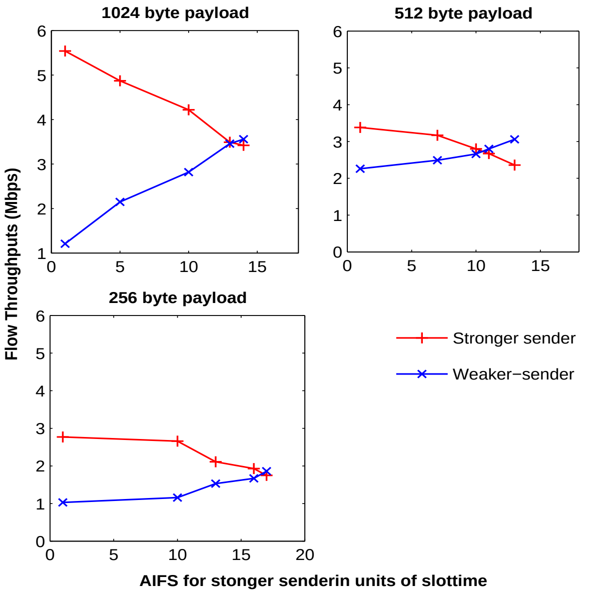


Figure 12: Effect of AIFS on the flow throughputs

Table 2: Fairness achieved by each method

Method	Flow 1 Throughput (Mbps)	Flow 2 Throughput (Mbps)
Default	5.54	1.21
TxPower	3.9	3.27
Retries	3.93	3.58
CWmin	3.31	3.64
TXOP	3.77	3.7
AIFS	3.49	3.46

5.4 Multiple flows and joint adaptation

Based on our observations, we studied the fairness behavior of five different flows chosen such that two out of the five senders had a significantly weaker RSSI at the receiver (approx. 20 units lesser than the stronger stations). Each sender always had a packet to transmit. We used 802.11b channel 1 with fixed rate setting of 11 Mbps. The RSSI and throughput distribution for each flow are shown in Figure 13. We present only the case for 512 byte packets.

It can be seen that the flows with sufficiently higher RSSI always get a much higher proportion of the total throughput; where as the weaker senders suffer due to repeated collisions. The maximum throughput imbalance was as high as 5x. Based on our earlier observations, we employed a two step heuristic approach to mitigate this unfairness:

1. We increased the TXOP for flow 4 (around 2 packets per channel access) and flow 5 (around 3 packets per channel access) based on their respective flow throughputs. Default settings were used for all other parameters. This was done to give weaker senders an opportunity to send additional packets during their successful channel accesses. Figure 14a shows that the throughputs of these flows improve as compared to the default case.

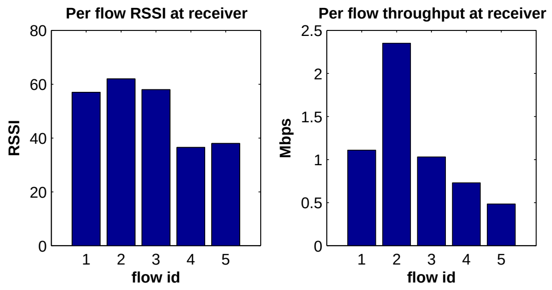


Figure 13: Per-flow RSSI and throughput at receiver. The first three flows are received approx. 20 RSSI units stronger than the last two flows

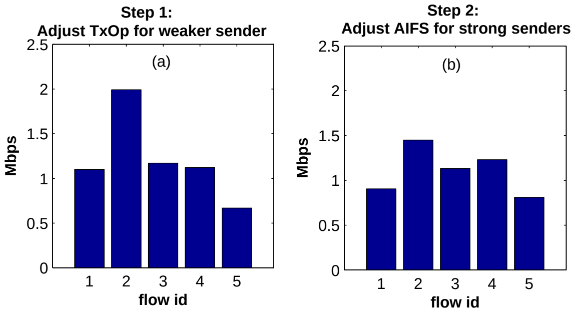


Figure 14: Per Flow throughput distribution after (TXOP, AIFS) correction

2. Flow 2 still has a higher throughput compared to the other flows. We additionally adjust the AIFS of this flow to DIFS + 10 slot times. After step 2, the flow throughputs are more balanced as seen in Figure 14(b).

We quantify the effective fairness gain in terms of Jain's fairness index [16]. The index, F, is calculated as $F = \frac{(\sum_i x_i)^2}{n \times \sum_i {x_i}^2}$ where x_i is the individual flow throughput and n is the total number of flows. An index value equal to one is considered to be perfectly fair.

Table 3 evaluates the gains of our approach w.r.t. the default case with no adaptation.

Table 3: Fairness comparison

Scheme	Fairness Index
Default (no adaptation)	0.7584
Step 1 (Adjust TXOP)	0.8877
Step 2 (Step 1 + Adjust AIFS)	0.9588

Our heuristic approach yields an improvement of about 25% in throughput fairness. For further improvement, the problem has to be studied jointly in the context of all the parameters previously described, and is the scope of our future work. Also, all our experiments were performed using fixed PHY rate settings to eliminate possible effects of rate adapation on PLC. We expect the performance of the weaker sender to deteriorate further if the auto-rate selection implementation in the driver drops the PHY rate to lower values upon encountering repeated losses due to capture. In future work, we plan to study its impact in more realistic variable bit-rate environments.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have experimentally verified the physical layer capture effect in 802.11 network cards as reported by earlier work. We address the related throughput fairness issue by evaluating several PHY and MAC layer options and their effectiveness in restoring fairness. Based on our observations, we apply a heuristic correction method (combined AIFS and TXOP) that yields an improvement of 25% in throughput fairness as compared to default settings. We plan to extend this work by developing efficient algorithms for capture detection as well as restoring fairness using a combination of frame level analysis from distributed sniffers and flow specific feedback from the receiver. Also, the use of RTS/CTS mechanism and evaluating its effectiveness in mitigating the unfairness caused by PLC is also the topic of our future work.

Acknowledgments

The authors would like to thank all three anonymous reviewers for their valuable comments. Part of J. Deng's work was performed while he was visiting WINLAB, Rutgers University in the Fall of 2005.

7. REFERENCES

[1] A. Kochut, A. Vasan, U. Shankar, and A. Agrawala. Sniffing out the correct Physical Layer Capture model in 802.11b. In *Proceedings of 12th IEEE International Conference on Network Protocols (ICNP'04)*, pages 252–261, May 2004.

[2] M. Soroushnejad and E. Geraniotis. Probability of Capture and Rejection of Primary Multiple Access Interference in Spread Spectrum Networks. In *IEEE Transactions on Communications*, pages 986–994, June 1991.

[3] C. Ware, J. Chicaro, and T. Wysocki. Modeling of capture behavior in IEEE 802.11 radio modems. In *Proceedings of the IEEE International Conference on Telecommunications (ICT'01)*, 2001.

[4] IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical layer (PHY) Specifications, June 1997.

[5] C. Ware, J. Judge, J. Chicaro, and E. Dutkiewicz. Unfairness and capture behavior in 802.11 ad-hoc networks. In *Proceedings of the IEEE International Conference on Communications (ICC'00)*, 2000.

[6] Anthony C. H. Ng, David Malone, and Douglas J. Leith. Experimental evaluation of TCP performance and fairness in an 802.11e test-bed. In *E-WIND '05: Proceeding of the 2005 ACM SIGCOMM workshop on Experimental approaches to wireless network design and analysis*, pages 17–22, New York, NY, USA, 2005. ACM Press.

[7] D. J. Leith and P. Clifford. Using the 802.11e EDCF to achieve TCP upload fairness over WLAN links. In *WiOpt '05: Proceedings of the Third International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, pages 109–118, Washington, DC, USA, 2005. IEEE Computer Society.

[8] IEEE 802.11e Wireless Medium Access control (MAC) and Physical layer specifications: Medium Access Control (MAC) Quality of Service(QoS) Enhancements, August 2004.

[9] Haitao Wu, Yong Peng, Keping Long, Shiduan Cheng, and Jian Ma. Performance of reliable transport protocol over IEEE 802.11 wireless LAN: analysis and enhancement. In *Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE INFOCOM 2002*, volume 2, pages 599–607, June 2002.

[10] S. Pilosof, R. Ramjee, D. Raz, Y. Shavitt, and P. Sinha. Understanding TCP fairness over wireless LAN. In *Proceedings of IEEE INFOCOM*, 2003.

[11] Dipankar Raychaudhuri, Ivan Seskar, Maximilian Ott, Sachin Ganu, Kishore Ramachandran, Haris Kremo, Robert Siracusa, Hang Liu, and Manpreet Singh. Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols. In *IEEE Wireless Communications and Networking Conference*, March 2005.

[12] ORBIT testbed, Open Access Research Testbed for Next-generation Wireless Networks. http://www.orbit-lab.org.

[13] Multiband atheros driver for wifi. http://www.madwifi.org.

[14] TCPdump. http://www.ethereal.com.

[15] Iperf traffic generator. http://dast.nlanr.net/Projects/Iperf/.

[16] R. Jain, D. Chiu, and W. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. Technical report, DEC Tech Report TR-301, September 1984.