

# An Experimental Evaluation of Rate Adaptation for Multi-Antenna Systems

Wonsoo Kim, Muhammad O. Khan, Kien T. Truong, Soon-Hyeok Choi, Robert Grant,  
Hyrum K. Wright, Ketan Mandke, Robert C. Daniels, Robert W. Heath, Jr., Scott M. Nettles  
Wireless Networking and Communications Group (WNCG)

Department of Electrical and Computer Engineering  
The University of Texas at Austin

1 University Station C0803, Austin, TX 78712-0240

Email: {wkim, mkhan, ktruong, schoi, bgrant, hwright, mandke, rdaniels, rheath, nettles}@ece.utexas.edu

**Abstract**—Increasingly wireless networks are built around multi-antenna nodes, such as those used for IEEE 802.11n and 802.16. The Physical layer (PHY) in such systems may use the antennas to provide multiple streams of data (spatial multiplexing) or to increase the robustness of fewer streams. These physical layers also provide support for sending packets at different rates by changing the modulation and coding used during transmission. Rate adaptation is essentially the problem of choosing the best transmission mode for the current channel and in these systems requires choosing both the level of spatial multiplexing and the modulation and coding.

Hydra is an experimental wireless network prototype that implements nodes in which both the MAC and PHY are highly programmable. Hydra is well suited to studying rate adaptation in multi-antenna systems. Hydra's PHY is essentially the 802.11n PHY, and currently supports two antennas and the same modulations and codings as 802.11n. Because of limitations of our hardware platform, the actual rates are a factor of 10 smaller than 802.11n. The MAC is essentially the 802.11 MAC with extensions, including the ability to feedback channel state or rate information from the receiver. Hydra also includes the ability to perform experiments over emulated channels using exactly the same MAC and PHY used for RF transmissions. This allows us to perform experiments that control aspects of the channel that are hard to control in real-world measurements.

We present rate control experiments based on transmission over both real and emulated channels. Our experiments include measurements for single antenna systems and two antenna systems using a single or multiple spatial streams. We study rate adaptation algorithms that use explicit feedback from the receiver and ones that use implicit feedback. A novel aspect of our results is the first experimental study of adaptation between single and multiple spatial streams for 802.11n style systems.

Increasingly wireless networking technologies, including IEEE 802.11n and IEEE 802.16, support radios with multiple antennas. These antennas can be used to support multiple data streams (spatial multiplexing) or to increase robustness by taking advantage of channel diversity. Choosing between these two is a basic tradeoff between rate and reliability. Modern Physical layers (PHYs) also support a tradeoff between rate and reliability by allowing a transmissions modulation and coding to be changed. System performance can be improved by using rate adaptation and choosing the combination of these parameters with the highest rate and acceptable performance.

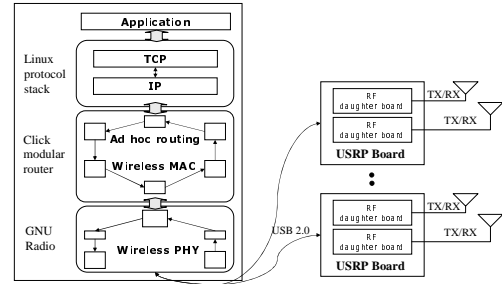


Fig. 1. Block diagram of a Hydra node.

We have developed a wireless node/network prototype, Hydra [1], [2], [3], which implements both a multi-rate, multi-antenna PHY and a complimentary Media Access Control (MAC) layer, in software. Software implementation allows us to easily experiment with sophisticated PHY and MAC algorithms and protocols and to closely couple these two layers, thus providing excellent support for studying cross-layer designs in the context of an operational wireless node. Hydra's PHY is essentially that of 802.11n, while the MAC is the 802.11 distributed coordination function (DCF) MAC with some enhancements, in particular for rate adaptation using both implicit and explicit feedback about the channel. Hydra currently supports the use of two antennas and thus can have two data streams when spatial multiplexing is used. The rates supported, 0.65, 1.3, 1.95, 2.6, 3.9, 5.2, 5.85, 6.5, 7.8, 10.4, 11.7 and 13.0 Mbps, are a factor of ten less than 802.11n due in large part of bandwidth limitations in our RF frontend. In addition to the ability to transmit over real channels, we have channel emulation capabilities that allow us to control many aspects of the channel that are hard to control in real world experiments. Taken together, Hydra's capabilities and flexibility makes it ideal for experimentation with rate adaptation in multi-antenna systems.

The focus of this paper is the presentation of a variety of rate control experiments conducted using Hydra, both over real and emulated channels. We consider single antenna systems and multi-antenna systems using both a single data stream and multiple data streams. We explore a number of algorithms. In particular, we study Auto Rate Fallback (ARF) [4], which is

based on implicit feedback and needs no information from the PHY and no modifications to the standard packet format. We also study an implementation of Receiver Based Auto Rate (RBAR) [5], a cross-layer protocol that uses explicit feedback and requires information from the PHY and packet modifications. We also use a feedback-based technique that, unlike RBAR, allows us to adapt between single and multiple data streams when using multiple antennas. We believe these are the first experimental results to be reported for adapting between using diversity and spatial multiplexing for 802.11-like systems. Our results show that rate adaptation can be effective for multi-antenna systems, but that adapting between a single stream taking advantage of diversity and several streams using spatial multiplexing is more complex than rate adaptation for single antenna systems.

We begin with Section I, which presents background and related work. Section II overviews the Hydra testbed and Section III discusses the experimental setup and in particular both the real and emulated channels used. The heart of the paper are the experimental results, which are divided between those based on real channels, in Section IV, and those based on emulated channels, in Section V. Section VI presents future work and brief conclusions.

## I. BACKGROUND

The goal of rate adaptation is to balance data transmission rate with the rate of packet failure. Choosing too low of a rate will cause transmissions to take longer than needed, while choosing too high a rate cause the transmission to fail. For the systems considered here, three factors determine the rate: how the data is modulated, the amount of redundancy in the error correcting code, and the degree of spatial multiplexing. If we fix the degree of spatial multiplexing, then for the systems under study here, the probability that a given modulation and coding rate will result in a successful transmission is determined primarily by the signal-to-noise ratio (SNR) at the receiver, which is in turn a function of the transmit power and the channel between the transmitter and the receiver. Changes in the channel mean that the best rate to use for transmission will change, and any approach to rate adaptation must track such changes to deliver the best rate.

Many rate adaptation protocols use implicit feedback to track the changing channel and adjust the rate. ARF [4] is typical of such algorithms. In ARF, ACKs are used as implicit feedback. If an ACK is not received, the current rate is assumed to be too high and the rate is lowered. If an ACK is received, the current rate is known to be equal to or lower than the ideal rate and if enough ACKs are received in a row, the protocol increases the rate to see if a higher rate is feasible. ARF raises the rate if 10 consecutive ACKs are received and lowers it if there are two consecutive ACK failures. Other protocols taking a similar approach include [6], [7].

Advantages of using ARF include that no information is needed from the PHY and packet formats are not impacted. However, ARF has the disadvantage that it can only adapt to changes in the channel that take place on time scales

greater than a few packet exchange times. Also, even for an unchanging channel, it occasionally raises the rate to see if a higher rate can be supported, resulting in packet loss. Finally, ARF assumes that failure to receive an ACK is due to the rate being too high, even though the failure may have some other cause, such as a collision.

Another class of protocols use explicit feedback to track the channel. A good example is RBAR [5], which uses the request-to-send (RTS) preceding the data transmission to measure the SNR. The receiver then determines the best rate based on this SNR and returns this rate to the transmitter, which uses it to transmit the data. Although there are a multitude of feedback algorithms, RBAR captures the basic architecture.

An advantage of RBAR is it can adapt to channels that change at a rate faster than tens of packet times and it does not need to probe when the channel does not change. One limitation is that it may not work well for channels that change so fast that the SNR for the RTS is different than for the packet. However, the main disadvantage is that it requires a potentially expensive RTS/CTS handshake to measure and feedback the channel related information, as well as additional space in the CTS for the feedback information, which impacts the packet format. Avoiding this overhead may cause the loss of the advantage of having timely feedback.

Unfortunately, the situation becomes more complex when there is a need to adapt between diversity and spatial multiplex modes in multi-antenna systems. Spatial multiplexing algorithms, only perform well in certain types of wireless channels with *good* spatial structure. In other words, increasing the number of data streams in a channel with a *bad* spatial structure will significantly reduce reliability and decrease throughput. The Demmel condition number is a good indicator of the spatial structure of MIMO channels [8]. Thus what rate (including the degree of spatial multiplexing) is best is a function of both the SNR and the Demmel condition number, both of which vary depending on the channel.

This creates a severe problem for techniques based on implicit feedback because it is no longer clear when and how to either increase or decrease the rate. For example, when ARF needs to increase the rate, should the protocol use a more aggressive modulation and coding or should it increase the level of spatial multiplexing? Similarly, which way to decrease the rate is unclear, should the degree of spatial multiplexing be decreased or should the modulation and coding be less aggressive. Even more confusing, on a loss, it might be possible to increase the chances of reliable communication while actually raising the rate by using a less aggressive coding and modulation but raising the degree of spatial multiplexing. We are not aware of any implicit feedback schemes that currently try to address this problem.

Fortunately, these issues can be addressed in systems that have explicit feedback. In particular, in Hydra we have implemented a system that uses the RTS to measure both the SNR and the Demmel condition number. The CTS is then used to return both the modulation/rate and the degree of spatial multiplexing to the transmitter. Details are found in Section II.

## II. THE HYDRA TESTBED

Our experiments are performed using the Hydra testbed. Hydra has been presented in some detail in [2] and an overview of a variety of PHY issues that arise when using Hydra (and when experimenting with PHYs, MACs, and cross-layer design) appear in [3]. Here we present a brief overview of Hydra, including the rate adaptation protocols supported.

Figure 1 shows a block diagram of a Hydra node. The programmable RF front-end is the Universal Software Radio Peripheral (USRP) [9], which interfaces to the general purpose host through a USB 2.0 connection. The diagram shows several USRPs with multiple antennas, but here we use a single USRP with two antennas. All other aspects of Hydra, the PHY, MAC, and higher layers, run on a general purpose processor (GPP) running Linux. The PHY is written in C++ using the GNU Radio framework [10]. The MAC is also written in C++, but using the Click programmable router framework [11]. Click also provides ad-hoc routing and interfaces to the standard Linux stack. Implementing the PHY and MAC in C++ using general frameworks greatly eases the task of creating working cross-layer prototypes.

The Hydra PHY implements a multi-input multi-output (MIMO) orthogonal frequency division multiplexing (OFDM) transceiver that is based on 802.11n [12]. Data is carried on 52 subcarriers that are modulated using BPSK, QPSK, 16-QAM, or 64-QAM. Forward error correction codes (punctured convolutional codes) are used with coding rates of 1/2, 2/3, 3/4, or 5/6. Our current  $2 \times 2$  multiple-antenna system is able to support both single-stream and double-stream transmissions. The Hydra PHY can thus support various rates through a combination of modulation scheme, code rate, and the number of data streams. In particular, the rates supported by the prototype are (in Mbps) 0.65, 1.3, 1.95, 2.6, 3.9, 5.2, 5.85, and 6.5 for single-stream mode and 1.3, 2.6, 3.9, 5.2, 7.8, 10.4, 11.7, and 13.0 for double-stream (spatial multiplexing) mode. Due to limitations of our current hardware, these rates are exactly 10 times less than those defined for 802.11n [12].

Hydra implements 802.11 DCF MAC [13] with some enhancements for rate adaptation algorithms, namely ARF and RBAR. Hydra implements ARF as described in Section I and raises rates after 10 consecutive ACKs and lowers it after 2 ACK failures. For RBAR, the RTS is used to measure the SNR and Demmel condition number by utilizing the channel estimation function of the training sequences in the PHY header. When not adapting between single stream and spatial multiplexing, the SNR compared to the cutoffs found experimentally in Subsection IV-A and the best rate is returned in the CTS.

If we wish to adapt between single stream and spatial multiplexing, we must take the Demmel condition number into account as well. Because of our hardware's bandwidth limitation, we observe a flat channel in practice, which means the average SNR per carrier is a good predictor of performance. Therefore, for adapting the spatial multiplexing level, Hydra implements a PHY-based rate adaption algorithm that

is based on a two-dimensional look-up table of the average SNR per subcarrier and the average Demmel condition number per subcarrier. The entries of the table were determined for different packet lengths and target packet error levels by simulation. For the experiments that involve adapting the degree of spatial multiplexing, the rate is looked up in the SNR-Demmel-condition-number table before being returned in the CTS.

Finally, implemented by software, Hydra's receiver PHY takes a significant amount of time to process a packet and that time is dependent on packet length. Thus, Hydra uses spacing between packets that are longer relative to the packet transmission times than for the 802.11n standard. Our goal is to provide insights about systems with more typical, hardware-based PHYs, and so we have processed the throughput results presented in Sections IV and V so that they reflect the inter-packet spacing defined in the IEEE 802.11n standard [12].

## III. EXPERIMENTAL SETUP

Since Hydra is implemented modularly in software and can be run on laptops, we can easily configure the system for different experiments. Many of the experiments described below are run over real wireless channels using the USRP RF frontend. However, it can be difficult to create a variety of channels, and achieving reproducibility in real channels is challenging. One advantage of Hydra's software implementation is that it is easy to create an emulator that processes the baseband output of the Hydra transmitter and then sends it to a Hydra receiver. Here, we discuss the setup for both our real and emulated channel experiments.

### A. Real channels

For our measurements over real channels, we use two laptops running the Hydra transceiver code. Each transceiver is connected to a USRP with daughter cards operating in the 2.4 GHz band. Two antennas are connected to daughter cards, which allows us to do multi-antenna experiments.

Most of our experiments are set up in a small lab space. We used directional antennas that allow us to create various kinds of spatial correlations and fading. Changing the angles of the antennas allows us to change the relative power of the LOS and non-LOS signals in the channel, which impacts the Demmel condition number. The distance between the nodes is such that given the power constraints of the USRPs, we can explore a range of SNRs that allow us to adapt over the full set of rates. We vary the average SNR in these experiments by changing the transmission power.

Many of our experiments are done with stationary antennas. For these experiments, we calibrated the channel to find angles for the antennas that have good Demmel condition numbers. Fixing the angles allows us to have relatively static spatial correlation. In Subsection IV-D we show measurements of the distribution of Demmel condition numbers.

To create real channels that vary significantly in the time and spatial domains, our current best solution involves an oscillating metal fan along with one antenna mounted on each

of the left and right sides of the fan. We present some results from these channels in Subsection V-A, but we continue to investigate how to create interesting, but controlled channels.

### B. Emulated channels

The difficulty in creating and controlling real channels means that channel emulation is also important. Our experimental setup for an emulated channel consists of three GPP machines, two acting as Hydra nodes and one running the emulator code.

The channel emulator allows us to experiment with a variety of channel models and impairments. Thus far, we have implemented a Jakes model [14] and Kronecker channel model [15] to create time-varying and spatially correlated channels.

For spatially uncorrelated time-varying multi-antenna channels, we used Jakes channel model, which is a model for rayleigh fading channel that includes Doppler effects. For the multi-antenna systems we generated independent identically distributed (i.i.d.) time-varying single antenna channels up to the number of multi-antenna channel elements. In addition, to control average SNR, we changed the noise power in the additive white Gaussian noise channel model.

One of our goals is to study the impact of the Demmel condition number on the performance of multi-antenna systems. Using the Kronecker channel model allows us to control the Demmel condition number but it is limited in that we can control the average, but not the instantaneous Demmel condition number. Our solution, instead of using Kronecker channel model, we manipulated the singular values of the time-varying multi-antenna channel generated by using Jakes model. Taking the singular value decomposition allows us to disassemble the channel into three matrixes which are a left and right unitary matrixes, and a diagonal matrix. The diagonal matrix consists of non-negative singular values. Manipulating the singular values allows us to target a Demmel condition number because the condition number is a function of the singular values. Then, two unitary matrixes and the modified diagonal matrix are reassembled to create the channel which has the target Demmel condition number.

### C. Experimental Details

Our experimental results follow. Unless otherwise stated, all of these results are use 1370 byte packets, which produce 1500 byte packets (a maximum size Ethernet frame) at the MAC layer. Unless noted, all real channels are based on stationary antennas as discussed above. We use the abbreviation ARF-W for results based on ARF with a RTS/CTS exchange and ARF-WO for results based on ARF without such an exchange. We have used the term *single antenna* when referring to results for a single antenna system, and *single stream* and *spatial multiplexing*, when referring to a multi-antenna system using diversity and a single data stream or spatial multiplexing and two data streams.

## IV. EXPERIMENTS ON REAL CHANNELS

The first group of experiments we present were conducted over real channels as described in Section III. The standard

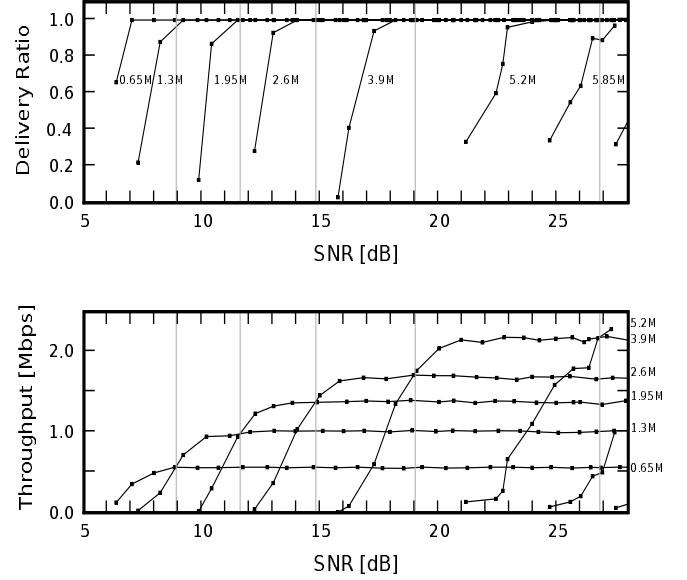


Fig. 2. Delivery Ratio and Throughput vs. SNR for single antenna fixed data rates. Dotted vertical lines indicate rate transition points for RBAR.

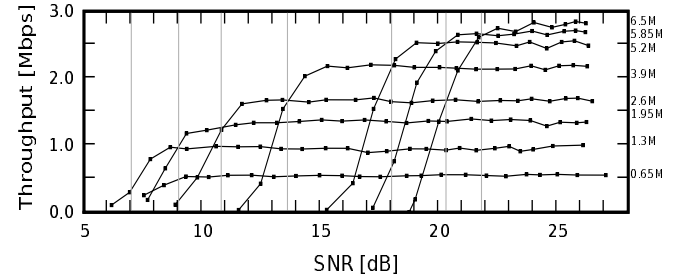


Fig. 3. Throughput vs. SNR for single stream fixed data rates. Dotted vertical lines indicate rate transition points for RBAR.

description of the delivery ratio and throughput graphs is found in Subsection IV-A. In general, the delivery ratio graphs provide little additional insight over the ones of throughput, so we have omitted all but the initial one.

### A. Calibrating the RBAR transition points

Any RBAR implementation will need to make a choice about when to transition from one rate to another. As with Holland [5], we use SNR as the criteria for channel quality. We choose the transition points by experimental calibration using the fixed rates supported by Hydra as a function of SNR. Separate calibration is needed for single antenna, single stream, and spatial multiplexing. Note, because it depends on the channel, we can not control the SNR directly. We can however, measure it and RBAR uses measurements of the RTS SNR to determine the rate. In these experiments, we control the transmit power and then measure the SNRs of the packets to find the required cutoffs.

Figures 2-4 show the experimental results we used to choose

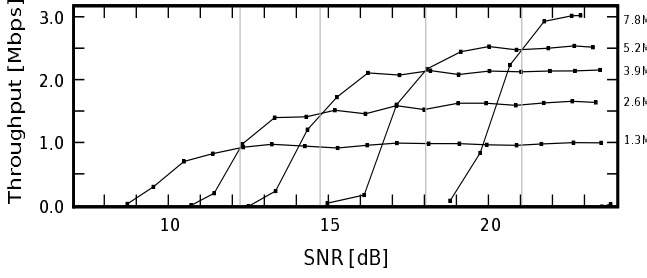


Fig. 4. Throughput vs. SNR for spatial multiplexing fixed data rates. Dotted vertical lines indicate rate transition points for RBAR.

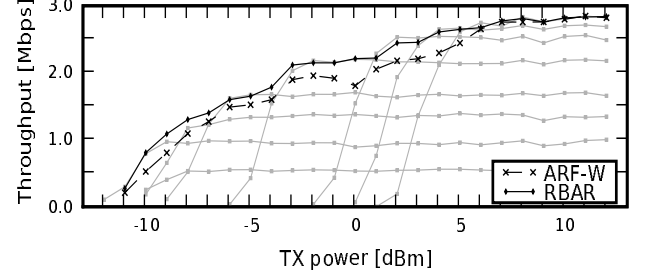


Fig. 6. Throughput vs. SNR for single stream RBAR and ARF-W. Fixed rate lines are shown lightly for reference.

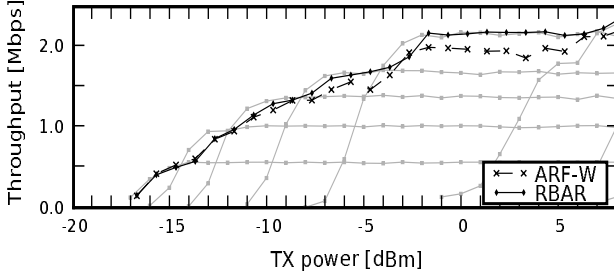


Fig. 5. Throughput vs. SNR for single antenna RBAR and ARF-W. Fixed rate lines are shown lightly for reference.

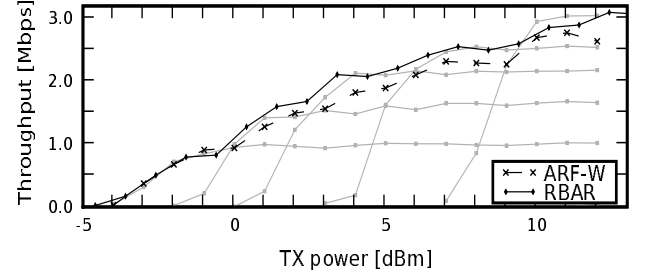


Fig. 7. Throughput vs. SNR for spatial multiplexing RBAR and ARF-W. Fixed rate lines are shown lightly for reference.

our transition points. All graphs have SNR on the X-axis. First two graphes show the results for all of the fixed rates supported by Hydra for that version of the system. On the other hand, for the spatial multiplexing experiment, the graph shows the results up to 7.8 Mbps. The channel used for these measurements was unable to support higher rates. For Figure 2 the Y-axis of the top graph shows the packet delivery ratio and of the bottom graph shows the throughput achieved in Mbps. The packet delivery ratio for the other measurements was essentially identical and so have not been presented. The vertical lines show the transition points we chose for RBAR. Note that single stream supports higher rates than single antenna and the cutoffs chosen are lower than for single antenna. This reflects the improvement we get in the channel from using diversity.

One design detail is what criteria to use for choosing the rate transitions. We choose to transition from one rate to another when doing so would result in higher throughput. Thus the transition points are located at the points where the throughput of a higher rate crosses that of the best performing lower rate. As shown by the vertical lines, the transitions we use are (in dB): for single antenna, 8.9, 11.6, 14.8, 19.0 and 26.8; for single stream, 7.0, 9.0, 10.8, 13.6, 18.0, 20.3 and 21.8; and for spatial multiplexing, 12.2, 14.7, 18.0 and 21.0. Note that the first cutoff for single stream is an estimate since the crossing point is not observed in the data.

Our choice of throughput as the criteria represents a tradeoff in terms of reliability. Since throughput is the product of the rate and the delivery ratio, choosing to transition to a higher

rate when the throughput is equal implies that the delivery ratio will go down at the transition. We can see this on the delivery ratio graph because the vertical lines intersect the curves at points where the delivery ratio is still increasing. One might wish to impose a reliability requirement as well and choose somewhat higher SNR transitions if it did not overly impact throughput.

### B. RBAR and ARF with RTS/CTS

The goal of our first rate adaptation experiment is to compare the performance of RBAR and ARF when both incur the overhead of the RTS/CTS exchange. Figures 5-7 show the experimental results comparing RBAR and ARF-W in the standard format for all system variants. The fixed rate data is shown lightly for reference. Note here, we plot the data in terms of transmit power. Since transmit power is what is actually controlled by the transmitter, these results reflect the conditions actual experienced by ARF and RBAR.

We see that RBAR tracks the top of the curves formed by the fixed rate data relatively closely, indicating that RBAR is generally choosing the best rate possible. ARF-W also tracks the fixed rate data reasonably well, but not as closely as RBAR. ARF has some dips at about the points the fix rate results cross. We believe at these points, ARF is sometime increasing its rate and incurring drops. We are investigating why ARF seems to consistently under perform in the spatial multiplexing case.

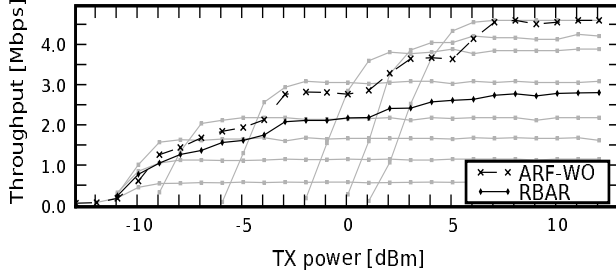


Fig. 8. Throughput vs. SNR for single stream RBAR and ARF-WO. Fixed rate lines are shown lightly for reference.

### C. RBAR and ARF-WO

The goal of this experiment is to study the impact of the RTS/CTS overhead on ARF. Figure 8 shows the experimental results comparing RBAR (using the same data as Figure 6) and ARF-WO in the standard format for the single stream system. The fixed rate data without RTS/CTS is shown for reference. The other variants show similar results.

Considering throughput, ARF-WO is almost always better than RBAR indicating that, for this channel, RTS/CTS overhead dominates gains achieved by better channel adaptivity. However, in general, ARF-WO often does not achieve the highest possible throughput, as shown by the fixed rate data. We suspect this is because ARF periodically probes for a higher rate resulting in drops. We also note that as the rate increases, the gap between ARF-WO and RBAR increases. This reflects the fact that at higher rate, the fixed duration transmit overheads begin to dominate over the decreasing time to actually transmit the data.

### D. Demmel Condition Number

Our previous experiments keep the degree of spatial multiplexing constant. The next two subsections introduce data that reflects on adapting between single stream and spatial multiplexing.

Figure 9 shows the distribution of the Demmel condition number for one channel with stationary antennas and Figure 10 when the antennas are fan mounted. The X-axis is the Demmel condition number and the Y-axis is the probability with which that Demmel condition number occurred. For the stationary antennas, the average Demmel condition number is 2.9, and for fan mounted, it is 5.0. Although the stationary case shows much less variation than the fan mounted case, we will see later (Subsection V-D) that there is still enough variation that sometimes spatial multiplexing may not work well. This may account for ARF's poorer performance for that system, since the resulting drops may cause ARF to reduce its rate, when the correct response would be not to use spatial multiplexing.

### E. Adapting Between Single Stream and Spatial Multiplexing

Figure 11 shows results for feedback-based adaptation between single stream and spatial multiplexing (as well as adaptation of modulation and coding). The figure, shows a

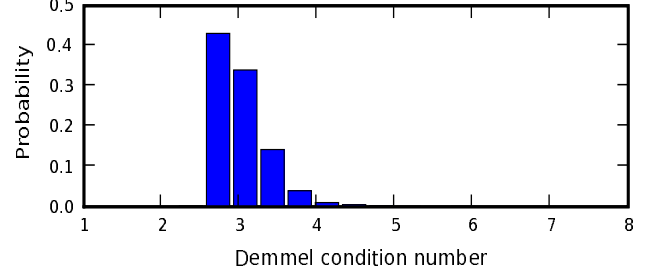


Fig. 9. Distribution of the Demmel condition number for stationary antennas

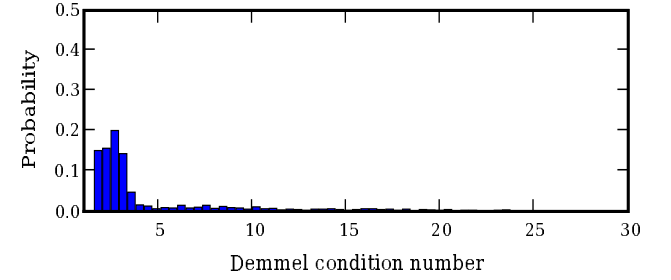


Fig. 10. Distribution of the Demmel condition number for two fan mounted antenna.

variety of data for a scenario in which approximately 50 packets (as shown on the X-axis) are transmitted over a channel formed when the two antenna are fan mounted. This is the same data that the distribution on Figure 10 was computed from. As shown by the legend, the figure shows when and at what rate (right X-axis) each packet was transmitted. It also shows whether the packet was transmitted using single stream or spatial multiplexing and whether the packet was transmitted successfully or not. It also shows the SNR at which each

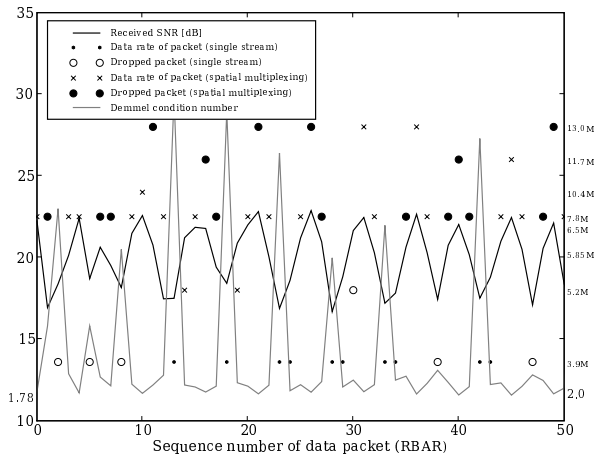


Fig. 11. Adapting between single stream and spatial multiplexing (Average SNR=19.9 dB, TX antennas fan mounted)

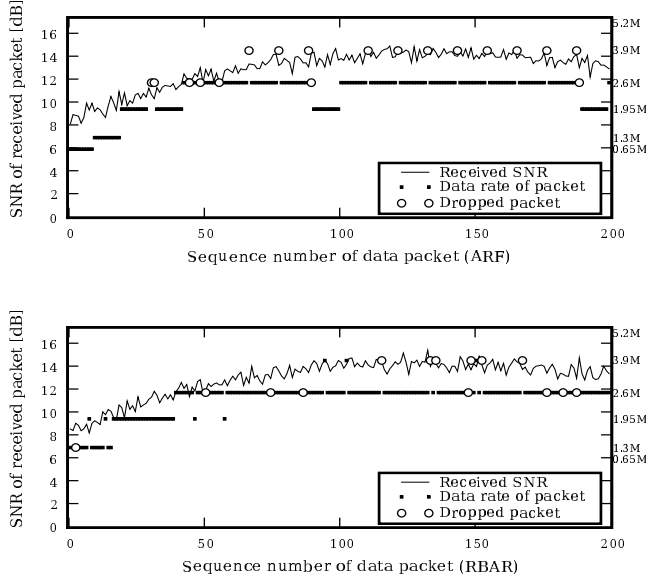


Fig. 12. MAC trace in a slow channel (Doppler=0.001) (Single stream)

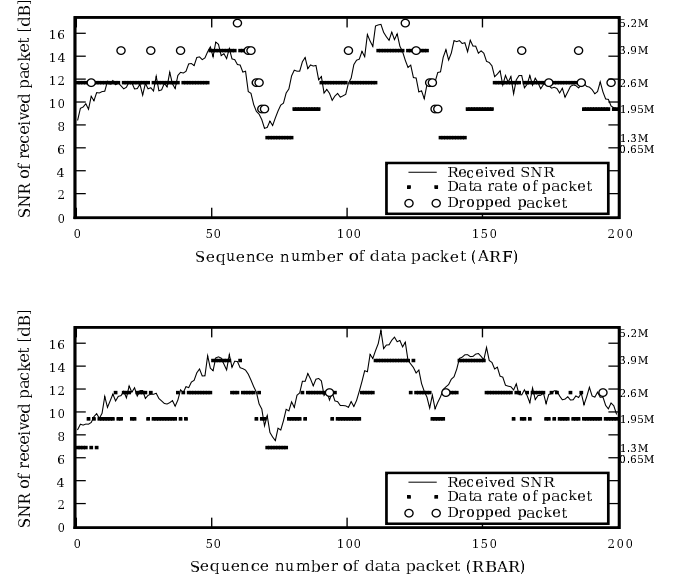


Fig. 13. MAC trace in a faster channel (Doppler=0.01) (Single stream)

packet was received (left Y-axis) and the Demmel condition number of the channel for the transmission. Note, the lowest Demmel condition number was about 1.8 and the highest 30.

We observe that the spikes in the Demmel condition number result in single stream transmissions, but that correlated drops in the SNR result in significant numbers of drops of those packets. Low Demmel condition numbers favor spatial multiplexing, but the drop pattern for these packets is less clear.

## V. EXPERIMENTS ON EMULATOR-BASED CHANNELS

Understanding the behavior of rate adaptation is aided by using reproducible channels with controllable time and spatial variations. Thus this section focuses on measurements using our channel emulator.

### A. Time varying channels

The critical difference between ARF and RBAR is how they behave as the channel varies in time. We show detailed traces of the behavior of the MAC for both RBAR and ARF in slow, faster, and fast channels. This allows us to confirm our understanding of how these algorithms work and to develop some intuitions about what detailed behavior to expect. For these MAC tracing plots, we set the TX power to -5 dBm which allowed us to see some rate variation in the results, and we sent 1370 Byte CBR traffic at a 1 sec interval. The traces each represent 200 packets. We used a Doppler shift of 0.001 for our slow channel, 0.01 for our faster channel, and 0.1 for our fast channel. These results are for single stream, but similar results were found for the other system types.

The top graph in Figure 12 shows a trace of the MAC behavior of ARF in the slowest channel (Doppler 0.001). The X-axis is the sequence number of the packet. The left Y-axis

shows the SNR of the received packet, while the right Y-axis shows the rate chosen for each packet. The line shows the received SNR for each packet and small dots show packets that were successfully received and larger open circle show packets that had errors.

This is the channel in which one would expect ARF to track the channel the best. We see that it does track the channel well, but we also see that when it probes the channel every tenth packet is likely to fail, which is as expected.

The bottom graph of Figure 12 shows a trace of the MAC behavior of RBAR in the slowest channel (Doppler 0.001) in the same format. We see that it tracks the channel as well as ARF, but without ARF's periodic failures. It does have an occasional failure when the channel becomes worse between sending the RTS and sending the data. From these results, we would expect that RBAR would outperform ARF in terms of packet delivery ratio and for throughput when ARF is used with RTS/CTS. However, ARF may outperform RBAR for throughput when it does not need RTS/CTS.

Figure 13 shows a trace of ARF on the top and of RBAR on the bottom both in a faster channel (Doppler 0.01) in the same format. As expected, ARF can no longer track the channel well and it takes some time to “ramp up” during periods when the channel is good. However, we see that drops usually occur when the channel gets worse and such fades are well tracked. Looking closely it is possible to see the double drops that trigger rate reduction. Also as expected, RBAR tracks the channel well, with failures coming chiefly when the channel is falling, making it more likely that a data packet experiences a worse channel than its RTS. For this case, it seems likely that RBAR will outperform all versions of ARF for all metrics.

Finally, Figure 14 shows a trace of ARF on the top and RBAR on the bottom both in a fast channel (Doppler 0.1)

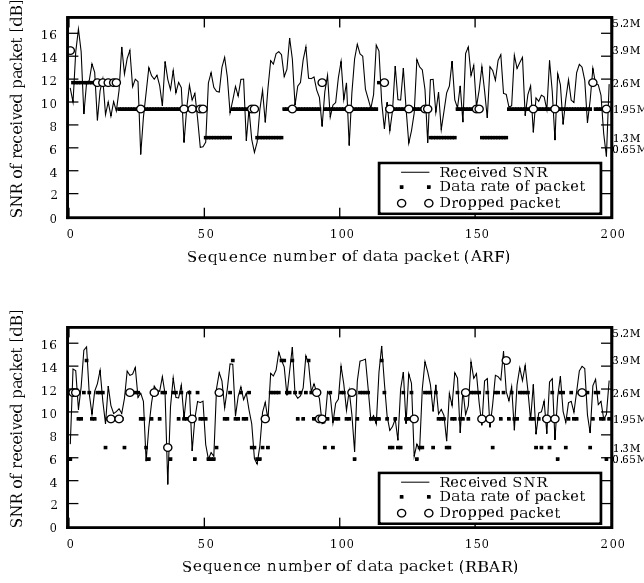


Fig. 14. MAC trace in a fast channel (Doppler=0.1) (Single stream)

in the same format. ARF tends to track the bottom of the envelope formed by the fast varying channel. In some case, probes succeed and for some period the rate increases, but in other cases they fail and the rate remains low. RBAR shows more complex behavior as it attempts to track the fast changing channel. When the channel goes up between the RTS and the data, the packet is successful, although perhaps at a lower rate than could have been achieved. On the other hand, when the channel goes down the data packet is sent at too high a rate and fails. Here, we would expect ARF to perform significantly better in terms of delivery ratio, but it is harder to predict the results for throughput.

#### B. Throughput as a Function of Normalized Doppler

In addition to gathering the MAC traces, we also measured the throughput as a function of Doppler shift, as shown in Table I. Here we compare RBAR to ARF-WO so that RBAR is penalized for the cost of the RTS/CTS exchange. As we might expect from the results of the previous section, ARF-WO outperforms RBAR at the slow and fastest Doppler shifts. However, RBAR outperforms ARF-WO slightly at the middle Doppler, despite the significant overhead penalty. This suggests that for some channels, the other benefits of the RTS/CTS exchange, such as reduced hidden nodes, can be gained with no loss in raw performance.

TABLE I  
AVERAGE THROUGHPUT VS. NORMALIZED DOPPLER

	0.001	0.01	0.1
RBAR	0.86 Mbps	1.32 Mbps	1.15 Mbps
ARF-WO	1.25 Mbps	1.29 Mbps	1.43 Mbps

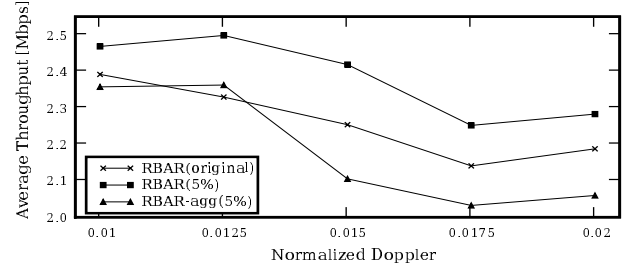


Fig. 15. Effect of Reducing Rate on Retransmission

#### C. Adapting Rate as a Function of Retransmission Count

In Subsection IV-C we observed that the impact of transmission overheads increases significantly as the rate increases. This is to be expected since at higher rates much of the overhead is of fixed duration, but the data itself takes less time to transmit.

When there is a transmission failure, the 802.11 MAC exponentially backs off and then retransmits up to 7 times. With each retransmission, the fixed penalty has a component that is growing exponentially. The result is that the performance gap between the rates decreases. Since transmitting at a lower rate is more reliable, this suggests a strategy in which on retransmissions a lower rate than that predicted by the base rate control algorithm is chosen so as to avoid possible (more expensive) retransmissions arising from the higher rate.

We are just beginning to explore this idea, but we present some preliminary results that are promising. Here, we use RBAR as our base rate control protocol, but implicit methods could also be used. When a retransmission is needed, we first get the rate predicted by the receiver from the CTS. Then we check to see if there is a lower rate that would have no greater than a 5% performance penalty if it were used, given the current state of the backoff window and the other fixed overheads. If so, we use the lower rate.

Figure 15 shows the result of our preliminary experiments with this idea. Here the emulated channel has an average SNR of 21.5 dB and we emulated a single stream system. The X-axis shows the normalized Doppler, while the Y-axis shows the throughput in Mbps. We measured three protocols, standard RBAR, RBAR with the rate lowering base on the 5% threshold, and RBAR with the 5% threshold, but using more aggressive cutoffs for the RBAR rate calculation.

From the figure, we see that the results from the new approach using the original rate prediction are modestly better than always retransmitting at the predicted rate. However, using more aggressive cutoffs did not result in better performance in general. We had hoped that it would, with the idea being that the initial transmission would be more aggressive and then retransmissions less so. Still, we believe these results show that the idea of coupling rate adaptation to the retransmission strategy is promising and we expect to pursue this idea in the future.



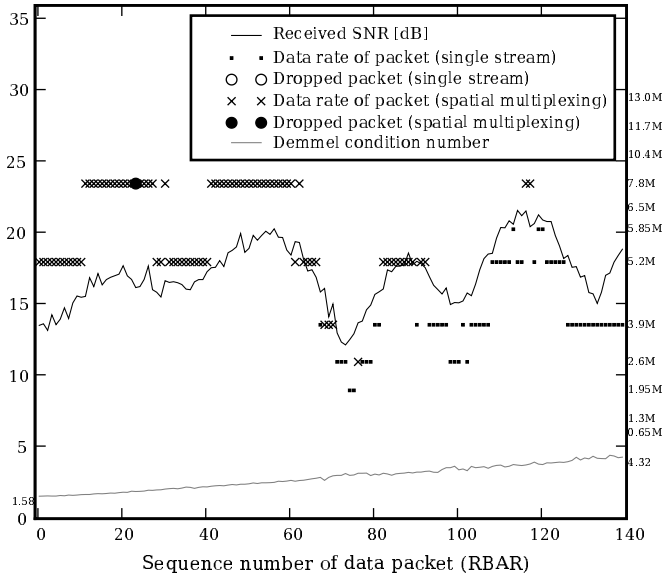


Fig. 16. Adaptation between spatial multiplexing and single stream rates using the channel emulator

#### D. Adaptation over Varying Demmel condition Number

Our final experiment presents the results of adapting between single stream and spatial multiplexing as we vary the Demmel condition number (and channel quality) in the emulator. In this case we controlled the Demmel condition number by manipulating the singular values of the channel matrix.

Figure 16 shows a MAC trace of the transmission of 140 packets. The format is the same as for the previous MAC traces (Figures 12-14) except for the addition of the Demmel condition number, which varies from 1.58 to 4.32. This trace was taken at a relatively high SNR (SNR=17.2 dB) and for a relatively slow channel (Normalized Doppler=0.01). Results for other SNRs and channel speeds are similar.

The key observation is that when the Demmel condition number is low, spatial multiplexing is effective and we are able to track the channel and achieve the higher rate promised by spatial multiplexing. However, when the Demmel condition number crosses about 3.5 the systems switches and especially for lower SNRs must use single stream. We note that our Demmel condition number distributions even for stationary antenna suggests that it may be common for this to be true.

#### VI. FUTURE WORK AND CONCLUSIONS

Possible future work falls broadly into two categories, experimental work and algorithm development and validation. Experimentally, probably the most important avenues to explore next involve varying the kinds of real channels we can experiment with. Ideally, we would be able to create reasonably reproducible time varying channels with deeper fades than those we explored here. Ideally we would like to be able to manipulate the Demmel condition number as well, but that seems difficult. Also, here we have not explored any of the issues that arise when the channel become frequency selective,

in which case, for example, average SNR will become a less reliable predictor of the best rate. Finally, we are working on support for more antennas, which will allow us to experiment more freely with the degree of spatial multiplexing.

Algorithmically, there is also significant work to do. We have developed an algorithm, Transmitter Base Autorate (TBAR), that uses channel reciprocity to measure the channel, without the need for feedback. Space limitations have prevented us from presenting it here. We are also working on understanding the relation between rate adaptation and retransmit policy.

In conclusion, we have used the Hydra testbed, which provides a flexible software platform to implement a wide variety of cross-layer protocols, to experiment with rate adaptation for multi-antenna systems. These experiments have been performed using realistic PHYs and MACs and over both actual and emulated channels. The results show that rate adaptation is feasible and effective for such systems, but also that it is more challenging than for single antenna systems.

#### REFERENCES

- [1] "Hydra – A Wireless Multihop Testbed." [Online]. Available: <http://hydra.ece.utexas.edu/>
- [2] K. Mandke, S.-H. Choi, G. Kim, R. Grant, R. C. Daniels, W. Kim, S. M. Nettles, and R. W. Heath, Jr., "Early Results on Hydra: A Flexible MAC/PHY Multihop Testbed," in *Proceedings of the 65th IEEE Vehicular Technology Conference*, Apr. 2007, pp. 1896–1900.
- [3] K. Mandke, R. C. Daniels, S.-H. Choi, S. M. Nettles, and R. W. Heath, Jr., "Physical Concerns for Cross-Layer Prototyping and Wireless Network Experimentation," in *The Second ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization*, Montreal, QC, Canada, Sep. 2007.
- [4] 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, 1997.
- [5] G. Holland, N. H. Vaidya, and P. Bahl, "A Rate-adaptive MAC Protocol for Multi-Hop Wireless Networks," in *Mobile Computing and Networking*, 2001, pp. 236–251. [Online]. Available: [citeseer.ist.psu.edu/holland01rateadaptive.html](http://citeseer.ist.psu.edu/holland01rateadaptive.html)
- [6] M. Lacage, M. H. Manshaei, and T. Turletti, "IEEE 802.11 Rate Adaptation: A Practical Approach," in *Proceedings of the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'04)*, Venice, Italy, Oct. 2004, pp. 126–134.
- [7] J. C. Bicket, "Bit-rate Selection in Wireless Networks," Master's thesis, MIT, Feb. 2005.
- [8] R. W. Heath, Jr. and A. J. Paulraj, "Switching Between Diversity and Multiplexing in MIMO Systems," *IEEE Transactions on Communications*, vol. 53, no. 6, pp. 962–968, Jun. 2005.
- [9] "GNU Radio: Universal Software Radio Peripheral." [Online]. Available: <http://www.comsec.com/wiki/UniversalSoftwareRadioPeripheral>
- [10] "GNU Software Radio." [Online]. Available: <http://www.gnu.org/software/gnuradio/>
- [11] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The Click Modular Router," *ACM Transactions on Computer Systems*, vol. 18, no. 3, pp. 263–297, 2000. [Online]. Available: [citeseer.ist.psu.edu/article/kohler00click.html](http://citeseer.ist.psu.edu/article/kohler00click.html)
- [12] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification-Draft 5.0: Enhancements for Higher Throughput, Part 11 Standard ed.*, IEEE 802.11n Working Group, 2008.
- [13] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE 802.11 Working Group, Piscataway, NJ, 1997.
- [14] W. C. Jakes, *Microwave Mobile Communications*. Wiley, 1974.
- [15] J. P. Kermoal, L. Schumacher, K. I. Pedersen, P. E. Mogensen, and F. Frederiksen, "A Stochastic MIMO Radio Channel Model with Experimental Validation," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 6, pp. 1211–1226, Aug. 2002.