A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks *

Gavin Holland Nitin Vaidya
Dept. of Computer Science
Texas A&M University
College Station, TX 77843-3112
{gholland,vaidya}@cs.tamu.edu

Paramvir Bahl
Microsoft Research
One Microsoft Way
Redmond, WA 98052-6399
bahl@microsoft.com

ABSTRACT

Wireless local area networks (W-LANs) have become increasingly popular due to the recent availability of affordable devices that are capable of communicating at high data rates. These high rates are possible, in part, through new modulation schemes that are optimized for the channel conditions bringing about a dramatic increase in bandwidth efficiency. Since the choice of which modulation scheme to use depends on the current state of the transmission channel, newer wireless devices often support multiple modulation schemes, and hence multiple data rates, with mechanisms to switch between them. Users are given the option to either select an operational data rate manually or to let the device automatically choose the appropriate modulation scheme (data rate) to match the prevailing conditions. Automatic rate selection protocols have been studied for cellular networks but there have been relatively few proposals for W-LANs. In this paper we present a rate adaptive MAC protocol called the Receiver-Based AutoRate (RBAR) protocol. The novelty of RBAR is that its rate adaptation mechanism is in the receiver instead of in the sender. This is in contrast to existing schemes in devices like the WaveLAN II [15]. We show that RBAR is better because it results in a more efficient channel quality estimation which is then reflected in a higher overall throughput. Our protocol is based on the RTS/CTS mechanism and consequently it can be incorporated into many medium access control protocols including the widely popular IEEE 802.11 protocol. Simulation results of an implementation of RBAR inside IEEE 802.11 show that RBAR performs consistently well.

1. INTRODUCTION

Wireless local area networks are becoming increasingly popular. This is due to the ratification of standards, like IEEE 802.11 [12], that have laid the foundation for off-the-shelf wireless devices capable of transmitting at high data rates. For example, devices are now available that can transmit at 11Mbps, with 54Mbps expected in the near future.

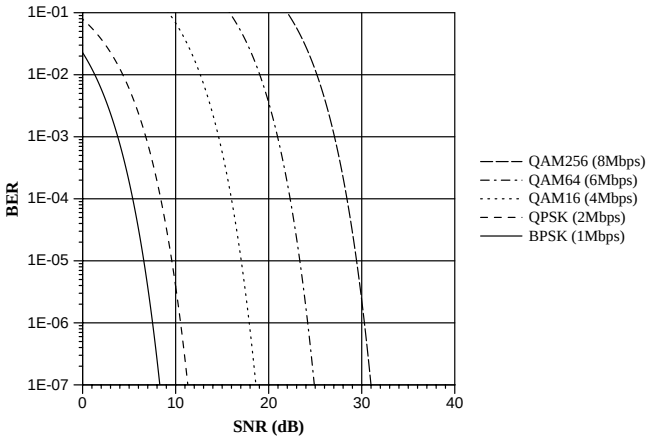


Figure 1: Theoretical bit error rates (BER) as a function of the signal-to-noise ratio (SNR) for several modulation schemes and data rates.

Higher data rates are commonly achieved by more efficient modulation schemes. *Modulation* is the process of translating an outgoing data stream into a form suitable for transmission on the physical medium. For digital modulation, this involves translating the data stream into a sequence of *symbols*. Each symbol may encode a certain number of bits, the number depending on the modulation scheme. The symbol sequence is then transmitted at a certain rate, the *symbol rate*, so for a given symbol rate, the data rate is determined by the number of encoded bits per symbol.

The performance of a modulation scheme is measured by its ability to preserve the accuracy of the encoded data. In mobile wireless networks, path loss, fading, and interference cause variations in the received signal-to-noise ratio (SNR). Such variations also cause variations in the bit error rate (BER), because the lower the SNR, the more difficult it is for the modulation scheme to decode the received signal. Since high rate schemes typically use denser modulation encodings, a tradeoff generally emerges between data rate and

*This research was supported in part by a grant from the National Science Foundation and a gift from Microsoft Research.

ACM SIGMOBILE 7/01 Rome, Italy

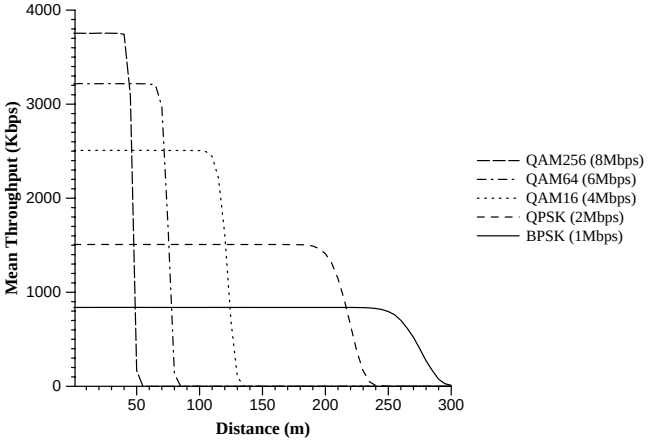


Figure 2: Comparison of throughput versus distance for several modulation schemes. Data was obtained by the simulation of two nodes at fixed positions, with one sending a continuous stream of UDP packets to the other. The propagation model was the log-distance path loss model, with a path loss exponent typical of an urban environment. Transmit power was constant.

BER: the higher the data rate, the higher the BER. This tradeoff is illustrated in Figure 1, which shows the theoretical BER as a function of the SNR for several different modulation schemes. Notice that for each modulation scheme the BER decreases with increasing SNR. Also notice that for a given SNR, an increase in data rate results in an increase in BER. For example, given an SNR of 10dB, a packet transmitted at 4Mbps using QAM16 modulation could experience a BER of 0.07, in comparison to 4×10^{-6} for the same packet transmitted at 2Mbps using QPSK modulation.

To illustrate the impact that this tradeoff can have on performance, Figure 2 shows throughput as a function of distance for each of the modulation schemes shown in Figure 1. For the sake of this illustration only large-scale path loss was modeled (in comparison to the rest of our results, in which we modeled Rayleigh fading) [20]. Notice that the lower rate schemes have greater transmission ranges than the higher rate schemes. As the distance increases, the signal attenuates until the received SNR drops below the threshold required to maintain a tolerable bit error rate. This appears as a sharp drop in throughput in Figure 2, corresponding to the steep curve in Figure 1. Of course, factors other than path loss contribute to variations in the SNR, such as fading and interference, which further impact performance.

Many existing wireless local area networking devices are designed with the capability of transmitting at multiple data rates. Examples of such devices include WaveLAN II from Lucent [15], and PC4800 from Aironet [1].

1.1 Rate Adaptation

Rate adaptation is the process of dynamically switching data rates to match the channel conditions, with the goal of selecting the rate that will give the optimum throughput for the given channel conditions. A proven technique for wireline modems [6], rate adaptation has attracted attention as a technique for use in wireless systems as well (e.g., [18], [3], [11], [24], [2]). In fact, the Lucent WaveLAN II and Aironet PC4800 devices also contain proprietary rate adaptation mechanisms.

There are two aspects to rate adaptation: channel quality estimation and rate selection. Channel quality estimation involves measuring the time-varying state of the wireless channel for the purpose of generating predictions of future quality. Issues include: which metrics should be used as indicators of channel quality (e.g., signal-to-noise ratio, signal strength, symbol error rate, bit error rate), which predictors should be used, whether predictions should be short-term or long-term, etc. [3], [10]. Rate selection involves using the channel quality predictions to select an appropriate rate. Techniques vary, but a common technique is threshold selection, where the value of an indicator is compared against a list of threshold values representing boundaries between the data rates [22], [3].

Among the factors that influence the effectiveness of rate adaptation, of particular importance is the accuracy of the channel quality estimates. Clearly, inaccurate estimates will result in poor rate selection. Thus, it is advantageous to utilize the best information available when generating an estimate, and since it is the channel quality seen by the *receiver* that determines whether a packet can be received, the best information is available on the receiver. It is equally important that once estimates are generated they be used before they become outdated. Thus, it is also advantageous to minimize the delay between the time of the estimate and the time the packet is transmitted.

Few rate adaptation techniques have been designed for wireless local area networks (e.g., mobile nodes communicating peer-to-peer over CSMA/CA links). Among those that are available, the following are most relevant. In [19], the authors present a protocol for a dual-channel slotted-aloha MAC, in which a separate control channel is used by the receiver to transmit explicit feedback to the sender, which the sender uses to adapt the rate. In [15], the authors present the "Auto Rate Fallback (ARF)" protocol for IEEE 802.11, used in Lucent's WaveLAN II devices. In ARF, the sender selects the best rate based on information from previous data packet transmissions, incrementally increasing or decreasing the rate after a number of consecutive successes or losses, respectively. Finally, in [9], the authors propose protocol for point-to-point links, that selects transmission settings (e.g. code rate and power level) based on cached per-link information. The settings are stored in separate *transmit* and *receive* tables, which are then used by the sender and receiver to transmit and receive data packets on the link. The tables are maintained, in part, by exchanging settings in control packets, such as those in the RTS/CTS protocol. The RTS/CTS protocol is a common MAC protocol for wireless local area networks (e.g., SRMA [21], MACA [16], MACAW [4], FAMA [8], IEEE 802.11 [12]). The purpose of the protocol is twofold: 1) to coordinate the transfer of the data packet between the sender and receiver, and 2) to announce the duration of the packet transfer to nodes that are in range of the sender and the receiver. The latter is important in multi-hop networks because of the potential for

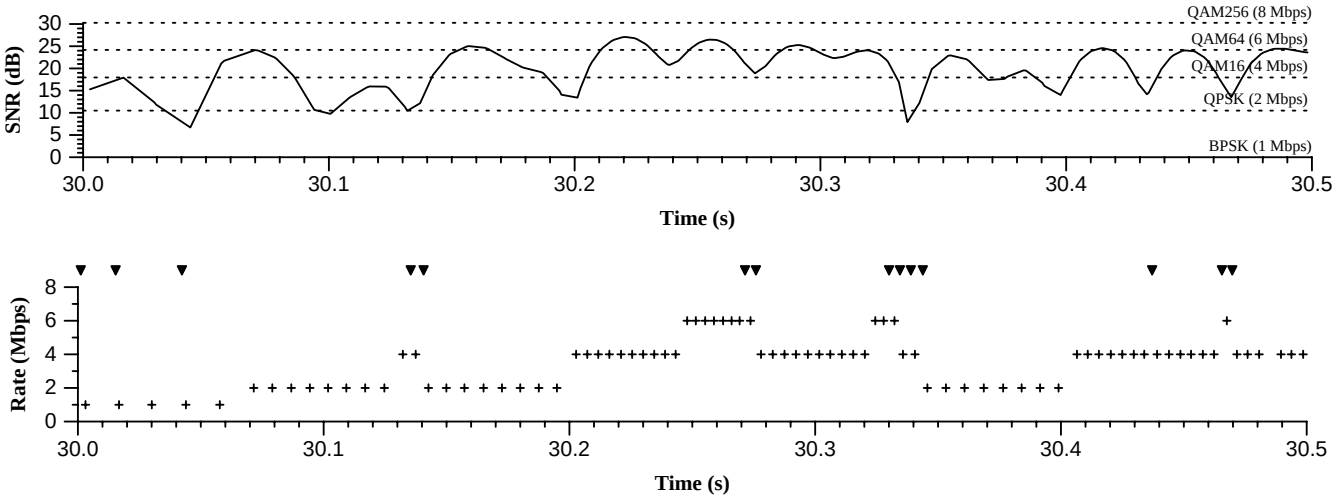


Figure 3: Performance of ARF for a single CBR connection between two nodes in a Rayleigh fading channel. Here, the sender is fixed and the receiver is moving at a speed of 2 m/s away from the sender. The lower graph shows the time at which packets were transmitted, and the modulation rate chosen by ARF for each packet. The tick marks along the top show the time at which packets were dropped by the receiver due to errors. The upper graph shows the SNR at the receiver for the packets in the lower graph. Also shown are thresholds representing the SNR values above which the next higher modulation rate has a theoretical mean $BER \leq 10^{-6}$. At the start, both nodes were at the same location, so the leftmost edges represent the point in time at which the two nodes were 60m apart.

collisions caused by *hidden terminals*. Hidden terminals are nodes that are in range of the receiver but not the sender. Collisions occur when hidden terminals, unable to sense the sender's transmission, attempt to transmit simultaneously, causing a collision at the receiver. The RTS/CTS packets reduce the probability of such collisions by announcing the packet transfer to potentially interfering nodes, who, in turn, react by deferring their own transmissions for the duration of the transfer. Thus, the RTS/CTS protocol provides *virtual* carrier sensing to supplement the *physical* carrier sensing of the devices. The protocol is simple. Prior to transmitting a data packet, the sender transmits a small RTS (*Ready to Send*) control packet to the receiver. If the receiver is capable of receiving the packet, it replies with a CTS (*Clear to Send*) control packet. The sender responds to the RTS by transmitting the data packet. Nodes that overhear either the RTS or CTS then defer their own transmissions for the duration of the packet transmission. In [9], the RTS is also used by the sender to tell the receiver what settings the sender will use to transmit the data packet (which it gets from its *transmit* table). The receiver uses the settings in the RTS to update its *receive* table. If the receiver chooses, it may use the CTS to suggest a different power level, but, otherwise, no other changes to the transmit settings are made during the RTS/CTS exchange. Instead, changes to the sender's *transmit* table are made by information in acknowledgment (ACK) or negative acknowledgment (NACK) packets sent at the end of the data packet transmission. These changes are then used for subsequent packet transmissions.

Note that, in all three protocols ([19], [15], and [9]), rate selection is performed by the sender, and, in [15], channel quality estimation is also performed by the sender. Also note that only [15] is based on a widely used wireless local area networking standard (IEEE 802.11).

Much of the other work on rate adaptation in wireless networks has assumed a cellular network (e.g., mobile nodes communicating to a base station over a TDMA link) [3], [18], [22]. We have observed that many of these techniques have the following characteristics: (a) channel quality estimation is performed by the receiver and periodically fed to the sender either on the same channel or on a separate subchannel; (b) rate selection is performed by the sender using the feedback provided by the receiver; and (c) they often reside at the physical layer, adapting rates on a symbol-by-symbol or slot-by-slot basis, transparent to upper layers. Although it may appear that such approaches are also applicable to wireless local area networks, several important differences exist. For instance, conventional local area networks generally use half-duplex radios on single RF channels, making simultaneous subchannel feedback impossible. Furthermore, conventional local area networks use distributed, contention-based medium access control protocols that require accurate estimates of packet transmission times for efficient operation (e.g. RTS/CTS). Thus, if transparent physical layer rate adaptation were to be employed, it would be difficult for the MAC layer to acquire accurate transmission time estimates, causing a decrease in efficiency.

1.2 Motivation

In this paper we propose a new approach to rate adaptation for wireless local area networks. In our approach, the rate selection and channel quality estimation are located on the *receiver*, and rate selection is performed on a per-packet basis *during* the RTS/CTS exchange, just prior to packet transmission. The motivation for our approach is based on the following observations:

- Rate selection can be improved by providing more timely and more complete channel quality information.
- Channel quality information is best acquired at the receiver.
- Transmitting channel quality information to the sender can be costly, both in terms of the resources consumed in transmitting the quantity of information needed as well as the potential loss in timeliness of the information due to transmission delays.

To emphasize the need for better rate adaptation mechanisms, consider Figure 3, which illustrates the behavior of the ARF protocol. Shown is the packet activity (shown in the lower graph) over a period of 500ms for a CBR connection between two nodes in a Rayleigh fading channel (shown in the upper graph). Here, the sender was held fixed while the receiver moved away at a speed of 2 m/s. At the start both nodes were at the same location, so the leftmost edges of the graphs represent the point at which the nodes were exactly 60m apart. From Figure 3, it is clear that ARF is slow to adapt to changes in SNR, as evidenced by the relative dissimilarity between the upper and lower graphs. In particular, consider its failure to rapidly increase the data rate after the deep fades at the 30.2s and 30.35s marks, and the attempt it makes to increase the rate in the middle of a fade at the 30.13s mark.

1.3 Paper Organization

The remainder of this paper is organized as follows. We start in Section 2 by giving some background material on the IEEE 802.11 standard. The proposed protocol is described in Section 3, followed in Section 4 by a detailed description of how it might be incorporated into 802.11. Section 5 presents the simulation environment used to generate the performance results in Section 6, which is followed by future work in Section 7. Finally, we summarize and conclude the paper in Section 8.

2. OVERVIEW OF IEEE 802.11

In this section, we briefly present relevant details of the features and operation of the IEEE 802.11 MAC. Readers familiar with 802.11 can skip this section without loss of continuity. Readers desiring more information on 802.11 are referred to [12].

2.1 Distributed Coordination Function (DCF)

The Distributed Coordination Function (DCF) in 802.11 is an implementation of the RTS/CTS protocol, and is illustrated in Figure 4, which is a time-line portraying the sequence of events that occur for a single packet transfer. Here, the source Src has a data packet to transmit to the destination Dst with a duration of length L. Node A is in range of Src but not Dst, and node B is in range of Dst but not Src. The protocol proceeds as follows. When Src has a packet to send, it calculates the length of time it will take to transmit the data packet at the current data rate, and then adds to that the transmission time of the CTS and ACK packets, which forms the duration of the reservation (D_{RTS}). The Src then transmits D_{RTS} in the RTS to Dst, using one of the rates in the *basic rate set*. The *basic rate set* is the set of rates that all nodes are required to support, which ensures that all nodes that are in transmission range are able to receive and demodulate the RTS/CTS packets. Since node A is in range of Src, it overhears the RTS and summarily defers its own transmissions for the duration of the reservation in the RTS (D_{RTS}), starting from the moment that it received the RTS (T_1). If Dst is capable of receiving the data packet, it responds by transmitting a CTS packet back to Src containing the time remaining in the reservation (D_{CTS}), which it calculates by subtracting the transmission time of the CTS from D_{RTS}. Node B, overhearing the RTS, learns of the requested reservation and, like A, defers for length D_{CTS}. At this point, transmission of the data packet and subsequent ACK can now proceed without interference from A or B. However, in the off-chance that A did not receive the RTS, due to, for example, an RTS collision caused by another node, the data packet also carries the time remaining in the reservation D_{DATA} to ensure that A defers during the transmission of the ACK.

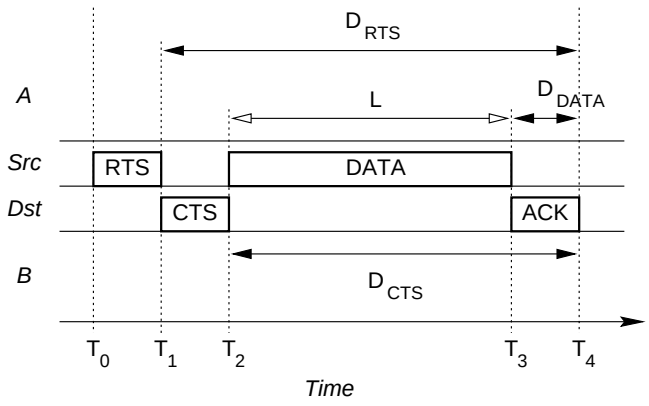


Figure 4: Timeline showing the RTS/CTS protocol in the IEEE 802.11 Distributed Coordination Function (DCF) for transmitting a data packet. Here, A and B are nodes that are in range of the transmitter and receiver, respectively. D_{RTS}, D_{CTS}, and D_{DATA} are the lengths of the reservations given in the RTS, CTS and DATA packets, and L is the duration of the data packet transmission.

2.2 Network Allocation Vector (NAV)

Since a node may overhear many different, potentially overlapping, reservation requests, it needs a means by which it can efficiently manage them. This is the purpose for the maintenance of a structure called the *Network Allocation Vector* (NAV) [12]. The NAV is a data structure that stores the aggregate duration of time that the medium is presumed to be "busy," based on the reservation requests that have been received. Maintenance of the NAV is straightforward, since reservations are not allowed to change. Thus, nodes that overhear a reservation request are free to update their NAVs without regard to any further communication, such as if the reservation was actually granted by the receiver.

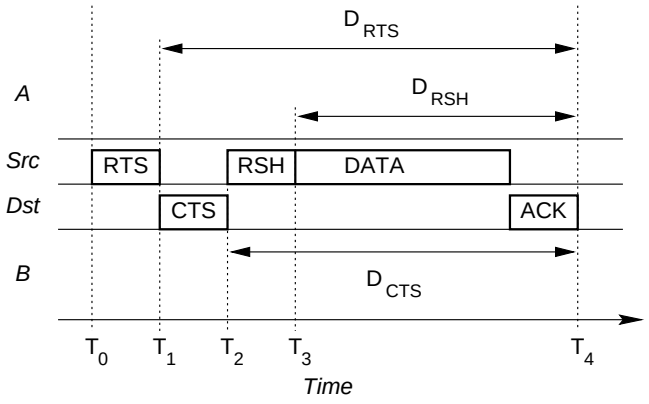


Figure 5: Timeline showing changes to the DCF protocol as needed for the proposed Receiver-Based Autorate Protocol.

3. THE PROPOSED RECEIVER-BASED AUTORATE (RBAR) PROTOCOL

The core idea of RBAR is to allow the *receiver* to select the appropriate rate for the data packet *during* the RTS/CTS packet exchange. Advantages to this approach include:

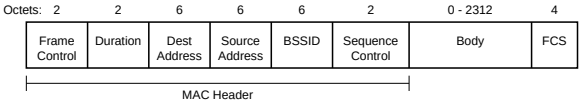
- Both channel quality estimation and rate selection mechanisms are now on the receiver. This allows the channel quality estimation mechanism to directly access all of the information made available to it by the receiving hardware (such as the number of multipath components, the symbol error rate, the received signal strength, etc.), for more accurate rate selection.
- Since the rate selection is done *during* the RTS/CTS exchange, the channel quality estimates are nearer to the actual transmission time of the data packet than in existing sender-based approaches, such as the protocol in [15] which attempts to predict channel conditions based on conditions experienced during previous data packet transmissions.
- It can be implemented into IEEE 802.11 with minor changes, as we will show in a later section.

In the remainder of this section, we present the RBAR protocol in more detail. Note that although our discussion is in the context of the RTS/CTS protocol in the DCF of the 802.11 standard, the concepts are equally applicable to other RTS/CTS based protocols such as SRMA [21], MACA [16], MACAW [4], and FAMA [8].

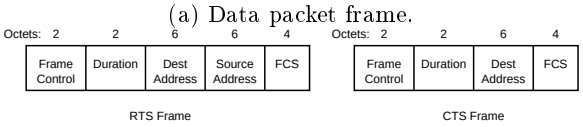
In RBAR, instead of carrying the duration of the reservation, the packets carry the modulation rate and size of the data packet. This modification serves the dual purpose of providing a mechanism by which the receiver can communicate the chosen rate to the sender, while still providing neighboring nodes with enough information to calculate the duration of the requested reservation. The protocol is as follows.

Referring to Figure 5, the sender Src chooses a data rate based on some heuristic (such as the most recent rate that was successful for transmission to the destination Dst), and then stores the rate and the size of the data packet into the RTS. Node A, overhearing the RTS, calculates the duration of the requested reservation D_{RTS} using the rate and packet size carried in the RTS. This is possible because all of the information required to calculate D_{RTS} is known to A. A then updates its NAV to reflect the reservation. While receiving the RTS, the receiver Dst uses information available to it about the channel conditions to generate an estimate of the conditions for the impending data packet transmission. Dst then selects the appropriate rate based on that estimate, and transmits it and the packet size in the CTS back to the sender. Node B, overhearing the CTS, calculates the duration of the reservation D_{CTS} similar to the procedure used by A, and then updates ts NAV to reflect the reservation. Finally, Src responds to the receipt of the CTS by transmitting the data packet at the rate chosen by Dst.

In the instance that the rates chosen by the sender and receiver are different, then the reservation D_{RTS} calculated by A will no longer be valid. Thus, we refer to D_{RTS} as a *tentative* reservation. A tentative reservation serves only to inform neighboring nodes that a reservation has been requested but that the duration of the *final* reservation may differ. Any node that receives a tentative reservation is required to treat it the same as a final reservation with regard to later transmission requests; that is, if a node overhears a tentative reservation it must update its NAV so that any later requests it receives that would conflict with the tentative reservation must be denied. Thus, a tentative reservation effectively serves as a placeholder until either a new reservation is received or the tentative reservation is confirmed as the final reservation. Final reservations are confirmed by the presence or absence of a special subheader, called the *Reservation SubHeader* (RSH), in the MAC header of the data packet. The reservation subheader consists of a subset of the header fields that are already present in the 802.11 data packet frame, plus a check sequence that serves to protect the subheader. The fields in the reservation subheader consist of only those fields needed to update the NAV, and essentially amount to the same fields present in an RTS. Furthermore, the fields (minus the check sequence) still retain the same functionality that they have in a standard 802.11 header. The reservation subheader is used as follows. Referring again to Figure 5, in the instance that the tentative reservation D_{RTS} is incorrect, Src will send the data packet with the special MAC header containing the RSH subheader. A, overhearing the RSH, will immediately calculate the final reservation D_{RSH}, and then update its NAV to account for the difference between D_{RTS} and D_{RSH}. Note that, for A to update its NAV correctly, it must know what contribution D_{RTS} has made to its NAV. One way this can be done, is to maintain a list of the end times of each tentative reservation, indexed according to the $< sender, receiver >$ pair. Thus, when an update is required, a node can use the list to determine if the difference in the reservations will require a change in the NAV.



(a) Data packet frame.



(b) RTS/CTS control packet frames.

Figure 6: MAC frame formats used in IEEE 802.11 for ad-hoc networks (IBSS).

4. INCORPORATION OF RBAR INTO 802.11

In this section we describe how RBAR may be incorporated into 802.11. We start by presenting background information on the formats of the relevant 802.11 frames, and then describe in detail how these frames can be modified to accommodate RBAR.

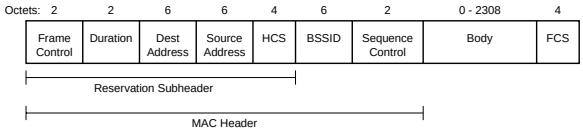
The IEEE 802.11 frame formats are shown in Figure 6. Figure 6(a) shows the format of the MAC frame used for sending unicast data packets in an ad-hoc network (IBSS). The *frame control* field carries frame identification information, such as the type of frame (e.g. management, control, or data), as well as protocol version information and control flags; the *duration* field contains the time remaining (in μs) until the end of the packet transfer (e.g. D_{DATA} in Figure 4); the *BSSID* is the unique network identifier; *sequence control* is a sequence number used to detect duplicate frames; and *FCS* is the frame check sequence. Figure 6(b) shows the format of the RTS and CTS control frames. The fields they share in common with the data frame serve the same purpose, except the duration fields contain the D_{RTS} and D_{CTS} values shown in Figure 4. For a more complete description, the reader is referred to [12].

Modifications to the standard 802.11 frames for RBAR are minimal, and are illustrated in Figure 7. A description of each modification is given next, followed by the design rationale.

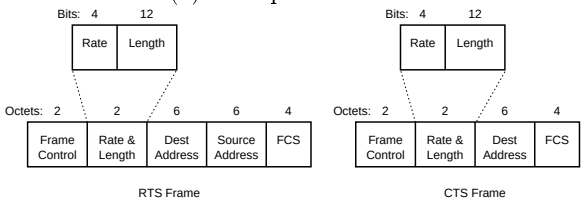
1. A new MAC data frame is introduced, shown in Figure 7(a), in which the standard 802.11 data frame has been changed to include a 32-bit check sequence positioned immediately after the source address field. The check sequence is used to protect the *reservation subheader*, which consists of the *frame control*, *duration*, *destination address*, *source address* and *address 2* fields of the header. The new frame is distinguished by other MAC frames by a unique type/subtype code in the frame control field (see [12] for more information on frame type codes).

2. The RTS and CTS control frames, shown in Figure 7(b), have been changed to encode a 4 bit *rate* subfield and a 12 bit *length* subfield, in place of the 16 bit *duration* field in the standard IEEE 802.11 frames. The *rate* subfield uses an encoding similar to the *rate* field in the PLCP header for the 802.11a supplement standard [13], and the *length* subfield gives the size of the data packet in octets.

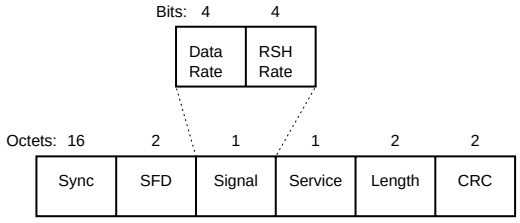
3. The physical layer header (PLCP), shown in Figure 7(c), has been divided into two 4 bit *rate* subfields, which use the similar rate encodings as those in 802.11a [13]. The first subfield, if non-empty, indicates the rate at which the reservation subheader will be transmitted, and the second subfield indicates the rate at which the remainder of the packet will be transmitted.



(a) Data packet frame.



(b) RTS/CTS control packet frames.



(c) Physical layer (PLCP) header.

Figure 7: MAC and physical layer frame formats used in the RBAR protocol.

The rationale for the modifications shown in Figures 7(a) and 7(b) was discussed at length in the previous section. Following, is a discussion of the modifications to the physical layer (PLCP) header shown in Figure 7(c).

In standard 802.11, the PLCP header contains an 8 bit *signal* field that encodes the rate at which the payload of the physical frame (the MAC packet) should be transmitted. These fields are used as follows. When the physical layer has a packet to transmit, it first transmits the PLCP header at a fixed rate that is supported by all nodes (1Mbps). It then switches to the rate encoded in the *signal* field for the transmission of the payload. After verifying that the PLCP header is correct, using the *CRC*, the receiving physical layer switches to the rate given in the *signal* field to receive the packet payload. The end of the transmission is determined by the receiver from the *length* field, which stores the duration of the transmission (in μs).

In RBAR, the physical layer may be required to switch rates twice during transmission of the payload: once for the

reservation subheader, and again for the remainder of the payload. To enable the use of an additional rate for the reservation subheader, our protocol requires that two rate changes occur during transmission of the data packet. Thus, instead of a single 8 bit *signal* field, we subdivide it into two 4 bit subfields, as shown in Figure 7(c), where the first rate is used to send the reservation subheader, and the second for the remainder of the data packet. Thus, the PLCP transmission protocol is modified as follows. When the MAC passes a packet down to the physical layer, it specifies two rates: one for the subheader and one for the remainder of the packet. The physical layer then encodes the rates into the appropriate *signal* subfields shown in Figure 7 and then transmits the packet. The receiving physical layer, after verifying that the PLCP header has been received correctly, will then switch to the first RSH rate for receipt of the subheader, and then to the data rate for the remainder of the packet. Note that, as specified in the IEEE 802.11 standard, as each byte is received, it is immediately available to the MAC. Thus, nodes that rely on the RSH to update their reservations, will be free to do so as soon as the RSH has been received.

5. SIMULATION ENVIRONMENT

The results in this paper were generated using the *ns-2* network simulator from LBNL [7], with extensions from the CMU Monarch project [5] for modeling mobile ad hoc networks. Included in the simulation are models for a number of traffic generators, as well as networking stacks incorporating common ad hoc routing, MAC, and physical layer protocols. To this, we added more detailed MAC and physical layer models, including the addition of the modulation schemes and rate adaptation mechanisms that are the focus of this study, as well as the addition of a Rayleigh fading simulator for studying the impact of multipath fading. The Rayleigh fading simulator we used is based on the well known Jakes' [14] simulator, which generates a continuous time-varying Rayleigh fading envelope. Additionally, we enhanced the realism of the existing network interfaces using the Intersil Prism II chipset and accompanying reference interface designs as our model. The Prism II chipset is an IEEE 802.11, direct-sequence spread-spectrum (DSSS) radio that is used in many commercially available network interfaces, including the Aironet PC4800 [1] (now known as the Cisco 350). Most of our network interface parameters were drawn directly from the Intersil documentation, including power constraints, receiver noise factors, reference antenna gains, and sensitivity thresholds. Since our interest in these experiments was only to observe how the individual rate adaptation protocols reacted to the changing channel conditions, and not to evaluate the exact performance of currently available network devices. We differed slightly from the reference design of the Prism II chipsets and did not model the CCK modulation schemes, instead choosing to use the more widely known and well documented M-ary QAM modulation schemes [20]. However, similar results can be expected for CCK, MOK, and and other modulation schemes. Apart from the aforementioned changes, the nodes in our simulations were otherwise configured similar to those in [5].

5.1 Autorate Fallback Algorithm (ARF)

As a basis for comparison, we implemented Lucent's Autorate Fallback (ARF) protocol into the simulator. ARF is

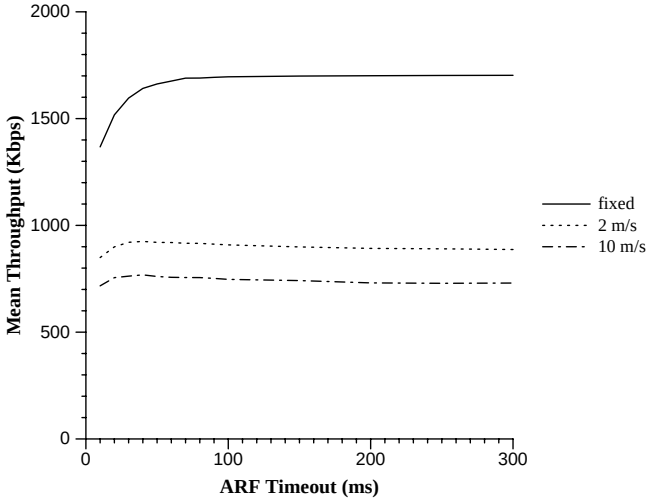


Figure 8: Comparison of the average throughput of ARF for various values of the timer it uses to indicate when it should attempt to increase the data rate in lieu of its usual indicator: the receipt of 10 consecutive ACKs. The curves shown are the average throughputs measured across a single CBR connection in a Rayleigh fading channel between two nodes oscillating near to far at different mean speeds. The *fixed* curve is the mean throughput between the nodes spaced at various intervals over the range of distance traveled by the oscillating nodes in the mobility curves.

the rate adaptation scheme used in Lucent's 802.11 WaveLAN II networking devices [15]. The protocol, as specified in [15], is summarized as follows. If ACKs for two consecutive data packets are not received by the sender, then the sender drops the transmission rate to the next lower data rate and starts a timer. If ten consecutive ACKs are received, then the transmission rate is raised to the next higher data rate and the timer is cancelled. However, if the timer expires, the transmission rate is raised as before, but with the condition that if an ACK is not received for the very next packet, then the rate is lowered again and the timer is restarted. In our implementation we attempted to adhere as closely as possible to the description given in [15]. However, values for the timeout were unspecified.

Therefore, prior to initiating our study, we experimented with several timeout values to determine a reasonable value for our simulations. The results of these experiments are shown in Figure 8, which shows the average throughput as a function of the timeout value for several different mean node speeds. From these results it appears that ARF is relatively insensitive to the choice of timeout, for the given scenarios. However, there is a clear threshold region in the 40ms-60ms range, depending on the degree of mobility, beyond which there is little performance change but below which there is a noticeable drop. The drop can be attributed to the greater frequency at which packets are lost due to rate increases triggered by timeouts during times in which the channel conditions are poor. For the experiments with mobility, the peak in performance is in the 40ms range, whereas for the

experiment without mobility, the performance rises sharply until the 60ms range and then levels off. The slight differences in peak values between the mobility experiments most likely represents those regions in which the timeouts are frequent enough to respond well to the variations in the Rayleigh channel, but not too frequent that the failed packet attempts significantly impact performance. Based on these results, we chose a value of 60ms for our simulations, which appears to be a reasonable compromise for the fixed and mobile simulations that we have used in our performance analysis.

5.2 Receiver-Based AutoRate Protocol (RBAR)

So far in our discussions of RBAR we have deliberately neglected to specify the channel quality estimation and rate selection protocols. This is because there are already a number of existing protocols in the literature (e.g. [3], [11]), [22]), any of which may be used in RBAR. However, for our performance analysis we chose the following.

For the channel quality estimation and prediction algorithm, we used a sample of the instantaneous received signal strength at the end of the RTS reception. In practice, of course, much more accurate techniques could be used, such as those in [3], [17], and [10].

For the rate selection algorithm, we used a simple threshold based technique. Threshold based techniques have been widely studied (e.g. [22], [3], [11]). In a threshold scheme, the rate is chosen by comparing the channel quality estimate against a series of thresholds representing the desired performance bounds of the available modulation schemes. The modulation scheme with the highest data rate that satisfies the performance objective for the channel quality estimate, is the chosen rate. The protocol we used was the following. Suppose we wish to select the modulation scheme that has the highest data rate among those with bit error rates $\leq$ 1E-5 for the estimated SNR of the next packet. The protocol would then choose the modulation scheme as follows. Let $M_1, \ldots, M_N$ represent the set of modulation schemes in increasing order of their data rate, and $\theta_i, \ldots, \theta_N$ represent the SNR thresholds at which $BER(M_i) = $ 1E-5. Choose modulation scheme

M_1 if $SNR < \theta_1$

M_i if $\theta_i \leq SNR < \theta_{i+1}, \quad i = 1, \ldots, N-1$

M_N otherwise

Notice that this protocol assumes that the values of $\theta_1, \ldots, \theta_N$ are known. In practice, however, it is impossible to determine the BER characteristics precisely, necessitating the use of approximations. For our simulations we used the BER equations found in [20], which are presented in the next section.

5.3 Error Model

Our error model was based on the detailed simulation of a Rayleigh fading channel, using the well known Jakes' method [14]. In this section, we describe, in detail, how we used this method to model packet errors in our simulations.

Jakes' method is a technique for simulating a signal with Rayleigh fading characteristics. The technique is based on the simulation of a finite number of oscillators with Doppler shifted frequencies, whose outputs are combined to produce the simulated Rayleigh fading signal. The resultant signal $\alpha(t) = x_c(t) + jx_s(t)$, where x_c and x_s are the signal's in-phase (real) and quadrature (imaginary) components, is computed as follows

$$x_c(t) = \frac{1}{\sqrt{N}} \sum_{n=1}^{N} \cos\beta_n \cos(\omega_n t + k\beta_n) \quad (1)$$

$$x_s(t) = \frac{1}{\sqrt{N}} \sum_{n=1}^{N} \sin\beta_n \cos(\omega_n t + k\beta_n) \quad (2)$$

where N is the number of oscillators, $k = 1$, and

$$\omega_n = \frac{2\pi v}{\lambda} \cos\left(\frac{\pi n}{2N+1}\right) \quad (3)$$

$$\beta_n = \frac{\pi n}{N} \quad (4)$$

The instantaneous gain of the channel is then the magnitude of the signal

$$|\alpha(t)| = \sqrt{x_c^2(t) + x_s^2(t)} \quad (5)$$

Given the gain, we computed whether a packet was received with errors using well known methods for calculating the pre-gain signal to noise ratio (SNR) and resultant bit error rate for the modulation schemes that were used.

To compute the value of the pre-gain received signal, we used the log-distance path loss model. This model gives the path loss P_l at a distance d from the transmitter based on the path loss at some close-in reference distance d_0.

$$P_l(d) = P_l(d_0) + n10\log_{10}(d/d_0) \quad (6)$$

where n, the *path loss exponent*, determines the rate of loss. A number of values for n have been proposed for different simulated environments. We used $n = 3$, which is commonly used to model loss in an urban environment [20]. To estimate $P_l(d_0)$, we used the Friis free space propagation model

$$P_r(d_0) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d_0^2 L} \quad (7)$$

where P_r and P_t are the receive and transmit powers (in Watts), G_t and G_r are the transmit and receive antenna gains, λ is the carrier wavelength (in meters), and L is a system loss factor ($L = 1$ in our simulations).

Noise was modeled as a combination of the noise floor of the interface and the aggregate energy of neighboring transmissions that were to weak to cause a collision. The noise floor was computed by first calculating the thermal noise N_t using the well known equation

$$N_t = kTB_t \quad (8)$$

where k is Boltzmann's constant (1.38×10^{23} Joules/Kelvin), T is the temperature (in Kelvin), and B_t is the unspread bandwidth of the interface; and then factoring in the published noise figure of the interface. For our simulations, we used a noise figure provided by Intersil for their Prism I chipset.

Finally, the received bit error rates were computed using the following bit error rate equations for the different modulation scheme that were used. For BPSK and QPSK [20]

$$P_e(t) = Q\left(\sqrt{\frac{2|\alpha(t)|^2 E_b}{N_o}}\right) \quad (9)$$

and for M-ary QAM

$$P_e(t) \approx 4\left(1 - \frac{1}{\sqrt{M}}\right) Q\left(\sqrt{\frac{3|\alpha(t)|^2 log_2(M) E_b}{(M-1)N_o}}\right) \quad (10)$$

where E_b/N_o is the bit-energy-to-noise ratio of the received signal and $|\alpha(t)|$ is the instantaneous gain of the Rayleigh channel (from Jakes'). The E_b/N_o of the received signal is derived from the SNR using the following relation

$$\frac{E_b}{N_o} = SNR \cdot \frac{B_t}{R_b} \quad (11)$$

where R_b is the maximum bit-rate of the modulation scheme and B_t is the unspread bandwidth of the signal.

Since portions of a packet may be transmitted at different modulation schemes, the probability that a packet was in error was based on separate calculations for each portion. Furthermore, since gain and noise may vary with time, we also accounted for those in our calculations by the following. For the gain, we used an approximation for the coherence time [20]. The coherence time T_c is the period over which the channel can be assumed to be effectively constant. To calculate T_c, we used the following approximation [20]

$$T_c(t) \approx \frac{9\lambda}{16\pi v(t)} \quad (12)$$

where $v(t)$ is the speed along the line-of-sight between the sender and receiver at time t, and $\lambda = c/f_c$ is the wavelength of the carrier frequency f_c (c is the speed of light). For the noise, we accounted for the changing conditions by tracking the beginning and ending times of each of the neighboring transmissions and adjusting SNR appropriately.

5.4 Network Configurations

In our analysis, we used several different network configurations.

Configuration 1

The first configuration consisted of two identically configured nodes communicating on a single channel. One of the nodes was held in a fixed position, while the other traveled along a direct-line path to and from the fixed node in a repetitious, oscillatory motion. The length of the path was 300m, which was the maximum effective transmission range of the modulation schemes as simulated (see Figure 2). The purpose of this configuration was to stress the rate adaptation schemes, but doing so within the bounds of a plausible scenario.

Configuration 2

The second configuration consisted of 20 nodes in continuous motion within a 1500x300 meter arena. For each experiment, nodes were placed in randomly chosen starting positions and followed randomly chosen paths according to the *random waypoint* mobility pattern used in [5] and elsewhere. The speeds at which nodes traveled were also chosen randomly, but were held to within ±10% of the mean node speed for the trial. For most experiments, we used mean node speeds of 2, 4, 6, 8, and 10 m/s. Unlike in the 2 node configuration, in this configuration we were interested in observing the performance characteristics of the proposed protocols in a plausible ad hoc networking environment. Thus, the nodes were configured to use the DSR routing protocol found in [5] instead of static routing. Unless otherwise stated, all results were based on the average of 30 runs using 30 precomputed scenarios, or *patterns*. Each pattern, generated randomly, designated the placement, heading, and speed of each node over the simulated time. For each pattern, the starting position and direction of the mobile node on the path was random, as well as its speed. For each subsequent traversal of the path, a different speed was chosen at random, uniformly distributed in an interval of $0.9v - 1.1v$, for some mean speed v. For experiments in which the mean speed v was varied, we used the same precomputed patterns so that the same sequence of movements occurred for each experiment. For example, consider one of the patterns, let's call it I. A node x in I that takes time t to move from point A to point B in the 5 m/s run of I will take time $t/2$ to traverse the same distance in the 10 m/s run of I. So, x will always execute the exact same sequence of moves in I, just at a proportionally different rate. The patterns we used had a duration of 600s at a mean node speed of 2 m/s. To provide a fair comparison, the exact same set of patterns were used for each protocol tested.

6. PERFORMANCE EVALUATION

In this section we present the results of our performance evaluation.

6.1 Overhead of the Reservation Subheader

There are several sources of overhead caused by the reservation subheader. The most obvious is the addition of the four byte check sequence to the MAC header. Additional overhead is encountered when the data rate used to transmit the RTS packet is lower than the rate used to transmit the data packet. Recall that the purpose of the reservation subheader is to update the tentative reservations that were made by the the RTS packet. If a node succeeds in hearing the RTS but fails to hear the subheader, then it may defer for an incorrect amount of time. Too short a time, and its next transmission may collide with the ACK coming back for the data packet. Too long a time, and the channel may be idle. Thus, the subheader must be sent at the same or lower rate to reach those nodes that heard the RTS. The per-packet overhead of the difference in rates is easy to calculate.

However, to gauge the impact that the per-packet overhead has on overall performance, we simulated the network in Configuration 1 with a single UDP connection for a range of packet sizes: 32, 256, 512, 1024, and 1460 bytes. Data was generated by an 8Mbps CBR source, and the data rate for the control packets (and, summarily, the reservation subheader), was fixed at 1Mbps. The results of these experiments are shown in Figure 9, which presents the throughput for both protocols as a percentage of ARF's throughput. Note that, even for small packet sizes, the overhead

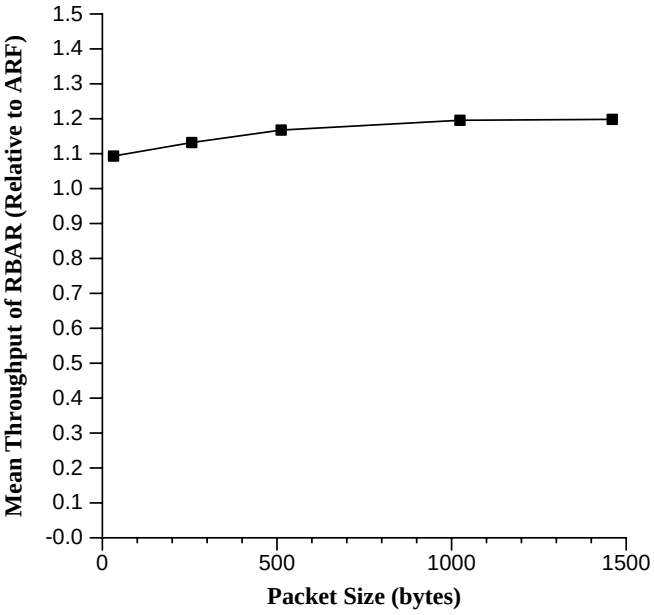


Figure 9: Impact of the reservation subheader on performance (relative to ARF) as a function of the packet size.

of RBAR's reservation subheader has a relatively modest performance impact. Even for the smallest packet size (32 bytes), RBAR maintains an approximate 10% improvement over ARF.

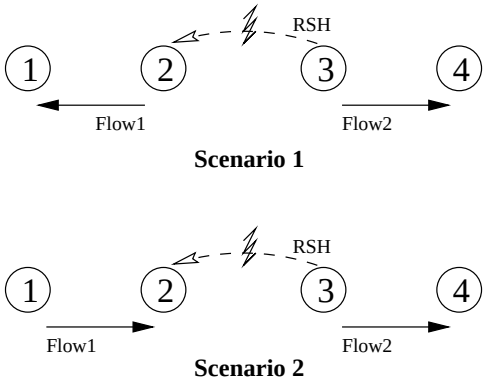


Figure 10: Network scenarios used to analyze the performance impact caused by the loss of the RSH subheader.

Even when sent at a low rate, a node may still fail to receive the subheader, such as, for example, when an RTS from a neighboring node collides with it or a deep fade causes excessive errors. Although such circumstances can also occur in standard 802.11, use of the reservation subheader may cause them to occur more frequently, and with more impact on performance, as touched on earlier. Thus, to gauge the sensitivity of RBAR to the loss of the reservation subheader, we simulated the networks shown in Figure 10 for varying loss rates. In Scenario 1, the network consisted of four nodes with two flows directed away from the center of the network such that the source nodes were able to hear each other but the sink nodes were out of range of all but the source of their flow. The distance between the nodes was such that the optimal rate along each flow was 2Mbps, and the rate announced in the RTS was always 1Mbps. In Scenario 2, the network was similar except the direction of one of the flows was reversed. In both scenarios, the reservation subheaders from Node 3's packets were corrupted with varying probability, so it was expected that Flow 1 would experience a decrease in performance with an increase in the probability of loss.

Table 1: Mean per-flow throughput for varying reservation subheader loss probability for the network scenarios shown in Figure 10

Err	Scenario 1			Scenario 2		
Prob	Flow1	Flow2	Total	Flow1	Flow2	Total
0.00	727	665	1393	222	1167	1389
0.05	708	690	1398	194	1187	1382
0.10	684	716	1400	187	1192	1380
0.15	676	724	1400	161	1220	1381
0.20	644	758	1402	153	1236	1389
0.25	634	761	1395	134	1260	1395
0.30	552	843	1394	123	1261	1385
0.35	537	855	1393	100	1285	1385
0.40	562	831	1393	93	1300	1393
0.45	498	896	1394	62	1330	1392
0.50	446	954	1400	47	1343	1390

The results of both experiments are shown in Table 1. Each row represents the measured throughput (in Kbps) for the probability of loss shown in the leftmost column. As a basis of comparison, the measured throughput for ARF in Scenario 1 was 576Kbps for Flow 1 and 572Kbps for Flow 2, and in Scenario 2 it was 278Kbps for Flow 1 and 867Kbps for Flow 2. The difference in the throughputs between the flows in Scenario 2 is due to problems with fairness in 802.11 [23]. For Scenario 1, there is only a moderate impact on performance. At 5% loss there is only a 3% decrease in performance, and the decline stays below 10% beyond a loss of 15%. However, in Scenario 2 we see a larger impact, starting at a decline of 14% at a 5% loss, increasing rapidly to 38% at a loss of 15%. Thus, it is evident that situations in which reservation subheaders are lost for nodes that are on the receiving end of a flow are more sensitive to that loss, most likely because the sender on that flow is subject to repeated backoff when its RTS's are ignored with increasing probability.

6.2 Slow Changing Channel Conditions

To observe the performance of the protocols under conditions when the channel conditions are static or slow changing, we again simulated the network in Configuration 1, but the mobile node was moved in 5m increments over the range of mobility (0m - 300m), and held fixed for a 60s transmission of CBR data over a single UDP connection. Here, data was generated at a rate of 8Mbps and sent in 1460 byte packets.

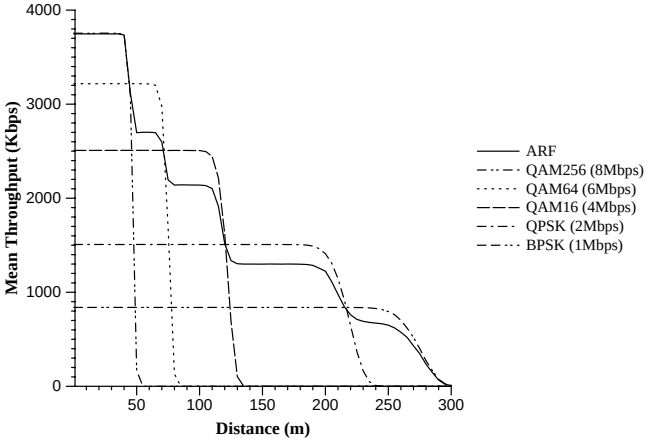


Figure 11: Performance of ARF for a single CBR connection between two nodes at fixed distances.

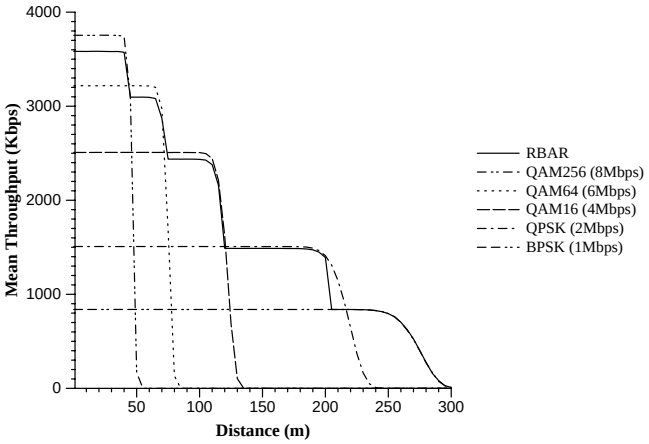


Figure 12: Performance of RBAR for a single CBR connection between two nodes at fixed distances.

The results of these experiments are shown in Figure 11 for ARF and Figure 12 for RBAR. Also shown are the results when the fixed rates are used. Notice that ARF fails to perform as well as the fixed rates at each distance except beyond that which is optimal for the highest rate. This is because ARF periodically tries to send data packets at the next highest rate in an attempt to gauge the channel conditions. In situations where the conditions are such that those packets are lost with high probability, then there is repeated packet loss resulting in the consistent performance degradation shown in the results.

RBAR, on the other hand, generally performs better at all distances except close in, where ARF excels. This is because of the increased impact of the reservation subheader. Recall that the reservation subheader has to be sent at one of the basic rates (in this instance, 1Mbps). Thus, at higher data rates the overhead of the subheader becomes more significant. One way to reduce this overhead is to employ a mechanism that predicts the best data rate for the channel conditions. One such technique is to simply cache the most

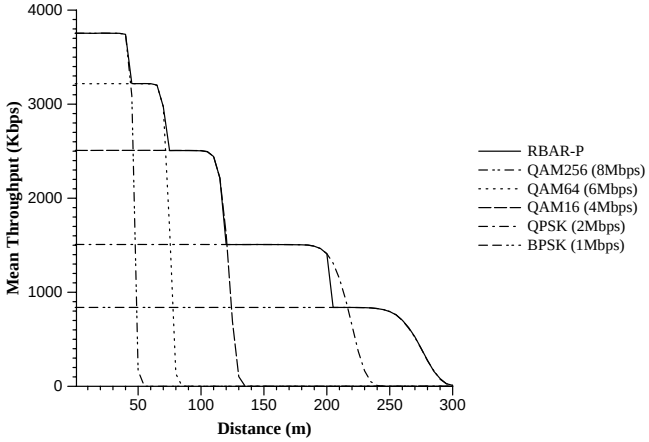


Figure 13: Performance of RBAR when a simple heuristic is used by the sender to try and predict the best data rate for the conditions, in an effort to reduce the frequency of the necessity for reservation subheaders.

recent rates as they are discovered. Figure 13 shows the results when such a technique is employed. Clearly there is a significant improvement in the instance shown here, due to the high predictability of the channel. However, better techniques such as those proposed in [3] may also work well for RBAR. This is a topic of future study.

6.3 Fast Changing Channel Conditions

In a Rayleigh fading channel, variations in the wireless signal are induced at a rate that depends, in part, on the speed along the line-of-sight between the transmitter and the receiver. For a conventional local area network with nodes moving at walking speeds (e.g., node speed $\leq$ 2 m/s communicating at 2Mbps over a 2.4GHz channel), changes generally occur slowly enough that the channel is effectively constant for the duration of a packet exchange (the coherence time). However, as the speed increases, changes occur much more rapidly, decreasing the predictability of the channel. Thus, by simulating a fading channel and varying the mean node speed, we can evaluate the adaptability of the protocols.

To observe this impact, we performed experiments for five different speeds, 2, 4, 6, 8, and 10 m/s, for Configuration 1. Results were generated for a UDP connection carrying CBR traffic that was generated at a rate of 8Mbps and sent in 1460 byte packets. These results are shown in Figure 14. Also included in the figure are results for the fixed rates (as a basis of comparison). Notice that:

- RBAR outperformed ARF for all mean node speeds, with the performance improvement ranging from 6% (10 m/s) to 20% (2 m/s).
- An increase in mean node speed resulted in a decrease in performance. As expected, the increase in variability of the signal resulted in a decrease in performance.

  Also notice that the performance improvement for RBAR also decreased as the mean node speed increased. Re-

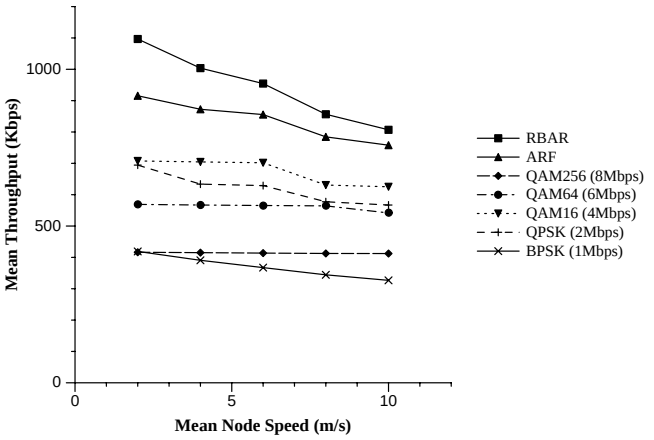


Figure 14: Performance for CBR traffic over a single UDP connection in a Rayleigh fading channel.

call that the simple channel quality prediction mechanism used in RBAR for these results works best when the channel coherence time is larger than the time it takes to transmit the CTS packet and the DATA packet. For 2 m/s, the coherence time was sufficiently large that this was true for packets transmitted at all data rates. However, as the node speed increased, the coherence time shortened and the higher data rates were also affected, resulting in a decline in performance. As mentioned previously, we expect that this decline can be improved significantly with better channel quality prediction techniques.

The adaptability of RBAR to the rapidly changing channel conditions can be more clearly seen in Figure 15. Compared to the similar figure for ARF, it is clear that RBAR is much better at reacting and adapting to the channel conditions.

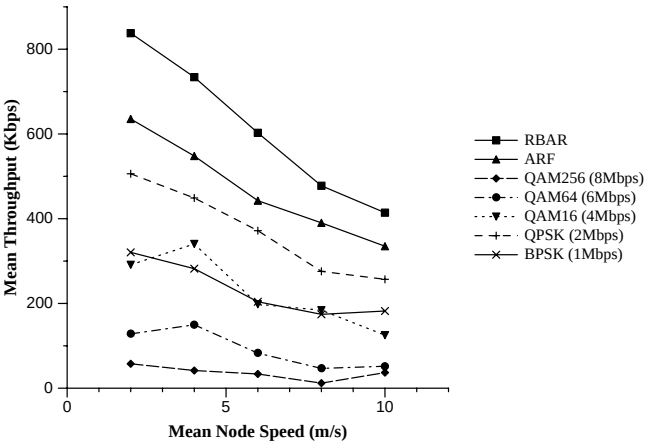


Figure 16: Performance for FTP traffic over a single TCP connection in a Rayleigh fading channel.

We also simulated a single TCP connection under the same conditions. These results are shown in Figure 16. Notice that the performance improvement is more significant, which can be attributed to TCP's sensitivity to packet loss due to wireless errors.

6.4 Impact of Variable Traffic Sources

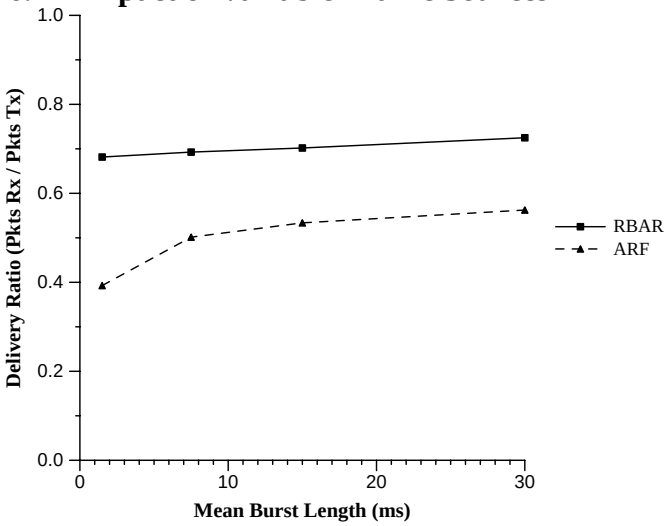


Figure 17: Protocol efficiency for an ON/OFF Pareto source generating traffic on a single UDP connection in a Rayleigh fading channel. The mean OFF time = 1s.

In this section, we study the impact of bursty data sources on the performance of the RBAR and ARF protocols. For this study, we performed a series of experiments using an ON/OFF traffic source, with ON ($\bar{\tau}_{on}$) and OFF ($\bar{\tau}_{off}$) times drawn from a Pareto distribution. During an ON period, data was generated at a rate of 8Mbps and sent in 1460 byte data packets, resulting in mean packet bursts ranging from $\approx 1-2$ packets ($\bar{\tau}_{on} = 1.5ms$) to ≈ 20 packets ($\bar{\tau}_{on} = 30ms$). Traffic was generated for a single UDP connection across a Rayleigh fading channel. The mean node speed was 2 m/s, and we used Configuration 1.

The results of these experiments are presented in Figures 17 and 18, for mean OFF times of 1s and 500ms respectively, which show the average delivery ratios for each protocol, where the delivery ratio is defined as the number of data packets successfully received over the total number of data packets sent.

Note that:

- RBAR outperforms ARF in all traffic conditions, with improvements ranging from 26% to 70%.
- RBAR shows the greatest improvement when the traffic is the lightest, and the least improvement when the traffic is heavy.

6.5 Multi-Hop Performance

In this section we present results for Configuration 2: 20 nodes in continuous motion within a 1500x300 meter arena. Here, we simulated a single CBR source generating traffic on

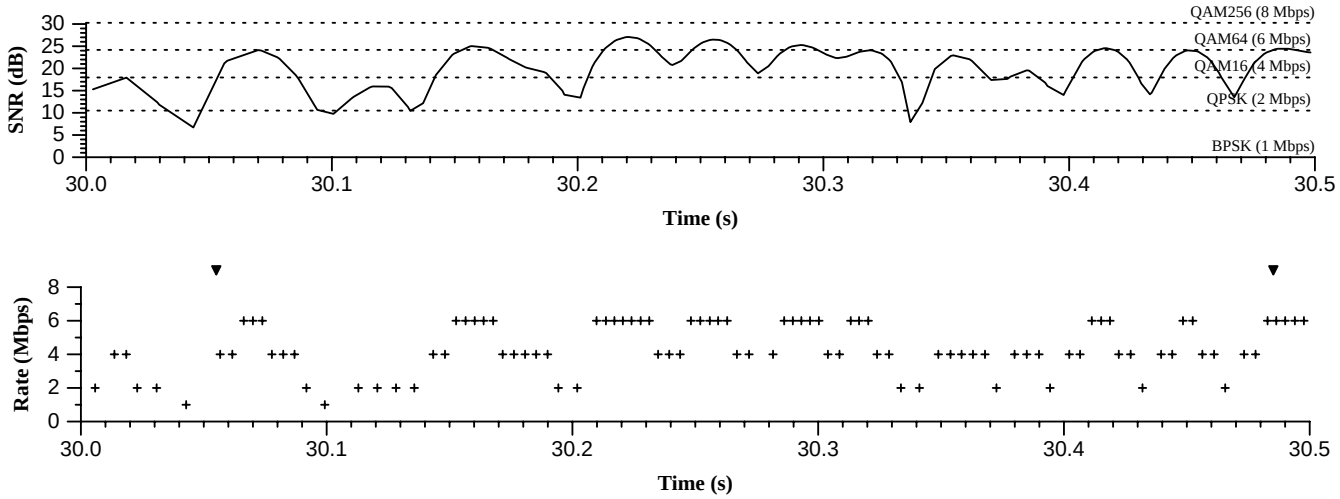


Figure 15: Performance of RBAR for a single CBR connection between two nodes in a Rayleigh fading channel. Here, the sender is fixed and the receiver is moving at a speed of 2 m/s away from the sender. The lower graph shows the time at which packets were transmitted, and the modulation rate chosen by ARF for each packet. The tick marks along the top show the time at which packets were dropped by the receiver due to errors. The upper graph shows the SNR at the receiver for the packets in the lower graph. Also shown are thresholds representing the SNR values above which the next higher modulation rate has a theoretical mean $BER \leq 10^{-6}$. At the start, both nodes were at the same location, so the leftmost edges represent the point in time at which the two nodes were 60m apart.

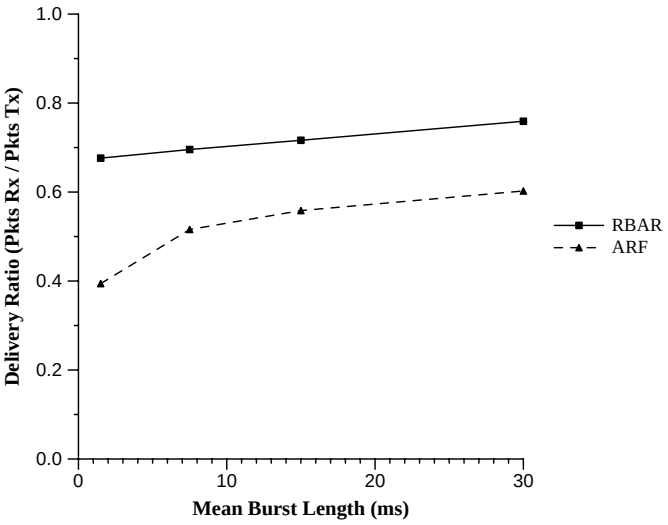


Figure 18: Protocol efficiency for an ON/OFF Pareto source generating traffic on a single UDP connection in a Rayleigh fading channel. The mean OFF time = 500ms.

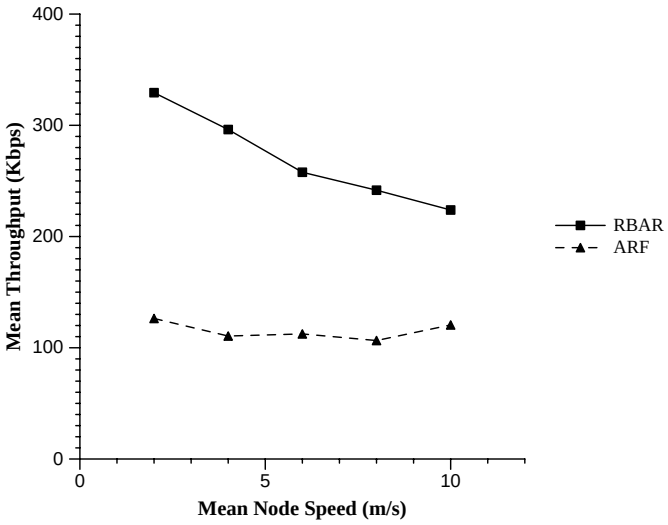


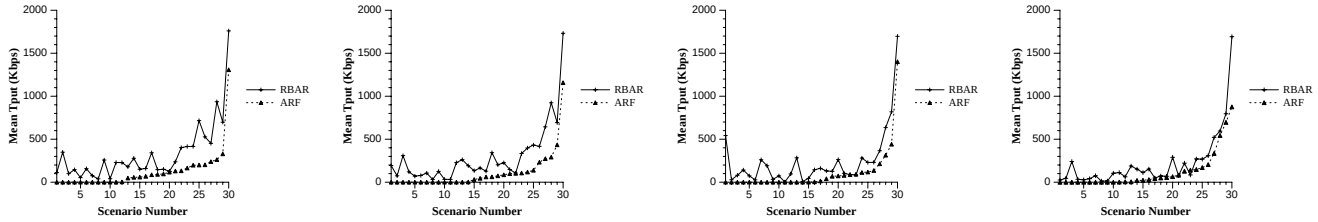
Figure 19: Performance comparison for a single CBR connection in a multihop network.

a single UDP connection between two nodes in the ad-hoc network. The results are shown in Figure 19 and Figure 20. Notice that RBAR consistently outperforms ARF.

Similar results are shown in Figure 21 for an FTP source generating traffic over a TCP connection. Clearly, the performance gains observed earlier are also applicable to a multihop scenario. We believe that ARF's increase in throughput with increased speed is due to its poor ability to select the correct rate when nodes are far apart, resulting in repeated backoff by TCP early in the simulations. However, with increased speed it may occur that the connection is established sooner due to the speed at which nodes in the sparse starting alignment are brought into range.

7. FUTURE WORK

We intend to explore several topics of future work based on our work in this paper. One idea is to extend RBAR to sit-



(a) mean speed = 2m/s (b) mean speed = 4m/s (c) mean speed = 8m/s (d) mean speed = 10m/s

Figure 20: Performance comparison across multiple hops. Shown here are the results for the individual scenarios, sorted according to the increasing throughput of ARF.

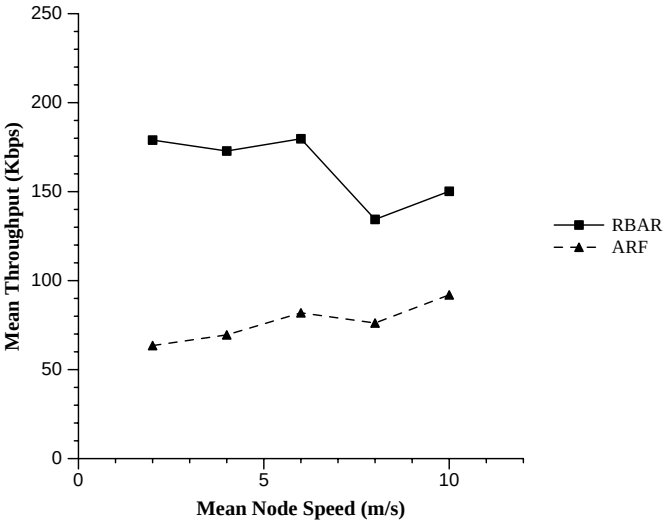


Figure 21: Performance comparison for a single FTP connection across a multihop network.

uations where the RTS/CTS protocol is not used for every packet, such as in the Basic Access mode in 802.11, by using a hybrid scheme where the RTS/CTS is used only when several ACKs are lost, or a length of time has expired since the last packet was transmitted. Another idea is to incorporate the packet size into the rate selection, since smaller packets have a lower probability of error than larger packets. Finally, we are currently looking at routing techniques that will take advantage of the autorate capabilities of RBAR, by probing for routes that satisfy certain quality characteristics, such as the highest capacity or the most stable.

8. CONCLUSION

In this paper, we addressed the topic of optimizing performance in wireless local area networks using rate adaptation. We presented a new approach to rate adaptation, which differs from previous approaches in that it uses the RTS/CTS protocol to enable receiver-based rate adaptation. Using this approach, a protocol based on the popular IEEE 802.11 standard was presented, called the *Receiver-Based AutoRate* (RBAR) protocol. Simulation results were then presented comparing the performance of the proposed protocol against the performance of an existing 802.11 protocol for mobile nodes across Rayleigh fading channels. These results showed that RBAR consistently performed well.

9. ACKNOWLEDGMENTS

The authors wish to thank K. Narayanan for the many helpful discussions on modulation and fading, and for the invaluable help on the Rayleigh fading simulator, including the contribution of code implementing Jakes' method. The authors would also like to thank the reviewers for their helpful comments.

10. REFERENCES

[1] Aironet. *PC4800 User Guide*, 1998. http://www.aironet.com/support/ftp/.

[2] S. M. Alamouti and S. Kallel. Adaptive trellis-coded multiple-phase-shift keying for rayleigh fading channels. *IEEE Transactions on Communications*, 42:2305–2314, June 1994.

[3] K. Balachandran, S. R. Kadaba, and S. Nanda. Channel quality estimation and rate adaption for cellular mobile radio. *IEEE Journal on Selected Areas in Communications*, 17(7):1244–1256, July 1999.

[4] V. Bharghavan. MACAW: A media access protocol for wireless LAN's. In *Proceedings of SIGCOMM'94*, London, 1994.

[5] J. Broch, D. A. Maltz, D. B. Johnson, Y. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *ACM/IEEE Int. Conf. on Mobile Computing and Networking*, pages 85–97, Oct. 1998.

[6] M. Eyugoglu, C. Forney, P. Dong, and G. Long. Advanced modulation techniques for v.fast. In *Eur. Trans. Telecommun.*, volume 4, pages 243–256, May-June 1993.

[7] K. Fall and K. Varadhan. ns *Notes and Documentation*. LBNL, August 1998. http://www-mash.cs.berkeley.edu/ns/.

[8] C. L. Fullmer and J. J. Garcia-Luna-Aceves. Solutions to hidden terminal problems in wireless networks. In *ACM SIGCOMM '97*, pages 14–18, Cannes, France, September 1997.

[9] J. H. Gass, M. B. Pursley, H. B. Russell, R. J. Saulitis, C. S. Wilkins, and J. S. Wysocarski. Adaptive transmission protocols for frequency-hop radio networks. In *Proceedings of the 1998 IEEE Military Communications Conference*, volume 2, pages 282–286, October 1998.

[10] D. L. Goeckel. Adaptive coding for time-varying channels using outdated fading estimates. *IEEE Transactions on Communications*, 47(6):844–855, June 1999.

[11] A. Goldsmith and S. G. Chua. Adaptive coded modulation for fading channels. *IEEE Transactions on Communications*, 46:595–602, May 1998.

[12] IEEE Computer Society. *802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, June 1997.

[13] IEEE Computer Society. *802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High Speed Physical Layer in the 5 GHz Band*, September 1999.

[14] W. C. Jakes, editor. *Microwave Mobile Communications*. IEEE Press, 1994.

[15] A. Kamerman and L. Monteban. WaveLAN-II: A high-performance wireless LAN for the unlicensed band. *Bell Labs Technical Journal*, pages 118–133, Summer 1997.

[16] P. Karn. MACA – a new channel access method for packet radio. In *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, pages 134–140. ARRL, 1990.

[17] M. B. Pursley and C. S. Wilkins. Adaptive transmission for direct-sequence spread-spectrum communications over multipath channels. *International Journal of Wireless Information Networks*, 7(2):69–77, 2000.

[18] X. Qiu and K. Chawla. On the performance of adaptive modulation in cellular systems. *IEEE Transactions on Communications*, 47(6):884–895, June 1999.

[19] R. Ramanathan and M. Steenstrup. Hierarchically-organized, multihop mobile wireless networks for quality-of-service support. *Mobile Networks and Applications*, 3(1):101–119, June 1998.

[20] T. S. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall, 1996.

[21] F. A. Tobagi and L. Kleinrock. Packet switching in radio channels: Part ii - the hidden terminal problem in carrier sense multiple-access modes and the busy-tone solution. *IEEE Transactions on Communications*, COM-23(12):1417–1433, 1975.

[22] T. Ue, S. Sampei, N. Morinaga, and K. Hamaguchi. Symbol rate and modulation level-controlled adaptive modulation/TDMA/TDD system for high-bit-rate wireless data transmission. *IEEE Transactions on Vehicular Technology*, 47(4):1134–1147, November 1998.

[23] N. Vaidya and P. Bahl. Fair scheduling in broadcast environments. Technical Report MSR-TR-99-61, Microsoft Research, 1999.

[24] W. T. Webb and R. Steele. Variable rate QAM for mobile radio. *IEEE Transactions on Communications*, 43:2223–2230, July 1995.