A Rate-Adaptive MAC Protocol for Wireless Networks

Gavin Holland, Nitin Vaidya
Texas A&M University
({gholland,vaidya}@cs.tamu.edu)

Paramvir Bahl
Microsoft Research
(bahl@microsoft.com)

Technical Report TR00-019

Dept. of Computer Science
Texas A&M University

August 7, 2000

Abstract

Wireless local-area networks are becoming increasingly popular. This is due, in part, to the recent availability of devices capable of communicating at data rates approaching that of conventional wired networks. These high rates are made possible through new modulation and coding techniques that dramatically increase bandwidth efficiency. However, maintaining reliable communication at higher data rates requires more signal power. Consequently, wireless devices often support multiple data rates, providing the user the ability to choose the rate that best suits their application. Alternatively, an automatic rate adaption mechanism may be used. Rate adaption is the process of automatically selecting the rate that gives the optimum throughput for the channel conditions. Although rate adaption mechanisms for cellular wireless networks have been studied at length, few have been proposed for wireless local-area networks. This paper presents one such mechanism: a rate adaptive MAC protocol based on the RTS/CTS collision avoidance handshake, called the Receiver-Based AutoRate (RBAR) protocol. The protocol is unique in that the rate adaption mechanism is located on the receiver, instead of the sender. Simulation results of an implementation of RBAR into IEEE 802.11 show that this arrangement performs well, in comparison to an existing protocol.

1 Introduction

Wireless local-area networks are becoming increasingly popular. This is due to the ratification of standards, like IEEE 802.11 [11], that have laid the foundation for wireless devices capable of transmitting at data rates approaching that of conventional wired networks. For example, devices are now available that can transmit at 11Mbps, with 54Mbps expected in the near future. With the promise of anytime, anywhere communication, at rates previously available only on the desktop, it is easy to see why wireless local-area networks are becoming popular.

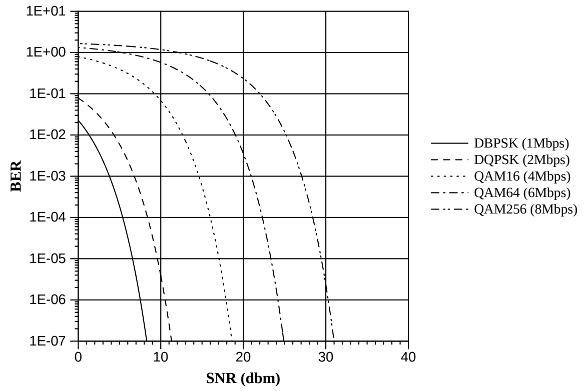


Figure 1: Theoretical bit error rates (BER) as a function of the signal-to-noise ratio (SNR) for several modulation schemes and data rates in an AWGN channel.

Higher data rates are commonly achieved by increasing the bandwidth efficiency of the modulation scheme. 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 signal pulses, or *symbols*. Each symbol may encode a fraction of a bit, or several bits, depending on the scheme. The ratio *bits/symbol* is called its *bandwidth efficiency*. The symbol sequence is then transmitted at a certain rate, the *symbol rate*, which is usually fixed. The data rate, then, is determined by the symbol rate and the number of bits encoded per symbol. High rate modulation schemes simply encode more bits per symbol – i.e. they are more bandwidth efficient.

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 in the channel all contribute to variations in the received signal-to-interference plus noise ratio (SINR). The variation

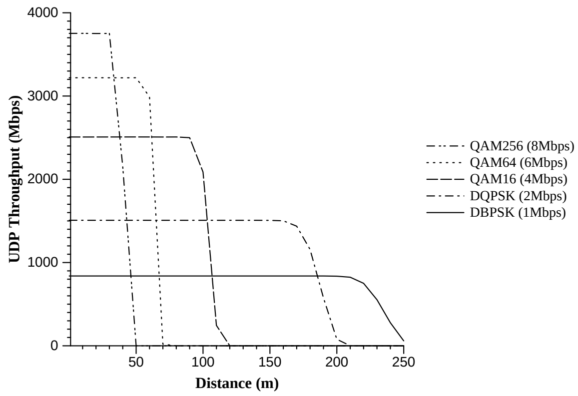


Figure 2: Comparison of throughput versus distance for several modulation schemes. The results were obtained by simulation of a single UDP connection with a CBR source in an AWGN channel with Friis free-space path loss.

in SINR results in variations in the bit error rate (BER), because the lower the SINR, the more difficult it is for the modulation scheme to decode the received signal, resulting in a higher (BER). Since an increase in bandwidth efficiency means denser encoding, a tradeoff emerges between data rate and power: the higher the data rate, the higher the required signal power.

This tradeoff is illustrated in Figure 1, which shows the theoretical BER as a function of the signal-to-noise ratio (SNR) for several different modulation schemes in an additive white Gaussian noise (AWGN) channel. Notice that, for an increase in data rate, an increase in signal power is required to maintain the same BER. For example, to achieve a bit error rate of 1E-5, a packet transmitted at 8Mbps (QAM256) requires 158× more signal power (22dBm gain) than the same packet transmitted at 1Mbps (DBPSK).

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 in Figure 1. Here, for the sake of illustration, only free-space path loss is modeled. Transmit power is constant. 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 SINR 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 SINR, such as fading and interference, which further impact performance.

Consequently, many conventional wireless local-area networking devices are designed with the capability of transmitting at multiple data rates, providing users with the flexbility to choose the rate that best suits their environment and application. For example, users who value high coverage might opt to use a lower rate. Alternatively, a rate adaption technique may be employed.

1.1 Rate Adaption

Rate adaption 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 conditions. A proven technique for wireline modems [5], rate adaption has recently attracted attention as a technique that can also be used to great effect in wireless systems [15] [9], [20], [1].

There are two aspects to rate adaption: 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-interference plus noise ratio (SINR), signal strength, symbol error rate, bit error rate), which predictors should be used, whether predictions should be short-term or long-term, etc. [2], [8]. 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 [19], [2].

Among the factors that influence the effectiveness of rate adaption, of particular importance is the accuracy of the channel quality estimates. Inaccurate estimates cause poor rate selection. Thus, it is advantagous to utilize the best information available when generating channel quality estimates. Furthermore, since it is the channel quality seen by the receiver that determines whether a packet can be received, the best information is found on the receiver – e.g. SINR samples, error rates, and fading estimates provided by the receiver hardware. It is equally important that, once the estimates are generated, they be used before they become stale. Thus, it is also advantageous to minimize the delay between the time the channel quality is estimated and the packet is transmitted.

Much of the previous work on rate adaption in wireless has assumed a cellular network (e.g. mobile nodes communicating to a base station over a TDMA/TDD link) [2], [15], [19]. We have observed that many of these techniques have the following characteristics: rate selection is performed by the sender; channel quality estimation is performed by the receiver and periodically fed to the sender either on the same channel (e.g. in alternating TDMA/TDD slots) or on a separate subchannel; and they operate at the physical layer, adapting rates on a symbol-by-symbol or slot-by-slot basis, transparent to upper layers.

Few rate adaption techniques have been designed for wireless local-area networks (e.g. mobile nodes communicating peer-to-peer over CSMA/CA links). There are two papers that address rate adaption in wireless local-area networks. In [16], the authors present a protocol for a dual-channel slotted-aloha MAC, in which the sender uses explicit feedback via a control channel to select the best rate for the data channel. And, in [13], the authors present a protocol for 802.11, used in Lucent's WaveLAN II devices, in which the sender uses data packets to probe for the best rate, basing rate selections on whether probe packets are dropped. Note that, in both protocols, rate selection is done by the sender, and in [13] channel quality estimation is also performed by the sender. Also note that only [13] is based on a widely used, wireless local-area networking standard.

In this paper, we propose a new approach to rate adaption in wireless local-area networks. Our approach differs from those in [16] and [13] in that rate selection and channel quality estimation are both located on the *receiver*, avoiding the costly transmission of channel quality feedback and packet probing. This is made possible by utilizing the

RTS/CTS collision avoidance handshake for the purpose of rate adaption.

2 Proposed Approach

In this paper, we propose a new approach to rate adaption in wireless local-area networks, which differs from existing approaches in that rate selection and channel quality estimation are both located on the receiver. The motivation for this approach is based on the following observations:

- Rate selection can be improved by providing more, and more accurate, channel quality information.
- Channel quality information is best acquired at the receiver.
- Transmitting channel quality information to the sender is costly.

To demonstrate this approach, we have developed the Receiver-Based Autorate (RBAR) protocol, which is a rate adaptive MAC protocol for wireless local-area networks.

2.1 The Receiver-Based Autorate (RBAR) Protocol

The Receiver-Based Autorate (RBAR) protocol is based on the RTS/CTS collision avoidance handshake, common in MAC protocols for wireless local-area networks (e.g. SRMA [18], MACA [14], MACAW [3], FAMA [7], IEEE 802.11 [11]).

The purpose of the RTS/CTS handshake is to reserve the wireless channel for the duration of a packet transmission, to avoid 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. In conventional RTS/CTS protocols, the *sender* selects the data rate at which to transmit the packet, and then calculates the duration of the reservation based on the packet size and the selected rate. The reservation is then transmitted in an exchange of RTS/CTS control packets with the receiver. The *RTS* (Ready to Send) and *CTS* (Clear to Send) packets serve two purposes: 1) to request and acknowledge the reservation between the sender and receiver, and 2) to announce the duration of the reservation to all nodes that are in range. Nodes that overhear the RTS/CTS messages react by deferring their own transmissions for the duration of the reservation.

In RBAR, the RTS/CTS handshake is modified to allow the *receiver* to choose the rate at which the packet will be transmitted. Instead of containing the duration of the reservation, the RTS/CTS packets carry two fields: data rate and data packet size. Together, these fields provide the information needed to allow nodes that overhear either RTS or CTS packet to calculate the duration of the reservation. However, in the RTS, the rate field carries the rate which the sender *intends* to use for the data packet. Whereas, in the CTS, it carries the *actual* rate that will be used, as selected by the receiver. When the rates differ, reservations based on the outdated rate in the RTS are updated by the data packet's header.

The protocol is illustrated in the example shown in Figure 3. Here, node S has a data packet of size n to send to node R, and A and B are nodes in range of S and R, respectively. The protocol behaves as follows.

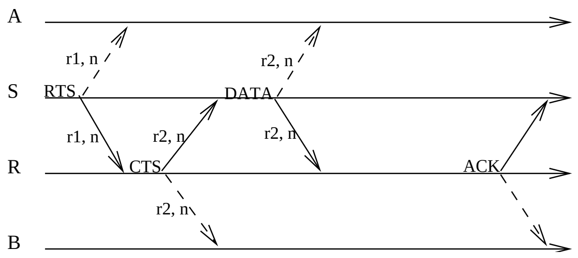


Figure 3: Example packet transfer using the proposed Receiver-Based Autorate (RBAR) protocol.

- S chooses a data rate $r1$, using some heuristic, and sends $r1$ and the size of the data packet n in the RTS to R.
- A, overhearing the RTS, uses $r1$ and n to calculate the duration of the reservation, marking it as *tentative.*
- R, having received the RTS, uses some channel quality estimation and rate selection technique to select the best rate $r2$ for the channel conditions, and sends $r2$ and n in the CTS to S.
- B, overhearing the CTS, calculates the reservation using $r2$ and n.
- S responds to the CTS by placing $r2$ into the header of the data packet and transmitting the packet at the selected rate. If $r1 \neq r2$, S uses a unique header signaling the rate change.
- A, overhearing the data packet, looks for the unique header. If it exists, it recalculates the reservation to replace the tentative reservation it calculated earlier.

Benefits to this design include:

- The rate selection mechanism has local access to all of the channel quality information available at the receiver, such as information from the receiver's hardware acquired during receipt of the RTS.
- The RTS can be used for estimating channel quality, very near to the time the data packet is transmitted.
- A separate channel for feedback of channel quality information to the sender from the receiver is not required.
- Rate adaption is performed on a per-packet basis.
- It can be implemented into 802.11.

Note, we have not specified the techniques for channel quality estimation and rate selection. The objective of this work is to demonstrate the usefulness of the receiver-based rate adaption approach using the RTS/CTS mechanism, and the potential performance improvement that it can achieve over existing approaches, not to advocate any particular physical layer channel quality estimation or rate selection technique. The ideas in this paper should apply equally well for use with any of such techniques. However, for the purposes of our performance evaluation, we used the channel quality estimation and rate selection techniques that are described in [2] for slow feedback-based rate adaption. Their approach is a threshold based rate selection scheme, which uses average SINR as an indicator of channel quality. We deviated slightly from their scheme, using instantaneous SINR, sampled at the end of a packet reception.

3 Implementation of RBAR into IEEE 802.11

In this section, we show how RBAR can be implemented into IEEE 802.11.

3.1 Preliminaries and Assumptions

In this section, we briefly describe features of the IEEE 802.11 MAC that are relevant to later sections. We refer the reader to [11] for more information on 802.11.

3.1.1 802.11 Reservation Access Control

The reservation access control protocol is an implementation of the RTS/CTS collision avoidance protocol, and is part of the Distributed Coordination Function (DCF) in the IEEE 802.11 MAC.

In the reservation access control protcol, the duration of a reservation is carried in the *duration* field of the RTS, CTS, and ACK control packets, as well as in the *duration* field in the MAC header of data packets.

Nodes track reservations in a data structure called the *Network Allocation Vector* (NAV). The NAV is consulted during carrier sensing to determine the current "busy" status of the channel. Thus, it provides MAC level *virtual* carrier sensing as a supplement to the physical carrier sensing provided by the device.

To illustrate the reservation access control protocol, consider the following example, where node x has a packet to send to node y. Node x first requests a reservation by calculating the duration of the reservation T and sending it in the *duration* field of the MAC header of an RTS to y. The duration T is the time that will be required from the moment after the RTS has been received, until the moment after the ACK has been received, and is calculated using $T = T_{CTS} + T_{DATA} + T_{ACK} + 3 * SIFS$. T_{CTS} and T_{ACK} are the estimated transmission times of the CTS and ACK packets at a rate chosen from the $BSSBasicRateSet$, and T_{DATA} is the estimated transmission time of the data packet using a rate chosen by x from the set of rates supported by both x and y. The $BSSBasicRateSet$ is the set of rates that all nodes are required to support. $SIFS$ is a physical layer constant. Each subsequent packet in the exchange carries the time remaining in the reservation in their *duration* field so that nodes in range of x and y are able to add the reservation to their NAVs. The time remaining is calculated for each packet by subtracting out the expected transmission time for the packet from the value of the *duration* field in the previous packet. For example, the *duration* field of the CTS packet sent by y would have a value of $T' = T - (T_{CTS} + SIFS)$, where T is the value of the *duration* field in the RTS that y received from x.

3.1.2 Support for Rate Adaption in 802.11

802.11 was designed to accomodate per-packet data rate selection. In 802.11, the physical layer prefaces every packet with a header (PLCP) that indicates the rate that will be used to transmit the packet. The PLCP header is then sent at a fixed rate that all nodes are required to support. Thus, when a node detects a transmission it first tunes its hardware to the fixed rate to receive the PLCP header, and then uses the contents of the header to tune its hardware to the appropriate rate for the packet. The algorithm for choosing which rate to use for a date packet was intentionally unspecified in the 802.11 standard.

3.2 Implementation Details

In this section, we describe the implementation of RBAR into 802.11. We start by presenting the issues that were addressed, followed by a description of specific changes to the 802.11 protocol.

In 802.11, the duration of a reservation does not change. Thus, nodes that overhear a request may update their NAVs without regard to any further communication about the reservation. To facilitate dynamic rate changes we introduce the notion of *tentative* reservations. Tentative reservations serve to inform neighboring nodes that a reservation has been requested but that the duration of the *actual* reservation may differ. Thus, a tentative reservation serves as a placeholder until the actual reservation is transmitted. The purpose of tentative reservations is to allow the sender and receiver to reserve bandwidth so they can negotiate the appropriate modulation rate without interruption. Any node that receives a tentative reservation is required to treat it the same as an actual reservation with regard to later requests; that is, if a node overhears a tentative reservation it must update its NAV so that any requests that are directed to it and conflict with the tentative reservation are denied. Several techniques can be used to integrate tentative reservations into 802.11. One technique would be to use additional control messages, such as a second round of RTS/CTS messages, to announce the tentative reservation. Another technique would be to modify the existing frames. In this implementation, we choose to do the latter. In the remainder of this section, we discuss the details of the frame modifications.

The following are the proposed changes to the 802.11 frames.

1. The encoding of the 16-bit *duration* field in RTS and CTS packets is changed to a 4-bit *rate* subfield and a 12-bit *length* subfield. The *rate* subfield uses an encoding similar to the *rate* field in the 802.11a PLCP header, and the *length* subfield gives the size of the data packet in octets.
2. A new data frame format is introduced, where the standard MAC header is changed to include a CRC-16 *duration check sequence (DCS)*. The DCS covers the *frame control*, *duration*, *address 1*, and *address 2* fields of the header, which together form the *reservation subheader*. The new frame will only be used for STA to STA data frames that update a previously announced reservation.
3. The encoding of the *signal* field in the PLCP header is divided into two 4-bit *rate* subfields that are encoded identically to the *rate* subfield in item 1. The first subfield indicates the rate at which the subheader in item 2 are transmitted, and the second subfield indicates the rate at which the remainder of the packet is transmitted.

As mentioned earlier, receiver-based rate adaption requires that the sender and receiver be able to exchange rate information about the data packet while still providing reservation information to neighboring nodes. This is accomplished by encoding the rate and packet length into the *duration* fields of the packets, according to the format in item 1. The protocol then proceeds as follows. When node x has a packet to send to node y, it chooses rates for the control and data packets, as in the current standard. However, instead of calculating the duration of the reservation, x encodes the rate and the length of the data packet into the *duration* field of the RTS and sends it to y. Nodes that overhear the RTS use the encoded data along with the rate at which they received the RTS to calculate the anticipated length of the reservation, using the previous equation for T. This is possible because all of the values required to calculate T are known: they are either physical-layer constants, or are provided by the RTS (note, we assume that all control packets are sent at the same data rate). However, since the rate for the data packet may be changed by the receiver, T is treated as a *tentative* reservation. After y receives the RTS from x it chooses the best rate and encodes it into the *duration* field of the CTS, along with the size of the data packet provided by the RTS. Nodes that overhear the CTS use the encoded information to calculate the length of the reservation, similar to that done for the RTS, only using the equation for T'. This is the *actual* reservation. The *duration* fields in the remaining packets are encoded similiarly so nodes that heard the tentative reservation in the RTS are able to calculate the actual reservation.

The RBAR protocol requires that all nodes be able to reliably receive and decode portions of data packets that they overhear. This is necessary because certain fields in the header are now used to announce reservations. However, in 802.11, it cannot be assumed that all nodes will be able to receive a packet since data packets may be sent at a rate that is not required to be supported by all nodes; that is, it may not be in the $BSSBasicRateSet$. Furthermore, even if the packet can be received, the packet data cannot be trusted until after the entire packet has been received and checked using the frame check sequence. To address these problems, we propose a new MAC data frame format that groups header fields carrying reservation information into a subheader protected by a checksum. To ensure that this information is available to all nodes, we also propose modifying the PLCP header and transmission protocol to enable transmission of the subheader at a rate independent of the rest of the packet. For example, the subheader could be transmitted at a rate in the $BSSBasicRateSet$, while the remainder of the packet is transmitted at a different rate. The proposed MAC and PLCP header modifications are described in items 2 and 3. Changes to the protocols are described below.

In 802.11, the PLCP header contains an 8-bit *signal* field that designates the rate at which the payload is transmitted. This field is used by the physical layer as follows. When the MAC passes a packet down to the PLCP it also specifies the rate at which to send the packet. The physical layer then encodes this rate into the *signal* field of the PLCP header. When the packet is sent, the physical layer first transmits the PLCP header at the fixed PLCP rate, and then switches to the rate specified by the MAC for transmitting the remainder of the packet. The physical layer at the receiver then uses the *signal* field to determine which rate to switch to for receiving the payload.

To enable the use of an additional rate for the reservation subheader, we propose the following. Instead of a single 8-bit *signal* field, we subdivide the field into two 4-bit subfields, as described in item 2. The transmission protocol is changed 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 will encode the rates into the *signal* subfields and transmit the PLCP header. After the PLCP header has been transmitted, the physical layer will switch to the first rate for the subheader, and then to the second rate immediately after the subheader has been transmitted. Furthermore, the reservation subheader will be made available to the MAC immediately after the header has been checked, to allow the MAC to update its NAV.

4 Performance Evaluation

In this section we present the results of our performance evaluation of the Receiver-Based AutoRate (RBAR) protocol. The evaluation is based on simulation results, using the *ns-2* network simulator. As a basis of comparison, we also simulated Lucent's Autorate Fallback (ARF) protocol, as presented in [13]. Next, we give a brief overview of the ARF protocol, followed by a description of the simulation environment and methodology.

ARF is the rate adaption scheme used in Lucent's 802.11 WaveLAN II networking devices. It uses the presence or absence of MAC ACKs as indicators of channel quality, incrementally selecting higher or lower rates when the quality changes. The protocol is simple. If two consecutive ACKs are lost then the rate is reduced and a timer is started. The rate remains reduced until either ten consecutive ACKs are received or the timer expires. Upon expiration of the timer, the rate is increased for the next data packet (in our discussion, we refer to this packet as a *probe* packet, since it serves the purpose of probing the channel to see if conditions have improved.) If the ACK for the probe packet is lost, then the rate is immediately reduced and the timer is restarted; otherwise, the protocol continues at the new rate. In our simulations, the timeout was set to 100ms.

All of the results are based on simulations using a modified version of the *ns-2* network simulator from LBNL [6], with extensions from the CMU MONARCH project [4]. The extensions include a set of mobile ad-hoc network routing protocols and an implementation of BSD's ARP protocol, as well as an 802.11 MAC layer. Also included are mechanisms to model node mobility, using precomputed mobility patterns that are fed to the simulation at run-time. For more information about the extensions, we refer the reader to [4]. Additional modifications were made to model the modulation schemes shown in Figure 1, and Rayleigh fading. The Rayleigh fading implementation is described in the Appendix.

Our network model consisted of two identically configured nodes communicating on a single channel, using radios partially modeled after the commercially available Aironet 4800 2.4GHz DSSS IEEE 802.11b-based wireless network interfaces. Since we are only interested in each protocol's ability to adapt to changing channel conditions, we chose not to simulate the CCK modulation of 802.11b in favor of M-ary QAM. However, similar results can be expected for CCK and other modulation schemes. Thus, the set of modulation schemes used in the performance evaluation was the same as those shown in Figure 1: DBPSK (1Mbps), DQPSK (2Mbps), QAM16 (4Mbps), QAM64 (6Mbps), and QAM256 (8Mbps). The 802.11 basic rate, which is the rate at which control packets are transmitted, was set to 1Mbps DBPSK. Routing was static, and the TCP results used TCP-Reno with delayed acks. The remaining parameters were similar to those in [10].

For simulations involving mobility, one node was held in-place while the other was in constant motion along a straight path extending outward from the fixed node. The length of the path (250m) was chosen to extend beyond the effective range of the modulation schemes so that the channel would vary from very good to very bad during each traversal of the path. The intent was to stress the rate adaption schemes in a plausible usage scenario.

Unless otherwise stated, all simulation results are based on the average of 20 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 occured 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.

4.1 Impact of Node Speed

In this section, we consider the impact of node speed on the performance of the rate adaption protocols in a Rayleigh fading channel. In a fading channel, variations in the signal are induced at a rate that depends, in part, on the relative speed 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 (this duration is often called the *coherence time*, which is described in the Appendix). However, as the node speed increases, changes occur much more rapidly, decreasing the predictability of the channel. Thus, varying the mean node speed will enable us to evaluate the adaptability of the two protocols.

To observe the impact of mean node speed, we performed experiments for five different speeds: 2, 4, 6, 8, and 10 m/s. Results were generated for UDP and TCP connections carrying continuous data traffic. A CBR traffic source was used for the UDP experiments, and an FTP source with unlimited data was used for the TCP experiments. For each, data was generated at a rate of 8Mbps and sent in 1460 byte packets.

Results of the UDP experiments for each protocol are shown in Figure 4. Also shown are the results obtained for a fixed rate of 2Mbps (DQPSK), which was the best performer of the fixed rate measurements. For these results, notice that:

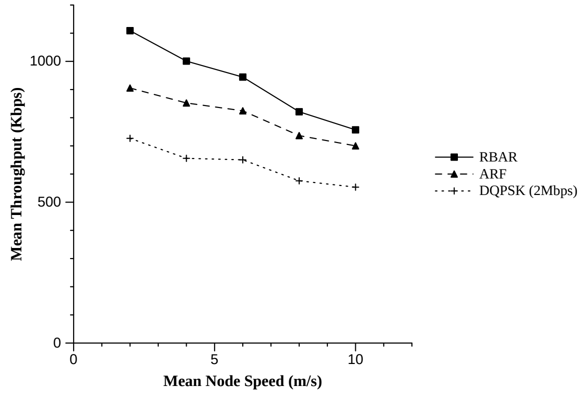


Figure 4: Performance for a CBR source generating traffic on a single UDP connection in a Rayleigh fading channel.

- RBAR outperformed ARF for all mean node speeds, with the performance improvement ranging from 8% (10 m/s) to 22% (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 spead increased. Recall that the simple channel quality prediction mechanism used in RBAR for these results works best when the channel coherence time (described in the Appendix) 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 (except 1Mbps, by a small margin). However, as the node speed increased, the coherence time shortened and the higher data rates were also affected, resulting in a decline in performance. We expect that this decline can be improved significantly with better channel quality prediction techniques, such as those in [2]. This is a topic of future work.
- Intuitively, ARF should perform at its best, relative to RBAR, when packet arrivals are frequent. This is because ARF tracks the channel quality using data packets as periodic probes. On the other hand, since RBAR uses the collision avoidance handshake to track the channel state on a *per-packet* basis, it should perform the same regardless of the traffic pattern. However, these results show that this is not necessarily the case. We suspect that this is because of the following reasons. If the channel is in a degraded state, ARF periodically probes the channel to see if conditions have improved by sending a data packet at the next higher rate. If conditions haven't improved, then there is a good chance that the packet will be dropped due to wireless errors. We observed that even though the steady stream of data packets improved ARF's ability to track and adapt to the changes in the channel state, the bandwidth wasted on dropped probe packets in regions where the highest rate was unavailable significantly degraded overall performance. For example, Table 5 shows the number of data packets received versus the number of packets dropped due to wireless errors, during one run of the simulator. In this instance, ARF lost 21% of its packets to wireless errors, versus RBAR's 10%.

Protocol	Received	Dropped
ARF	50,921	13,927
RBAR	62,755	7,034

Figure 5: UDP data packet statistics for one run of the simulator.

The remaining performance difference can be attributed to RBAR's more accurate rate adaption. We observed that, even with the steady traffic flow, RBAR was able to adapt more quickly to the changing channel conditions than ARF, which not only contributed to fewer dropped packets, but also resulted in better rate choices and, consequently, higher throughput. This can be seen in Table 5, which shows that RBAR was able to transmit nearly 5,000 more packets than ARF, and deliver nearly 12,000 more. Thus, not only was RBAR able to transmit data packets at a faster rate, but it was also able to deliver packets more reliably.

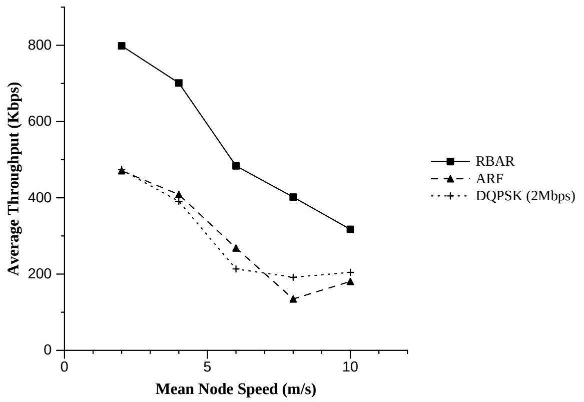


Figure 6: Performance for an FTP source with unlimited data generating traffic across a single TCP connection in a Rayleigh fading channel.

The results of the TCP experiments are shown in Figure 6. From these results, notice that:

- RBAR again outperformed ARF for all mean node speeds, with the performance improvement ranging from 77% (2 m/s) to 198% (8 m/s).
- The larger performance gain seen in the TCP results can, again, be attributed to TCP's sensitivity to packet loss. In the UDP results shown earlier, we noted that ARF had a packet loss percentage that was twice that of RBAR, for the example given. This higher loss percentage is the reason for RBAR's much better performance. Consider the following example, showing the TCP results for the same scenario used in Table 5. Table 7 shows the number of data packets received and dropped for each of the two protocols. Note that RBAR lost only 6% of its packets, versus 17% for ARF. Also note that, as observed in the previous case for UDP traffic, RBAR's ability to adapt more quickly and accurately to the state of the channel again results in a larger number of packets sent and received.

Protocol	Received	Dropped
ARF	38,088	7,834
RBAR	44,522	2,683

Figure 7: TCP data packet statistics for one run of the simulator.

4.2 Bursty Data Sources

In this section, we compare the performance of the RBAR and ARF protocols for bursty traffic.

First we consider the performance of the two protocols for traffic over a UDP connection. Here, the results we present are for 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. The results of these experiments are shown in Figure 8. Note that:

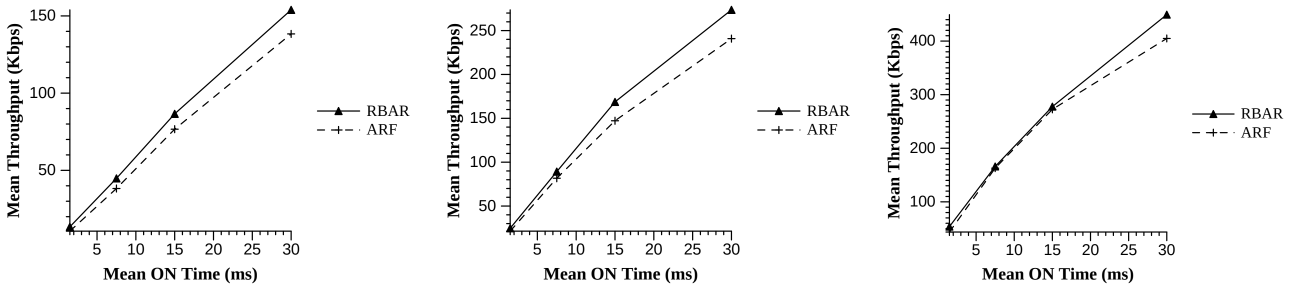
- RBAR outperforms ARF for all traffic scenarios simulated, with the improvement ranging from 2% to 26%.
- RBAR shows the most performance improvement (26%) when the traffic is the lightest ($\bar{\tau}_{off}$=1000ms, $\bar{\tau}_{on} = 1.5ms$).
- RBAR shows the least performance improvement (2%) when the traffic is moderate ($\bar{\tau}_{off}$=250ms, $\bar{\tau}_{on} = 7.5ms$).

The behavior illustrated by the latter two points is explained as follows. As mentioned in the previous section, intuition suggests that ARF should perform at its best, relative to RBAR, when the traffic is frequent. Previously, we showed that this was not always the case for very heavy traffic. Here, however, we see that this does appear to be the case for light to moderate traffic: RBAR's performance, relative to ARF, is at its best when the traffic is the lightest.

Next, we consider the performance of the two protocols for traffic over a TCP connection. Presented here are the results of a series of experiments for a Telnet source with interarrival times from the "tcplib" distribution generating traffic across a single TCP-Reno connection in a Rayleigh fading channel. The mean node speed was 2 m/s, and packet sizes were varied: 16, 64, 256, 512, 1024, and 1460 bytes. The results are shown in Figure 9, where Figure 9-a shows the actual measured throughput, and Figure 9-b shows throughput for both protocols as a percentage of ARF's throughput. Note that RBAR outperforms ARF for all experiments, with the improvement ranging from 29% for 1460 byte packets to 47% for 64 byte packets. This improvement is notably better than the improvement we observed for bursty UDP traffic. The reason can, again, be attributed to TCP's sensitivity to packet loss.

4.3 Overhead of RBAR Reservation Subheader

Finally, in this section we address the impact that the additional overhead of RBAR's reservation subheader has on performance.



(a) mean OFF time = 1s (b) mean OFF time = 500s (c) mean OFF time = 250ms

		Throughput (Kbps)			
$\bar{\tau}_{off}$ (ms)	Protocol	$\bar{\tau}_{on} = 1.5ms$	$\bar{\tau}_{on} = 7.5ms$	$\bar{\tau}_{on} = 15ms$	$\bar{\tau}_{on} = 30ms$
1000	ARF	10.6	38.2	76.7	138.3
	RBAR	13.4	44.8	86.6	153.9
500	ARF	21.4	81.7	147.2	240.8
	RBAR	24.7	89.2	168.7	273.6
250	ARF	43.7	162.9	272.5	404.9
	RBAR	53.7	166.0	277.7	449.3

Figure 8: Performance comparison for an ON/OFF Pareto source generating traffic on a single UDP connection in a Rayleigh fading channel. For clarity, the data used for the graphs on top is also shown in the table underneath.

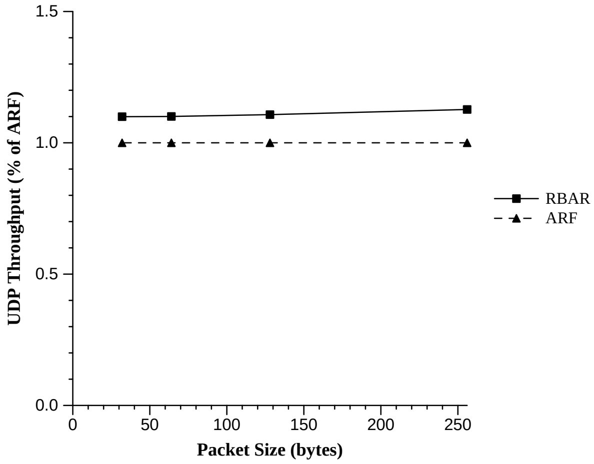


Figure 10: Performance comparison for a CBR source generating traffic on a single UDP connection in a Rayleigh fading channel.

To observe the impact of the overhead of the reservation subheader, we performed experiments for a single CBR data source with several small packet sizes: 32, 64, 128, and 256 bytes. In these experiments, data was generated at a rate of 8Mbps and sent across a single UDP connection in a Rayleigh fading channel. The results are presented in Figure 10, which shows the throughput for both protocols as a percentage of ARF's throughput. Note that, even for small packet sizes, the overhead of RBAR's reservation subheader does not appear to have a significant performance impact. Although there is a slight drop, RBAR still shows a 10% improvement over ARF.

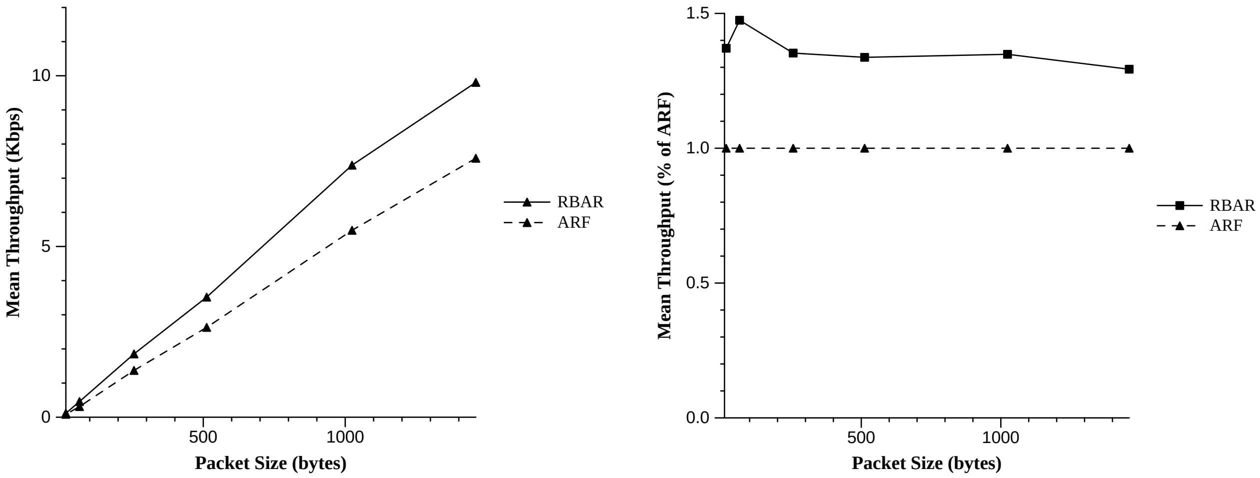
5 Protocol Variations

In this section we present a variation to the RBAR protocol.

Basic Access / Reservation Access Hybrid Protocol In the IEEE 802.11 standard, there is a variable that allows selective use of the DCF reservation access control protocol based on packet size. This variable, called the *RTSThreshold*, stores the maximum packet size for packets that *should not* be sent using reservation access. Instead, any packets that are smaller than the RTSThreshold will be sent using the DCF basic access control protocol (CSMA/CA). The objective is to reduce overhead by eliminating the RTS/CTS exchange for small packets. In situations where use of the RTSThreshold is desireable, a hybrid rate adaption scheme could be used where packets below the threshold are sent using a probing approach similar to ARF, while packets above the threshold are sent using RBAR. However, instead of sending data packets as probes, the probe packet would be sent using RBAR. This would reduce the overhead of lost probes, while still resulting in an overhead reduction for small packets.

6 Conclusion

In this paper, we addressed the topic of optimizing performance in wireless local-area networks using rate adaption. We presented a new approach to rate adaption, which differs from previous approaches in that it uses the RTS/CTS collision avoidance handshake to enable receiver-based rate adaption. 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 outperformed the existing protocol, with performance gains usually in the 20%-40% range.



(a) Measured throughput. (b) Relative Throughput.

Figure 9: Performance comparison for a Telnet source with interarrival times from the "tcplib" distribution generating traffic on a single TCP-Reno connection in a Rayleigh fading channel.

Acknowledgement

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 (see the Appendix), including the contribution of code implementing Jakes' method. The authors would also like to thank P. Abhiram for the helpful discussions on CCK. This work was supported, in part, by the National Science Foundation. Part of this work was performed while G. Holland and N. Vaidya were visiting Microsoft Research, Redmond, WA.

Appendix

A Simulation of Rayleigh Fading

This appendix describes the procedure used to simulate Rayleigh fading.

We first calculate the coherence time [17]The coherence time is the period over which the channel can be assumed to be effectively constant.

$$T_c(t) \approx \frac{9\lambda}{16\pi v(t)} \tag{1}$$

where $v(t)$ is the relative speed between the sender and receiver at time t, and $\lambda = c/f_c$ is the wavelenth of the carrier frequency f_c (c is the speed of light). The relative speed is calculated as follows. For some node i, let $\vec{p_i}$ be its position, $\vec{d_i}$ be its destination, and s_i be its speed. Its velocity is then

$$\vec{v_i} = \frac{s_i(\vec{d_i} - \vec{p_i})}{|\vec{d_i} - \vec{p_i}|} \tag{2}$$

where $||$ represents the magnitude of the vector difference (e.g. $|\vec{a} - \vec{b}| = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2 + (a_z - b_z)^2}$). Thus, the relative velocity between the sender s and the receiver r is

$$v = |\vec{v_r} - \vec{v_s}| \tag{3}$$

Next, we calculate the instantaneous bit error probability P_e that the packet will encounter over T_c. For DBPSK and DQPSK, this is [17]

$$P_e(t) = Q\left(\sqrt{\frac{2|\alpha(t)|^2 E_b}{N_o}}\right) \tag{4}$$

and for M-ary QAM [17]

$$P_e(t) \approx 4\left(1 - \frac{1}{\sqrt{M}}\right) Q\left(\sqrt{\frac{3|\alpha(t)|^2 log_2(M) E_b}{(M-1)N_o}}\right) \tag{5}$$

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. The procedure used to compute the E_b/N_o is given in the following section. The value of α is computed using Jakes' method, which is a common technique for simulating a signal with Rayleigh fading characteristics [12]. Jakes' method combines the output of a finite number of oscillators with Doppler shifted frequencies to produce a Rayleigh fading 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 and are computed as follows

$$x_c(t) = \frac{1}{\sqrt{N}} \sum_{n=1}^{N} \cos\beta_n \cos(\omega_n t + k\beta_n) \tag{6}$$

$$x_s(t) = \frac{1}{\sqrt{N}} \sum_{n=1}^{N} \sin\beta_n \cos(\omega_n t + k\beta_n) \tag{7}$$

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

$$\omega_n = \frac{2\pi v}{\lambda} \cos\left(\frac{\pi n}{2N+1}\right) \tag{8}$$

$$\beta_n = \frac{\pi n}{N} \tag{9}$$

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

$$|\alpha(t)| = \sqrt{x_c^2(t) + x_s^2(t)} \tag{10}$$

This procedure is then repeated until the entire packet has been processed.

A.1 Computation of E_b/N_o

To compute the E_b/N_o of the received signal, we calculate SNR and use the relation

$$\frac{E_b}{N_o} = SNR \cdot \frac{B_t}{R_b} \tag{11}$$

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

To compute the value of SNR, we use the following

$$SNR = 30 + 10 * \log_{10}(P_r) - (N_t + N_r + N_I) \tag{12}$$

where P_r is the power (in watts) of the received signal, N_t is the thermal noise (in dBm), N_r is the circuitry noise (in dBm), and N_I is the aggregate noise (in dBm) caused by concurrent transmissions that are too weak to cause a collision.

P_r is computed using the Friis free space path loss equation [17]

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \tag{13}$$

where d is the distance (in meters) between the sender and receiver, P_t is the transmit power (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 miscellaneous system loss factor (we assume $L = 1$). N_t is calculated using

$$N_t = 30 + 10 * \log_{10}(kTB_t) \tag{14}$$

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. For N_r, we use a value provided by Intersil for their Prism I chipset. Finally, we compute N_I using

$$N_I = 30 + 10 * \log_{10}\left(\sum_{i=1}^{n} P_i\right) \tag{15}$$

where P_i is the power (in watts) of the i^{th} transmission.

References

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

[2] 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*, vol. 17, pp. 1244–1256, July 1999.

[3] V. Bharghavan, "MACAW: A media access protocol for wireless LAN's," in *Proceedings of SIGCOMM'94*, (London), 1994.

[4] 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*, pp. 85–97, Oct. 1998.

[5] M. Eyugoglu, C. Forney, P. Dong, and G. Long, "Advanced modulation techniques for v.fast," in *Eur. Trans. Telecommun.*, vol. 4, pp. 243–256, May-June 1993.

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

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

[8] D. L. Goeckel, "Adaptive coding for time-varying channels using outdated fading estimates," *IEEE Transactions on Communications*, vol. 47, pp. 844–855, June 1999.

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

[10] G. Holland and N. Vaidya, "Analysis of tcp performance over mobile ad hoc networks," in *Proceedings of MOBICOM'99*, (Seattle), 1999.

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

[12] W. C. Jakes, ed., *Microwave Mobile Communications.* IEEE Press, 1994.

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

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

[15] X. Qiu and K. Chawla, "On the performance of adaptive modulation in cellular systems," *IEEE Transactions on Communications*, vol. 47, pp. 884–895, June 1999.

[16] R. Ramanathan and M. Steenstrup, "Hierarchically-organized, multihop mobile wireless networks for quality-of-service support," *Mobile Networks and Applications*, vol. 3, pp. 101–119, June 1998.

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

[18] 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*, vol. COM-23, no. 12, pp. 1417–1433, 1975.

[19] 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*, vol. 47, pp. 1134–1147, November 1998.

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