Reversing the Collision-Avoidance Handshake in Wireless Networks

J.J. Garcia-Luna-Aceves and Asimakis Tzamaloukas *
Computer Engineering Department
Baskin School of Engineering
University of California, Santa Cruz, California 95064
{jj, jamal}@cse.ucsc.edu

Abstract

Many medium-access control (MAC) protocols for wireless networks proposed or implemented to date are based on collision-avoidance handshakes between sender and receiver. In the vast majority of these protocols, including the IEEE 802.11 standard, the handshake is sender initiated, in that the sender asks the receiver for permission to transmit using a short control packet, and transmits only after the receiver sends a short clear-to-send notification. We analyze the effect of reversing the collision-avoidance handshake, making it receiver initiated and compare the performance of a number of these receiver-initiated protocols with the performance of protocols based on sender-initiated collision avoidance. The receiver-initiated protocols we present make use of carrier sensing, and are therefore applicable to either baseband or slow frequency-hopping radios in which an entire packet can be sent within the same frequency hop (which is the case of FHSS commercial radios that support IEEE 802.11). It is shown that the best-performing MAC protocol based on receiver-initiated or sender-initiated collision avoidance is one in which a node with data to send transmits a dual-purpose small control packet inviting a given neighbor to transmit and asking the same neighbor for permission to transmit.

1 Introduction

There is a large body of work on the design of MAC (medium access control) protocols for wireless networks with hidden terminals. Kleinrock and Tobagi [7] identified the hidden-terminal problem of carrier sensing, which makes carrier-sense multiple access (CSMA) perform as poorly as the pure ALOHA protocol when the senders of packets cannot hear one another and the vulnerability period of packets becomes twice a packet length. The BTMA (busy tone multiple access) protocol was a first attempt to solve the hidden-terminal problem by introducing a separate busy tone channel [12]. The same authors proposed SRMA (split-channel reservation multiple access) [13], which attempts to avoid collisions by introducing a control-signal handshake between the sender and the receiver. A station that needs to transmit data to a receiver first sends a request-to-send (RTS) packet to the receiver, who responds with a clear-

to-send (CTS) if it receives the RTS correctly. A sender transmits a data packet only after receiving a CTS successfully. ALOHA or CSMA can be used by the senders to transmit RTSs.

Several variations of this scheme have been developed since SRMA was first proposed, including MACA [6], MACAW [1], IEEE 802.11 [5], and FAMA [3]. These examples of MAC protocols, and most protocols based on collision-avoidance handshakes to date are sender-initiated, in that the node wanting to send a data packet first transmits a short RTS asking permission from the receiver. In contrast, in the MACA by invitation (MACA-BI) protocol [11], the receiver polls one of its neighbors asking if it has a data packet to send. A receiver-initiated collision avoidance strategy is attractive because it can, at least in principle, reduce the number of control packets needed to avoid collisions. However, as we show in this paper, MACA-BI cannot ensure that data packets never collide with other packets in networks with hidden terminals.

In this paper, we present MAC protocols with receiver-initiated collision avoidance that do provide *correct collision avoidance*, i.e., prevent data packets addressed to a given receiver from colliding with any other packets at the receiver. We analyze the effect of reversing the collision-avoidance handshake used to eliminate the hidden-terminal problem of carrier sensing. Our study of receiver-initiated collision avoidance focuses on single-channel networks with asynchronous transmissions, but many of our results extrapolate to networks with multiple channels.

The key contributions of this paper are recasting collision avoidance dialogues as a technique that can be controlled by senders, receivers, or both; showing that receiver-initiated collision avoidance can be even more efficient than sender-initiated collision avoidance; and presenting a method for proving that a receiver-initiated collision avoidance strategy works correctly.

We use a fully-connected network topology to discern the relative performance advantages of different protocols. We opted to focus on fully-connected networks in our analysis because of two reasons: (a) it allows us to use a short analysis that can be applied to several protocols; and (b) our focus on protocols that provide correct collision avoidance means that the relative performance differences in a fully-connected network are very much the same when networks with hidden terminals are considered. In particular, results presented for FAMA protocols [2, 3] indicate that, in a network with hidden terminals, the performance of a MAC protocol with correct collision avoidance is almost identical to the performance of the same protocol in a fully-connected network if the vulnerability period of a control packet is made proportional to the length of the entire packet. This is intuitive, if a MAC protocol prevents data packets from colliding with other packets in any type of topology, hidden terminals can degrade the protocol's performance from that obtained in a fully-connected network only to the extent that control packets used to prevent data collisions are subject to

1. REPORT DATE **1999**	2. REPORT TYPE	3. DATES COVERED **00-00-1999 to 00-00-1999**

4. TITLE AND SUBTITLE **Reversing the Collision-Avoidance Handshake in Wireless Networks**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of California at Santa Cruz,Department of Computer Engineering,Santa Cruz,CA,95064**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**

13. SUPPLEMENTARY NOTES **The original document contains color images.**

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES **12**	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**			

additional interference caused by the fact that nodes cannot sense the transmissions of control packets by hidden sources.

The receiver-initiated protocols we introduce in this paper require that nodes accomplish carrier sensing. This can be done with baseband radios and today's commercial slow frequency hopping radios, in which complete packets are sent in the same frequency hop. The receiver-initiated protocols we present, as well as the sender-initiated protocols introduced in the past based on carrier sensing and a single channel (e.g., FAMA [3]), do not really apply to DSSS (direct sequence spread spectrum) radios, because radios capture none or one of multiple overlapping transmissions depending on the proximity and transmission power of the sources. Fortunately, there are many commercial radios, specially at the 2.4GHz band, which can make use of our collision-avoidance approach.

Section 2 introduces fundamental aspects of receiver-initiated collision-avoidance handshake, and Section 3 presents a number of MAC protocols based on receiver-initiated collision-avoidance. Section 4 proves that, in the absence of fading, all these protocols solve the hidden-terminal problem, i.e., they eliminate collisions of data packets. Section 5 analyzes the throughput of these protocols in fully-connected networks. Our analysis shows that receiver initiated multiple access with dual-use polling (RIMA-DP) is the most efficient approach among all the sender- and receiver-initiated MAC protocols proposed to date for single-channel networks with asynchronous transmissions.

2   Receiver-Initiated Collision Avoidance

Critical design issues in receiver-initiated MAC protocols over a single channel are: (a) whether or not to use carrier sensing, (b) how to persist transmitting packets, (c) how to resolve collisions, and (d) deciding how a receiver should poll its neighbors for data packets.

Carrier sensing has been shown to increase the throughput of sender-initiated collision avoidance tremendously [2]; furthermore, carrier sensing has also been shown to be necessary to avoid collisions of data packets in sender-initiated collision avoidance over single-channel networks in which transmissions occur in an asynchronous way, i.e., without time slotting [3].

We describe all receiver-initiated schemes assuming carrier sensing and asynchronous transmissions. To simplify the analysis of the protocols, we also assume non-persistent carrier sensing, which has been shown to provide better throughput characteristics than persistent disciplines for CSMA and CSMA/CD [8] at high loads. Furthermore, our treatment of receiver-initiated collision avoidance assumes simple back-off strategies; however, the benefits of using sophisticated back-off strategies or collision resolution algorithms has been analyzed for a number of sender-initiated MAC protocols [1, 4], and it should be clear that the same schemes could be adopted in any of the receiver-initiated approaches we address in this paper.

In sender-initiated collision avoidance, a node sends a request-to-send packet (RTS) whenever it has data to send and, in protocols using carrier sensing, the channel is free. However, deciding how to send polling packets in receiver-initiated protocols is not as immediate as sending transmission requests in sender-initiated protocols; furthermore, as we show in this paper, the polling discipline chosen determines to a large extent the performance of the protocol. A polling rate that is too small renders low throughput and long average delays, because each sender with a packet to send is slowed down by the polling rate of the receiver. Conversely, a polling rate that is too high also renders poor performance, because the polling packets are more likely to collide with each other and no source gets polled.

The polling discipline used in a receiver-initiated MAC protocol can be characterized by three different factors:

- Whether or not the polling rate is independent of the data rate at polling nodes,

- Whether the poll is sent to a particular neighbor or to all neighbors,

- Whether the polling packet asks for permission to transmit as well.

In terms of the relationship between the polling rate and the data rate, we can categorize polling disciplines in two major classes: independent polling and data-driven polling.

With independent polling, a node polls its neighbors at a rate that is independent of the data rate of the node or the perceived data transmission rate of its neighbors. In contrast, with data-driven polling, a node attempts to poll its neighbors at a rate that is a function of the data rate with which it receives data to be sent, as well as the rate with which the node hears its neighbors send control and data packets. The specification of the MACA-BI protocol by Talucci et al. [11] assumes this type of polling. Throughout the rest of the paper, we assume data-driven polling, because it is very difficult in a real network to determine a good independent polling rate by the receivers, and because data-driven polling is far simpler to analyze.

In practice, to account for data rate differences at nodes and to eliminate the possibility of a data-driven polling discipline never allowing a node to receive data, a protocol based on data-driven polling should send a poll based on its local data to be sent or after a polling timeout elapses without the node having any packet to send to any neighbor.

The intended audience of a polling packet can be a single neighbor, a subset of neighbors, or all the neighbors of a node. A large audience for a poll packet introduces the possibility of contention of the responses to the poll, and either the collisions of responses need to be resolved, or a schedule must be provided to the poll audience instructing the neighbors when to respond to a poll.

The intent of a polling packet can be simply to ask one or more neighbors if they have data to send to the polling node, or it can both ask for data and permission to transmit in the absence of data from the polled neighbors. Intuitively, the latter approach should have better channel utilization, because data will be sent after every successful handshake, and more data per successful handshake are sent as traffic load increases even if the polled node does not have data for the polling node. We also note that a polling packet asking for data from a neighbor could allow the polled node to send data to *any* destination, not just to the polling node; however, this strategy would not work efficiently in multihop networks, because there is no guarantee that the recipient of a data packet who did not ask for it will receive the transmission in the clear.

It is clear that polls that specify transmission schedules can address the three key functions of a polling discipline that we have just discussed. In this paper, however, we concentrate on single-node polling and broadcast polling only. Receiver-initiated protocols based on schedules is an area of future research.

3   Receiver-Initiated Protocols

This section introduces new MAC protocols based on receiver initiated collision avoidance and relate them to the taxonomy of polling disciplines presented in Section 2. To our knowledge, these protocols are the first based on receiver-initiated collision avoidance that eliminate the collisions of data packets with any other control or data packets in the presence of hidden terminals.

For simplicity, we describe the new MAC protocols without the use of acknowledgments (ACKs); in practice, ACKs will be used. However, it should be clear that, because the protocols support correct collision avoidance, an acknowledgment to each data packet

can be sent collision-free by the receiver immediately after it processes the data packet. The only caveat is that the time that a node must back off to let data flow without collisions must include the time needed for the sender to receive the acknowledgment in the clear.

3.1 Protocols with Simple Polling

3.1.1 MACA-BI

The original MACA-BI [11] protocol uses a ready-to-receive packet (RTR) to invite a node to send a data packet. A node is allowed to send a data packet only if it has previously received an RTR, whereas a node that receives an RTR that is destined to a different node has to back off long enough for a packet to be sent in the clear.

According to the description of MACA-BI, a polled node can send a data packet intended to the polling node or any other neighbor. In a fully-connected network, whether the data packet is sent to the polling node or not is not important, because all the nodes must back off after receiving an RTR in the clear. However, this is not the case in a network with hidden terminals.

By means of two simple examples, we can show that MACA-BI does not prevent data packets sent to a given receiver from colliding with other data packets sent concurrently in the neighborhood of the receiver. The first example illustrates the fact that, in order to avoid the transmission of data packets that the intended receiver cannot hear because of other colliding data packets, a polled node should send data packets only to the polling node. The second example illustrates the possibility that collisions of data packets at a receiver may occur because the receiver sent an RTR at approximately the same time when data meant for another receiver starts arriving.

In Fig. 1, nodes a and d send RTRs to nodes b and e at time t_0, respectively. This prompts the polled nodes to send data packets at time t_1; the problem in this example occurs when at least one of the polled nodes sends a data packet addressed to c, which cannot hear either packet.
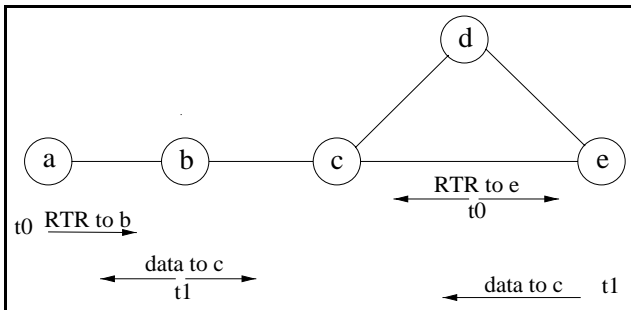

Figure 1: Data packets colliding in MACA-BI when packet is not sent to polling node

In the example shown in Fig. 2, node a sends an RTR to b at time t_0. This RTR makes node b start sending data to node a at time t_1 which in order to provide good throughput must be larger than γ seconds, where γ is the length of an RTR. At time t_2 node c starts sending an RTR to node d. Because of carrier sensing, t_2 must be within τ seconds (maximum propagation delay) of t_1. In this example, after receiving node c's RTR, node d replies with data that must start arriving at node c at time t_3. Because the maximum propagation delay is τ, it must be true that $t_3 \leq t_2 + \gamma + 2\tau \leq t_1 + \gamma + 3\tau$. Hence, if data packets last longer than $\gamma + 3\tau$ seconds, the data packets from b and d collide at node c. In practice, data packets must be much longer than RTRs to provide good throughput, and it thus follows that MACA-BI cannot prevent all data packets from experiencing collisions.
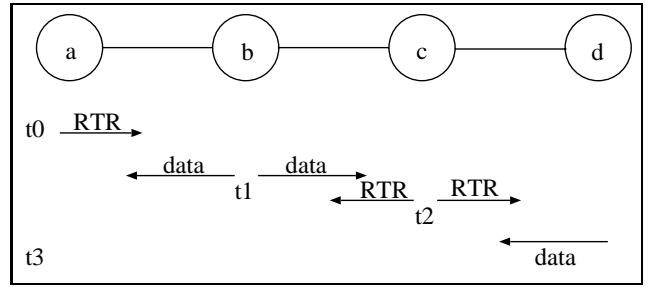

Figure 2: Data packets colliding in MACA-BI due to RTR not being heard

3.1.2 RIMA-SP

The above problems in MACA-BI went unnoticed in the specification by Talucci et al. [11]. To make the RTR-data handshake in MACA-BI collision free, the following two minor modifications are required:

- The polled node should transmit data packets only if they are addressed to the polling node.

- A new control signal is also required, which we call No-Transmission-Request (NTR), and an additional collision-avoidance waiting period of ξ seconds is required at a polled node prior to answering an RTR. During that period, if any channel activity is heard, the receiver (polling node) that originated an RTR sends an NTR telling the polled node not to send any data. Otherwise, if nothing happens during the waiting period, the polled sender transmits its data, if it has any to send to the polling node.

We call the protocol resulting from modifying MACA-BI with the above two rules RIMA-SP (receiver initiated multiple access with simple polling). Fig. 3 illustrates the operation of RIMA-SP. The complete proof that RIMA-SP provides correct collision avoidance when $\xi = \tau$ is given in Section 4.

In RIMA-SP, every node initializes itself in the START state, in which the node waits twice the maximum channel propagation delay, plus the hardware transmit-to-receive transition time (ϵ), before sending anything over the channel. This enables the node to find out if there are any ongoing transmissions. After a node is properly initialized, it transitions to the PASSIVE state. In all the states, before transmitting anything to the channel, a node must listen to the channel for a period of time that is sufficient for the node to start receiving packets in transit.

If a node x is in the PASSIVE state and senses carrier, it transitions to the REMOTE state to defer to ongoing transmissions. A node in REMOTE state must allow enough time for a complete successful handshake to take place, before attempting to transition from remote state.

Any node in PASSIVE state that detects noise in the channel must transition to the BACKOFF state. If node x is in PASSIVE state and obtains an outgoing packet to send to neighbor z, it transitions to the RTR state. In the RTR state, node x uses non-persistent carrier sensing to transmit an RTR. If node x detects carrier when it attempts to send the RTR, it transitions to the BACKOFF state, which makes the node back off immediately for a sufficient amount of time to allow a complete handshake between a sender-receiver pair to occur; otherwise, x sends its RTR.

If node z receives the RTR correctly and has data for x, it waits for ξ seconds. If during the waiting period there is no activity in the channel, node z transitions to the XMIT state, where it transmits a data packet to x (Fig. 3(a)); otherwise, node z assumes that
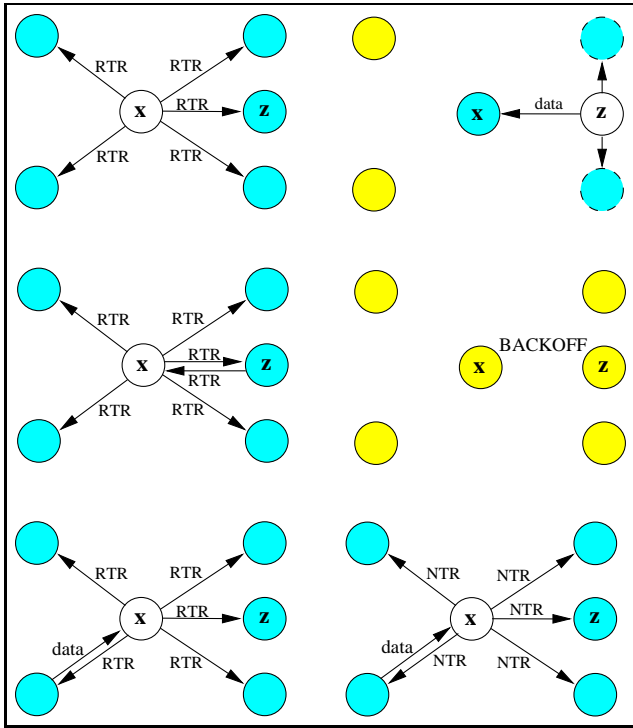
Figure 3: RIMA-SP illustrated

there was a collision and transitions to the BACKOFF state to allow floor acquisition by some other node. After sending its RTR, node x senses the channel. If it detects carrier immediately after sending its RTR, node x assumes that a collision or a successful data transfer to a hidden node is taking place. Accordingly, it sends a No transmission Request (NTR) to z to stop z from sending data that would only collide at x (Fig. 3(b)).

When multiple RTRs are transmitted within a one-way propagation delay a collision takes place and the nodes involved have to transition to the BACKOFF state and try again at a later time chosen at random, as shown in Fig. 3(b).

Node x determines that its RTR was not received correctly by z after a time period equal to the maximum round-trip delay to its neighbors plus turn-around times and processing delays at the nodes, plus the waiting period ξ. After sending its RTR, node x listens to the channel for any ongoing transmission. Because of non zero propagation delays, if node x detects carrier immediately after transmitting its RTR, it can conclude that it corresponds to a node other than z, which would take a longer time to respond due to its need to delay its data to x to account for turn-around times. [1]

The lengths of RTRs and NTRs are the same. The same argument used in [2] to show that the length of an RTS must be longer than the maximum propagation delay between two neighbors to ensure correct collision avoidance can be used to show that RTRs and NTRs must last longer than a maximum propagation delay. In ad-hoc networks in ISM bands, propagation delays are much smaller compared with any packet that needs to be transmitted.

To reduce the probability that the same nodes compete repeatedly for the same receiver at the time of the next RTR, the RTR specifies a back-off-period unit for contention. The nodes that must enter the BACKOFF state compute a random time that is a multiple of the back-off-period unit advertised in the RTR. The simplest case consists of computing a random number of back-off-period units using a uniformly distributed random variable from 1 to d,

where d is the maximum number of neighbors for a receiver. The simplest back-off-period unit is the time it takes to send a small data packet successfully.

3.2 Protocols with Dual-Use Polling

The collision avoidance strategy described for RIMA-SP can be improved by increasing the probability that data will follow a successful RTR, without violating the rule that data packets should be transmitted only if they are addressed to the polling nodes. A simple way to achieve this with data-driven polling is to make an RTR entry both a request for data from the polled node, and a transmission request for the polling node to send data. The RIMA-DP (receiver-initiated multiple access with dual-purpose polling) protocol does exactly this. Fig. 4 illustrates the modified collision avoidance handshake to permit the polling node to either receive or send data without collisions.

As Fig. 4(a) illustrates, a key benefit of the dual-use polling in RIMA-DP is that both polling and polled nodes can send data in a round of collision avoidance. This is possible because the RTR makes all the neighbors of the polling node back-off, and the data from the polled node make all its neighbors back-off, which can then be used by the polling node to send its data.

RIMA-DP gives transmission priority to the polling nodes. When a node z is polled by node x and has data for node x, z waits ξ seconds before sending a data packet. In contrast, if the polled node does not have data for x, it immediately sends a CTS (Clear-To-Send packet) to x. This permits a polling node x exposed to a neighbor sending data to hear part of that neighbor's data packet after sending its RTR; in such a case, node x can send an NTR to the polled node to cancel its RTR. Section 4 shows that this prevents collisions of data packets, provided that z waits for $\xi > \gamma + 7\tau$ seconds before sending any data after being polled and the length of a CTS is 2τ seconds longer than the length of an RTS. As in RIMA-SP, the lengths of RTRs and RTSs are the same.

As in RIMA-SP, every node starts in the START state and transitions to to the PASSIVE state when it is initialized. If a node x is in the PASSIVE state and senses carrier, it transitions to the REMOTE state to defer to ongoing transmissions. A node in REMOTE state must allow enough time for a complete successful handshake to take place, before attempting to transition from remote state.

Any node in PASSIVE state that detects noise in the channel must transition to the BACKOFF state where it must allow sufficient time for complete successful handshakes to occur. If node x is in PASSIVE state and obtains an outgoing packet to send to neighbor z, it transitions to the RTR state. In the RTR state, node x behaves as in RIMA-SP.

If node z receives the RTR correctly and has data for x, it waits for ξ seconds before sending a data packet to x. If during the waiting period there is no activity in the channel, node z transitions to the XMIT state, where it transmits a data packet to x. Otherwise, z assumes a collision or data transfer to a hidden node and goes to the BACKOFF state. If z has no data for x, it sends a CTS to x immediately.

If node x detects carrier immediately after sending an RTR, it defers its transmission attempt and sends an NTR to the node it polled. The CTS length, which is τ seconds longer than an RTR, forces polling nodes that send RTRs at about the same time when a polled node sends a CTS to detect carrier from the CTS and stop their attempt to send or receive data. Any node other than x receiving the CTS for x transitions to the BACKOFF state. When node x receives the CTS from z, it transitions to the XMIT state and transmits a data packet to z.

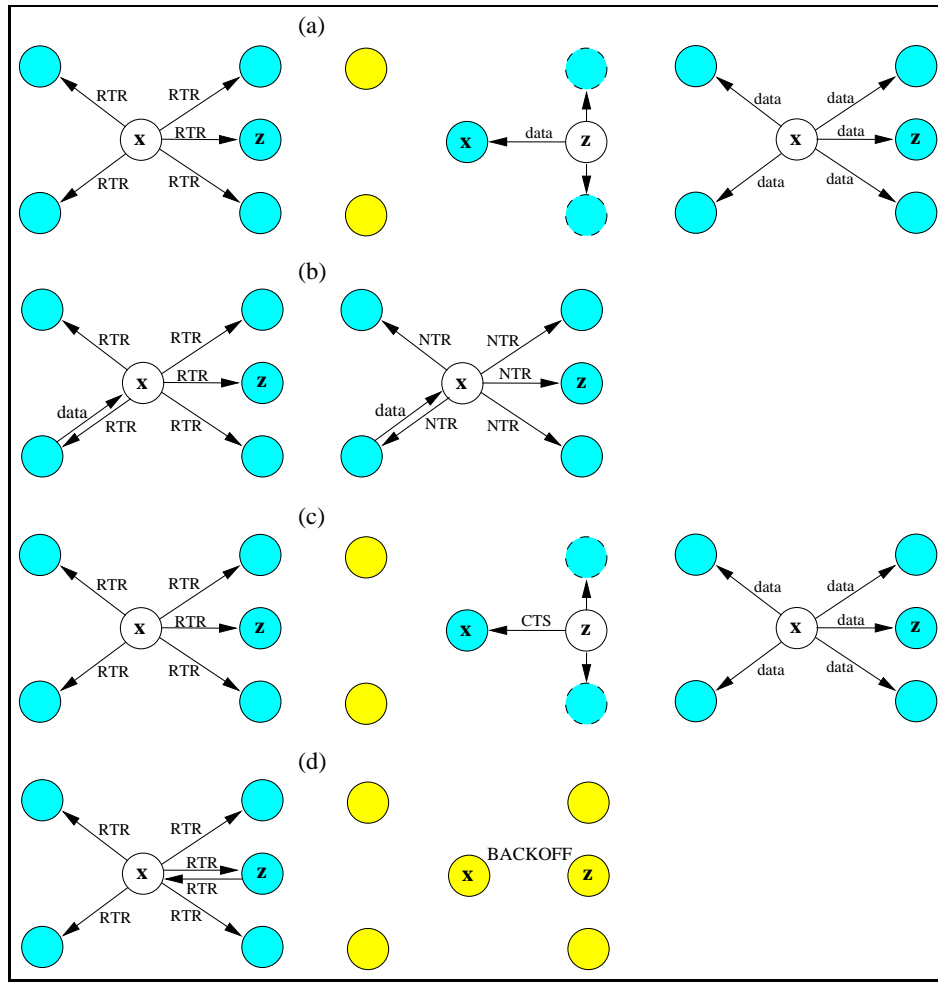[1] Our analysis assumes 0 turn-around times and 0 processing delays for simplicity.

Figure 4: RIMA-DP illustrated

3.3 Protocols with Broadcast Polling

Contrary to the prior two approaches, an RTR can be sent to multiple neighbors. We describe a modification of RIMA-SP based on this variant.

A node broadcasts an RTR only when there is a local data packet (data-driven polling). Only after a node has received an invitation, it is allowed to send any data. Because a poll broadcast to all the neighbors of a node can cause multiple nodes to attempt sending data to the polling node, an additional control packet is needed to ensure that transmissions that collide last a short period and do not carry user data. Accordingly, a polled node sends a short RTS (Ready-To-Send packet) before sending data. Furthermore, after sending its RTS, the polled node must wait for ξ seconds to allow the polling node to send an NTR when collisions of RTSs occur at the polling node. We call this protocol RIMA-BP (Broadcast Polling).

It can be shown that RIMA-BP provides correct collision avoidance if $\xi = 4\tau$. Fig. 5 illustrates the receiver-initiated handshake of RIMA-BP. As it is shown in the figure, the key difference with RIMA-SP is the use of an RTS prior to the transmission of a data packet.

4 Correct Collision Avoidance in RIMA protocols

Theorems 1 and 2 below show that RIMA-SP and RIMA-DP ensure that there are no collisions between data packets and any other transmissions. A similar proof to that of Theorem 1 can be used to show that RIMA-BP provides correct collision avoidance if $\xi = 4\tau$. The following assumptions are made to demonstrate correct collision avoidance in RIMA protocols [3]:

A0) A node transmits an RTR that does not collide with any other transmissions with a non-zero probability.

A1) The maximum end-to-end propagation time in the channel is $\tau < \infty$.

A2) A packet sent over the channel that does not collide with other transmissions is delivered error free with a non-zero probability.

A3) All nodes execute a RIMA protocol correctly.

A4) The transmission time of an RTR and a CTS is γ, the transmission time of a data packet is δ, and the hardware transmit-to-receive transition time is zero; furthermore, $2\tau < \gamma \leq \delta < \infty$.

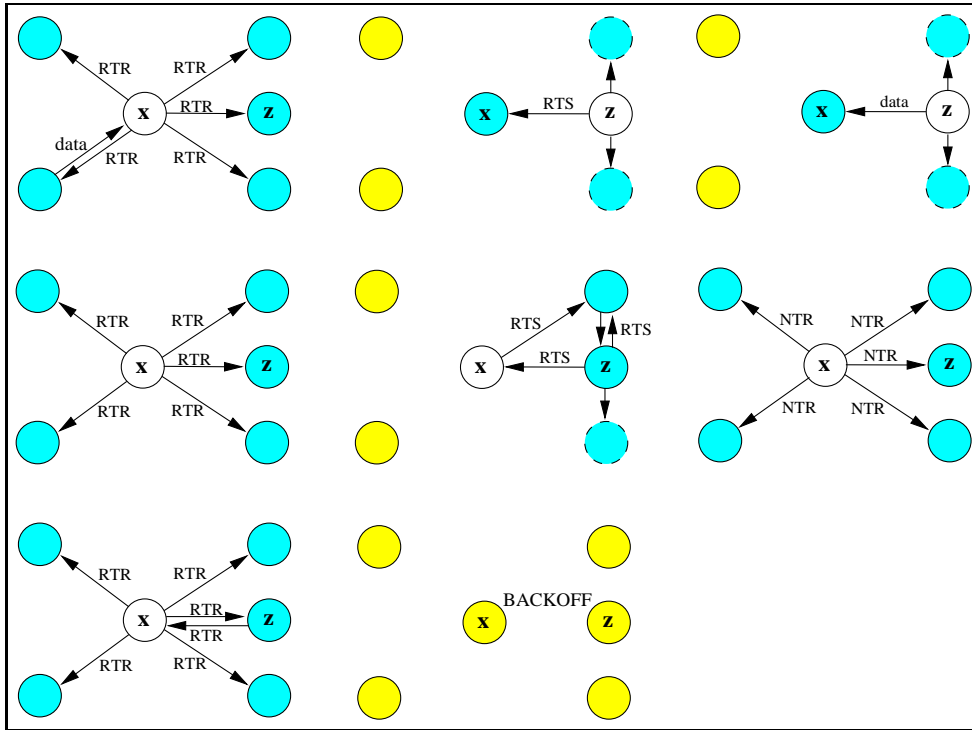A5) There is no capture or fading in the channel.

Figure 5: RIMA-BP illustrated

The approach used to show that a collision-avoidance protocol works correctly, i.e., that it prevents data packets from colliding with any type of packets, consists of showing that, once a data packet is sent by a node, the intended receiver obtains the packet without interference. In any receiver-initiated collision avoidance scheme, we have polling nodes and polled nodes, and we must show that any interference at the polled node prevents it from sending data, while any detected interference at a polling node that has sent an RTR forces the node to jam the polled node to prevent data from arriving and collide at the polling node.

Because interference must be detected by polling and polled nodes, the receiver-initiated collision-avoidance protocols we are describing require carrier sensing. The ability to detect carrier is applicable only to baseband radios or slow frequency hopping radios, and periods of fading disrupt any type of collision avoidance dialogue, i.e., data packets may experience collisions in the presence of fading.

Assuming zero processing and turn-around delays is done for convenience; however, the same type of proofs, with adjusted parameters, apply for non-zero hardware delays.

Theorem 1 *RIMA-SP provides correct collision avoidance in the presence of hidden terminals, provided that $\xi = \tau$.*

Proof: Consider a polling node A and a polled node X and assume that A sends an RTR at time t_0. If X does not receive the RTR correctly due to interference from any neighbor hidden from A, it does not send any data. Else, X waits $\xi = \tau$ seconds after receiving A's RTR before sending its data to A. Because propagation delays are positive, the earliest time when X can start sending data to A is $t_1 > t_0 + \gamma + \tau$. On the other hand, if A detects interference immediately after sending its RTR, i.e., at time $t_0 + \gamma$, it starts sending an NTR to X, and this NTR must start arriving at X no later than $t_2 \le t_0 + \gamma + \tau$. Because $t_1 > t_2$, it follows that X cannot send data to A that can collide with any other transmission arriving at A. □

Theorem 2 *RIMA-DP provides correct collision avoidance in the presence of hidden terminals, provided that $\xi > \gamma + 7\tau$ and a CTS lasts 2τ seconds longer than an RTR.*

Proof: Consider a polling node A and a polled node X and assume that A sends an RTR to X at time t_0. If A is exposed to a polled node Y sending data or a CTS, A must have started its RTR within τ seconds of Y's start of transmission; for otherwise A would have detected carrier caused by Y and would not have sent its RTR. Accordingly, because data packets and CTSs are at least $\gamma + 2\tau$ in length, A must detect carrier from Y's transmission immediately after sending its RTR, which forces node A to send an NTR at time $t_0 + \gamma$. Therefore, regardless of what happens at the polled node X, the polling node A must send an NTR immediately following its RTR and back-off, and cannot send any data if there is any exposed polled node sending or requesting data.

Assume that A is not exposed to a polled node sending or requesting data, but is exposed to a polling node B. Let A poll node X and B poll node Y.

For both A and B to send their RTRs they must do so within τ seconds of each other, for otherwise one of the two would detect carrier and back-off. For X to send any packet to A (data or CTS), A's RTR must be received collision free at X. X can receive A's RTR successfully no earlier than $t_1 > t_0 + \gamma$, because propagation delays are positive. If X has no data for A, it sends a CTS immediately to X. This CTS can arrive in its entirety at A no earlier than $t_2 > t_0 + 2\gamma + 2\tau$, because a CTS lasts $\gamma + 2\tau$ and propagation delays are positive. The same is the case for the node Y polled by B. Therefore, if both X and Y send data, Y must be hidden from A and X must be hidden from B, and no data packets collide.

There are three cases to consider now. In one case one polled node sends a CTS and the other sends data, in another case both polled nodes send data, and in the last case each polled node sends a CTS.

Without loss of generality, assume that Y sends a CTS and X is ready to send data. The earliest time when X can start sending

data is $t_2 > t_0 + \gamma + \xi$, because propagation delays are positive. On the other hand the latest time when A must start receiving data sent by B, after B receives the CTS from Y, is $t_3 \le (t_0 + \tau) + (2\gamma + 3\tau) + \tau$. The first $t_0 + \tau$ stems from the fact that B can send its RTR up to τ seconds after A starts sending its RTR. The second term $(2\gamma + 3\tau)$ corresponds to B's RTR and Y's CTS plus the corresponding maximum propagation delays, and the last term τ is the maximum propagation delay of data from B to A. Accordingly, A must detect carrier and starts sending its NTR at time t_3, and X must detect carrier from A's NTR at time $t_4 \le t_3 + \tau \le t_0 + 2\gamma + 7\tau$. Because $t_2 > t_0 + \gamma + \xi$ and $\xi > \gamma + 7\tau$, it follows that X cannot send its data to A and no collision occurs.

If both X and Y send data or CTSs after being polled no collision occurs, because we have shown that X must be hidden from B and Y must be hidden from A. Therefore, RIMA-DP provides collision avoidance correctly. $\square$

5 Approximate Throughput Analysis

The objective of our analysis is to contrast the various polling polices introduced for RIMA protocols, and to compare them against sender-initiated collision avoidance protocols, namely, MACA [6] and FAMA-NCS [3]. The choice of protocols was made because MACA is the simplest sender-initiated collision avoidance protocol and FAMA-NCS is the best performing MAC protocol based on sender-initiated collision-avoidance reported to date.

Our analysis shows a number of interesting results. By making collision-avoidance a joint effort by sender and receiver (as we do in RIMA-DP), a much better performance is obtained than what can be achieved with FAMA-NCS; this should be expected, because dual-use polling doubles the opportunity for collision-free data to be sent. Protocols based on simple polling (RIMA-SP) perform much better than MACA, but their performance degrades with the number of neighbors that a node has. RIMA-SP exhibits lower performance than FAMA and RIMA-DP, because polled nodes can send data packets only to polling nodes to avoid collisions; therefore, at low and moderate loads, there is a non-zero probability that a polled node has nothing to send to the polling node. An interesting result of the analysis is that undirected polling (RIMA-BP) always has lower performance than dual-purpose polling, which should be expected, but is more attractive than simple polling when the node neighborhood is large.

MACA-BI exhibits the best performance in a fully-connected network, which results from the fact that a polled node can transmit data to any node, not just the polling node, and this eliminates the case in which there is no data packet after a successful poll. As we have shown in Section 3, a polled node should transmit only if it has data meant for the polling node in order to avoid sending data to a neighbor that cannot receive the data in the clear. Nevertheless, the good performance of MACA-BI reported by Talucci et al. [11] indicates that a receiver-initiated collision-avoidance protocol should perform very well when nodes have traffic to send to most of their neighbors; this fact is illustrated in the performance results we obtain under a heavy-traffic assumption, which show that RIMA-DP performs much better than FAMA-NCS, and that RIMA-SP performs better than FAMA-NCS.

5.1 Assumptions

We analyze the throughput of receiver initiated protocols using the model first introduced by Kleinrock and Tobagi [7] for CSMA protocols and used subsequently to analyze MACA [6], FAMA [3], MACA-BI [11] and several other collision-avoidance protocols. According to this model, the following assumptions are made:

1. There are N nodes in the fully-connected network.

2. A single unslotted channel is used for all packets, and the channel introduces no errors.

3. All nodes can detect collisions perfectly.

4. The size for a data packet is δ seconds and the size of an RTR and RTS is γ seconds, the size of a CTS in RIMA protocols is γ seconds, and the size of a CTS for FAMA-NCS is $\gamma + 2\tau$ [3].

5. The turn-around time ϵ is considered to be part of the duration of control and data packets.

6. The propagation delay of the channel between any two nodes is τ seconds.

For simplicity, our previous analysis of receiver-initiated MAC protocols [15] assumed that a polled node could send a packet only if it scheduled a data packet to be sent within the duration of the polling packet. This assumption was used to introduce independence among transmissions; however, although the relative comparison among RIMA protocols is valid, it overstates the likelihood of having a polled node with no data to send to a polling node at low to medium traffic load levels, and provides an unfair comparison of RIMA protocols against sender initiated protocols.

To provide a more accurate comparison between sender initiated and receiver initiated protocols while preserving the tractability of the analytical model, we assume that a polled node receiving an RTR always has a data packet to send, but the probability that that packet is addressed to the polling node is $\frac{1}{N}$. Furthermore, we assume that each node sends its RTR according to a Poisson distribution with a mean rate of $\frac{\lambda}{N}$, and that (when applicable) the polling node chooses the recipient of the RTR with equal probability. This model is slightly unfair to RIMA protocols compared to MACA-BI, because the likelihood that a polled node can transmit remains constant even as the traffic load increases. To account for this, we also discuss a heavy-traffic approximation of our results, in which a polled node always has data to send to any polling node.

The corresponding assumptions for sender-initiated protocols are that a node always has packets to send, but schedules the transmission of RTSs according to a Poisson distribution with a mean rate of $\frac{\lambda}{N}$, and chooses to which neighbor to send the RTS with probability $\frac{1}{N}$. These assumptions preserve the validity of prior analytical results for FAMA and MACA [3].

Because the arrival of RTSs or RTRs to the channel is Poisson, the average channel utilization is:

$$S = \frac{\overline{U}}{\overline{B} + \overline{I}} \qquad (1)$$

where $\overline{B}$ is the expected duration of a busy period, defined to be a period of time during which the channel is being utilized; $\overline{I}$ is the expected duration of an idle period, defined as the time interval between two consecutive busy periods; and $\overline{U}$ is the time during a busy period that the channel is used for transmitting user data successfully.

5.2 MACA-BI

The following theorem provides the throughput of MACA-BI in a fully-connected network. In a network with hidden terminals, MACA-BI's performance would degrade substantially according to two factors: (a) the probability of bad busy periods is increased by the probability that either a node sends an RTR within τ seconds of any neighbor sending a data packet, or a node receives a data packet addressed to it while it also receives other data packets; and (b) the length of a bad busy period is proportional to the length of a data packet, rather than the length of an RTR as in RIMA-SP.

Theorem 3 *The throughput for MACA-BI in a fully-connected network is given by*

$$S = \frac{\delta}{\delta + \tau + \frac{1}{\lambda} + (\gamma + 2\tau)e^{\lambda\tau}} \tag{2}$$

Proof: Because a successfully polled node can send data to any neighbor, the probability that a successful transmission occurs equals the probability that an RTR is transmitted successfully, that is,

$$P_S = e^{-\lambda\tau} \tag{3}$$

The duration of every successful busy period is $\gamma + \delta + +2\tau$, and the first and the last packet of the busy period is the successful packet of the period.

Because the network is fully connected, a failed busy period can occur only when there is a collision between RTRs, which occurs with probability $1 - P_S$.

The average duration of any busy period always consists of at least an RTR and the associated propagation delay (i.e., $\gamma + \tau$) plus the average time between the first and the last RTR of the busy period, which we denote by $\overline{Y}$ and is the same as in CSMA [12], i.e., $\overline{Y} = \tau - \frac{1 - e^{-\lambda\tau}}{\lambda}$. If the busy period is successful, a data packet is also sent; therefore, the length of the average busy period in MACA-BI is given by

$$
\begin{aligned}
\overline{B} &= \gamma + 2\tau - \frac{1 - e^{-\lambda\tau}}{\lambda} + e^{-\lambda\tau}(\delta + +\tau) \\
&= \gamma + 2\tau - \frac{1}{\lambda} + e^{-\lambda\tau}\left[\tau + \frac{1}{\lambda} + \delta\right]
\end{aligned} \tag{4}
$$

The length of the average idle period is $\frac{1}{\lambda}$, and the length of the average utilization period is

$$\overline{U} = \delta P_S = \delta e^{-\lambda\tau} \tag{5}$$

The theorem follows by substituting the values of $\overline{U}, \overline{B}$ and $\overline{I}$ in Eq. (1). □

The throughput of MACA-BI has been reported before by Talucci et al. [11]. However, that prior derivation did not take into account that, in computing the length of an average busy period, the first and the last RTR of a busy period is the same, and that there is a non-zero probability that a polled node has no packets to send to any node if RTRs are sent when packets arrive and arrivals are Poisson. Nevertheless, the results in Theorem 2 and [11] are practically the same for the model we have assumed in our analysis, in which a polled node always has something to send, even if it is not the polled node. We should also point out that our own prior analysis of MACA-BI [15] incorrectly assumed that a polled node could only transmit packets to the polling node, which is unfair to MACA-BI in a fully-connected network.

5.3 RIMA-SP

The following theorem provides the throughput of RIMA-SP in a fully-connected network. In a network with hidden terminals, the performance of RIMA-SP would degrade by the increase of the vulnerability period of RTRS from one propagation delay to essentially twice the length of the RTR, and by the need for the polling nodes to send NTRs after detecting interference.

Theorem 4 *The throughput for RIMA-SP in a fully-connected network is given by*

$$S = \frac{\delta\frac{1}{N}}{\frac{\delta}{N} + \xi + \tau + \frac{1}{\lambda} + (\gamma + 2\tau)e^{\lambda\tau}} \tag{6}$$

where $\xi = \tau$ to ensure collision avoidance.

Proof: Because of our independence assumptions, the probability that a successful transmission occurs equals the probability that an RTR is transmitted successfully, times the probability that the polled node has a data packet for the polling node at the head of its queue, that is,

$$P_S = e^{-\lambda\tau}\left(\frac{1}{N}\right) \tag{7}$$

The duration of every successful busy period is $\gamma + \delta + \xi + 2\tau$. Notice that, in this case, the first and the last packet of the busy period is the successful packet of the period.

In RIMA-SP, a failed busy period can occur when there is a collision between RTRs, and when an RTR is sent in the clear but the polled sender does not have a data packet to send to the polling node. The first case occurs with probability:

$$P_{F1} = 1 - e^{-\lambda\tau} \tag{8}$$

The probability of the second case of a failed busy-period scenario occurring is given by

$$P_{F2} = e^{-\lambda\tau}\left(1 - \frac{1}{N}\right) \tag{9}$$

As it was the case for MACA-BI, any busy period always consists of at least an RTR and the associated propagation delay (i.e., $\gamma + \tau$) plus the average time between the first and the last RTR of the busy period, denoted by $\overline{Y}$. When the busy period fails due to the collision of two or more RTRs, there are no additional time components in the busy period. When the busy period is successful, $Y = 0$, of course, and additional time due to the collision-avoidance waiting time and a data packet is incurred, i.e., $\delta + \xi$. Finally, if the busy period fails because the polled node does not have a packet for the polling node, then an additional propagation delay and a collision-avoidance waiting time are incurred. Accordingly, the length of the average busy period is given by

$$
\begin{aligned}
\overline{B} &= \gamma + 2\tau - \frac{1 - e^{-\lambda\tau}}{\lambda} + \frac{1}{N}e^{-\lambda\tau}(\delta + \xi + \tau) \\
&+ e^{-\lambda\tau}(1 - \frac{1}{N})(\xi + \tau) \\
&= \gamma + 2\tau - \frac{1}{\lambda} + e^{-\lambda\tau}\left[\xi + \tau + \frac{1}{\lambda} + \frac{\delta}{N}\right]
\end{aligned} \tag{10}
$$

The length of the average idle period is $\frac{1}{\lambda}$, and the length of the average utilization period is

$$\overline{U} = \delta P_S = \delta e^{-\lambda\tau}\left(\frac{1}{N}\right) \tag{11}$$

The theorem follows by substituting the values of $\overline{U}, \overline{B}$ and $\overline{I}$ in Eq. (1). □

5.4 RIMA-DP

The following theorem provides the throughput for RIMA-DP in a fully-connected network. The performance of RIMA-DP in a network with hidden terminals would degrade by the increase in the vulnerability period of RTRs, which is one propagation delay in a fully-connected network and is twice an RTR in a network with hidden terminals.

Theorem 5 *The throughput of RIMA-DP for a fully connected network is given by*

$$S = \frac{\delta(1 + \frac{1}{N})}{\gamma + \delta + 2\tau + \frac{1}{\lambda} + \frac{1}{N}(\delta + \xi - \gamma) + (\gamma + 2\tau)e^{\lambda\tau}} \tag{12}$$

where $\xi > \gamma + 7\tau$.

Proof: Because the network is fully connected, whenever an RTR is transmitted successfully a packet always follows, either from the node sending the poll or the polled node. Therefore, the probability of success, P_S, is equal to the probability with which an RTR is transmitted successfully. Because all nodes are connected, an RTR from node w is successful if there are no other RTRs transmitted within τ seconds from the start of the RTR. After this vulnerability period of τ seconds, all the nodes detect the carrier signal and act appropriately. Accordingly,

$$P_S = e^{-\lambda\tau} \tag{13}$$

The probability, P_{S1}, with which the polled node has data to send to the polling node is equal to the probability that an RTR is sent in the clear, times the probability that the polled node has a packet to send to the polling node, that is:

$$P_{S1} = e^{-\lambda\tau}\left(\frac{1}{N}\right) \tag{14}$$

The second case of a successful busy period happens when the polled sender does not have a packet to send and therefore it sends a CTS back to the sender of the RTR enabling the node to send a data packet. The probability, P_{S2}, with which this scenario occurs is equal to the probability that an RTR is sent in the clear, times the probability that the polled node has no data packet for the polling node, that is,

$$P_{S2} = e^{-\lambda\tau}\left(1 - \frac{1}{N}\right) \tag{15}$$

As it was the case with RIMA-SP, the length of an average busy period always includes an RTR and a propagation delay, plus the average time between the first and the last RTR of the busy period. When the busy period fails, there are no additional components in it. With probability P_{S1}, a successful busy period case contains two data packets, one from the polled node followed by one from the polling node, plus the associated propagation delays and the collision-avoidance waiting period of ξ seconds. With probability P_{S2}, a successful busy period contains a single data packet from the polling node, plus a CTS from the polled node and the associated propagation delays. It follows that the duration of the average busy period is given by

$$
\begin{aligned}
\overline{B} &= \gamma + 2\tau - \frac{1 - e^{-\lambda\tau}}{\lambda} \\
&+ e^{-\lambda\tau}\left[\frac{1}{N}(2\delta + \xi + 2\tau) + (1 - \frac{1}{N})(\gamma + \delta + 2\tau)\right] \\
&= \gamma + 2\tau - \frac{1 - e^{-\lambda\tau}}{\lambda} \\
&+ e^{-\lambda\tau}\left[\frac{1}{N}(\delta + \xi - \gamma) + \gamma + \delta + 2\tau\right] \tag{16}
\end{aligned}
$$

Because inter-arrival times for RTRs are exponentially distributed, it follows that $\overline{I} = \frac{1}{\lambda}$. The average utilization time at node w is the proportion of time in which useful data are sent, consequently,

$$
\begin{aligned}
\overline{U} &= P_{S1}(2\delta) + P_{S2}(\delta) \\
&= e^{-\lambda\tau}\frac{2\delta}{N} + e^{-\lambda\tau}(1 - \frac{1}{N})\delta = e^{-\lambda\tau}\delta(1 + \frac{1}{N}) \tag{17}
\end{aligned}
$$

Eq. (12) follows from substituting $\overline{B}$, $\overline{I}$ and $\overline{U}$ into Eq. (1). $\square$

5.5 RIMA-BP

Theorem 6 *The throughput of RIMA-BP is given by*

$$S = \frac{\delta(1 - \frac{1}{N})^{N-1}}{\gamma + \xi + 2\tau + \frac{1}{\lambda} + (1 - \frac{1}{N})^{N-1}(\delta - \gamma - \xi) + e^{\lambda\tau}(\gamma + 2\tau)} \tag{18}$$

where $\xi = 4\tau$.

Proof: Given our independence assumptions, the probability of success, P_S, equals the probability with which an RTR is transmitted successfully, times the probability with which an RTS is transmitted successfully. Because all nodes are connected, an RTR from node w is successful if there are no other RTRs transmitted within τ seconds, where τ is the time needed for all the nodes connected to detect the carrier signal. After this vulnerability period of τ seconds, all the nodes detect the carrier signal and act appropriately. Because the arrivals of RTRs to the channel follow the Poisson distribution with rate λ, we can write:

$$P_{SRTR} = e^{-\lambda\tau} \tag{19}$$

The probability that only one of the nodes that receive a successful RTR transmits an RTS is equal to the probability that only one neighbor has a packet ready for the polling node. Because at each neighbor this is the case with probability $\frac{1}{N}$, and because each node has $N - 1$ neighbors, this can be expressed as follows:

$$P_{SRTS} = (N - 1)\left(\frac{1}{N}\right)\left(1 - \frac{1}{N}\right)^{N-2} \tag{20}$$

Therefore, the probability with which a packet is transmitted successfully is

$$
\begin{aligned}
P_S &= P_{SRTR}P_{SRTS} \\
&= e^{-\lambda\tau}(N - 1)\left(\frac{1}{N}\right)\left(1 - \frac{1}{N}\right)^{N-2} \tag{21}
\end{aligned}
$$

There are three ways in which a busy period can be unsuccessful, i.e., contain no data packet. First, the RTRs sent in the busy period may collide with one another, which occurs with probability $1 - e^{-\lambda\tau}$ because all nodes can hear one another. A busy period can also fail if a single RTR is sent in the clear but none of the polled nodes has a packet to send to the polling node; the probability with which this scenario takes place is equal to:

$$P_{F2} = e^{-\lambda\tau}\left(1 - \frac{1}{N}\right)^{N-1} \tag{22}$$

The last case of a failed busy period is when an RTR is successful, but more than one RTSs are sent in response. In this case, the polling node sends an NTR immediately after detecting the collision. The probability with which this type of busy period occurs is:

$$P_{F3} = e^{-\lambda\tau}\left[1 - 2\left(1 - \frac{1}{N}\right)^{N-1}\right] \tag{23}$$

The average length of a busy period always includes the length on an RTR, a propagation delay, and the average time between the first and the last RTR in the busy period, which is the same as in the previous proofs of this section. When RTRs collide, the busy period has no additional components.

With probability P_s, the busy period also includes an RTS from a polled node, a collision-avoidance waiting time at the polling node, and the data packet from the polled node, plus the associated propagation delays. With probability P_{F2}, the busy period

also contains a waiting time of 2τ after which the polling node detects no RTSs. With probability P_{F3}, the busy period also contains the length of the RTSs that collide and its propagation delay, a collision-avoidance waiting time at the polled nodes, and a propagation delay with which the polling node starts sensing the collision. Accordingly, the duration of an average busy period is

$$
\begin{aligned}
\overline{B} \;=\;& \gamma + 2\tau - \frac{1 - e^{-\lambda\tau}}{\lambda} + e^{-\lambda\tau}\left(1 - \frac{1}{N}\right)^{N-1}(2\tau) \\
& + \; e^{-\lambda\tau}\left(1 - \frac{1}{N}\right)^{N-1}(\gamma + \delta + \xi + 2\tau) \qquad (24) \\
& + \; e^{-\lambda\tau}\left(1 - 2\left(1 - \frac{1}{N}\right)^{N-1}\right)(\gamma + \xi + 2\tau)
\end{aligned}
$$

According to the RIMA-BP specification the channel will be idle after every data transmission for a period of $\tau + \epsilon$ seconds. In addition, the channel is idle for a time period equal to the inter-arrival rate of RTRs, so $\overline{I} = \frac{1}{\lambda}$.

The average utilization time at node w is the proportion of time in which useful data are sent. Consequently,

$$
\overline{U} = \delta P_S = \delta e^{-\lambda\tau}\,(N - 1)\left(\frac{1}{N}\right)\left(1 - \frac{1}{N}\right)^{N-2} \qquad (25)
$$

Substituting the equations for $\overline{U}$, $\overline{I}$ and $\overline{B}$ into Eq. (1) we obtain Eq. (18). $\square$

5.6 Numerical Results

To compare the various RIMA protocols with MACA, FAMA-NCS, and MACA-BI, we introduce the variables in Table 2, and Table 1 shows the normalized throughput for the MAC protocols based on those variables. In our comparison, we assume a fully-connected network topology with a propagation delay of $1\mu s$; we used 500 byte data packets; a length of 20 bytes for RTRs, CTSs and NTRs for the various RIMA protocols; CTSs of length $\gamma + \tau$ for FAMA-NCS; a channel data rate of 1 Mb/s; and zero preamble and processing overhead for convenience. Figs. 6, 7 and 8 plot the throughput of MACA, FAMA-NCS, MACA-BI, RIMA-SP, RIMA-DP, and RIMA-BP against the average offered load when the network consists of 5, 10, and 50 nodes, respectively.

$a = \frac{\tau}{\delta}$ (normalized propagation delay)	
$b = \frac{\gamma}{\delta}$ (normalized control packets)	
$G = \lambda \times \delta$ (Offered Load, normalized to data packets)	

Table 2: Normalized variables

The performance attained by RIMA-DP is much better than the performance of the other MAC protocols that provide correct collision avoidance (FAMA-NCS, RIMA-SP, and RIMA-BP). This should be expected, because RIMA-DP permits one or two packets to be sent with each successful handshake, while the other protocols allow just one packet per handshake.

As Figs. 6 to 8 illustrate, the throughput of RIMA-SP degrades as the size of a node neighborhood increases. Even though our model is only a rough approximation of the impact of the number of neighbors a node has, this illustrates the fact that simple polling is inherently limited compared to dual-use polling, because at light and moderate loads there is a non-zero probability that the polled node has no data to send to the polling node.

It is also interesting to observe that the throughput of RIMA-BP is independent of the number of nodes and is always lower than
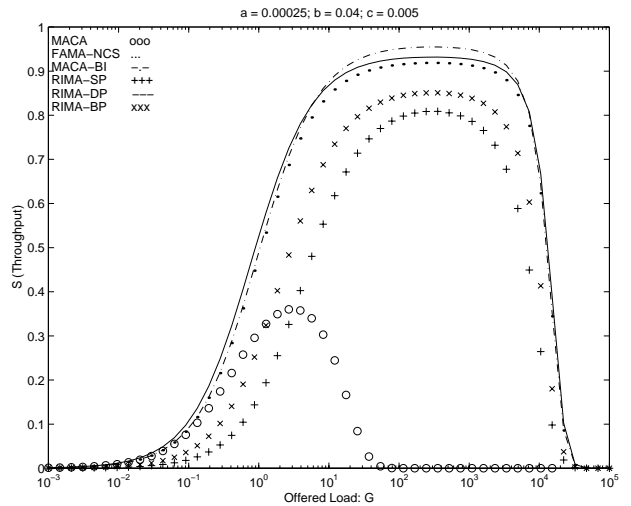

Figure 6: Throughput vs. offered load for 1Mbit/sec channel and 500 Byte data packets; network of 5 nodes
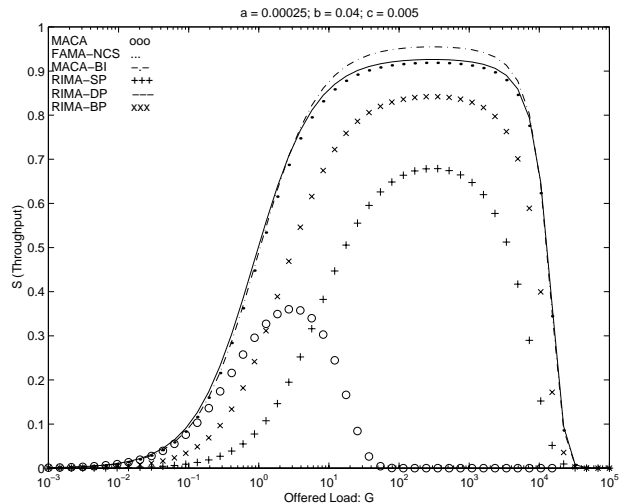

Figure 7: Throughput vs. offered load for 1Mbit/sec channel and 500 Byte data packets; network of 10 nodes

RIMA-DP. There are two reasons for this behavior: a node receiving a broadcast poll can only transmit packets to the polling node, and multiple responses (RTSs) to the poll are likely to be sent, incurring wasted busy periods.

Figs. 6 to 8 also illustrate that carrier sensing is needed to provide high throughput in addition to correct collision avoidance. MACA's poor performance is due to the long durations of busy periods in which collisions occur, which are bounded by a maximum round trip delay and a control packet length with carrier sensing. In fairness to MACA and variants of collision avoidance protocols that do not use carrier sensing, it should be emphasized once more that, with the COTS radios available today, carrier sensing is possible only with FHSS radios in ISM bands, with which entire packets are sent in a single frequency hop. In contrast, collision avoidance without carrier sensing can be applied to FHSS and DSSS radios. However, given the performance advantage of collision avoidance using carrier sensing, FHSS radios appear more attractive than DSSS radios for ad-hoc networks.

In Figs. 6 to 8, MACA-BI achieves the maximum throughput

Protocol	Throughput
MACA [6]	$$\frac{1}{e^{(2b+a)G}\left(b+a+\frac{1}{G}+F'\right)+e^{bG}\left[b+\frac{a}{2}+P'(a-F')\right]+1+\frac{3a}{2}+F'+P'(a-F')}$$ where $F'=\frac{e^{bG}-1-bG}{bG(1-e^{-bG})}$ and $P'=\frac{e^{-bG}-e^{-G(a+b)}}{1-e^{-G(a+b)}}$
FAMA-NCS [3]	$$\frac{1}{b+4a+1+\frac{1}{G}+e^{aG}(b+4*a)}$$
MACA-BI [10]	$$\frac{1}{1+\frac{1}{G}+a+(b+2a)e^{aG}}$$
RIMA-SP	$$\frac{\frac{1}{N}}{\frac{1}{N}+\frac{1}{G}+a+(b+2a)e^{aG}}$$
RIMA-DP	$$\frac{1+\frac{1}{N}}{1+b+2a+\frac{1}{G}+\frac{1}{N}(1+7a)+(b+2a)e^{aG}}$$
RIMA-BP	$$\frac{(1-\frac{1}{N})^{N-1}}{b+6a+(1-\frac{1}{N})^{N-1}(\frac{1}{G}-2b-6a)+(b+2a+\frac{1}{G})e^{aG}}$$

Table 1: Throughput of sender-initiated and receiver-initiated MAC protocols

among all the protocols. The reason for this is that a polled node can transmit a data packet to any node, not just the polling node; however, as we have shown, this should not be done in networks with hidden terminals in which the protocol is meant to operate.

To provide a fairer comparison between MACA-BI and RIMA protocols without having to consider a more complex model involving hidden terminals, we can use a heavy-traffic approximation consisting of assuming that a polled node always has data to send to any polling node. This approximation is actually not far from reality in large networks in which a node always has packets in its transmission queue meant for different destinations and has to distribute them among its various neighbors. With this approximation, the probability that a successful RTR generates two data packets in RIMA-DP is 1, and the probability that an RTR is not answered with data in RIMA-SP is 0; Fig. 9 shows the corresponding results. As could be expected, under the heavy-traffic assumption, RIMA-DP achieves the best throughput under any average load, and RIMA-SP exhibits essentially the same throughput as MACA-BI.

It is evident from Figs. 6 to 9 that making collision avoidance a joint effort by sender and receiver, instead of placing all functionality at the sender or the receiver is the best way to avoid collisions while maintaining a high throughput.

6 Conclusions

We have presented the first treatment of collision avoidance based at the receiver instead of the sender that demonstrates the required features of such handshakes in order to eliminate the possibility of

data packets colliding with any other packets at the intended receiver in networks with hidden terminals. Our simple comparative analysis of throughput of receiver-initiated multiple access protocols shows that a receiver-initiated collision avoidance strategy can be made more efficient than any of the sender-initiated strategies used and proposed to date. We have proposed one strategy based on dual-purpose polling (RIMA-DP) that is always better than sender-initiated collision avoidance schemes.

Although we have analyzed RIMA protocols in fully-connected networks only, the importance of our analysis is in showing which type of collision avoidance handshake should be investigated further. Because RIMA protocols provide correct collision avoidance in any topology, the relative performance differences among these protocols apply also to networks with hidden terminals. A more lengthy analysis for hidden terminals similar to that presented by Tobagi and Kleinrock for CSMA [12] and Fullmer and Garcia-Luna-Aceves for FAMA [3] should be used to verify that this is the case. It is clear from our results that strategies in which dual-purpose polls are used and in which polls are directed to specific neighbors are the ones that should be implemented. Our approach assumes the ability of radios to sense the channel; developing correct collision-avoidance strategies that do not rely on carrier sensing constitutes a fruitful area of research.

References

[1] V. Bharghavan, A. Demers, S. Shenker and L. Zhang, "MACAW: A Media Access Protocol for Wireless LAN's"
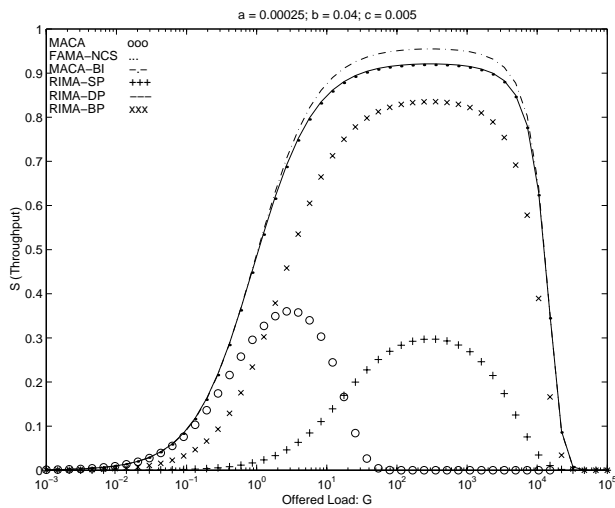
Figure 8: Throughput vs. offered load for 1Mbit/sec channel and 500 Byte data packets; network of 50 nodes
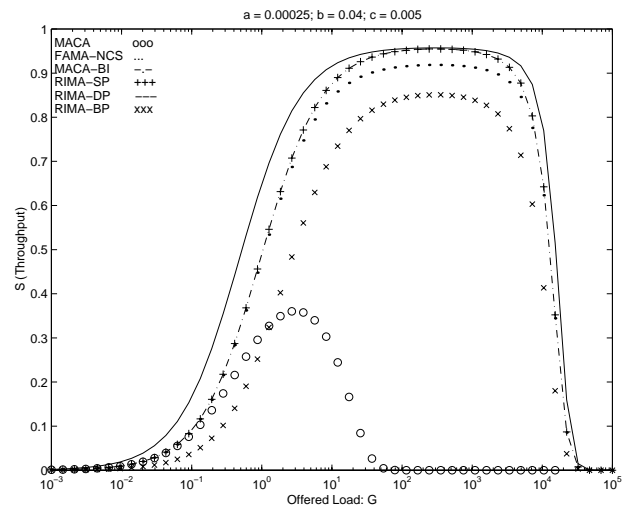

Figure 9: Heavy-traffic approximation: Throughput vs. offered load for 1Mbit/sec channel and 500 Byte data packets; network of 50 nodes

Proc. ACM SIGCOMM 94, London, UK, Aug. 31 - Sep. 2, 1994.

[2] C. L. Fullmer and J.J. Garcia-Luna-Aceves, "Floor Acquisition Multiple Access for Packet-Radio Networks," *Proc. ACM SIGCOMM 95*, Cambridge, MA, August 30-September 1, 1995.

[3] C. L. Fullmer and J.J. Garcia-Luna-Aceves, "Solutions to Hidden Terminal Problems in Wireless Networks," *Proc. ACM SIGCOMM 97*, Cannes, France, September 14-18, 1997.

[4] R. Garces and J.J. Garcia-Luna-Aceves, "Floor Acquisition Multiple Access with Collision Resolution," *Proc. ACM/IEEE MobiCom 96*, Rye, New York, November 11–12, 1996.

[5] *P802.11–Unapproved Draft: Wireless LAN Medium Access Control (MAC) and Physical Specifications*, IEEE, January 1996.

[6] P. Karn, "MACA - a new channel access method for packet radio," in *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, pp. 134–40, ARRL, 1990.

[7] L. Kleinrock and F. A. Tobagi, "Packet switching in radio channels: Part I - carrier sense multiple-access modes and their throughput-delay characteristics," *IEEE Trans. Commun.*, vol. COM-23, no. 12, pp. 1400–1416, 1975.

[8] R. Rom and M. Sidi, "Multiple Access Protocols Performance and Analysis," *Springer-Verlag*, 1990.

[9] H. Takagi and L. Kleinrock, "Output processes in contention packet broadcasting systems," *IEEE Transactions on Communications*, vol. COM 33, no. 11, pp. 1191-9, 1985.

[10] F. Talucci and M. Gerla, "MACA-BI(MACA by invitation). A Wireless MAC Protocol for High Speed Ad hoc Networking," *Proc. IEEE ICUPC '97*

[11] F. Talucci, M. Gerla and L. Fratta, "MACA-BI(MACA by invitation)-A Receiver Oriented Access Protocol for Wireless Multihop Networks," *Proc. IEEE PIMRC '97*

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

[13] F. A. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part III - polling and (dynamic) split-channel reservation multiple access," *IEEE Trans. Commun.*, vol. COM-24, no. 8, pp. 832–845, 1976.

[14] F. A. Tobagi and L. Kleinrock, "The Effect of Acknowledgment Traffic on the Capacity of Packet-Switched Radio Channels," *IEEE Trans. Commun.*, vol. COM-26, no. 6, pp. 815—826, 1978.

[15] A. E. Tzamaloukas and J.J. Garcia-Luna-Aceves, "Poll-before-Data Multiple Access," *Proc. IEEE ICC '99*, Vancouver, Canada, June 6–10, 1999