10-2004

Design and Analysis of a Cooperative Medium Access Scheme for Wireless Mesh Networks

Arup ACHARYA
IBM T. J. Watson Research Center

Archan MISRA
Singapore Management University, archanm@smu.edu.sg

Sorav BANSAL
Stanford University

DOI: https://doi.org/10.1109/BROADNETS.2004.29

Citation

ACHARYA, Arup; MISRA, Archan; and BANSAL, Sorav. Design and Analysis of a Cooperative Medium Access Scheme for Wireless Mesh Networks. (2004). *First International Conference on Broadband Networks BROADNETS: Proceedings: San Jose, CA, October 25-29, 2004*. 621-631. Research Collection School Of Information Systems.
Available at: https://ink.library.smu.edu.sg/sis_research/693

Design and Analysis of a Cooperative Medium Access Scheme for Wireless Mesh Networks

Arup Acharya, Archan Misra
IBM T J Watson Research Center
19 Skyline Drive, Hawthorne, NY 10532, USA
{arup, archan}@us.ibm.com

Sorav Bansal
Stanford University, Gates 508
Stanford, CA 94035, USA
sbansal@stanford.edu

Abstract

This paper presents the detailed design and performance analysis of MACA-P, a RTS/CTS based MAC protocol, that enables simultaneous transmissions in wireless mesh networks. The IEEE 802.11 DCF MAC prohibits any parallel transmission in the neighborhood of either a sender or a receiver (of an ongoing transmission). MACA-P is a set of enhancements to the 802.11 MAC that allows parallel transmissions in situations when two neighboring nodes are either both receivers or transmitters, but a receiver and a transmitter are not neighbors. The performance of MACA-P in terms of system throughput is obtained through a simulation of the protocol using ns and is compared with the 802.11 RTS/CTS MAC. Experiments with the base MACA-P protocol reveal the need for certain enhancements, especially to avoid the drawbacks associated with attempts at parallel transmissions in scenarios where such parallelism is not feasible. Studies with the enhanced MACA-P protocol also demonstrate how significant performance gains in wireless mesh network performance may be realized if the radio transceiver behavior is modified in tandem with the MAC protocol.

1. Introduction

IEEE 802.11-based [1][2] wireless LANs (WLANs) offer an increasingly popular access networking model, especially as transmission rates of 54 Mbps and above enable a range of high-bandwidth multimedia wireless applications. Currently deployments of 802.11-based networks are purely single hop, with the various mobile or client devices connecting to the access point (AP) via a direct wireless link. There is, however, great excitement surrounding the notion of *multi-hop, wireless 802.11-based mesh networks*, where the wired backbone is reachable only via *multiple wireless hops*. Potential examples of this include *in-building* wireless networks in malls, hotels and apartment blocks, and *community* networks where rooftop antennas are used to create an ad-hoc wireless network in specific residential communities.

Such multi-hop networks however currently exhibit very poor performance in terms of overall throughput. It is important to realize that a significant factor contributing to this poor performance is the MAC's *inability to efficiently support multi-hop packet forwarding*, as distinct from the usually suspected causes such as mobility-induced link breakages and error-prone wireless channels. Indeed, as several studies have shown, even in scenarios where all the nodes are static, and the wireless channel is reasonably error-free, the achieved throughput is extremely low. For example, [3] [4] showed how TCP sessions suffer from a sharp drop in throughput when transmitted over multiple 802.11-based hops. The 802.11 MAC is primarily responsible for this degradation, since it does not allow multiple simultaneous transmissions, even if these are ideally feasible. The 802.11 CSMA-CA (Carrier Sense Multiple Access with Collision Avoidance) mechanism for distributed access to the shared channel is extremely restrictive and prohibits any concurrent transmission or reception activity in the vicinity of either an active sender or receiver. This overly restrictive design principle may be appropriate for a *single-hop* wireless LAN where nodes form a *clique*, but is particularly bad in exploiting the *spatial diversity* available in multi-hop wireless settings.

In this paper, we present the detailed design and performance evaluation of **MACA-P**, an enhancement to the 802.11 MAC for obtaining higher concurrency in spatially diverse wireless networks. MACA-P's tries to schedule multiple transmissions in parallel as long as it does not violate the fundamental constraint needed to avoid collisions at any receiver:

> *If any node is currently receiving information from another neighboring node, no node (other than the transmitting node) within the one-hop neighborhood of*

the receiver can engage in a simultaneous transmission.

Accordingly, MACA-P's aims to coordinate the reception and transmission times of neighboring nodes (in a distributed manner) to avoid collisions at a receiver node. The basic design philosophy of MACA-P has been presented earlier in [5]. This paper addresses several questions related to the optimal choice of various MACA-P parameters and investigates the potential benefits of designing better radio receivers to exploit the parallelizing capabilities of MACA-P. In addition, we present an adaptive learning scheme to combat realistic wireless scenarios where nodes often interfere with one another's transmissions but cannot communicate with one another.

Like other CSMA-CA based MAC protocols (such as [6], [7], [8], [1]), MACA-P also contains a contention-based reservation or signaling phase. Unlike these protocols, *the data transmission interval in MACA-P does not always occur immediately after the reservation phase, but can be delayed by a variable, yet bounded, interval*. Incorporating this interval (called a ***control phase gap***) enhances the likelihood of parallel transmissions by allowing multiple sender-receiver pairs to synchronize their data transfer intervals.

1.1 Background Work

The early research on the 802.11 MACA algorithm, such as [7] and [8], alludes to the possibility of parallel transmissions, but does not present any specific solutions. More recently, [9] describes PCMA, a power control scheme to increase the number of simultaneous transmissions within an ad-hoc wireless network. PCMA uses power control to effectively partition the total network into a larger number of non-interfering regions, each of which can engage in transmission activity independently. In contrast, our current version of MACA-P does not use power control but instead extends the RTS/CTS based MAC to increase the number of situations where parallelism is feasible. Very recently, [10] combined the design philosophies of PCMA and MACA-P into a concurrent transmission protocol that combines delayed packet transfer with power control. However, [10] does not discuss the improvement of MACA-P through adaptive learning of feasible concurrent transmission schedules, or the potential benefit from a better design of wireless transceivers. Another interesting recent work is [11]: though fundamentally different in design and goals, MACA-P uses a similar philosophy of sharing information with neighboring nodes for a better channel access.

The rest of this paper is organized as follows. In section 2, we review the operation and limitations of the 802.11 DCF MAC and the basic design components of MACA-P. Section 3 presents comparative simulation studies between our basic implementation of MACA-P and the standard 802.11 MAC. Such studies motivate the contents of Section 4, which discusses several additional protocol refinements to improve MACA-P's performance in practical wireless environments. Finally, section 5 concludes the paper with a list of unresolved issues.

2. 802.11 Limitation and MACA-P Fundamentals

For ease of discussion, we make a distinction in the use of four common terms: the terms *sender and recipient* refer to the source and destination nodes of a particular packet transfer, and the terms *transmitter and receiver* refer to the corresponding nodes associated with a specific transmission activity. An ACK-based link layer packet transfer involves at least two distinct activities: (1) packet transmission by the sender to a recipient and (2) a corresponding ACK transmitted by the recipient to the sender. A link-layer packet transmission between a *(sender, recipient)* node pair thus involves a *role-reversal*, with the both the sender and the recipient alternately act as transmitter and receiver respectively.

The 802.11 MAC's *Distributed Coordination Function (DCF)* uses a 4-way distributed handshake mechanism to resolve contention between peers. We now discuss why the 802.11 DCF MAC does not permit two nodes to transmit simultaneously that are either neighbors or have a common neighboring node. Consider the following observation, which must be supported by a*ny wireless MAC* to avoid collisions at a receiver:

Observation SRS : *If any node is currently a transmitter, there can be only one receiver node in the transmitter's 1-hop neighborhood. Conversely, if any node is a receiver, only one node in its 1-hop neighborhood is allowed to be a transmitter.*

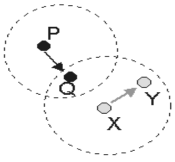


Fig. 1

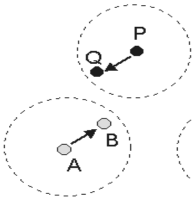


Fig. 2a

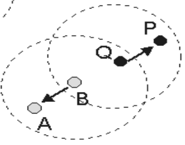


Fig. 2b

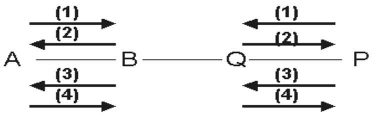


Fig. 2c

Consider Fig.1 where the transmission from X (to Y) would interfere would P's transmission to Q, since Q is within range of both X and P. Therefore, the two transmissions cannot occur simultaneously. Now consider Fig. 2 where Q and B are one-hop neighbors, and A's transmission range does not include Q (and vice versa), and P's transmission range does not include B (and vice versa). The transmission patterns shown in cases (3) and (4) shown in Fig 2c are clearly inherently infeasible, since they would cause collisions at a receiver node (at node Q in case 3 and at node B in case 4). However, there is no fundamental constraint in parallel transmissions when the neighbors are either both transmitters, or both receivers. For example, in case 1 in Fig 2c, the transmissions A-to-B and P-to-Q (shown in Fig. 2a) can proceed in parallel, since A's transmission range does not include Q and P's transmission range does not include B. However, the 802.11 MAC does not support such parallel transmissions: when B sends a CTS in response to A's RTS, Q is aware that B has reserved the channel. If now P sends a RTS to Q, Q cannot respond with a CTS to P since it is aware of an existing channel reservation[1]. A similar situation exists for the scenario in Fig 2b (and case 2 in Fig 2c) for the case of two neighboring senders, where the RTS transmitted by the first sender effectively prohibits the 2nd sender from sending out an RTS.

The failure of the 802.11 MAC to support these two types of concurrent activity occurs for two distinct reasons:

a. *In any packet transfer, a node reverts between a transmitter (tx) and receiver (rx) roles multiple times* without a precise, explicit knowledge of when these role reversals take place. For example, in Fig 2a, A is in a tx role for the RTS and DATA transmission phases, while B is in a rx role during the same two phases. In the CTS and ACK phases, B is in a tx role while A is in a rx role. Assuming the P-to-Q 4-way handshake is initiated while the DATA transmission is in progress from A to B, P's RTS would be received correctly by Q. However, to reply with a CTS, Q would take on a tx role and that would violate observation *SRS* stated earlier, i.e. Q's transmission of a CTS would interfere with A's data transmission at B.

b. In the 802.11 MAC, the 4-way handshake mechanism is effectively contiguous—once a node pair initiates a packet transfer, neighboring nodes cannot assume the role of a transmitter until the original 4-way handshake is complete. Clearly, the RTS/CTS exchange between a sender-recipient pair (e.g. for a P-to-Q transmission) cannot proceed simultaneously with a DATA transmission between a neighboring pair (e.g A-to-B).

These observations motivate our fundamental design decision to introduce a *control gap*, between the RTS/CTS exchange and the subsequent DATA/ACK exchange, in MACA-P. This *variable* gap provides two important functions:

- Subsequent to a RTS/CTS exchange by a tx/rx pair (e.g. A-to-B), it allows other neighboring pairs to exchange RTS/CTS messages (e.g. P-to-Q) within the control phase gap of the first pair.
- It allows subsequent pairs (e.g. P-to-Q) to align their DATA and ACK transmission phases with that of the first pair- the DATA transmissions are scheduled at the end of the control gap.

Note that the control gap is put in place by the first pair (A-to-B). A subsequent RTS/CTS exchange by a neighboring pair (P-to-Q) uses the remaining portion of the control gap to align their data transmission with the first pair. MACA-P's principal goal is the enhancement of the 4-way handshake to allow parallel communication in cases 1 and 2 of Fig. 2c.

2.1 Overview of MACA-P Behavior

In MACA-P, each (sender, receiver) uses the initial RTS/CTS exchange to establish a future reference time instant at which the DATA and ACK phases will commence. An explicit delineation of these time instants allows neighboring nodes to then proceed with their RTS/CTS exchanges (in the ensuing control gap) and synchronize their own DATA and ACK phases with the already established schedule. We now revisit the basic building blocks of MACA-P, an initial version of which had been described in very condensed form in [5].

Control Phase: In addition to permitting a variable gap between the RTS/CTS and DATA/ACK phases, the MACA-P protocol adds extra information in the RTS and CTS messages to explicitly delineate the intervals for both the DATA and ACK transmissions, thereby allowing neighboring nodes to know exactly when the two nodes associated with the DATA/ACK switch between tx and rx roles. To avoid the requirement of synchronized clocks, the following two time instants are both specified *relative to the time of receiving* the associated control packet:

T_{DATA}: indicates the start time of DATA transmission.

T_{ACK}: indicates the start time of the ACK transmission.

---

[1] The data structure at each node that records current or impending channel activity is called a NAV (Network Allocation Vector), as per the 802.11 MAC specification.

In figure 3 below, node Q overhears the RTS sent from A to B. If Q has a packet to transmit, it will initiate a RTS whose T_{DATA} is aligned with the start time of B's data transmission. Both RTS and CTS messages carry the two intervals so that nodes that are neighbors of either the sender or the recipient learn of the scheduled data and ACK transmissions.

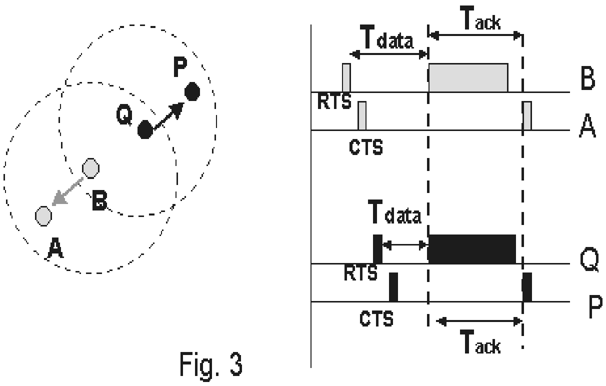


Fig. 3

State of neighboring nodes: As in 802.11, each MACA-P node is required to maintain the state of its neighboring nodes by overhearing the RTS/CTS exchanges from its neighbors. Consider Fig 3, where B initiates a RTS/CTS exchange with A. Since Q hears the RTS from B, it will update its NAV to indicate that B has scheduled a transmission to A. For each neighbor from which a RTS or a CTS has been overheard, a node maintains an entry in the NAV consisting of the neighbor's MAC address, sender or recipient, T_{DATA} and T_{ACK} intervals. This information is used as follows: if a node wishes to send a data packet, it first must check that no entry in its NAV is marked as a recipient (as otherwise the SRS observation made earlier would be violated). Similarly, a node receiving an RTS cannot respond with a CTS if any entry in its NAV is marked as a sender. In addition to this basic test, the NAV allows a node to figure out if there is a transmission already scheduled in its neighborhood and use this information to schedule an overlapping data transmission of its own. For example, in Fig.3, Q updates its NAV on overhearing B's RTS to A, and then uses this information to schedule an overlapping transmission of its own, as explained next.

Inflexible Bit in RTS : The RTS message is further enhanced to carry a bit called the **inflexible** bit, which indicates to the RTS receiver whether the transmission schedule proposed in the RTS message can be changed : if the bit is set, then this schedule cannot be changed. Consider figure 3 again. When B sends its RTS to A, this bit is *unset* since there are no transmissions in B's neighborhood. However, after that, assume Q wishes to send a packet to P. Q's NAV has already been updated with B's scheduled transmission as a result of overhearing B's RTS. Consequently, Q sends a RTS to P with the inflexible bit *set* and the data transmission aligned with that of B. There are situations where the proposed schedule from a sender may be infeasible for a recipient based on its own neighborhood information—a modified schedule may however be feasible for the recipient. If the inflexible bit is set in the RTS, then the recipient has to either accept the proposed schedule (by sending a CTS back with the same T_{DATA} and T_{ACK} as the RTS[2]) or reject it completely (by not sending a CTS back); it cannot send a modified schedule back on the CTS in this case.

Modification of T_{DATA} and T_{ACK} by CTS: When a node receives a RTS where the inflexible bit is *not* set, it may change the proposed schedule by modifying the T_{DATA} and T_{ACK} of the RTS, and sending back the modified values on the CTS. Consider figure 4, where B has overheard the CTS from Q and is aware of a scheduled reception in its neighborhood. On receiving a RTS from A with the inflexible bit unset, B responds with a modified T_{DATA} and T_{ACK} (shown as t1 and t2) so that its reception of data from A overlaps with Q's reception.

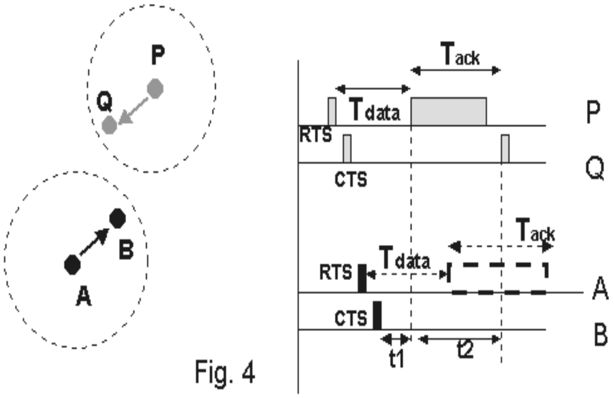


Fig. 4

RTS' message: Nodes update their respective NAVs on overhearing a RTS. However, unlike 80.211, a MACA-P recipient may modify the schedule proposed in the original. To avoid inconsistencies at a neighbor of the RTS sender (who would otherwise be unaware of the changed schedule), the RTS sender always sends a gratuitous RTS message (**RTS'**) with updated T_{DATA} and T_{ACK} (received from the CTS) immediately after the CTS.

---

[2] The T_{DATA} and T_{ACK} on the CTS will be slightly different than the RTS to account for the fact that the T_{DATA} and T_{ACK} on the RTS reflect the intervals after the RTS, while those on the CTS reflect the intervals after the CTS, but in both cases they refer to the same start times of the data and ACK transmissions.

A second use of RTS' is to cancel a transmission schedule, when the sender does not receive a CTS from the intended recipient. After waiting a CTS timeout period, the RTS sender transmits a RTS' with zero T_{DATA} and T_{ACK}**;** neighbors hearing this cancellation message flush the corresponding entry in their NAVs. The RTS' is specially important to cancel a proposed schedule and thus prevent the problem of *cascading lockouts*. For example, in a chain of nodes,

S1 — R1 — S2 — R2 — S3 — R3

assume that S3 has successfully exchanged CTS/RTS with R3, and S2 sends a RTS to R2 during the control gap or DATA transmission of the S3-to-R3 transfer. However, R2 cannot respond with a CTS, since there exists a scheduled transmission (not reception) in its neighborhood. Following the RTS from S2, assume that S1 sends a RTS to R1. Since R1 has heard S2's RTS, it cannot respond to S1. In effect, the S3-to-R3 transmission has locked out both S2-to-R2 and S1-to-R1[3]. Using the RTS' however allows both S3-to-R3 and S1-to-R1 transmissions to proceed in parallel: when S2 does not receive a CTS from R2, it sends a RTS' thereby freeing the channel for use by any neighbor.

MACA-P preserves 802.11's mechanism of exponential backoffs for contention resolution. As in 802.11, a MACA-P node wishing to transmit on the channel must ensure that the channel is idle for a DIFS period to avoid collisions. To accommodate the additional duration of the RTS' message, the DIFS period is slightly longer in MACA-P.

Master Transmission Schedules: MACA-P essentially works by creating an extended neighborhood of (sender, receiver) pairs that synchronize to a common transmission schedule for the DATA/ACK phases. This behavior can be formalized by the notion of a *master* transmission schedule, which is chosen by a master sender node (one that is unaware of any scheduled activity in its neighborhood) to a master recipient. Note that the use of the word "master" does not imply any form of centralized control—each node may become a master if it happens to choose a transmission schedule to which others synchronize. Due to the spatial diversity in wireless meshes, a node can have multiple master transmission sets (each set is a collection of nodes following a common schedule for the DATA/ACK phases) in its neighborhood. To regulate parallel transmissions, MACA-P imposes the following rule:

A sender/recipient pair can schedule a data transmission only if there is at most one master transmission in the sender's neighborhood or at most one master reception in the recipient's neighborhood, but not both.

The rationale is as follows. In Figure 5a, Y is neighbor of B and Q, but B is not a neighbor Q. The two transmissions A-to-B and P-to-Q have been scheduled, i.e. Y has two masters, B and Q. X then sends a RTS to Y. If Y has to fit in this transmission, it must align X's data transmission with P's data transmission (Q's reception) and stretch out its (Y's) ACK to X to align with B's ACK to A. While such extended forms of concurrency may be possible in certain scenarios, our current implementation uses a simple policy of allowing a new (sender, recipient) pair to synchronize with at most one existing schedule. (A similar approach, outlined in Fig. 5b, can be used to synchronize with one master sender in the sender's neighborhood.) If both the sender and recipient nodes have distinct pre-existing master schedules in their individual neighborhoods, then MACA-P does not allow the initiation of a new transmission.

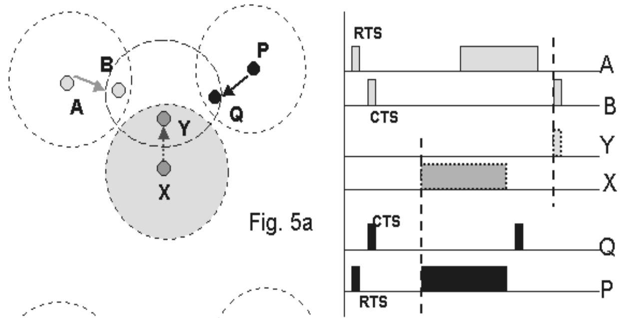


Fig. 5a

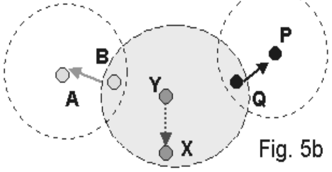


Fig. 5b

Parallel MACA-P transmissions are feasible only when the "slave" transmissions take less time than the master. To avoid situations where concurrency proves impossible due to an excessively small master packet size, master senders in provide a control gap only for "large" packets. For smaller packets, a master node uses the standard 802.11 with contiguous RTS/CTS/ DATA/ACK .phases, avoiding the higher signaling overhead for "small" data packets.

3. Performance Studies on Base MACA-P

We now describe studies on the relative performance of the basic MACA-P protocol vs. 802.11, using the ns-2 (version 2.1b8) simulator, which motivate the subsequent

---

[3] Note that, it does not lock out R2 from sending data to S2, aligning the R2-to-S2 transmission with S3-to-R3.

introduction of additional features in MACA-P. Before proceeding further, it is however, important to clarify an important physical layer feature.

3.1 Physical-Layer Capture Effects and MACA-P

Wireless receivers are designed to exploit the phenomena of *packet capture*—i.e., the ability to recover a stronger signal in the presence of an interfering signal, as long as the interfering signal is significantly weaker than the primary signal. This requirement is usually expressed in terms of a capture threshold; typical receivers are able to perform capture if the primary signal is ~6-10 dB stronger than the weaker signal. Of course, transmission power levels also define both a transmission range (beyond which the received signal strength is too low), and an interference range (beyond which the signal power drops even below the carrier sense threshold). Capture allows parallel transmissions even if a receiver lies within the interference range of a secondary transmitter. For example, in Fig 3, A can receive B's transmission even if Q is within the interference range, as long as the signal from B is exceeds that from A by the capture threshold. In terms of distance, for a given distance between a sender and recipient, we can define an equivalent *capture radius,* such that capture works only if the interfering node lies outside this capture radius.

In the current ns-2 implementation of 802.11, such capture works only if the first bit of the stronger signal is received before the first bit of the weaker signal; if the stronger signal arrives later, a collision is declared and both packets are dropped. We believe that there this is no fundamental reason for this restriction; many cellular transceivers can "lock onto" (via detection of appropriate training sequences) a stronger signal even if it arrives later than the weaker one. For our purposes, this difference is important since the small propagation delays between neighbors on an 802.11-type network (1 µsec for a distance of 300 meters) may cause slight offsets between apparently "synchronized" master and slave transmissions. For now, we thus make the reasonable assumption that *a second stronger-signal packet, arriving later, can be captured as long as it arrives "shortly after"(4 µs in our simulation environment) the arrival of the first packet* (at the physical layer, this corresponds to the arrival of the training sequence for the 2nd packet *before the completion of the training sequence of the 1st packet.*)[4] This modification rectifies a small but significant flaw in the current ns-2 implementation and *does not require any changes in real-life receiver cards*. In a later section (IV.F), we shall show that MACA-P can post even more impressive performance gains if receiver designs are modified to perform such capture *anytime* during the reception of the 1st packet (even if the 2nd packet arrives after the receiver has "locked on" to the 1st packet).

3.2 MACA-P Implementation

We implemented MACA-P by extending the 802.11 DCF MAC available in *ns*. The RTS/CTS/RTS' exchange was implemented with an extra 2-byte field T_{data} (see Fig 3) in the header of each of these packets. Since the original 802.11 RTS/CTS packet (including the physical headers) is around 40 bytes, the *increase in the size of the control packets is around 5%*. Moreover the use of RTS' for modifying/canceling the original RTS imposes an additional 25% signaling overhead. The *nav*, which stores the neighborhood activity information, is now maintained as a table, with each entry maintaining the state of the neighbor (SENDER, RECEIVER, IDLE), T_{data} (start time of data transfer), T_{ack} (start time of ACK Transfer). Table 1 lists the values for the various MACA-P and simulation parameters; apart from the ones defined explicitly for MACA-P, the remaining parameters have the usual meaning (as in 802.11).

Table 1 : Simulation Parameter Values

Channel Capacity	1 Mbps
Propagation Model	TwoRayGround
Reception Range	250m
Carrier Sense Range	550m
Capture Threshold	6 dB
CONTROL_GAP	512 Bytes
SIFS (Short Inter-frame Space)	10 µs
DIFS (DCF Inter-frame Space)	50 µs
CWmin and CWmax (used for Random Backoff)	31 µs,1023 µs
Size of RTS/GratRTS/RTS-Cancel Packets	177 bits
Size of CTS Packets	177 bits

3.3 Base Performance Studies

We first used the topology shown in Fig. 6 to verify the basic operation of MACA-P and the role of capture. Since the carrier sense range (550 m in our studies) is typically more than twice the reception range, the performance of MACA-P depends heavily on the capture effect. We used

---

[4] This behavior can be replicated even in receivers that are unable to perform capture during the training of the 1st packet, simply by modifying every transmitter to send an initial "garbage" in the preamble of every packet transmission.

two traffic patterns: one with UDP traffic from nodes 1 and 4 to nodes 2 and 3 respectively, and another with UDP traffic from nodes 2 and 3 to nodes 1 and 4 respectively. By varying the angle 2-3-4,we observed that the capture effect breaks down, and MACA-P suffers from an extremely sharp drop in cumulative throughput, for angles less than $\pi/2$ for a capture threshold of 6dB (as expected under the Two-Ray propagation model which causes d^4 attenuation).

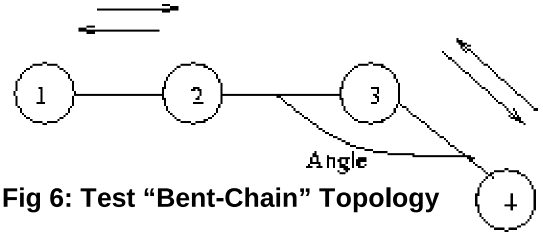


Fig 6: Test "Bent-Chain" Topology

To study the performance gain of MACA-P, we performed simulations on a concentric ring scenario illustrated in Figure 7. This layout consists of an equal number of nodes, placed in inner and outer concentric circles, with all the inner nodes form a *clique.* While 802.11 does not allow more than one transmission at any given time for the concentric ring, the number of simultaneous transmissions for MACA-P can be as high as *n/2* (for *n* nodes). We considered two traffic patterns:

Case 1: **Outer Senders**, with traffic going from each outer node to its corresponding inner node.

Case 2: **Inner Senders**, with traffic going from each inner node to its corresponding outer node.

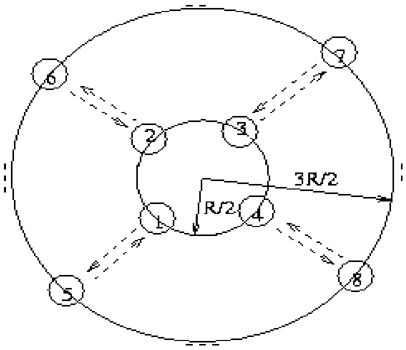
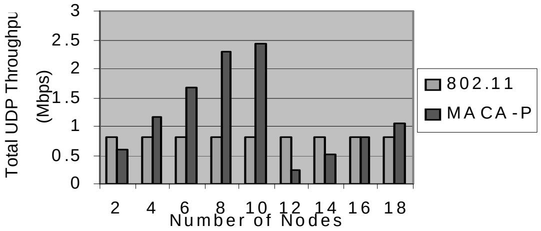
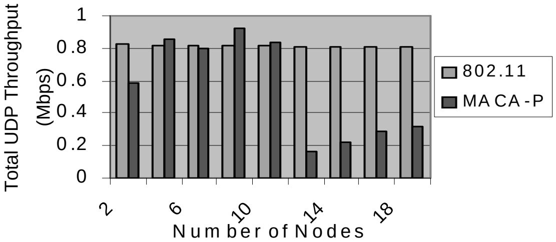


Figure 7: Concentric Ring Topology

We measured the cumulative throughput by the number of successful packets successfully received. Figure 8 shows the relative performance of 802.11 and MACA-P for packet sizes of 1536 bytes, and a MACA-P control gap of 640 Bytes. While throughput in 802.11 never exceeds the channel capacity, MACA-P results in almost 200% improvement in some scenarios.



8a: Inner Senders



8b: Outer Senders

Figure 8: MACA-P/802.11 on Concentric Ring

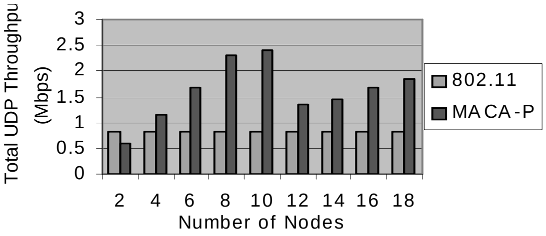
The figure illustrates two important points. Firstly, MACA-P suffers a sharp drop in throughput when the number of inner nodes exceeds 5. This is due to the scheduling of infeasible concurrent transmissions; when the outer nodes get too crowded, an outer node comes within the capture radius (but outside the transmission range) of other inner nodes. Concurrent transmissions scheduled by MACA-P end up causing collisions and a dramatic drop in throughput. Secondly, we see that MACA-P achieves much lower throughput in the outer-sender case. In the inner-sender case, the inner senders simply suppress their RTS packets if parallel transmissions are infeasible. In the outer-senders case, concurrency control is achieved by the suppression of CTS packets by the inner receivers; the outer nodes (the senders) then pay the penalty of (possibly multiple) exponential backoffs. On analyzing the trace, we observed that the poor performance in the outer sender case was primarily due to the fact that an inner node would *often* fail to hear the master CTS packet due to a collision with an interfering RTS packet from one of the outer senders. This leads to stale NAV in the neighboring inner nodes causing wastage of the control gap.

4. Enhancements to the Base MACA-P Specifications

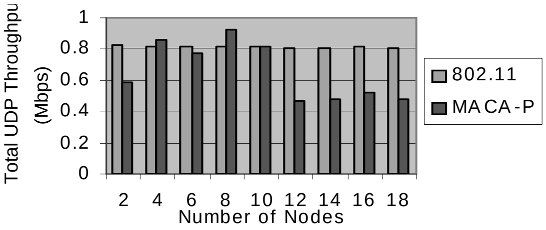
In this section, we look at a variety of enhancements to base MACA-P behavior, and transceiver properties, that significantly improve the inadequacies observed with MACA-P performance in our initial studies.

4.1 Adaptive Learning in MACA-P

Base MACA-P can exhibit serious performance degradation in dense environments, where interfering nodes lie *inside the capture but outside the transmission radius*. To avoid the collisions caused by MACA-P's attempts at concurrent transmissions in such situations, we devise an enhanced version of MACA-P with adaptive learning. In essence, each node learns from the success or failure of concurrent transmissions to eventually avoid infeasible concurrent data transmissions.



9a: MACA-P (with Adaptation): Inner Senders



9b: MACA-P (with Adaptation) : Outer Senders

Figure 9: MACA-P Adaptive Learning on Concentric Ring

Each node maintains a function F : $2^N \rightarrow (0,1]$ where N is the set of all nodes. This function maps a set of participating nodes to a number between (0,1] which indicates the likelihood of a successful parallel data transmission between the participating nodes. Note that we do not provide a distinction between a receiver and a sender in this function, since any data transfer always involves bi-directional transmissions. When a node is attempting to set up a concurrent transmission, it obtains the value F(S), where S is the set of nodes to which it is attempting to synchronize its current transmission (i.e., those nodes that have already completed their RTS/CTS exchange). *The node will then attempt to perform a concurrent transmission only with probability F(P).* MACA-P then performs adaptive adaptive learning by updating F(S) using exponential weighing: for every attempted concurrent transmission, *F(S) is updated according to the rule:*

$$F(S) = F(S)*(1-\alpha) + O*\alpha,$$

where O=1 if the packet transfer was successful and O=0 otherwise. α is a forgetting factor that should ideally depend on the dynamicity of the network topology (e.g., if nodes are mobile or not); for static environments, α can be set close to 0. For our experiments, we set α=0.6, allowing for rapid learning updates. Figure 9 shows the performance of MACA-P with adaptive learning. While there is clearly an improvement in both the inner and outer-sender cases (MACA-P throughput does not degrade as sharply with increasing N), the performance of the outer-sender case is still poor. While adaptation prevents the inner nodes from taking bad decisions, it does not stop the outer senders from sending RTS packets which lead to collisions.

4.2 Preferential Triggering in MACA-P Grid

To further study MACA-P, we performed simulation studies on a 2XN grid-like layout (see Fig 10) that represents a very common wireless mesh network topology. Nodes are placed 250m from each other (the transmission range is also 250m); each node in the top row streams UDP data to the corresponding node in the lower row.

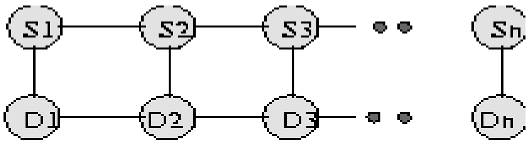


Figure 10: A 2xN Grid

Figure 11 shows the throughput results for MACA-P (with adaptive learning) vs. 802.11 as the number of columns is varied; the MACA-P control gap is set to 512 bytes. We see that MACA-P shows considerable improvement over 802.11. However, an investigation of simulation traces showed that the degree of concurrency in MACA-P was not always maximized. Quite often, sources separate by a few hops independently set up independent master schedules, effectively preventing intermediate nodes (which had more than one master transmission set) from exploiting the control gap for concurrent transmissions. (For example, if S1 and S3 in Fig 11 set up independent transmission schedules, causing S1 to block. Ideally, MACA-P should schedule transmissions from S1, S2 and S3 all in parallel (in a distributed manner).

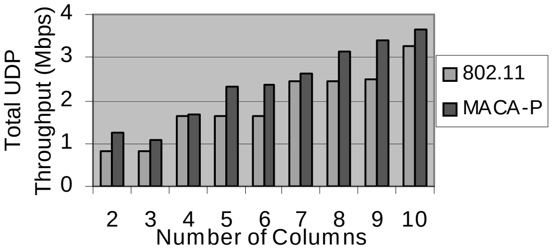


Figure 11: MACA-P on a 2xN Grid

To reduce this problem, we introduce a *preferential triggering* mechanism, whereby we increase the *probability* that a potential slave transmitter node, overhearing a master RTS/CTS exchange, initiates its own RTS signaling *before* RTS/CTS exchanges by nodes outside the transmission range of the master transmitter. The idea is to cascade the slave transmissions over multiple hops, thereby allowing more sender-recipient pairs to synchronize with a single master transmission schedule. To avoid collisions among multiple slave nodes, we preserve the basic philosophy of random back-off. To provide such slave nodes a higher priority, a prospective sender (that can proceed in parallel) *halves its residual back-off time* upon hearing a master RTS packet. Although halving the backoff timer may increase the collision probability, this strategy appears to work well since the number of feasible slave transmitters is not very high in practical topologies. (For example, any node that is within the transmission range of both the master sender and its recipient cannot feasibly transmit anyway). Figure 12 plots the throughput observed in 802.11, MACA-P and MACA-P with trigger for a 2xN grid. We can see that the triggering mechanism leads to an observable, although not substantial, improvement in throughput.

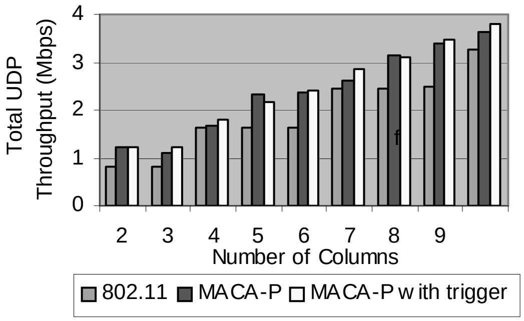


Figure 12: MACA-P with trigger (2xN grid)

4.3. Choice of Control Gap Length

The performance of MACA-P is clearly heavily dependent on the control gap. If the control gap is too small, there is lesser opportunity for the slave nodes to schedule in parallel. However if the control gap is too large, a lot of time is wasted idling. An optimum value of the control gap will depend on the number of neighbors. Fig 13 illustrates the effect of varying the control gap on MACA-P performance for the concentric ring topology. As expected, the optimum control gap size increases with increasing number of nodes in the concentric ring (since it gives more nodes a chance to squeeze in their RTS/CTS packets), at least in the inner-sender case wherever MACA-P is effective. A minimum of ~256B is required to observe any advantage through MACA-P (for a second RTS/CTS packet exchange to successfully take place).

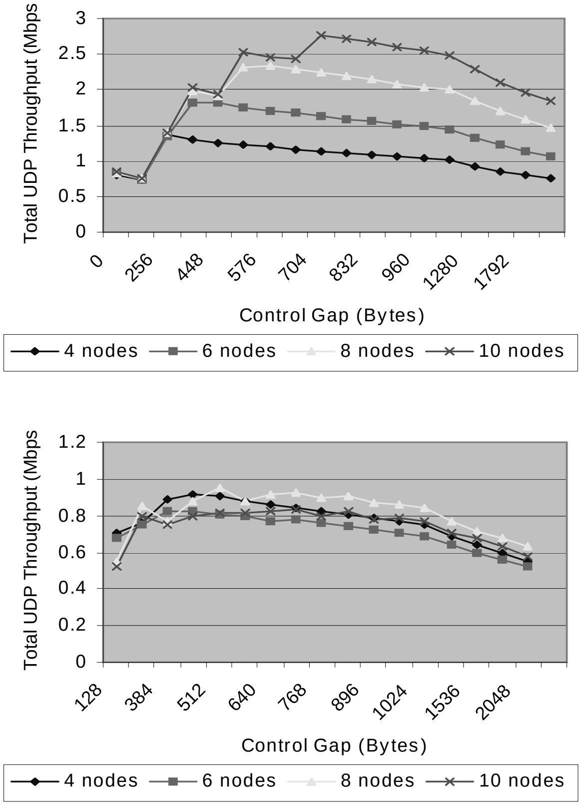


Figure 13: MACA-P for varying control gap in the Concentric Ring (Top: Inner Senders, Bottom: Outer Senders)

4.4. Modification of Capture Behavior

We have so far studied MACA-P under the assumption that the radio is able to capture a packet with a stronger signal only if it arrives `almost simultaneously' (within 4μs) or before the interfering packet—this assumption is satisfied by current radio receivers. There is, however, no fundamental reason why radio receivers cannot be designed to capture a stronger second packet *anytime* during the reception of the first packet. MACA-P performance would significantly benefit from such a physical layer capability, as this would significantly extend the range of layouts over which concurrent transmissions are feasible. Fig 15 plots the dramatic increase in throughput for the outer-sender concentric ring scenario if the physical layer allowed capture at any point (labeled MACA-P(2nd capture) in the figure). Analysis of the simulation trace showed that the performance gain resulted primarily from the fact that this modification allowed an inner receiver to correctly receive the CTS from another receiver and appropriately update their NAV, even if they had already begun receiving a colliding RTS from any other outer sender. In other words, second capture solved the *CTS-Loss* problem discussed in Section 3.3. The result illustrates how future broadband wireless mesh networks may significantly benefit from joint design of MAC and radio transceiver behaviors.

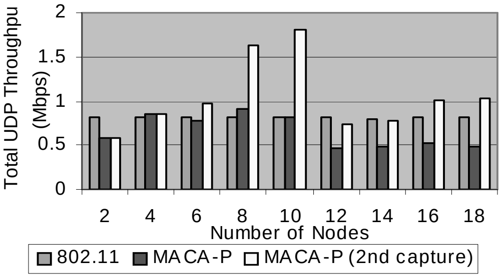


Figure 15: MACA-P with 2nd capture for the outer-sender concentric ring

5. Conclusions and Future Work

This paper first showed that the 802.11 MAC does not permit many feasible concurrent transmissions, due to the fact that each node of a sender-recipient pair switches roles between a transmitter and a receiver multiple times during the course of the RTS/CTS/DATA/ACK exchange. For parallel transmissions to take place, two neighboring sender/recipient pairs must switch roles for the DATA and ACK transmissions in lockstep, as well as provide a control gap between the CTS and DATA phases for other sender/recipient pairs to complete the necessary signaling. These observations drive the design of MACA-P, a parallelizing MAC specifically designed to improve throughput in wireless mesh networks.

Simulation experiments verified the performance gains over 802.11 with the base MACA-P design. While the gains were sometimes spectacular (more than 200% in some instances), we observed lingering performance drawbacks due to MACA-P's attempts at parallel transmissions in some cases. We then introduced the notion of an adaptive learning algorithm that helps avoid infeasible transmissions, especially in dense topologies. We also introduced an additional preferential triggering mechanism that extends the degree to which synchronized transmission schedules propagate over multiple hops. Our investigation of MACA-P also highlighted the significant impact of radio transceiver capabilities on the the MAC performance. Our performance studies show that, as long as the propagation delays are small, MACA-P can work without any changes in the physical layer capabilities of current WLAN receivers. Moreover, MACA-P performance improves tremendously if receivers can be modified to capture a stronger signal, irrespective of its arrival instant.

Significant opportunities exist for future work. Since MACA-P performance depends significantly on the optimal size of the control gap, we need to investigate adaptive algorithms for adjusting this gap. Performance studies on larger, random network topologies are also needed to quantify MACA-P's performance gains.

6. References

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

[2] B. Crow, I. Widjaja, J. G. Kim and P. T. Sakai. *IEEE 802.11 Wireless Local Area Networks.* IEEE Communications Magazine, Sept '99.

[3] S. Xu and T. Saadawi. *Does the IEEE 802.11 MAC Protocol work well in multihop wireless ad hoc networks?* IEEE Communications Magazine, June 2001.

[4] M. Gerla, K. Tang and R. Bagrodia. *TCP Performance in Wireless Multi-hop networks.* IEEE WMCSA '99, Feb '99.

[5] A. Acharya, A. Misra and S. Bansal, *MACA-P: A MAC for Concurrent Transmissions in Multi-hop Wireless Networks*, IEEE Percom 2003.

[6] P. Karn. *MACA – a new channel acces method for packet radio*. ARRL/CRRL Amateur Radio 9th Computer Networking Conference, 1990.

[7] V. Bhargavan, A. Demers, S. Shenker and L. Zhang. *MACAW : A Media Access Protocol for Wireless LAN's*. Proceedings of ACM SIGCOMM '94.

[8] J. Weinmuller, H. Woesner, J. Ebert, A. Wolisz. *Analyzing the RTS/CTS Mechanism in the DFWMAC Media Access Protocol for Wireless LANs*. IFIP TC6 Workshop on Personal Wireless Communications, April '95.

[9] J. P . Monks, V. Bhargavan and W. Hwu. *A Power Controlled Multiple Access Protocol for Wireless Packet Networks*. Proceedings of IEEE Infocom 2001.

[10] A. Muqattash and M. Krunz, *A Single Channel Solution for Transmission Power Control in Wireless Ad Hoc Networks*, Proceedings of ACM MOBIHOC, May 2004.

[11] R. Rozovsky and P. R. Kumar. *SEEDEX: A MAC protocol for ad hoc networks*. Proceedings of ACM Mobihoc 2001.