RC22325 (W0202-014) February 5, 2002
Computer Science

IBM Research Report

MACA-P : A MAC Protocol to Improve Parallelism in Multi-Hop Wireless Networks

Arup Acharya, Archan Misra
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Sorav Bansal
IBM India Research Labs

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

MACA-P : A MAC Protocol to Improve Parallelism in Multi-Hop Wireless Networks

Arup Acharya
IBM TJ Watson Research Center
arup@us.ibm.com

Archan Misra
IBM TJ Watson Research Center
archan@us.ibm.com

Sorav Bansal
IBM India Research Labs
soravban@in.ibm.com

Abstract

Several studies have shown that multi-hop networks using IEEE 802.11 wireless LANs, based on a RTS/CTS MAC, exhibit significant throughput degradation. A reason for this degradation is that a RTS/CTS based MAC does not allow simultaneous transmissions, even if these are ideally feasible. This paper presents the design of MACA-P, a RTS/CTS based MAC protocol that supports simultaneous transmissions in ad-hoc wireless networks. When no transmitter (tx) node is a neighbor of a receiver (rx) node, MACA-P strives to support parallel transmissions even when some of the rx nodes are neighbors or some of the tx nodes are neighbors. The paper first specifies the conditions under which such parallel transmissions are possible. MACA-P contains a contention-based reservation phase to reserve the channel for a subsequent data transmission interval. A key feature of MACA-P is that the data transmission interval does not occur immediately after the reservation phase but is delayed by a control phase interval. This phase allows multiple sender-receiver pairs to synchronize their data transfer and ensures that neighboring nodes coordinate their reception or transmission times in a distributed manner such that multiple transmission do not cause a collision at a receiver. In addition to the RTS and CTS messages, MACA-P uses a new RTS' control message that is used to either alert neighboring nodes of a re-alignment of the data transmission interval or turn down a proposed re-alignment by a data receiver.

1 Introduction

IEEE 802.11-based [ieee] [wlan] wireless LANs (WLANs) are clearly becoming a popular way for connecting to the backbone infrastructure. Recent innovations to the IEEE 802.11 standard, including 802.11a operating at 54Mbps and enhanced versions operating at speeds up to 108Mbps, will further increase the capacity of the wireless channel. Such enhancements are expected to usher in the deployment of multi-hop, wireless 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.

Several studies have, however, shown that multi-hop networks based on the 802.11 MAC protocol exhibit significant performance degradation. For example, [multi] [tcp] showed how TCP sessions suffer from a sharp drop in throughput when transmitted over multiple 802.11-based hops. A major reason for this poor performance is the unduly restrictive nature of the 802.11 MAC, which 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 was principally designed for the *single-hop* wireless LAN scenario, where all nodes would usually be within the transmission range of one another (thus forming a *clique*). Clearly, multiple simultaneous transmissions would not be possible in this case. Multi-hop wireless networks are however *spatially diverse*, with different nodes able to communicate directly with *different* sets of one-hop neighbors.

In this paper, we present the design and performance evaluation of **MACA-P,** a CSMA-CA based MAC protocol designed to significantly increase the potential for multiple simultaneous transmissions in ad-hoc networks. MACA-P is based on simple modifications to the 802.11 MAC and retains its fundamental property of distributed, contention-based access to the common physical channel. Moreover, MACA-P also retains the RTS-CTS based handshake mechanism employed for unicast transmissions, although it does introduce the possibility of an additional *RTS'* control message that may be needed to alert neighboring nodes of a re-alignment of the data transmission interval. MACA-P was designed by observing that common-channel networks, which use the CSMA-CA principle, have only one fundamental constraint[1] on multiple simultaneous transmissions in non-clique topologies:

> *If any node is currently engaged in receiving information from another neighboring node, no other node (other than the transmitting node) within the one-hop neighborhood of the receiver may also be engaged in a simultaneous transmission*.

Accordingly, a major component of the MACA-P multiple access algorithm is to ensure that neighboring nodes coordinate their reception or transmission times (in an essentially distributed manner) to avoid the collisions caused by multiple simultaneous transmissions at a receiver node.

Like all the other proposed MAC protocols based on the CSMA-CA algorithm (such as [maca], [macaw], [fama], [ieee]), MACA-P also contains a contention-based reservation or signaling phase that is used by the sender and receiver to mutually negotiate the data transmission interval. Unlike most of these protocols, *the data transmission interval does not always occur immediately after the reservation phase but can be delayed by a variable, yet bounded, interval*. We shall see that incorporating this interval (which we call a ***control phase gap***) enhances the likelihood of parallel transmissions by serving two distinct purposes:

- It allows other sender-receiver pairs to exchange their control information to set up additional feasible concurrent transmissions.
- It allows multiple sender-receiver pairs to synchronize their data transfer interval, thereby avoiding the possibility for collisions during these transfers.

While the specification of a control gap does increase the duration of the individual transmission (and thus lowers the ideal maximum throughput between a specific node pair), it significantly increases the probability of multiple parallel transmissions by additional node-pairs. MACA-P achieves such increased concurrence by requiring each node to maintain any information that it may have overheard about current or future activity by neighboring nodes. Note that such information is already maintained in a *NAV* (Network Activity Vector) structure in the current 802.11 protocol [ieee] -- MACA-P enhances this data structure with additional information and intelligent processing of this information.

Given the volume of work in the area of wireless medium access protocols in general, we mention only those which are directly relevant. In the past, [macaw] and [dfwmac] have alluded to the possibility of parallel transmissions but no solutions were presented. More recently, [pcma] describes a power control scheme to increase the number of simultaneous transmissions within ad-hoc wireless network. MACA-P, in contrast, does not use power control mechanisms but instead extends the RTS/CTS based MAC to increase parallelism. Another interesting recent work is [seedex] which attempts to avoid collisions without explicitly contending for the channel, by each node propagating its seed for a random transmission schedule within a two-hop neighborhood. While [seedex] does not attempt to increase parallelism, it represents a cooperative, rather than a pure contention-based approach for channel access. Though fundamentally different in design and goals compared to [seedex], MACA-P shares the broad approach of sharing information with neighboring nodes for a better channel access. [direct] describes another approach to increase parallelism in ad-hoc networks through the use of directional antennas.

---

[1] Rigorously speaking, this constraint applies to bi-directional links, where both neighbors can communicate directly with each other. We do not consider uni-directional links in this paper.

The rest of this paper is organized as follows. In section 2, we first use examples to frame the fundamental constraint problem in a more rigorous manner, and then describe why 802.11 4-way handshake mechanism prohibits simultaneous transmissions in several of these examples. Section 3 explains the design components of MACA-P and sets the ground for a description of the protocol in the subsequent section. The concluding section presents our future work on extending MACA-P.

2 Problem Definition

In this section, we consider the various scenarios where concurrent transfer of data between various node pairs may be possible, and explain why the 802.11 MAC does not allow for such concurrency in most scenarios. Before proceeding further, it will be helpful to make an important observation about the ACK-based mechanism used for reliable transfer of data packets over a specific link. We wish to make a distinction between the *sender and recipient* of a particular packet, and the *transmitter and receiver* associated with a specific transmission activity. An ACK-based link layer packet transfer involves at least two distinct phases, with the actual data being transmitted by the sender and received by the recipient, and the corresponding acknowledgement being transmitted by the recipient and received by the sender. Any link-layer packet transmission between a *(sender, recipient)* node pair thus involves a *role-reversal*, with the both the sender and the recipient alternating between transmission and reception phases.

The 802.11 MAC is designed to provide shared access to the wireless medium in two basic modes:

- *Point-coordination Function (PCF)*, where a master node regulates access to the medium between one or more slave nodes
- *Distributed Coordination Function (DCF),* where all nodes resolve potential contentions using a distributed algorithm.

When used in multi-hop ad-hoc networks, the 802.11 MAC is best used in the DCF mode where each node behaves as a peer to all other nodes within its transmission range. In this paper, any reference to the 802.11 MAC will implicitly refer to the RTS/CTS based DCF mode of operation. This section discusses why any RTS/CTS based MAC such as 802.11 when used in the DCF mode, does not allow two nodes to transmit simultaneously that are either neighbors or have a common neighboring node.

When multiple sender/recipient pairs are disjoint, i.e. no sender node is a neighbor of a recipient node, a MAC that allows two or more simultaneous transmission when either some of the recipient nodes are neighbors or some of the sender nodes are neighbors, would greatly enhance the overall system throughput. The RTS/CTS based 802.11 MAC protocol disallows such a parallel operation. To see why, consider the following observation which must be supported by a wireless MAC:

- **Observation SRS** : If any node is currently acting as a sender, only one other node (the recipient of the transmitted information) in its 1-hop neighborhood can be receiving. No other node in the transmitter's neighborhood can be receiving a simultaneous transmission from another transmitter. Conversely, if any node is acting as a recipient, only one other node (the sender) in its 1-hop neighborhood is allowed to transmit.

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 at the same time.

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). It is clear that the transmission patterns shown in cases (3) and (4) shown in Fig 2c are not inherently feasible. In case (3), B's transmission to A would collide with P's transmission at Q, while in case (4), A's transmission (to B) would collide with Q's transmission (to P) at B (similar to the discussion for Fig1). Next consider the two transmissions A-to-B and P-to-Q, as shown in Fig. 2a and case(1) in Fig 2c. Since A's transmission range does not include Q and P's transmission range does not include B, the two transmissions should be allowed to proceed in parallel. Same with case(2) and Fig2b : since B's transmission range does not include P, and A is outside Q's range, B and Q should be able to transmit to A and P respectively at the same time. There is no fundamental constraint on two one-hop neighbors (Q and B in this case) that prevent both from simultaneously acting as senders or both simultaneously acting as recipients. However, as explained below, the 802.11 MAC disallows simultaneous transmissions in such situations (Fig 2a and Fig 2b).

Under the current 802.11 Distributed Control Function (DCF), the 4-way handshake[2] effectively precludes any other sender/recipient from operating during the entire duration of an ongoing packet transmission. In other words, the 802.11 MAC does not allow two one-hop neighbors to either be simultaneously recipients or senders. The four-way handshake consists of 4 different message transfers (between nodes A and B):

1. The RTS (request-to-send), sent by node A, specifying a time interval T_{RTS} that includes B's response through a CTS (clear-to-send), followed by data transmission by A and time to send an ACK from A to B[3]. This is in effect informs anyone within A's neighborhood, that the medium is "reserved" for the duration T_{RTS}.

2. The CTS, sent by node B, specifying the time interval T_{CTS} during which A is permitted to send this data--in 802.11, the interval specified in the CTS is equal to the transmission times of the data and the ACK. The CTS informs all neighbors of B that the channel is reserved for the duration T_{CTS}.

3. The data itself, sent by node A, during the slot reserved for it by the CTS--this data transfer phase immediately[4] follows the reception of the CTS. Note that the data transmission interval is typically larger that than control message transmission times (CTS/RTS/ACK). The max data frame that can be sent is 2346 bytes, while the RTS, CTS and ACK control frames are 20 , 14 and 14 bytes respectively.

4. The final data ACK, sent by node B, indicating successful reception--this ACK is sent after the end of the transmission of data by A[5].

[2] It is actually the two-way handshake of RTS/CTS that precludes anyone within the range of either the sender or receiver of the data transmission from using the channel during the subsequent data transmission interval.

[3] The interval also includes short inter-frame spacing (SIFS)periods between the RTS and CTS, CTS and data, data and ACK. For the rest of the paper, a SIFS is assumed between transmissions even if not stated explicitly.

[4] Actually, as noted earlier, after a SIFS period.

[5] With a gap of a SIFS period.

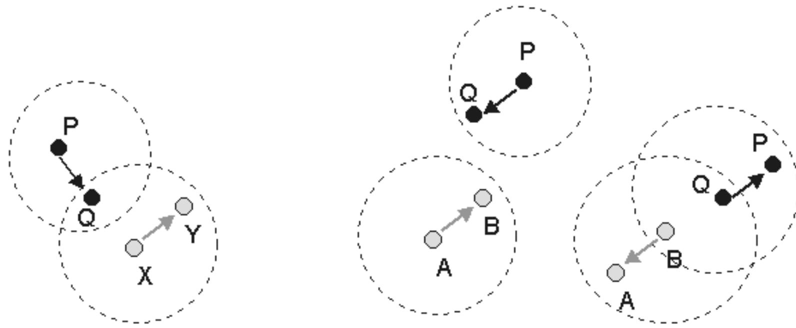


Fig. 1 Fig. 2a Fig. 2b

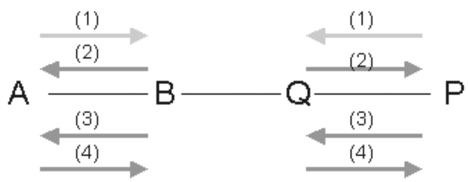


Fig. 2c

With the above description of DCF in mind, we now explain why the transmissions scenarios outlined in Figs 2a and 2b are not allowed by 802.11 MAC. In Fig 2a, B sends a CTS in response to A's RTS. Since Q is within B's range, it is now aware that B has reserved the channel for T_{CTS} interval. During this time interval, assume P sends a RTS to Q to schedule a data transmission from P. However, Q cannot respond with a CTS to P since it is aware that B has reserved the channel for a period that would overlap with P's data transmission[6]. Thus, though it seems possible that P and A should be able to overlap their transmissions, the 802.11 MAC precludes this operation. A similar line of reasoning for Fig 2b shows that B's RTS reserves the channel for a subsequent time interval T_{RTS} so that Q cannot send out a RTS (for scheduling its data transmission to P) within that interval.

One reason why 802.11 is unduly restrictive, as shown in the two cases above, is because a node reverts between a tx (transmitter) and rx (receiver) role multiple times during a packet transfer 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 phase in the A-to-B 4-way handshake is in progress, 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. A similar reasoning also explains why the transmission scheme of Fig2b is disallowed. Assuming Q wishes to initiate a RTS to P during the data transmission phase from B to A is in progress, the RTS from Q would reach P without interference, but the CTS from P would not be received correctly at Q since the CTS and data transmission from B would interfere at Q, i.e it violates observation SRS stated earlier.

Another key observation that we make here is that a RTS/CTS exchange between a sender-recipient pair (e.g. for a P-to-Q transmission) cannot take place simultaneously with a data transmission between another pair (e.g A-to-B). On the other hand, the RTS/CTS exchanges of the two pairs cannot overlap either. This motivates a need for introducing a gap after a RTS/CTS exchange and the subsequent data transmission. This gap, which we term as a *control phase* serves two purposes:

[6] The data structure at each node that records channel reservations that it knows about, is called a NAV (Network Allocation Vector), as per the 802.11 MAC specification.

- Subsequent to a RTS/CTS exchange by a tx/rx pair (e.g. A-to-B), it allows other neighboring pairs (which may be able to schedule a overlapping data transmission), to exchange RTS/CTS messages (e.g. P-to-Q) within the control phase gap.

- It allows subsequent pairs such as P-to-Q to align their data transmission phase with the first pair, by scheduling their data transmission 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) does not redefine the gap. Instead, subsequent pairs use the remaining portion of the control gap to align their data transmission with the first pair.

In summary, this section examined under which conditions it should be possible for two neighboring pairs to overlap data transmissions and why RTS/CTS mechanism used by the 802.11 MAC prevents such simultaneous transmissions. We then introduced the notion of a control gap to serialize the RTS/CTS exchanges followed by simultaneous data transmissions. The next section describes our proposed enhancement to the MAC to enable parallel data transmissions.

3 Building blocks of the MACA-P protocol

Control Phase : The first enhancement we make to the RTS/CTS protocol is to add extra information in the RTS and CTS messages to explicitly delineate the intervals for both the data transfer phase and the ack phase, thereby allowing other nodes to know exactly when the two nodes associated with the transmission under consideration switch between tx and rx roles. This is done by explicitly introducing two time intervals to the RTS and CTS control messages:

- T_{DATA} : specified as a time interval after the reception of this control message, this indicates the start time of the data transmission.

- T_{ACK} : specified as a time interval after the reception of this control message, this indicates the start time of the ACK control message.

This is a departure from the format of the RTS/CTS messages of the 802.11 MAC, which include a single time interval. By introducing the two time intervals, we are explicitly demarcating the control phase, the data transmission phase and the ACK. The time interval T_{DATA} informs all neighboring nodes of the duration of the control phase and allows for the possibility that an overlapping transmission may be scheduled aligned with the end of this interval[7]. The second interval T_{ACK} is necessary to align the ACKs of overlapping transmissions. In figure 3 below, B overhears the RTS sent from Q to P, and becomes aware that a neighboring node is initiating a transmission. If it has a packet to transmit during this time, it will initiate a RTS whose T_{DATA} is aligned with the start time of Q'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 scheduled data and ACK transmissions after T_{DATA} and T_{ACK} intervals.

[7] The conditions under which parallel transmissions can be scheduled will be described shortly.

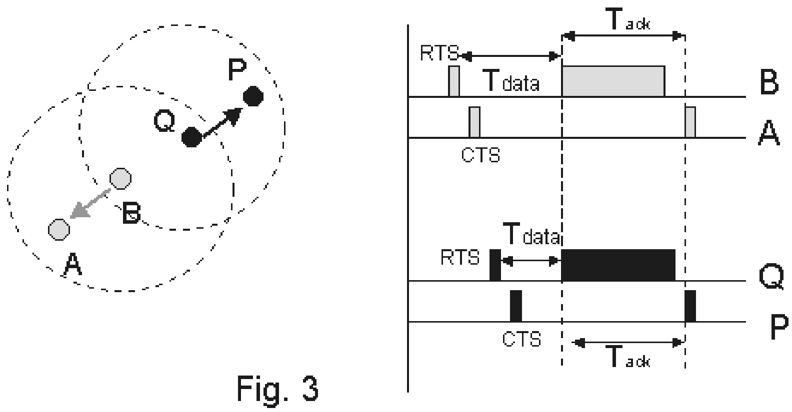


Fig. 3

State of neighboring nodes : Each node is required to maintain the state of its neighboring nodes by overhearing the RTS/CTS exchanges from its neighbors. Consider Fig 4a. B initiates a RTS/CTS exchange with A. Since Y 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 must check that no entry in its NAV is marked as a recipient. Otherwise, it would violate the SRS observation made earlier. Similarly, if a node receives a RTS, it cannot respond with a CTS if any entry in its NAV is marked as a transmitter. 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, B updates its NAV on overhearing Q's RTS to P, 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 which we call the **inflexible** bit. The purpose of this bit is to indicate to the receiver of the RTS message 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 Q sends its RTS to P, this bit is unset since there are no transmissions in Q's neighborhood. However, after that, assume B wishes to send a packet to A. B's NAV has already been updated with P's scheduled transmission as a result of overhearing Q's RTS. Consequently, B sends a RTS to A aligning its data transmission with that of Q and sets the inflexible bit in the RTS. Note that there are situations where the proposed schedule from a sender may be infeasible for a recipient based on its own neighborhood information but a modified schedule may be feasible for the recipient. However, 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[8]) or reject it completely (by not sending a CTS back); it cannot send a modified schedule back on the CTS.

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 of the sender so that it can align the proposed data transmission by the sender with an existing scheduled reception in its neighborhood. This is done by modifying the T_{DATA} and T_{ACK} received on the RTS message, and sending back the modified values on the CTS message. Consider figure 4, where B has overheard the CTS from Q an is aware of a scheduled reception in its neighborhood. Thus, when it receives a RTS from A with the inflexible bit unset, it responds with a modified T_{DATA} and T_{ACK} (shown as t1 and t2) so that B's reception of data from A overlaps with Q's reception.

---

[8] 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.

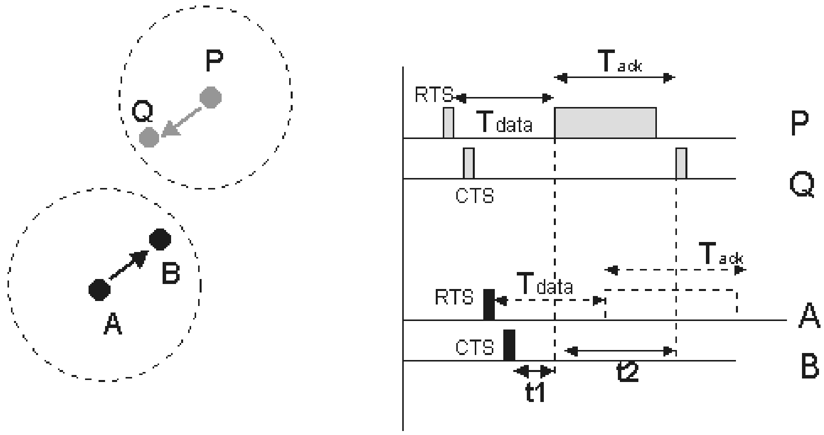


Fig. 4

RTS' control message : When a sender transmits a RTS message, neighbors of the sender have updated their respective NAVs based on overhearing the RTS. However, if the schedule proposed in the RTS is modified by the receiver of the RTS message (as discussed earlier), neighbors of the sender are not updated of the modified schedule. To avoid such a situation, the sender always sends a gratuitous RTS message containing the updated T_{DATA} and T_{ACK} that it received from the CTS. The purpose of this gratuitous RTS (which we term **RTS'**) is *not* to send an acknowledgement of the CTS to the recipient, but instead inform all other neighbors of the sender of the modified schedule. The RTS' is sent following a SIFS (Short Inter-frame Space) period of time after receiving the modified CTS.

A second use of the RTS' message is to cancel a prior schedule made through the matching RTS, when the sender did not receive a CTS from the intended receiver (of the RTS). The receiver may be unable to respond (to the RTS) for a variety of reasons : there may be an ongoing reception in the receiver's neighborhood or the proposed (inflexible) schedule may violate the SRS observation at the receiver. The CTS could also be lost due to channel errors. However, neighbors of the sender have updated their respective NAVs based on overhearing the initial RTS. In this case too, after waiting a SIFS period plus a CTS timeout period, the sender transmits a RTS' message with zero T_{DATA} and T_{ACK}. On reception of this message, neighbors remove the entry corresponding to the sender from their respective NAVs. This lets neighbors use the channel during the time originally reserved by the RTS.

An important benefit of using RTS' to cancel an earlier proposed schedule is to avoid cascading lockouts of the medium in certain scenarios. Consider a chain of nodes,

S1 — R1 — S2 — R2 — S3 — R3

where sender S3 has successfully exchanged CTS/RTS with a recipient R3. Now, if during the control or data transmission phases of the S3-to-R3 schedule, assume S2 sends a RTS to R2 to schedule S2-to-R2 transmission. However, R2 cannot respond with a CTS, since there exists a scheduled transmission (not reception) event in its neighborhood. Following the RTS from S2, assume that S1 sends a RTS to R1 to schedule a S1-to-R1 transmission. 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[9]. However, in principle, both S3-to-R3 and S1-to-R1 transmissions could proceed in parallel. This is achieved by the use of RTS'. When S2 does not receive a CTS from R2, it sends a RTS' thereby freeing the channel for use by any neighbor. This is why we introduce an additional message RTS' in MACA-P.

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

Enhanced Basic Access Mechanism : MACA-P is designed to preserve 802.11's mechanism of exponential backoffs for contention resolution. Whenever a node using the 802.11 MAC is first requested to transfer a packet, it senses the channel till the channel beomes inactive. Following that, it waits a period of time equal to DIFS (Distributed Inter-Frame Space). A random backoff timer in the range (0, Congestion Window) is then selected in terms of number of slots. If the chanel is idle on expiration of the timer, packet transmission is initiated. The timer is frozen whenever the channel becomes busy. Else, the congestion window is doubled and a random timer is chosen from the new window.

DIFS is the period of time in 802.11 DCF MAC that a node must wait to transmit a message after sensing an idle channel. In MACA-P, given the possibility of a RTS' following a RTS, a node that overhears the RTS and then senses an idle channel must wait long enough to allow for the possibility that the sender may transmit a RTS' . Thus, we require that a node that senses an idle channel after overhearing a RTS must use a sufficiently large DIFS period to allow for the possibility of hearing a RTS', i.e. should this node chose to transmit a RTS of its own, it must do so only after a time that allows the overheard node to send a RTS'.

Master Transmission Schedules : A master transmission schedule is one that allows neighbors of either the sender or recipient to schedule their own *data transmission* overlapping with the master. A set of aligned data transmissions is referred to as a transmission set. The data sender of a master transmission will be referred to as a master sender, and similarly the data recipient will be referred to as a master recipient. In general, we will simply use the term "master" when it is clear from the context whether it refers to a sender or a recipient.

We now state a key requirement whether it is possible for a sender/recipient pair to schedule a transmission depending on the number of masters in their neighborhood:

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.

To see why we mandate such a requirement, consider Figure5. 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. In general, if a node has more than 1 master, it has to align the proposed data transmission with that of the master with earliest data transmission and align the ACK with that of the master with the latest ACK. First, this adds complexity to our solution. Second, all master (recipient) nodes other than the master with the latest ACK, are blocked from scheduling any further receptions till the master transmission with the latest ACK, completes. In the figure, this means Q cannot schedule any further reception (from P, say) before Y sends its ACK to X (aligned with the ACK from B to A). Otherwise, a subsequent CTS from Q could interfere with Y's reception of data. For MACA-P, we take a conservative approach and disallow a node with more than one master from participating in a parallel transmission/reception. Figure 5b shows the analogue for a node with more than one master sender.

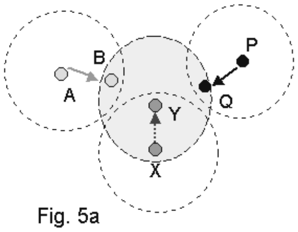


Fig. 5a

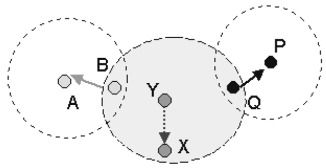
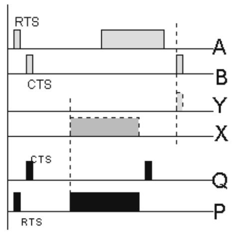


Fig. 5b



Next, consider Figure 6, where both the sender and the recipient of a proposed transmission schedule, has a master each. In Fig 6, assume that both B-to-A and P-to-Q transmissions have been scheduled. B is a master of Y while Q is a master of X. Y sends a RTS with the inflexible bit set. Since X has a master already, it must align any reception of its own with master Q. However, Y's proposed transmission schedule is aligned with its (Y's) master B. In this case, X will not be able to respond to Y's RTS. What this example shows is that if both the sender and recipient have a master transmission and reception scheduled in their respective neighborhoods, then it is not possible for the sender/recipient pair to have a data transfer without collision.

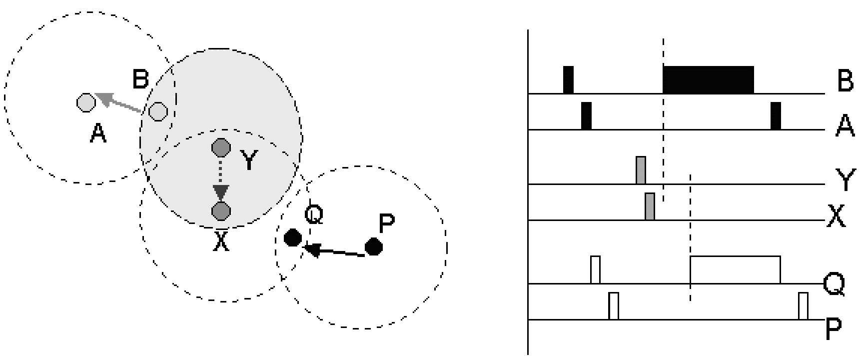


Fig. 6

Packet size for a master transmission: An important implication of a transmission being a master is that all overlapping transmissions must transfer data packets whose size is less than that of the master transmission. Otherwise, the data transfer of an overlapping transfer will interfere with the ACK of the master. In addition, MACA-P introduces a control gap before the data transfer phase. Therefore, (1) a sender with no existing master transmission in its neighborhood applies MACA-P on a packet only if the packet size is greater than the threshold, i..e for "large" packets, MACA-P is applied, while the base RTS/CTS mechanism is used for "small" packets, and (2) a node can overlap transmission with a master only if the node's data transmission time is no more than the master. Note that (2) implies that if the sender has a "small" packet to send and there exists a master transmission, then the sender will try to overlap the packet's transmission with that of the master.

MACA-P decrements the RTS backoff timer only when it is legitimate to send out an RTS, thus either when the channel is idle (including a potential control phase) and the sender is aware of at most 1 additional master transmission schedule and no neighboring receivers. This behavior is independent of the packet size; even for small sized packets (smaller than the threshold), MACA-P tries to align that packet with existing transmission schedules if possible. If the timer expires and the RTS attempt fails, then MACA-P follows the default 802.11 behavior of exponential doubling of the backoff timer

Re-transmitting a RTS: When a node sends a RTS and does not receive a CTS in response, it will transmit a RTS' to cancel the transmission schedule proposed in the RTS. As per the 802.11 basic access mechanism outlined earlier, it will double its congestion window and retry using a random timeout value within the congestion window. During this period, it is possible that the state of its neighborhood has changed. For example, on the first try, the node may be a master transmitter (i.e none of its neighbors had a scheduled transmission). However, during the backoff period, a neighbor may have issued a RTS and becoming a master transmitter in that process. Thus, when the node retries sending the RTS, it may not be able to use the same T_{DATA} as it used for the first/earlier try. It must retry the RTS based on its current neighborhood state.

4 MACA-P algorithm

The algorithm consists of three parts: a) *sender* : the steps executed by a node when it wishes to send a data packet b) *recipient* : the steps executed by a recipient to accept/reject a proposed transmission schedule, and c) *listener* : actions taken by a node on overhearing control messages from its neighbors that are not directly intended for itself.

Sender Node:

Step 1 *: If (node N has packet P to send)*
then wait till no neighbor is a recipient.
Step 2 *: If (#(master transmissions) > 1)*
then idle till exactly one master scheduled.
Step 3 *:*
If (exactly 1 master transmission scheduled)
then
if (size of P) > (master transmission size)
then
wait till master transmission complets
else
send RTS with inflexible-bit set with T_{data} and T_{ack} aligned with master
wait till CTS is received
if no CTS is received
then
send RTS' canceling prior RTS
retry RTS using 802.11 basic access method
else
Schedule transmission of P at T_{data}
Wait for ACK at T_{ack}
If (ACK is not received at T_{ack})
then Goto ***NoSuccess***

else / there are no master transmissions scheduled */*
if (size of packet) < threshold
then
use 802.11 RTS/CTS DCF / no control gap before data*/*
else
Node N is the master

```
                    Send RTS with inflexible-bit unset
                    Wait till CTS is received
                    If (CTS is not received)
                    then
                              send RTS' canceling prior RTS
                              N is no longer a  master
                              retry RTS using 802.11 basic access method
                    else
                               if  (CTS is received with modified T_data and T_ack)
                               then
                                        send RTS' with inflexible-bit set with modified T_data and T_ack
                                        schedule transmission of P after modified T_data
                                        wait for ACK after modified T_ack
                                        If (ACK is not received at modified  T_ack)
                                        then
                                                   Goto NoSuccess
                                        else
                                                  schedule transmission after T_data
                                                  Wait for ACK after T_ack
                                                  If (ACK is not  received at  T_ack)
                                                  then Goto NoSuccess

Goto Step1
NoSuccess :       Update #tries for packet P
                  If #tries > max-tries
                 then  Drop P from transmission queue
                 Goto Step1
```

Recipient Node R:

```
Step 1 : Wait till RTS is received from a node N
If  inflexible bit is set
then
           if  (R has no master recipient))
                    then
                              Set R as a master recipient
                              Send CTS back with same T_data and T_ack as RTS
                     else       /* cannot schedule a inflexible sender with an existing master */
                              No CTS is sent
                               Goto Step1
else
          if  #(master recipients for R) == 0
                     then
                              set R as a master recipient
                              Send CTS back same T_data and T_ack as RTS
                              Goto WaitforData

           If #(master recipients for R) == 1
                     then
                              Send CTS with modified T_data and T_ack aligned with existing master
                    else      /* (more than 1 master recipients) */
                              Do not send CTS
                              Goto Step1

WaitforData : Wait  for transmission of data from N at (modified)  T_data
               If transmission received correctly, then send ACK at  (modified) T_ack
              Goto Step1
```

Listener node L :

If a RTS or a CTS message is heard
then create an entry in L's NAV containing
a) MAC address of the sender of RTS/CTS
b) T_{data} and T_{ack} of the transmission
c) Sender is the data transmitter or recipient

If a RTS' message is heard,
Then update entry corresponding to sender of RTS' (or create one if none exists)

5 Conclusions and Future Work

In this paper, we first identified the reasons why the RTS/CTS based 802.11 MAC is incapable of supporting parallel transmissions. An important observation that we noted is that each node of sender-recipient pair switches roles between a transmitter and a receiver during the course of a 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. The key idea behind MACA-P is to introduce a control phase gap between the RTS/CTS and DATA/ACK phases. This allows two neighboring senders (recipients) to synchronise the start of DATA transmission (reception) and ACK reception (transmission). We defined the notion of a master transmission (reception) and identified the condition for parallel data transfers, viz. either the sender or recipient has at most one master transmission/reception in its immediate 1-hop neighborhood. We introduced an additional control message RTS' that is used by a sender node to inform its neighbors of a change in the DATA transmission and ACK reception times that was specified in a preceding RTS control message.

6 References

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

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

[dfwmac] 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.

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

[multi] 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.

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

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

[rate] G. Holland, N. Vaidya and P. Bahl. *A Rate-adaptive MAC protocol for multi-hop wireless networks.* Proceedings of ACM/IEEE Mobicom, July 2001.

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

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

[fama] C. L. Fullmer and J. J. Garcia-Luna-Aceves. *Solutions to hidden terminal problems in wireless networks*. Proceedings of ACM Sigcomm ’97, Sept ’97.

[direct] Y.Ko, V. Shankarkumar and N.H. Vaidya, Medium Access Control Protocols using Directional Antennas in Ad Hoc Networks. IEEE INFOCOM'2000, March 2000.

.