

Time Bounded Medium Access Control for Ad Hoc Networks

Raymond Cunningham
Distributed Systems Group
Trinity College Dublin
Ireland

raymond.cunningham@cs.tcd.ie

Vinny Cahill
Distributed Systems Group
Trinity College Dublin
Ireland

vinny.cahill@cs.tcd.ie

ABSTRACT

Most previous work on medium access control (MAC) protocols for wireless ad hoc networks has focused on the twin goals of maximising throughput and minimising average packet delay as required for general-purpose applications.

In this paper we describe a new MAC protocol for use in multi-hop ad hoc networks whose goal is to provide, with high probability, time-bounded access to the wireless medium for applications with guaranteed response time requirements.

The Time-Bounded Medium Access Control (TBMAC) protocol is based on time-division multiple access with dynamic but predictable slot allocation. TBMAC uses a light-weight atomic multicast protocol to achieve distributed agreement on slot allocation and employs location information to minimise contention for slots.

TBMAC is the first time-bounded MAC protocol for multi-hop wireless ad hoc networks. In this paper we describe the protocol and provide a number of time bounds for the transmission of messages.

Categories and Subject Descriptors

C.2 [Computer-Communication Networks]: Wireless Communication

General Terms

Algorithms, Design

Keywords

Mobile Ad Hoc Networks, Medium Access Control, Time Bounded

1. INTRODUCTION

With the increased research interest in ad hoc networks in recent years, a number of new application domains such as mobile robotics, traffic management and computer games

are possible. Communication in a timely fashion is an important building block to allow applications in these domains to be realised. In real-time systems in wired networks, ensuring timely communication is typically supported at the medium access control (MAC) layer (examples include Time Division Multiple Access, TDMA [16], and Controller Area Network, CAN [13]).

These approaches are inappropriate for ad hoc networks due to their underlying assumptions about the static nature of the wired network. Most previous work on medium access control protocols for multi-hop ad hoc networks has focused on maximising throughput in the ad hoc network and minimising average packet delay as required for general-purpose applications.

In this paper, we describe a new MAC protocol for use in multi-hop ad hoc networks whose goal is to provide, with high probability, time-bounded access to the wireless medium for real-time applications, i.e., those with guaranteed response time requirements. The Time-Bounded Medium Access Control (TBMAC) protocol is based on TDMA with dynamic but predictable slot allocation. TBMAC uses an atomic multicast protocol to achieve distributed agreement on slot allocation and employs location information to minimise contention for slots. TBMAC is the first time-bounded MAC protocol for multi-hop wireless ad hoc networks.

In the next section, we review existing MAC protocols for ad hoc networks paying particular attention to their timeliness properties. The following section gives a basic introduction to the TBMAC protocol. In section 3, we cover the atomic broadcast protocol used by TBMAC, followed by a more in depth description of the TBMAC protocol in section 5. We finish the paper with some conclusions and a discussion of future work.

2. RELATED WORK

In this section, we review some of the existing techniques used to control medium access in wireless ad hoc networks. Typically, there are two main approaches used to access the wireless medium in an ad hoc network:

1. Contention
2. Scheduled access

These two techniques will be covered with particular emphasis on their timeliness properties.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

POMC'02, October 30-31, 2002, Toulouse, France.

Copyright 2002 ACM 1-58113-511-4/02/0010 ...\$5.00.

2.1 Contention-based Approaches

In these approaches, mobile hosts contend with each other for access to the wireless medium when they have data to transmit. In MACA [10] sensing of the wireless medium before transmitting is not done. Each mobile host transmits a “Request To Send” (RTS) control packet causing mobile hosts that receive this packet to defer. The destination of the RTS packet replies with a “Clear To Send” (CTS) control packet. On receiving the CTS packet, the successful mobile host then sends its data packet.

There is a possibility of collisions of the RTS packets using MACA. If a mobile host does not receive the expected CTS packet correctly then a binary exponential back-off algorithm is executed.

MACAW [2] builds on MACA by introducing a Data-Sending (DS) packet to indicate to mobile hosts that the RTS/CTS packet exchange was successful. In addition, MACAW changes the back-off algorithm to include the current value of the back-off counter in the packet header field. Both IEEE 802.11 and FAMA-NTR [7] also build on top of MACA with each mobile host using carrier sensing before transmitting.

In each of these schemes, a mobile host does not have access to the wireless medium in a timely manner. In both MACA and MACAW, RTS control packets can easily be corrupted due to two or more mobile hosts transmitting at approximately the same time. The possibility of this occurring in 802.11 is reduced (but not eliminated) by each mobile host sensing the medium for a random number of time periods before transmitting.

In contrast to the above schemes, Sobrinho and Krishnakumar [15] use a black burst contention period to determine whether a mobile host gains access to the wireless medium or not. The mobile host sends a burst of energy of a known duration that is proportional to the length of time that the mobile host has been waiting to access the medium. After transmitting the burst, the mobile host listens for other mobile hosts transmitting a longer burst. If there is a longer burst, then the mobile host defers its transmission. Otherwise the mobile host transmits its packet.

Markowski and Sethi [12] use a Contention Resolution Algorithm (CRA) to support the co-existence of hard, soft, and non real-time data. A mobile host transmits its packet (of a fixed size) and then listens for feedback information from other mobile hosts as to whether a collision has occurred or not. If a collision has occurred, then the CRA is executed. The CRA at each mobile host uses a window of size n (where n is the known number of hosts participating in the protocol) and divides this into an active part and an inactive part. A mobile host in the active window transmits while those in the inactive window defer. If another collision occurs, then the hosts in the active window are split again into an active and an inactive part. This continues until a mobile host can successfully transmit.

The main disadvantage of the Sobrinho and Krishnakumar and the Markowski and Sethi medium access schemes is that they assume that participating mobile hosts are all within communication range of each other. Therefore, neither of these protocols deal with the well-known hidden terminal problem [17] and are not suitable for multi-hop ad hoc networks.

2.2 Scheduled-based Approaches

In this approach, mobile hosts negotiate a set of TDMA slots in which to transmit, so that these transmissions are as free of collisions as possible. Once a mobile host has a slot allocated to it, it then has access to the medium until a new set of TDMA slots is negotiated. However, if two or more mobile hosts have inconsistent views of the negotiated slots there is a possibility that these mobile hosts will corrupt each others’ transmissions. These schedule-based approaches can be broken into two categories: topology-dependent and topology-transparent scheduling.

Topology-dependent scheduling assigns slots to mobile hosts (or links between mobile hosts) within a two-hop neighbourhood. As the topology changes, the slots are reassigned in a distributed manner by the mobile hosts.

One approach, from Cidon and Sidi [4], uses a dedicated set of slots as a control segment to resolve conflicts and broadcast channel reservations. The number of dedicated slots used is proportional to the number of nodes in the network. Therefore, as the number of mobile hosts in the network increases, the number of dedicated slots also increases.

Another topology-dependent approach, from Bao and Garcia-Luna-Aceves [11], uses identifiers for one-hop and two-hop neighbours and a pseudo-random number generator to decide whether or not a mobile host (or a link between two mobile hosts) can transmit in the current time slot. Every mobile host uses the same initial seed for the pseudo-random number generator.

From a timeliness point of view, whether or not a mobile host can transmit in the slot depends on the current state of the pseudo-random number generator and the mobile hosts local two-hop topology information. Therefore a mobile host is unable to know when it will gain access to the wireless medium again in a predictable way. In addition, collisions can still occur using this approach when a mobile host does not have complete knowledge of the local one-hop and two-hop topology information, e.g., when network partitions begin to occur (or when they begin to merge).

In contrast to topology-dependent approaches, topology-transparent approaches, as proposed by Chlamtac and Farago [3], and Ju and Li [9], allocate a collection of slots to a mobile host to use. The underlying idea is that if a mobile host transmits in each of its allocated slots then any neighbour of this mobile host will correctly receive at least one transmission from the mobile host. This is made possible by each mobile host using a unique code that determines which slots a mobile host will use.

However, one limitation of these two topology-transparent approaches is that the maximum number of neighbours of any mobile host is known and bounded. Another limitation is that a transmitting mobile host does not know in which slot a particular neighbour can correctly receive their transmissions. Therefore, the mobile host must use all the slots allocated to it to ensure that at least one message is received correctly.

Another topology transparent protocol by Amouris [1] uses the position of a mobile host to determine in which slot that mobile host transmits. In this work, space is divided into virtual geographic cells called space slots in a similar way to cellular networking techniques. A slot is allocated to a cell and due to spatial reuse, this slot can be reused in cells sufficiently far away. If there is more than one mobile host

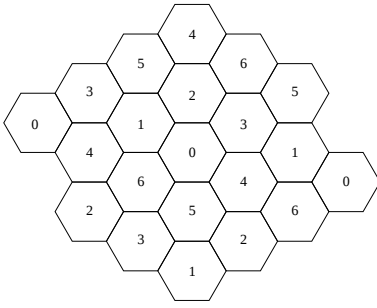


Figure 1: Possible Cell and Channel allocation

in a cell, then the slot is allocated in a round-robin fashion by each mobile host maintaining a sorted list of the mobile hosts in the cell.

On entering a cell, a mobile host broadcasts a packet to inform mobile hosts in the new cell and those in the old cell of its arrival/departure. Another mobile host, in the new cell, replies to this packet by transmitting a packet including the sorted list of mobile hosts located in the new cell.

There are a number of problems with this protocol in terms of a mobile host having predictable access to the wireless medium. Firstly, the author of the paper does not specify how a mobile host that powers on in a cell obtains a slot. A similar problem arises when two mobile hosts enter an empty cell or power on in an empty cell at the same time. Another problem is that the sorted list of mobile hosts is replicated by each mobile host in the cell. Therefore, updates to this replicated data structure need to be carried out in a consistent manner, otherwise there is a possibility of the sorted list becoming inconsistent across mobile hosts, which eventually results in collisions in the assigned slot.

3. TBMAC PROTOCOL BASICS

To provide each mobile host with time-bounded access to the wireless medium with high probability, TBMAC needs (i) to reduce the probability of the transmissions of two or more mobile hosts colliding and (ii) to detect collisions in bounded time and to take some action to prevent these collisions from recurring.

To reduce the probability of the transmissions colliding, the geographical area occupied by the mobile hosts is statically divided into a number of geographical cells in a similar approach to [1]. Each cell is numbered and can have arbitrary shape and size but for simplicity, we assume that the cells are hexagons of equal size as illustrated in Figure 1. In contrast to [1], each numbered cell is allocated a distinct radio channel (or CDMA spreading code) to use, maximising the total overall bandwidth available in the ad hoc network.

The motivation behind dividing the area of coverage into a collection of cells is to increase the probability of one mobile host hearing the transmission of any other mobile host in the cell and thus reducing the possibility of the well-known hidden terminal problem [17]. In order to achieve this, the width of a cell is related to the transmission range of the wireless technology being used.

The boundaries of each of these cells are known to each mobile host in the ad hoc network. To meet this requirement, each mobile host requires access to location informa-

tion (such as GPS). By a mobile host knowing the cell that it is in, it can then infer the correct radio channel to use.

We assume the presence of a higher layer above the TBMAC that decides in which cell the mobile host is currently located and therefore which radio channel the mobile host should use. The higher layer would continually monitor the current position of the mobile host (based on the update rate of GPS for example) and notify the TBMAC protocol of a change of cells. The operation of this higher layer will not be discussed further in this paper.

To further reduce the possibility of collisions, access to the wireless medium within a cell is divided into two time periods:

1. Contention Free Period (CFP)
2. Contention Period (CP)

Both the CFP and the CP are divided into slots and each period lasts a well-known period of time. Once a mobile host has been allocated a CFP slot, it has predictable access to the wireless medium. The mobile host can then transmit data in its slot until it leaves the cell or fails.

Mobile hosts, that do not have CFP slots allocated to them, contend with each other to request CFP slots to be allocated to them in the CP. The CP is used by mobile hosts that arrive into the cell or that have recently powered on in the cell. The steps carried out by a mobile host to be allocated a CFP slot are covered in section 5.

Dividing access to the medium into these two well-known time periods requires the clocks of all the mobile hosts in the network to be synchronised. Again equipping each mobile host with a GPS receiver would satisfy this clock synchronisation requirement. In addition to GPS, an internal clock synchronisation protocol could be executed to increase fault tolerance among the mobile hosts [14].

4. SYNCHRONOUS ATOMIC BROADCAST

The division into a CFP and a CP is similar to the Point Coordination Function in the 802.11 standard [8] with the exception that the TBMAC CFP does not rely on one particular mobile host to act as an access point. In the PCF of 802.11, agreement on accessing the wireless medium is enforced by the access point polling the mobile hosts for data to send. In the ad hoc environment that we are considering, we cannot assume there is an access point present in the area covered by the geographical cell.

One option would be to elect an access point from the mobile hosts present in the cell. This option has a number of problems. The mobile hosts have to reach a distributed agreement on which mobile host is to become the access point. Mobile hosts would also have to monitor the access point for failure and to reach agreement that the failure has occurred. This option has simply moved the problem of reaching distributed agreement to distributed leadership election and failure detection.

Instead of using an access point, the TBMAC participants reach a distributed agreement on the order of allocations and deallocation of CFP slots. Therefore, updates to the allocation of CFP slots in a cell need to be totally ordered across all mobile hosts in the cell. Two or more messages (e.g., allocating CFP slots) are seen in the same order by every mobile host in the cell.

The approach we use to provide a total ordering protocol within a cell is to use the synchronous atomic broadcast protocol from Flaviu Cristian [5]. Typical uses of this protocol within a cell are to allocate and de-allocate slots in the CFP and for requests to communicate across cell boundaries.

To explain how the synchronous atomic broadcast protocol works, consider a mobile host with a CFP slot that wishes another CFP slot to be allocated to it.

The mobile host creates a message requesting a slot, including information related to the atomic broadcast such as the mobile host's address, a sequence number and the current time (a timestamp). The mobile host then broadcasts this message a number of times using its CFP slot.

A receiving mobile host stores this packet if it has not received it before based on the sequence number. In addition to storing the packet, the mobile host also rebroadcasts the packet in its CFP slot until the delivery time of the information in the packet arrives.

The delivery time of the message is equal to the original timestamp plus the delay to delivery, Δ , which is a parameter of the atomic broadcast protocol. For example, the Δ for the TBMAC protocol would typically be 2 CFPs. When the delivery time of the message arrives, all the mobile hosts in the cell then update their information consistently and allocate the mobile host a new slot.

On first reading of the above description, it would appear that if a mobile host wishes to reach agreement with the other mobile hosts in a cell then it needs to retransmit the same message a number of times in its slot. However, since the information specific to the atomic broadcast is relatively small, this information can easily be piggybacked on data packets being transmitted.

5. TBMAC PROTOCOL DETAILS

In this section, we give an in depth description of the TBMAC protocol. Firstly, the data structures and packet types used by TBMAC are covered. Management of CFP slots in the TBMAC protocol is then described. Following this, the case when a mobile host enters an empty cell is discussed. Finally, a description is given of some possible extensions to the TBMAC protocol.

5.1 Data Structures

There are two important data structures maintained by each mobile host running the TBMAC protocol, these are Slot Owners and Slot Bitmap structures. Slot Owners is an array of addresses, CFP in size, which stores the address of each mobile host that has a CFP slot allocated to it. The Slot Bitmap contains two bits for each slot in the CFP to represent the four possible states of a slot: Owner, Other, Collision, Available.

When a mobile host sets a position in the Slot Bitmap to Owner, it is indicating to other mobile hosts that it is using this slot. Similarly, by setting a slot position to Other, the mobile host is indicating that another mobile host is using that slot. The identity of the mobile host that is using the slot is stored in the mobile host's Slot Owners array. A CFP slot is marked Available in the Slot Bitmap if the mobile host is not aware of any other mobile host using that slot. Finally, a CFP slot is marked in the Slot Bitmap as Collision when a mobile host has received a message (in another CFP slot) indicating that a collision is occurring in this slot (section 5.4).

Source Address	Destination Address	Slot Bitmap	Current Slot Number	Message Type	Protocol Extensions	Additional Info
----------------	---------------------	-------------	---------------------	--------------	---------------------	-----------------

Figure 2: CFP Slot Header

5.2 Protocol Header

When a CFP slot is allocated to a mobile host in a cell, the mobile host includes the CFP header (Figure 2) in each of its transmissions in its CFP slot. As part of this header, the mobile host includes its Slot Bitmap, the source and destination address and the current CFP slot number in the header.

The Message Type field is used to indicate the type of message being transmitted. The different possible types of messages are:

1. Data
2. Null
3. Slot Allocation Request
4. Slot Deallocation Request
5. Inter-Cell Communication Request

The need for the first message type is obvious. When a CFP slot is allocated to a mobile host, the mobile host sends a Null message in its slot even if it does not have a message to send. The remaining three message types are control messages.

Briefly, the Slot Allocation and Deallocation Request messages are used by mobile hosts to allocate and deallocate CFP slots. The Inter-Cell Communication Request message is used when a mobile host wishes to communicate across its current cell boundary with a neighbouring cell.

Finally, the Additional Info field is a variable length field and is used for extra information depending on the type of the message being transmitted.

5.3 Slot Management

Management of CFP slots can be broken down into three parts: Allocating a Slot, Deallocating a Slot and Inter-Cell Communication.

5.3.1 Allocating a Slot

When a mobile host powers on in a cell and requires a slot in the CFP to be allocated to it, the mobile host first needs to learn which CFP slots have been allocated and which CFP slots are available. The mobile host waits for the beginning of the next CFP and then listens for one full CFP to pass before requesting a slot in the following CP. The Slot Bitmap field in each occupied CFP slot indicates the number of slots allocated in the cell. By receiving one message in the CFP correctly, the listening mobile host obtains the number of CFP slots allocated and the position of each of these allocated CFP slots. The mobile host then requests some CFP slot to be allocated to it by sending a message in the CP.

A simple approach would be for a mobile host to request a slot by choosing a random slot in the CP and broadcasting to every mobile host requesting a slot to be allocated to it.

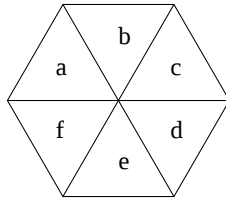


Figure 3: Subdivision of a Cell

Mobile hosts with CFP slots allocated, that correctly receive this request, then atomically broadcast this request. After the delivery of this atomic broadcast message, the mobile host can then access its allocated slot in the following CFP.

With this approach, there is a possibility of two or more hosts powering on within a cell at approximately the same time and choosing the same slot in the CP in which to broadcast their request and thus corrupting each other's packets. A way to reduce this possibility of collisions would be to further subdivide the cell as illustrated in Figure 3. Each subdivision of the cell would correspond to one or more slots in the CP.

When a mobile host powers on, it calculates which cell it is in and then which subdivision it is in. It then sends a request in the CP slot corresponding to the subdivision that it is in. By increasing the number of subdivisions, we decrease the possibility of two or more mobile hosts powering on in the same subdivision and therefore further reducing the possibility of collisions in the CP. The obvious disadvantage with increasing the number of subdivisions is that the number of slots in the CP needs to be increased accordingly, therefore reducing the network throughput and increasing the time between CFPs.

In the absence of collisions, the time bound before a mobile host has a CFP slot allocated to it equals $CFP + CP + CFP + CP + 2 * (CFP + CP)$. When a mobile host enters a non-empty cell, the worst case time bound, in the absence of collisions, occurs when a mobile host enters the cell just as the CFP begins. The mobile host must then wait until the end of this CFP and the following CP before listening at the beginning of the next CFP. The mobile host then broadcasts a request resulting in an atomic broadcast being transmitted that takes $2 * (CFP + CP)$ to be delivered.

5.3.2 Deallocating a Slot

The simplest way for a mobile host to deallocate one of its CFP slots is to atomically broadcast a message requesting the deallocation. This could happen if a mobile host is powering down or realises, based on location information, that it is leaving the cell.

It is more difficult to deallocate a CFP slot of a mobile host that has failed or has left the cell without firstly atomically broadcasting a request to deallocate the CFP slot (note that in the following discussion, departure from the cell and failure can be considered equivalent). In addition, we would also like to avoid deallocating a CFP slot of a mobile host that does not want its slot deallocated.

Each mobile host, that has been allocated a CFP slot, monitors each of the other allocated CFP slots for correct reception. If a mobile host does not correctly receive a number of messages from a failed mobile host, then it includes

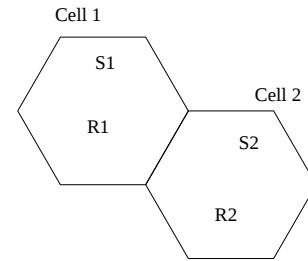


Figure 4: Adjoining Cells

this information in the Additional Info field of each of its CFP transmissions indicating the failure of the mobile host to use its slot.

When other mobile hosts receive this transmission, they incorporate the received Additional Info field into their CFP transmissions if they also have not correctly received a number of messages from the failed mobile host. After receiving a majority of such indications from mobile hosts in the cell, a mobile host atomically broadcasts a request for the CFP slot to be deallocated. After the delivery of the atomic broadcast message, the CFP slot of the failed mobile host is deallocated by each mobile host in the cell.

5.3.3 Inter-Cell Communication

As the area occupied by the mobile hosts has been divided up into geographical cells and each cell has been allocated a particular radio channel to use, an obvious question is how does a mobile host communicate with mobile hosts in neighbouring cells.

A simple solution to this problem would be to statically allocate the same CFP slot of two adjoining cells for communication between mobile hosts in these cells. When a mobile host wishes to use the inter-cell CFP slot, it atomically broadcasts a request to be allocated the CFP slot. Once this request is delivered, the mobile host begins to use the slot.

There is a problem with this approach since transmitting mobile hosts in each cell could collide with each others transmission. This is illustrated in Figure 4 where S1 in cell 1 has requested access to use the same inter-cell slot as S2 in cell 2. When S1 and S2 transmit, their packets collide.

If S1 uses cell 2's radio channel and S2 uses cell 1's radio channel then their transmissions will not collide. R1 will be listening for S2's transmission and R2 will be listening to receive S1's transmission. However, S1 will not be able to receive S2's transmission since it transmitted at the same time as S2.

The above problem can be avoided by using two CFP slots, one slot for communication from Cell 1 to Cell 2 and another slot from Cell 2 to Cell 1. As before, mobile hosts in each cell would atomically broadcast a request to be granted the inter-cell CFP slot. When a mobile host transmits in the inter-cell slot, all the mobile hosts in the adjoining cell will be listening for this transmission.

5.4 Empty Cells

Until this point in the paper, we have made a simplifying assumption that there is at least one mobile host with a CFP slot allocated to it in a particular cell. Therefore the obvious

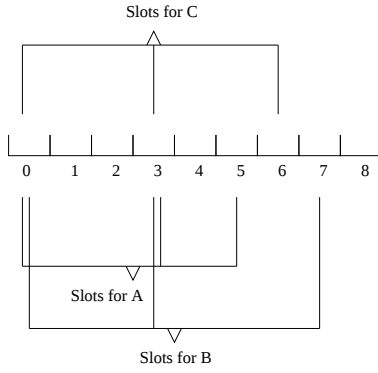


Figure 5: Generated list of slots

question is how is the first CFP slot allocated? Another question is what happens if two or more mobile hosts have an inconsistent view of the allocation of CFP slots? If an inconsistent view of the allocation of slots does arise, this would result in collisions in one or more of the conflicting slots.

5.4.1 Allocating a Slot

When dividing access to the medium into CFP slots and the CP slots, the most difficult task is to allocate the first CFP slot to a mobile host in a cell. Once there is one mobile host in the cell with a CFP slot then it is possible to use the atomic broadcast protocol described in section 4 to allocate and deallocate slots.

A simple approach would be for a mobile host to wait for another mobile host to transmit in the CP. These two mobile hosts could then negotiate with each other in the CP to agree on an allocation of slots in the CFP. The main problem with this approach is that two mobile hosts could choose the same CP slot in which to transmit and therefore corrupt each other's packets and therefore be unable to reach agreement. The likelihood of this occurring increases as the number of mobile hosts powering on within the cell increases and is also exacerbated by the fact that the CP is relatively small compared to the CFP.

Instead of the above approach, a mobile host in an empty cell generates a list of CFP slots to use after listening for one full CFP as illustrated in Figure 5. For each CFP slot in the generated list, the mobile host sets the corresponding two bit value of its Slot Bitmap to Owner. For all other slots, the two bit value for that slot is set to Available.

In Figure 5, mobile host A's list consists of slots 0, 3 and 5 while mobile host B's list consists of slots 0, 3 and 7. Mobile host A would also set position 0, 3 and 5 of its Slot Bitmap to Owner and set every other position to Available. By generating a list of slots to use, the probability of all slots in two or more mobile hosts' lists colliding is very small.

During the next CFP, the mobile host transmits in each CFP slot in its generated list (including the Slot Bitmap in each transmission). The mobile host also listens in each of the other CFP slots that it does not transmit in to gain knowledge about the presence of other mobile hosts and the CFP slots that they are using.

5.4.2 Obtaining a Consistent Allocation of Slots

When two or more mobile hosts enter an empty cell and

generate a list of CFP slots to use, each mobile host has an inconsistent view of the allocation of CFP slots. During the next CFPs, the goal of the TBMAC protocol is for each mobile host in the cell to converge on a consistent view of the allocation of CFP slots.

When a mobile host successfully receives a message from another mobile host, the receiving mobile host obtains a list of slots being used by the transmitting mobile host from the Slot Bitmap in the message (those positions in the Slot Bitmap marked as Owner).

For each slot, marked as being owned by the transmitting mobile host, the receiving mobile host checks if the slot is marked Available, Collision or Other in its own Slot Bitmap. If the position in the Slot Bitmap is Available, then the receiving mobile host marks the corresponding position in its Slot Bitmap as Other and stores the address of the transmitting mobile host in its Slot Owners array. However, if the position in the Slot Bitmap is marked as causing a Collision, then the mobile host does not update its Slot Bitmap. Finally, if the position in the Slot Bitmap is marked as Other, then the address of the transmitting mobile host is checked against the stored address in the Slot Owners array. If the address of the transmitting mobile host does not match the address stored then this slot is causing a collision. The receiving mobile host marks the corresponding position in its Slot Bitmap as causing a Collision and transmits the updated Slot Bitmap in its next CFP slot.

For each slot marked as Other in the transmitting mobile host's Slot Bitmap, the receiving mobile host checks the corresponding position in its Slot Bitmap. If the position in the receiving mobile host's Slot Bitmap is Available, then it is changed to Other. On the other hand, if it is already set to Other or Collision, then the receiving mobile host does not change its Slot Bitmap.

Finally, for each slot marked as causing a Collision in the transmitting mobile host's Slot Bitmap, the receiving mobile host marks the corresponding position in its Slot Bitmap as causing a collision. If a mobile host realises that one of its CFP slots is causing a collision, then the mobile host stops transmitting in this slot.

The above algorithm is run for a number of CFPs (M say). At the end of these CFPs, the mobile hosts in a cell converge on a consistent view of their Slot Bitmaps (in the presence of collisions and packet corruption due to transient errors). The underlying reason the above algorithm obtains a consistent view of the Slot Bitmap is because it is an example of an epidemic algorithm [6].

The Slot Bitmap of a mobile host, transmitted in a CFP slot, represents all the updates that that mobile host has performed on its allocation of CFP slots. In a similar way to [6], the probability of one or more mobile hosts having an inconsistent view of the allocation of CFP slots can be calculated.

If another mobile host arrives during these M CFP rounds, then the mobile host must wait until the end of the M CFP rounds, performing any necessary updates to its Slot Bitmap, before requesting a CFP slot to be allocated to it. Similarly, if all CFP slots in a mobile hosts list are causing a collision, the mobile host must also listen until the end of the M CFP rounds, before requesting a CFP slot to be allocated to it in the next CP.

To allow an arriving mobile host to know the number of rounds of the algorithm left to be executed, it is necessary

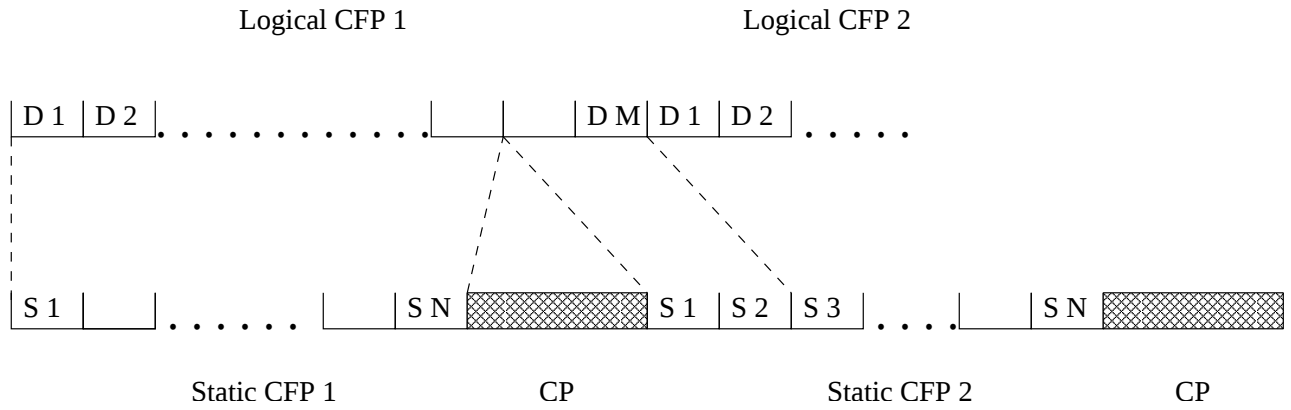


Figure 6: Mapping logical slots to static slots

for the mobile hosts to include the current CFP round in the Additional Info field of the CFP header.

The time bound before a mobile host in an empty cell has a CFP slot allocated to it equals $CFP + CP + CFP + CP$. However, the mobile host will not have a consistent view of the allocation of CFP slots until M CFPs later. Therefore, the more important time bound is $CFP + CP + CFP + CP + M * (CFP + CP)$.

5.4.3 Deallocating a Collision Slot

The obvious next question is how a slot marked as causing a collision is garbage collected and converted into a free slot available to be reused. TBMAC takes an approach similar to garbage collecting unused slots from failed or recently departed mobile hosts described earlier. A mobile host waits for a majority of mobile hosts to acknowledge the collision occurring in the slot before atomically broadcasting a message to deallocate the slot.

5.5 Extensions

By fixing the number of slots in the CFP, it would appear that we are placing an upper bound on the number of mobile hosts that can be allocated a CFP slot in the cell at any one time. This would mean that the TBMAC protocol is very restrictive and has only limited use.

It is possible however to overcome this restriction by allowing a dynamic logical CFP (LCFP) to grow and shrink in the repeating static CFPs. This is illustrated in Figure 6. Each static CFP has the same number of slots, N and it repeats every $CFP + CP$ time units.

To allow the LCFP to grow and shrink, the TBMAC needs to include a Repeat Duration value in the Protocol extensions field in the header of each TBMAC message. The Repeat Duration field contains the number of CFP slots before the allocation of slots repeats itself. Additionally, the size of the Slot Bitmap in the CFP header of each TBMAC message would also need to grow and shrink.

Another possible extension to the TBMAC protocol would be allow a dynamic number of CFP slots to be allocated for inter-cell communication between two adjoining cells. To allocate such an extra CFP slot would require two atomic broadcasts, one in each of the two adjoining cells.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have introduced a new MAC protocol, called TBMAC, for use in multi-hop ad hoc networks. This new protocol provides time bounded access to the wireless medium in a multi-hop ad hoc network. Future work includes calculating probabilities of each of these time bounds being obtained and comparing these calculated probabilities to the performance of the TBMAC protocol.

7. ACKNOWLEDGMENTS

The work described in this paper was partly supported by the Irish Higher Education Authority's Programme for Research in Third Level Institutions cycle 0 (1998-2001) and by the Future and Emerging Technologies programme of the Commission of the European Union under research contract IST-2000-26031 (CORTEX - CO-operating Real-time sentient objects: architecture and EXperimental evaluation). The authors are grateful to past and current colleagues at Trinity College Dublin as well as to Jörg Kaiser of the University of Ulm for their valuable input.

8. REFERENCES

- [1] K. G. Amouris. "Space-Time Division Multiple Access (STDMA) and Coordinated, Power-Aware MACA for Mobile Ad Hoc Networks". In *IEEE Symposium on Ad Hoc Wireless Networks (SAWN2001)*, San Antonio, Texas, 2001.
- [2] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. "MACAW: a media access protocol for wireless LAN's,". *ACM SIGCOMM'94*, pages 212–225., 1994.
- [3] I. Chlamtac and A. Farago. "Making transmission schedules immune to topology changes in multi-hop packet radio networks". *IEEE/ACM Transactions on Networking*, 2(1):23–29, 1994.
- [4] I. Cidon and M. Sidi. "Distributed Assignment Algorithms for Multihop Packet Radio Networks". *IEEE Transactions on Computers*, 38(10):739–746, Oct. 1989.
- [5] F. Cristian. "Synchronous Atomic Broadcast for Redundant Broadcast Channels". In *Journal of Real-time Systems*, pages 195–212. Kluwer Academic Publishers, 1990.

- [6] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry. "Epidemic Algorithms for Replicated Database Maintenance". In *The Sixth Symposium on Principles of Distributed Computing*, pages 1–12, Aug. 1987.
- [7] J. J. Garcia-Luna-Aceves. "Floor Acquisition Multiple Access (FAMA) for Packet-Radio Networks,". In *ACM SIGCOMM'95*, pages pp. 262–273., 1995.
- [8] IEEE. "IEEE std 802.11 - wireless lan medium access control (MAC) and physical layer (PHY) specifications", 1997.
- [9] J. Ju and V. Li. "An optimal topology-transparent scheduling method in multihop packet radio networks". *IEEE/ACM Transactions on Networking*, 6(3):298–306, June 1998.
- [10] P. Karn. "MACA - a new channel access method for packet radio." In *ARRL/CRRL Amateur Radio 9th Computer Conference*, pages 134–140, Ontario, Canada, 1990.
- [11] B. Lichun and J. J. Garcia-Luna-Aceves. "A new approach to channel access scheduling for ad hoc networks". In *7th Annual International Conference on Mobile Computing and Networking*, Rome, Italy, 2001.
- [12] M. J. Markowski and A. S. Sethi. "Fully Distributed Wireless MAC Transmission of Real-Time Data". In *Fourth IEEE Real-Time Technology and Applications Symposium*, June 1998.
- [13] Robert Bosch GmbH. "CAN specification version 2.0", 1991.
- [14] K. Romer. "Time Synchronisation in Ad Hoc Networks". In *The ACM Symposium on Mobile Ad Hoc Networking & Computing*, Long Beach, California, USA, Oct. 2001.
- [15] J. L. Sobrinho and A. S. Krishnakumar. "Real-time Traffic over the IEEE 802.11 Medium Access Control Layer". Technical report, Bell Labs, 1996.
- [16] A. S. Tanenbaum. "*Computer Networks*". Prentice Hall, 1996.
- [17] F. Tobagi and L. Kleinrock. "Packet Switching in Radio Channels: Part II—The Hidden Terminal Problem in Carrier Sense Multiple Access and the Busy-Tone Solution". *IEEE Transactions on Communications*, 23(12):1417–1433, Dec. 1975.