An Energy Efficient Real-Time Medium Access Control Protocol for Wireless Ad-hoc Networks

Maria Adamou, Insup Lee, and Insik Shin
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104
email:{adamou, lee, ishin}@saul.cis.upenn.edu

ABSTRACT

This paper presents the design and analysis of an energy efficient real-time Medium Access Control (MAC) protocol for wireless ad-hoc networks. We propose a fully distributed reservation scheme to provide bandwidth guarantees to real-time sessions as well as significant energy savings, taking advantage of location information of the nodes. In this paper, we describe the protocol and present the correctness proof.

I. INTRODUCTION

The issues of power saving mechanisms and real-time traffic support over wireless ad hoc networks have received considerable research attention. Several studies have been performed focusing on real-time communications [1][2][3][4] and power management [5][6] over ad hoc networks. However, to our knowledge, no study has been performed on addressing the both issues simultaneously. In this paper, we propose a Power-Aware Reservation-based Medium Access Control (PARMAC) protocol for wireless ad hoc networks with the following objectives. Firstly we provide an energy efficient communication mechanism which eliminates the causes of unnecessary power consumption and allows the node's network interface to be switched off as much as possible. Secondly, our protocol supports real-time traffic with specific delay and bandwidth constraints.

In order to achieve the above goals, our protocol employs a fully distributed reservation scheme. Several characteristics of ad hoc networks such as the hidden terminal problem [7] and mobility, make it difficult for a distributed reservation scheme to work correctly. To overcome this difficulty, our protocol takes advantage of the location information of the nodes [8]. We prove the correctness of our protocol by establishing that all the nodes have consistent information on the reservations.

The rest of the paper is organized as follows. Section II describes the system assumptions and the protocol. Section III discusses the correctness of our protocol and Section IV presents the salient features and trade-offs of our protocol. Finally, our conclusions and some future work are discussed in Section V.

II. ENERGY EFFICIENT REAL-TIME MAC PROTOCOL

A. System assumptions

We assume a network of a total of N hosts, each having a unique ID i, $0 \leq i < N$. Due to limited transmission range, not all hosts are in range of each other, thus each host has a set of neighbors, $R(i) = \{k | k$ is in the wireless transmission range of $i\}$. Each node knows the ID's of its neighbors. We also denote the set $R(\{i, j\})$ as the nodes that are in $R(i) \bigcup R(j)$, for two nodes i and j. The set of hosts that are hidden from i is denoted as $H(i) = \{k | k$ is not in the wireless transmission range of $i\}$. The nodes in $H(i)$ cannot hear the messages from node i and cannot transmit to i.

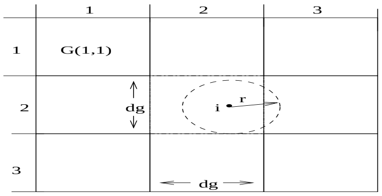

Fig. 1. A grid G(x,y) and its neighboring grids.

We assume that the geographic area of the network is divided into rectangular *grids* of size d_gxd_g. Each grid is labeled as $G(x, y)$, where (x, y) are its xy-coordinates. Each host can identify its location, i.e., its xy-coordinates through a positioning device (e.g., GPS) and can determine its grid number. Thus, there is a mapping between each host's i position (x_i, y_i) to a grid $G(x, y)$. Let r be the transmission range of the hosts. Then $d_g \geq r$, thus a host in grid $G(x, y)$ cannot reach a host in grids $G(x \pm 2, y \pm 2)$. A host i can only reach by one hop the hosts in neighboring grids (see Figure 1).

We also make the following assumptions under which our protocol operates.
(a) Power constraints: We assume that all stations have limited power.
(b) Synchronization: All stations are synchronized to a global time, according to some synchronization scheme; for example, they may use their GPS receiver to achieve this.
(c) No channel error: The packets can be lost only due to collisions or missed deadlines.
(d) Admission control has been performed already, so that the offered load can be sustained by the network.
(e) Mobility: We assume that the stations are not moving.

Our protocol is designed to support real-time and non-real time traffic. A real-time session is specified by an establishment deadline d_{res}, which is the time by which the session has to send its first packet. Each real-time packet has also a deadline d_p, which is the time by which the packet should be sent, otherwise it is discarded. We consider that these deadlines are *"soft"*, i.e.,

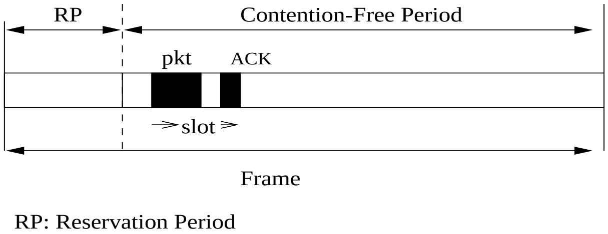

Fig. 2. The structure of a Frame

	1	2	3	4	5
1	t1	t2	t3	t4	t5
2	t6	t7	t8	t9	t10
3	t11	t12	t13	t14	t15
4	t16	t17	t18	t19	t20
5	t21	t22	t23	t24	t25

Fig. 3. Assigning reservation slots in a 5x5 grid area

the number of missed deadlines should not exceed a maximum packet loss rate.

B. Protocol Description

We now describe our protocol, called Power-Aware Reservation-based Medium Access Control (PARMAC). We assume that all stations are synchronized, using a global synchronization scheme, and time is divided in frames of fixed length. Each frame is divided as shown in Figure 2, into the Reservation Period (RP), during which nodes contend to establish new reservations or cancel reservations, and the Contention-Free Period (CFP), during which they send data packets without contention during the reserved transmission windows and sleep when they do not have a packet to send or receive. The lengths of the frame, CFP and RP are pre-specified in the network. The Reservation Period is divided in "Reservation Period slots (RP-slots)", each having a fixed length (which is enough for three reservation messages to be transmitted). The Contention-Free period is divided in a Contention-Free (CF) slots. Each CF-slot has fixed length, long enough for a transmission of a data packet and an acknowledgment. As in [1], each station i keeps the reservation information in a Reservation Table (RT_i), which keeps track of the IDs of the nodes (within range) that are transmitting or receiving data during each CF slot. When a node joins the network it has to stay awake for some period (multiple of frame size) to hear the ongoing transmissions and update its RT.

Reservation Period

During the Reservation Period, two types of "reservation procedures" can take place, the *Reservation Establishment* procedure and the *Reservation Cancelation* procedure. A station that needs to initiate a Reservation Establishment or Reservation Cancelation can do so in the pre-specified Reservation Slot for its grid. A host in a grid $G(x, y)$ can initiate a reservation procedure only during the reservation slot t such as $t = 5x + y + 1$ (see Fig. 3). This ensures that only one reservation procedure in a rectangular area of **5x5** grids can take place in one reservation slot. This is essential to the correctness of our protocol as shown in Section III.

When a station needs to make a reservation establishment or cancellation, it senses the channel to determine if another station of the same grid is transmitting. The station proceeds with a transmission if the medium is determined to be idle for a specified interval. If the medium is found to be busy, or the first message is not sent successfully, by a sender then the exponential backoff algorithm is followed and the station chooses a subsequent frame to re-initiate its request.

The *Reservation Establishment* procedure (see Figure 4(a)) takes place for a real-time station every time a new real time session begins, e.g., when it produces a new talkspurt. Datagram (non-real-time) traffic is not sensitive to delay, thus nodes may buffer the packets up to a "burst length" β and then make a request for sending the whole burst. The reservation establishment involves the exchange of three control messages:
(a) A Reservation Request (`RR(i,j)`) is sent by a node i to a neighbor j for which it has real-time or non-real-time packets. The `RR` includes the free slots in the RT of the sender and the packet length. We define the notion of a free slot and how a node i chooses its free slots in Section III. The `RR` for a real-time session also includes the deadline of its packets, while for a non-real-time station it includes the number of buffered packets of the sender.
(b) A Reservation Acknowledgment (`RA(j,i)`) is sent by a node j to a neighbor i. The receiver (node j) compares (as described later) the free slots of the sender (which are included in the `RR` message) with its own free slots to find the common free slots. Then, it schedules the requested packet in a common free slot in its Reservation Table. Then, the receiver indicates in the `RA` which reserved slot(s) the sender should use. With this message, all the nodes in $R(j)$ become aware of the reservation.
(c) A Reservation Broadcasts (`RB(i,j)`) is sent by a node i to a node j and includes the reserved slots that i has reserved. Thus all the nodes in $R(i)$ become also aware of the reservation.

The *Reservation Cancelation* procedure is illustrated in Fig.4(b). It is invoked when a sender has no more packets to send during its real-time session. Two messages are involved in the Reservation Cancelation: (a) The Reservation Cancel (`RC(i,j)`) sent by a node i to j and (b) the Reservation Cancel Acknowledgment (`RC-ACK(j,i)`), sent by node j to i. Both messages include the session ($i \rightarrow j$) that needs to cancel its reservation and the respective slot.

Contention-Free Period

During the CFP, the stations wake up in the predetermined transmission slots according to their RTs, to send or receive data packets, and sleep the rest of the period. In each slot, the sender sends a data packet, with size specified by the sender and receives an acknowledgment (ACK) sent by the receiver. If a node does not have any data to send or receive during a CF-slot, then it switches off its NIC. Once the reservation for a real-

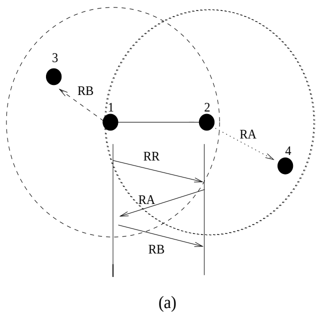

Fig. 4. (a) Reservation Establishment (b) Reservation Cancellation

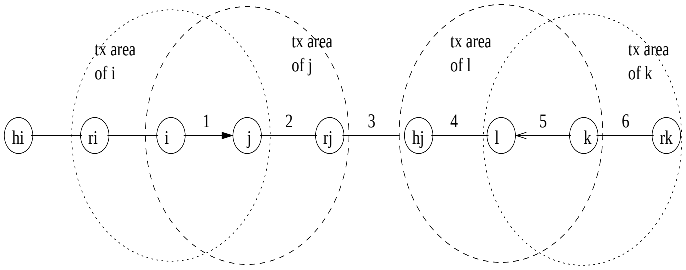

Fig. 5. Two simultaneous reservations in a 5-hops away neighborhood

time session is established, it is kept until an explicit reservation cancelation is performed as described above. The sender will use the reserved slot to send its data packets until the session is over. Reservations for datagram traffic are valid only for the current frame and the hosts clear their RTs for those slots that have non-real-time transmissions, after the CFP is over. Thus no explicit cancelation is needed in case of datagram reservations.

III. CORRECTNESS OF PARMAC

In this section we discuss the correctness of PARMAC.

Definition 1: A Reservation Protocol is correct if

- When there is no ongoing reservation, the Reservation Tables of all nodes in the networks are consistent
- When a node i needs to reserve a slot for sending to j and there is a common available slot s, then the protocol makes a reservation $i \rightarrow j$ during the slot s, as long as it is allowed by RTs consistency rules. The reservation means that i will send its data packets to j during slot s without collision.
- When a node i no longer needs to use a reserved slot s for sending to j, then the protocol removes the reservation from all RTs that include it.

A. Reservation Tables Consistency

In reservation-based protocols, like ours, the Reservation Tables of the nodes need to be consistent to perform correctly. We now give a definition of the reservation tables consistency and then explain why it is important. Assume that node i has a reservation to send to node j during the slot s of the Contention-Free Period.

Definition 2: The Reservation Tables of all nodes in the network are *consistent* if there is an entry in slot s in the RTs of $R(i)$ and $R(j)$ that indicates the transmission $i \rightarrow j$, then
(1) There is no entry in slot s, in the RT of any node in the network, which indicates that node j is receiving a packet by any node, other than i, and
(2) There is no entry in slot s in the RT of any node in the network, that indicates that any node in $R(i)$ is the receiver of any transmission during slot s.
Furthermore, there is no slot t in the RT of any node in the network that indicates the transmission $i \rightarrow j$ if i is not sending to j during slot t.

According to the above definition, all nodes in $R(i)$ need to know about the reservation, since they will not be able to receive any other packet during that slot. Also, all nodes in $R(j)$ should be informed about the reservation, since, if they send another packet during the same data slot, they cause a collision at j. Thus, all nodes in $R(\{i,j\})$ must have the reservation $i \rightarrow j$ for slot s in their tables. Also, the packet that i sends to j will not collide with any other packet sent to (or overheard by) j and also will not cause any collision with any other packet that is sent to a node in $R(i)$. Finally, there is always a transmission indicated by a reservation, which means that the slot is not mistakenly taken for reserved. Thus, the reservation tables consistency is very important so as no collisions in data packets can occur during the CF Period and also all the slots can be successfully reserved.

In order for the nodes to preserve the consistency of their RTs, they have to choose free slots in their RTs, according to the following definition.

Definition 3: Any slot s in RT_i of a node i, is considered free depending on whether i wants to send or receive during the slot, according to the following rules.
Case (a): If node i wants to make a reservation to send to node j, then a slot s is free for i if:

- Node i is not receiving or sending during x.
- No neighbor k of i (i.e., $k \in R(i)$) is receiving during x.

Case (b): When node j wants to make a reservation to receive from node i, a slot t is free if:

- Node j is not receiving or sending during y.
- No neighbor l of j (i.e., $l \in R(j)$) is sending during slot t.
- Slot t is also free for node i.

B. Correctness proof

We now discuss how our protocol performs correctly under the assumptions of Section II. In a multihop network the hidden stations can cause collisions to control packets and reservation information be lost. If reservation information is lost then the reservation tables can become inconsistent and collisions in data packets may occur or CF slots maybe wasted. When a `RR` packet is collided, no reservation information is lost, since `RR` message does not carry any reservation establishment information. When `RA` or `RB` packets are lost, conflicting reservations can happen, which may result in data packets collisions. When `RC` or `RC-`

ACK packets are lost, then reservation cancelation information maybe lost and the slots may not be able to be reserved for other hosts, thus data slots remain unused.

We assume that a node i initiates a reservation procedure with node j, that involves CF-slot s, which we indicate as $i \to j$, and we refer to the network shown in Figure 5. In order to prove the correctness of our protocol we use the following lemmas. Their proofs are ommited due to lack of space.

Lemma 1: A node k in the network can cause a reservation message to be missed by a node in $R(i,j)$, during a Reservation Procedure $i \to j$ is taking place, iff k is 1,2,3 or 4 hops-way from the sender i.

Lemma 2: All Reservation Messages are received successfully by the nodes in $R(\{i,j\})$ during the time that any reservation procedure $i \to j$ is taking place, iff any node i in $G(x,y)$ initiates a reservation procedure at RP-slot $t = 5x + y + 1$.

Lemma 3: The protocol ensures that all nodes in $R(\{i,j\})$ successfully update their reservation tables whenever a reservation procedure $i \to j$ is taking place.

Lemma 4: In order for all RTs to be consistent, whenever a reservation procedure $i \to j$ is completed, a node k needs to update its RT only iff $k \in R(\{i,j\})$.

We also give the following theorems.

Theorem 1: The RTs in all the nodes in the network are consistent when no reservation procedure is taking place and after the completion of any reservation procedure.

Theorem 2: The PARMAC protocol ensures that the Reservation Tables are consistent before any Reservation Procedure is made and after it is completed. Also the protocol always establishes a reservation whenever there is a free slot and cancels a reservation whenever it is not needed.

IV. DISCUSSION OF SALIENT FEATURES AND TRADE-OFFS

We now discuss the properties of our protocol regarding power savings and real-time traffic guarantees. Energy is consumed when a node is sending or receiving a packet and also when it is idle, waiting for a packet to receive. Finally some energy is consumed when a node is sleeping, but this is significantly less than the others. The energy savings we expect from our protocol to achieve are as follows. Firstly, our protocol minimizes the idle time and allows the nodes to sleep during the Contention Free Period, during the slots that they do not have reservations for either transmitting or receiving packets. Secondly, by following the reservations the data packets are less possible to collide, thus retransmissions are minimized and the total number of packets sent or received is less. Finally, only five (or three for non-real-time traffic) control messages are needed to be exchanged successfully, for a real-time session (or a burst of non-real time packets), whereas in other protocols, such as 802.11 at least two control messages are needed for each *packet* transmission, thus the energy for sending and receiving the control messages is reduced.

We now discuss how our protocol can guarantee the support of real-time sessions. According to the protocol, when a node succeeds in sending RR, then the RA and RB are also sent successfully. If there is an available slot then the real-time session will make a successful reservation for its whole duration (see Theorem 2). Thus, if a station succeeds in establishing a reservation before the deadline d_{res}, then it can be guaranteed that the real-time packets are sent before the deadline d_p.

Two of the assumptions we make, in order to prove the correctness of our protocol are that the nodes are not moving and there are no errors in the channel. If we allow any of these, then any Reservation Message can be lost, thus, the nodes in $R(i,j)$ are not guaranteed to receive successfully all the reservation messages. In these cases, the consistency of the reservation tables cannot be guaranteed.

V. CONCLUSION AND FUTURE WORK

The design and analysis of an energy efficient real-time MAC protocol have been presented. The protocol aims to provide energy-efficient real-time communications in a mobile ad hoc network, using a fully distributed reservation scheme. The protocol is proved to be correct, that is, to provide consistent reservations while satisfying the traffic requirements. This is true when no errors and no mobility is assumed. As future work we plan to simulate our protocol and compare its performance with similar reservation-based protocols and 802.11. It is expected that PARMAC achieves high power savings, while being able to support real-time traffic with minimum packet deadline misses. We also need to consider the issues of mobility and errors in the channel and how they can be treated so as to preserve the correctness of our protocol. Finally, we are also considering how to dynamically adjust some of the system's parameters (such as the RP length) to better achieve our goals.

REFERENCES

[1] C.R. Lin and M. Gerla "MACA/PR: An Asynchronous Multimedia Multihop Wireless Network", Proceedings of IEEE INFOCOM'97.

[2] Andrew Muir, J. J. Garcia-Luna-Aceves, "An Efficient Packet Sensing MAC protocol for wireless networks", Mobile Networks and Applications, Vol 3, pp 221-234, 1998.

[3] Z. Tang and J.J. Garcia-Luna-Aceves, "A Protocol for Topology-Dependent Transmission Scheduling", Proc. IEEE Wireless Communications and Networking Conference 1999 (WCNC 99), New Orleans, Louisiana, September 21–24, 1999.

[4] C. Zhu and M. S. Corson, "A Five-Phase Reservation Protocol (FPRP) for Mobile Ad Hoc Networks", in Proceedings of the Conference on Computer Communications (IEEE Infocom), (San Francisco, California), pp. 322, March/April 1998.

[5] S. Singh and C.S. Raghavendra. "PAMAS - Power Aware Multi-Access protocol with Signalling for Ad Hoc Networks", ACM Computer Communication Review, July 1998.

[6] Laura Feeney and Martin Nilsson. "Investigating the Energy Consumption of a Wireless Network Interface in an Ad-Hoc Networking Environment", In Proceedings of the IEEE Infocom 2001 Conference, Anchorage, Alaska USA , April 2001.

[7] F. A. 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, Vol. COM-23, No. 12, pp. 1417-1433, Dec. 1975.

[8] Wen-Hwa Liao, Yu-Chee Tseng, and Jang-Ping Sheu "GRID: A Fully Location-Aware Routing Protocol for Mobile Ad Hoc Networks", http://citeseer.nj.nec.com/310846.html.