

# Soft Real-Time Chains for Multi-Hop Wireless Ad-Hoc Networks

Bach D. Buy, Rodolfo Pellizzoni, Marco Caccamo, Chin F. Cheah, Andrew Tzakris  
*Department of Computer Science, University of Illinois at Urbana-Champaign*  
{bachbui2, rpelliz2, mcaccamo, cheah, tzakis}@uiuc.edu

## Abstract

*Prioritized MAC protocols are needed to support soft real-time communication in wireless networks. In this paper, we introduce real-time chain, a new prioritized MAC protocol to support soft real-time data flows in multi-hop wireless ad-hoc networks. By avoiding packet collisions and limiting the effect of priority inversions, real-time chain is able to provide soft real-time and bandwidth guarantees. Furthermore, the use of multiple channels enables high spatial reuse and transmission rates. Finally, we can achieve compatibility with IEEE 802.15.4 after a minor modification to the standard. The protocol has been fully implemented on Crossbow MICAz hardware and its performance has been validated with a large set of both indoor and outdoor experiments.*

## 1. Introduction

In recent decades, computer systems have been embedded into physical environments for automated real-time sensing and control. This trend will continue, and even expand, to improve and secure our quality of life in many areas such as defence, counter-terrorism, emergency rescue, and bio-sensing. Sensor networks are becoming an attractive solution for a variety of applications ranging from forest/wildlife monitoring to surveillance of borders. Multi-hop wireless ad-hoc communication is the popular networking paradigm for sensor networks: an example of embedded sensing device is Crossbow MICAz which has a wireless transceiver compatible to IEEE 802.15.4 standard. Since many sensory applications are characterized by low rate sensory data (temperature, humidity, pressure), and end-to-end data delivery can have a latency of seconds/minutes, researchers have not focused much attention on devising communication mechanisms able to deliver high bandwidth data flows subject to end-to-end soft real-time guarantee. As embedded devices become cheaper and more powerful, Wireless Multimedia Sensor Networks (WMSNs) can be envisioned as next generation multi-hop wireless ad-hoc networks where sensor nodes are equipped with micro-

phones/cameras and real-time audio/video streams<sup>1</sup> can be established on demand [16, 5]. Wireless multimedia sensor networks will enable new applications and will enhance existing surveillance and monitoring systems used for border control, disaster management, emergency rescue.

Several works have provided some degree of temporal QoS in multi-hop wireless ad-hoc networks (see Section 6), but to the best of our knowledge, none of them is capable of supporting non-realtime (CSMA/CA like) traffic and prioritized real-time data flows (subject to soft real-time guarantee) without assuming a regular network structure. Each real-time data flow can be characterized by a real-time priority assigned according to data criticality. This work assumes statically located or slow moving nodes (lifetime of existing routes is of the order of seconds/minutes); however, nodes can be placed irregularly, do not need synchronized clocks and do not rely on centralized mechanisms (i.e., masters, network coordinators).

To make the idea of real-time wireless ad-hoc networks work, two main problems need to be addressed: 1) packet collisions on wireless channel; 2) priority inversions when accessing wireless medium. This work is one of the first attempts at addressing these challenging problems together and providing a suitable real-time MAC protocol. A packet collision occurs when two nodes nearby decide to transmit a packet at the same time resulting in a collision; priority inversion occurs each time a node carrying a high priority packet loses wireless medium contention against another node that has a lower priority packet: an ideal real-time ad-hoc network should not experience any packet collision and should experience only bounded priority inversions. To avoid packet collisions and mitigate the priority inversion problem caused by the multi-hop nature of large scale networks, our work exploits a prioritized medium access scheme based on Black-Burst (BB) protocol [20]. The original BB scheme was designed to achieve fair channel sharing in wireless LAN. Here, we are only interested in the BB channel contention scheme, and we adapted it to cope with a multi-hop network scenario (our implementation on Crossbow MICAz platform allows to differentiate among eight real-time priority levels). Experimental test-

---

<sup>1</sup> In this work we used Crossbow MICAz motes to build a multi-hop sensor network able to carry in real-time low quality audio sampled at about 4KHz.

ings in an indoor and outdoor multi-hop environment drove our design choices and the introduction of the novel idea of **Real-Time Chain**: a multi-hop real-time data flow that can be established on demand, it is characterized by a real-time priority, it allows good spatial reuse of the wireless medium by exploiting multiple channels, it is not subject to any wireless interference caused by best effort (non real-time) traffic and it is compatible with IEEE 802.15.4 (after a minor modification to the IEEE standard). It is worth noticing that even if *real-time chains* were implemented on low power embedded devices like MICAz motes, the devised multi-hop mechanism can also be suitable for other classes of devices: in fact, real-time chain only assumes that the wireless transceiver supports CSMA/CA medium access control and provides a reasonable number of non-overlapping channels (e.g., IEEE 802.11a physical layer provides eight distinct channels). As a final remark, it is important to notice that the wireless medium is essentially unreliable despite any efforts of building an ideal collision free MAC; as such, it was out of the scope of this work trying to guarantee hard real-time constraints. Instead, we focused on providing soft real-time guarantees through the development of a robust, practical and easy to implement protocol that does not rely on strong assumptions hard to be met in large scale distributed systems like sensor networks. Therefore, all the soft real-time bounds expressed in the paper are provided subject to the assumption that the wireless medium is not affected by jamming or EMI. The rest of the paper is organized as follows. Section 2 introduces some terminology and assumptions we will use in the rest of the paper; Section 3 introduces the network architecture and describes the basic idea behind the proposed approach; Section 4 illustrates important implementation details; Section 5 shows some experimental results; Section 6 presents the related work; and Section 7 contains our conclusions and future work.

## 2. Terminology and Assumptions

We consider a wireless ad-hoc network composed of nodes with static position and a single radio transceiver, capable of transmitting at a maximum bandwidth of  $B$  bytes/second. Each node  $N_i$  is able to communicate with a set of neighboring nodes but not with the rest of the network, i.e. the network is not fully-linked. Hence, whenever a source node  $N_i$  wants to transmit a packet to a destination node  $N_j$  which is not within its neighborhood, the packet must be relied through a series of intermediate nodes (hops). We assume that a suitable geographic routing protocol is available in the network [9] and implemented at the network layer, such that if  $N_i$  knows the physical position of node  $N_j$  and embeds the information inside a packet, each intermediate node can determine the next hop towards the destination node. Traffic differentiation is provided at the MAC layer. Best effort traffic is served by a CSMA/CA protocol with random backoff like the standard IEEE 802.15.4 protocol, which is especially tailored for sensors or home automation. Periodic real-time traffic is served by our new MAC protocol and specified as a set of data flows, i.e. com-

munication channels between a source and a destination node over which periodic traffic is sent for a defined interval of time  $[t'_i, t''_i]$ . Let  $r_i$  be the desired rate at which packets are injected in the flow by the source node, and  $n_i$  be the constant length in bytes of all packets. For simplicity, we assume that the intermediate hops selected by the network protocol remain constant in the interval  $[t'_i, t''_i]$ . In practice, network routes are unlikely to remain stable over long intervals of time. If a change in intermediate nodes is necessary at time  $t''_i$ , we simply assume that the source closes the old data flow and opens a second one along the new route<sup>2</sup>. For each flow  $f_i$  we can define a hop length  $M_i$  and an ordered set of nodes  $\{N_i^1, N_i^2, \dots, N_i^{M_i}\}$ , with  $N_i^1$  being  $f_i$ 's source node,  $N_i^{M_i}$  being the destination, and each intermediate hop  $N_i^j$  selecting  $N_i^{j+1}$  as its next hop for  $f_i$ . Finally, each flow is characterized by a flow priority  $p_i \geq 1$ . The priority determines the urgency of the flow, with higher values corresponding to more urgent traffic. In the following section we describe a deterministic channel contention scheme based on the Black Burst protocol [20, 21] that is able to correctly prioritize real-time traffic over best effort traffic.

### 2.1. Background: Black Burst Contention Scheme

The Black Burst (BB) protocol was first proposed in [20] to achieve fair channel sharing in wireless LAN. Here, we are only interested in the BB channel contention scheme, and we adapt it to cope with a multi-hop network scenario. Whenever a node has a real-time packet ready for transmission, the packet is assigned a priority  $p$  based on the priority of the flow to which the packet belongs (we support up to 8 packet priorities in our implementation; see Section 4). For simplicity, in this section we assume that whenever two nodes contend with real-time packets on the wireless channel, the packets have different priorities. In Section 3.4 we detail how to cope with this limitation. Nodes with real-time packets access the channel using a medium interframe spacing of length  $t_{med}$ . Upon hearing idle channel for  $t_{med}$ , the real-time node starts transmitting a jamming signal, called a black burst, with length  $t_{BB}^p$  proportional to its packet priority  $p$ . After sending a BB, a node reverts to channel sensing for an interval  $t_{short}$  shorter than the medium interframe spacing  $t_{med}$ . If the channel remains idle for such interval, the node begins transmitting its packet, otherwise it reverts to channel sensing until the channel is sensed idle for at least  $t_{med}$  before transmitting another BB. In order to integrate real-time and best effort traffic, we propose a small modification to the IEEE 802.15.4 MAC. Instead of transmitting a packet immediately when the backoff timer expires, each node must first perform channel sensing for a long interframe spacing  $t_{long} > t_{med}$ ; if the channel is perceived busy, the node enters backoff again. This ensures that real-time packets as a group are given precedence over best

<sup>2</sup> More efficient solutions can be devised, but due to lack of space they are out of the scope of this paper.

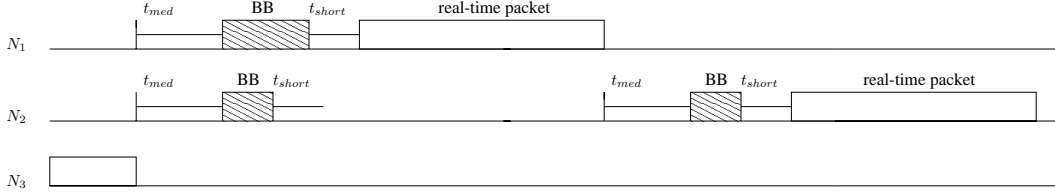


Figure 1: Example: BB contention

effort traffic. An example of BB contention is shown in Figure 1, where node  $N_1$  has the highest priority packet.

### 3. Protocol Design

In this section, we discuss the design choices behind the real-time chain protocol. We believe that in order to reliably support periodic real-time traffic in a wireless multi-hop environment, the MAC protocol must show several characteristics. First, it should avoid collisions of real-time packets, as they result in unacceptable waste of bandwidth and added unpredictability. Second, it must provide a mechanism to prioritize packet flows based on priority and with respect to best effort traffic. Third, it must support relatively high rates and soft real-time guarantees typical of real-time applications. The BB contention scheme provides a mechanism to prioritize the transmission of single packets, but further structure is needed to support flow communication. Furthermore, since our goal is to provide a practical and implementable protocol, our design choices must be grounded on physical realities.

**Design choices driven by experimental data:** We conducted a series of experiments with real hardware to evaluate the sources of packet collisions. In particular, we are interested in three quantities. The communication range  $R_C$  is the maximum range at which two nodes  $N_i$  and  $N_j$  can communicate reliably<sup>3</sup>. The sensing range  $R_S$  is the maximum range at which  $N_j$  can sense  $N_i$  transmitting. Finally, the interfering range  $R_I$  is the maximum range between nodes  $N_j, N_k$  such that packet transmissions with destination  $N_j$  that overlap in time with a transmission from  $N_k$  are lost. An important property depending on the defined ranges is the following:

**Definition 1 (Hidden node avoidance)** A network is said to conform to the hidden node avoidance property iff  $R_C + R_I \leq R_S$ .

It is easy to see that if the hidden node avoidance property holds any node  $N_k$  within interference range of  $N_j$  is able to sense any transmission from node  $N_i$  to  $N_j$ . Experiments were conducted on the Crossbow MICAz platform in a grass field, with nodes elevated about one meter from the ground, highest transceiver power level<sup>4</sup> and packet length

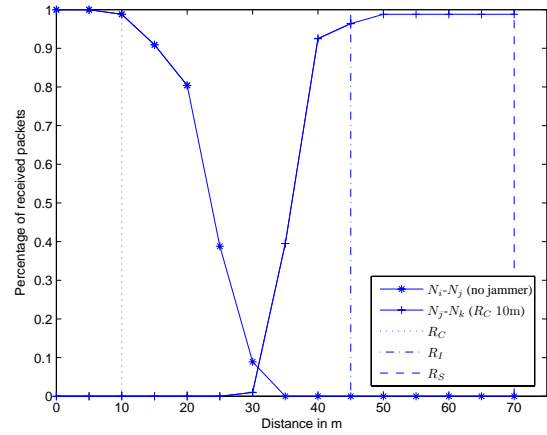


Figure 3: Ranges for MICAz motes

$n_i = 66$ . Results are reported in Figure 3. To determine the communication range, we changed the distance between transmitter  $N_i$  and receiver  $N_j$  and measured the percentage of correctly received packets. Based on the figure, we determined a range  $R_C = 10\text{m}$ . We then fixed the distance between  $N_i$  and  $N_j$  at  $R_C$  and activated a jammer node  $N_k$  at variable distances from the receiver. Again, based on the results we determined a range  $R_I = 45\text{m}$ . Finally, the sensing range was measured at  $R_S = 70\text{m}$ . In accordance with the experimental data, the hidden node problem did not represent a major issue when designing real-time chains; in addition, when we deployed intersecting real-time chains (see Section 5), we did not experience collisions due to hidden nodes<sup>5</sup>. Furthermore, we expect real-time chains to be robust against the hidden node problem even with less friendly communication, interference and sensing ranges due to the following argument: real-time chains do not interfere with non real-time traffic as they use dedicated channels. As such, the number of potential hidden nodes is expected to be very limited and only due to intersecting chains contending on the same channel. An interesting consequence is that if the hidden node avoidance property holds, it is in fact straightforward to see that the BB contention can avoid all collisions for real-time packets:

<sup>3</sup> Reliable communication is experimentally defined as 95% received packets

<sup>4</sup> Different power levels were tested but the ratios between  $R_C, R_I$  and  $R_S$  were very similar. The same holds for indoor experiments.

<sup>5</sup> An exhaustive analysis of the hidden node problem with real-time chains is out of the scope of this paper: as future research, we plan to use a simulator and to extensively test our scheme for different communication, interference and sensing ranges.

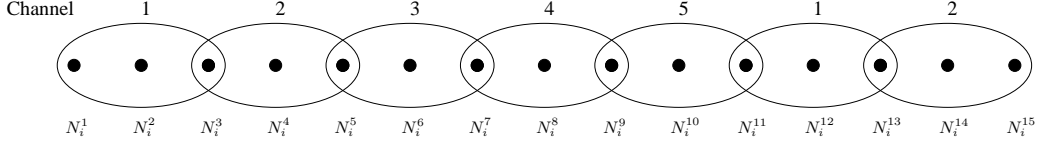


Figure 2: Flow cells and channels

**Property 1** *If the hidden node avoidance property holds, no two interfering real-time packets have the same priority, and furthermore  $t_{short} < t_{med} < t_{long}$ , no collision of real-time packets is possible under the BB contention scheme.*

Furthermore, notice that sensing and interference ranges are much larger than what commonly assumed, with the first being around 7 times larger than the reliable communication range. This means that at least for the category of nodes tested channel reuse is extremely low; since no more than one in every 8 intermediate nodes can simultaneously transmit in a contention based protocol, flow bandwidth can be no higher than  $B/8$ . Based on the above results, we claim that the use of a single shared wireless channel is not sufficient to provide high bandwidth for real-time wireless communication. Fortunately, transceivers of wireless embedded networks are typically able to transmit on several independent channels. In particular, the 802.15.4 compliant transceiver of MICAz motes can switch with minimum overhead among 16 non overlapping channels, numbered 0-15.

**Real-Time Chain:** We now introduce the key ideas of real-time chains. While no real-time flow is present in the network, all nodes are available for best effort traffic, delivered through CSMA/CA MAC on channel 0. Whenever a source node  $N_i^1$  starts flow  $f_i$  at time  $t'_i$ , it must first send a single *chain open* packet towards the destination. The chain open packet is transmitted using the BB scheme on channel 0, and is used to reserve the intermediate nodes for the chain serving flow  $f_i$ . Whenever an intermediate node receives a chain open packet, it switches to a set of channels  $[1, \dots, C]$  reserved for chain communication; in our implementation, we chose  $C = 5$ . When  $f_i$  ends at time  $t''_i$ , all intermediate nodes revert to servicing best effort traffic on channel 0. Hence, best effort traffic can not interfere with real-time traffic. Furthermore, by reserving multiple channels we can provide higher bandwidth to real-time flows. Again, while the protocol has been developed for the MICAz motes, it can be easily extended to other platforms with different range values by tuning the protocol parameters; in particular, the number of reserved channels  $C$  is a function of the sensing range  $R_S$  as detailed in Section 3.1. Notice that the choice of reserving the nodes used by an opened real-time chain is driven by the idea that a real-time chain is best used for carrying high bandwidth real-time traffic that requires most/all of the available resources of the relay embedded nodes. To carry very low bandwidth and time sensitive traffic would be best to use CSMA/CA (on shared channel 0) and differentiated traffic classes [4] (compatible with real-time chains). In the next Section we describe how chain

communication works.

### 3.1. Real-Time Chain Operation

After receiving the chain open packet for a flow  $f_i$ , nodes  $N_i^1, \dots, N_i^{M_i}$  switch to a channel in the set  $[1, \dots, C]$  and remain reserved for  $f_i$  communication until the end of the flow. We distinguish two types of nodes in the flow. Nodes with even indexes are *single channel* nodes: they receive and forward packets on the same channel. More precisely, each single channel node  $N_i^{2k}$  is associated with channel  $(k - 1) \bmod C + 1$ . Each node  $N_i^{2k+1}$  with odd index is a *dual channel* node: it receives packets on the channel of its predecessor node  $N_i^{2k}$  and it forwards packets on the channel of its successor node  $N_i^{2k+2}$ . The source node  $N_i^1$  is an exception, as it does not receive any packet; instead, packets are injected at the MAC layer with the desired flow frequency  $r_i$ . Similarly, the destination node  $N_i^{M_i}$  does not forward any received packet. An example of channel distribution for  $C = 5$  is reported in Figure 2, where we encompass with a oval the set of adjacent nodes that receive or transmit on a given channel; such nodes are said to form a *cell* (note that each dual channel node belongs to two cells at once). The value of  $C$  is strictly dependent on the sensing range. Since we want to enable maximum spatial reuse within a single chain, we must ensure that nodes belonging to different cells with the same channel are not in sensing range of each other. Since the minimum distance among any such nodes is equal to  $2C - 1$  hops, given a minimum hop distance of  $d$  we can choose  $C$  such that  $(2C - 1)d > R_S$  holds, from which we derive the condition:

$$C > \frac{R_S + d}{2d} \quad (1)$$

Both single and dual channel nodes transmit using the BB contention scheme. Given a priority  $p_i$  for the flow, single channel nodes use BB priority  $2p_i$  while dual channel nodes use BB priority  $2p_i - 1$ ; hence, a single channel node is always given precedence over a dual channel node in the same cell. Also note that given 8 different packet priorities, the number of different flow priorities is equal to 4. Furthermore, independently of the value of  $C$ , whenever two different chains are spatially close to each other it is always possible that two different cells of the two chains operating on the same channel interfere with each other. In this case, as long as the two flows have different priorities the BB contention scheme completely avoids collisions and prioritizes the packets of the higher priority flow over the packets of the lower priority one. An extension to the case when multiples interfering flows can have the same priority is dis-

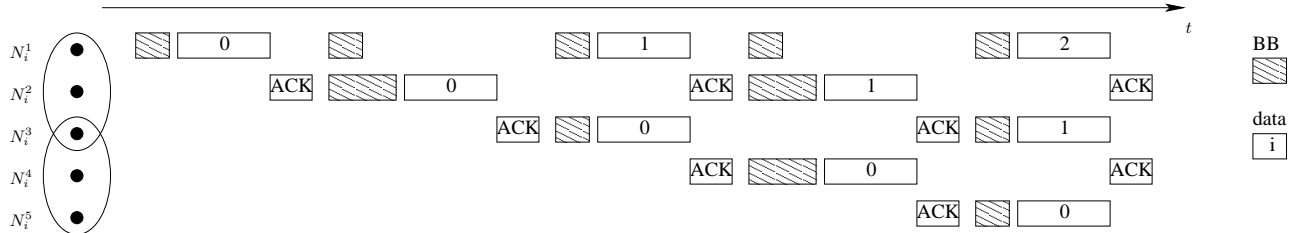


Figure 4: Example schedule: single flow

cussed in Section 3.4. Packet reliability is ensured by the receiver node sending back an ACK message immediately after the end of the data packet. Each time a packet is injected at the source, a progressive packet id is attached to the packet as the only MAC header information. We now detail the protocol rules for a dual channel node; single channel nodes behave in the same way, except for the fact that they never switch channel. Each intermediate node holds a single packet buffer in the MAC layer to store the last received packet and a counter to store the id of the last packet.

- Whenever the buffer is empty, the node listens on its reception channel and immediately acknowledges any packet sent to it. If the id of the received packet is greater than the counter, its value is updated and the packet is copied in the buffer.
- Upon copying a packet in the buffer, the node switches to its transmission channel and uses the BB contention scheme to transmit the packet. While the buffer is full, the node does not acknowledge any packet sent to it.
- If the node receives an ACK after winning a BB contention and sending the packet, it removes it from the buffer and switches back to listening. Otherwise, it contends again on the transmission channel until it correctly receives an ACK.

An example transmission schedule for a single flow is shown in Figure 4, where the desired rate  $r_i$  is high enough that the source node is never idle. Therefore, after  $N_1^1$  transmits packet 0 to  $N_1^2$ , it immediately starts another channel contention; however, since the BB priority of  $N_1^2$  is higher,  $N_1^2$  wins the contention and forwards packet 0 to node  $N_1^3$ , which is listening on channel 1.  $N_1^3$  then switches to channel 2 and transmits the packet. At the same time,  $N_1^1$  contends again on channel 1 and sends packet 1. The pattern repeats itself until each packet reaches the destination. The maximum packet rate sustainable by the flow, called  $\rho_i$ , can be easily computed based on the flow characteristics. Let  $t_{ACK}$  be the time to transmit an ACK,  $t_{pack}^i = n_i/B$  be the time to transmit a packet of  $f_i$ , and finally let us define  $t_{over}^i = 2(t_{med} + t_{short}) + t_{BB}^{2p_i} + t_{BB}^{2p_i-1} + 2t_{ACK}$  as the cell overhead. Then it can be seen from Figure 4 that the source transmits one packet every  $2t_{pack}^i + t_{over}^i$  seconds. Hence,  $\rho_i$  can be bounded as follows:

$$\rho_i \leq \rho_i^{\max} = \frac{1}{2t_{pack}^i + t_{over}^i} \text{ (packets/s)} \quad (2)$$

In particular, if  $t_{over}^i \ll t_{pack}^i$ , then the raw transmission bandwidth converges to a value of  $n_i \rho_i^{\max} = B/2$  which is optimal for packet forwarding. In practice,  $t_{over}^i$  is not negligible. Furthermore, in the current implementation each packet with BB priority  $p$  incurs in an additional processing overhead  $t_{proc}^p$ ; to take it into account, we can simply modify  $t_{over}^i$  by adding an additional implementation dependent term  $t_{proc}^{2p_i} + t_{proc}^{2p_i-1}$  (see Section 4). A good property of our approach is that queuing is done at the source node only. Hence, if a chain sustainable rate  $\rho_i$  decreases below the required rate  $r_i$ , the source is quickly notified as injected packets start to be queued for transmission. The MAC layer can then notify the application layer, thus giving the application a chance to react by dynamically adjusting the required rate.

### 3.2. Chain Opening

As detailed in Section 3.1, in order to operate a chain intermediate nodes must switch to a dedicated set of channels and remain there for the entire interval  $[t'_i, t''_i]$ . Hence, a suitable mechanism must be implemented to open a chain by reserving all intermediate nodes for chain communication. Since the chain must service real-time traffic, the opening phase must have bounded delay. This is achieved by means of a chain open packet which is sent by the source at time  $t'_i$  and forwarded by the routing protocol towards the destination. The chain open packet contains two fields: the flow priority, and a node id which is incremented at each hop. By using the flow priority and node id, each intermediate node can automatically compute its reception and transmission channel and its BB priority. The chain open packet is transmitted on channel 0, using the BB contention mechanism with BB priority equal to the flow priority. As detailed in Section 2.1, as long as no two chains with the same priority are opened simultaneously, no collision is possible for chain open packets and furthermore they receives precedence over both best effort packets and chain open packets with lower priority. We believe this assumption to be reasonable as chain opening should be infrequent with respect to the time it takes to complete the opening phase. Let  $t_{open}$  be the time that it takes to transmit the chain open packet; then the per-hop delay should ideally be:

$$t_{hop}^i = t_{med} + t_{BB}^{p_i} + t_{short} + t_{open} + t_{ACK} \quad (3)$$

Notice that this is not possible in a multi-hop environment. To understand why, consider a case where the chain open

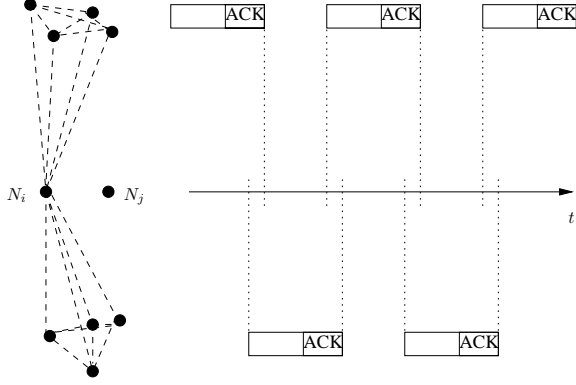


Figure 5: Starvation Problem

packet is forwarded from node  $N_i$  to node  $N_j$ , while another node  $N_k$  has a best effort packet ready to transmit, such that  $N_k$  is within sensing range of  $N_j$  but not  $N_i$ . Then while  $N_i$  is transmitting, the backoff timer on  $N_k$  can count down. After  $N_j$  receives the packet from  $N_i$ , it must sense the idle channel for  $t_{med}$  before transmitting. Now assume that the backoff timer expires and  $N_k$  starts transmitting immediately before  $t_{med}$  seconds from the reception at  $N_j$ . The overall result is that  $N_j$  will be blocked for as long as  $t_{med} + t_{max}$  seconds, where  $t_{max}$  is the maximum time to transmit a best effort packet (including the ACK). We refer to this situation, where a real-time packet is blocked by a lower priority or best effort packet, as a *priority inversion*; since a packet transmission can not be stopped once initiated, the described situation is effectively unavoidable.

Unfortunately, in a multi-hop environment priority inversion can even cause starvation of high priority packets. This situation is shown in Figure 5, where dotted lines are used to represent sensing between nodes. A node  $N_i$  is trying to send a real-time packet to node  $N_j$ , while two separate groups of nodes are engaged in best effort communication. The key here is that  $N_i$  is able to sense transmissions from both groups, while nodes in each group can not sense nodes in the other group. If we are unlucky enough that a transmission from one group always overlaps with idle time in the other group and viceversa as shown in the figure, then  $N_i$  can continuously perceive busy channel. In order to solve the starvation problem, we implement the following mechanism. If  $N_i$  perceives the channel busy for more than  $t_{max}$ , it jams channel 0 with a high-power jamming signal. The signal lasts for  $t_{max}$  and the power is high enough that it is reliably perceived by all nodes that  $N_i$  is able to sense. At the end of the jamming signal, all communication in the neighborhood of  $N_i$  has thus stopped. In the absence of higher priority packets,  $N_i$  wins the channel contention and transmits the packet. Therefore, we can guarantee that in spite of the starvation problem a chain open packet suffering no interference from higher priority packets traverses each hop in at most:

$$t_{hop}^i = 2t_{max} + 2t_{med} + t_{short} + t_{BB}^i + t_{open} + t_{ACK} \quad (4)$$

In practice, due to the nature of CSMA/CA random back-

off we expect that starvation is rare even with high bandwidth best effort traffic; hence, the average single hop delay is much shorter, as shown in Section 5. If the packet suffers interference from a chain open packet for a higher priority flow  $f_j$ , it incurs  $t_{hop}^j$  additional delay each time it loses a BB contention. In order to close the chain at time  $t''_i$ , the source simply sends a chain close packet on the reserved channel, which is again forwarded to the destination. After successfully transmitting the flow close packet, each node reverts to serving best effort traffic on channel 0.

### 3.3. Rate Analysis

In this section, we detail how rate bounds can be computed for a network with  $m$  simultaneously active flows  $f_1, \dots, f_m$ . Without loss of generality, assume  $p_1 < p_2 < \dots < p_m$ . Note that while in our model nodes are reserved for each chain, flows can still freely intersect thus creating interference with each other. In particular, we say that a set of neighboring nodes belonging to different flows constitutes an *interference point* if the nodes use the same channel and are in sensing range of each other. Nodes from any number of flows can belong to the same interference point, but as detailed in Section 3.1, we assume that no two such flows have the same priority. In order to determine the sustainable bandwidth for all flows, we start by analyzing the case of an interference point with two flows  $f_i, f_j, i > j$ . Ideally, if  $f_i$  rate approaches the maximum sustainable rate  $\rho_i^{\max}$ , due to interference  $f_j$  should be stopped from transmitting. However, this is not the case due to priority inversion. In particular, every dual channel node of  $f_i$  can suffer priority inversion from a packet of  $f_j$ ; however, single channel nodes of  $f_i$  cannot suffer priority inversion as they immediately contend on the same channel after receiving a packet. Hence, each packet of  $f_i$  can suffer in the worst case one priority inversion every two packet transmissions. In practice, due to processing and channel switching overhead if  $f_j$  rate is high enough the communication always converges to such worst-case packet schedule in which two packets of  $f_i$  followed by one packet of  $f_j$  are sent periodically in interfering cells. Following the discussion in Section 3.1 and Equation 2, we can therefore bound  $\rho_i$  as follows:

$$\rho_i \leq \frac{1}{2t_{pack}^i + t_{pack}^j + t_{over}^i + t_{over}^j/2} \quad (5)$$

Since  $f_j$  uses the bandwidth left free by  $f_i$ , the percentage of time it occupies the channel is at most  $1 - \rho_i/\rho_i^{\max}$ . Hence, the constraint on  $\rho_j$  is:

$$\rho_j \leq (1 - \rho_i/\rho_i^{\max})\rho_j^{\max} \quad (6)$$

Any other flow  $f_k, k < j$  in the interference point does not affect the schedule as it always loses contention from both  $f_i$  and  $f_j$ . Hence, the flow simply takes any additional bandwidth remaining from both  $f_i$  and  $f_j$ . Note that due

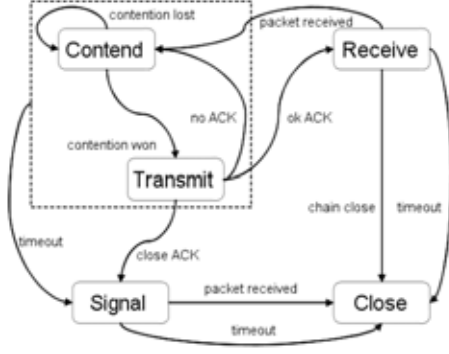


Figure 6: Protocol State Machine

to the chain design, if a flow belongs to multiple interference points, its bandwidth is effectively limited by the lowest sustainable rate among all points. Hence, we can easily compute the rates by building a set of linear constraints in  $\rho_1, \dots, \rho_m$  as follows:

1. for each active flow  $f_k, 1 \leq k \leq m$  in the network:

$$\rho_k \leq r_k \quad (7)$$

2. for each interference point, let  $I$  be the set of flows belonging to it, with  $f_i$  and  $f_j$  being the highest priority flows for that specific point as defined above. Add Equation 5 for the highest priority flow  $f_i$  to the constraint set, and furthermore for each other flow  $f_k \in I, k \neq i$ :

$$\rho_k \leq (1 - \sum_{l \in I \wedge l > k} \rho_l / \rho_l^{\max}) \rho_k^{\max} \quad (8)$$

It is then possible to solve the system by individually maximizing each flow rate starting from the highest priority  $\rho_m$  to the lowest priority  $\rho_1$  subject to all constraints.

### 3.4. Extensions

While our currently implemented protocol exhibits good properties in term of guaranteed bandwidth and delay, it also suffers from some limitations. First of all, collisions can not be avoided if two flows with the same priority interfere with each other. Consider two flows  $f_i, f_j$  with the same priority, and assume that either two single channel nodes or two double channel nodes  $N_i^k, N_j^l$  are in interference range of each other; note that a single channel node can never interfere with a dual channel node since they use different BB priorities. Then whenever  $N_i^k$  and  $N_j^l$  try to access the channel simultaneously, they can both win the contention and therefore transmit at the same time causing interference. Since each node continue to contend until it receives an ACK, both chains could be blocked. We now discuss a possible extension to our protocol in order to cope with this problem. Instead of using a single channel set for all flows, we can use multiple sets  $[1, \dots, C], [C +$

$1, \dots, 2C], [2C + 1, \dots, 3C], \dots$ . Using our MICAz implementation with  $C = 5$  as an example, we can support up to three channel sets:  $[1, \dots, 5], [6, \dots, 10], [11, \dots, 15]$ . Whenever a chain is blocked from transmission for a significant amount of time, we close it and reopen it at the source with a different channel set, either selected through some form of distributed agreement or simply at random<sup>6</sup>. A simplified state machine for an intermediate node is shown in Figure 6. While the chain is working correctly, each node transitions among the states<sup>7</sup> *receive*, *contend* and *transmit* as detailed in Section 3.1. Upon receiving a chain close packet, the node transitions to the *close* state and reverts to channel 0. The node also transitions to *close* if it does not receive any packet for timeout seconds. Furthermore, upon receiving a packet and transitioning to the *contend* state the node also starts a timer. If the node can not transmit the packet correctly within timeout seconds, it assumes that the chain has failed and transitions to *signal*. In this state, the node reverts to channel listening and acknowledges any received packet with a special *close ACK*, then transitions to *close*; upon receiving a close ACK, the sender immediately transitions from *transmit* to *signal*. In this way, the failure information can be quickly propagated towards the source; a single bit is sufficient to distinguish normal ACKs from close ACKs. Finally, since multiple nodes in the chain can timeout in the *contend* state, a third timeout must be used to transition from the *signal* to the *close* state should the node not receive any packet. Note that while this extension has been developed to cope with the problem of interfering flows with the same priority, it is in fact useful to relieve all situations in which a chain is blocked, including the least priority flow in a three flow interference point not receiving any bandwidth or violations of the hidden node avoidance property. A second possible limitation is that nodes can not simultaneously belong to more than one chain. If multiple flows are sent at the same time from different sources toward a single wired base station, several solutions are possible to overcome the limitation. First, we can simply provide the base station with multiple transceivers, in number equal to the maximum amount of simultaneously active flows. Second, we can reserve a separate channel for the base station; all chain nodes transmitting to it switch and contend with each other on the base station's reserved channel. A mixed approach is also possible, by reserving one separate channel for each transceiver on the base station. Each incoming chain then transmits on one of the reserved channels, either chosen at random or through some form of local arbitration by the base station.

## 4. Implementation

This section describes our implementation of the proposed protocol on MICAz motes. This state-of-the-art sen-

<sup>6</sup> It is possible to implement a more efficient but complex solution by reopening the chain from the blocked node instead of going back to the source. We do not detail it due to space limitations

<sup>7</sup> *Transmit* and *receive* states include the exchanging of an ack.

Unit	$t_{med}$	$t_{short}$	$t_{long}$	$t_{slot}$	$t_{ACK}$	$t_{extra}$
ms	0.64	0.32	0.96	0.32	0.544	0.32
bytes	20	10	30	10	17	10

Table 1: Implementation parameters

$p$	1	2	3	4	5	6	7	8
$t_{proc}^p$ (ms)	1.6	2	2.2	2.4	2.7	3	3.1	3.4

Table 2:  $t_{proc}^p$  values for different packet priorities  $p$ .

sor node platform uses Chipcon CC2420 RF transceiver which conforms to IEEE 802.15.4 standard. The radio chip provides a raw bandwidth of 250kbps and is especially tailored to the standard with hardware automatic acknowledgement and address filter. The BB MAC protocol was implemented in MICAz TinyOS 1.1.7. To reduce the protocol overhead, we make use of hardware-supported facilities of CC2420. In particular, a black burst is sent as a packet with a packet type value different from standard 802.15.4 supported values. Since CC2420 silently drops packets with unsupported packet types, a listening mote perceives a BB packet as a burst of noise without receiving it, thus reducing processing overhead. Details of implementation parameters are shown in Table 1. Due to limited sensing capabilities of the transceiver, we actually require all BBs to be at least as long as  $t_{med}$ ; hence, for a packet with priority  $p$  we use a BB length  $t_{BB}^p = pt_{slot} + t_{extra}$ . Since we are constrained by the maximal CC2420 packet size of 128 bytes, this means that we are only able to provide 8 priority levels in the current implementation. Note that the maximal BB packet size is comparable to the size of the CSMA backoff window implemented in MICAz TinyOS 1.1.17, which varies from 10 to 160 bytes time. It is worth noting that, since CC2420 is specially designed to support IEEE 802.15.4 standard with a CSMA/CA MAC, it does not accommodate BB transmissions well. A more BB-friendly radio chip would help to reduce BB overhead. Providing a good estimation of  $t_{over}^i$  is more complex. As detailed in Section 3.1, in practice packet transmission rate is limited by additional processing overhead in both the hardware and software components. We model it with an overhead term  $t_{proc}^p$  for a BB packet with priority  $p$ . Since computing the value of  $t_{proc}^p$  analytically is hard, we evaluated it experimentally. To this end, we determined the maximum single-hop packet transmission rate for each priority  $p \in [1, \dots, 8]$  by simply letting a transmitter node send packets as fast as possible to a base station without any interference. Since for priority  $p$  the single-hop packet rate is effectively equal to  $1/(t_{med} + t_{BB}^p + t_{short} + t_{pack} + t_{ACK} + t_{proc}^p)$ , we can easily compute the processing overhead term. Table 2 shows the obtained values for different packet priorities. Finally, Table 3 shows the final computed values for  $t_{over}^i$  for chain priority  $p_i \in [1, \dots, 4]$ .

$p_i$	1	2	3	4
$t_{over}^i$ (ms)	8.2	10.5	12.9	15.0

Table 3:  $t_{over}^i$  values for different flow priorities  $p_i$ .

## 5. Experimental Results

To measure the performance of the real-time chain protocol and to validate our proposed bounds, we conducted several experiments for multi-hop single chain communication, multiple chains communication and for the chain opening procedure. For each experiment we provide both measured and predicted results (based on the  $t_{over}^i$  values of Section 4) for flows with packet length  $n_i = 66$  bytes. Results are reported both in term of sustainable packet rate  $\rho_i$  and MAC level bandwidth. Note that since the physical layer adds a 6 bytes frame header, the effective packet length at the MAC layer is  $n_i - 6 = 60$  bytes. Hence, MAC bandwidth is simply computed as  $60\rho_i$  bytes/sec.

**Single chain performance:** We measured the maximal packet rate and the per-hop delay of the real-time chain protocol using indoor chains with  $M_i = 10$  hops,  $C = 5$  channels and different priorities. We set the desired packet rate  $r_i$  high enough that the source is always busy, and the number of packets received at the destination node was used to calculate the packet rate. To provide a comparison with existing MAC protocols, we also show the measured packet rate for the TinyOS CSMA/CA MAC. However, a comparison with the standard protocol would not be fair, as we use more channels. Hence, for this experiment we modified the TinyOS protocol to use multiple channels similarly to real-time chain. Results are shown in Figure 7 and Table 4. As analyzed in Section 3.1, when the source rate is high enough, thanks to the deterministic nature of the protocol a node in a chain should be either receiving or transmitting a packet at all times. In fact, the measured packet rate is around half that of single-hop communication, much higher than what achievable with single channel communication. Furthermore, BB chain almost achieves the predicted upper bound rate  $\rho_i^{max}$ , while the measured per-hop delay is only slightly higher than the predicted value of  $2t_{pack}^i + t_{over}^i$ . On the contrary, due to the random nature of the backoff timer, a 10-hop CSMA/CA chain sustains a bandwidth of only 24.6 kbps, which is actually 2.9 times lower than the measured single-hop bandwidth of 71 kbps. In order to determine the optimal value for the number of real-time channels  $C$ , we conducted outdoor experiments with a 20 hops chain with  $p_i = 1$ . The bandwidth of the chain was measured while changing the numbers of channels. We found that with  $C = 5$  channels we are able to achieve a bandwidth equivalent to the one shown in Figure 7; note that with this value, each channel is shared by two cells in the chain. The value is consistent with Equation 1 and the experimental values of Figure 3.

**Multiple chains performance:** To validate the analysis of Section 3.3, we measured the packet rates of two chains  $f_1, f_2$  each having ten hops with  $C = 5$ . The two chains were assigned priorities  $p_1 = 1$  and  $p_2 = 2$  and were placed

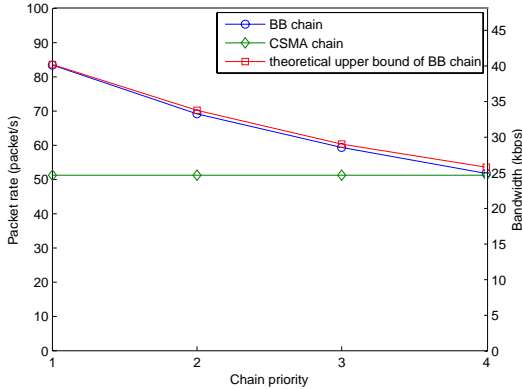


Figure 7: Bandwidth for a single chain

$p_i$	1	2	3	4
Measured per-hop delay (ms)	13.8	15.6	17.8	19.6
$2t_{pack}^t + t_{over}^t$ (ms)	12.4	14.7	17.1	19.2

Table 4: Per-hop delay for a single chain

in parallel<sup>8</sup> in the interfering range of each other. The desired rate  $r_1$  of the lowest priority chain was set higher than its sustainable rate while the desired rate  $r_2$  of the highest priority chain was varied. The measured sustainable rates  $\rho_1$  and  $\rho_2$  for the two chains are plotted in Figure 8 together with the predicted bounds from Equations 5, 6. The first observation is that the sustainable packet rate of the highest priority chain is roughly one third lower than its maximal rate shown in Figure 7. As detailed in Section 3.3, this reduction is due to ineluctable priority inversion of the lowest priority chain over the highest one. However, the obtained measurements follow closely the values predicted by Equations 5, 6, 7. Along with the results on single chain performance, this shows that our theoretical analysis with the adjustment of  $t_{proc}^p$  is able to accurately predict the operation of real-time chain.

**Chain opening performance:** We evaluated the time needed for chain opening on a chain with 10 hops and priority  $p_i = 1$  indoor. Eight further nodes were placed along the chain to simulate interference from best effort traffic; all such nodes tried to transmit at their maximum possible rate. In order to obtain precise timing information, each chain open packet was effectively sent from the source node to the destination node and then back to the source; the per-hop delay for that specific packet was then computed by measuring the overall delay at the source node and dividing it by the number of hops. We repeated the experiment for 300 packets, and computed average and maximum per-hop delay. Using Equation 3 modified to account for the processing overhead  $t_{proc}^p$ , we can compute a lower bound on the per-hop delay of 5.86ms. Similarly, from Equation 4 the up-

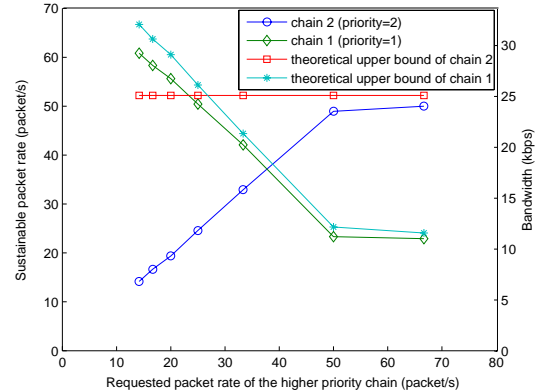


Figure 8: Bandwidth of two crossing chains

per bound on the per-hop delay is 11.81ms. The measured average per-hop delay is equal to 6.11ms, which is rather close to the lower bound. This shows that due to the effect of random backoffs, priority inversions are actually rare in the tested scenario. However, the measured maximum per-hop delay is significantly higher at 9.44ms. Hence, in rare cases the chain open packet can indeed be slowed down by best-effort traffic. Still, the measured maximum is well within the bound provided by Equation 4.

## 6. Related Work

MAC protocols have been widely studied in wireless networks. In what follows, we focus on MAC protocols that provide real-time or QoS traffic support. We can broadly categorize them into two classes [10]: *Synchronous schemes* that require global time synchronization among different nodes, and *asynchronous schemes* which do not require such support. Synchronous schemes include Implicit-EDF [2], TBMAC [3], Cluster TDMA [12], Cluster Token [11] and MAX [15]. All these protocols work by having nodes implicitly agree on a transmission slot assignment. The agreement requires both time synchronization and either a regular network structure or some form of clustering. In [16, 19] a TDMA based scheme is introduced that is able to transmit full-duplex, compressed data voice. The authors presented an efficient power-aware scheme that requires a global routing tree and network schedule: it can support one communication path at a time and it requires an infrastructure of fixed nodes. Compared to it, the approach of real-time chains requires only local knowledge for routing and does not rely on a global schedule for medium access. IEEE 802.11 Distributed Coordination Function (DCF) is a widely used, standard asynchronous protocol, but its ability to provide consistent bandwidth to differentiated multi-hop traffic is poor [8]. Several schemes have been developed to improve over DCF by either providing support for differentiated traffic categories [1, 4, 14, 23] or by reducing collisions in multi-hop networks [13]. Similarly, in [6] a routing protocol is introduced to pro-

<sup>8</sup> Note that real-time chains can be intersected and deployed in a 2D or 3D configuration.

vide soft real-time delivery service by predicting delay at each hop. Though these schemes are easy to implement over the IEEE 802.11 protocol, they are limited by the randomness of the medium access protocol. Several more deterministic protocols based on Black Burst [20, 21] have been proposed to support real-time traffic [18, 7]. However, all such protocols except BTPS [24] assume a fully connected network, and are not easily extensible to multi-hop environment. BTPS provides the ability to differentiate two classes of packets in a multi-hop environment using Black Burst for channel jamming, but it supports only one priority and it relies on random back-off to establish a connection. Out-of-band signaling has been used for channel reservations in multi-hop networks in different works [22, 17]. Although these protocols use multiple wireless channels like real-time chains, the key difference is that they need the transceiver to be able to simultaneously listen on two different channels at once, which is not practical in resource constrained embedded nodes.

## 7. Conclusions and Future Work

We have introduced real-time chain, a new prioritized MAC protocol for real-time data flow delivery over ad-hoc multi-hop wireless networks. Collision-free channel contention and per-packet prioritization are achieved through a modified version of the Black Burst contention scheme. Based on experimental data, we designed a transmission scheme over multiple channels in order to enhance spatial reuse and provide high bandwidth for real-time communication. Finally, delay and bandwidth soft guarantees have been proposed and validated through an implementation based on MICAz motes. As future work, we plan to investigate how to integrate real-time chains with adaptive transmission power control mechanisms to enhance communication reliability in spite of environmental changes.

## References

- [1] R. O. Baldwin, I. V. Nathaniel, J. Davis, and S. F. Midkiff. A real-time medium access control protocol for ad hoc wireless local area networks. *SIGMOBILE Mobile Computer Communication Review*, 3(2):20–27, 1999.
- [2] M. Caccamo, L. Zhang, L. Sha, and G. Buttazzo. An implicit prioritized access protocol for wireless sensor networks. In *IEEE RTSS*, December 2002.
- [3] R. Cunningham and V. Cahill. Time bounded medium access control for ad hoc networks. *Principles of Mobile Computing*, 2002.
- [4] J. Deng and R. S. Chang. A priority scheme for IEEE 802.11 DCF access method. *IEICE Transactions on Communications*, E82-B(1):96–102, 1999.
- [5] M. Rahimi et al. Cyclops: in situ image sensing and interpretation in wireless sensor networks. In *ACM SenSys*, November 2005.
- [6] T. He, J. Stankovic, C. Lu, and T. Abdelzaher. Speed: A stateless protocol for real-time communication in sensor networks. In *International Conference on Distributed Computing Systems*, Providence, Rhode Island, 2003.
- [7] S. Jang-Ping, L. Chi-Hsun, W. Shih-Lin, and T. Yu-Chee. A priority MAC protocol to support real-time traffic in ad hoc networks. *Wireless Networking*, 10(1):61–69, 2004.
- [8] JinyangLi and et al. Capacity of ad hoc wireless networks. In *ACM MobiCom*, 2001.
- [9] B. Karp. *Geographic Routing for Wireless Networks*. PhD thesis, Harvard University, 2000.
- [10] S. Kumar, V.S. Raghavan, and J. Deng. Medium access control protocols for ad-hoc wireless networks: a survey. *Elsevier Ad-Hoc Networks Journal*, To appear.
- [11] C. H. Lin. *A multihop adaptive mobile multimedia network: architecture and protocols*. PhD thesis, University of California at Los Angeles, 1996.
- [12] C. R. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. *IEEE Journal of Selected Areas in Communications*, 15(7):1265–1275, 1997.
- [13] C. R. Lin and M. Gerla. Real-time support in multihop wireless networks. *Wireless Networks*, 5(2):125–135, 1999.
- [14] C. Lu, B. Blum, T. Abdelzaher, J. Stankovic, and T. He. Rap: a real-time communication architecture for large-scale wireless sensor network. In *Proc. of the 8<sup>th</sup> RTAS*, San Jose, California, September 2002.
- [15] R. Mangharam and R. Rajkumar. Max: a maximal transmission concurrency MAC for wireless networks with regular structure. In *IEEE Broadnets*, October 2006.
- [16] R. Mangharam, A. Rowe, R. Suzuki, and R. Rajkumar. Voice over sensor networks. In *IEEE RTSS*, December 2006.
- [17] J. P. Monks, V. Bharghavan, and W. Hwu. A power controlled multiple access protocol for wireless packet networks. In *Proceedings of IEEE Infocom*, 2001.
- [18] A. Pal, A. Dogan, and F. Ozguner. MAC layer protocols for real-time traffic in ad-hoc wireless networks. In *ICPP*, 2002.
- [19] A. Rowe, R. Mangharam, and R. Rajkumar. Rt-link: A time-synchronized link protocol for energy constrained multi-hop wireless networks. In *IEEE SECON*, September 2006.
- [20] J. Sobrinho and A. Krishnakumar. Real-time traffic over the IEEE 802.11 medium access control layer. *Bell Labs Technical Journal*, 1(2):172–187, 1996.
- [21] J. Sobrinho and A. Krishnakumar. Quality-of-service in ad hoc carrier sense multiple access networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1353–1368, August 1999.
- [22] 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*, 23(12), 1975.
- [23] X. Yang and R. Kravets. Distributed qos guarantees for real-time traffic in ad hoc networks. In *Proc. of the 1<sup>st</sup> IEEE Secon*, 2004.
- [24] X. Yang and N. H. Vaidya. Priority scheduling in wireless ad hoc networks. In *Proc. of the 3<sup>rd</sup> ACM MobiHoc*, 2002.