Proposition of a Hard Real-Time MAC Protocol for Wireless Sensor Networks

Thomas Watteyne, Isabelle Augé-Blum
CITI Laboratory, INSA Lyon, France
{thomas.watteyne, isabelle.auge-blum}@insa-lyon.fr

Abstract

Many wireless sensor network applications are emerging nowadays. For critical, safety related applications, the network needs to provide bounded transmission delays. Hard real-time guarantees need therefore to be given by wireless sensor network communication protocols. In this paper, we propose a new hard real-time MAC protocol, and we give the time constraints that can be reached.

1. Introduction

Monitoring or event detection applications on a wide range of areas contribute to a general growing interest in wireless sensor networks (WSNs) [1]. We consider static sensors that report their sensed data to a sink or gateway, an intelligent node capable of taking the appropriate actions.

WSNs can be regarded as a sub-category of ad-hoc networks. Nodes cooperatively organize themselves into a communicating radio network, adapting to changes in the environment such as loss of connectivity. Communication among nodes can be done in a multi-hop way, each node being a router for other nodes' communications.

The low performance nodes (in terms of processing power, available energy, memory and bandwidth, mainly) have to cope with applications' demanding constraints, such as network lifetime, scalability, fault tolerance or environmental constraints. Due to the applications' nature, timeliness appears as a "natural" constraint and the network should give a guaranteed worst case transmission time [8].

To free ourselves from routing considerations, we assume a linear network (a 2-D extension being future work) and propose a new hard real-time MAC protocol. Possible applications include highway car accident monitoring or production line surveillance.

Related work is summarized in Section 2; Hypotheses are presented in Section 3. The proposed protocol is detailed in Section 4; Analysis and validation is done in Section 5. Section 6 concludes this paper and presents issues for further research.

2. Related work

Real-time has two definitions: soft and hard real-time. Whereas the first type focuses on mean times and tries to reduce deadline miss ratio using flow differentiation, hard real-time systems only take worst case times into account (missing a deadline not being an option) and offer bounds to those times.

In [6], presenting a soft real-time communication architecture called RAP, each message is assigned a velocity it has to keep in order to reach its destination on time, and message scheduling inside the network is done accordingly (velocity is mapped to a MAC-layer priority). Packets can be discarded if the required velocity is too high.

SPEED [4], a soft real-time stateless and localized routing protocol, maintains a desired delivery speed across the network using feedback control and non-deterministic geographic forwarding.

To our knowledge, only [2] presents a communication protocol (MAC) giving hard real-time guarantees. As depicted in Figure 1, nodes must be organized in hexagonal cells. Intra-cellular and inter-cellular communications are differentiated. Each cell is assigned a frequency; using 7 different frequencies, interference between neighboring intra-cellular communications is avoided. By running the Earliest Deadline First algorithm [2] at each node, each one constructs the same scheduling table which regulates medium access among nodes of the same cell to avoid intra-cellular communication interference. A router node is placed at the center of each cell for inter-cellular communication. The six directions are noted A, B, etc, and by emitting at the receiver cell's frequency, directional emission is possible. Using a global time slotted frame (shown at the bottom of Figure 1) intra-cellular and inter-cellular communication in a given direction alternate.

We argue that the assumptions are hard to meet. Indeed, rigid hexagonal shaped cell structure seems only poorly compatible with random deployment and nodes are costly (router nodes have two transceivers; all nodes are GPS-enabled for global synchronization).

Proceedings of the 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS'05)
1526-7539/05 $20.00 © 2005 **IEEE**

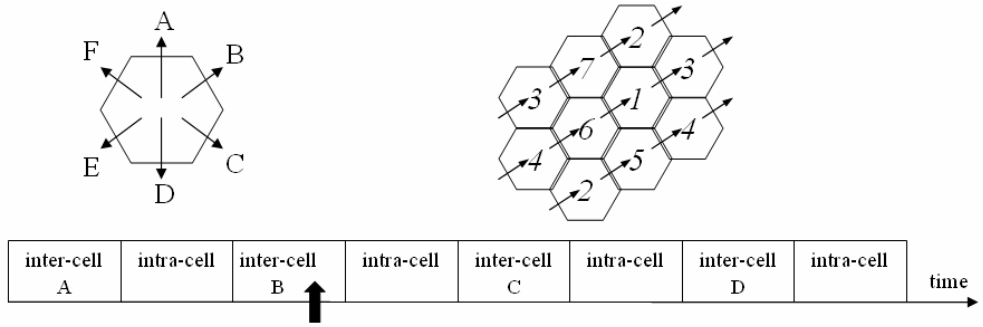

Figure 1. Inter-cellular communication in I-EDF [2]

3. Hypothesis

We focus our work on proposing a hard real-time MAC protocol for a network of low-cost sensors (e.g. only one frequency), deployed randomly, with no differentiated nodes (e.g. no router nodes), and without synchronization on a global clock (e.g. no GPS). To free ourselves from routing considerations, in the supported application class the covered area is linear (any node's transmission reaches both borders). A sink is placed at one network end to collect the alarms.

We assume that each node knows its x-coordinate, uniquely identifying it. This position can be obtained during deployment. We assume that neighboring nodes are separated by a geographic distance between $dist_{min}$ and $dist_{max}$ (neighboring nodes can communicate).

The radio link is supposed to be bi-directional, and when a node is placed between two potentially communicating nodes, it can communicate with either one of them. Nodes communicate using the constant bandwidth BW (in bps); the message lengths in bits are noted $length_{<message\ type>}$. Each node knows a priori max_{range} (in meters), the maximum range of an emitting node, in optimal conditions. Interference range is assumed never to exceed 1.5 times communication range (assumption loosened in 4.2.2).

Alarms are considered aperiodic; they can be generated by any node, and are all equally important.

4. The proposed MAC protocol

The protocol consists of an initialization phase, followed by a run-time phase, subdivided in two modes: unprotected and protected mode.

Each node can only access the medium if it has waited for a backoff time proportional to its distance to the last emitting node, and it has not heard any other message during this waiting time. With nodes at least separated by $dist_{min}$, collisions are avoided. We introduce the novel concept of virtual "waves" of expiring backoff timers. Each such wave has a predetermined limited speed noted $W_{<type\ of\ wave>}$.

4.1. Initialization of the sensor network

Initialization phase creates half-cells of strongly connected nodes: each node of a half-cell can reliably communicate with any node of a neighboring half-cell. Cells are numbered in ascending order away from the sink. Two messages are used: *creation* and *ack*.

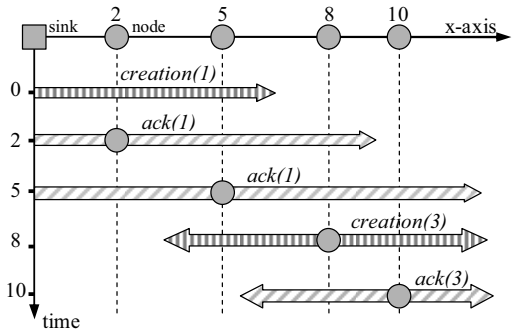

Figure 2. Cell creation

(1) The sink creates cell 1 with *creation(1)*.
(2) Nodes that hear this message set their timers according to their distance to the sender.
(3) Each node sends out an *ack* message when its timer elapses, potentially heard by nodes further away.
(4) A node which heard only *ack* messages will send out a new *creation* message (incrementing the cell number by 2 -cells will be cut in half later on).

The maximal acceptable Bit Error Rate (BER_{max}) is translated into a minimal acceptable Signal to Noise Ratio (SNR_{min}) (details can be found in [7]). To achieve reliability, a creation message is considered heard only if received with $SNR \geq SNR_{min}$ (a node cannot directly sense a message's BER).

Each node is identified by the 3-tuple *[*cell number *I*, relative position in the cell (in percentage) *R,* absolute position *A]*. To meet full connectivity between neighboring half-cells, each cell is split at relative position 50% (each node recalculates its identifier).

Proceedings of the 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS'05)
1526-7539/05 $20.00 © 2005 **IEEE**

The network is synchronized on the periodical passing of synchronization waves. Waves are equally separated by 6 half-cells to avoid interference between two simultaneous emissions (as in Section 3, a node's interference range equals 3 half-cells). All waves run at the same speed $W_{synchronization}$, in percent of a half-cell per second. In protected mode, a node emits when its timer elapses.

(1) The sink starts the process by sending out *sync*.
(2) All new nodes hearing this message set their timers.
(3) The first node of each half-cell uses the timer elapsing instant to re-emit *sync*. Go to step (2).
(4) All nodes use the timer elapsing instant to start a periodic timer. Its period is set to the time needed by the wave to run through 6 half-cells.

4.2. Run-time

A possibly long alarm (due to alarm aggregation) needs to reach the sink within a bounded time. Two modes are used: unprotected and protected mode. Unprotected mode is used when collision probability is low (few alarms), multi-hop propagation speed is near-optimal. When a collision occurs, the network switches to a slower but collision-free protected mode.

4.2.1. Unprotected mode

This mode is used initially; it provides a near-optimal transmission speed but is collision-prone.

(1) A node sends out an alarm message.
(2) All nodes hearing this message set their timers, based on the difference between their absolute position and the furthest position the message can reach (calculated using max_{range}).
(3) The first node which timer elapses is elected relaying node, and immediately resends the alarm.

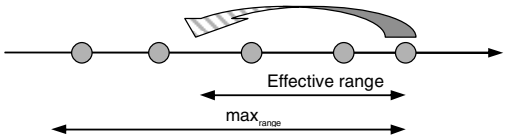

Figure 3. Unprotected mode

When a node does not hear its alarm relayed after a given time, it sends out a jam message that floods the network to switch to protected mode.

4.2.2. Protected mode

Before an alarm is sent, a portion of the network where no new alarms can be generated is reserved using signaling messages. This portion is long enough for two simultaneously emitted alarms to be separated by enough distance not to collide. For signaling messages not to collide, wave synchronization is used. To provide reliability, signaling messages are transmitted between neighboring half-cells (Figure 4).

(1) A node at half-cell *j* willing to emit sends out a *silence(1)* message.
(2) All nodes of half-cell *j-1* (closer to the sink) put themselves in a *reserved(1)* state.
(3) A message *silence(2)* is sent by a node at half cell *j-1* to reserve all nodes of half cell *j-2*. The relaying node is elected using backoff timers.
(4) Multi-hop reservation goes on until half-cell *j-5* is put into *reserved(5)* state. This cell then sends back an *ack_exp* message.
(5) The *ack_exp* message travels in unprotected mode until it reaches the initiating node.

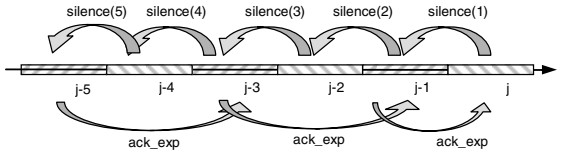

Figure 4. Reservation mechanism

After protection, the alarm is transmitted as in unprotected mode. The elected relaying node will send out a *silence(1)* message, launching a new protection phase. Alternation between protection and transmission phases provides the alarm's multi-hop transmission.

For the mean transmission speed to be as high as possible, the network needs to switch back to unprotected mode. The sink decides when to perform this switching back based on the rate of alarms received -as receiving few alarms heightens the probability of having a less occupied network- and sends out a jamming signal.

Remark: Augmenting the number of reserved half-cells and the distance between synchronization waves lets the protocol adapt to other interference ranges.

5. Protocol analysis and validation

5.1. Protocol Analysis

In this part, we present the Worst Case Execution and Transmission Times (*WCET*/*WCTT*) of the different parts of the proposed protocol, obtained by analytical analysis. It is essential to have bounded worst case times for all parts, for the system to meet hard real-time constraints.

$WCET_{initialization}$ is given by (1); $network_{length}$ refers to the length of the network, expressed in meters. The summed up terms refer to the duration of the passage of the cell-creation and the synchronization waves, respectively. For each of these terms, we have considered the worst case scenario; in particular the case where each node sends out a creation message. Re-initializing the network at run-time in a time- or event-driven manner without affecting the hard-real time characteristic of the protocol is possible.

$$WCET_{initialization} = \frac{network_{length} + \max_{range}}{W_{cell\ formation}} + \frac{number_{cells} \times 100}{W_{synchronization}} \quad (1)$$

As for the run-time phase, we present the $WCTT_{unprotected}$ and $WCTT_{protected}$ where one alarm is present on the network.

$WCTT_{unprotected}$ (2) is found when each node can only communicate with its two neighbors (all nodes will need to relay the alarm), and those nodes are extremely close to each other (maximizing the furthest reached node election time).

$$WCTT_{unprotected} = number_{nodes} \times \left[\frac{length_{data}}{BW} + \frac{\max_{range}}{W_{emission}} \right] \quad (2)$$

$WCTT_{protected}$ (3) is found when all half-cells only have one node; all nodes will then need to relay the alarm. Synchronization waves are separated by 6 half-cells, thus by 600%. Signaling messages are sent each synchronization wave period, leading to 6 wave periods until a relaying node is elected at half-cell *reserved(5)*. In the worst case, the *ack_exp* message will need to be relayed at each half-cell (sent 5 times, with 4 furthest reached node elections). Finally, the alarm of length $length_{alarm}$ is sent using the bandwidth BW and a furthest reached relaying node is elected.

$$WCTT_{protected} = number_{nodes} \times \left[\frac{6 \times 600}{W_{synchronization}} + \left(5 \times \frac{length_{ack\_exp}}{BW} + 4 \times \frac{\max range}{W_{emission}} \right) + \left(\frac{length_{alarm}}{BW} + \frac{\max_{range}}{W_{emission}} \right) \right] \quad (3)$$

$WCET_{switch}$ (4) is obtained when each node can only reach its two neighbors. This WCET is the same for either switching to or from to unprotected mode.

$$WCET_{switch} = number_{nodes} \times \frac{length_{jam}}{BW} \quad (4)$$

5.2. Modeling and validation

A behavioral validation of the proposed protocol is needed, followed by a formal validation of its timeliness characteristics. For modeling purposes, we needed a modeling language capable of expressing distributed system issues such as concurrence or message exchange, together with temporal constraints. It had to be possible to use the created model for formal behavioral and timeliness validation. UPPAAL [5] is an integrated tool environment for modeling and model-checking real-time protocols. [3] justifies its use for modeling and analysis of hard real-time networks.

Using UPPAAL, we have validated the behavior of our protocol, and have formally confirmed the WCET and WCTT presented in the previous section. Due to space limitations, a detailed explanation will not be given here.

6. Conclusion and future work

Throughout this paper, we have presented a novel MAC protocol for linear wireless sensor networks with realistic assumptions. It supports hard real-time constraints and provides WCET/WCTT guarantees.

We are currently working on simulating this protocol in order to compare its mean time performances with existing non-real time protocols.

Future work include studying and adding fault tolerance mechanisms to our protocol (message and node loss), determining the protocol's real-time capacity in terms of number of alarms per time unit and per half-cell, studying its scalability, and adding a hard-real time routing layer in order to extend the protocol to two dimensions.

7. References

[1] I. F. Akyildiz and I. H. Kasimoglu. Wireless sensor and actor networks: research challenges. 1st IEEE International Conference in Mobile Ad hoc and sensor systems (MASS'04), December 2004.

[2] M. Caccamo, L. Y. Zhang, L. Sha, and G. Buttazzo. An implicit prioritized access protocol for wireless sensor networks. IEEE Real-Time System Symposium (RTSS'02), December 2002.

[3] K. Godary, I. Augé-Blum, and A. Mignotte. SDL and Timed Petri Nets versus UPPAAL for the validation of embedded architecture in automotive. Forum on specification and Design Language (FDL'04), September 2004.

[4] T. He, J. A. Stankovic, C. Lu, and T. F. Abdelzaher. SPEED: a stateless protocol for real-time communication in sensor networks. International Conference on Distributed Computing Systems (ICDCF'03), May 2003.

[5] G. Larsen, P. Pettersson, and W. Yi. UPPAAL in a nutshell. Int. Journal on Software Tools for Technology Transfer, 1(1):134 152, December 1997.

[6] C. Lu, B. M. Blum, T. F. Abdelzaher, J. A. Stankovic, and T. He. RAP: a real-time communication architecture for large-scale wireless sensor networks. IEEE Real Time Technology and Application Symposium (RTAS'02), September 2002.

[7] Simon R. Saunders. Antennas and propagation for wireless communication systems. Wiley, 1999.

[8] J. A. Stankovic. Research challenges for wireless sensor networks. SIGBED Review: Special Issue on Embedded Sensor Networks and Wireless Computing, 1(2), July 2004.