Preliminary Discussion on Globally Prioritized Medium Access for Multi-Channel Wireless Systems

Björn Andersson, Nuno Pereira and Eduardo Tovar
IPP Hurray Research Group
Polytechnic Institute of Porto, Portugal
{bandersson,npereira,emt}@dei.isep.ipp.pt

Abstract

We discuss the development of a simple globally prioritized multi-channel medium access control (MAC) protocol for wireless networks. This protocol provides "hard" pre-run-time real-time guarantees to sporadic message streams, exploits a very large fraction of the capacity of all channels for "hard" real-time traffic and also makes it possible to fully utilize the channels with non real-time traffic when hard real-time messages do not request to be transmitted.

The potential of such protocols for real-time applications is discussed and a schedulability analysis is also presented.

1. Introduction

We consider the problem of designing a multi-channel medium access control (MAC) protocol for wireless networks. That is, our MAC protocol is designed for nodes equipped with radio transceivers that can receive in multiple channels at the same time. Such design aims at providing pre-run-time guarantees for real-time traffic, and thus we design a globally prioritized MAC protocol.

The fact that it is globally prioritized means that the transmission of a message is only delayed if *all* channels are being used to transmit higher-priority messages. Such a protocol would give application developers the ability of achieving "hard" pre-run-time real-time guarantees to sporadic message streams and exploit a very large fraction of the capacity of all channels for "hard" real-time traffic. It would also make it possible to fully utilize the channels with non real-time traffic when hard real-time messages do not request to be transmitted. Unfortunately, the current research literature does not (as far as we know) offer such a protocol [1].

We give a preliminary discussion on such a protocol and show how it can be designed. We also propose a schedulability analysis for it. Initially, we assume that each computer node is equipped with one transmitter module that can transmit to any selected channel and the computer node is also equipped with *CH* receiver modules where each receiver module is assigned to a specific channel (*CH* denotes the number of channels). Since contemporary hardware does not have this capability, we will later on in this paper discuss how our new protocol can be adapted to contemporary hardware.

The remainder of this paper is structured as follows. Section 2 presents the system model and the assumptions we make. Section 3 presents the MAC protocol whereas Section 4 presents its schedulability analysis. Section 5 discusses practical aspects and previous works on multi-channel MAC protocols. Finally, Section 6 gives conclusions and future work.

2. System model

Consider *m* computer nodes in a single broadcast domain, that is, every computer node can hear every other transmission. We assume $1 \leq m \leq UBm$, where *UBm* is an upper bound on *m*. It is assumed that computer nodes do not know *m* but they know *UBm*.

Each computer node is equipped with one transmitter module that can transmit to any channel and the computer node is also equipped with *CH* receiver modules where each receiver module is assigned to a specific channel. We say that channel 1 is the control channel, meaning that it will be used for arbitration. It is assumed that the transmission on one channel does not interfere with a transmission on another channel.

If an arbitrary computer node broadcasts an unmodulated carrier wave for *TFCS* time units, then any other node will reliably detect the existence of that carrier wave. Let *CMAX* denote the maximum packet size. We assume that the computer node can call a function `carrierOn` which causes the transmitted module to immediately start transmitting a carrier. There is also a function `carrierOff` which causes the transmitted module to immediately stop transmitting a carrier. This is close to a realistic transceiver; typically they are also able to start and stop the transmission of an unmodulated carrier within just one microsecond. Each message is assigned a priority. It is assumed that priorities are assigned such that any set

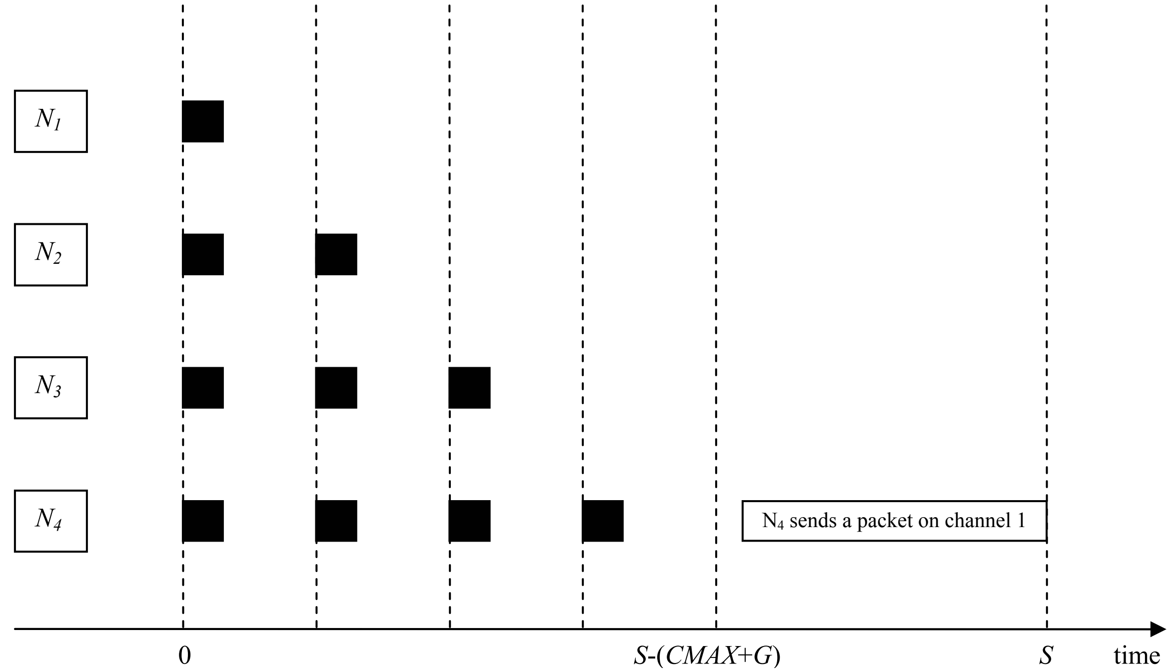

Figure 1. The illustrated MAC protocol is prioritized, but it does not allow parallel transmissions. The black filled rectangles indicate that the computer node sends an unmodulated carrier.

of messages that are contending with each other has unique priorities. One way to achieve this is to use the sporadic model (see Section 4) and assign unique priorities to message streams. It is assumed that a high number means high priority. Occasionally, we speak about the priority of a computer node and then it means the priority of the highest-priority message of that computer node.

3. Multi-Channel MAC

We will present three multi-channel MAC protocols. First, Section 3.1 presents a multi-channel MAC protocol assuming a slotted system; that is, an external device notifies all computer nodes that a slot starts. We let S denote the slot size. We do not bother about identifiers of slots; we only care about ensuring that all computer nodes know the start time of a slot.

Then we will (in Section 3.2) extend this protocol to a slotted system but without any external reference signal. Finally, we will (in Section 3.3) show how the protocol (in Section 3.2) can be designed to be used for existing transceivers. It will be designed for computer nodes equipped with only a transceiver and this transceiver can at a moment only transmit to one channel (which may be changed at run-time) or receive from one selected channel (which can be changed at run-time); that is, it cannot listen to all channels at the same time.

3.1. Multi-Channel MAC for Slotted Systems

A prioritized MAC protocol should select the CH highest priority messages among all messages that are contending at an instant. In order to build up the intuition to understand the design of the new protocol, consider Example 1.

Example 1. Consider Figure 1. It shows four computer nodes N_1, N_2, N_3 and N_4 and there are two channels available (CH=2). Each computer node requests to transmit a message at time 0. The message requested to be transmitted by node N_i has priority i.

We use a scheme similar to black-bursts [5] but we will modify it slightly. Every computer node waits for the beginning of a time slot. A time slot starts at time 0; time S, time $2S$, time $3S$, etc. We will now consider the time interval $[0,S)$. This time interval consists of the interval $[0,S-(CMAX+G))$ and $[S-(CMAX+G),S)$. (G is a parameter that will be discussed later on). The former interval is used for arbitration for the medium and the latter is for transmitting the data payload. The time interval $[0,S-(CMAX+G))$ is split into UBm subintervals. Let us index these subintervals $1,2,3,\ldots,UBm$. In Figure 1, it is assumed that $UBm=m=4$. A node sends a pulse of an unmodulated carrier wave in the beginning of a subinterval.

A node with priority i does this for the subintervals with index $1..i$. For example, node 1 sends a pulse in the

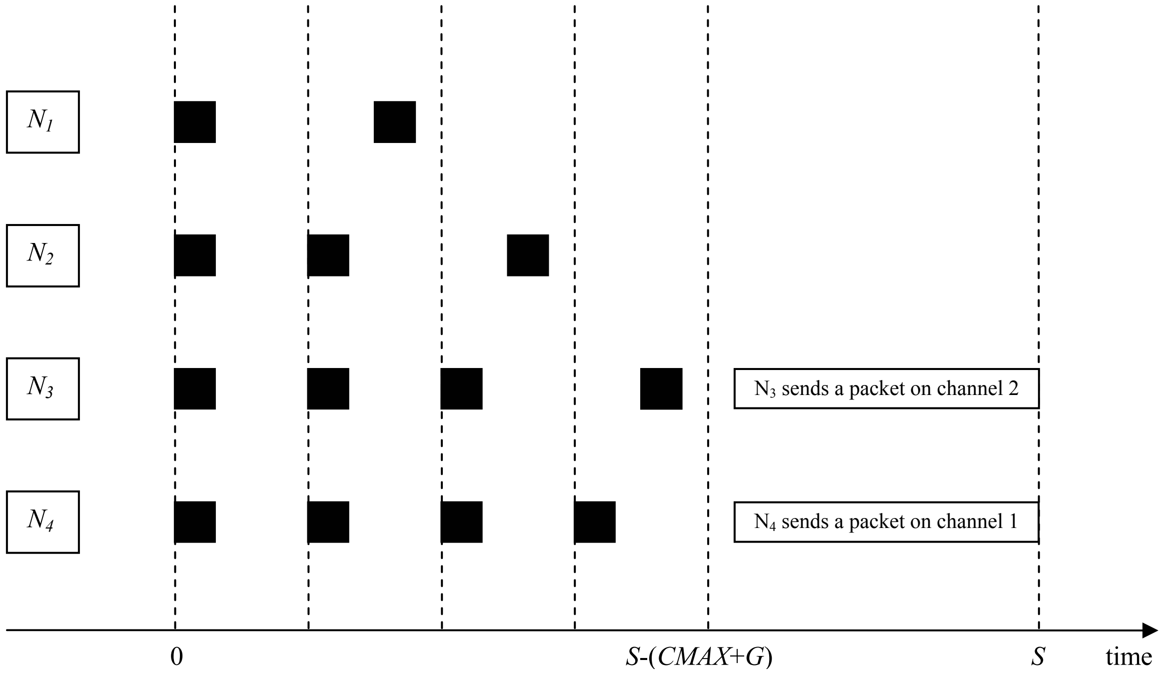

Figure 2. This MAC protocol is prioritized and allows parallel transmissions. The filled black rectangles indicate that the computer node sends an unmodulated carrier.

beginning of subinterval 1. Node 2 sends a pulse in the beginning of subinterval 1 and in the beginning of subinterval 2. For those subintervals, for which a node does not send an unmodulated carrier wave, the node performs carrier sensing in the beginning of the subinterval. For example, node 1 performs carrier sensing in the beginning of subinterval 2.

If a computer node detects a carrier then it declares itself as a loser. For example, node N_1 declares itself as a loser in subinterval 2. A computer node declares itself as a winner if it did never detect a carrier wave and then it sends its packet during [S-($CMAX$+G),S). Unfortunately, there is only one winner. □

It can be seen that the MAC protocol illustrated in Example 1 achieves prioritization. But only one computer node sends so it does not exploit the opportunity for parallel transmission on different channels. Example 2 illustrates how the behavior should be changed to allow parallel transmissions.

Example 2. Consider Figure 2. It shows four computer nodes N_1, N_2, N_3 and N_4 and there are two channels available (CH=2). Each computer node requests to transmit a message at time 0. The message requested to be transmitted by node N_i has priority i.

The computer nodes send unmodulated carriers as in Example 1. But in addition to that, a computer node sends a carrier in the later part of a subinterval if it lost in this time interval. For example, node 1 lost in the second subinterval and hence it sends an unmodulated carrier in the later part of the second subinterval. In each subinterval, there can be at most one node that lost (assuming that priorities are unique). For this reason, all computer nodes will not only know the priority of the winner, but they will also know for each priority level if this priority level lost. And in this way, all nodes will know the priority of the node with the highest priority, the priority of the node with the second highest priority, and so on. This allows us to design a MAC protocol that achieves global prioritization. □

Having seen the main idea on how to design a globally prioritized MAC protocol we are now in position to formally state the new protocol.

To simplify the presentation of the protocol, we do it using timed-automata like notation. States are represented as vertices and transitions are represented as edges. An edge is described by its guard (a condition which has to be true in order for the protocol to make the transition) and an update (an action that occurs when the transition is made). We let "/" separate the guards and the updates; the guards are before "/" and the update is after. Let "=" denote test for equality and let ":=" denote assignment to a variable. For those transitions with an update having many lines of code, it is assumed that the lines are executed sequentially.

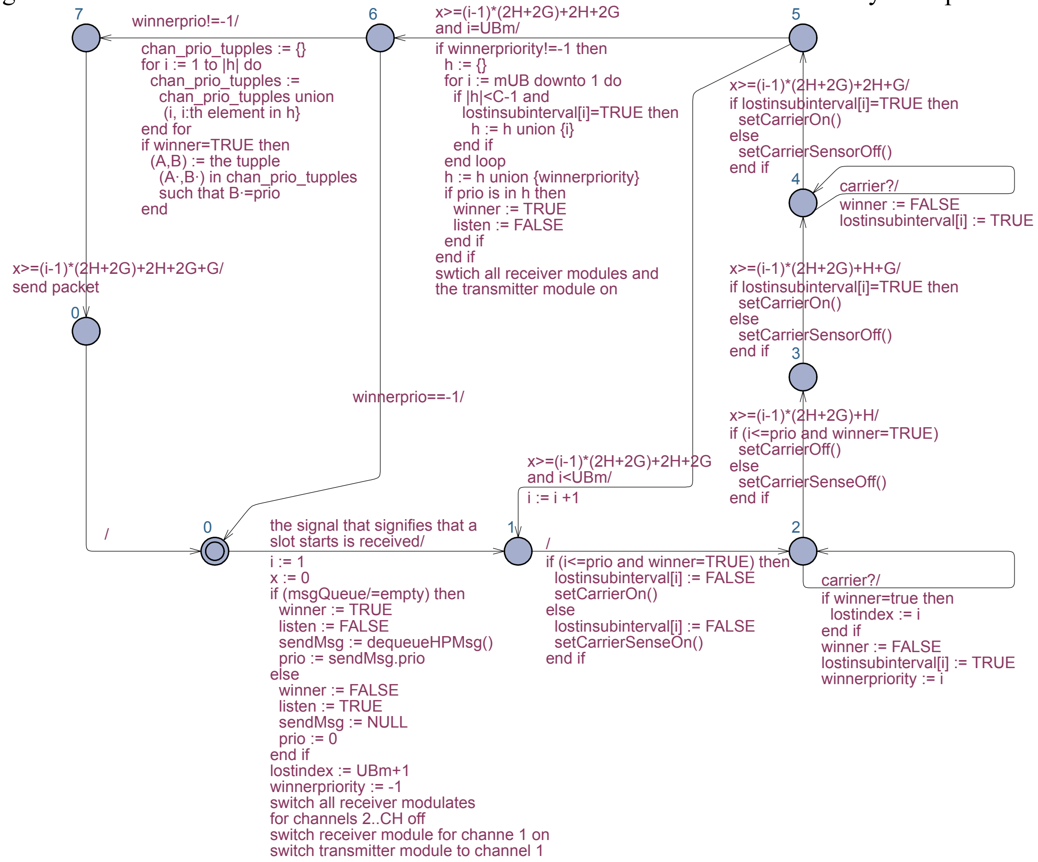

Figure 3. A timed automaton description of the proposed multi-channel MAC protocol.

Figure 3 shows the automaton. It describes the behaviour that every computer node does. First a computer node waits until it receives a signal that signifies that a time slot has begun. When that happens it makes a transition from state 0 to state 1 (state 0 is the initial state). Then the protocol iterates through all subintervals (in states 1-5). A computer node makes the transition to state 6 when all subintervals have been executed. When this transition takes place, the computer node computes a set `h` which contains all the *CH* highest priorities that contended. A node may have lost the tournament but it had still high enough priority so for this reason, it may be declared as a winner (one of the computer nodes that will send). Computer nodes make the transition to state 7 where a mapping from priorities to channels is computed. The computer nodes that are winners send their packets on the channel given by this mapping. Computer nodes that are not winners do nothing; they already listen to all channels.

The protocol depends on timeout parameters *G* and *H*. They should be selected to satisfy the following constraints:

$$2H + 2G + G + CMAX \le S \qquad (1)$$

and

$$TFCS \le H \qquad (2)$$

and

$$\begin{array}{c}\textit{difference in time that computer node receive the}\\ \textit{signal that signifies that a time slot starts} \le G\end{array} \qquad (3)$$

It can be seen that *H* is duration of a pulse and *G* is a guard band.

3.2. Multi-Channel MAC for Slotted Systems without External Synchronization

The protocol described in Section 3.1 can be easily extended to slotted systems without external synchronization. We simply let computer nodes wait for a long period of silence and then send an unmodulated synchronization pulse (as was done in WiDom [6], a prioritized MAC protocol for single-channel wireless systems).

3.3. Using Contemporary Transceivers

Typical contemporary transceivers are only able to either receive from a single (selectable at run-time) channel or transmit to a single (selectable at run-time) channel. For such transceivers, the MAC protocol from Section 3.2 can be used with the only restrictions that (i) if a computer node is a winner then it cannot hear any transmitted packet and (ii) if a computer node is not a winner then it can only hear one transmitted packet. Assuming that transmissions are unicast then we can require that the intended receiver sends an ACK (this can be considered to be part of *CMAX*) to the sender on the channel the sender used. If the sender receives an ACK then it knows that the receiver listened on that channel and hence the packet is successfully transmitted. If no ACK was received then the sender retries in the next "slot".

4. Schedulability analysis

In order to perform schedulability analysis, it is necessary to describe a model of the traffic. We consider the sporadic model. It is as follows. A computer node is assigned zero, one or many message streams. A message stream is assigned to exactly one computer node.

A message stream τ_i is characterized by D_i and T_i, where D_i is the relative deadline and T_i is the minimum inter-arrival time. A message stream τ_i performs (a possible infinite) sequence of message transmission requests. The time between two consecutive message transmission requests in a message stream is at least T_i. The time to transmit a message in message stream τ_i is at most *CMAX*. For this reason, we can assume (from the perspective of schedulability analysis) that the length of a message is S and this includes the time for arbitration. We also assume the constrained deadline case, that is, $\forall i$: $D_i \leq T_i$. We assume that priorities are assigned according to deadline-monotonic (DM) [7]; that is, the priority of a message stream is inversely proportionate to its deadline.

Inspired by results in static-priority scheduling on multiprocessors [2] and combining this way of thinking with results from the CAN analysis [3] gives that we can calculate an upper bound on the response time when the MAC protocol in Section 3.1 is used. The upper bound is obtained as follows. Find the minimum value of RUB_i that satisfies both (4) and (5):

$$RUB_i \leq S + S + \left\lceil \frac{1}{CH} \times \sum_{j \in hp(i)} \left\lceil \frac{RUB_i + S}{T_j} \right\rceil \right\rceil \times S \qquad (4)$$

$$RUB_i \leq S + S + \sum_{\substack{j \in hp(i) \wedge j \text{ is a message stream} \\ \text{on the same computer node as } i}} \left\lceil \frac{RUB_i + S}{T_j} \right\rceil \times S \qquad (5)$$

where $hp(i)$ denotes the set of message streams in the entire network that have higher priority than message stream τ_i. Observe that messages in $hp(i)$ may be assigned to other nodes than message stream τ_i. The first term in (4) is due to blocking; the second term is due to transmission and the third term is due to interference. Observe that we add S to the window that is used to compute interference. This is because the blocking due to lower-priority messages can increase the window of interference. The reasoning for deriving (5) is similar to that of deriving (4). The inequality (5) accounts for the fact that even if many channels are available, a single computer node can send at most one packet per time slot.

It can be seen that a large number of channels help to reduce the interference and hence it reduces the response time. Observe that the real response-time may be smaller than RUB_i. This is due to (i) pessimism in the analysis of non-preemptive static-priority scheduling and (ii) pessimism in the "multi-channel" aspect. We know however that if $\forall i$: $RUB_i \leq D_i$ then all deadlines are met.

The response-time calculations from (4) and (5) is valid for the MAC protocol in Section 3.1 But it is not valid for the MAC protocol in Section 3.2 and Section 3.3 because those protocol use a specific type of synchronization and it depends in a technique called "delayed dequeing" which complicates the analysis. See [6] for details.

One may ask whether DM is optimal for the system that we assume. We guess the answer is no. One may also ask whether Dhall´s effect [4], a scenario that occurs on multiprocessor scheduling that can cause deadlines misses although the multiprocessor/multiple channels are almost idle all the time but still a deadline is missed, can occur. Our guess is that if all messages have similar length then this is not a big problem.

5. Practical Aspects and Previous work

5.1. Practical Aspects

We assumed that a radio channel offers reliable broadcast. Whether this is reasonable in practice is debatable and it depends on the exact location of computer nodes, radio conditions, transmission power and detection techniques. Nonetheless, we have in previous work shown that there are environments where such assumption is reasonable [6].

We assumed it is possible to detect a carrier wave if this wave is transmitted for a duration of *TFCS* time units. The exact value of *TFCS* depends on the hardware being used. In CC2420 (a transceiver for 802.15.4), this time is 128μs. It has been reported that other radios can has *TFCS* of 5μs [10] and 20μs [5].

We assumed initially that each computer node is equipped with multiple receiver modules and one transmitter module. Such a computer node is more costly than computer nodes with normal transceivers and we are not aware of any such computer node on the market today. A computer node with a separate transmitted module and receiver module has been built (by others [8] in a collaborative project with us) and it would be possible to add more receiver modules. Consequently, the construction of a computer node with multiple receiver modulate is at least technically possible.

We assumed that when a computer node transmits to a channel then it causes no interference to transmission son other channels. Many current standards (such as IEEE 802.11 and IEEE 802.15.4) support many channels and they occupy different frequency band. The standard IEEE 802.11b has 14 channels (5 MHz each) but in order to for transmissions to be totally non-overlapping it is necessary that channels are separated by 30 MHz. Hence channel 1,6,11 in the IEEE 802.11 standard can transmit in parallel. It would be possible to use our protocol by letting channel 1 in IEEE 802.11 be our channel 1, letting channel 6 in IEEE 802.11 be our channel 2 and 11 in IEEE 802.11 be our channel 3. Then we would have *CH*=3.

5.2. Previous work

The scientific advances in multi-channel MAC protocol originates from two time periods: (i) before the IEEE 802.11 standard and (ii) after IEEE 802.11.

Before the IEEE 802.11 standard was proposed, significant research was performed on ground packet radio network, particularly in the U.S. One of the earliest multi-channel MAC protocols was proposed *receiver-directed transmission* [9]. Here it was assumed that each computer node is assigned a channel and it listens only to that channel. A computer node knows, for every of its neighbors, which channel this neighbour listens to. When a computer node wishes to send, it switches to the channel that the receiver listens to.

After IEEE 802.11 several multi-channel MAC protocol were proposed and they were more flexible. They can be categorized [1] as (i) dedicated control channel, (ii) common hopping, (iii) split phase and (iv) multiple rendezvous. They all have in common that a node that wishes to send a packet first sends a request-to-send (RTS) packet and if the receiving computer node receives this RTS packet, it responds with a Clear-to-Send (CTS) packet. They differ in when and on which channel this RTS/CTS exchange is performed and on which channel the subsequent data transmission is performed.

6. Conclusions and Future work

We have presented a globally prioritized multi-channel MAC protocol for wireless systems and a schedulability analysis for it.

We left four problems open and we consider them as future work. First, this protocol was based on the black-burst scheme [5] which offers few priority levels. So a natural question is: Can the prioritized MAC protocol WiDom [6] be extended to multi-channel wireless systems. If so a large number of priorities can be supported even with a small overhead. It is clear that one can run the normal WiDom *CH* times but we would like to design a protocol with an overhead lower than that. Second, we would like to make the schedulability analysis tighter and explore alternative priority-assignment schemes. Third, we would like to design a schedulability analysis technique that takes into account that fact that a message may be transmitted but a receiver does not listen to it and hence the message must be retransmitted. Fourth, we would like to implement the new protocol in contemporary transceivers.

References

[1] J. Mo, H. Sheung, W. So and J. Walrand, "Comparison of multi-channel MAC protocols", International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems, pages 209 – 218, 2005, Montréal, Quebec, Canada.

[2] B. Andersson and J. Jonsson, "Some Insights on Fixed-Priority Preemptive Non-Partitioned Multiprocessor Scheduling", in Proc. of the IEEE Real-Time Systems Symposium, Work-in-Progress Session, November 27-30, 2000, Orlando, Florida.

[3] R. I. Davis, A. Burns, R. J. Bril and J. J. Lukkien, "Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised". Journal of Real-Time Systems, pages 239-272, vol 35, issue 3, 2007.

[4] S. Dhall and C. Liu, "On a real time scheduling problem", Operations Research, 26(1), 1978, pages 127-141.

[5] J. L. Sobrinho and A. S. Krishnakumar, "Real-time traffic over the IEEE 802.11 medium access control layer," Bell Labs Tech. J., vol. 1, no. 2, pp. 172-187, 1996.

[6] N. Pereira, B. Andersson and E. Tovar, "WiDom: A Dominance Protocol for Wireless Medium Access", IEEE Transactions on Industrial Informatics, Vol 3, No 2, May 2007.

[7] J. Y. Leung and J. Whitehead, "On the complexity of fixed-priority scheduling of periodic, real-time tasks," Performance Evaluation, 2(4):237-250, December 1982.

[8] http://www.nanork.org:8000/nano-RK/wiki/wings

[9] N. Shacham and P. King, "Architectures and Performance of Multichannel Multihop Packet Radio Networks", IEEE Journal on Selected Areas in Communications, 5(6), July 1987, pages 1013- 1025.

[10] F. Tobagi and L. Kleinrock, "Packet Switching in Radio Channels: Part II--the Hidden Terminal Problem in Carrier Sense Multiple-Access and the Busy-Tone Solution", IEEE Transactions on Communications, Vol. COM-23, No. 12, pp. 1417-1433, December 1975.