IPP HURRAY!

www.hurray.isep.ipp.pt

Technical Report

New Schedulability Analysis for WiDom

Maryam Vahabi

Björn Andersson

HURRAY-TR-100507

Version:

Date: 05-25-2010

New Schedulability Analysis for WiDom

Maryam Vahabi, Björn Andersson

IPP-HURRAY!

Polytechnic Institute of Porto (ISEP-IPP)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8340509

E-mail: mmvi@isep.ipp.pt, bandersson@dei.isep.ipp.pt

http://www.hurray.isep.ipp.pt

Abstract

WiDom is a wireless prioritized medium accesscontrol (MAC) protocol which offers a very large number ofpriority levels. Hence, it brings the potential for employingnon-preemptive static-priority scheduling and schedulabilityanalysis for a wireless channel assuming that the overhead ofWiDom is modeled properly. One schedulability analysis forWiDom has already been proposed but recent research has created a new version of WiDom with lower overhead (we call it: WiDom with a master node) and for this version of WiDom no schedulability analysis exists. Also, common to the previously proposed schedulability analyses for WiDom is that they cannot analyze message streams with release jitter. Therefore, in this paper we propose a new schedulability analysis for WiDom (with a master node). We also extend the WiDom analyses (with and without master node) to work also for message streams with release jitter.

New Schedulability Analysis for WiDom

Maryam Vahabi and Björn Andersson
IPP-HURRAY Research Group, CISTER/ISEP
Polytechnic Institute of Porto
Porto, Portugal
mmvi@isep.ipp.pt, bandersson@dei.isep.ipp.pt

***Abstract*—WiDom is a wireless prioritized medium access control (MAC) protocol which offers a very large number of priority levels. Hence, it brings the potential for employing non-preemptive static-priority scheduling and schedulability analysis for a wireless channel assuming that the overhead of WiDom is modeled properly. One schedulability analysis for WiDom has already been proposed but recent research has created a new version of WiDom with lower overhead (we call it: WiDom with a master node) and for this version of WiDom no schedulability analysis exists. Also, common to the previously proposed schedulability analyses for WiDom is that they cannot analyze message streams with release jitter. Therefore, in this paper we propose a new schedulability analysis for WiDom (with a master node). We also extend the WiDom analyses (with and without master node) to work also for message streams with release jitter.**

Keywords-Medium Access Control; Dominance/binary count-down protocol; Schedulability Analysis

I. INTRODUCTION

Medium access control (MAC) protocols, by assuring a collision-free transmission, can have a significant impact on whether timing requirements are met. For example an unsuccessful transmission needs a retransmission which results in a longer delay.

WiDom [1] is an adaptation of the dominance protocol used in the CAN bus to the wireless domain. It is a collision-free prioritized MAC protocol in which each message is assigned a unique priority. During contention resolution, the wireless channel is granted to a node with the highest priority message among the messages that are requested to be transmitted. WiDom supports a large number of priority levels without imposing a huge overhead on the system compared to other protocols. This low-overhead protocol is already implemented [1,2] on MicaZ and Firefly [3] platforms. Most of the other wireless protocols cannot be analyzed to offer pre-run-time guarantees that message streams meet their deadlines. Some protocols that offer such guarantees rely on polling, which is inefficient when the deadline is short, while WiDom can provide such schedulability analysis by calculating the response time of a message stream and comparing it to the deadline of the message stream. However, the existing analysis does not consider release jitter. It also does not provide any response time analysis for a new type of WiDom where synchronization is made by a master node.

In this paper we present a schedulability analysis of WiDom considering its overhead and for message streams with release jitter. We also develop this work for WiDom including a master synchronization node. However, it is important to mention that we just focus on a single broadcast domain and assume that there is no noise in the system that cause faultiness in the functionality of WiDom.

The following section presents a brief background on schedulability analysis of static-priority scheduling on CAN bus and the previous version of WiDom. Section III explains the mechanism of the WiDom protocol followed by the new analyses which incorporate release jitter.

II. BACKGROUND ON SCHEDULIBILITY ANALYSIS OF NON-PREEMPTIVE STATIC-PRIORITY SCHEDULING

This section presents the previously known schedulability analysis of CAN and then presents the previously proposed analysis of WiDom. Our new schedulability analysis (in Section IV) will be based on these.

A. Controller Area Network (CAN)

The CAN bus implements non-preemptive static-priority scheduling on a wired channel and for this reason, early researchers [4] realized that the uniprocessor preemptive static-priority scheduling theory [5] could be modified for non-preemptiveness and applied to CAN. Davis *et al* [6] proposed the first correct analysis of the CAN bus by revising this analysis by considering the fact that in the non-preemptive static-priority scheduling, for a given message m, a higher priority message can be awaiting for transmission when message m completes transmission. Thus the busy period can extend beyond of the period of message m. To be more accurate in the calculation they first determine the duration of *level-m busy period* as follows:

$$t_m = B_m + \sum_{\forall i \in hp(m) \cup m} \left\lceil \frac{t_m + J_i}{T_i} \right\rceil \times C_i \qquad (1)$$

where $hp(m) \cup m$ is the set of message streams with priority m or higher; B_m is maximum blocking time that can be imposed by a lower priority message; release jitter or queuing jitter [4], J_j, is defined as the largest difference between initiating time of the event and the time in which that message has been queued; C_i and T_i are transmission time and minimum inter-arrival time of message stream i respectively. Then for calculating the response time for message stream m, the response time for all the instances of this message stream located in the level-m busy period should be calculated. Finally the response time of a message instance which has the largest value among other instances during the busy period will be considered as the worst case response time (WCRT) of the message stream m and is computed as follow:

$$R_m = \max_{q=0,\ldots Q_m-1} (J_m + w_{m,q} - qT_m + C_m) \quad (2)$$

where Q_m is the number of message instances located in the level-m busy period and is given by:

$$Q_m = \left\lceil \frac{t_m + J_m}{T_m} \right\rceil \quad (3)$$

and $w_{m,q}$ can be defined as follows:

$$w_{m,q} = B_m + qC_m + \sum_{\forall i \in hp(m) \cup m} \left\lceil \frac{w_{m,q} + J_i + \tau_{bit}}{T_i} \right\rceil \times C_i . \quad (4)$$

B. Wireless Dominance MAC (WiDom)

The existing response time analysis [1] of WiDom assumes no release jitter. As stated before, the WCRT of a message stream is the longest response time of all message instances q that enter the ready queue for a period of time which is called busy period, so:

$$R_i = \max_{q=0,\ldots,Q_i-1} \left\{ w_{i,q} + C''_i - q \times T_i \right\} \quad (5)$$

where Q_i is given by:

$$Q_i = \left\lfloor \frac{L_i}{T_i} \right\rfloor + 1 \quad (6)$$

and L_i is the length of the longest level-i busy period and can be formulated as follow:

$$L_i = \max_{j \in lp(i)} \left\{ C'_i - Q_{bit} \right\} + \sum_{j \in hp(i) \cup i} \left\lceil \frac{L_i}{T_j} \right\rceil \times C''_j \quad (7)$$

where $hp(i)$ is similarly the set of message streams with priority higher than i, and $lp(i)$ is the set of message streams with priority lower than that. Chip duration Q_{bit} is the time granularity which is similar to τ_{bit} in CAN analysis. In the current implementations of WiDom [1,2], the radio uses direct-sequence spread-spectrum (DSSS) in which every 4 bits is modulated as 32 chips so that the data rate reaches 2Mchip/s which is equivalent to 250 Kbits/s. For such a platform, we have Q_{bit} =4/250000=16μs. C''_i is the time span needed to finish transmission. It consists of synchronization time, F, together with tournament duration, C'_i .Then waiting time $w_{i,q}$ is:

$$w_{i,q} = q \times C''_i + \max_{k \in lp(i)} \left\{ C'_k - Q_{bit} \right\} \quad (8)$$

$$+ \sum_{j \in hp(i)} \left\lceil \frac{w_{i,q} + F + E + \max\{TFCS, SWX\} + H + Q_{bit}}{T_j} \right\rceil \times C''_j$$

where F is a long period of silence that nodes should wait before contending for the channel, E is the duration of time that is considered for encompassing clock drift between the nodes and to guarantee that all nodes have time to listen for F time units of silence. Time for carrier sense (*TFCS*) is the duration of time that a node needs in order to detect a carrier transmission. To have a good perception of these parameters it is needed to know how WiDom works. In the next section we will describe the WiDom functioning procedure in brief.

III. WiDom Mechanism

As stated before, WiDom is a prioritized MAC protocol for wireless networks and hence the message with the highest priority (corresponding to the lowest priority number) is granted the channel. When messages contend for the channel, a conflict resolution phase (called tournament) similar to the dominance/binary countdown arbitration is performed. During the tournament nodes transmit the priority of the message contending for the medium bit-by-bit. A bit is said to be dominant if it is "0"; it is said to be recessive if it is "1". The protocol is composed of three phases, namely: synchronization, tournament and data exchange — see Figure 1. The synchronization is needed to provide a common reference point in time so that all nodes can start the competition at the same time. Hencethe synchronization should happen before the tournament and at last the node that wins the tournament will start transmission.

At the start of synchronization, nodes should wait for a long period of idle time F — see Figure 1, such that no node disrupts an ongoing tournament. Then nodes with a pending message wait for another time span E to compensate for the potential clock drift and also ensuring that all nodes have enough time to listen for F time units. Afterwards, nodes start sending a carrier pulse for a duration of H time units that signals the start of a tournament and establishes a common time reference. To do so they have to switch from receive mode to send mode which takes *SWX* units of time.

By sending this signal, all nodes restart their timers and synchronization ends.

In the tournament phase if a node loses the contention of a bit (i.e. it transmits a recessive bit and receives a dominant bit), it does not continue further bits and only proceed listening to the medium to find out the priority of the winner. If a node does not lose the contention during the current bit it will proceed with the contention for the next bit. The nodes that have dominant bit, start transmitting a pulse of carrier for duration of *H* unit of time, while nodes with recessive bit, perform carrier sensing. Also, note that the fact that wireless transceiver are not able to send and transmit simultaneously poses no problem to WiDom since when a node has a dominant bit, there is no need for reception and when a node has a recessive bit, it sends nothing; it performs carrier sensing. There is also a guarding time *G* interval to separate pulses of carrier wave. This guarding time interval makes the protocol robust against clock inaccuracies, and takes into account that signals need a non-zero time to propagate from one node to another. At the end of tournament, the node that does not receive a pulse wins the competition and wait for *ETG* time before starting data transmission.

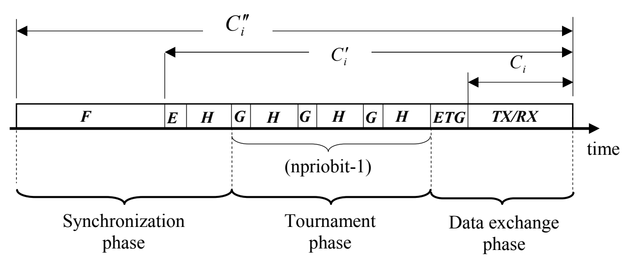

Figure 1. Timing Order of WiDom without Master Node.

There is another version for WiDom [2] that uses a master node to reduce the overhead of synchronization. To do so, the master node broadcasts a synchronization pulse periodically on a separate radio channel. Thus, nodes are not forced to wait for a long period of *F* time duration in order to be synchronized.

IV. New Analysis

In this section we compute the WCRT of a given message stream. We develop this analysis by considering release jitter and assuming that all nodes are in a single broadcast domain. We will consider two different type of WiDom, one with the master node and the other one without.

A. WiDom without Master Node

The idea of this extension is obtained from [6]. Following the analysis in [1], the response time calculation can be made up of two distinct elements:

- $w_{i,q}$, corresponding to the longest time from the start of the busy period till when the instance *q* starts successful transmission [6].
- C_i'' , Time span needed to finish transmission. It consists of synchronization time, *F*, together with tournament duration, C_i'. Or it can be expressed in a mathematical way:

$$C_i'' = C_i + F + E + ETG + H + (npriobits - 1)(G + H) \tag{9}$$

In general, queuing of a message can occur with jitter [7]. To provide a precise response time calculation, we follow the previous analysis by considering this release jitter. Accordingly the WCRT can be computed as follow:

$$R_i = \max_{q=0 \ldots Q_i - 1} \left\{ w_{i,q} + J_i + C_i'' - q \times T_i \right\} \tag{10}$$

where Q_i is given by:

$$Q_i = \left\lfloor \frac{L_i + J_i}{T_i} \right\rfloor + 1 . \tag{11}$$

L_i is the length of level-*i* busy period — see Figure 2.

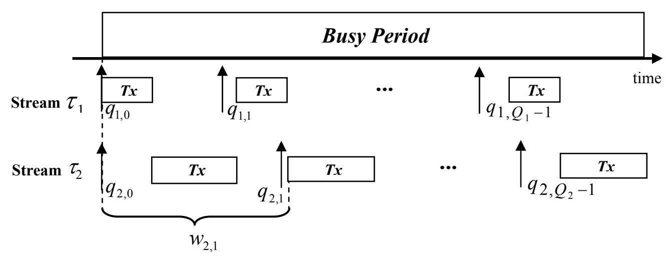

Figure 2. An Example of $w_{i,q}$.

The key characteristic of a busy period is that all messages of priority *i* or higher queued strictly before the end of the busy period are transmitted during this period. Therefore, level-i busy period is the smallest value given by:

$$L_i = \max_{j \in lp(i)} \left\{ C_i' - Q_{bit} \right\} + \sum_{j \in hp(i) \cup i} \left\lceil \frac{L_i + J_j}{T_j} \right\rceil \times C_j'' \tag{12}$$

where *hp(i)* is the set of message streams with priority *i* and analogously, *lp(i)* comprises all message streams with priority less than *i*. Here again we assume that the radio transceivers use DSSS so, the transmission of a data bit of duration t_b is replaced by transmission of a finite chip sequence c. Hence, Q_{bit} , which is the chip duration, is considered as the time granularity [1]. The longest time from the start of the busy period to the time in which instance *q* begins transmission successfully — see Figure 2, is given by:

$$w_{i,q} = q \times C_i'' + \max_{k\in lp(i)}\{C_k' - Q_{bit}\} \qquad (13)$$

$$+ \sum_{j\in hp(i)}\left(\left\lceil \frac{w_{i,q} + F + E + \max\{TFCS, SWX\} + H + J_j + Q_{bit}}{T_j}\right\rceil\right) \times C''.$$

B. *WiDom with Master Node*

In this subsection, we consider a master node in a single broadcast domain to achieve more accurate synchronization and also to reduce the overhead. This extension has been proposed in [2] in which nodes have been equipped with an extra radio circuit.

Unfortunately, no schedulability analysis for WiDom with master node was presented. In this case, a master node sends pulses periodically with period of P_S on a separate channel to announce the start of the tournament and immediately afterwards, nodes start executing the tournament. The advantage of using the separate receiver is the possibility of setting it perpetually in reception mode and eliminating the switch time. As it is shown in Figure 3 the second element of response time will be changed as follow:

$$C_i' = C_i + G + (H + G)(npriobits - 1) + ETG + L \qquad (14)$$

$$C_i'' = C_i' + H \qquad (15)$$

It is obvious that it reduces the overhead by replacing the H units of time for synchronization instead of waiting for the long period of time F, switch time SWX and clock drift E. For analyzing the response time, we can follow the equations (10) and (12) by considering the new value for C_i'' and C_i' in equations (14) and (15).

The longest time from the start of the busy period to the time in which instance q begins transmission successfully is given by:

$$w_{i,q} = q \times C_i'' + P_s - (H - t) \qquad (16)$$

$$+ \sum_{j\in hp(i)}\left(\left\lceil \frac{w_{i,q} + J_j + Q_{bit}}{T_j}\right\rceil\right) \times P_s$$

where P_S is the periodicity in which synch signal is broadcast through the network. This period should be chosen in a way that a message with the longest transmission time (C_i) could be able to finish its transmission before the start of next synch signal. This constraint is formulated as follow:

$$P_S \geq (H + G)(npribits) + ETG + L + \max\{C_i\} + TFCS \qquad (17)$$

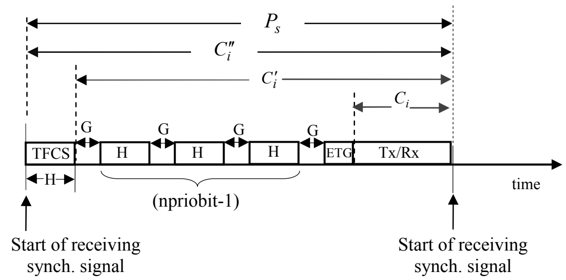

Figure 3. Timing Order of WiDom with Master Node.

where L is the time needed for some computational purposes inside the WiDom protocol.

V. CONCLUSION AND FUTURE WORK

In this paper we focused on a recent prioritized MAC protocol, WiDom, and developed its schedulability analysis by considering release jitter for two different implementation models of this protocol. For future work we plan to compare the numerical results from the presented calculation with the ones that will be obtained through experiments using real-world implementations of WiDom.

ACKNOWLEDGEMENTS

This work was partially supported by CONET, the Cooperating Objects Network of Excellence, funded by the European Commission under FP7 with contract number FP7-2007-2-224053, ARTISTDesign Network of Excellence on Embedded Systems Design, funded by the European Commission under FP7 with contract number ICT NoE-214373 and the Portuguese Science and Technology Foundation (Fundação para a Ciência e a Tecnologia - FCT).

REFERENCES

[1] N. Pereira, B. Andersson and E. Tovar, "WiDom: a dominance protocol for wireless medium access," *Industrial Informatics, IEEE Transactions on,* vol. 3, pp. 120-130, 2007.

[2] N. Pereira, R. Gomes, B. Andersson and E. Tovar, "Efficient aggregate computations in large-scale dense WSN," in *Real-Time and Embedded Technology and Applications Symposiumin, 2009, 15th IEEE International*, pp. 317-326, 2009.

[3] R. Mangharam, A. Rowe and R. Rajkumar, "FireFly: a cross-layer platform for real-time embedded wireless networks," *Real-Time Systems,* vol. 37, pp. 183-231, 2007.

[4] K. Tindell, H. Hansson and A. J. Wellings, "Analysing real-time communications: controller area network (CAN)," in *Real-Time Systems Symposium,1994. Proceedings.*, pp. 259-263, 1994.

[5] M. Joseph and P. Pandya, "Finding response times in a real-time system," *Computer journal,* vol. 29, pp. 390-395, 1986.

[6] R. Davis, A. Burns, R. Bril and J. Lukkien, "Controller area network (CAN) schedulability analysis: refuted, revisited and revised," *Real-Time Systems,* vol. 35, pp. 239-272, 2007.

[7] N. Audsley, A. Burns, M. Richardson, K. Tindell and A. J. Wellings, "Applying new scheduling theory to static priority pre-emptive scheduling," *Software Engineering Journal,* vol. 8, pp. 284-292, 1993.