MAC-Level Communication Time Modeling and Analysis for Real-Time WSNs

Valentin STANGACIU, Mihai MICEA, Vladimir CRETU
Department of Computer and Software Engineering, Politehnica University of Timisoara, Romania
valys@dsplabs.cs.upt.ro, mihai.micea@cs.upt.ro, vladimir.cretu@cs.upt.ro

***Abstract*—Low-level communication protocols and their timing behavior are essential to developing wireless sensor networks (WSNs) able to provide the support and operating guarantees required by many current real-time applications. Nevertheless, this aspect still remains an issue in the state-of-the-art. In this paper we provide a detailed analysis of a recently proposed MAC-level communication timing model and demonstrate its usability in designing real-time protocols. The results of a large set of measurements are also presented and discussed here, in direct relation to the main time parameters of the analyzed model.**

***Index Terms*—Wireless sensor networks (WSNs), Real-time systems, Wireless communication, Access protocol, Time measurement**

I. Introduction

Some of the most studied domains during the last decade are wireless sensor networks and real-time systems. Furthermore, consistent research has been made in a field where wireless sensor networks and real-time systems are combined together in order to solve problems in critical, time bounded situations.

Generally speaking, a real-time application proves its usefulness when it is generating a correct result or action in a specified amount of time. In such a system, the same importance is given to the amount of time a result is generated as well as the result itself [1, 2].

The same idea is applied when using a wireless sensor network in a real-time environment. Achieving real-time constraints in a WSN depends on the fact that not only each node of the network must meet its time requirements but also the whole network has to function in a real-time manner. On the node side this can be achieved by using embedded real-time operating systems with the necessary adaptations and improvements to meet the specifications [3-5]. On the other hand, if the whole wireless sensor network has to function in a real-time manner, the communication delays have to be predicted and controlled. This implies that the time of the arrival of a frame from its transmitter to its intended destination is of key importance for the communications in WSNs [6].

Many of today's applications have been implemented using WSNs, thus proving their usefulness [7] in fields like monitoring and controlling the environment [8, 9], security systems, robotic environments [10], industrial process control [11, 12], fire detection systems [13, 14] and military systems with hard real-time demands [15].

II. Related work

In order for a wireless sensor network to function as a real-time system, the communication protocols have to meet specific time constrains in order to deliver time bounded messages. Higher level protocols such as the ones that represent the application layer and the network layer have been intensely studied and important advances have been made in routing protocols [16, 17].

A very important aspect regarding the communication protocols for WSNs used in applications described above is that they also have to take energy efficiency into consideration [18], but without degrading the real-time support [19]. In many situations, energy efficient protocols were developed without taking into consideration the real-time aspects, even if some of them actually offer a very weak support for real-time [18, 20].

On the other hand, lower level protocols are not so evolved regarding their real-time behavior and issues still remain to be studied. In many situations the lower level protocols are either considered to operate in a real-time manner thus meeting all their deadlines or they are totally ignored. Assumptions are also often made that lower level protocols do not introduce any delay or unpredictability to the system [21].

Many studies on WSN communications rely on a vastly used MAC level protocol, defined by the IEEE 802.15.6 standard [22], which provides a very weak support for real-time applications [23, 24], even with the latest improvements [25].

The cluster based network organization is the present in almost all of the studies presented. In this configuration, the communication relies on special nodes with crucial importance. This can introduce high unreliability into the network, because these special nodes are highly active regarding communication and can seriously affect the network in case of failure. Also, in many situations, researchers assume that no mobile nodes are present in the network and the position of the nodes is known at deployment [26].

Soft real-time requirements are satisfied by most of the medium access protocols found in the literature [27]. Significant advances have also been made in real-time communications at MAC level when using FDMA and TDMA based protocols [28], but with important drawbacks,

This work was partially supported by the strategic grant POSDRU/159/1.5/S/137070 (2014) of the Ministry of National Education, Romania, co-financed by the European Social Fund – Investing in People, within the Sectoral Operational Programme Human Resources Development 2007-2013.

This work was partially supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS – UEFISCDI, project number PN-II-RU-TE-2014-4-0731

Digital Object Identifier 10.4316/AECE.2016.01005

such as the presence of a network coordinator node thus enforcing a cluster based organization of the network. These solutions are not valid in an ad-hoc, unorganized network architecture, as well as on a dynamic network in terms of mobility.

Another possible issue present in today's literature is that many MAC protocol models and many WSN applications are validated using simulation environments, which do not take into account the behavior of the nodes in real life conditions or the delays introduced by the radio modules [29].

This paper extends the work presented in [30] by making a detailed analysis of the timing aspects of MAC communication in wireless sensor network and by proposing a comprehensive model to determine the time parameters involved in a point-to-point MAC level communication. The proposed model parameters are then experimentally evaluated using a custom-designed measurement framework. The results gathered after testing several platforms with different radio modules, are also presented and discussed in details.

III. MAC-Level Communication Timing Model

A set of assumptions and basic specifications have been considered for the design of the proposed communication timing model. Each node of the network features a wireless communication module which is driven by a host microcontroller. One of the main functions of the microcontroller is to implement the real-time communication protocol stack by taking into account the MAC layer processing delay, besides the message delivery time between two nodes. This processing time consists mainly of the delay introduced by the wireless module driver. The wireless radio module is also assumed to be either a transmitter or a receiver, not a transceiver.

In all the cases considered here, the transmitter node is denoted as "A" and the receiver node is denoted as "B". The nodes are assumed to have only these capabilities and their roles are not interchangeable in terms of transmitter and receiver.

In (1) we present the model described in [30], which represents the total delay for a transaction between a transmitter node A and a receiving node B

$$T(A,B) = T_{transfer_A} + T_{TXON_A} + T_{TX\_SFD_A} + T_{RX\_SFD_B} + T_{RX\_PACK_B} + T_{transfer_B} \quad (1)$$

The time parameters in (1) are defined as follows:

- $T_{transfer\_A}$ – the time needed for the host processor of node A to transfer the packet to the wireless module. This time component is predictable and calculable. It depends on the packet size and on the wired communication protocol between the host process and the wireless module.
- T_{TXON_A} – the time needed by the wireless module to deactivate the receiver and to activate the transmitter. This parameter is depended on the wireless module characteristics. Usually this parameter is given by the manufacturer of the module in its datasheet, as a maximum value.
- $T_{TX\_SFD\_A}$ – the time needed by the wireless module to transmit the Start of Frame Delimiter (SFD) sequence. This component usually includes the clear channel assessment (CCA) procedure. This parameter is highly dependent on the wireless medium.
- $T_{RX\_SFD\_B}$ – the time needed for the SFD sequence to arrive at its destination node B. This parameter depends on the wireless medium characteristics as well as on the bitrate and modulation used by the wireless modules.
- T_{RXPACK_B} – the amount of time needed by module B to receive the whole packet.
- $T_{transfer\_B}$ – the time needed by the host processor of node B to transfer the newly arrived packet from the wireless module to its internal memory. This parameter has the same characteristics as $T_{transfer\_A}$.

Another important component of the wireless communication timing is represented by the time needed by the host microcontroller to control the transmission and reception process. This aspect is important when designing the driver for the wireless module in a real-time context. Usually in a WSN with real-time capabilities, a real-time operating system (RTOS) is needed, as the main software component of each node. A driver design for a wireless module within such an RTOS needs to take into consideration several time components. For a transmitter node (A), these time components can be described as follows:

$$T(A) = T_{transfer} + T_{TXON} + T_{TX\_SFD} + T_{TX\_PACK} + T_{RX\_ON} \quad (2)$$

The time parameters in (2), like $T_{transfer}$, $T_{transfer\_A}$, T_{TXON}, T_{TX_SFD} are similar to those in (1). The remaining parameters have the following meaning:

- T_{TX_PACK} – the time needed for the module to transmit the entire packet. This component is usually predictable and is dependent on the packet size
- T_{RX_ON} – the time needed by the module to change from the transmitting mode to receiving mode after the packet has been sent. This parameter is usually a characteristic of the wireless module.

The same analysis is performed on the receiver side (B):

$$T(B) = T_{RX\_PACK} + T_{transfer} \quad (3)$$

Here, parameter T_{RX_PACK} can be described as the time from the moment the module signals the arrival of a SFD sequence to the moment the module receives the whole packet. The $T_{transferB}$ parameter is the time needed for the host microcontroller to transfer the received data from the communication modules.

IV. Communication timing model analysis

All of the time components described above may be further decomposed to increase the accuracy of the model.

The time needed to transfer a packet between the host processor and the radio module ($T_{transfer\_A}$, $T_{transfer\_B}$, $T_{transfer}$) is present in all of the above models. This parameter may be analyzed as follows:

$$T_{transfer} = 8 \cdot P_S \cdot T_{transfer\_BIT} \quad (4)$$

In (4), we define P_S as the packet size in bytes and $T_{transfer\_BIT}$ the time needed to transfer 1 bit of information. Giving the fact that this latter term is constant and known in most of the cases, we can deduct that the whole $T_{transfer}$ component is calculable using (4). We can make this assumption mainly because in all of the cases, the communication between the host processor and the radio module is implemented using line bus protocols which are predictable and calculable, as is the case of the SPI or the UART bus.

Another component that needs to be studied is the time needed by the radio module to turn off the receiver and switch on the transmitter (T_{TXON_A}, T_{TXON}), in other words the time needed by the radio module to switch to transmitting mode. Usually this operation does not take a lot of time but in a worst case scenario this time component can be equal to the time needed by the radio module to recalibrate its internal PLL. In a real-time environment one has to consider the worst case scenario thus resulting equation (5), where T_{PLL} represents the maximum amount of time for the radio module PLL to be stabilized. This value is usually given by the manufacturer in the datasheet of the radio module.

$$T_{TXON} = T_{PLL} \tag{5}$$

A similar component is T_{RXON} which is the time needed by the radio module to switch to receiving mode. Usually, the radio modules are in receiving mode most of the time. When a transmission needs to be made, the module has to switch to transmitting mode. After the transmission has been completed the module needs to switch back to receiving mode. In a worst case scenario the switching between transmitting mode to receiving mode and vice versa may involve a PLL recalibration. In this situation, we state the following equations:

$$T_{RXON} = T_{PLL} \tag{6}$$

$$T_{RXON} = T_{TXON} \tag{7}$$

One of the most unpredictable time parameters presented above describes the time needed by the radio module to transmit the start-of-frame (SFD) sequence. This component normally includes the CCA operation and the actual transmission of the sequence. The unpredictability is given by the CCA operation. This time component is described in the following equation:

$$T_{TX\_SFD} = T_{CCA} + SFD\_SEQ\_SIZE \cdot \frac{1}{DTR} \tag{8}$$

The newly introduced terms in (8) are T_{CCA} – the time needed by the CCA operation, DTR (Data Transfer Rate), measured in bytes/second, which defines the transfer rate, a characteristic of the radio module and the actual size in bytes of the SFD sequence.

The next analyzed component is the time needed by the receiver module to get the start of frame sequence. This is usually equal to the latency of the radio module and is specified by most manufacturers in the datasheet:

$$T_{RX\_SFD} = L_{RX} \tag{9}$$

Another time parameter that does not introduce any unpredictability intro the system, mainly because it depends on the transfer rate, is T_{RX_PACK}. This component can be described as in (10):

$$T_{RX\_PACK} = P_S \cdot \frac{1}{DTR} \tag{10}$$

Using the parameters defined in (4), (5), (6), (7) and (9) and applying them to (1), we can define a much more accurate model of the total delay introduced by a transaction between the transmitting node A and receiving node B:

$$\begin{aligned} T(A,B) = P_S \cdot \left(16 \cdot T_{transfer\_BIT} + \frac{1}{DTR}\right) + \\ + T_{PLL} + T_{CCA} + SFD\_SEQ\_SIZE \cdot \frac{1}{DTR} + L_{RX} \end{aligned} \tag{11}$$

In the model described by (11), most of the components do not introduce any unpredictability to the systems, thus they are perfectly calculable. Two of the parameters in (11) need to be further analyzed, though. The T_{PLL} time component describes the time needed to fully recalibrate the PLL, in a worst case scenario.

In the same manner, using (4), (6), (7) and (8), we can obtain a much more accurate and simplified version of the time model described in (2), which refers to the total delay introduced by the radio module of the transmitting node A with respect to the real-time operating system running on the node:

$$\begin{aligned} T(A) = P_S \cdot \left(8 \cdot T_{transfer\_BIT} + \frac{1}{DTR}\right) + \\ + 2 \cdot T_{PLL} + T_{CCA} + SFD\_SEQ\_SIZE \cdot \frac{1}{DTR} \end{aligned} \tag{12}$$

The same method can be used to describe the case of the receiving node B, by applying (4) and (10) in (3):

$$T(B) = P_S \cdot \left(\frac{1}{DTR} + 8 \cdot T_{tramsfer\_BIT}\right) \tag{13}$$

V. Experimental Results and Model Validation

A set of experiments have been conducted to validate the time analysis and measurement framework presented above. The results were collected from measuring the communication delays using two platforms as case studies.

The measurement framework used to collect the results and the method used to identify the time parameters is described in [30] which extends this work. The measurement framework has as its main component an Olimex LPC-H2294 board [31] with an ARM7TDMI architecture based microcontroller [32]. The main reason this microcontroller is used is that it has numerous capture channels and interrupt sources. Another important aspect is that the two dedicated timers of LPC2294 were synchronized with an error of 1 tick (62.82 ns). We have also considered the interrupt latency which is 12 clock cycles for the fast interrupt (FIQ) and 25 clock cycles for ordinary interrupt (IRQ) [33]. The timer input captures interrupts which are mapped as ordinary interrupts and the additional 4 external interrupts are mapped on the FIQ channel. In a simplified situation we can assume a worst case situation were both interrupt types have a maximum latency of 25 clock cycles. Giving the fact that the core is clocked at 58.9824 MHz we can calculate the interrupt latency as a maximum of 424 ns. In a worst case scenario, the maximum capture error of an event is no more than 486 ns which we consider to be negligible.

The first platform is represented by two modules of the

CrossBow MicaZ hardware platform [34] which consists of a host processor and a ChipCon CC2420 wireless module [35]. This platform usually has as its main software component the TinyOS operating system, but within these measurements this system was not used. A low level driver for the wireless module was used instead. The second platform is represented by two Olimex LPC-H2294 boards with a ChipCon CC2500 [36] wireless module attached to it. In each case the platforms were programmed to generate the necessary events needed by the measurement platform.

For both platforms a custom firmware with the necessary low level drivers was developed in order not only to control the radio module but also to generate the needed events via the GPIO system. The firmware of the two platforms was designed to control the whole node from a local PC through an AT command protocol.

To properly validate the time components presented earlier, a significant number of measurements were needed. For instance, in the case of the CC2420 platform the maximum packet size was considered 64 bytes and were performed 600 transmissions and receptions. Giving the fact that 10 time components were measured in each case, the total number of measurements for this platform is about 384000. In the case of the CC2500 platform the total number of measurements reaches 200000. The packet delays were measured in various conditions: high and low transmit gain, low noise medium, high noise medium, etc.

Noise has also been introduced by generating wireless traffic on adjacent channels, using other similar radio modules. For example, we configured the platforms to operate on ZigBee channel B and used XBee modules [37], configured on ZigBee channel C, to generate traffic noise.

We present in Fig. 1 the experimental analysis of time component described in (9) that was measured using the CC2420 platform. This practically represents the measured latency of the wireless module. The same time component is displayed in Fig. 2 with data gathered from the measurement using the CC2500 platform. Using these plots one can identify the behavior of the internal latency of the module.

In Fig. 3 we present the measurement of another important time parameter, i.e. the time needed by the wireless module to switch to transmitting mode, defined in (5). It was measured using CC2500 the platform. The same component relative to the CC2420 platform is presented in Fig. 4. Using these measurements we can identify a normal behavior of this time component. However, in a real-time environment one has to consider this component having a worst case value which according to many wireless module datasheets is equal to the time needed for PLL recalibration.

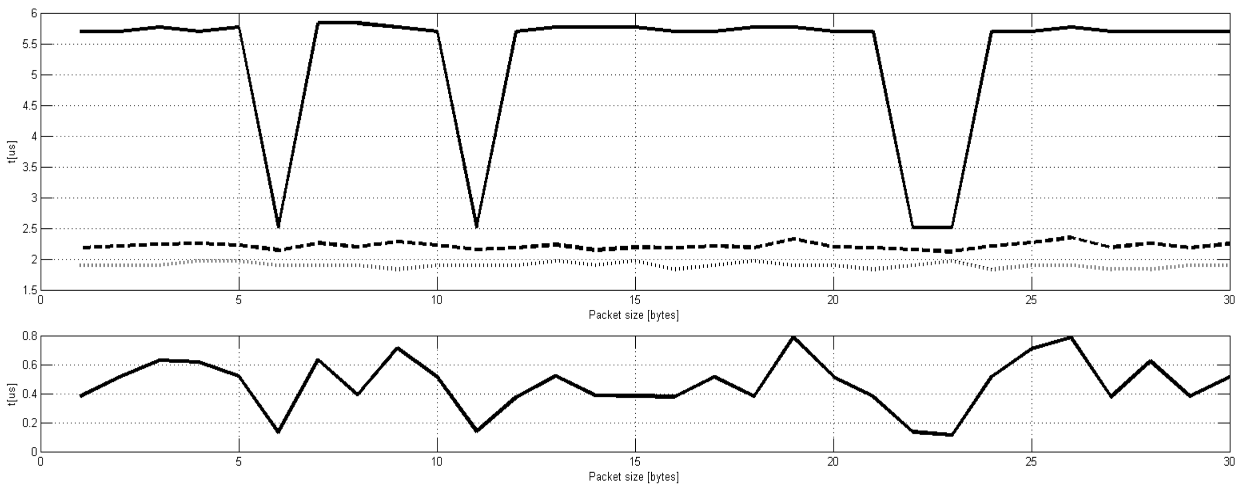

Figure 1. CC2420 based platform – Wireless module internal latency. a) Maximum, average and minimum values, b) Sigma values

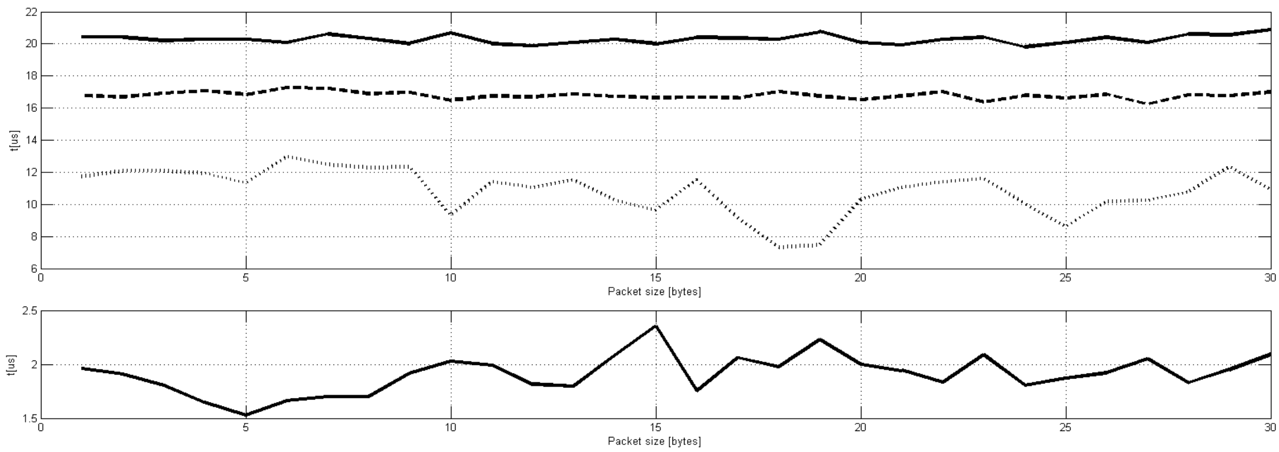

Figure 2. CC2500 based platform – Wireless module internal latency. a) Maximum, average and minimum values, b) Sigma values

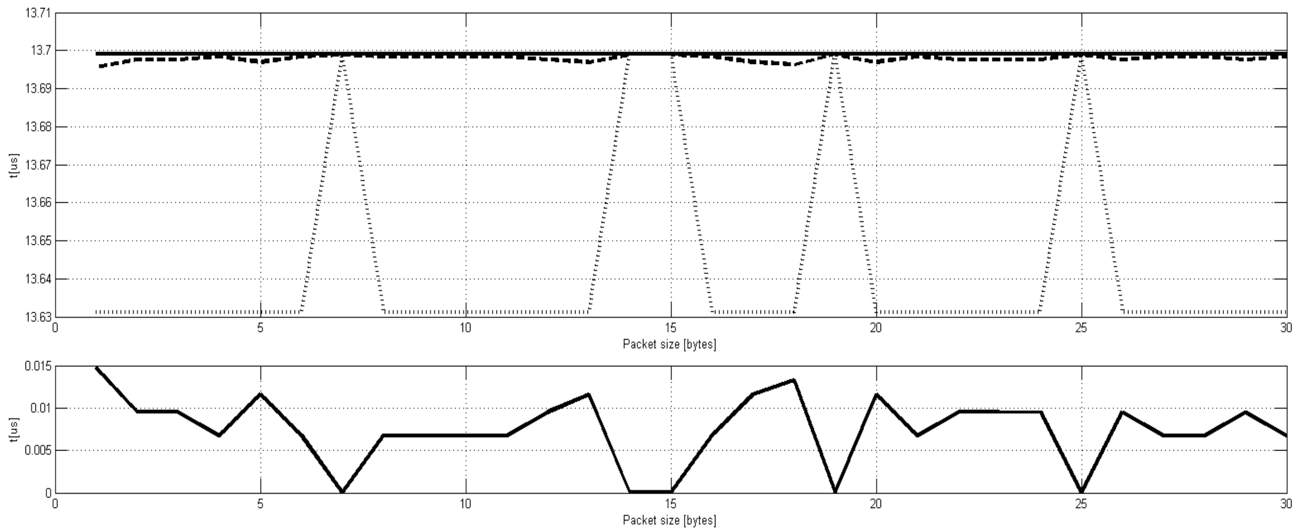

Figure 3. CC2420 based platform – Delay generated when swithing to transmit mode. a) Maximum, average and minimum values, b) Sigma values

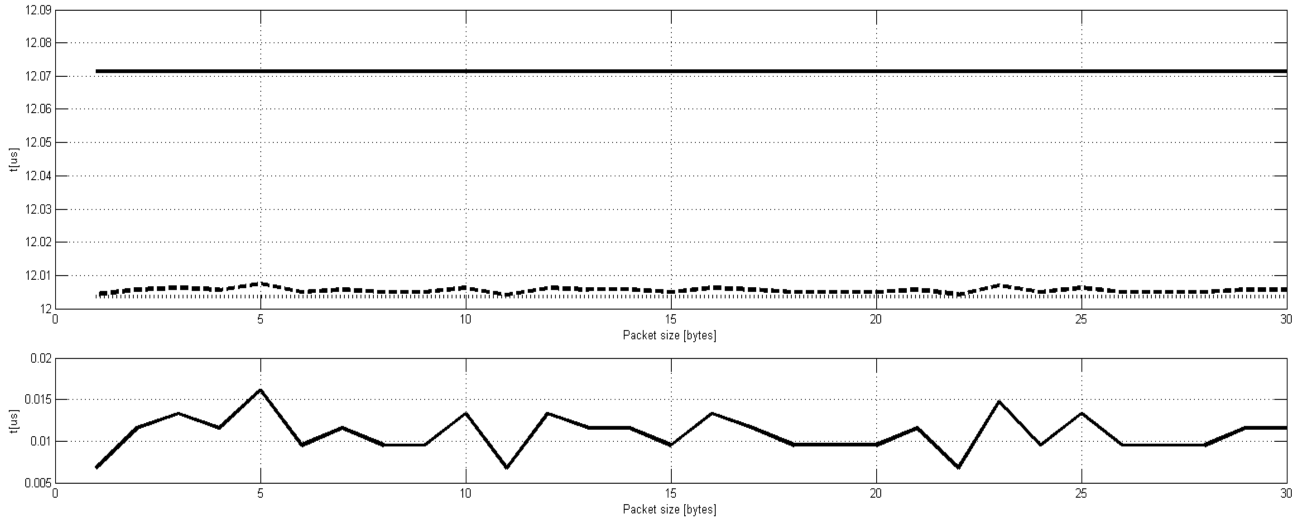

Figure 4. CC2500 based platform – Delay generated when swithing to transmit mode. a) Maximum, average and minimum values, b) Sigma values

VI. Conclusion

In this paper we have presented a model that cat be used to identify, calculate and approximate the components of the total delay introduced by a wireless communication at MAC level. Furthermore, this model can offer a way to calculate the total delay introduced by a radio module when transmitting or receiving data in order to properly design a driver suitable for a real time operating environment. From a more practical point of view this model is suitable for making a time analysis of a chosen radio module in order to determine its time behavior in many situations and thus deciding whether it can or cannot be used in a real time application.

References

[1] G. C. Butazzo, Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications (Real-Time Systems Series). Pavia, Italy: Springer, 2005, ISBN: 978-0387231372, doi:10.1007/0-387-27578-9.

[2] G. Butazzo and P. Gai, "Efficient EDF Implementation for Small Embedded Systems," in Workshop on Operating Systems Platforms for Embedded Real-Time applications OSPERT, Dresden, Germany, 2006.

[3] M. V. Micea, V. Cretu, and V. Groza, "Maximum Predictability in Signal Interactions With HARETICK Kernel," IEEE Transactions on Instrumentation and Measurement, vol. 55, no. 4, pp. 1317 - 1330, August 2006. doi:10.1109/TIM.2006.876530.

[4] M. A. Rivas and M. G. Harbour, "MaRTE OS: Overview and Linux Version," Talk for the Real-Time Systems Group, York Nov 2004.

[5] R. Inam, J. Maki-Turja, M. Sjodin, S. M. H. Ashjaei, and S. Afshar, "Support for hierarchical scheduling in FreeRTOS," in Emerging Technologies & Factory Automation (ETFA), 2011 IEEE 16th Conference on, 2011, pp. 1-10. doi:10.1109/ETFA.2011.6059016.

[6] K. K. Z. Teng, "A Survey on Real-Time MAC Protocols in Wireless Sensor Networks," Communications and Network, vol. 2, no. 2, pp. 104-112, 2010. doi:10.4236/cn.2010.22017.

[7] L. B. Ruiz, J. M. Nogueira, and A. A. F. Loureiro, "Sensor Network Management," in Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems, M. Ilyas and I. Mahgoub, Eds., ed Boca Raton, Florida: CRC Press, 2005, ISBN: 0-8493-1968-4, pp. 57-84.

[8] J. V. Capella, A. Bonastre, J. J. Serrano, and R. Ors, "A pollution monitoring system based on an energy efficient Wireless Sensor Networks architecture," in 17th Telecommunications forum TELFOR, Serbia, Belgrade, 2009, pp. 1145-1148.

[9] M. Xufei, M. Xin, H. Yuan, L. Xiang-Yang, and L. Yunhao, "CitySee: Urban CO2 monitoring with sensors," in INFOCOM, 2012 Proceedings IEEE, 2012, pp. 1611-1619. doi:10.1109/INFCOM.2012.6195530.

[10] M. V. Micea, G. N. Carstoiu, L. Ungurean, and D. Chiciudean, "PARSECS: A Predictable Data Communication System for Smart Sensors and Hard Real-Time Applications," IEEE Transactions on Instrumentation and Measurement, vol. 59, no. 11, pp. 2968-2981, November 2010. doi:10.1109/TIM.2010.2046363.

[11] H. Pei, X. Li, S. Soltani, M. W. Mutka, and X. Ning, "The Evolution of MAC Protocols in Wireless Sensor Networks: A Survey," Communications Surveys & Tutorials, IEEE, vol. 15, no. 1, pp. 101-120, 2013. doi:10.1109/SURV.2012.040412.00105.

[12] C. Deji, M. Nixon, H. Song, A. K. Mok, and Z. Xiuming, "WirelessHART and IEEE 802.15.4e," in Industrial Technology (ICIT), 2014 IEEE International Conference on, 2014, pp. 760-765. doi:10.1109/ICIT.2014.6895027.

[13] J. V. Capella, A. Bonastre, J. J. Serrano, and R. Ors, "A new robust, energy-efficient and scalable wireless sensor networks architecture applied to a wireless fire detection system," in International Conference on Wireless Networks and Information Systems WNIS, 2009, pp. 395-398. doi:10.1109/WNIS.2009.104.

[14] P. N. N. Reddy, P. I. Basarkod, and S. S. Manvi, "Wireless Sensor Network based Fire Monitoring and Extinguishing System in Real Time Environment," International Journal Advanced Networking And Application, vol. 3, no. 2, pp. 1070-1075, 2011.

[15] M. J. Markowski and A. S. Sethi, "Wireless MAC Protocols for Real-Time Battlefield Communications," in Military Communications Conference MILCOM, Monterey, Canada, 1997. doi:10.1109/MILCOM.1997.645033.

[16] P. Rezayat, M. Mahdavi, M. GhasemZadeh, and M. AghaSarram, "A novel real-time routing protocol in wireless sensor networks," in Current Trends in Information Technology CTIT International Conference on the, 2009, pp. 1-6. doi:10.1109/CTIT.2009.5423117.

[17] Li-Ming He, "A Novel Real-Time Routing Protocol for Wireless Sensor Networks," in 10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing SNPD, 2009, pp. 411-416. doi:10.1109/SNPD.2009.110.

[18] Y. Sun, S. Du, O. Gurewitz, and D. B. Johnson, "DW-MAC: A Low Latency, Energy Efficient Demand-Wakeup MAC Protocol for Wireless Sensor Networks," in 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing MOBIHOC, 2008, pp. 53-62. doi:10.1145/1374618.1374627.

[19] F. J. Atero, J. J. Vinagre, E. Morgado, and M. R. Wilby, "A Low Energy and Adaptive Architecture for Efficient Routing and Robust Mobility Management in Wireless Sensor Networks," in 31st International Conference on Distributed Computing Systems Workshops ICDCSW, Minneapolis, Minnesota USA, 2011, pp. 172-181. doi:10.1109/ICDCSW.2011.39.

[20] B. Fateh and M. Govindarasu, "Energy-Aware Adaptive MAC Protocol for Real-Time Sensor Networks," in Communications (ICC), 2011 IEEE International Conference on, Kyoto, Japan, 2011, pp. 1-5. doi:10.1109/icc.2011.5962413.

[21] G. S. A. Kumar, G. Manimaran, and Z. Wang, "End-to-End Energy Management in Networked Real-Time Embedded Systems," Parallel and Distributed Systems, IEEE Transactions on, vol. 19, no. 11, pp. 1498-1510, Nov 2008. doi:10.1109/TPDS.2008.124.

[22] IEEE Computer Society, "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless personal Area Networks (LR-WPANs)," in 802.15.4d, ed: Institute of Electrical and Electronics Engineers, Inc., 2009.

[23] G. Anastasi, M. Conti, and M. D. Francesco, "The MAC unreliability problem in IEEE 802.15.4 wireless sensor networks," presented at the Proceedings of the 12th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems, Tenerife, Canary Islands, Spain, 2009. doi:10.1145/1641804.1641839.

[24] G. Anastasi, M. Conti, and M. Di Francesco, "A Comprehensive Analysis of the MAC Unreliability Problem in IEEE 802.15.4 Wireless Sensor Networks," Industrial Informatics, IEEE Transactions on, vol. 7, no. 1, pp. 52-65, 2011. doi:10.1109/TII.2010.2085440.

[25] Y. Seong-Eun, C. Poh Kit, K. Daeyoung, D. Yoonmee, P. Minh-Long, C. Eunchang, and H. Jaedoo, "Guaranteeing Real-Time Services for Industrial Wireless Sensor Networks With IEEE 802.15.4," Industrial Electronics, IEEE Transactions on, vol. 57, no. 11, pp. 3868-3876, 2010. doi:10.1109/TIE.2010.2040630.

[26] A. Ali, L. A. Latiff, M. A. Sarijari, and N. Fisal, "Real-time Routing in Wireless Sensor Networks," in 28th International Conference on Distributed Computing Systems Workshops ICDCS, 2008, pp. 114-119. doi:10.1109/ICDCS.Workshops.2008.70.

[27] 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," in Eighth IEEE Real-Time and Embedded Technology and Applications Symposium RTAS, Proceedings. Eighth IEEE, 2002, pp. 55 - 66. doi:10.1109/RTTAS.2002.1137381.

[28] D. Abdeli, S. Zelit, and S. Moussaoui, "RTH-MAC: A real time hybrid MAC protocol for WSN," in Programming and Systems (ISPS), 2013 11th International Symposium on, 2013, pp. 153-162. doi:10.1109/ISPS.2013.6581481.

[29] S. Lohier, A. Rachedi, E. Livolant, and I. Salhi, "Wireless Sensor Network simulators relevance compared to a real IEEE 802.15.4 Testbed," in Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International, 2011, pp. 1347-1352. doi:10.1109/IWCMC.2011.5982734.

[30] V. Stangaciu, M. V. Micea, and V. Cretu, "Low-Level Communicaton Time Analysis in Real-Time Wireless Sensor Networks," presented at the IEEE International Workshop on Robotic and Sensors Environments (ROSE), Timisoara, Romania, 2014. doi:10.1109/ROSE.2014.6952988.

[31] Olimex. (2005). LPC-H2294 Get Started Guide. Available: https://www.olimex.com/Products/ARM/NXP/LPC-H2294/resources/H2294.chm

[32] NXP Semiconductors, "LPC2119/2129/2194/2292/2294 User Manual," Koninklijke Philips Electronics N.VMay 2004.

[33] T. Martin, The Insider's Guide To The Philips ARM7-Based Microcontrollers, An Engineer's Introduction to the LPC2100 Series. UK: Hitex Ltd., 2007, ISBN: 0-9549988 6.

[34] Crossbow, "MPR-MIB Users Manual," Crossbow Technology Inc, Revision B 2006.

[35] ChipCon Products from Texas Instruments, "CC2420 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver," Texas Instruments 2012.

[36] ChipCon Products from Texas Instruments, "CC2500 Low-Cost Low-Power 2.4 GHz RF Transceiver," Texas Instruments 2011.

[37] Digi International, "XBeeTM Series 2 OEM RF Modules Product manual v1.x.2x - ZigBee Protocol," Digi International Inc. 2007.