Improving NS-2 Network Simulator for IEEE 802.15.4 standard operation *

André Guerreiro, Jeferson L. R. Souza, and José Rufino

University of Lisbon - Faculty of Sciences
Large-Scale Informatics System Lab. (LaSIGE)
aguerreiro@lasige.di.fc.ul.pt, jsouza@lasige.di.fc.ul.pt, ruf@di.fc.ul.pt

Abstract The IEEE 802.15.4 standard was designed to support the specification of wireless sensor networks (WSNs) and wireless sensor and actuator networks (WSANs), which the utilization is emerging within environments with real-time requirements such as industrial and aerospace. The network simulator NS-2 supports the test, simulation and evaluation of such type of networks, although the real-time support offered by the standard is not yet available in the NS-2 release. This paper presents improvements in the IEEE 802.15.4 NS-2 module to provide a better support for the emulation of networks with real-time requirements, through the incorporation of the contention free period (CFP) and of guaranteed time slot (GTS) defined within the IEEE 802.15.4 module present in the NS-2. Additionally, we also complement this module with IEEE 802.15.4 standard management operations not implemented in the official NS-2 release.

Keywords: NS-2 simulator; wireless sensor and actuator networks; wireless communications; real-time.

1 Introduction

Wireless networking technology has experienced a thriving development in recent years. Due to their unique features such as reduced size, weight, cost, power consumption and mainly the absence of cables, there is a huge interest in developing applications that use wireless sensor networks (WSNs) and wireless sensor and actuator networks (WSANs) in different sectors such as natural resources monitoring [13], aerospace [18], vehicular [9], and industrial [8]. Most of these environments have real-time communication constraints, which implies that WSNs and WSANs must be capable to provide support on real-time communication services.

Several studies have been focused in evaluating wireless technologies for supporting reliable and real-time communication services [15,20,21,18], although

* This work was partially supported: by the EC, through project IST-FP7-STREP-288195 (KARYON); by FCT/DAAD, through the transnational cooperation project PROPHECY; and by FCT, through the project PTDC/EEI-SCR/3200/2012 (READAPT), the Multiannual Funding Program, and the Individual Doctoral Grant SFRH/BD/45270/2008.

dependability and safety of wireless communications are not yet addressed properly [14]. A recent study presents a *Mediator Layer*[16] approach, which is capable to enhance the dependability and timeliness of medium access control (MAC) sublayer. This study represents a motivation for the current research on the test and evaluation of WSNs and WSANs through network simulators, where we take the IEEE 802.15.4 network standard as a case study.

The use of network simulators is a suitable way to test and evaluate network behaviours using several environmental conditions that can be represented by different environment models and setups. There are several network simulators available [11], some with commercial license, such as OMNeT++ [4] and OPNET [5], and others with open source or academic license, like Prowler [6], TOSSIM [12], NS-3 [3] and NS-2 [1]. NS-3 is the next generation and the evolution of the NS-2 simulator, but does not yet support the IEEE 802.15.4 standard. In the presence of such limitation in the NS-3 we choose the NS-2, which is a widely accepted and used network simulation tool, being open source and modular, supporting the simulation of WSNs and WSANs through the IEEE 802.15.4 standard [10].

However the NS-2 IEEE 802.15.4 module [19] does not have a native support for features that address real-time aspects of communications, such as emulation of a contention free period (CFP) where time slots can be allocated for exclusive access to the network. This paper presents improvements in the IEEE 802.15.4 NS-2 module to provide a better support for the test, simulation and evaluation of IEEE 802.15.4 networks with real-time requirements. We include all the management functions needed to support the use of guaranteed time slots (GTS) for frame transmissions, adapting and extending an implementation of a CFP module proposed by [7]. It is our objective to overcome the existing limitation which, natively, only allow contention-based communications in the 802.15.4 NS-2 module. We evaluate and validate our implementation through test cases that uses different network loads, and performance metrics such as delivery ratio, latency, and energy consumption, allowing a better characterization of IEEE 802.15.4 networks in the support of real-time communications.

To present our contributions, this paper is organized as follows: Section 2 presents an overview of the IEEE 802.15.4 standard. Section 3 presents an overview of the NS-2 and its IEEE 802.15.4 module. Section 4 describes our enhancements in the IEEE 802.15.4 NS-2 module. Section 5 presents, compares, and discusses the results obtained from different network load patterns and GTS utilization through the simulation work. Finally, section 6 draws some conclusions and future research directions.

2 IEEE 802.15.4 overview

The IEEE 802.15.4 specification [17] is a standard that allows the creation of wireless networks, being more specifically oriented for the creation of WSNs and WSANs. Each IEEE 802.15.4 network has a special node dubbed network coordinator, which defines a set of characteristics of the network such as addressing,

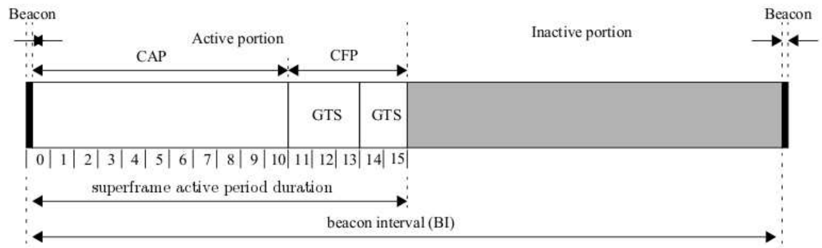

Figure 1: Superframe structure [17]

supported channels, and operation mode. The network can operate either in a beacon-enabled mode or in a nonbeacon-enabled mode. In the beaconless mode, the protocol is essentially a simple Carrier Sense Multiple Access with collision avoidance (CSMA-CA) protocol. Since most of the unique features of IEEE 802.15.4 are in the beacon-enabled mode, like support for communications with real-time restrictions we will focus our attention on this mode. In the beacon-enabled mode the network coordinator coordinates the access to the network by periodically transmitting a special frame dubbed Beacon, which delimits the structure dubbed superframe that specifies the intrinsic rules to perform such access. The period that specifies the consecutive beacon transmissions is dubbed beacon interval (BI).

The superframe structure depicted in Fig. 1 may comprise two periods: a mandatory active period, and an optional inactive period. Each active period is divided into a contention access period (CAP), and an optional contention free period (CFP). The CAP was designed for general purpose traffic, using a contention-based approach in the access of the network. The CFP was designed to support real-time traffic, being divided in transmission windows dubbed guaranteed time slots (GTSs) that uses an exclusive and contention-free approach in the access of the network. Once a given GTS slot is allocated to a node, only this node can transmit in this time interval. Finally, the inactive period was designed to power-saving purposes, where all nodes use such period to save the energy spent in the listen process.

3 IEEE 802.15.4 in NS-2 Simulator

The network simulator NS-2 is a discrete event simulator developed in a collaborative effort by many institutions, containing contributions from different researchers [2]. As a discrete-event simulator, all actions in NS-2 are associated with events rather than time. NS-2 consists of two key languages: C++ and Object-oriented Tool Command Language (OTcl). While the C++ defines the internal mechanism (i.e., a backend) of the simulation objects, the OTcl sets up

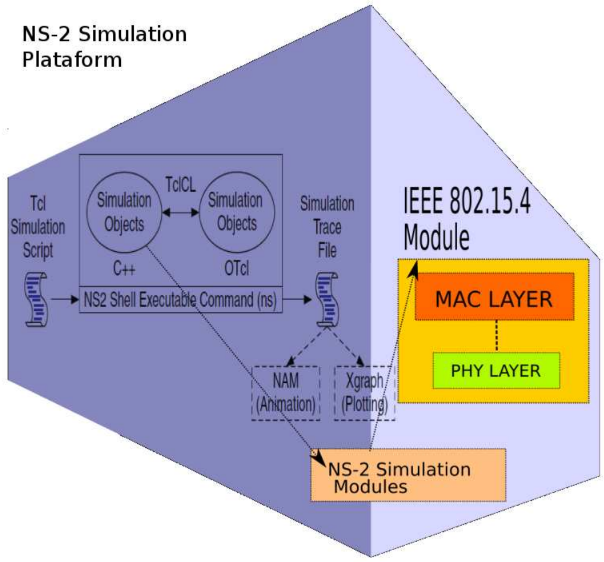

Figure 2: NS-2 802.15.4 module architecture.

simulation by assembling and configuring the objects as well as scheduling discrete events (i.e., a frontend). The C++ and the OTcl are linked together using TclCL as illustrated in the rightmost part of Fig. 2. Mapped to a C++ object, variables in the OTcl domains are sometimes referred to as handles. Conceptually, a handle (e.g., n as a node handle) is just a string in the OTcl domain, and does not contain any functionality. Instead, the functionality (e.g., receiving a packet) is defined in the mapped C++ object (e.g., of class Connector). In the OTcl domain, a handle acts as a frontend which interacts with users and other OTcl objects. It is possible to define its own procedures and variables to facilitate the interaction. The member procedures and variables in the OTcl domain are called instance procedures. The basic architecture is represented in Fig. 2.

The IEEE 802.15.4 NS-2 module is provided in the form of methods of each layer specified in the IEEE 802.15.4 standard [10]. The module came with different functionalities, and support different network topologies (star and point-to-point), two types of operation (beacon and non-beacon enabled), and basic MAC management actions such as Association, Channel Scan, energy model, etc. However, the communication during CFP is not implemented in a modular way in the current IEEE 802.15.4 NS-2 module. The absence of the GTS mechanism is a major drawback once is fundamental for real-time WSAN applications, allowing a node to operate on the channel within a portion of the superframe that is dedicated exclusively to it. **IEEE 802.15.4 MAC:** The MAC sub-layer handles all access to the physical radio channel and is responsible for tasks such as network synchronization based on beacons, network association and disassociation, data security, network access, and handling and maintaining the GTS mechanism. The current IEEE 802.15.4 module within the NS-2 simulator has some differences in the behaviour regarding the implementation of the standard MAC management actions. Actions needed for the support of real-time data transmissions such as GTS allocation and deallocation are not implemented. Additionally, other management actions such as disassociation and network ID conflict notification, and all auxiliary mechanisms needed to support the execution of such actions are not implemented and should be developed and incorporated in the IEEE 802.15.4 module, enhancing the compliance with the IEEE 802.15.4 standard.

4 Improvements in the 802.15.4 NS-2 Simulator module

In order to achieve a better real-time support from the 802.15.4 simulation module in NS-2, we extend the existent module to provide the GTS mechanism for network nodes. Adapting the CFP implementation in [7], to the latest version of the simulator used in this study, revealed some issues and incompatibilities. The referenced implementation proposed in [7] prevents the use of MAC management commands such as Orphan Notification and Coordinator Realignment. In our implementation this was corrected and the GTS can be activated from the NS-2 script without affecting other MAC services. A description of the implemented GTS mechanism and enhanced MAC management operations is given below.

4.1 CFP and GTS implementation in NS-2

Data can be transmitted in three ways, *Direct transmission* which means that data is sent during the CAP. *Indirect transmission* is only available for coordinators, data is placed on the indirect transmission queue and is sent during the CAP when polled. And finally *GTS transmission*, which requires that a node has to use a GTS slot to transmit its data. To allow this, a GTS slot is allocated to the node for the specified data transmission. Although a data transmission

Algorithm 1 Transmission Data Frame using GTS

1: Begin.
2: $MAC.Data.Send.Request(data)$;
3: **when** allocated GTS is reached **do**
4: $\quad MAC.Data.transmit(data)$;
5: **end when**
6: End.

request can occur at any-time in a superframe, a data transmission request using a GTS is required to transmit the data only during the allocated GTS. Therefore, in our implementation described in Algorithm:1, it is checked if the data

Algorithm 2 Coordinator processing a GTS request command

1: Begin.
2: $MAC.Mgmt.GTS.Request(node_{addr}, nr\_slots)$;
3: **if** nr_slots are available **then**
4: $MAC.Mgmt.GTS.allocate(node_{addr}, nr\_slots)$;
5: $MAC.Mgmt.GTS.updateGTSList(node_{addr})$;
6: **else**
7: $MAC.Mgmt.GTS.Response(slots\_not\_available)$;
8: **end if**
9: End.

transmission request using a GTS is in the allocated GTS duration or not, as represent on line 3. Since the procedure to check the remaining GTS time is also implemented, multiple data can be transmitted during a GTS, which complies with the IEEE 802.15.4 standard. When a coordinator receives a GTS request command from a node, willing to transmit data, Algorithm:2 is executed by the coordinator. After checking if the node GTS slot is valid (line 3), which means the $node_{addr}$ is already known by the coordinator and nr_slots are available, the allocation is made (line 4). If the operation is successfully concluded, the final CAP slot is updated as in line 5, and the updated beacon is sent. If all the GTS slots are occupied at the time, an information regarding $slots\_not\_available$ is sent to the node, as described in line 7. The information from the GTS allocation or deallocation is delivered in the next beacon frame to the nodes that sent the GTS request command, letting them know the result of the requesting process.

4.2 Enhancing MAC management actions according to the IEEE 802.15.4 standard

The NS-2 simulator module has some differences in the behaviour regarding the implementation of IEEE 802.15.4 standard MAC management actions. Taking this in consideration we added the functions presented in Fig. 3. Some operations are implemented in the original module but they are not fully functional. We corrected them and implemented other additional and needed operations. The Coordinator conflict is one of those such operations. Coordinator conflict occurs when more than one coordinator is active within the same network. By default, each network has an identifier, the source identifier, which identifies the network uniquely and is used by the coordinator in beacon transmissions. If some other (possibly old) coordinator enters the network operational space, e.g., after having been away from some period of time, the network may have two different coordinators transmitting beacons with the same source identifier.

The resolution in turn will request the MAC layer to perform an active scan. This scan is realized in all currently used logical channels. If the protocol management entities decide that the node was orphaned, a request is issued to the MAC layer to start an orphan scan recovery action, over a specified set of logical channels. For each logical channel: a MAC orphan notification command is sent; as reply, a MAC realignment command from the previously associated coordinator, is awaited for during a given period. Once such MAC command is received the node terminates the scan and the network becomes accessible. At the coordinator the need to assist MAC layer management actions starts when a MAC orphan notification command is received. Upon processing by protocol management entities, the acknowledged transmission of a MAC realignment command is requested. Relatively to the channel scan process carried out by the different nodes, they should be able to scan all channels available to either associate to a network or to establish one. However, since we aim to simulate a network operating in 2.4 Ghz we removed the limitation of scanning only the first 3 channels. This limitation was removed and the scan of all the 16 channels defined by the standard is now allowed. The duration of each scan was also incorrect, once its parametrization was not set in compliance with the standard. Again, the issue was corrected.

MAC Manegement Action	IEE 802.15.4 Standard Behaviour	NS-2 Original Module	NS-2 implemented module
Orphan	A request is issued to the MAC layer to start an orphan scan recovery action	Not Functional	Operational
Coordinator Realignment	On the reception of Orphan notific. is required an acknowledged transmission of a realignment command	Not Functional	Operational
Coordinator conflict	If two coordinators establish a network with the same Network identifier, a Coordinator conflict occurs	Not implemented	Implemented
Channels Available to Scan	16 channels on 2.4Ghz	Only the first 3 channels	16 channels
Scan Duration atribute	$aBaseSuperframeDuration \times (2^n + 1)$, where n is the value of the $ScanDuration$ parameter.	incorrect definition	in compliance with the standard
Network Information Base (NIB) attribute	The Management Entity checks to see if the NIB attribute is a MAC or a PHY layer attribute.	This verification is not performed	in compliance with the standard

Figure 3: NS-2 IEEE 802.15.4 Module behaviour comparison

5 Results

We now describe our evaluation metrics, the simulation setup and finally the results achieved in improving the IEEE 802.15.4 MAC NS-2 module.

5.1 Evaluation Metrics

The evaluation metrics that we used are applied to the network, taking into consideration all the nodes involved in the simulation. The metrics are the following:

– *Data frame delivery Ratio (DFDR)*, which is the ratio between the total number of frames received in MAC sub-layer and the total number of data frame transmit requests during the simulation period. In our simulation we consider the data frames transmit requests issued by all the nodes but the coordinator.

$$DFDR = \frac{Total\ Data\ frames\ received \times 100}{Total\ Data\ frames\ transmit\ requests} \quad (1)$$

– *Latency*, which represents the transfer time of a data frame to a one-hop neighbour. For each individual data frame transfer the frame transfer latency represents the interval between the data frame reception instant (T_{rxData}) and the instant the corresponding data frame transmit request is issued (T_{txData}). This frame includes the data frame processing and queueing time at the nodes, the data frame transmission time and the back off interval (if applicable). The average latency can then be calculated over all successful end-to-end transmissions within the simulation run.

$$AverageLatency = \frac{\sum_{allreceivedframes}(T_{rxData} - T_{txData})}{Total\ number\ of\ received\ frames} \quad (2)$$

On the other hand, the worst case value is given by:

$$WorstCaseLatency = \max_{allreceivedframes}(T_{rxData} - T_{txData}) \quad (3)$$

– *Energy*, the energy model present in NS-2 is used to calculate the amount of energy consumed by the nodes during the simulation time.

$$Energy\ Used\ per\ Node = \frac{Total\ Energy\ Used}{Number\ of\ Nodes} \quad (4)$$

5.2 Simulation Setup

The characteristics of the simulation setup scenario are shown in Table 1. The star topology network was chosen in this simulation. The network was simulated with seven nodes, where one of these nodes, in the center, was the coordinator. All other nodes are in the radio transmission range of the coordinator. Additionally all nodes are in a single broadcast domain, which means that all the nodes are within the range of each other. To evaluate the network behaviour, the six remaining nodes constantly transmit data frames to the coordinator during CAP or CFP. The traffic generator is set to produce Constant Bit Rate traffic (CBR), which means data frames are transmitted at a constant rate from the nodes to the coordinator. The interval between each data transmission request is successively set to 1, 0.1, 0.01 and 0.001 seconds. Given the packet size of 70 bytes, this means the network load is monotonically increased, adopting the values of 15, 40, 70, 95 kilobytes. The MAC management actions required for node association with its coordinator and the GTS allocation times (if required) are excluded from the evaluation scope in the present simulation run.

Table 1: NS-2 Simulation Parameters

Simulation Parameters	
NS-2 Version	2.35 updated with GTS features
Network Topology	Star Topology
Nodes	7
Traffic	Constant Bit Rate (CBR)
Reception range	15m
Carrier Sense range	15m
Packet Size	70 bytes
CAP Transmission Type	Direct, using CSMA/CA
CFP Transmission Type	GTS transmission
Transmission/Reception Power	30mW
Beacon	Enabled
Beacon Order	3
Superframe Order	3
Maximum CSMA/CA Attempts	4
Simulation Time	600 seconds

5.3 Simulation Results

Figure 4 represents the delivery ratio on the network, providing a comparison of the results achieved for transmissions requests issued during the CAP and CFP periods. During CFP, nodes use the allocated GTS and get direct and exclusive network access, which allows to achieve about 100% delivery rate. In CAP the delivery ratio drops in function of the increase in the network load. This is explained by the occurrence of collisions during CAP, or due to the number of nodes attempting to access the medium. In the CSMA/CA protocol a data frame transmit request is dropped, if the number of transmission attempts exceed a given threshold defined by the Maximum CSMA/CA attempts (Table 1).

Figure 5 shows the latency comparison between a data frame transmission using CFP and CAP. While the latency remains almost constant when data frames are transmitted during CFP, using allocated GTS, the latency highly increases while using CAP. The constancy achieved in data frame transfer times during CFP is a sign of determinism and predictability and shows in Figure 5 in two ways: an (almost) constant worst-case data frame transmission latency; the optimal value of this latency, which does not exceed 0.002936 seconds. This is due to nodes during CFP get exclusive network access, meaning nodes do not have to check if the media is idle and no collisions occur for those nodes. These results show the importance of the GTS mechanism in applications with real-time requirements on which deterministic data frame transmission times are mandatory. Additionally, the data frame transmission latency increases in CAP, up to the worst-case value of 0.010512 seconds, given the worst-case network load in the simulation setup. Finally, Figure 6 represents the average energy consumption by the nodes during the simulation period. It noticed that the

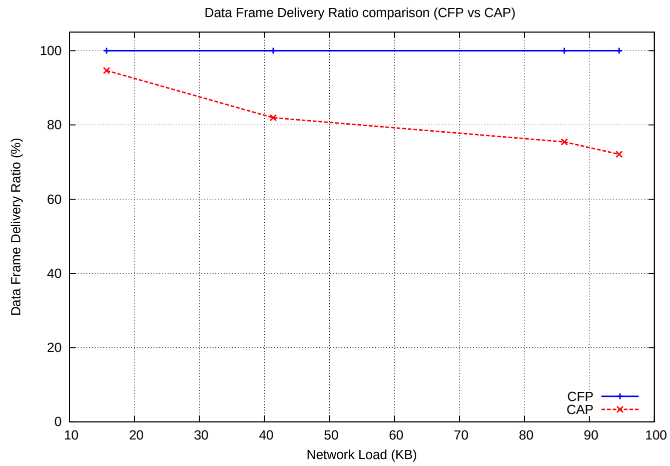

Figure 4: Data Frame Delivery Ratio comparison between transmission during CAP and CFP

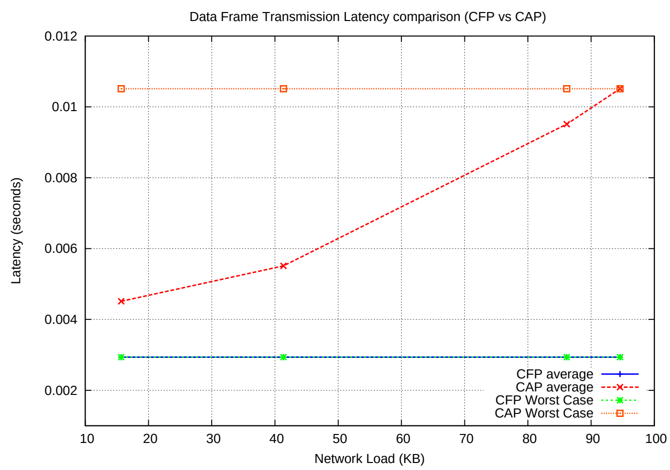

Figure 5: Data frame transmission Latency comparison between transmission during CAP and CFP

energy consumption increases when using CFP in comparison with CAP as result of required beacon frame reception tracking by the node, an action that obliges the node to switch-on its transceiver during the active period of every superframe instance. Contention-based access is more efficient under light network loads, whereas contention-free access becomes preferable when the background network load increases.

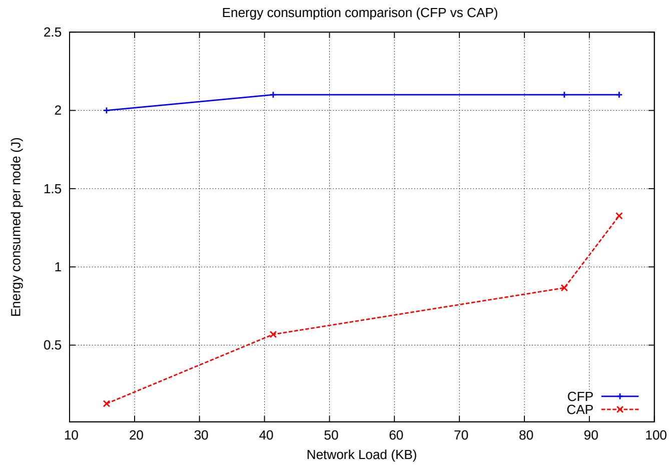

Figure 6: Energy consumption per node for data transmission during CAP and CFP

6 Conclusion and future directions

In this paper the current IEEE 802.15.4 module implemented in NS-2 is modified and extended to include the use of the GTS mechanisms based on the standard. So, the operations of the GTS allocation, use and deallocation are implemented. The addition of unimplemented MAC operations enhanced the simulation module so that is in accordance to the standard.

Based on NS-2 simulations, we evaluate the performance of various features in the IEEE 802.15.4 MAC. We find that data transmission during the CAP reduces energy cost due to idle listening in the backoff period but increases the collision at higher rate and larger number of sources. While the use of GTS in the CFP can allow dedicated bandwidth to a device to ensure low latency, the device need to track the beacon frames in this mode, which increases the energy cost. The addition of available channels to scan during association revealed an increase of the association time an energy cost, but made the NS-2 more compliant to the standard. Having better tools allow us to better understand the temporal aspects, in this case, of IEEE 802.15.4 networks. Thus, the knowledge of the network performance allows a better analysis and definition of a timeliness model, representing a first though crucial step for achieving an effective support to real-time operation in IEEE 802.15.4 networks.

This greatly assists the IEEE 802.15.4 standard related research. To benefit the research community, our NS-2 implementation of this protocol is publicly available online at: `http://www.karyon-project.eu/wp-content/uploads/2012/10/ns-2-2.35-with-gts.tar.gz`.

References

1. Ns-2: The network simulator version 2. http://www.isi.edu/nsnam/ns/
2. Ns-2: The network simulator version 2 contributed code. http://nsnam.isi.edu/nsnam/index.php/Contributed_Code/
3. Ns-3: The network simulator version 3. http://www.nsnam.org/
4. Omnet++ discrete event simulation system. http://www.omnetpp.org/
5. Opnet:application and network performance tool. http://www.opnet.com
6. Prowler: Probabilistic wireless network simulator. http://www.isis.vanderbilt.edu/Projects/nest/prowler/
7. Choi, W., Lee, S.: Implementation of the IEEE 802.15.4 module with CFP in NS-2. Telecommunication Systems (Aug 2011), http://link.springer.com/10.1007/s11235-011-9548-7
8. Han, S., Zhu, X., Mok, A., Chen, D., Nixon, M.: Reliable and real-time communication in industrial wireless mesh networks. In: 17th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS). pp. 3–12 (2011)
9. Hartenstein, H., Laberteaux, K.: A tutorial survey on vehicular ad hoc networks. IEEE Communications Magazine 46(6), 164 –171 (June 2008)
10. IEEE 802.15.4: Part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANs) - IEEE standard 802.15.4. IEEE P802.15 Working Group (2011), Revision of IEEE Standard 802.15.4-2006
11. Korkalainen, M., Sallinen, M., Kärkkäinen, N., Tukeva, P.: Survey of Wireless Sensor Networks Simulation Tools for Demanding Applications. 2009 Fifth International Conference on Networking and Services (2009)
12. Levis, P., Lee, N.: TOSSIM : A Simulator for TinyOS Networks pp. 1–17 (2003)
13. Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., Anderson, J.: Wireless sensor networks for habitat monitoring. Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications - WSNA '02 p. 88 (2002)
14. Åkerberg, J., Gidlund, M., Björkman, M.: Future research challenges in wireless sensor and actuator networks targeting industrial automation. In: 9th IEEE International Conference on Industrial Informatics (INDIN) (July 2011)
15. Shuai, X.Y., Zhang, Z.C.: Research of real-time wireless networks control system MAC protocol. Journal of Networks (April 2010)
16. Souza, J.L.R., Rufino, J.: Towards resilient real-time wireless communications. In: 25th Euromicro Conference on Real-Time Systems (ECRTS-WiP) (July 2013)
17. Standard, I., Society, I.C.: IEEE Standard for Local and metropolitan area networks, Part 15 . 4 : Low-Rate Wireless Personal Area Networks (LR-WPANs) IEEE Computer Society S ponsored by the (2011)
18. Stone, T., Alena, R., Baldwin, J., Wilson, P.: A viable COTS based wireless architecture for spacecraft avionics. In: IEEE Aerospace Conference. pp. 1–11 (2012)
19. Zheng, J., Lee, M.J.: A Comprehensive Performance Study of IEEE 802.15.4, chap. 4, pp. 218–237. IEEE Press, Wiley Interscience (June 2006)
20. Zhou, T., Sharif, H., Hempel, M., Mahasukhon, P., Wang, W., Ma, T.: A novel adaptive distributed cooperative relaying MAC protocol for vehicular networks. IEEE Journal on Sel. Areas in Comm. (Jan 2011)
21. Zhu, X., Han, S., Huang, P.C., Mok, A., Chen, D.: MBStar: A real-time communication protocol for wireless body area networks. In: 23rd ECRTS (July 2011)