Multi-Stack Architecture Implementation to Enhance the QoS in WSN with Prioritization of Packets

Panimozhi K
Assistant Professor,
Dept. of CSE
BMSCE, Bangalore, India

G. Mahadevan, Ph.D
Professor,
Dept. of CSE
AMCEC, Bangalore, India

ABSTRACT

The main aspect in Wireless Sensor Network is to improve the overall performance of the network, to reduce the battery usage, and also to enrich all QoS by using some protocols. Now a it is seen that a single network that is deployed have to handle numerous with varied QoS requirements applications. These aspects can be achieved by using multi-stack architecture which works with various several combinations of MAC & NWK protocols. The combinations of MAC & NWK protocols are adapted in multi-stack mechanism by dividing time into time intervals. At each time interval each combination of the protocols are activated. Simulations mitigated QoS among the combinations of protocols. The queue exchange algorithm used allows frames from one time interval of the node that is to be sent during another time interval at same node is proposed. This eases the complexity of time intervals dimensioning and also improves the overall performance of the network.

General Terms

QoS, MAC protocols, Performance

Keywords

Sensor Networks, Multi-stack architecture, Priority Queuing

1. INTRODUCTION

Wireless Sensor Network (WSN) consists of spatially distributed autonomous sensors in order to monitor physical or environmental conditions such as temperature, sound, pressure etc. and co-operatively pass their data through network to main location. Hence these types of network are widely used in real time and critical real time applications. These networks handling such type of applications have to handle several types of packets with varied constraints.

Quality of service is the level of service provided by the sensor networks to the end users. The researchers define Quality of Service (QoS) for sensor networks as the optimum number of sensors sending information towards information-collecting sinks or a base station. Researchers [1], [2] identified traffic types that can be forwarded efficiently for several protocols. The conclusion shared by several works is that no single protocol can be adapted to all QoS [3].

Wireless Sensor Networks (WSNs) are resource-constrained networks, which are expanding as a result of a wide range of potential applications. However, there are several difficulties in developing energy-efficient, scalable and reliable WSN applications. This is due to the fact that WSNs are constrained in terms of energy, short communication range, low bandwidth, and limited processing and storage. Another typical problem in the wireless network is that all the signals

are treated as same, even if some nodes detect emergency signals they have to wait until all the previous sensed signals reach the base station. One of the main goals of researchers is to reduce the battery usage by designing new medium access control (MAC), network (NWK) protocols and to prioritize the packets.

This topic uses a novel multi-stack architecture where several MAC/NWK combinations are used. The aim is to benefit from several MAC/NWK combinations to leverage QoS management. A synchronized round-robin strategy ensures that at a given time, the same MAC/NWK combination is used by all the nodes of the network. The architecture is enhanced by using Queue scheduling algorithm to benefit the QoS management.

The contributions of this paper are four-fold. First, it shows that Zigbee architecture is compatible for the Multi-stack implementation. Second, an evaluation of queuing mechanism concludes that it can improve the network performance. Third, it is described how packets can be exchanged from one time-interval to another, in order to maximize the usage of each time-interval. Fourth is how the emergency signals are prioritized. The remaining of the paper is organized as follows. Section 2 presents the architecture of Zigbee architecture. Section 3 presents the queuing mechanism used in this paper. Section 4 describes the multi-stack architecture used in this paper. Section 5 gives the evaluation result. Section 6 resumes the advantages of the multi-stack architecture supporting the single network to handle several applications and gives promising perspectives. Finally Section 7 is the conclusion.

2. STUDY OF ZIGBEE ARCHITECTURE

The joint efforts of the IEEE 802.15.4 task group [20] and the Zigbee Alliance [21] have ended up with the specification of a standard protocol stack for Low-Rate Wireless Personal Area Networks (LR-WPANs), a promising technology for Wireless Sensor Networks (WSNs) [22-26]. In what follows, we denote by *Zigbee* the entire IEEE 802.15.4/Zigbee protocol stack. Zigbee is gaining an exponentially increasing interest from industry and is considered as a universal solution for low-cost low-power wirelessly connected monitoring and control devices [24-26]. This interest is mainly driven by the potentially large number of emerging applications including home automation (as the current principal commercial target of the Zigbee Alliance), health care monitoring, industrial automation, environmental monitoring, surveillance, and so on. These applications have essentially been triggered by the wireless sensor network paradigm, which represents the new generation of network infrastructure for large-scale distributed embedded systems.

The reputation of Zigbee, even though not already widely commercially available, is closely related to the objectives for which it was designed [20, 21] and to its flexibility to fit different network and application requirements. While it was designed for low-cost wireless devices (such as wireless sensors), the most important technical features of Zigbee are to provide *low power consumption* and *real-time guarantees*. However, the benefit gained from these features typically depends on the configuration of the Medium Access Control (MAC) sublayer, whether operating in beacon-enabled (with synchronization) or in non beacon-enabled (without synchronization) modes.

Some of the characteristics of ZigBee include:

- Global operation in the 2.4GHz frequency band according to IEEE 802.15.4

- Regional operation varies from country to country for e.g: 915 MHz (Americas) and 868 MHz (Europe).

- Frequency agile solution operating over 16 channels in the 2.4GHz frequency

- Incorporates power saving mechanisms for all device classes

- Discovery mechanism with full application confirmation

- Pairing mechanism with full application confirmation

- Multiple star topology and inter-personal area network (PAN) communication

- Various transmission options including broadcast

- Security key generation mechanism

- Utilizes the industry standard AES-128 security scheme

- Supports Alliance standards (public application profiles) or manufacturer specific profiles

3. QUEUE SCHEDULING MECHANISM

The queue scheduling provides various different services based on application specification and prioritize the data packets. The scheduling queue has a multi-queue based queuing architecture that classifies the data packets according to their application and MAC layer abstraction. To determine the next packet to be served from the multi-queue mechanism within each node, Packet based Multi-priority queue (PBMPQ) scheduling algorithm in order to allocate rate and to select the next serviced packet are used. The multi- queue queuing architecture is given as in Fig. 2. The received packets are classified according to its critical condition i.e. based on the packet type and are stored into the appropriate queue. Thus the number of queues determines the number of network service levels.

The Fig. 1 show the priority based packet assignment to the various queues. The explanation is given below.

The incoming packet priority is determined by the application and the MAC layer abstractions. The application layer abstraction prioritizes the packets based on its content importance. The MAC layer abstraction, on provide fair, efficient network services among self-generated and relayed packets. Packets that have real time critical data go to a higher priority. For example, when the originating packets

are delay constrained packets then the factor of MAC layer abstraction is determined by normalizing with the maximal delay permitted for such type of packets. The queuing architecture consists of 3 queues or more queues with one specified as an *instant queue*, or deterministic queue, meaning that any packet stored in this queue will be instantly served.
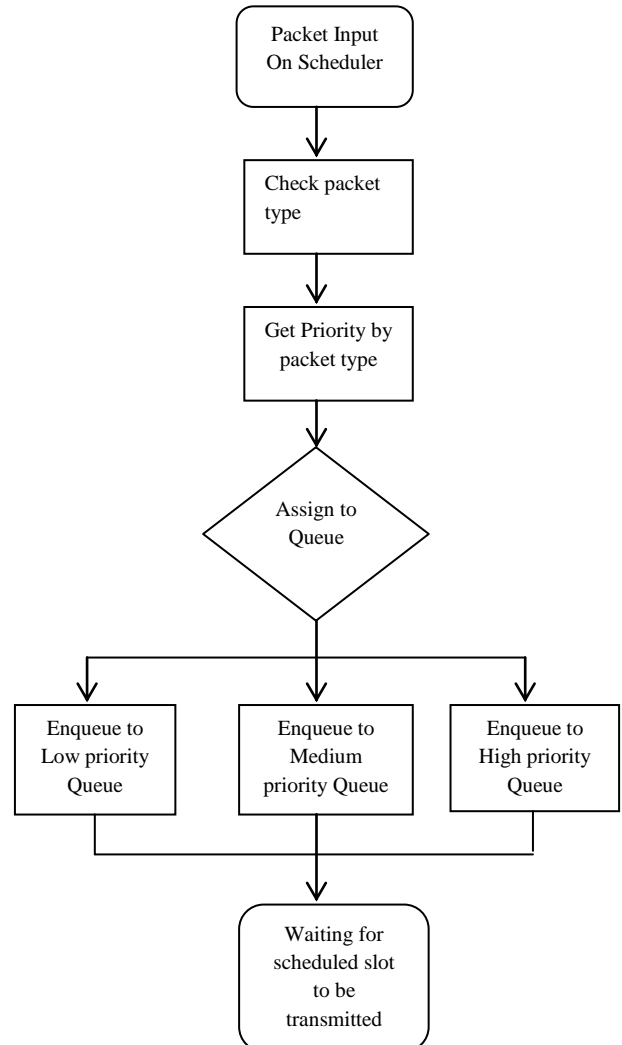

Fig. 1 Scheduler operation in Multi- Queue

Such design allows the allocation of a trapdoor for centralized network management traffic (e.g. network synchronization) and offers extremely urgent traffic with a path for rapid service. The remaining rest of the queues uses Packet based Multi-priority queue (PBMPQ) scheduling algorithm in order to allocate rate and to select the next serviced packet.
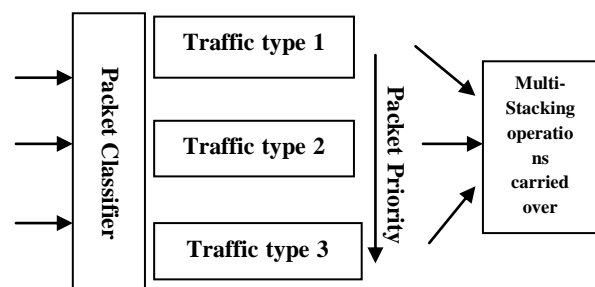

Fig. 2 The multi-queue queuing architecture

The most important function of the MAC layer is that it coordinates and schedules the data transmissions among multiple nodes by sharing the same channel. The packets are immediately dropped when the difference between the sensing time and the current time is beyond this predefined threshold.

The Packet based Multi-priority queue (PBMPQ) scheduling algorithm is given as follows:

Algorithm 1: Packet based Multi-priority queue
Function Queue Assignment (M [ ], E [ ], P [ ], mi, ns)

Begin Function

$\quad$ For all m ∈ M ∧ e ∈ ∧ E ∧ p ∈ P do
$\quad\quad$ If m< e then
$\quad\quad\quad$ Enqueue (e, m)
$\quad\quad$ End if
$\quad\quad$ If e < p then
$\quad\quad\quad$ Enqueue (p, e)
$\quad\quad$ End if
$\quad$ End for
$\quad$ For all p ∈ P do
$\quad\quad$ Sort (p)
$\quad$ End for
$\quad$ For all e ∈ ∧ E do
$\quad\quad$ Sort (e)
$\quad\quad$ Priority (e)
$\quad$ End for
$\quad$ Repeat
$\quad$ E ← MaxprioirtyinQueueHead(P)
$\quad\quad$ m_e ← Sensor Number(e)
$\quad\quad$ p_e ← Packet Number(e)
$\quad\quad$ if $p_{e \leq}$ ns then
$\quad\quad\quad$ AssignAll1By1 (e)
$\quad\quad$ Else
$\quad\quad$ Assign Proportion (e)
$\quad\quad$ End if
$\quad\quad$ mi ← mi- m_e
$\quad\quad$ ns ← ns - p_e
$\quad$ until mi = 0 ∨ ns = 0 ∨ Empty (E)
End function

where M- number of Sensor nodes,

$\quad$ E – List of events,

$\quad$ P- Array containing packets with priority,

$\quad$ mi- number of assignments

$\quad$ ns – time slot for packets

4. STUDY OF MAC AND NWK PROTOCOLS

4.1 CSMA/CA [Carrier Multiple Access/ Collision Avoidance

CSMA/CA are the IEEE 802.15.4 standards [4] that describes the lower layers of a low-power wireless personal area network (LP-WPAN). It defines two types of devices: full-functional devices (FFD) and reduced-functional devices (RFD). IEEE 802.15.4 operates in two modes: the beacon-enabled mode and the non-beacon enabled mode. We focus on the beacon-enabled mode using slotted CSMA/CA that provides larger energy savings. In this beacon-enabled mode, some FFDs broadcast a periodic beacon in order to synchronize the neighboring devices. Generally the throughput is the same under real world conditions due to radio propagation factors

CSMA/CA is a stochastic MAC protocol. A node waits for a random back off and assesses the channel, before transmitting. The node draws a new, larger back off, when the channel is busy.

4.2 Upstream MAC Protocol

Some MAC protocols activate nodes in sequentially in order to reduce the collisions in the network. A schedule is the order in which routers are activated. It has a significant impact on the delay to relay frames. A schedule in which the children nodes are placed before their parents are via an upstream scheduling. In such a scheduling, a frame can be forwarded from any node to the root of the tree in a single global cycle. In order to forward a frame from the root of the tree to a node at depth *l*, *l* global cycles are required.

The technique of the upstream scheduling is used to create a new MAC protocol called: the upstream MAC protocol. In this upstream MAC protocol, time is divided into some *k* intervals, where *k* represents the number of nodes in the tree. Each interval *τi* corresponds to the activation time of node *i*. The nodes are activated in the order of reverse breadth-first search of the tree, which is always an upstream scheduling. In the upstream MAC protocol, all the neighbors of a node will be listening during the activation time of the node. This allows the use of any routing protocol in conjunction with the upstream MAC protocol.

4.3 Hierarchical Tree Routing Protocol

The hierarchical tree routing protocol of ZigBee [27] is referred to as the tree protocol in the following of this paper. Using this protocol, communications follow the links of a tree. When a router at depth d receives a packet, the router checks whether the destination is within its own address space or not. If it is the case, the destination is a descendant of the router: the router computes for which child the packet has to be sent. Otherwise, the router sends the packet to its parent.

The tree protocol allows high energy savings [28]. Indeed, the routing decision can be made without exchanging routing tables between routers. Thus, the control overhead is limited to the tree maintenance. Moreover, the energy overhead is limited: when a router r is active, only its parent and children have to be active, while the other potential neighbors can be inactive. However, the tree protocol produces non-optimal routes in terms of hop count.

4.4 Shortcut Tree Routing Protocol

A node examines its neighbor table to route a packet in the shortcut tree routing [6]. For each neighbor, the node computes the hop-count (on the tree), from this neighbor to the destination of the packet. Then, it chooses as a next hop to the neighbor providing the shortest hop-count (on the tree).
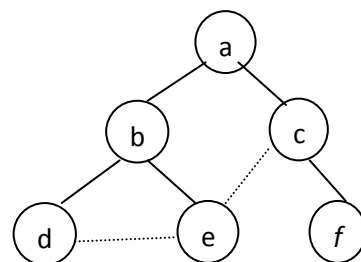
Fig. 3 The order (*f, e, d, c, b, a*) corresponds to an upstream scheduling.

For an example, consider the network shown in Fig. 3. Assume node *e* has a frame to transmit to node *f*. *e* has three neighbors: *b*, *c* and *d*. The hop-count (on the tree) from *b* to *f* is 3. The hop-count (on the tree) from *c* to *f* is 1. The hop-count (on the tree) from *d* to *f* is 4. Thus, *e* chooses *c* as its next hop. Here the shortcut tree routing protocol uses only a local knowledge to compute the next hop. For example, if node *d* had a frame to send to *f*, *d* would assume that the hop-count from *e* to *f* is 4, and 3 from *b*. *d* would forward the frame to *b*, which would not be an optimal decision. Indeed, *d* would not be able to determine that *e* knew a shortcut on the tree through *c*.

5. MULTI-STACK ARCHITECTURE

In this section, multi-stack is described based on multi-layering approach. Multi-layer refers to a single protocol that spans various several layers of the network stack, the term *multi-stack* is used in contrast to *cross-layering*, which is possible to fit several MAC protocols at the MAC layer, and several NWK protocols at the NWK layer [11]. This multi-stack architecture is orthogonal with cross-layering strategies.

Fig. 4 illustrates the multi-stack architecture which composed of several MAC/NWK combinations. This supports different types of traffic. And as it is shown figure, every combination is associated to a specific queue. This queue can receive only one type of traffic. Using multi-stack architecture, the QoS of the network is improved in terms of throughput, delay, and packet loss. The main aim of this paper is to provide increased throughput, decreased delay and decreased or no packet loss.
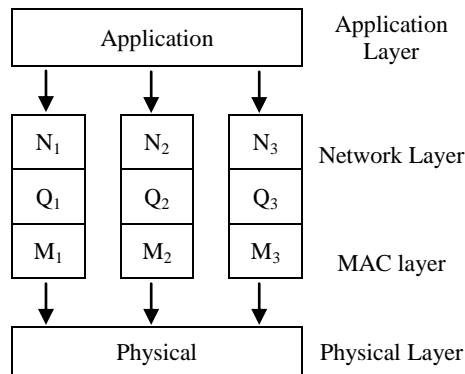

Fig. 4 Architecture of multi-stack architecture

5.1 Explanation of the Architecture:

The multi-stack architecture uses a time schedule repeated cyclically. The time schedule divides time into *t* intervals. During each interval t_i, a MAC protocol M_i and a NWK protocol N_i are activated. To ensure a coherent medium access control and routing in the whole network, all the nodes are synchronized and are in the same interval at a given instant. Nodes can be put into sleep mode during an interval *i* if $M_i = N_i = \emptyset$. They also switch to sleep mode according to the properties of each M_i.

Two major elements that have to be shared between the nodes are: the instant when the first interval of the schedule starts, and the length and protocols that have to be activated for each interval. To share this knowledge, we assume that there is a synchronization mechanism which precedes the

first interval. This synchronization mechanism can be repeated at the beginning of each time schedule.

Fig. 5 shows an example of the time schedule with a synchronization period and *n* = 2 intervals.
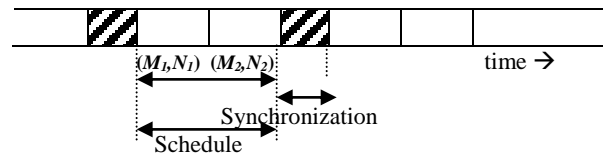

Fig. 5 Repeated time-intervals in the multi-stack schedule

Assume that each packet is marked by the application or by an upper layer providing QoS management. According to first hypothesis, a packet marked with a number *i* can be dealt with during interval t_i. The centralized approach has been chosen to simplify the description. The architecture can be easily distributed, provided that the synchronization mechanism is distributed too.

5.2 Technical specification:

In this architecture a specific queue Q_i is used for each combination (M_i, N_i). Queue Q_i only receives frames marked by *i*. Recall that a queue between the NWK and the MAC layer already exists in the network stack. There is no overhead in terms of frame length. While each application has to provide the NWK layer with the number *i*, this information is not stored at the NWK layer. The packet is simply handed over to the NWK protocol N_i, which eventually stores it in Q_i. The MAC protocol M_i sends the frame without including the value *i*, which can be deduced by any receiving node, as the frame is received during interval t_i.

Table 1 Simulation Parameters

Data Generation Period	1 packet / 100 millisecond
Data Sending Period	1 forwarding message / 25 millisecond
Multi-Queue	Three Queue High Medium Low
No. of Nodes	30 – 150
Packet types	Real-time, Critical, delay-constrained

5.3 Improvement of the architecture:

This architecture assumes that the application chooses to mark the packets according to the properties of the traffic. It is also possible to improve this by marking packets according to the current interval too. An application requires a small delay that decides to send the frame in the interval t_i corresponding to the MAC/NWK combination with the smallest delay. It is possible that *ti* is only scheduled after a long period. This happens if the current interval is t_{i+1} for example. If the application (or an intermediate layer) knows the current interval and the whole schedule, it could decide to send the packet in a interval t_j scheduled before the next t_i.

6. EVALUATION

The simulation is done using JProwler. The proposed work was simulated using a open source simulator called JProwler. Prowler (ISIS 2006) is a probabilistic sensor simulator written in Matlab, and has a version build in java (JProwler). JProwler is built for MICA Mote hardware platform, which is running on Tiny OS. It also has a very efficient throughput, but it provides only one MAC protocol of TinyOS. Other required MAC protocols were designed to support the current work. All the nodes of the network had no clock drift and are supposed to be synchronized with perfection. Simulation parameter setting is given in table 1.

Table 2 Structure of Data Packet

Field	Source ID	Data	Data Type	Priority
Byte	**4**	**8**	**1**	**1**

The structure of data packet used is shown in the table 2. The data packet format contains the sources ID, the payload. It also contains the type of data based on which the priority of the packet is assigned which is used for scheduling.
The time is divided into two time-intervals: the first time interval is used by the slotted CSMA/CA, and the second time-interval is used by the upstream MAC and shortcut routing protocols.

→ Different MAC protocols
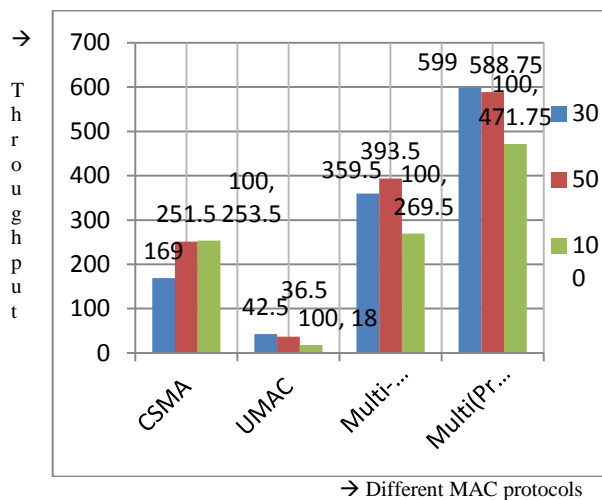
Fig. 6 Throughput Analysis with different MAC Protocols

Fig. 6 gives the throughput behavior of the network for transferring the packets with respect to the number of nodes. The table given below shows the simulation results obtained by considering different set of nodes.

Graph details

No. Of Nodes / Mac Protocol	30 (Blue)	50 (Red)	100 (Green)
CSMA	169	251.5	253.5
UMAC	42.5	36.5	18
Multi-stack	359.5	393.5	269.5
Multi(Priority)	599	588.75	471.75

CSMA/CA protocol shows the consistency in throughput. Multi-stack architecture using both CSMA/CA and UMAC provides much better performance when compared to both used individually. Multi-Stack architecture when priority queue is used shows a highly good performance when compared to Multi-Stack without packet priority assigned and all the other two schemes.

7. ADVANTAGES OF MULTI-STACK ARCHITECTURE

Use of multi-stack architecture supports multiple applications on the same or single network. This supports the current technology. This reduces the network deployment cost and hence provides scalability. More over the packets are transferred from source to sink with or no packet loss, less delay and with increased throughput. For e.g. When a packet of size 100 bytes to be transferred from source to sink, it is divided in two of 50 bytes each and are stored in the two stacks. With the help of required protocol the divided two packets of 50 bytes are transferred from source to sink parally thus by increasing the overall network throughput, decreased delay and packet loss.

8. CONCLUSION

By using different MAC and NWK protocols in our architecture, new properties of routing protocols are identified, that allow some packets to be routed safely according to any of the routing protocols. This architecture has been validated by simulations with respect to three main QoS i.e, throughput, delay and packet loss. Among which only throughput comparison is only discussed in this paper. Other parameters would be studied elaborately and discussed in the next paper. The perspectives for this work focuses on testing its performance in a real network deployment.

9. ACKNOWLEDGEMENT

10. REFERENCES

[1] M. Petrova, J. Riihijarvi, P. Mahonen, and S. Labella, "Performance study of IEEE 802.15.4 using measurements and simulations," in *IEEE WCNC*, 2006, pp. 487–492.

[2] T. O. Kim, H. Kim, J. Lee, J. S. Park, and B. D. Choi, "Performance analysis of IEEE 802.15.4 with non-beacon-enabled CSMA/CA in nonsaturated condition," in *Embedded and Ubiquitous Computing*, ser. LNCS, vol. 4096, 2006, pp. 884–893.

[3] P. Baronti, P. Pillai, V. W. C. Chook, S. Chessa, A. Gotta, and Y. Fun Hu, "Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards," *Computer communications*, vol. 30, pp. 1655–1695, 2007.

[4] IEEE 802.15, "Part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANs)," ANSI/IEEE, Standard 802.15.4 R2006, 2006.

[5] ZigBee, "ZigBee Specification," ZigBee Standards Organization, Standard ZigBee 053474r17, January 2008.

[6] T. Kim, D. Kim, N. Park, S.-E. Yoo, and T. S. L´opez, "Shortcut tree routing in ZigBee networks," in *ISWPC*, 2007.

[7] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *Operating Systems Design and Implementation*, vol. 36, 2002, pp. 147–163.

[8] S. Ganeriwal, R. Kumar, and M. B. Sribastava, "Timing-sync protocol for sensor networks," in *ACM Embedded Networked Sensor Systems*, 2003, pp. 138–149.

[9] G. Chalhoub, A. Guitton, and M. Misson, "MAC specifications for a WPAN allowing both energy saving and guaranted delay - Part A: MaCARI: a synchronized tree-based MAC protocol," in *IFIP WSAN*, 2008.

[10] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," IETF, Request For Comments 3561, July 2003.

[11] El Rachkidy, N., Guitton, A., and Misson, M. Improving qos in wireless sensor networks using a multi-stack architecture. In IEEE Vehicular Technology Conference (2011).

[12] Petr Jurčík, Anis Koubâa, Mário Alves, Eduardo Tovar, Zdeněk Hanzále, "A Simulation Model for the IEEE 802.15.4 Protocol: Delay/Throughput Evaluation of the GTS Mechanism", 15th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2007. MASCOTS '07, 24-26 Oct. 2007, page(s): 109 - 116, Istanbul, ISSN: 1526-7539, Print ISBN: 978-1-4244-1853-4.

[13] Feng Chen Talanis, T. German, R. Dressler, F. "Real-time enabled IEEE 802.15.4 sensor networks in industrial automation" , IEEE International Symposium Industrial Embedded Systems, 2009. SIES '09. ISBN: 978-1-42444109-9, INSPEC Accession Number: 1081428

[14] Collotta M., Salerno V. M. "A real-time network based on IEEE 802.15.4 / ZigBee to control home automation environment", International forum "Modern Information Society Formation – problems, perspectives, innovation approaches". St.-Petersburg, Russia, June 6–11, 2010

[15] Patrick R. Casey, Kemal E. Tepe, and Narayan Kar, " Design and Implementation of a Testbed for IEEE 802.15.4 (Zigbee) Performance Measurements", Research Article, EURASIP Journal onWireless Communications and Networking, Volume 2010, Article ID 103406, 11 pages, doi:10.1155/2010/103406, Hindawi Publishing Corporation.

[16] Feng Chen, Nan Wang, Reinhard German and Falko Dressler, "Simulation study of IEEE 802.15.4 LR-WPAN for industrial applications", Wirel. Commun. Mob. Computer. 10: 609–621 (2010) Published online in Wiley InterScience (www.interscience.wiley.com) DOI: 10.1002/wcm.736

[17] D Rohm, M Goyal, N Dua, H Hosseini, K Vairavan, "Smart Parking Lots of the Future", Symposium (RTAS) 2007: Workshop on Smart Transportation, April 3, 2007, Bellevue.

[18] Mario Collotta, Valerio Mario Salerno, "A REAL-TIME NETWORK BASED ON IEEE 802.15.4/ZIGBEE TO CONTROL HOME AUTOMATION ENVIRONMENT", ISA European paper competition - Modern Information society formation - Problems Perspectives, innovation Approaches, ISBN/ISSN: ISBN 978-5-8088-0528-6 - 6-11 June 2010.

[19] Ricardo Severino, Manish Batsa, Mário Alves, Anis Koubâa, "A Traffic Differentiation Add-On to the IEEE 802.15.4 Protocol: implementation and experimental validation over a real-time operating system", 13th Euromicro Conference on Digital System Design, Architectures, Methods and Tools, DSD 2010, 1-3 September 2010, Lille, France. IEEE 2010, ISBN 978-0-7695-4171-6.

[20] IEEE-TG15.4, "Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)," *IEEE standard for Information Technology*, 2003.

[21] Zigbee-Alliance, "ZigBee specification" *ttp://www.zigbee.org/*, 2005.

[22] J. Zheng and J. L. Myung, "Will IEEE 802.15.4 Make Ubiquitous Networking a Reality?-- A Discussion on a Potential Low Power, Low Bit Rate Standard," *IEEE CommunicationsMagazine*, vol. 42, No. 6, pp. 140-146, 2004.

[23] D. Geer, "Users Make a Beeline for ZigBee Technology," *IEEE Computer Society Press*, vol. 38, Issue 12, pp. 16-19, Dec., 2005.

[24] J. Adams, "Building low power into wireless sensor networks using ZigBee technology," *Industrial Embedded Systems Resource Guide, Networking: Technology*, pp. 26-30, 2005.

[25] T. Culter, "Deploying ZigBee in existing industrial automation networks," *Industrial Embedded System Resource Guide, Networking: Technology*, pp. 34-36, 2005.

[26] C. Herzog, "Creating value with ZigBee Networks," *Industrial Embedded System Resource Guide, Networking: Technology*, pp. 31-33, 2005.

[27] ZigBee, "ZigBee Specification," ZigBee Standards Organization, Standard Zigbee 053474r17, January 2008.

[28] F. Cuomo, S. Della Luna, U. Monaco, and F. Melodia, "Routing in ZigBee: Benefits from exploiting the IEEE 802.15.4 association tree," in IEEE International Conference on Communications (ICC), 2007, pp. 3271–3276.