International Journal of Electronics and Communication Engineering (IJECE)
ISSN(P): 2278-9901; ISSN(E): 2278-991X
Vol. 3, Issue 1, Jan 2014, 17-34
© IASET

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations

PERFORMANCE EVALUATION OF CLUSTER TREE TOPOLOGY BASED WIRELESS SENSOR NETWORKS

K. AYYAPPA RAVI KIRAN[1], Y. CHALAPATHI RAO[2] & CH. SANTHI RANI[3]

[1]Research Scholar, ECE Pursuing, BIET, Pennada, Andhra Pradesh, India

[2]HOD, Department of ECE, BIET, Pennada, Bhimavaram, Andhra Pradesh, India

[3]Professor, Department of ECE, DMSSVH College of Engineering, MTM, Andhra Pradesh, India

ABSTRACT

Time-sensitive Wireless Sensor Network (WSN) applications require finite delay bounds in critical situations. WSN is the accepted technologies for emerging wireless communications standards. The activities in many wireless standardization bodies and forums, for example IEEE 802.15.4/ZigBee WSNs, attest to this fact. Evaluating the performance of sensor nodes in worst-case conditions (i.e. no sink) could make or mar the opportunities in these networks. It is essential to seek outstanding performance benchmarks to which various modeling schemes can be compared. Therefore, this paper provides a comprehensive cluster-tree WSNs with a mobile sink with Logical ZigBee tree topology in a way to ensure closed-form recurrent expressions for computing the worst-case end-to-end delays, buffering and bandwidth requirements in any network path in the cluster-tree assuming error free channel. In contrast, it has been observed that the performance of IEEE 802.15.4/ZigBee WSNs with its theoretical results in different sink node conditions. Out system results are accurate compared with its experimental results here we validate the theoretical results through experimentation.

KEYWORDS: Wireless Sensor Networks (WSNs), Worst Case Network Dimensioning, Network Calculus, IEEE 802.14.8/ ZigBee, Cluster Tree Topology

1. INTRODUCTION

In time-sensitive Wireless Sensor Network (WSN) applications, it is important that time-critical messages arrive to their destination prior to the expiration of their deadlines. This requires a priori dimensioning of the available resources of the WSN to provide an end-to-end guaranteed service from the source node to the sink

A wireless sensor network is a group of nodes organized into a cooperative network. Each node consists of processing capability, may contain multiple types of memory, have a RF transceiver, have a power source, and accommodate various sensors and actuators. The nodes communicate wirelessly and often self-organize after being deployed in an ad hoc fashion.

A medium access control (MAC) protocol coordinates actions over a shared channel. The most commonly used solutions are contention-based. One general contention-based strategy is for a node which has a message to transmit to test the channel to see if it is busy, if not busy then it transmits, else if busy it waits and tries again later. After colliding, nodes wait random amounts of time trying to avoid re-colliding. If two or more nodes transmit at the same time there is a collision and all the nodes colliding try again later. Many wireless MAC protocols also have a dozen mode where nodes not involved with sending or receiving a packet in a given timeframe go into sleep mode to save energy. Many variations exist on this basic scheme.

A cluster-tree WSN has the sink it is a central point that collects all sensory data attached to the root. In this paper we evaluate the worst-case network performance assuming a cluster-tree topology of balanced height and load based on the sink behavior [1]. Here we derive per-hop and an end-to-end resource requirement in worst-case delays of upstream flows and investigates the worst-case resource dimensioning and analysis of cluster-tree WSNs with mobile sink behavior with upstream flows (sink in the root) and downstream flows (sink not in the root). The paper proposes and describes a system model, an analytical methodology and software tool that permit the worst-case dimensioning and analysis of cluster-tree WSNs. We simplify the analysis by avoiding the mobility management.it is very important to use our methodology with IEEE 802.15.4/ZigBee protocols which are very promising technologies for WSNs. After a successful execution of this system we show the validity of our theoretical model by comparing worst-case results with the maximum and average values measured through experimentation.

II. BACKGROUND WORK

Network Calculus

Network Calculus is a set of recent developments that provide deep insights into flow problems encountered in networking [2]. The foundation of network calculus lies in the mathematical theory of tools, and in particular, the Min-Plus algebra. With network calculus, we are able to understand some fundamental properties of integrated services networks, window flow control, scheduling and buffer or delay dimensioning.

A basic system model S in Network Calculus consists of a buffered FIFO (First-In, First-Out order) node with the corresponding transmission link in figure 1. For a given data flow, the input function R(t) represents a cumulative number of bits that have arrived to system S in the time interval (0, t). The output function R*(t) represents the number of bits that have left S in the same interval (0, t). Both functions are wide sense increasing, i.e. R(s) ≤ R(t) if and only if s ≤ t.

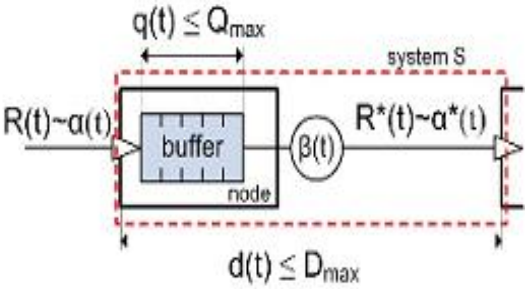


Figure 1: The Basic System Model in Network Calculus

Concatenation Theorem

Assume that S_i offers service curve $\beta_i(t)$, i=1,2 to the flow. Then the concatenation of these two systems offers a following single service curve $\beta\ (t)$ to the traversing flow:

$$\beta(t) = (\beta_1 \otimes \beta_2)(t) \tag{2.1}$$

Where $\otimes$ is the min-plus convolution defined for f, g ∈ **F**, where **F** is the set of wide-sense increasing functions, as: Min-plus convolution has several important properties, including being commutative and associative. Furthermore, convolution of concave curves is equal to their minimum. It can be also shown that service curve β and arrival curve α can be expressed using a min-plus convolution as follows:

$$R^* \geq (R \otimes \beta)(t) = \inf_{0 \leq S \leq t} \{R(t-s) + \beta(s)\} \text{ for } \forall t \geq 0$$

$$R \leq (R \otimes \alpha)(t) = \inf_{0 \leq S \leq t} \{R(t-s) + \alpha(s)\} \text{ for } \forall t \geq 0 \tag{2.2}$$

Due to the aggregation of the data flows in the direction of the sink, each router must provide a service curve $\beta(t)$ to the aggregated data flow. Thus, the delay and backlog bounds are computed for the aggregated data flow traversing the router. On the other hand, using the aggregate scheduling theorem, tighter bounds can be computed for each individual data flow traversing the network. In this work, both approaches are used to compare the results.

We consider a data flow constrained by the (b, r) arrival curve $\alpha(t)$ and traversing system S with a rate-latency service curve $\beta_{R,T}(t)$, Then, the guaranteed performance bounds $D_{\max}$ and $Q_{\max}$ are computed as:

$$D_{\max} = \frac{b}{R} + T \quad Q_{\max} = b + r.T \tag{2.3}$$

An upper bound of the outgoing flow with output function $R^*(t)$, called output bound, as

$$\alpha^*(t) = \alpha(t) \Theta \beta_{R,T}(t) = \alpha(t) + r.T \geq \alpha(t) \tag{2.4}$$

The nodes offer a service curve $\beta(t)$ to this aggregated data flow in the direction of the sink. Thus, the delay and backlog bounds can be calculated for the entire aggregate data flow at each node. Using the aggregate scheduling theorem, tighter bounds can be computed for individual flows traversing the network [3].

Aggregate Scheduling Theorem

Consider a lossless node multiplexing two data flows, 1 and 2, in FIFO order. Assume that flow 2 is constrained by the arrival curve $\alpha_2(t)$ and the node guarantees a service curve $\beta(t)$ to the aggregate of these two flows. Define the family of functions as:

$$\beta_1(t,\theta) = (R - r_2)\left[t - \left(\frac{b_2 + r_2(T-\theta)}{R - r_2} + T\right)\right]^+ .1_{\{t>\theta\}} \tag{2.5}$$

III. SYSTEM MODEL

This section defines general cluster-tree topology and data flow models that will be considered in the following analysis. It also elaborates on the worst-case cluster scheduling; that is, the time sequence of clusters' active portions leading to the worst-case end-to-end delay for a message to be routed to the sink [4]. To ensure predictable performance of a WSN, the network topology and data flows must be bounded. To provide closed-form recurrent expressions for computing the worst-case performance bounds in a WSN, the network topology and data load must be balanced [5].

Cluster-Tree Topology Model

The worst-case cluster-tree topology is graphically represented by a rooted balanced directed tree defined by the following three parameters. Height of the tree (**H**), Maximum number of end-nodes that can be associated to a router ($N_{end-end-node}^{MAX}$) and Maximum number of child routers that can be associated to a parent router (N_{router}^{MAX}).The depth of a node is defined as the number of logical hops from that node to the root. The root is at depth zero, and the maximum depth of an end-node is H+1.

Sink is a node that gathers the sensory data from all sensor nodes inside the network. For our experimental analysis we take sink to be an autonomous and topology-independent mobile node. The mobile behavior means that a sink moves arbitrarily within a static cluster-tree WSN and can be associated with any router within communication range. The router, to which the sink is in a given moment associated, is referred to as sink router. the depth at a given moment of the sink router in a cluster-tree topology is denoted as $H_{\sin k} \in (0, H)$.The $H_{\sin k} = 0$ whenever the sink is associated with router and network contains only upstream flows. Here we analyze the performance of WSN with $H_{\sin k} > 0$

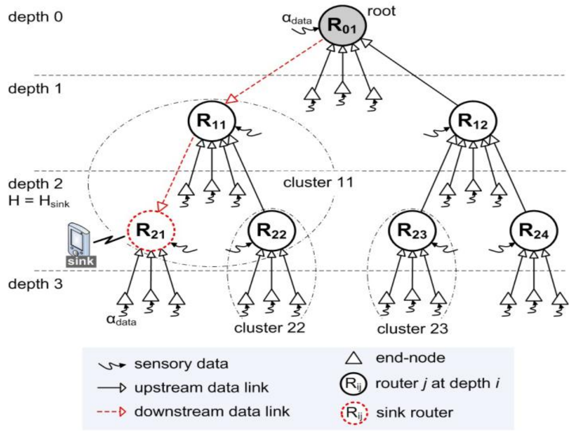


Figure 2: The Cluster-Tree Topology and Data-Flow Models

Data-Flow Model

Data traffic is routed to the sink router without any in-network processing on the way. In the worst-case, all sensor nodes are assumed to contribute equally to the network load; sensing and transmitting sensory data upper bounded by the affine arrival curve $\alpha_{data} = b_{data} + r_{data}.t$ .Where b_{data} is the burst tolerance and r_{data} is the average data rate. The affine arrival curve can represent any type of traffic, assuming that it can be bounded. It can represent a periodic or aperiodic traffic or any other random traffic (VBR traffic). This is the main reason for using this simple but effective and general arrival curve model: to be independent of any specific pattern/distribution of traffic.

Note that the data flows requiring real-time guarantees are only considered in the analysis. Other best-effort flows are also supposed to exist. Each end-node is granted a service guarantee from its parent router corresponding to the rate-latency service curve $\beta_{end-node} = R_{end-node}\left(t - T_{end-node}\right)$

Where $R_{end-node} \geq r_{data}$ is the guaranteed link bandwidth and $T_{end-node}$ is the maximum latency of the service. The same service curve is provided to all end-nodes by their parent routers. Upper bounds the outgoing data from any end-node is denoted as.

$$\alpha^{*}_{data} = \alpha_{data} + r_{end-node}.T_{end-node} \tag{3.1}$$

The amount of bandwidth allocated by each router depends on the cumulative amount of data at its inputs, which increases towards the sink. Thus, the total input function R(t) of each router depends on the depth, and consists of the sum of the output functions $R^{*}(t)$ of its end-nodes and child routers. Additionally, the router itself can be equipped with sensing capability producing a sensory data traffic bounded by α_{data}. Thus, in general case, the arrival curve constraining the total input function R(t) of a router at a depth i is expressed as

$$\overline{\alpha}_i = \alpha_{data} + N_{end-node}^{MAX}.\alpha_{data}^* + N_{router}^{MAX}.\alpha_{i+1}^* \tag{3.2}$$

The outgoing data of a router at depth i, that receives guaranteed service curve β_{i-1}, is constrained by the output bound as follows:

$$\alpha_i^* = \overline{\alpha}_i \Theta \beta_{i-1} \tag{3.3}$$

Hence, the data flow analysis consists in the computation of the arrival Curves $\overline{\alpha}_i$ and output bounds α_i^*, using iteratively Equations (3.2) and (3.3), from the deepest routers until reaching the, sink router. After that, the resource requirements of each router, in terms of buffer requirement Q_i and bandwidth requirement R_i, and the worst-case end-to-end delay bounds are computed. If the sink is associated to the root, i.e. $H_{\sin k} = 0$, all data flows only in upstream direction. In what follows, the upstream and downstream directions are marked by the subscripts U and D, respectively (e.g. $\alpha_{iU}^*, \overline{\alpha}_{iD}$). Each router at depth i provide two types of service curves to its child routers at depth i + 1, which are denoted as:

$$\beta_{iU} = R_{iU}.(t - T_{iU})^+ \quad \beta_{iD} = R_{iD}.(t - T_{iD})^+ \tag{3.4}$$

Here R_i is the guaranteed link bandwidth. T_i is the maximum latency that a data must wait for a service. The same downstream or upstream service curves must be guaranteed to all downstream or upstream flows at a given depth, respectively.

Time Division Cluster Scheduling

In case of single collision domain, the TDCS must be non-overlapping, i.e. only one cluster can be active at any time. Hence, the period of TDCS is given by the number of clusters and the length between clusters, it is mandatory to schedule the clusters' active portions in an ordered sequence that is called Time Division Cluster Schedule (TDCS) [6]. In case of single collision domain, the TDCS must be non-overlapping, i.e. only one cluster can be active at any time. Hence, the period of TDCS is given by the number of clusters and the length of their active portions. On the contrary, in a network with multiple collision domains, the clusters from different non-overlapping collision domains may be active at the same time. Note that the non-overlapping TDCS can be more pessimistic in networks with multiple collision domains.

IV. NETWORK ANALYSIS

We assume that in system the end-nodes have sensing capabilities, but the sensing capability of routers is optional. For an improved analysis, we introduce a binary variable S whose value is equal to 1 if routers have sensing capabilities; otherwise S is equal to 0. The total input data flow of each router as shown in Eq. (3.2) comprises, among other terms, the sum of the output flows of its end-nodes and, optionally, its own sensory data flow constrained by $\alpha_{data}(t)$. This part of the total input flow is the same for upstream and downstream flows, hence we introduce the substitution:

$$\overline{\alpha}_H(t) = S.\alpha_{data}(t) + N_{end-node}^{MAX}.\alpha_{data}^{*}(t) \tag{4.1}$$

Thus, using Eq. (3.1) we get:

$$\overline{\alpha}_H(t) = (N_{end-node}^{MAX} + S).\alpha_{data}(t) + N_{end-node}^{MAX}.r_{data}.T_{data} \tag{4.2}$$

Where, $\overline{r}_H = (N_{end-node}^{MAX} + S).r_{data}$ is the resulting aggregate rate of $(N_{end-node}^{MAX} + S)$ input data flows, and $\overline{b}_H = \left(N_{end-node}^{MAX} + S\right).b_{data} + N_{end-node}^{MAX}.r_{data}.T_{data}$ is the burst tolerance. Note that $\overline{\alpha}_H(t)$ is also equal to the total input upstream flow of the deepest routers (at depth H).

Upstream Data Flows

First, the arrival curves of the incoming data in upstream direction $\overline{\alpha}_{iU}$ and the upper bounds of the outgoing data in upstream direction α_{iU}^{*} are evaluated depth by depth, using the Network Calculus methodology, starting from depth H (i.e. the deepest routers). The analysis considers the general queuing model for the upstream direction as illustrated in Figure 3. The arrival curve, constraining the total input upstream flow of each router at depth i, is expressed as follows:

$$\overline{\alpha}_{iU} = \left(\sum_{j=0}^{H-i} N_{router}^{MAX}\right).\overline{\alpha}_H + \sum_{j=1}^{H-i}\left(\left(N_{router}^{MAX}\right)^j\right).\sigma_{i+j-1} \tag{4.3}$$

For $\forall i, 0 \le i \le H$ $\quad \sigma_n = \left(\sum_{k=0}^{H-(n+1)} (N_{router}^{MAX})^k\right).\overline{r}_H.T_{nU}$
$where$

The output bound for the upstream data flow from each child router at depth i, receiving a service curve $\beta_{i-1}(t)$ from a parent router at depth i-1, is then expressed as:

$$\alpha_{iU}^{*} = \overline{\alpha}_{iU} + \sigma_{i-1} =$$

$$\left(\sum_{j=0}^{H-i}\left(N_{router}^{MAX}\right)\right).\overline{\alpha}_H + \sum_{j=0}^{H-i}\left(\left(N_{router}^{MAX}\right)^j.\sigma_{i+j-1}\right) \text{ (4.4)}$$

For $\forall i, 0 < i \le H$

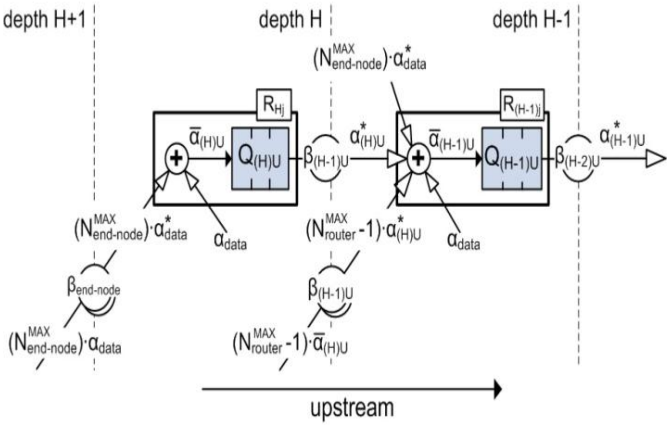

depth H+1
depth H
depth H-1
$(N_{end\text{-}node}^{MAX})\cdot\alpha_{data}^{*}$
R_{Hj}
$R_{(H-1)j}$
$\overline{\alpha}_{(H)U}$
$Q_{(H)U}$
$\beta_{(H-1)U}$
$\alpha_{(H)U}^{*}$
$\overline{\alpha}_{(H-1)U}$
$Q_{(H-1)U}$
$\beta_{(H-2)U}$
$\alpha_{(H-1)U}^{*}$
$(N_{end\text{-}node}^{MAX})\cdot\alpha_{data}^{*}$
α_{data}
$(N_{router}^{MAX}-1)\cdot\alpha_{(H)U}^{*}$
α_{data}
$\beta_{end\text{-}node}$
$\beta_{(H-1)U}$
$(N_{end\text{-}node}^{MAX})\cdot\alpha_{data}$
$(N_{router}^{MAX}-1)\cdot\overline{\alpha}_{(H)U}$
upstream

Figure 3: The Queuing System Model for the Upstream Direction

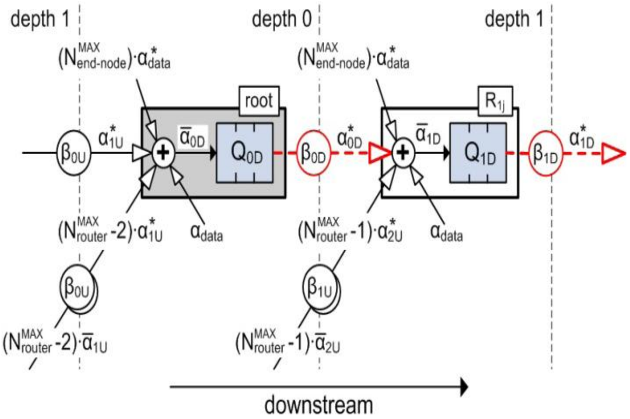

depth 1
depth 0
depth 1
$(N^{MAX}_{end\text{-}node})\cdot\alpha^*_{data}$
$(N^{MAX}_{end\text{-}node})\cdot\alpha^*_{data}$
root
R_{1j}
β_{0U}
α^*_{1U}
$\bar{\alpha}_{0D}$
Q_{0D}
β_{0D}
α^*_{0D}
$\bar{\alpha}_{1D}$
Q_{1D}
β_{1D}
α^*_{1D}
$(N^{MAX}_{router}-2)\cdot\alpha^*_{1U}$
α_{data}
$(N^{MAX}_{router}-1)\cdot\alpha^*_{2U}$
α_{data}
β_{0U}
β_{1U}
$(N^{MAX}_{router}-2)\cdot\bar{\alpha}_{1U}$
$(N^{MAX}_{router}-1)\cdot\bar{\alpha}_{2U}$
downstream

Figure 4: The Queuing System Model for Downstream Direction

Downstream Data Flows

The arrival curves of the incoming data in downstream direction $\bar{\alpha}_{iD}$ and the upper bounds of the outgoing data in downstream direction α^*_{iD} is evaluated depth by depth, using the Network Calculus methodology, starting from depth 0 (i.e. the root). The analysis considers the general queuing model for downstream direction as illustrated in Figure 4. The arrival curve constraining the total input downstream flow of a router at a given depth i, for

$i = 0,\ldots\ldots, (H_{\sin k} - 1)$, is expressed as:

$$\bar{\alpha}_{iD} = \left(\sum_{j=0}^{i} \left(N^{MAX}_{router}\right)^{H-j} \right) . \bar{\alpha}_{H} + \left(N^{MAX}_{router} - 1\right) \sum_{j=0}^{i} \delta_j + \sum_{j=0}^{i-1} \tau_j \tag{4.5}$$

For $\forall i, 0 \le i < H_{\sin k}$, $\delta_n = \sum_{k=0}^{H-(n+1)} \left(\left(N^{MAX}_{router}\right)^k . \sigma_{k+n} \right)$
where

$$\sigma_n = \left(\sum_{k=0}^{H-(n+1)} \left(N^{MAX}_{router}\right)^k \right) . \bar{r}_H . T_{nU}$$

$$\tau_n = \left(\sum_{k=0}^{n} \left(N^{MAX}_{router}\right)^{H-k} \right) . \bar{r}_H . T_{nD}$$

The upper bound of the output downstream flow from a parent router at depth i, providing a service curve $\beta_{iD}(t)$, towards its child router at depth i+1 is expressed as:

$$\alpha^*_{iD} = \bar{\alpha}_{iD} + \tau =$$

$$\left(\sum_{j=0}^{i} \left(N^{MAX}_{router}\right)^{H-j} \right) . \bar{\alpha}_{H} + \left(N^{MAX}_{router} - 1\right) \sum_{j=0}^{i} \delta_j + \sum_{j=0}^{i} \tau_j \tag{4.6}$$

For $\forall i, 0 \le i < H_{\sin k}$

Note that the sink can be associated to the router at a depth lower than the height of the cluster-tree, i.e. $H_{\sin k} < H$ Figure 3 or equal to the height of the Cluster-tree, i.e. $H_{\sin k} = H$, the arrival curve constraining the total input downstream flow is expressed as:

$$\overline{\alpha}_{(H_{\sin k}).D} = \overline{\alpha}_{H} + N_{router}^{MAX}.\alpha^{*}_{(H_{\sin k}+1).U} + \alpha^{*}_{(H_{\sin k}-1).D} \tag{4.7}$$

If $H_{\sin k} = H$, the arrival curve constraining the total input downstream flow is expressed as:

$$\overline{\alpha}_{(H_{\sin k}).D} = \overline{\alpha}_{H} + \alpha^{*}_{(H_{\sin k}-1).D} \tag{4.8}$$

V. WORST-CASE NETWORK DIMENSIONING

Per-Router Resources Analysis

The aim is at specifying the minimum bandwidth of each upstream and downstream data links and the minimum buffer size at each router needed to store the bulk of data incoming through the router's inputs.

Bandwidth Requirements

Consider a parent router at depth i providing a service curve β_{iU} or β_{iD} to its child routers at depth i+1 in upstream or downstream direction, respectively. In the upstream case, the outgoing data of a child router at depth i +1 is constrained by the output bound $\alpha^{*}_{(i+1).U}$ and dispatched through the upstream link to its parent router at depth i. Thus, to ensure a bounded delay, the guaranteed amount of bandwidth R_{iU} must be greater than or equal to the outgoing data rate $r^{*}_{(i+1).U}$ .As a result, by applying Eqs. (12) and (9) we obtain:

$$R_{iD} \geq \overline{r}_{iD} = r^{*}_{iD} = \left(\sum_{j=0}^{i} \left(N_{router}^{MAX} \right)^{H-j} \right) . \overline{r}_{H}$$

$$= \left(\sum_{j=0}^{i} \left(N_{router}^{MAX} \right) \right) . \left(N_{end-node}^{MAX} + S \right), r_{data} \tag{5.1}$$

For $\forall i, 0 \leq i < H_{\sin k}$.

Note that it is possible to determine the total number of routers in a network using Eq. (16) by having i = H and $\overline{r}_{H} = 1$, which is expressed as:

$$\sum \left(N_{router}^{MAX}, H \right) = \sum_{j=0}^{H} \left(N_{router}^{MAX} \right)^{H-j} \tag{5.2}$$

Buffer Requirements

The buffer of a downstream router at depth i must be able to store all incoming data for avoiding the buffer overflow, constrained by the arrival curve $\overline{\alpha}_{iD}(t)$, until it is dispatched through the downstream link to a child router at depth i+1. The required buffer size Q_{iD} of the downstream router at depth i must be at least equal to the burst tolerance b^{*}_{iD} of the output bound $\alpha^{*}_{iD}(t)$ .Hence, according to Eq. (4.6) we get:

$$Q_{iD} = b_{iD}^{*} = b_{iD}^{*BURST} + b_{iD}^{*UP\_LAT} + b_{iD}^{*DOWN\_LAT}$$

$$= \left(\sum_{j=0}^{i} \left(N_{router}^{MAX} \right)^{H-j} \right) . \bar{b}_{H} + \left(N_{router}^{MAX} - 1 \right) . \sum_{j=0}^{i} \delta_{j} + \sum_{j=0}^{i} \tau_{j}$$

For $\forall i, 0 \leq i < H_{\sin k}$ (5.3)

Observe that the buffer requirement is the sum of three terms. Similarly to the upstream case, the burst term is related to the burst tolerance b_{data}, and the second term is related to the cumulative effect of the service latencies of upstream data. The third term represents the cumulative effect of the service latency at each depth for downstream data. In case of a sink router at depth $H_{\sin k}$ the buffer requirement must be greater than or equal to the burst tolerance $\bar{b}_{(H_{\sin k}).D}$ of total incoming data $\bar{\alpha}_{(H_{\sin k}).D}$ given by Eq. (4.6) or Eq. (4.7).

End-To-End Delay Analysis

The worst-case end-to-end delay is the delay bound of a data flow router along the longest path in the network. It can be computed using two approaches, as follows.

Per-Hop End-To-End Delay

The first approach consists in computing the per-hop delay bounds of the aggregate flows, and then deducing the end-to-end delay bound as the sum of per-hop delays. In the upstream case .According to Eq. (2.3), the delay bound between a child router at depth i and its parent router at depth i-1 guaranteeing service curve $\beta_{(i-1).U}$ is expressed a $D_{iU} = \bar{b}_{iU} / R_{(i-1)U} + T_{(i-1)U}$. On the other hand, in the downstream case, the delay bound between a parent router at depth i, which guarantees service curve β_{iD} to its total incoming data constrained by arrival curve $\bar{\alpha}_{iD}$, and its child router at depth i+1 is expressed as $D_{iD} = \bar{b}_{iD} / R_{iD} + T_{iD}$. Hence, the worst-case end-to-end delay is the sum of all per-hop delay bounds along the longest routing path, as follows:

$$D_{e2e}^{MAX} = D_{data} + \sum_{i=1}^{H} D_{iU} + \sum_{i=0}^{H_{\sin k}-1} D_{iD} \qquad (5.4)$$

Where $D_{data} = b_{data} / R_{data} + T_{data}$ is the delay bound between an end-node and its parent router. This approach is a bit pessimistic, since the delay bound at each router is computed for the aggregation of all incoming flows. Tighter end-to-end delay bounds can be computed for individual flows, as described next.

Per-Flow End-To-End Delay

The idea of this approach is to derive the service curves guaranteed to a particular individual flow f by the routers along the path, using the aggregate scheduling theorem in Eq. (2.5), and then deduce the network-wide service curve for flow f based on the concatenation theorem. Finally, according to Eq. (2.3), the end-to-end delay bound of a given flow f is computed using the network-wide service curve applied to the arrival curve of the incoming flow. The worst-case end-to-end delay is equal to the delay bound of a data flow along the longest routing path in the network.

VI. IEEE 802.15.4/ZIGBEE APPLICATION

The aforementioned analysis is independent from any specific protocol. In addition, the proposed model is quite interesting for existing cluster-tree WSN protocols that provide guaranteed services, such as LEACH or IEEE 802.15.4/Zigbee, and it can be easily used for their worst-case dimensioning. In this section, we show the practical applicability of our approach by instantiating the general model proposed in Section 5 for IEEE 802.15.4/Zigbee cluster tree WSNs [7], and provide a methodology for its worst-case dimensioning. The computations are made using MATLAB.

IEEE 802.15.4/ZigBee Protocols Features

The IEEE 802.15.4/ZigBee protocols have several appealing properties for WSNs.The features of the MAC sub layer are beacon management, channel access, GTS management, frame validation, acknowledged frame delivery, association, and disassociation. In addition, the MAC sub layer provides hooks for implementing application-appropriate security mechanisms.

Beacon Interval (BI) is defined as the time interval between two consecutive beacons, and it is divided into an active portion and, optionally, a following inactive portion. The active period, corresponding to the Superframe Duration (SD), is divided into 16 equally-sized time slots. Each active period can be further divided into a Contention Access Period (CAP) and an optional Contention Free Period (CFP). Within the CFP, Guaranteed Time Slots (GTSs) can be allocated to a set of child nodes [8][9]. The CFP supports up to 7 GTSs and each GTS may contain multiple time slots. Each GTS can transfer data either in transmit direction, i.e. from child to parent (upstream flow), or receive direction, i.e. from parent to child.

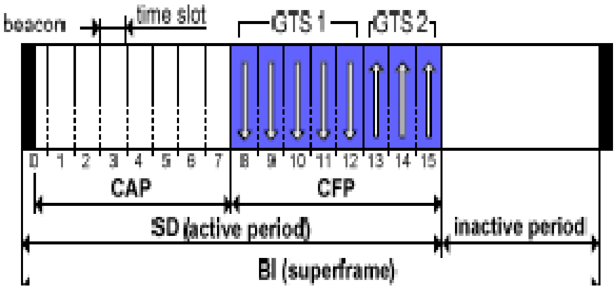


Figure 5: IEEE 802.15.4 Superframe Structure

The structure of the superframe is defined by two parameters, the Beacon Order (BO) and the Superframe Order (SO), as follows:

BI=α Base Superframe Duration. 2^{BO}

SD=αBase Superframe Duration. 2^{SO}

where aBase Superframe Duration = 15.36 ms assuming the 2.4 GHz ISM frequency band with 250 kbps data rate, and $0 \leq SO \leq BO \leq 14$. The TDCS is given by the non-overlapping sequence of equally-sized SDs Figure 5, and the duration of a TDCS cycle is equal to BI.

Guaranteed Bandwidth of a GTS Time Slot

The whole data transmission in a GTS, including the frame, inter-frame spacing (IFS) and potential acknowledgment, must be completed before the end of the GTS. The maximum time required for the whole transmission of a MAC frame, called MPDU (MAC Protocol Data Unit) is then expressed as:

$$T_{MPDU} = MPDU_{\max} / C + IFS + AckWaitDuration.\Omega \tag{6.1}$$

Where $MPDU_{\max}$ is the user defined maximum size of the frame, C is the data rate (we assume 250 kbps), an $\Omega = 1$ for an acknowledged transmission or $\Omega = 0$ for an unacknowledged transmission. The maximum number of MAC frames that can be transmitted during one time slot is expressed as:

$$N_{MPDU} = \left\lfloor \frac{TS}{T_{MPDU}} \right\rfloor \tag{6.2}$$

Where TS is the duration of a time slot and is equal to SD/16.In the remaining time, a frame smaller than $MPDU_{\max}$can be transmitted if the whole transmission can be completed before the end of the GTS. The transmission time of last frame is then expressed as:

$$T_{last} = TS - N_{MPDU}.T_{MPDU} - IFS - AckWaitDuration.\Omega \tag{6.3}$$

Finally, assuming a full duty cycle (i.e. SO = BO) the guaranteed bandwidth of one GTS time slot is expressed as:

$$R_{TS}^{100\%} = \frac{N_{MPDU}.MPDU_{\max} + \max(T_{last}, 0).C}{SD} \tag{6.4}$$

Characterization of the Service Curve

Each parent router must reserve a GTS with enough time slots for each of its child nodes (requiring guaranteed service). For downstream data link, a parent router at depth i must reserve a GTS with N_{iD}^{TS} time slots in receive direction to its child router at depth i+1 such that the resulting link bandwidth is greater than or equal to its total input arrival rate $\bar{r}_{iD}$ .It results that:

$$N_{iD}^{TS} = \left\lceil \frac{\bar{r}_{iD}}{R_{TS}} \right\rceil \tag{6.5}$$

Hence, a GTS with N_{iD}^{TS} time slots provides rate-latency service $\beta_{R_i T_i}(t)$, where $R_i = N_{iD}^{TS}.R_{TS}$ is the guaranteed band width and T_i is the service latency.

According to Figure 5, the worst-case service latency guaranteed to a flow over downstream data link at given depth is expressed as:

- The service latency guaranteed by a router at depth 0 to the child router at depth 1

$$T_{OD} = \left(N_{router}^{MAX} - 1\right).N_{OU}^{TS}.TS \tag{6.6}$$

- The service latency guaranteed by a router at depth i to the child router at depth i+1, for $\forall i, 0 < i < H_{\sin k}$:

$$T_{iD} = BI - SD - \left(N_{iD}^{TS} - N_{(i-1)D}^{TS}\right).TS \tag{6.7}$$

IEEE 802.15.4/ZigBee WSN Setup

The experimental setup consist of a simple cluster-tree WSN corresponding to the configuration where $H = 2$, $N^{MAX}_{end\_node} = 1$, $N^{MAX}_{router} = 2$.For the sake of simplicity, only end-nodes are equipped with sensing capability (i.e. S = 0) and generate data flows bounded by n the arrival curve $\alpha_{data}(t)$.We assume a minimum possible value of SO (e.g. SO = 4), imposed by some technological limitations, namely due to the non-preemptive behavior of the TinyOS operating system. According to Eq. (17), the total number of routers is equal to 7. Hence, BO must be set such that at least 7 SDs with SO = 4 can fit inside the BI without overlapping. In general, we obtain:

$$BI \geq \sum \left(N^{MAX}_{router}, H\right).SD \Leftrightarrow BO_{\min}$$

$$= \left| \log_2 \left(\sum \left(N^{MAX}_{router}\right).2^{SO} \right) \right| \tag{6.8}$$

As a result for SO = 4, the minimum BO is equal to 7, such that a maximum of $2^7 / 2^4 = 8$ SDs can fit in one BI.

The maximum duty cycle of each cluster is then equal to (1/8) = 12.5 %. Note that to maximize the lifetime of a WSN, the lowest duty cycles must be chosen. On the other hand, low duty cycles enlarge end-to-end delays. Hence, long lifetime is in contrast to the fast timing response of a WSN, so a trade-off must be found.

The minimum CAP is equal to 7.04ms, assuming the 2.4 GHz ISM band, which corresponds to 1 time slot with SO = 4. The remaining slots can be allocated for GTSs. Hence, the maximum CFP length is equal to $L_{CFP} = 15$ time slots. A router cannot reserve more than L_{CFP} time slots for 7 GTSs maximum, i.e. for its $N^{MAX}_{end\_node}$ end-nodes and N^{MAX}_{router} child routers.

Assuming that each end-node requires allocation of a GTS with N^{TS}_{data} time slots (i.e. $r_{data} \leq N^{TS}_{data}.R_{TS}$) from its parent router, then each child router can allocate a GTS with the maximum number of time slots equal to:

$$\left\lfloor \left(L_{CFP} - N^{TS}_{data}.N^{MAX}_{end\_node}\right) / N^{MAX}_{router} \right\rfloor$$

According to Eq. (5.1), the arrival rate r_{data} must not exceed the maximum bandwidth that a parent router can reserve. Obviously, due to the cumulative flow effect, the maximum bandwidth will be required by the sink router.

Hence, the corresponding link bandwidth guaranteed by the parent router at depth $H_{\sin k} - 1$to the sink router at $H_{\sin k}$ is equal to:

$$R_{(H_{\sin k}-1)} = \left\lfloor \frac{L_{CFP} - N^{TS}_{data}.N^{MAX}_{end\_node}}{N^{MAX}_{router}} \right\rfloor .R_{TS} \tag{6.9}$$

As a result applying Eq. (5.1), we obtain the maximum arrival rate of the sensory data flow as:

$$r_{data}^{MAX} = \frac{\left\lfloor \dfrac{L_{CFP} - N_{data}^{TS}.N_{end\_node}^{MAX}}{N_{router}^{MAX}} \right\rfloor R_{TS}}{\left(\sum_{j=0}^{\left(H_{\sin k}-1\right)} \left(N_{router}^{MAX}\right)^{H-j} \right).\left(N_{end\_node}^{MAX} + S_{ij}\right)}$$

(6.10)

Note that the aforementioned expressions are valid for $\forall H_{\sin k}, 1 \le H_{\sin k} \le H$ .The expressions for $H_{\sin k} = 0$ is

$$r_{data}^{MAX} = \left\lfloor \frac{L_{CFP} - N_{child}}{N_{router}} \right\rfloor . \frac{R_{TS}}{\gamma\left(N_{router}\right).\left(N_{child}+1\right)} \quad (6.11)$$

As a result for $L_{CFP} = 14$,we get $r_{data}^{\max} = 0.104$ kbps .The value of burst tolerance b_{data} is selected according to the burstiness of sensory data.

VII. PERFORMANCE EVALUATION

This section compares the analytical results based on Network Calculus with the experimental results obtained through the use of IEEE 802.15.4/ZigBee technologies. The analytical results are computed using a Matlab tool. and the experimental results are obtained using a test-bed based on the TelosB motes.

Network Setup

The experimental test-bed in Figure 6, consists of 7 clusters and 14 TelosB motes running the TinyOS 1.x operating system with open source implementation of the IEEE 802.15.4/ZigBee protocol stack. The TelosB is a battery-powered wireless module with integrated sensors, IEEE 802.15.4 compliant radio, antenna, low power 16-bit RISC microcontroller, and programming capability via USB. For debugging purposes, it has been used the Chipcon CC2420 packet snider that provides a raw list of the transmitted packets, and the Dain tree Sensor Network Analyzer (SNA) that provides additional functionalities, such as displaying the graphical topology of the network. Set the application running on the sensor nodes to generate 3 bytes at the data payload. Hence, the maximum size of the MAC frame is equal to $MPDU_{\max} = 192$ bits.

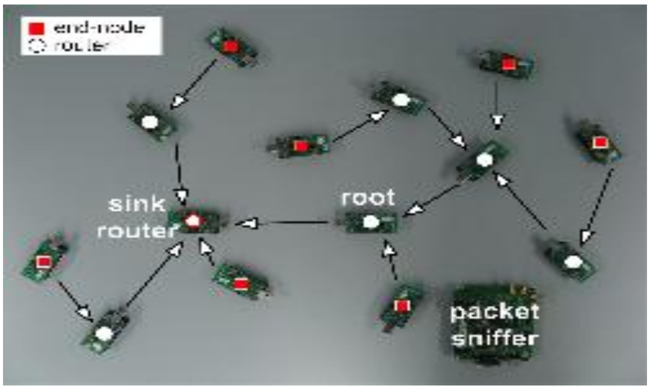


Figure 6: The Test-Bed Deployment for $H_{\sin k} = 1$

The problem has been already reported and fixed in TinyOS 2.x. Since our implementation of IEEE 802.15.4/ZigBee protocol stack was built over TinyOS 1.x, we overcame the aforementioned problem by setting the inter-frame spacing (IFS) time (i.e. time between two consecutive frames) such that no frame arrives during the frame processing times. The experimental value of IFS equal to 3.07ms was measured. According to Eq. (20), the bandwidth guaranteed by one time slot for SO = 4 is equal to 3.125 kbps with 100% duty cycle. Hence, in our experimental scenario with a 12.5 % duty cycle

$$BO = BO_{\min} = 7$$

The guaranteed bandwidth of one time slot is equal to RTS = 3.125 · 0.125 = 0.3906 kbps. Assume $N_{data}^{TS} = 1$ Then according to Eq. (6.10), we obtain the maximum arrival rates of the sensory data flow as follows:

- $r_{data}^{MAX} = 456$ bps for $H_{\sin k} = 2$
- $r_{data}^{MAX} = 684$ bps for $H_{\sin k} = 1$
- $r_{data}^{MAX} = 911$ bps for $H_{\sin k} = 0$ (root)

As a result of $r_{data} \leq \min\left(r_{data}^{MAX}\right)$ and $r_{data} \leq R_{TS}$, an average arrival rate equal to $r_{data} = 390$ bps, which corresponds to 4 frames (192 bits each) generated during one Beacon Interval (BI = 1.96608 sec), is considered. The burst tolerance is assumed to be equal to $b_{data} = 576$ bits, which corresponds to 3 frames generated at once. Hence, each sensor node transmits sensory data bounded by the arrival curve $\alpha_{data} = 576 + 390.t$ .Note that Network Calculus based analytical model is bit-oriented, which means that sensory data are handled as a continuous bit stream with data rate r_{data} ,while the experimental test-bed is frame-oriented, where data traffic is organized in frames of a given size. The frames can be generated at constant bit rate (CBR) or variable bit rate (VBR), but all data traffic must be upper bounded by the arrival curve α_{data} .Finally, let us consider the complete network setting:

- $N_{router}^{MAX} = 2$
- $N_{end\_node}^{MAX} = 1$
- $H = 2$
- $SO = 4$ (SD = 245.76 ms)
- BO = 7 (BI = 1966.08 ms)
- Duty Cycle = 12.5 %
- $MPDU_{\max} = 192$ bits
- $r_{data} = 390$ bits
- $b_{data} = 576$ bits

- IFS = 3.07 ms
- $L_{CFP} = 15$
- S=0

It is assumed the non-overlapping worst-case TDCS given by the following sequence of active portions of clusters 11, 01, 12, 24, 23, 21, 22. Note that the unacknowledged transmission is only assumed

Experimental vs. Theoretical Results

Buffer Requirements

The theoretical worst-case buffer requirements as compared to the maximum values obtained through real experimentation, for different $H_{\sin k}$ values in figure 8.

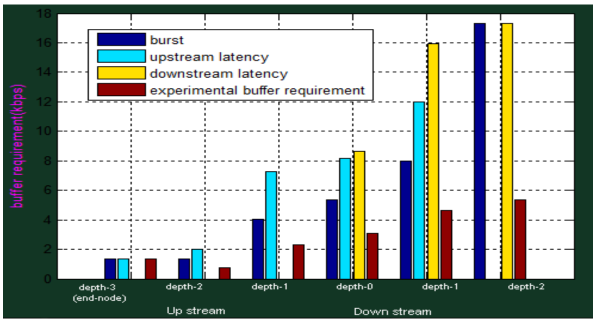
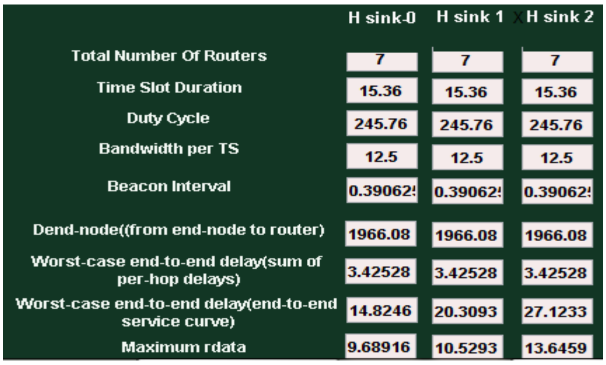


Figure 7: Buffer Requirements



	H sink-0	H sink 1	H sink 2
Total Number Of Routers	7	7	7
Time Slot Duration	15.36	15.36	15.36
Duty Cycle	245.76	245.76	245.76
Bandwidth per TS	12.5	12.5	12.5
Beacon Interval	0.39062!	0.39062!	0.39062!
Dend-node((from end-node to router)	1966.08	1966.08	1966.08
Worst-case end-to-end delay(sum of per-hop delays)	3.42528	3.42528	3.42528
Worst-case end-to-end delay(end-to-end service curve)	14.8246	20.3093	27.1233
Maximum rdata	9.68916	10.5293	13.6459

Figure 8: Experimental Values for Different H_{sink}

The numerical values of theoretical worst-case as well as experimental maximum buffer requirements are summarized in figure 7. The bandwidth requirements given by Eq. (5.1) and the corresponding number of time slots are also presented.

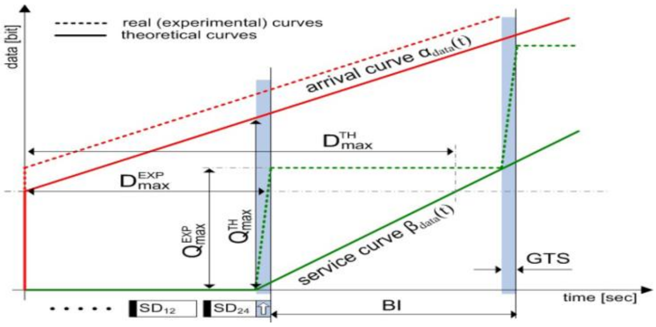


Figure 9: Theoretical vs. Experimental Data Traffic

Delay Bounds

Figure 10 compares the worst-case, maximum and average values of per-hop delays bound in each router and the end-to-end delay bounds for $H_{\sin k} = 2$ .A worst observation confirms that theoretical results upper bound the experimental results.

The difference in theoretical worst-case $D_{\max}^{TH}$ and experimental maximum $D_{\max}^{EXP}$ delays is given by the aforementioned continuous and stepwise behaviors of the analytical model and test-bed, respectively. The experimental delays comprise mainly the service latencies. Hence, the maximum per-hop delays also decrease in the direction of the sink, as can be observed in Figure 9.

Figure 10 presents the worst-case, maximum and average numerical values of per-hop and per-flow delay bounds, and the end-to-end delays for different sink positions. Note that the average values were computed from a set of 15 runs, involving the transmission of 1155 frames each. The theoretical worst-case end-to-end delays are obtained as the sum of per-hop delays using Eq. (5.4), or by per-flow approach, which results in the family of service curves as a function of $\Theta \geq 0$.

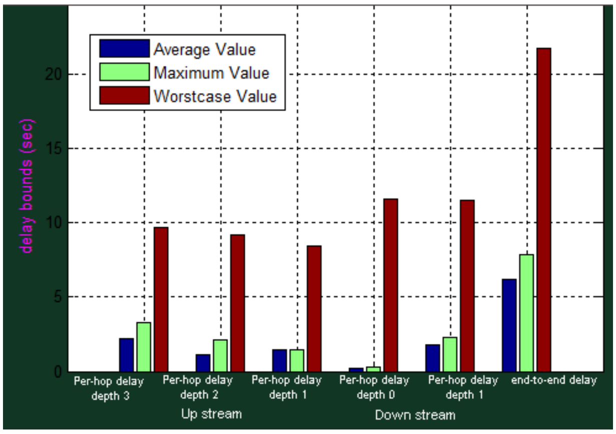


Figure 10: The Theoretical vs. Experimental Delay Bounds

This analysis assumes $\Theta = T + (b_2 / R)$ as a trade-o_ between computation complexity and optimality. The determination of the optimal service curve, leading to the lowest worst-case delay, will be addressed in future work.

CONCLUSIONS

Modeling the fundamental performance limits of Wireless Sensor Networks (WSNs) is of paramount importance to understand their behavior under the worst-case conditions and to make the appropriate design choices. In that direction this chapter contributes with a methodology based on Network Calculus, which enables quick and efficient worst-case analysis and dimensioning of static or even dynamically changing cluster-tree WSNs where the data sink can either be static or mobile, i.e. can be associated to any router in the WSN. The proposed analytical methodology enables to guarantee the routers' buffer size to avoid buffer overflows and to minimize clusters' duty-cycle (maximizing nodes' lifetime) still satisfying that messages' deadlines are met.

The future work includes improving the current methodology to encompass clusters operating at different duty-cycles and to provide a model that enables real-time control actions, i.e. the sink assuming the role of controlling sensor/actuator nodes.

REFERENCES

1. Koubaa, M. Alves, E. Tovar, “Modeling and Worst-Case Dimensioning of Cluster-Tree Wireless Sensor Networks,” In Real Time Systems Symposium (RTSS’06), Brazil, Dec. 2006.

2. J. B. Schmitt, F. Zdarsky, L. Thiele, “A Comprehensive Worst- Case Calculus for Wireless Sensor Networks with In-Network Processing,” In IEEE Real-Time Systems Symposium (RTSS’07), USA, Dec. 2007.

3. Koubaa, M. Alves, E. Tovar, “IEEE 802.15.4: a Federating Communication Protocol for Time-Sensitive Wireless Sensor Networks,“ Chapter of the book Sensor Networks and Configurations: Fundamentals, Techniques, Platforms, and Experiments, Springer-Verlag, Germany, pp. 19-49, Jan. 2007.

4. S. Prabh, T. F. Abdelzaher, “On Scheduling and Real-Time Capacity of Hexagonal Wireless Sensor Networks, “In Euromicro Conference on Real-Time Systems (ECRTS’07), Italy, July 2007.

5. J. B. Schmitt, F. Zdarsky, L. Thiele, “A Comprehensive Worst- Case Calculus for Wireless Sensor Networks with In-Network Processing,” In IEEE Real-Time Systems Symposium (RTSS’07), USA, Dec. 2007.

6. A. Cunha, R. Severino, N. Pereira, A. Kouba, M. Alves, “ZigBee over TinyOS: implementation and experimental challenges, “In the 8th Portuguese Conference on Automatic Control (CONTROLO 2008), Vila Real, Portugal, 2008.

7. A. Koubaa, A. Cunha, M. Alves, “A Time Division Beacon Scheduling Mechanism for IEEE 802.15.4/ZigBee Cluster-Tree Wireless Sensor Networks,” In Euromicro Conference on Real- Time Systems (ECRTS’07), Italy, July 2007.

8. J. Schmitt and U. Roedig, "Sensor Network Calculus - A Framework for Worst Case Analysis," in Proceedings of the IEEE/ACM International Conference on Distributed Computing in Sensor Systems (DCOSS'05), LNCS 3560, Marina del Rey, USA, 2005.

9. S.-e. Yoo, D. Kim, M.-L. Pham, Y. Doh, E. Choi, and J.-d. Huh, "Scheduling Support for Guaranteed Time Services in IEEE 802.15.4 Low Rate WPAN," in Proceedings of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'05), Hong Kong (CHINA), 2005