IPP HURRAY!

www.hurray.isep.ipp.pt

Technical Report

On the Capacity of Cluster-tree ZigBee Networks

Petr Jurcik

Anis Koubaa

Mario Alves

Eduardo Tovar

HURRAY-TR-091202

Version:

Date: 12-15-2009

On the Capacity of Cluster-tree ZigBee Networks

Petr Jurcik, Anis Koubaa, Mario Alves, Eduardo Tovar

IPP-HURRAY!

Polytechnic Institute of Porto (ISEP-IPP)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8340509

E-mail:

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

Abstract

Modeling the fundamental performance limits of Wireless Sensor Networks (WSNs) is of paramount importance to understand their behavior under worst-case conditions and to make the appropriate design choices. In that direction this paper contributes with an analytical methodology for modeling cluster-tree WSNs where the data sink can either be static or mobile. We assess the validity and pessimism of analytical model by comparing the worst-case results with the values measured through an experimental test-bed based on Commercial-Off-The-Shelf (COTS) technologies, namely TelosB motes running TinyOS.

On the Capacity of Cluster-tree ZigBee Network

Petr Jurčík*†, Anis Koubâa*‡, Mario Alves*, Eduardo Tovar*
*IPP-HURRAY Research Group, CISTER/ISEP, Polytechnic Institute of Porto, Porto, Portugal
†Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic
‡Al-Imam Muhammad Ibn Saud University, College of Computer Science and Information Systems, Riyadh, Saudi Arabia
{petr, aska, mjf, emt}@isep.ipp.pt

***Abstract*—Modeling the fundamental performance limits of Wireless Sensor Networks (WSNs) is of paramount importance to understand their behavior under worst-case conditions and to make the appropriate design choices. In that direction this paper contributes with an analytical methodology for modeling cluster-tree WSNs where the data sink can either be static or mobile. We assess the validity and pessimism of analytical model by comparing the worst-case results with the values measured through an experimental test-bed based on Commercial-Off-The-Shelf (COTS) technologies, namely TelosB motes running TinyOS.**

***Keywords*-cluster-tree; IEEE 802.15.4; ZigBee; worst-case dimensioning**

I. Introduction

The evaluation of the performance limits of Wireless Sensor Networks (WSNs) is a crucial task, particularly when the network is expected to operate under the worst-case conditions [1]. For achieving predictable resource guarantees (e.g. bandwidth and buffer size) all over the sensor networks, it is mandatory to rely on structured logical topologies such as cluster-tree with deterministic routing protocol, and collision-free access to the shared communication medium. Hence, in this paper, we aim at the performance limits of the worst-case cluster-tree WSN, i.e. the configuration of WSN that leads to the worst-case performance.

The nodes in cluster-tree WSNs are organized in logical groups, called *clusters*. Each node is connected to a maximum of one node at the lower depth, called *parent* node, and can be connected to multiple nodes at the upper depth, called *child* nodes (by convention, trees grow down). The routers and end-nodes are two types of wireless nodes in cluster-tree WSNs (refer to Fig. 1). The nodes that can participate in multi-hop routing are referred to as *routers* (R_{ij}, i.e the j^{th} router at depth i). The nodes that do not allow association of other nodes and do not participate in routing are referred to as *end-nodes* (N). The router that has no parent is called *root*. Each router forms a cluster and is referred to as its *cluster-head* (e.g. router

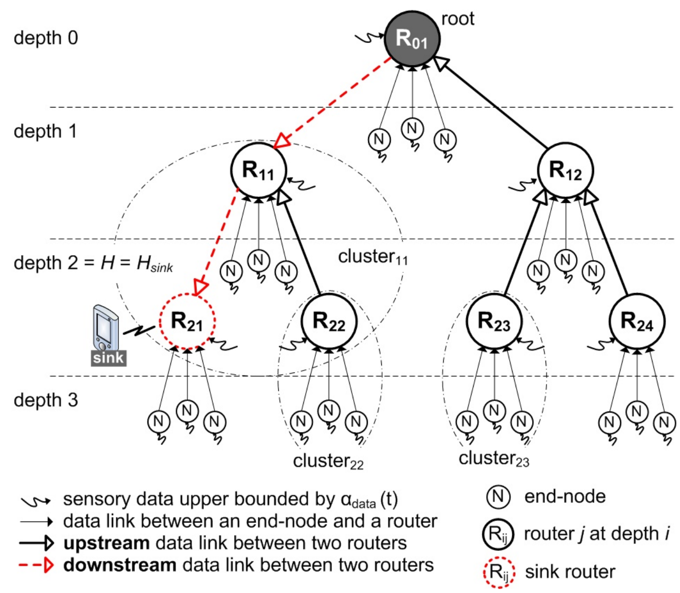


Figure 1. The worst-case cluster-tree topology corresponding to a configuration where $N^{MAX}_{end-node} = 3$, $N^{MAX}_{router} = 2$, $H_{sink} = 2$, and $H = 2$.

R_{11} is the cluster-head of cluster$_{11}$). All of its child nodes (i.e. end-nodes and routers) are associated to the cluster, and the cluster-head handles all their transmissions. The depth of a node is defined as the minimal number of logical hops from that node to the root. Note that the root is at depth zero.

The worst-case cluster-tree topology is graphically represented by a rooted balanced directed tree [2] defined by the following three parameters:

- H — Height of the cluster-tree, i.e. the maximum number of logical hops for a message from the deepest router to reach the root (including the root as a final hop).
- $N^{MAX}_{end-node}$ — Maximum number of end-nodes that can be associated to a router.
- N^{MAX}_{router} — Maximum number of routers that can be associated to a router.

Note that while a static or even dynamically changing practical cluster-tree WSN can assume different configurations, these can never exceed the worst-case topology, in terms of

This work was partially funded by FCT under the CISTER Research Unit (FCT UI 608), PLURALITY and CMU-PT projects, ARTIST2/ARTISTDesign NoEs., and by the Czech Republic MPO project (61 03001).

maximum depth and number of child routers/end-nodes.

We consider the sink as an autonomous entity that does not make part of the static cluster-tree WSN, but can be associated to any of its routers through any (wired or wireless) communication means. The router to which the sink is associated in a given moment is referred to as *sink router*. Hence, we specify another parameter, $H_{sink} \in (0, H)$, to represent the maximum depth of the sink router in a cluster-tree WSN.

We assume that all sensory data is sent 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 upper bounded data. In cluster-tree WSNs where the sink can be associated to any router, data may flow in the upstream and downstream directions. In the upstream case, data is sent from the child nodes to its parent router (so called *upstream flow*), and the parent router must reserve enough bandwidth for the outgoing data of its child nodes. On the contrary, in the downstream case, data is sent from a parent router to its child router (so called *downstream flow*), and the parent router must reserve enough bandwidth for its own outgoing data.

A. Related work

The evaluation of the fundamental performance limits of WSNs has been addressed in several research works. In [1], the energy-constrained limits of WSNs with respect to the network throughput and operational lifetime has been evaluated. In [3], the authors have evaluated the real-time capacity of multi-hop WSNs, identifying how much real-time data the network can transfer by their deadlines. A capacity bound has been derived for load-balanced as well as load-unbalanced sensor networks using (ideal) MAC protocols with fixed priority packet scheduling mechanisms. Both above mentioned papers consider unstructured WSNs with ad-hoc deployment.

The worst-case analysis and resource dimensioning of WSNs using Network Calculus has been pursued by Schmitt et al., who proposed the Sensor Network Calculus methodology. In [4], Sensor Network Calculus was introduced and basic components such as arrival and service curves were defined. The system model assumes generic tree-based topologies with nodes transmitting sensor data towards the sink, that is associated to the root. The authors also proposed a general iterative procedure to compute the network internal flows and, subsequently, the resource requirements and the delay bounds. On the contrary, our work provides recurrent equations so that to avoid iterative computations that are more complex and time consuming and not suitable for large-scale WSNs. In [5], the previous Sensor Network Calculus framework was extended to incorporate in-network processing features (e.g. data aggregation). In our work, we abstract from the computational resources in the network nodes and from data aggregation. In [6], the authors searched for the worst-case topology (i.e. the topology that exhibits the worst-case behavior in terms

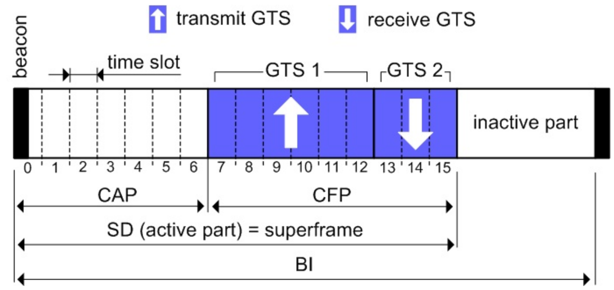


Figure 2. IEEE 802.15.4 superframe structure with two GTSs.

of buffer requirements, delay bounds and network lifetime) in networks with random nodes deployment. As compared to the aforementioned papers, our system model is more accurate for the specific case of cluster-tree topologies and the sink can be associated to any router in the WSN.

II. Overview of the IEEE 802.15.4/ZigBee Protocols

The IEEE 802.15.4/ZigBee [7], [8] protocols have recently been adopted as a communication standard for low data rate, low power consumption and low cost WSNs. The IEEE 802.15.4 [7] standard specifies the physical and data link layers, while the network and application layers are defined by the ZigBee [8] specification. The MAC (Medium Access Control) supports the beacon-enabled or non beacon-enabled modes that may be selected by a central controller of the WSN, called the *PAN coordinator*. We only consider the beacon-enabled mode, since it enables the provision of the guaranteed bandwidth through the *Guaranteed Time Slot* (GTS) mechanism.

While IEEE 802.15.4 in the beacon-enabled mode supports only the star-based topologies, the ZigBee specification has proposed its extension to mesh and cluster-tree topologies. In the particular case of ZigBee cluster-tree WSNs, a PAN coordinator is identified as the root of the tree and forms the initial cluster. The other routers join the cluster-tree in turn by establishing themselves as cluster-heads, starting to generate the beacon frames for their own clusters. Beacon frames are periodically sent at *Beacon Interval* (BI) to synchronize the child nodes that are associated with a given cluster-head and to describe the superframe structure (Fig. 2). Time between two consecutive beacons is divided to an active part, during which the cluster-head enables the data transmissions inside its cluster, and, optionally, subsequent an inactive part.

During the inactive part, each associated node may enter a low-power mode to save energy. The active part, corresponding to the *Superframe Duration* (SD), is divided into 16 equally-sized time slots, during which the data transmission is allowed. The active part can be further divided into a *Contention Access Period* (CAP) using slotted CSMA/CA for the best effort data delivery, and an optional *Contention Free Period* (CFP) supporting the time-bounded data delivery. Within the CFP, Guaranteed Time Slots can be allocated to a set of the child nodes. The CFP supports up to 7 GTSs and each GTS may contain one or more time slots. Each node may request up to one GTS in the *transmit direction*, i.e. from the child node to the parent router (upstream flow), and/or one GTS in the *receive direction*, i.e. from the parent router to the child node (downstream flow).

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

$$\begin{aligned} BI &= aBaseSuperframeDuration \cdot 2^{BO} \\ SD &= aBaseSuperframeDuration \cdot 2^{SO} \end{aligned} \quad (1)$$

where *aBaseSuperframeDuration* = 15.36 ms (assuming the 2.4 GHz frequency band and 250 kbps of bit rate) and denotes the minimum length of the superframe when $SO = 0$, and $0 \leq SO \leq BO \leq 14$. Note that the ratio of SD to BI is called duty-cycle (DC).

III. Models

A. Analytical model

In [9] and [10], we have proposed and derived simple yet efficient analytical methodology based on *Network Calculus* that enable the worst-case dimensioning of a static or even dynamically changing cluster-tree WSNs and the evaluation of the performance bounds for upstream and downstream flows, respectively. We have also showed how it is possible to guarantee the routers' buffers size to avoid overflows and to minimize clusters' duty-cycle (maximizing nodes' lifetime), still satisfying that messages' deadlines are met.

Network Calculus [11] is a mathematical methodology based on min-plus algebra that applies to the deterministic analysis of queuing/flows in communication networks. In Network Calculus, it is possible to bound the incoming and outgoing data flows of each node by *arrival curve* $\alpha(t) = b + r \cdot t$, where b is the burst tolerance and r is the average data rate, and *service curve* $\beta(t) = R \cdot (t - T)^{+}$, where $R \geq r$ is the guaranteed forwarding rate and T is the maximum service latency, respectively. Note that sensory flow of every sensor node is upper bounded by the same arrival curve $\alpha_{data} = b_{data} + r_{data} \cdot t$ which bounds the highest sensory flow in the network.

The knowledge of the arrival and service curves enables to determine the performance bounds for a lossless FIFO node, namely the *delay bound* D_{max}, which represents the worst-case delay of a message traversing the node, and the *backlog bound* Q_{max}, which represents the worst-case queue length of a flow, i.e. indicates the minimum buffer size requirement inside the node.

The worst-case *end-to-end delay* (D_{e2e}) is the delay bound of a data flow transmitted along the longest path in the network. It can be computed using *per-hop* or *per-flow* approach [10]. The former is computed as the sum of the delay bounds of all nodes on the longest path. This approach is a bit pessimistic, since the delay bound at each node is computed for the aggregation of all incoming flows. Tighter end-to-end delay bounds can be computed for individual flows, i.e. per-flow approach.

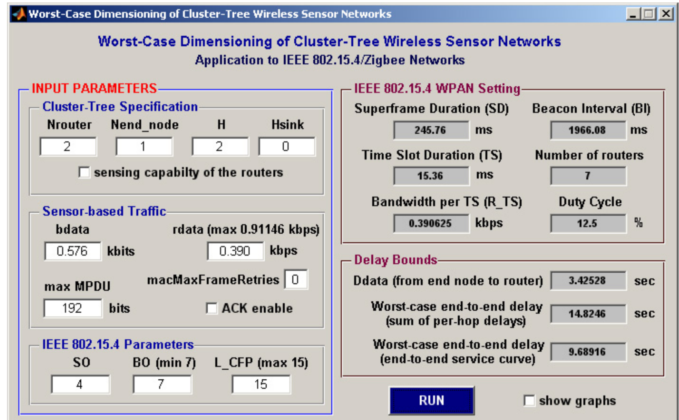


Figure 3. The GUI of the MATLAB analytical tool.

In cluster-tree WSNs, messages are forwarded from cluster to cluster until reaching the sink. The clusters may have collisions when they overlap. To avoid these collisions, it is mandatory to schedule the clusters' active parts (i.e. SDs) in an ordered sequence. In [10], we have elaborated on the worst-case cluster scheduling; that is, the time sequence of clusters' active periods leading to the worst-case end-to-end delay for a message to be routed to the sink. For the sake of simplicity, in this paper, we assume that all clusters have the same duty cycle, and whole WSN is inside one collision domain, i.e. the non-overlapping sequence of equally-sized SDs of duration equal to BI.

The analytical framework [12] was implemented as the MATLAB tool (Fig. 3) that enables WSN designers to efficiently predict network resources so that to guarantee a minimum performance.

B. Experimental model

The experimental test-bed (illustrated in Fig. 4) consists of 7 clusters and 14 TelosB motes running the TinyOS 1.x [13] operating system with open source implementation of the IEEE 802.15.4/ZigBee protocol stack [14]. 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, we have used the Chipcon CC2420 packet sniffer [15] that provides a raw list of the transmitted packets, and the Daintree Sensor Network Analyzer (SNA) [16] that provides additional functionalities, such as displaying the graphical topology of the network.

Note that, in practice, this experimental deployment could span over a wider region than the one illustrated in Fig. 4, provided that every end-node and child router is within radio

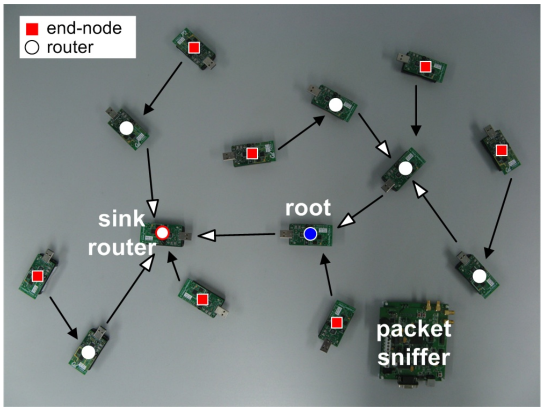


Figure 4. The test-bed deployment for $H_{sink} = 1$.

range of its parent router (TelosB radio range is around several tens meters). Number of end-nodes associated to each router can also be higher (not all nodes might need guaranteed bandwidth).

We assume $SO = 4$, which is the minimum value that is possible to use without resulting into synchronization problem [17], using open-ZB protocol stack [14] over TinyOS [13] and MICAz/TelosB motes. This constraint results from the non-preemptive behavior of the TinyOS operating system.

TinyOS 1.x flushes the reception buffer of the radio transceiver after processing the first arriving frame. Thus, the frames that arrive during the processing time of the first frame are discarded. This 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 time. A value of IFS equal to 3.07 ms was measured.

IV. Performance Evaluation

In this section, we compare the analytical results based on Network Calculus with the experimental results obtained through the use of IEEE 802.15.4/ZigBee technologies.

We assume the worst-case cluster-tree topology with network setting as follows:

$N_{router}^{MAX} = 2$	$r_{data} = 390$ bps
$N_{end-node}^{MAX} = 1$	$b_{data} = 576$ bits
$H = 2$	IFS = 3.07 ms
$SO = 4$ (SD = 245.76 ms)	$BO = 7$ (BI = 1966.08 ms)

For detailed results please refer to [9], [10].

A. Buffer requirements

Figure 5 shows the theoretical worst-case buffer requirements compared with the maximum values obtained through real experimentation, for $H_{sink} = 2$. First, it can be observed that end-nodes have the smallest buffer requirement as they are the leaves of the tree, and that the buffer requirement grows in the direction of the sink router. The next observation confirms that the theoretical values upper bound the experimental values. The pessimism of the theoretical bounds is justified by the fact that the Network Calculus analytical model is based on a continuous approach (arrival and service curves are continuous) in contrast to the real stepwise behavior of flows and services (in the test-bed). In practice, the data is actually transmitted only during its GTS, while in the analytical model we consider a continuous data flow during the whole BI, since it represents the average rate and not the instantaneous rate.

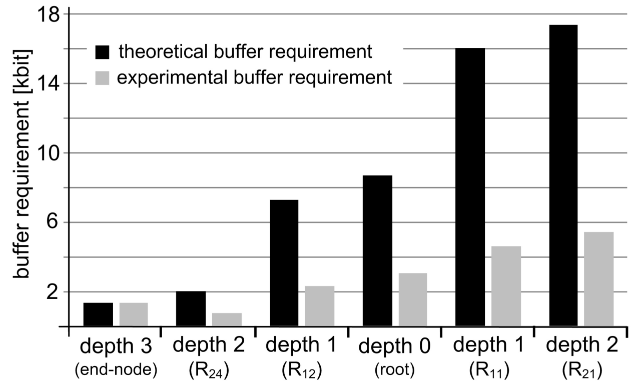


Figure 5. The theoretical vs. experimental buffer requirements.

B. Delay bounds

In Figure 6, we compare the worst-case, maximum and average values of per-hop delays bound in each router, and the end-to-end delay bounds for $H_{sink} = 2$.

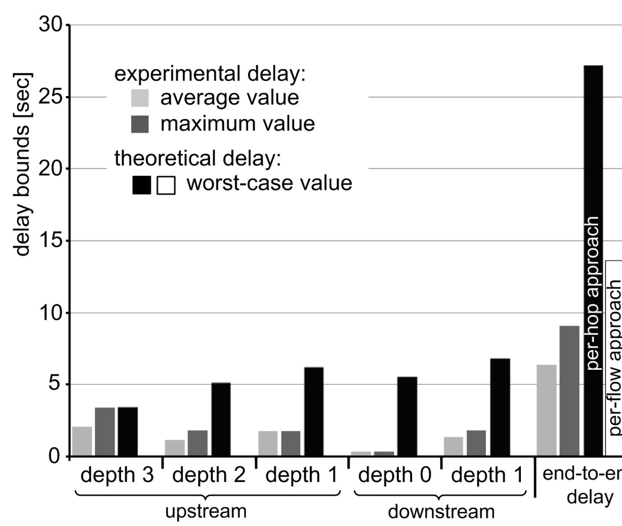


Figure 6. The theoretical vs. experimental delay bounds.

A first observation confirms that theoretical results upper bound the experimental results. The difference in theoretical

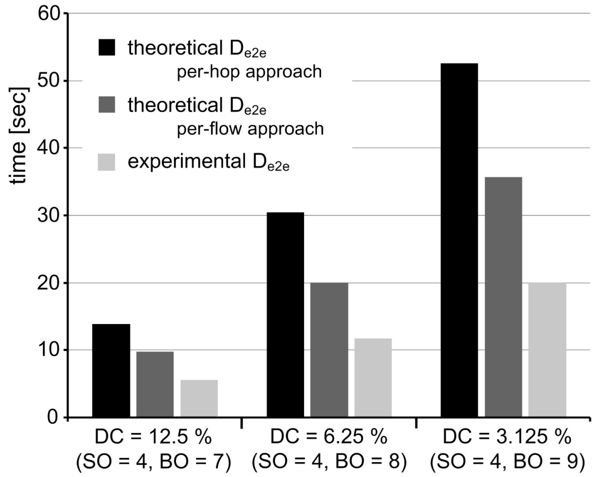


Figure 7. The theoretical worst-case and experimental maximum end-to-end delays as a function of duty-cycle for $H_{sink} = 0$.

worst-case and experimental maximum 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 T decreasing in the direction of the sink. Hence, the maximum per-hop delays also decrease in the direction of the sink, as can be observed in Fig. 6.

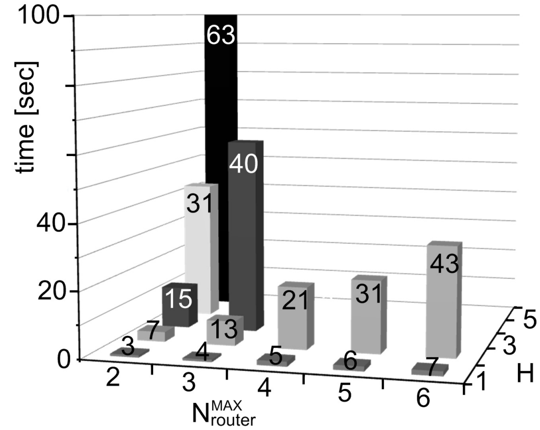
The end-to-end delays bounds are quite high, even though the b_{data} and r_{data} are low. This is mainly due to high value of $SO = 4$ (i.e. BI = 1.966 sec). Hence, the end-to-end delay bounds can be reduced using lower values of SO or higher bandwidth guarantees, using lower IFS, for example. Observe also that the worst-case end-to-end delay obtained by the per-flow approach introduces less pessimism than the delay from the per-hop approach (roughly 50% smaller).

C. Duty-cycle vs. Timing Performance

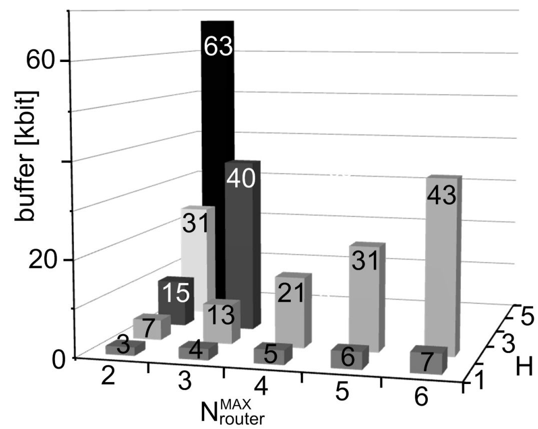
To maximize the lifetime of a WSN, low duty-cycles (DCs) are required (IEEE 802.15.4 supports duty-cycles under 1%). On the other hand, low duty-cycles enlarge the timing performance of a WSN. Our assumptions were confirmed as depicted in Fig. 7, which shows the theoretical worst-case and experimental maximum end-to-end delays (D_{e2e}) as a function of duty-cycle (DC) for $H_{sink} = 0$. The value of SO is set to 4 and the decreasing duty-cycles are obtained by increasing BO. To avoid the lack of bandwidth for smaller duty-cycles, the average arrival rate must be reduce to $r_{data} = 0.190$ kbps. The other network settings are the same as in previous experiments.

D. Network planning

Our analytical methodology can be used for the planning of the cluster-tree WSN as well. Let us consider the example of a convergecast application gathering sensory data at the root (i.e. $H_{sink} = 0$) and using the network settings as mentioned in Section IV. However, in this case, the largest



(a) The worst-case end-to-end delay.



(b) The buffer requirement of the sink router.

Figure 8. The worst-case delay and buffer requirement as a function of N_{router}^{MAX} and H.

feasible configuration of the worst-case cluster-tree topology is achieved for $N_{router}^{MAX} = 2$ and $H = 2$. This means that a feasible worst-case cluster-tree topology given by the parameters N_{router}^{MAX} and H satisfies the network constraints given by the other parameters.

To obtain more illustrative results, we reduce the length of the IFS to the minimum value defined by 802.15.4 standard (i.e. 0.64 ms), $r_{data} = 25$ bps, $SO = 2$, and keep the other parameters. Figure 8a presents the worst-case end-to-end delay and Figure 8b buffer requirement of the sink router as a function of the height of the tree H and the maximum number of child routers N_{router}^{MAX}. In other words, Figure 8 presents all feasible configurations of the worst-case cluster-tree topology, which satisfy a given network constraints. The numerical values at the columns represent the total number of routers in the network. It can be observed that there can be more feasible configurations for the same number of routers. For instance, the total number of 31 routers can be achieved with two configurations, namely $H = 2$ and $N_{router}^{MAX} = 5$ or $H = 4$ and $N_{router}^{MAX} = 2$. The buffer requirements at the sink router are almost the same for both configurations, but the first configuration provides around half of the worst-case end-to-end delay ($D_{e2e} = 22.18$ sec) compared with the second configuration ($D_{e2e} = 43.15$ sec). On the other side, the cluster-topology using the second configuration can spread out over a larger area due to the higher height H. So the system designer must find a trade-off for a given application-specific implementation.

V. Conclusions and future work

This paper shows how to support time-bounded communications in cluster-tree Wireless Sensor Networks (WSNs). We tackled the worst-case analysis and dimensioning of cluster-tree WSNs assuming that the data sink can be static or mobile, i.e. can be associated to any router in the WSN. We proposed the worst-case system model, an analytical methodology and a software tool that enable system designers to analyze and dimension these networks. In this way, it is possible 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.

We developed a multiple cluster test-bed based on Commercial-Off-The-Shelf technologies, namely TelosB motes [18] running open-ZB protocol stack [14] over TinyOS [13]. This test-bed enabled us to assess the validity and pessimism of our worst-case theoretical results (buffer requirements and message end-to-end delays), by comparing these to the maximum and average values measured in the experiments.

Ongoing and 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] Z. Hu and B. Li, *to appear in Ad Hoc and Sensor Networks, Yang Xian and Yi Pan, Editors*. New York, USA: Nova Science Publishers, 2004.

[2] R. Diestel, *Graph Theory*. Springer-Verlag, 2005.

[3] T. Abdelzaher, S. Prabh, and R. Kiran, "On real-time capacity limits of multihop wireless sensor network," in *Proceedings of the 25th IEEE International Real-Time Systems Symposium (RTSS)*. Washington, DC, USA: IEEE Computer Society Press, Dec. 2004, pp. 359–370.

[4] J. Schmitt and U. Roedig, "Sensor network calculus - a framework for worst case analysis," in *Proceedings of the 1st IEEE/ACM Conf. on Distributed Computing in Sensor Systems (DCOSS)*. Washington, DC, USA: IEEE Computer Society Press, Jun. 2005, pp. 141–154.

[5] J. Schmitt, F. Zdarsky, and L. Thiele, "A comprehensive worst-case calculus for wireless sensor networks with in-network processing," in *Proceedings of the 28th IEEE Real-Time Systems Symposium (RTSS)*. Washington, DC, USA: IEEE Computer Society Press, Dec. 2007, pp. 193–202.

[6] J. Schmitt and U. Roedig, "Worst case dimensioning of wireless sensor networks under uncertain topologies," in *Proceedings of the 1st Workshop on Resource Allocation in Wireless NETworks (RAWNET)*. Washington, DC, USA: IEEE Computer Society Press, Apr. 2005.

[7] *Part 15.4: wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs)*, IEEE SA Standards Board Std. 802.15.4, 2006.

[8] *ZigBee Specification*, ZigBee Standards Organization Std. 053 474r13, 2006.

[9] A. Koubaa, M. Alves, and E. Tovar, "Modeling and worst-case dimensioning of cluster-tree wireless sensor networks," in *Proceedings of the 27th Real Time Systems Symposium (RTSS)*. Washington, DC, USA: IEEE Computer Society Press, Dec. 2006, pp. 412–421.

[10] P. Jurcik, R. Severino, A. Koubaa, M. Alves, and E. Tovar, "Real-time communications over cluster-tree sensor networks with mobile sink behaviour," in *Proc. of the 14th IEEE International Conf. on Embedded and Real-Time Computing Systems and Applications (RTCSA)*. Washington, DC, USA: IEEE Computer Society Press, Aug. 2008, pp. 401–412.

[11] J. L. Boudec and P. Thiran, *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet Lecture Notes in Computer Science)*. New York, USA: Springer-Verlag, 2004.

[12] P. Jurcik. (2008) Matlab tool for the worst-case dimensioning of IEEE 802.15.4/ZigBee cluster-tree WSNs. [Online]. Available: http://www.open-zb.net/downloads.php

[13] TinyOS. (2009) TinyOS open-source OS for wireless embedded sensor networks. [Online]. Available: http://www.tinyos.net

[14] A. Cunha, A. Koubaa, R. Severino, and M. Alves, "Open-ZB: an open source implementation of the IEEE 802.15.4/ZigBee protocol stack on TinyOS," in *Proc. of the 4th IEEE International Conf. on Mobile Ad-hoc and Sensor Systems (MASS)*, Oct. 2007.

[15] Chipcon. (2009) C2420DK development kit datasheet. [Online]. Available: http://www.ti.com

[16] Daintree Networks. (2009) aintree sensor network analyzer (SNA). [Online]. Available: http://www.daintree.net

[17] A. Cunha, R. Severino, N. Pereira, A. Koubaa, and M. Alves, "ZigBee over TinyOS: implementation and experimental challenges," in *Proc. of the 8th Portuguese Conf. on Automatic Control (CONTROLO)*. Portugal: UTAD, Jul. 2008, pp. 911–916.

[18] Crossbow. (2009) TelosB mote datasheet. [Online]. Available: http://www.xbow.com