Journal Paper

A Cross-layer QoS Management Framework for ZigBee Cluster-Tree Networks

Ricardo Severino
Sana Ullah
Eduardo Tovar

CISTER-TR-160102

2016/11

A Cross-layer QoS Management Framework for ZigBee Cluster-Tree Networks

Ricardo Severino, Sana Ullah, Eduardo Tovar

*CISTER Research Centre

Polytechnic Institute of Porto (ISEP-IPP)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail: rar@isep.ipp.pt, sauah@isep.ipp.pt, emt@isep.ipp.pt

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

Abstract

Telecommun Syst (2016) 63:381–397
DOI 10.1007/s11235-015-0128-0

A cross-layer QoS management framework for ZigBee cluster-tree networks

Ricardo Severino[1] · **Sana Ullah**[1,2] · **Eduardo Tovar**[1]

Abstract Wireless sensor networks show great potential to successfully address the timeliness and energy-efficiency requirements of different cyber-physical system applications. Generally, these requirements span several layers of the stack and demand an on-line mechanism capable of efficiently tuning several parameters, in order to better support highly dynamic traffic characteristics. This work presents a cross-layer QoS management framework for ZigBee cluster-tree networks. The proposed framework carries out an on-line control of a set of parameters ranging from the MAC sub-layer to the network layer, improving the successful transmission probability and minimizing the memory requirements and queuing delays through an efficient bandwidth allocation at the network clusters. Through extensive simulations in a real datacenter monitoring application scenario, we show that the proposed framework improves the successful transmission probability by 10 %, and reduces the end-to-end delay by 94 %.

Keywords ZigBee · Cluster-tree · Quality of service · Cross layer control

1 Introduction

Wireless sensor networks (WSNs) have emerged as a promising technology for numerous cyber-physical system (CPS) applications [38] which require a close interaction with the physical world, enabling new applications in aerospace, transportation, manufacturing, and healthcare. These network infrastructures also have great potential to address various challenges posed by the large-scale energy consumption, cooling, and operational needs of large data centers, by providing a sensor layer to monitor parameters such as temperature, humidity, airflow and power consumption per server. Existing data centers consume around 50 % of the supplied energy in cooling related actions [19]. A WSN infrastructure can enable more precise and efficient control of the datacenter's equipment and may reduce the cooling cost.

WSN applications have different Quality of Service (QoS) requirements [32], particularly in what concerns timeliness. Structured logical topologies such as cluster-trees are generally used to address these QoS requirements [1,9,31]. They provide deterministic behavior instead of flat mesh-like topologies, where timeliness is not always guaranteed. The ZigBee [43] standard has proposed a cluster tree network topology to address different QoS requirements of heterogeneous applications and support synchronization and predictability through a hierarchical network structure. However, this topology is not directly adaptable due to implementation and flexibility problems. Few solutions have already been proposed to address these problems [20,21,36] however, to the best of our best knowledge, no attempt has been made to integrate them in one framework capable of supporting online and dynamic cross-layer QoS management mechanisms for cluster-tree networks [30,34]. This framework should be able to allocate network resources online

---

✉ Ricardo Severino
rar@isep.ipp.pt

Sana Ullah
sauah@isep.ipp.pt

Eduardo Tovar
emt@isep.ipp.pt

[1] CISTER Research Centre, ISEP/IPP, Rua Dr. Antonio Bernardino de Almeida, 431, 4200-072 Porto, Portugal

[2] Department of Computer and Software Technology, University of Swat, Odigram, Pakistan

and increase the network flexibility in terms of latency and bandwidth utilization, since these networks usually rely on a static cluster schedule. It should also adapt its medium access control (MAC) sub-layer to different traffic priority classes using a cross-layer approach. This would result in a framework capable of addressing not only the timeliness, but also other QoS aspects such as robustness, by providing the network infrastructure with self-adapting capabilities, and energy-efficiency, by providing traffic differentiation at the MAC sub-layer, improving the successful transmission probability to selected nodes, for instance, while relying upon the underlying cluster duty-cycling provided by the IEEE 802.15.4 beacon enabled mode.

This work presents a cross-layer QoS management framework that provides automatic and on-line control of two QoS mechanisms for ZigBee cluster-tree networks. At the MAC level, we improve the successful transmission probability of a tagged node, by carefully tuning the MAC parameters. The successful transmission probability is calculated by the fraction between the number of successfully acknowledged frames and the traffic generated at the application layer. At the network level, we reduce the queuing delays and memory requirements per node by carrying out an online efficient allocation of the available bandwidth for each cluster. This results in the elimination of bottlenecks in the network infrastructure, achieving a clear improvement in the end-to-end latency of the application. This work relies on two previously proposed and validated mechanisms, the Traffic Differentiation Mechanism (TRADIF) proposed in [20] and the Dynamic Cluster Scheduling (DCS) proposed in [36]. We extended the TRADIF mechanism to provide control of multiple nodes in the same cluster. An online performance evaluation mechanism is managed by a cross-layer Traffic Efficiency Control Module (TECM), which enables the necessary QoS mechanism where and when needed. The TECM also supports a mechanism to enable scalable and synchronized data acquisition in multiple clusters (SYNC) as proposed in [39]. However, this mechanism must be enabled on demand. To regulate access to the beacon payload from different modules, a Beacon Payload Management module (BPM) is also proposed in this paper.

We further validate and demonstrate the proposed mechanisms through simulations in a datacenter monitoring application scenario, which will be deployed in a new large-scale datacenter infrastructure in Portugal [34], using WSN as a sensing infrastructure to collect power and environmental data with high resolution and timing constraints. The simulation results show that the traffic rates are adjusted automatically by using TECM and by triggering DCS effectively. This results in a significant decrease of memory requirements, minimizing queuing overflow and end-to-end average latencies by 90 %. The results also show the possibility of improving the successful transmission probability of higher priority nodes by 8 % due to reduction in retransmissions, thus reducing the average energy consumption. Moreover, we conclude that the efficient pairing of TECM with DCS achieves a more efficient distribution of bandwidth than that of DCS only. This makes it possible for the network to accommodate higher traffic rates that would not be feasible otherwise.

The rest of the paper is organized as follows: Section 2 presents the related work. Section 3 provides an overview of the IEEE 802.15.4/ZigBee communication standards. Section 4 presents the QoS mechanisms supported by the proposed framework. Sections 6 and 7 discuss the proposed TECM and BPM mechanisms and their simulation results, respectively. The final section concludes our work.

2 Related work

In this section, we provide a brief overview of QoS in IEEE 802.15.4/ZigBee networks. This overview is classified into the following three groups.

2.1 QoS improvements to the IEEE 802.15.4 standard

Several research efforts focus on improving the performance of IEEE 802.15.4 slotted CSMA/CA protocol in terms of delay and reliability of time-critical events. Many approaches [12,14,16] adopt a priority toning strategy to privilege high priority frame transmissions. Although these solutions seem to improve the responsiveness of high priority frames in IEEE 802.15.4 slotted CSMA/CA, they require a non-negligible change to the IEEE 802.15.4 MAC protocol, thus affecting its standard compatibility. The toning mechanism imposes some changes to the hardware (using a tone signal transmitter) and also to the protocol itself, due to the frame tailoring strategy. Other approaches prioritize the nodes by tuning few IEEE 802.15.4 MAC parameters, such as the Backoff Exponent (BE) and Contention Window (CW) size. The authors of [16] presented a Markov chain model of IEEE 802.15.4 slotted CSMA/CA and analyze the impact of changing BE and CW parameters on delay and throughput. In [26], the authors studied a service differentiation scheme of two priority classes by changing the values of CWinit between one and two. The results of this work are interesting, however changing the CW to one may cause collisions with ACK frames. In [2], the authors focused on changing the backoff parameter to improve the response of network control system. They used Matlab/Simulink to simulate the control system and evaluate its response. In DBP [33], the authors introduced a (m,k)-firm deadline task model to assign priorities to messages. The backoff parameters were changed according to the proximity to lose m deadlines within a window of k service requests. This model was implemented on MICAz platforms.

ANGEL [18] presented an implementation and performance evaluation of a traffic differentiation mechanism based on a multi-queue service. Traffic differentiation was achieved by tuning some MAC parameters. However, the effects of changing each parameter, such as the macMinBE and the CWinit value, were not studied separately, and the performance evaluation only focused on changing the macMinBE and macMaxBE parameters. Moreover, the implementation was validated over TinyOS, which was found unreliable in [5] for a large amount of traffic due to lack of preemption and FIFO-based task management approach, thus making it difficult to precisely identify the impact of parameter variations at heavier traffic loads. In [20] the authors conducted a simulation study on the impact of different MAC parameters over several performance metrics, using a completely backward compatible approach with the IEEE 802.15.4 standard. This work proposed two queues to achieve traffic differentiation in IEEE 802.15.4 beacon-enabled networks: (1) a single FIFO queue supporting different traffic priorities by tuning the macMinBE, aMaxBE and CWinit MAC parameter, and (2) a multi-queue strategy in which different parameter values were assigned to different queues. Its improvement was verified by simulation using IEEE 802.15.4/ZigBee simulation model in OPNET [28] and was later implemented and validated over a real-time operating system in [35].

2.2 QoS improvements to the ZigBee protocol

The above work focused on traffic differentiation over the IEEE 802.15.4 networks, however, none of this work discussed on-the-fly alteration and implementation of different parameters. Considering that traffic and network performance may change during the network lifetime, it is important that the parameter tuning may be carried out periodically, adjusting to current network performance. This may avoid unnecessarily decrease in the performance of lower priority traffic. Concerning the network layer, general synchronized cluster-tree topologies tend to suffer from four technical challenges: (1) how to schedule the transmissions of different neighboring clusters avoiding interference, (2) how to predict the performance limits to correctly allocate resources, (3) how to change the resource allocation of the cluster-tree on-the-fly, and (4) the lack of available and functional implementations over standard WSN technologies, such as the IEEE 802.15.4/ZigBee protocols. This is particularly true for IEEE 802.15.4/ZigBee protocols, which support the cluster-tree network topology but do not provide a clear description of implementation problems including beacon collision problems. In [13], the IEEE 802.15.4b proposed some basic approaches to solve the aforementioned problems: the beacon-only period approach and the time division approach. Few other approaches targeted the scheduling of ZigBee cluster-tree networks. The work in [29] introduced the minimum delay beacon scheduling problem, however this work only addressed the latency problem by assuming the use of GTS slots for converge cast, and do not address the bandwidth problem. In [21], the authors proposed a Time Division Cluster Schedule (TDCS) algorithm for IEEE 802.15.4/ZigBee networks and implemented it in the Open-ZB stack [4]. This algorithm used a time-division approach and worked by assigning a different time offset to each cluster. The implementation of this work is available to the TinyOS and WSN communities [40], through the Open-ZB [27] framework. This work is of great importance since it solves the beacon scheduling issues for ZigBee cluster-tree networks. Other approaches, such as [3,24] followed a similar approach to [21] for mesh networks. The work in [15] addressed the problem of predicting resource needs by modeling the performance limits of ZigBee cluster-tree networks using GTS flows. In another work, the authors extended the latter by computing the optimal schedule for several GTS data flows [10]. Recently, [7] followed a similar approach to [29] proposing two heuristics to reduce the complexity of the otherwise NP-complete problem. Although the usage of GTS guarantees real-time performance within the IEEE802.15.4/ZigBee standards, the number of available GTS slots and their bandwidth is limited. The authors of [11] tried to improve the GTS bandwidth utilization by borrowing it from the neighboring nodes.

The above research efforts compute a static schedule based on periodic traffic assumptions, which remain active throughout the network lifetime. They follow a purely theoretical approach, lacking a clear description on how to implement such mechanisms on ZigBee cluster-tree networks. In [36], the authors presented a much simpler and low complexity DCS algorithm to satisfy bandwidth and delay requirements by rescheduling the clusters. It proposes two mechanisms: (1) carries out a rescheduling of the clusters ordering in the TDCS cycle aiming at minimizing end-to-end delays, and (2) rearranges the bandwidth allocation for the clusters involved in a stream, increases its bandwidth and decreases the overall data transmission time, and minimizes the queuing size and delay. Both techniques can be used together, or separately. The DCS algorithm was validated through simulations and implemented over TinyOS in real WSN platforms. Although each mechanism can be triggered on-the-fly, the user must specify a threshold based on a maximum predefined amount of traffic. This might be hard to correctly select without a simulation or experimental approach. Furthermore, to avoid increased complexity, the algorithm increases the bandwidth among all the clusters in the traffic stream which may not always be necessary and may create inefficiency. Nevertheless, we believe this is the most simple and practical approach and can improve the performance even further.

2.3 Online and cross-layer QoS proposals

As discussed above, most of proposed research efforts do not encompass an online mechanism to apply the necessary changes as the network performance decreases, and exactly where needed. Instead, they usually rely on the user to enable these services through application mode changes according to a set of assumptions usually obtained from simulation scenarios, and in most cases without performance feedback from the network. This suboptimal approach usually leads to an unnecessary decrease in the performance of low priority traffic at the MAC level, and also, to a waste of precious bandwidth resources at the network level, as traffic varies through time. Moreover, these approaches rely on the expertise of the user to control complex mechanisms, for instance, setting MAC parameters. Clearly this is a big impediment for a democratization of these networks, unnecessarily increasing their complexity, as most users do not hold the knowledge to fine tune these parameters. Since the traffic conditions may change, these settings must be updated throughout the entire network lifetime in many cases. Furthermore, as the QoS provisioning is not a one layer specific issue, the QoS management becomes a daunting task as the number of layers at the communication stack increases. The network layer performance for instance, is tightly coupled with the MAC sub-layer for efficient resource allocation. This resource allocation requires a cross-layer mechanism that must be able to address the QoS problems at MAC and network layers.

Cross-layer strategies have already been proposed in the literature [23,42] as an efficient way of solving many issues in WSNs. Most of these strategies focused on tuning parameters in different layers of the stack. Unfortunately, existing proposals which target QoS related issues, are either not compliant with ZigBee [37] or do not address the particular case of ZigBee cluster-tree networks such as in [25].

For instance, in [6], a cross-layer architecture was designed to address QoS in wireless multimedia sensor networks. The authors join the link layer and the network layer into a single communication module, to provide service differentiation to different classes in terms of soft delay, reliability and throughput, by relying in a cost function per hop. However, the proposal is not directed a clustered networks and it is not clear how it could be adapted to fit ZigBee-based networks. In [17] the authors proposed CCAR, another cross-layer strategy to support MAC level service differentiation and routing optimization. The proposal aims at providing QoS support for medical applications by monitoring a channel quality metric, looking at the buffer capacity and transmission delay at each node. Again, although the MAC layer service differentiation is achieved by tuning MAC parameters, the remaining of the cross-layer strategy cannot be applied to the case addressed in this paper.

In our proposal, by joining TRADIF and DCS, coupled with the proposed TECM online management capabilities, we achieve a cross-layer online QoS management service for ZigBee cluster-tree networks. We further show how the successful transmission probability of a higher priority best-effort traffic class can be dynamically improved through TECM/TRADIF and how the overall end-to-end delay can be reduced through TECM/DCS, by tuning MAC sub-layer parameters and the clusters duty-cycle respectively. Furthermore, this work results in a more efficient usage of energy and memory because of less retransmission attempts and reduced queue size.

3 On the IEEE 802.15.4/ZigBee protocols

IEEE 802.15.4 is a low-power communication standard designed for low data rate applications. It operates on three frequency bands: 868 MHz, 915 MHz, and 2.4 GHz bands. In addition, it supports two MAC operational modes; non-beacon-enabled and beacon-enabled modes. In the non-beacon-enabled mode, an unslotted CSMA/CA mechanism is used for resource allocation. In the beacon-enabled mode, the operation of the entire network is controlled by a central coordinator. The coordinator periodically transmits beacons for association and synchronization. The coordinator defines a superframe structure as illustrated in Fig. 1. The superframe is characterized by a Beacon Interval (BI), which specifies the time between two consecutive beacons, and a Superframe Duration (SD), which corresponds to the active period, as:

$$\begin{aligned} BI &= aBaseSuperframeDuration \times 2^{BO} \\ SD &= aBaseSuperframeDuration \times 2^{SO} \end{aligned} \tag{1}$$

where BO and SO represent Beacon Order and Superframe Order, respectively, and $0 \leq SO \leq BO \leq 14$. The Beacon Interval may optionally include an inactive period (for SO< BO), in which all nodes may enter into a sleep mode, thus saving energy. Sated otherwise, the coordinator communicates

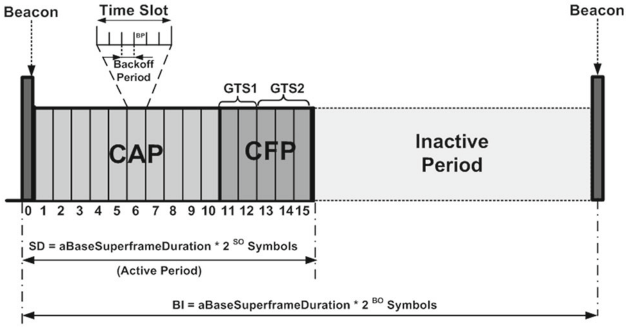

Fig. 1 IEEE 802.15.4 superframe structure

with the nodes during the active period and sleeps during the inactive period. The Contention Access Period (CAP) in the superframe employs a slotted CSMA/CA mechanism for resource allocation. The slotted CSMA/CA mechanism mainly depends on three variables: (1) the Backoff Exponent (BE), which enables computation of the backoff delay, (2) the Contention Window (CW), which represents the number of backoff periods during which the channel must be sensed idle before accessing it, and (3) the Number of Backoffs (NB), which represents the number of times the CSMA/CA algorithm goes into backoff while attempting to access the channel. Further details about the slotted CSMA/CA algorithm is given in [20]. Finally, the Contention Free Period (CFP) in the superframe consists of seven Guaranteed Time Slots (GTS) slots, which are used for time critical traffic. Figure 1 shows the superframe structure of the IEEE 802.15.4.

ZigBee defines network and application layer services on top of the IEEE 802.15.4 protocol. A ZigBee network is composed of three device types: (1) the ZigBee Coordinator (ZC), which identifies the network and provides synchronization services through transmission of the beacon frames containing identification of the network and other relevant information, (2) the ZigBee Router (ZR), which has the same functionalities as the ZC with the exception that it does not create its own networka ZR must be associated either to the ZC or to another ZR and must provide local synchronization to its cluster (child) nodes via beacon frame transmissions, and (3) the ZigBee End-Device (ZED), which neither has coordination nor routing functionalities and is associated either to the ZC or to the ZR.

ZigBee/IEEE 802.15.4 enables three network topologies star, mesh and cluster-tree. In the star topology a unique node operates as a ZC. The communication paradigm of the star topology is centralized, i.e. each device joining the network and willing to communicate with other devices must send its data to the ZC, which forwards it to the destination. The mesh topology also includes a ZC that identifies the entire network. However, the communication paradigm in this topology is decentralized, i.e. each node directly communicates with any other node within its radio range. The cluster-tree network topology is a special case of a mesh network where there is a single routing path between any pair of nodes and there is a distributed synchronization mechanism (IEEE 802.15.4 beacon-enabled mode). There is only one ZC which identifies the entire network and one ZR per cluster. Any of the FFD can act as a ZR providing synchronization services to other devices and ZRs. This topology is the most interesting to support time sensitive applications due to its scalability, predictability and the possibility to support GTS.

To manage the cluster's active periods the Time Division Cluster Scheduling (TDCS) approach is usually used in these networks, so that one can schedule the different cluster transmissions in way that no overlapping occurs. This is important to ensure the stability of the network, since it depends of correct beacon synchronization.

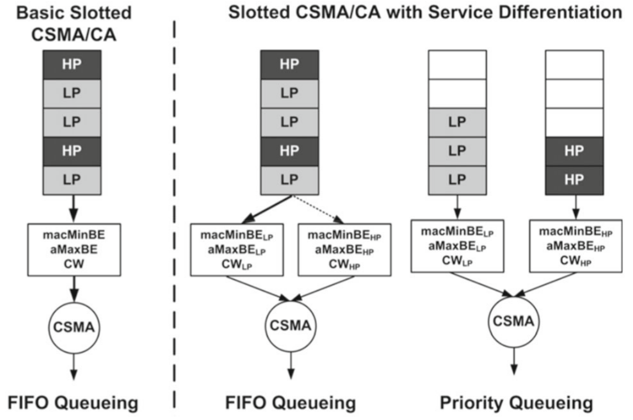

Fig. 2 TRADIF traffic-differentiation strategies

4 On the supported QoS mechanisms

4.1 TRADIF

TRADIF is a traffic differentiation mechanism, fully backward compatible with the IEEE 802.15.4 standard, which works by tuning a few MAC parameters. This mechanism is implemented in OPNET Open-ZB IEEE 802.15.4/ZigBee simulation model [27]. It is also validated over the ERIKA [8] real-time operating system in [35], using real WSN platforms.

The differentiated service strategies of TRADIF are presented in Fig. 2. Two approaches are proposed to achieve traffic differentiation: (1) a single FIFO queue supporting different traffic priorities by tuning the macMinBE, aMaxBE and CWinit MAC parameter, and (2) a multi-queue strategy in which different parameter values are assigned to different queues. Concerning the former, we denote [macMinBEHP, aMaxBEHP] and CWHP the backoff interval and the contention window initial values for high priority traffic related to high priority traffic, and [macMinBELP, aMaxBELP] and CWLP the initial values for low priority traffic. While, the slotted CSMA/CA remains unchanged, the adequate initial parameters that correspond to each service class must be applied. In addition to the specification of different CSMA/CA parameters, Priority Queuing can be applied to reduce queuing delays of high priority traffic. In this case, slotted CSMA/CA uses priority scheduling to select frames from queues, and then applies the adequate parameters corresponding to each service class. TRADIF differentiates traffic classes previously defined at network setup time, by tuning different MAC parameter values. Originally, these settings remain active regardless of the network performance. While simulation could be carried out to assess the correct settings, there are several implications: (1) it is assumed that the user has enough expertise to carry out this task; (2) because the settings remain active throughout the network lifetime, even if no traffic arrives from a higher priority class, the lower priority will still experience an unnecessary service downgrade.

We address the above problems by adding intelligence to the triggering of TRADIF. Performance indicators (successful transmission probability and the ratio of retransmissions) are used to trigger the proposed mechanism only if a service decrease for a high priority node is noticed. This is independent from the user, as it no longer chooses the parameter settings. Instead, TRADIF will change the respective MAC parameters in successive steps, until the performance indicator reaches an acceptable level. For instance, since the highest impact on the successful transmission probability metric is caused by changing CW, a decrease of service at a higher priority class will trigger TRADIF which will automatically increase the CW value of the lower priority classes to the next value. This process will be repeated until there is no noticeable degradation. To extend the capabilities of TRADIF at network level, a simple communication protocol is used to enable an interface between the TRADIF modules at different nodes. This enables a high priority node to ask for an increase of the CW of remaining nodes in the cluster, thus improving its success probability. Figure 3 presents a timing diagram of the communication protocol, when TECM triggers TRADIF to improve the success probability of a high priority node.

Upon the request, the TRADIF module must reduce the successful transmission probability of the lower priority nodes which compete in the same cluster. It then sends a TRADIF Request to its parent with three fields. The first field in Fig. 3 indicates the kind of nodes it is targeting (high (HP) or low priority (LP)), the second indicates the metric to be affected (successful transmission probability or throughput) and the third indicates the direction (increase or decrease). The request is received by the parent's TECM module and the contents of the request are forwarded as a TECM message to the Beacon Payload Manager module (BPM). This TECM message is incorporated in the beacon payload, which is received by all the cluster's nodes. Upon reception, the BPM forwards the payload to TECM, which will trigger TRADIF-SERVICE. TRADIF will then increment CW (of low priority nodes). If the higher priority node's performance does not improve, the process will be repeated.

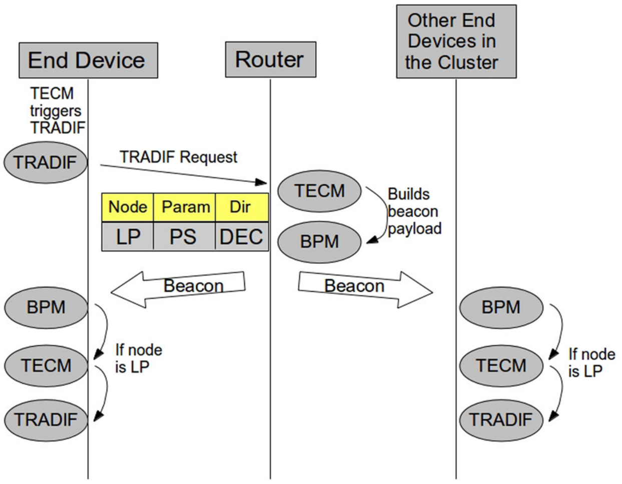

Fig. 3 TECM timing diagram

4.2 Dynamic cluster scheduling

With TDCS [21] it is possible to find the best schedule for the routers active periods in order to avoid interference, and to support most of the network bandwidth requirements. However, the schedule is done at network setup time, which assumes a static network that will remain unchanged. Thus, the choice of TDCS schedule has a strong impact in the end-to-end delays and on the available bandwidth for each cluster throughout the network lifetime.

The DCS reacts to different data flow changes on-the-fly, while simultaneously minimizing the network inaccessibility time using two techniques as proposed in [36]: (1) DCS Cluster Re-ordering (DCR), which re-orders the clusters' active periods to favor one set of streams and reduces end-to-end delays, and (2) DCS Bandwidth Re-allocation (DBR), which tunes the size of the clusters' duration and increases the bandwidth of the clusters serving a specific stream, while decreasing others bandwidth if needed. The DCR consists of a rescheduling of the clusters order in the TDCS cycle, aiming at minimizing end-to-end delays, while the second technique consists of rearranging the bandwidth allocation for the clusters involved in a stream, to increase its bandwidth and decrease the overall time for a data transmission, minimizing the queuing size and delay.

There is however a room for improvement in DCS. For instance, in DCS a node triggers the mechanism if the size of the stream of data which is to be transmitted is larger than a threshold. However, specifying this threshold is not easy and usually simulation must be done to choose this value. In DBR, on the other hand, the mechanism distributes a fixed amount of bandwidth throughout all the nodes in a stream in an equal fashion, without any added benefit, wasting precious bandwidth that may be required by other nodes. Often, only the nodes at the lower depths need extra bandwidth due to the higher concentration of traffic at nodes near the sink (assuming sink is at the root). In fact, even when the others at higher depths need bandwidth, it is not always in an equal amount.

In this paper, by joining DCS with TECM we can carry out a better and fairer redistribution of the bandwidth in an on-the-fly fashion. TECM relies on a performance indicator at each node which accounts for the input/output traffic ratio as well as the queue size. If the indicator drops below a threshold, TECM will trigger the DCS-DBR mechanism to increase the available bandwidth exactly where needed, improving the efficiency of the mechanism. To do this, only a change to the triggering mechanism is required. This process is discussed in detail in the next section.

5 On the traffic efficiency control mechanism

5.1 TECM architecture

The TECM consists of an online cross-layer module aiming at improving the QoS in ZigBee cluster-tree networks. It can improve the cluster scheduling, reduce end-to-end delays and queuing sizes using DCS, and improve the successful transmission probability for a higher priority traffic class using TRADIF. It relies on an online algorithm that periodically assesses the performance of the network and triggers the necessary QoS mechanisms. Figure 4 shows the proposed system architecture. The figure shows how the TECM module is wired into the IEEE 802.15.4/ZigBee stack and its most relevant services and Service Access Points (SAPs). The top three layers of the communication stack are implemented by the official TinyOS 2.x IEEE 802.15.4/ZigBee stack [41].

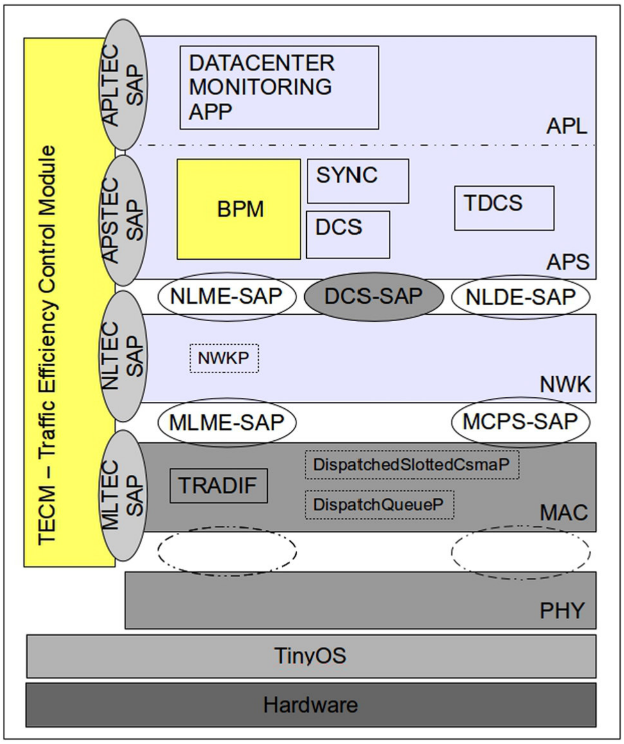

Fig. 4 TECM system architecture

The Application Layer (APL) is stacked at the top and connects to an example of a Datacenter Monitoring Application. The Application Support Sublayer (APS) provides the interface between the APL and the Network Layer (NWK) of the ZigBee communication stack through the NLDE and NLME ZigBee Service Access Points (SAPs). The DCS-SAP is also shown since it supports the DCS mechanism as described in [36]. The NWK also supports several network management service modules such as DCS, TDCS, BPM and SYNC. The TDCS module is only implemented at the routers and the Coordinator. It is responsible for the scheduling negotiation and is triggered at network setup, upon successful association by the application. The SYNC module, supports the application network wide synchronized data acquisition as described in [39], enabling all clusters to synchronize to any moment in time. By doing this, we can ensure that despite the clusters' different offsets between them, all can wake up or, for instance, sample a sensor simultaneously. This is of course supported by the underlying IEEE 802.15.4 beacon-enabled mode.

The DCS module is responsible for the dynamic rescheduling of each cluster. All the three kinds of nodes (ZC, ZR and End Devices) implement the service. The necessary interfaces to the MAC sub-layer are done through the regular IEEE 802.15.4 service access points. Within NWK in the figure, a dotted box shows a set of files of the communication stack implementation which were changed to support the necessary TECM packet counters for the performance assessment. At the MAC sub-layer, the TRADIF is implemented according to [35], however, an extra set of interfaces was introduced to enable control of the CSMA/CA parameters by TECM. Also a few interfaces were connected to different modules of the communication stack to retrieve information from the implemented packet counters. Again, these modules are shown in dotted boxes. These interfaces are part of the MLTEC-SAP.

The cross-layer module, TECM, manages the DCS and the TRADIF modules and uses different SAPs to interface each layer. In addition, it periodically pools through MLTEC-SAP and NLTEC-SAP a set of counters at the NWK and MAC layer to carry out the performance analysis. After processing a set of algorithms, it can trigger changes to the network scheduling and MAC parameters using the DCS or TRADIF modules respectively. The TECM SAPs implement a set of interfaces: (1) GET, which is called from TECM to receive the value of an attribute, (2) SET, which is used to set an attribute, and (3) RESET, which is always called at the beginning or if an issue is detected, similarly to the way the IEEE 802.15.4 and ZigBee SAPs are implemented.

The communication with the TECM module by the Application Layer is done using the TECM-SAP set of interfaces. These include: A TECM-control interface to control the starting and stopping of the TECM module at any time; the

TECM-setmode, to choose the TECM mode of operation (auto or fixed-rate), and TECM-setup, to program the different TECM parameters as follow.

TECM-setup (sampling window size, enabled modules, thresholds); where the sampling window size sets the period of the TECM performance analysis, enabled modules field informs the TECM module of which services are available (DCS and TRADIF by default), and the threshold field is used to specify the performance thresholds for each module. If none are specified, TECM will use a default setup.

TECM provides two modes of operation: auto-rate and fixed-rate. In the auto-rate mode, the TECM module will try to maximize the application traffic rate, by increasing it to a pre-defined steps established by the TECM-setmode interface, and by changing the network setup as necessary to accommodate the increase. This can be useful when there is no specific constraint concerning the traffic rate, but the objective is to optimize the use of the network resources, while keeping an acceptable performance. In the fixed-rate mode, the user defines at any point a traffic rate that should be maintained. The TECM module will trigger the DCS and TRADIF mechanisms as needed to guarantee an acceptable network performance. If at any point the TECM module is no longer capable of maintaining an acceptable network performance, due to reaching the maximum available bandwidth for that particular cluster, it will inform the application layer through the APLTEC-SAP, and the user can chose its action. This mode is also able to detect a reduction in the traffic rate, and reduce the previously assigned bandwidth if it is no longer required.

5.2 Beacon payload management module

The Beacon Payload Management Module (BPM) module in Fig. 5 consists of an interface layer for beacon processing between the NWK, the several APS service modules and the APL. Its objective is twofold: to manage the received beacon payload and deliver it to the corresponding module (DCS, SYNC, or TECM) or the APL, and to manage the concurrency from different modules which try to access and modify the beacon payload before sending it to NWK layer. To avoid long processing delays at beacon reception, only one module at a time is allowed to access the beacon payload.

Thus, each module delivers the content to the BPM module which places it within a FIFO queue. When ready, it builds the new beacon payload structure and signals the NWK layer.

The resulting beacon payload is illustrated in Fig. 5, where BPM message type field identifies the payload contents, the Module ID field identifies the information being integrated in the beacon payload, the Size field identifies the length of the payload, and finally, the Module Payload identifies the beacon contents.

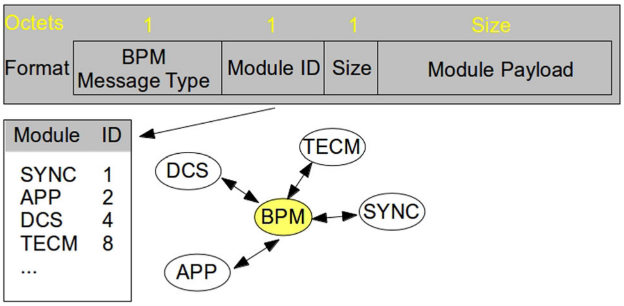

Fig. 5 BPM module description

5.3 Performance indicators

TECM periodically carries out a performance analysis based on a set of indicators at each node, every i-th interval. It relies on two performance indicators, which assesses the network QoS concerning bandwidth requirements and successful transmission probability.

The first performance indicator denoted by d_i represents the relationship between incoming and outgoing traffic, which gives a measurement of the bandwidth requirements of a node. This performance indicator can be computed as:

$$d_i = a\frac{c_{csma}}{c_{NWK}} + (1-a)\frac{c_{csma}}{c_{queue}} \tag{2}$$

where the first term represents the ratio between the number of packets transmitted by Slotted CSMA-CA algorithm (c_{csma}) and the number of packets delivered by the NWK (c_{NWK}) and eventually transmitted during a time window. A decrease in this term indicates that not all packets delivered by the NWK will be transmitted, resulting in an accumulation of packets in the queue. The second part of the performance indicator concerns the size of the queue size (c_{queue}) by considering the amount of packets that will be transmitted at each period. If this performance indicator decreases, it indicates that the queue is growing. It will grow as a penalty in d_i until the outbound/inbound traffic ratio inverts become higher than one, which indicates that the node is serving a higher amount of packets than the ones delivered by the NWK, showing that the queue is being emptied. A weighted mean is used to balance the two terms, where a defines its weight, represented by a number between 0 and 1.

If the d_i indicator reaches one, it means the node's queue is emptied at each active period, thus no more bandwidth is needed. To smoothen out the result, avoiding sharp transient oscillations that could trigger the mechanism inadvertently, the last result for D_i is always considered in an exponential moving average, where α is used to balance the average, resulting in:

$$D_i = \alpha d_i + (1-\alpha)D_{i-1}, i > 0 \tag{3}$$

The second performance indicator, t_i, concerns behavior of the MAC layer concerning successful transmissions, and it is represented by two ratios. The first term calculates the regular success probability metric as it is usually computed: a ratio between the number of successfully transmitted packets ($c_{success}$) and the number of packets which entered the Slotted CSMA-CA algorithm ($c_{success} + c_{fail}$). This is the most important indicator, since a decrease on in it immediately shows that the packets are being dropped. However, to know this is not enough since the IEEE 802.15.4 Slotted CSMA algorithm allows retransmissions. Hence, the second indicator takes this into consideration and shows a ratio between the number of successfully transmitted messages and the number of successfully transmitted packets. A decrease in this indicator shows that packets are not being successfully transmitted during the first attempt.

$$t_i = b\frac{c_{success}}{c_{success} + c_{fail}} + (1-b)\frac{c_{success}}{c_{success} + c_{ret}} \tag{4}$$

If there is no retransmission the second part of the indicator will tend towards one. If one retransmission occurs per packet it will tend towards 0.5, as the number of tries doubles to transmit one packet. Both ratios are averaged giving a higher weight to the first one. A weighted mean is used to balance the two terms, where b defines its weight, represented by a number between 0 and 1. To smoothen out the result and avoid sharp transient oscillations that could trigger the mechanism inadvertently, an exponential moving average is again used, in which β defines the average's weight, resulting in the indicator T_i:

$$T_i = \beta t_i + (1-\beta)T_{i-1}, i > 0 \tag{5}$$

In order to compute the performance indicators, only four counters must be implemented. Access to them by TECM must be granted using the MLTEC-SAP and NLTEC-SAP as described in Sect. 5.1. These will account for the NWK delivered packets (c_{NWK}), successfully transmitted packets ($c_{success}$), the number of packets which avail the Slotted CSMA-CA service (c_{csma}) and number of retransmissions (c_{ret}).

5.4 The TECM online algorithms

The TECM algorithm is presented in what follows. A sampling window is chosen at setup time, adjusted to at least two times the Beacon Interval. This ensures that each sample will always measure at least one transition from the ZR to the parent. However, this sampling window can be increased to save energy.

Algorithm 1 TECM Algorithm

```
Input:$r_i, d_i, t_i$
Output:$R_i, D_i, T_i, CW, DBRatios$

for every i do
    compute $R_i$ //traffic rate $r_i \leftarrow C_{NWK}$
    compute $D_i$
    if $D_i < Threshold_{DBR}$ then
        call DCS-Set(Inc. Bandwidth)
        $DBRratios[BW] \leftarrow R_i$
        $BW \leftarrow BW + 1$

        if (Auto-Rate Mode) then
            call APLTEC-Set (Dec. Data Rate)
        end if
    else
//we are fine with current available bandwidth

        if $BW > 0$ then //if bandwidth has been increased before we check for possible reduction due to a lower traffic rate
            $\Delta R_i \leftarrow \frac{R_i}{DBRratios[BW]}$

            if $\Delta R_i < Threshold_{\Delta R}$ then
                call DCS-Set(Dec. Bandwidth)
                $BW \leftarrow BW - 1$
            end if
        end if

        if (Auto-Rate Mode) then
            call APLTEC-Set (Inc. Data Rate)
        end if
    end if

    if Node is HP then
        compute $T_i$

        if $T_i < Threshold_{TRADIF}$ then
            call TRADIF-set (Inc. PS)
            $CW \leftarrow CW + 1$
        end if
    else
        if $CW > 0$ then //node is no longer HP but CW was changed
            call TRADIF-Set (RESET)
        end if
    end if
end for
```

The algorithm is focused in analyzing bandwidth requirements and packet delivery success probability. We can partition the algorithm into the following two phases:

(1) The first phase computes if any changes to the scheduling are needed namely if more bandwidth is needed for each node by looking into the D_i indicator. If so, TECM will trigger the DCS mechanism to get more bandwidth. A mechanism is also in place to decrease the service when the increased amount of bandwidth is not needed anymore. However, instead of trying to compute which is the optimum amount of bandwidth, we choose a simpler approach. The NWK incoming traffic rate is measured at each sampling window at the beginning of the algorithm. At each DCS-

SERVICE increase it will save the incoming traffic rate and will subsequently compare at each interval the current traffic rate with the saved value. If the current rate decreases beyond the saved value, it concludes it can safely reduce the bandwidth to the previous amount. In this case the DCS-SERVICE interface is used to ask for a decrease in the service. This has substantially modifies the original DCS mechanism which, after a schedule change, remains with that schedule for a per-programmed amount of time, and does not check if the bandwidth is still needed or not. This leads to two issues. First, it may result in unused bandwidth and starvation, second, when the timer expires, it might result in reducing the bandwidth while it is still needed, leading to unnecessary reschedules of the network, and increased inaccessibility time. Thus, there is a great advantage in providing on-line management of this mechanism, joining DCS with TECM. (2) The second phase of the algorithm computes if any changes are needed to improve the success probability. The algorithm starts by checking if the node is selected as high priority (HPN). Only HPNs can use the TRADIF service and request for improved success probability. They are selected using the TECM setup interface to the APL layer. If so, the performance metric T_i will be computed. If the overall weighed result goes beyond the previously set threshold, the TRADIF mechanism is triggered using the TRADIF-SERVICE interface.

6 Framework validation and results

To validate TECM, we relied on a real-world application scenario based on a data-center monitoring application. This application was previously addressed and implemented in [30]. However, few problems were identified in its design. The static network topology had troubles providing low latencies while still coping with the large amount of data generated at individual racks, and bottlenecks would appear at certain routers. There was also an impossibility to manage the priority of the data originating at different racks, for improved success probability. Using TECM we expect to solve these issues by improving on the flexibility of the network in terms of QoS, while maintaining a tight synchronization in terms of data acquisition.

6.1 Application description

A large portion of the power consumption in data centers is due to the control of physical parameters of the data center (such as temperature and humidity). This application features a data collection and distribution architecture that enables gathering physical parameters of a large data center at a very high temporal and spatial resolution of the sensor measurements. This is an important characteristic to enable more accurate heat-flow models of the data center and to optimize energy consumptions. Instrumenting data centers with very fine spatial and temporal granularity presents a twofold advantage. First, by providing the possibility of billing the consumed energy to their clients, and second by improving energy efficiency and having a better control of the micro-climate conditions in the rooms.

Figure 6a, b presents a view of the data-center testbed along with the WSN deployment. All the nodes are TelosB motes powered using USB hubs. For the moment we consider 8 racks and place a ZigBee router on top of each one, so that we have one cluster per rack. Each cluster consists of 6 sensing nodes (ZigBee End-Devices), capable of sensing temperature and humidity in one rack at the front and back. Other sensors will be added later on.

The application supports different modes of operation providing different data acquisition settings. It supports synchronous and asynchronous sampling using the SYNC module and enables to zoom in into specific parts of the datacenter, or receive finer data resolution on demand, focusing on a set of racks or one particular rack. This is done by choosing the application mode (AM), which can be changed during run time. The supported operational modes are listed below:

> AM1: Normal: asynchronous data acquisition of all racks with a relaxed report every 8 s.
> AM2: All Sync: synchronized data acquisition of all racks. The acquisition rate should be as high as the maximum allowed by the network.
> AM3: User Select Sync: user selects racks for a synchronized data acquisition every 2 s, while others maintain non-synchronized 8 s report.
> AM4: All Zoom In: non-synchronized data acquisition of all racks every 2 s.
> AM5: Rack Zoom In: user selects one rack for a non synchronized data acquisition with report every 0.8 s while the other racks report every 2 s.

Clearly, each application mode imposes different QoS requirements upon the network as data from specific racks may be of more interest than others in a particular moment, and traffic flows in the network may become quite demanding and unbalanced. This is especially visible in AM5, where data from one rack should have priority over the others, in AM4 where a higher report rate imposes bandwidth constraints to the network, and in AM2 where the objective is clearly to maximize the report rate. Thus, it is clear that this application presents interesting dynamics that could be improved by the use of the proposed TECM mechanism.

Concerning the MAC layer demands, the different application modes should provide differentiated traffic service to the application for each case. Namely, the successful transmission probability from the Zoomed In nodes should be

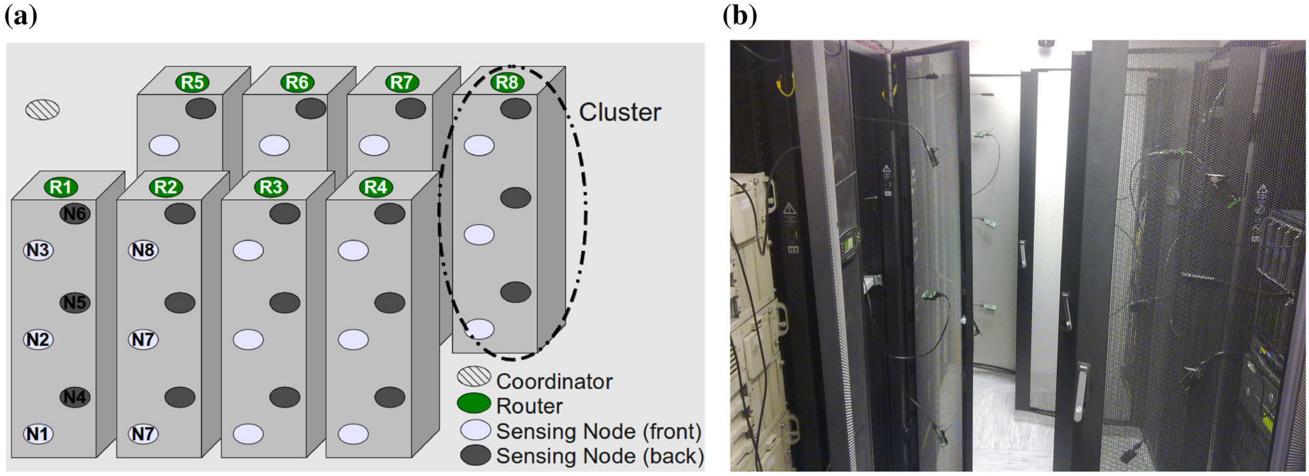

Fig. 6 Application scenario

higher. This can be achieved by using the TRADIF with TECM on-line mechanisms to trigger the TRADIF module when needed. At the NWK layer, the issue is mostly related to the bandwidth limitation. First, a relatively short delay is expected for the application, and second, long Beacon Intervals cannot be tolerated since it would increase the end-to-end latency. Therefore, a lower BO was chosen in order to minimize the latency (BO = 6). However, this has implications in the available bandwidth per cluster, since it cannot overlap and thus must be limited. Second, some application scenarios generate high traffic rates, which lead to a demand for more bandwidth at certain clusters. Failing to provide an increased bandwidth leads to higher queuing delays and memory demands and eventually packet drops and unpredictable behavior as the available memory limit is reached. Thus, a careful bandwidth allocation must be carried out at network setup, ensuring that the timing constraints of the application are taken into consideration, and then the bandwidth is increased as necessary at particular clusters with the DCS/DBR mechanism.

However, the DBR mechanism when triggered would increase bandwidth among all the clusters of a particular data stream by default for a fixed amount of time. Increasing the bandwidth in all the routers in a stream might not be always necessary, and since no network performance data is evaluated, this results in bandwidth depletion. Moreover, some applications may need a larger amount of bandwidth than what was given by the DBR rescheduling, which means that if it was miscalculated, the result will be suboptimal. This is especially important in AM2, where the objective is to maximize the data acquisition and report rate.

TECM can avoid these issues by using its on-line mechanisms to manage the DCS and in this particular the DBR mechanism. On the one hand it ensures the re-scheduling is carried out only for the nodes which sense a lack of bandwidth, and on the other hand the new schedule remains in place until the mechanism senses that a reduction of the bandwidth is in order.

6.2 Performance results

The TECM mechanism was evaluated through simulation using OPNET Modeler Wireless Suite v15, and the Open-ZB ZigBee simulation model. The mechanism and respective interfaces were also implemented over the official TinyOS Zigbee stack in using the TelosB [22] WSN platforms, which will be deployed in a large datacenter infrastructure in Portugal.

Several scenarios were setup to encompass the different application modes previously described. The network setup for each scenario is composed of 9 clusters. The cluster controlled by the Coordinator Node does not hold any sensing nodes. All the remaining clusters hold 6 end-devices each, resulting in a total of 48 sensing nodes, 8 Routers and one Coordinator for each simulation scenario. The network's BO was set to 6 to minimize latency and each Router's SO was set to 2. Routers were scheduled using TDCS in a downstream fashion to reduce downstream communication latency, minimizing the synchronization drift at each beacon period. For each scenario, 100 runs with duration of 10 minutes were carried out. Figure 7 presents one of the simulation scenarios in OPNET.

The application usually begins in AM1, with a very relaxed report from the sensing nodes every 8 s. This is the minimum report rate for the sensors to have an up-to-date birds eye view of the datacenter. However, as the report rate increases for the other modes, it is important to guarantee that data arrives with minimum delay otherwise, the user will always be looking into past data. This factor obviously depends on the amount of bandwidth available. If it

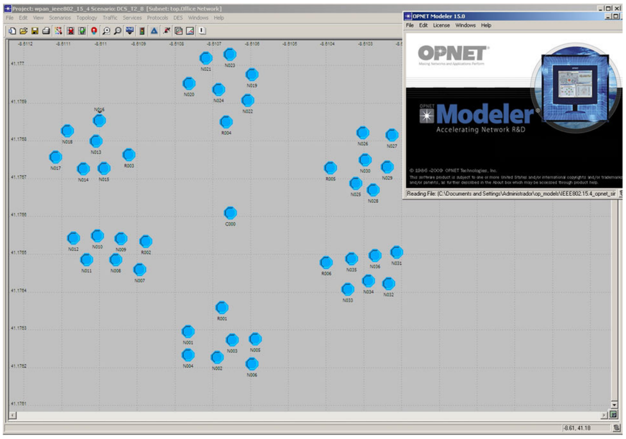

Fig. 7 OPNET simulation scenario

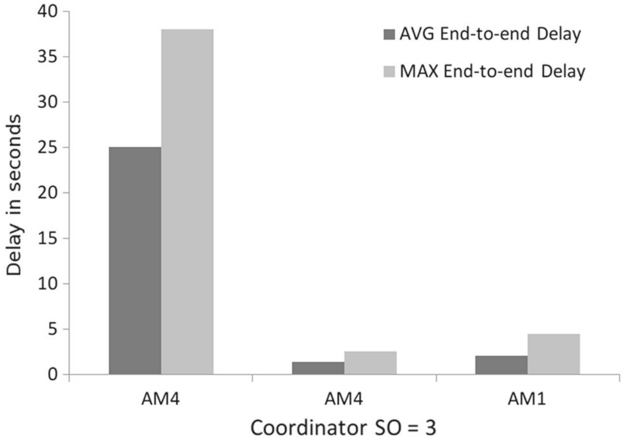

Fig. 8 Simulation average and maximum end-to-end delays for 2 scenarios (AM1 and AM4) when compared with an increase in the Coordinator SO

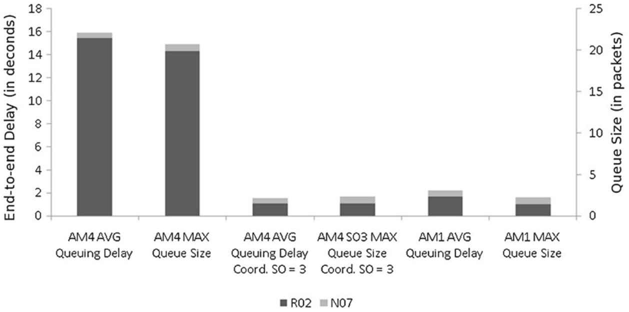

Fig. 9 Simulation results at node 7 and router 2 for the previous scenarios

is not enough, packets will be buffered waiting for service, increasing the end-to-end delay. To illustrate this problem at the network layer, we chose two asynchronous application modes which do not present changes at the MAC layer, (e.g. no high priority nodes) and are thus eligible for a fair comparison: Normal (AM1) and All Zoom In (AM4). Nevertheless, the same effect is to be expected whenever the traffic increases beyond the link capacity of a node.

Figure 8 shows the end-to-end delays for packets generated by the sensing nodes in the two application modes. The difference between AM1 and AM4 is quite significant as seen in the above figure, with an average end-to-end delay of 25 s in AM4 while for AM1 it is approximately 1.6 s. This results from an increase of the message rate in AM4 which forces all nodes to report every 2 s. Since the available bandwidth cannot cope with this increase in traffic, packets wait in the queue for service, thus increasing the delay. The problem is solved if the bandwidth for the Coordinator node is increased. This can be achieved by changing the SO allocated to it. Changing the SO from 2 to 3 will double the superframe size, thus doubling the available bandwidth. As shown in Fig. 8 the increase reduces the end-to-end delay to 1.4 s in AM4.

To better understand where the problem is located in the network, we look into the queue sizes and delay at R02 and Node 7 which belong to the same cluster C2. Figure 9 presents the results for a stream of data originating in a sensing node of cluster 2 with AM4. The other clusters present similar results since the same data rate is also applied to them. The lighter gray depicts the results for the queue of the sensing node, while the darker gray represents the results for the router queue. Average queuing delay and size is presented for the three scenarios shown before. As shown, the largest queuing delay is by far in the Router's queue. The queue size also grows considerably, reaching the memory limit.

Thus, we have identified the bottleneck. The available Coordinator bandwidth is not enough to cope with the increase traffic from C2, leading to an accumulation of packets in the queue of the respective router R2. Increasing the SO of the Coordinator node to SO = 3 reduces both queuing delay and size, approaching the results for AM1 scenario. The SO increase can be accomplished using the BDR option of the DCS mechanism. However, this mechanism will redistribute the bandwidth among all participants of the stream by default.

Indeed, this is the safest way to guarantee there will be no shortage of bandwidth, but it is rarely needed. With TECM it is possible to selectively increase the bandwidth where needed, in this case at the Coordinator. This can be achieved by carefully triggering the DBR mechanism in a slightly different way as presented before in Sect. 5. This saves bandwidth and makes its use more efficient. Figure 10 shows

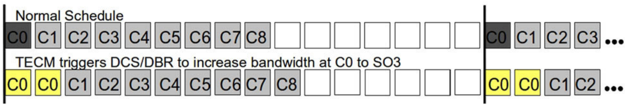

Fig. 10 Resulting schedule after TECM triggers de DCS/DBR mechanism

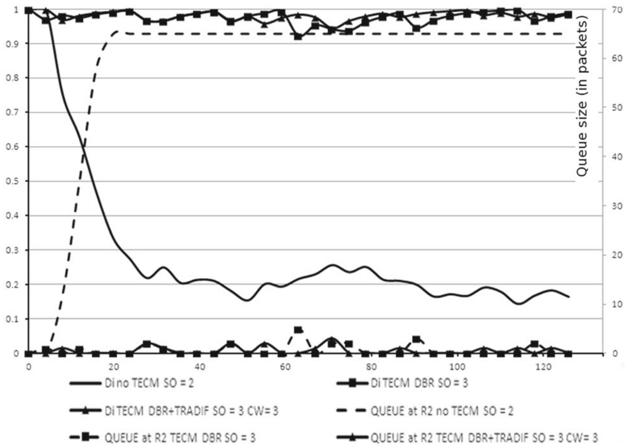

Fig. 11 Variation of D_i and queue size in AM5 when using TECM with DCS

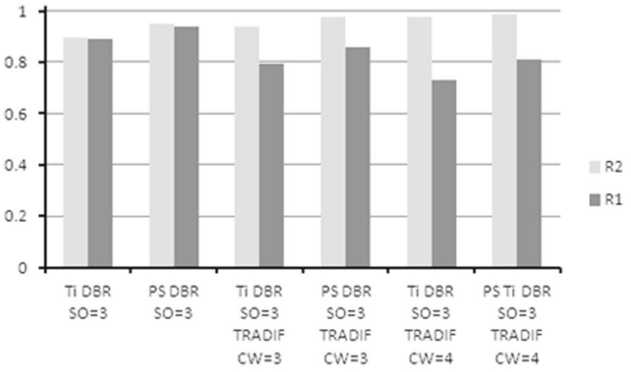

Fig. 12 Average probability of successful transmissions and average T_i for different contention windows at two routers

the resulting schedule, where C_x stands for cluster x active period.

TECM relies on two performance indicators to trigger the most adequate mechanism as presented in Sect. 5. D_i is mostly affected by the lack of available bandwidth and T_i by the Probability of Success. Figure 11 presents the variation of D_i at Router 2 with a sampling interval of 4 s, approximately 4 times the BI size. The application mode selected for this scenario is AM5 and the high priority rack selected is Rack 2 (C_2). As observed, using mode AM5 rapidly results in a reduction of D_i in Router 2 due to the lack of available bandwidth to accommodate the traffic generated by the sensing nodes. This is visible by the increasing rate at which the queue goes up to full capacity. Increasing the bandwidth at the Coordinator solves the above problem. By keeping the indicator above 95 % and the queue size near zero, the D_i indicator shows that all packets are successfully transmitted in each transaction as shown in Fig. 11. This is visible in the small queue size.

In another scenario, the contention window of the other lower priority nodes (considering C_2 is high priority) was increased using TRADIF-SERVICE, improving the probability of success for C_2. The objective is to understand the impact caused from an improved probability of success in the D_i indicator. As seen in Fig. 12, this change does not impact the indicator much, confirming that it is highly independent from the probability of success. On the other hand, changing the contention window impacts the T_i indicator as it is affected by the probability of success. Figure 12 shows the average probability of success and average T_i as contention window of lower priority nodes is increased from CW = 2 up to CW = 4. Note that for the case of CW = 3 and CW = 4 the probability of success is higher and quite similar between each other, however T_i shows a higher improvement for CW = 4. This is because T_i besides the probability of success also translates the number of retransmissions, and as expected, increasing CW on the lower priority nodes results in a reduced number of retransmissions for the high priority node and the opposite for the remaining. Notice that R1 is shown as example of a low priority router in Fig. 12, but the other low priority routers present the same behaviour, since they all share the same collision domain.

In this case the improvement in the success probability reaches 10 % for the case of CW = 4. R1, a lower priority node, on the other hand, gets decreased service so that R2 can increase its indicator. For the case of R1 with CW = 4 its successful transmission probability is reduced by 8 %.

6.2.1 *TECM fixed-rate mode*

Figure 13 shows the variation of the T_i and D_i indicators as TECM is applied to the network setup, both for R2 (high priority cluster) and R1 (regular cluster). The remaining low priority clusters present similar results to R1 as they share the same collision domain. Similarly, any other router can be elected as high priority, thus presenting similar results to R2. The application begins in AM5 mode, which is quite challenging for the network, due to its bandwidth requirements (sampling rate increases to 0.8 s at the high priority cluster while all the others must guarantee 2 s sampling rate), and the high priority cluster demands better service at the MAC layer during contention.

In the beginning we immediately observe a rapid decrease of the D_i indicator followed by a decrease of T_i. This is related to the lack of available bandwidth in the Coordinator node. T_i also decreases because the lack of bandwidth creates more collisions as nodes are competing for the medium, resulting in a decrease of the probability of success and increased queue size. Several reasons can justify this scenario. Bad network planning or the need for a reduced beacon interval, to keep a low latency in the communication, may result in an under dimensioned bandwidth distribution.

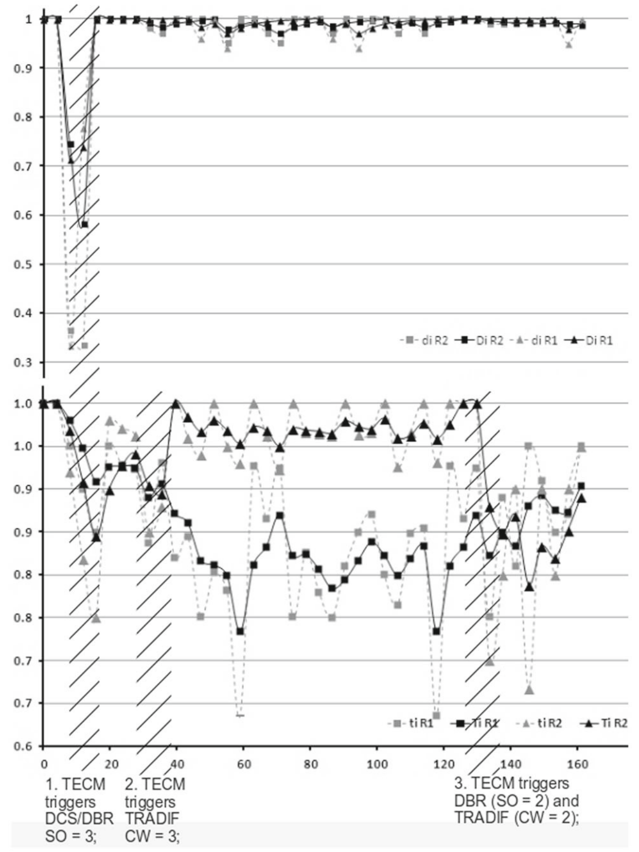

Fig. 13 Variation of the T_i and D_i indicators as TECM is applied to the network setup

As D_i decreases beyond the selected threshold of 90 %, the TECM triggers the DCS/DBR issuing a DCS Request. Both R2 and R1 trigger the mechanism as both feel the effects of reduced bandwidth, although the D_i indicator is worse for R2 due to a larger amount of packets accumulating in its queue. The same applies to the remaining Routers. After the positive DCS Response and the respective rescheduling occur, the D_i indicator immediately climbs.

In this particular application the buffers are emptied after this procedure. This is to remove the burden of the accumulated packets in memory. If this is not done, the cluster will require much of the bandwidth to transmit all the delayed packets. Instead, since delayed packets are not important in this application we give the system a clean start. However, this is optional. As observed in Fig. 13, the D_i indicator increases immediately and stabilizes above the 90 % threshold. On the other hand, the T_i indicator of R2 remains below the threshold. Since C2 is a high priority cluster, TECM triggers the TRADIF-SERVICE interface to correct this and TRADIF increases the contention window of all lower priority clusters, improving the probability of success for C2. Immediately we see a rise at T_i, as the probability of success increases for C2, and simultaneously a slight decrease of T_i at C1. This happens on all other clusters but is not shown in this figure for readability.

As observed T_i stabilizes around 85 % for C1. Similar values were found at the other clusters. After a while, the application mode is changed to AM1. The incoming traffic rate at C0 decreases and as it drops below the threshold as explained in Sect. 5, the DCS mechanism will trigger a DCS Request to decrease the service, as the bandwidth is not needed anymore. The schedule returns to the previous version. In Fig. 13, this is not noticeable in the D_i indicator, as it remains always high. This is because there is no lack of bandwidth. Concerning T_i at R2, a change is visible as C2 loses its high priority status. R2 issues a RESET using the TRADIF-SERVICE interface and all the clusters reset the TRADIF mechanism, leading to an immediate decrease of T_i in R2 and stabilization around 90 %. A slight increase is observed for C1 and all the remaining clusters as the contention window is reset to its default value (CW = 2), and now share the same priority in contention.

6.2.2 *TECM auto-rate mode*

TECM can also be used in Auto-Rate mode. This mode aims at maximizing the traffic rate of the selected nodes, using DCS or TRADIF when necessary. This is usually chosen when one does not care about enforcing a delivery rate but wishes to maximize the use of the available network resources within predefined performance limits. TECM will periodically tell the application to increase the traffic rate (the user must implement that in the APL) through the APLTEC-SAP, as long as the D_i performance indicator does not fall beyond a threshold. In that case, TECM will ask the APL to decrease the traffic rate and trigger a DCS Request in a similar fashion to the Fixed-Rate mode to increase available bandwidth. Upon re-scheduling, TECM continues the process until no further increase is possible, usually due to a negative DCS Response, at which point the last traffic rate is kept.

Figure 14, shows the variation of D_i in R2 and N7 (sensing node belonging to C2) for AM2, as the rate is increased using TECM Auto-Rate mode from 2 packets/second per sensing node up to 15. As the traffic rate climbs beyond 2.4 packets/second, the D_i indicator at R2 starts decreasing as its queue begins to grow. Note that D_i in N7 remains the same. As D_i goes beyond the threshold, DCS is triggered and similarly to the previous mode the indicator is reset and packets are purged. A decrease in the rate to the previous value is carried out as soon as the issue is detected, and D_i goes up while DCS re-scheduling is being carried out, increasing the coordinator node bandwidth by changing its SO to 3. Importantly, D_i at the sensing node is not affected as no more bandwidth is needed. The traffic rate resumes its increasing rate after the DCS re-schedule and D_i remains

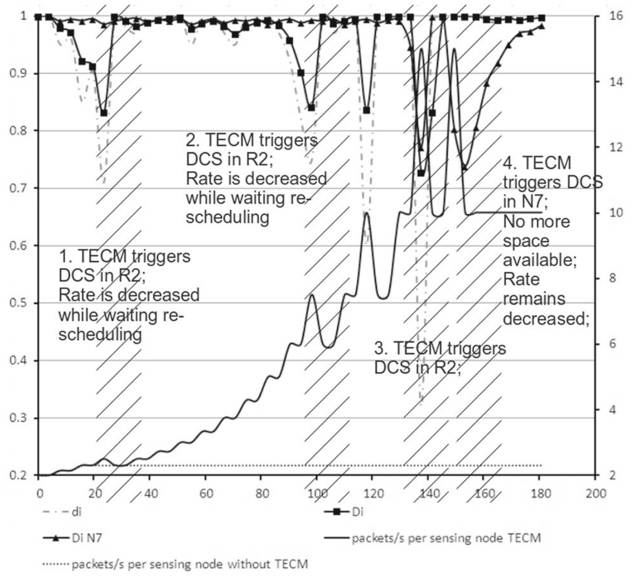

Fig. 14 Variation of D_i in R2 and N7 (sensing node belonging to C2) for AM2, as the rate is increased using TECM auto-rate mode

stable at the two nodes until a traffic rate of approximately 7 packets/second is reached. At this point, D_i at R2 decreases again beyond the threshold which triggers DCS. Traffic generation is decreased to the previous rate and DCS increases the Coordinator SO to 4. When a rate of 15 packets/sec is reached, the process is repeated. However, this time, the D_i of node N7 also indicates that more bandwidth is needed to transmit all the traffic to R2. DCS computes the new scheduling, increasing the coordinator's SO to 5 and all the routers SO to 3 (the next SO). At this point, all the BI space is used. The TECM Auto-Rate process resumes increasing the traffic rate again but D_i at node 7 is showing that the available bandwidth is still not enough to support the specified traffic rate. DCS is triggered, however, as no more space is available, a negative DCS response is sent. In response, the nodes at depth 2 must maintain a lower traffic rate. The TECM Auto-Rate process reaches a steady state and remains with those settings until the user resets TECM.

An interesting observation results from the fact that without TECM, solely relying on the DCS algorithm would result in a much lower traffic rate due to a non-optimized bandwidth redistribution. Figure 14 depicts the maximum traffic rate that could be achieved if only DCS was used, and Fig. 15 shows the resulting schedules.

As presented, solely using the original DCS algorithm would result in the last schedule, as bandwidth would be distributed among all the nodes in the streams, increasing each nodes' SO to 3, quickly depleting the available space in the BI, even if at that rate, an increase in the coordinator's bandwidth would suffice. By using TECM to trigger DCS we added intelligence to the process, only increasing the bandwidth where needed, allowing for much higher traffic rates as seen in Fig. 14.

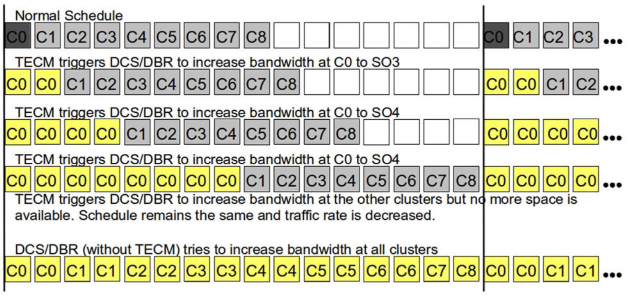
Normal Schedule
C0 C1 C2 C3 C4 C5 C6 C7 C8 C0 C1 C2 C3 ...
TECM triggers DCS/DBR to increase bandwidth at C0 to SO3
C0 C0 C1 C2 C3 C4 C5 C6 C7 C8 C0 C0 C1 C2 ...
TECM triggers DCS/DBR to increase bandwidth at C0 to SO4
C0 C0 C0 C0 C1 C2 C3 C4 C5 C6 C7 C8 C0 C0 C0 C0 ...
TECM triggers DCS/DBR to increase bandwidth at C0 to SO4
C0 C0 C0 C0 C0 C0 C0 C0 C1 C2 C3 C4 C5 C6 C7 C8 C0 C0 C0 C0 ...
TECM triggers DCS/DBR to increase bandwidth at the other clusters but no more space is available. Schedule remains the same and traffic rate is decreased.
DCS/DBR (without TECM) tries to increase bandwidth at all clusters
C0 C0 C1 C1 C2 C2 C3 C3 C4 C4 C5 C5 C6 C6 C7 C8 C0 C0 C1 C1 ...

Fig. 15 Resulting network schedules as TECM carries out network changes

7 Conclusions and future work

This work presents TECM, a cross-layer QoS management framework, providing an automatic and on-line control of two QoS mechanisms for ZigBee Cluster-tree network based applications. At the MAC, we improve on the successful transmission probability to achieve traffic differentiation using TRADIF. At the network level, through DCS we carry out an on-line efficient allocation of the available bandwidth, reducing the queuing delays and memory requirements per node, which results in the elimination of bottlenecks in the network infrastructure, and a clear improvement to the end-to-end latency. Interestingly, we achieve better results than if the mechanism were used separately. In our evaluation, we were able to achieve reductions in end-to-end delay in the order of 94 % and improvements in the successful transmission probability up to 10 % in a real datacenter monitoring application scenario. Latency for any specific stream can be further reduced if DCS/DCR is used in conjunction with the DBR technique.

We also proposed an extension to the TRADIF mechanism enabling network level communication, and BPM, a beacon payload management module. We validate and demonstrate our proposal through simulation in a datacenter monitoring application scenario, which is to be deployed in a new large-scale datacenter infrastructure, using the WSN as a sensing infrastructure to collect power and environmental data, with high resolution and timing constraints. The proposal was also implemented over the TinyOS operating system, and is awaiting deployment at the datacenter facilities, to enable a fair comparison between simulation and experimental results.

So far only two QoS properties were tackled in our proposal, (the ones that mostly hindered the prospective application) but we expect to include others in the near future, as the underlying TECM mechanism design can easily support this. In this line of work, we plan to enable hidden-node avoidance with this framework in the near future.

Acknowledgments This work was partially supported by National Funds through FCT/MEC (Portuguese Foundation for Science and Technology) and co-financed by ERDF (European Regional Development Fund) under the PT2020 Partnership, within project UID/CEC/04234/2013 (CISTER Research Centre); by FCT/MEC and the EU ARTEMIS JU within projects ARTEMIS/0001/2012-JU Grant Nr. 332987 (ARROWHEAD), ARTEMIS/0004/2013-JU Grant Nr. 621353 (DEWI, http://www.dewi-project.eu); also by FCT/MEC and the European Social Fund (ESF) through POPH (Portuguese Human Potential Operational Program), under PhD Grant SFRH/BD/71573/2010.

References

1. Abdelzaher, T., Prabh, S., & Kiran, R. (2004). On real-time capacity limits of multihop wireless sensor networks. In *Proceedings of 25th IEEE international on real-time systems symposium, 2004* (pp. 359–370).
2. Boughanmi, N., Song, Y. Q., & Rondeau, E. (2009). Online adaptation of the ieee 802.15.4 parameters for wireless networked control systems. In *8th IFAC international conference on fieldbuses and networks in industrial and embedded systems (FET 2009)*.
3. Burda, R., & Wietfeld, C. (2007). A distributed and autonomous beacon scheduling algorithm for ieee 802.15.4/zigbee networks. In *IEEE internatonal conference on mobile adhoc and sensor systems, 2007. MASS 2007* (pp 1–6).
4. Cunha, A., Koubaa, A., Severino, R., & Alves, M. (2007). Open-zb: An open-source implementation of the ieee 802.15.4/zigbee protocol stack on tinyos. In *IEEE internatonal conference on mobile adhoc and sensor systems, 2007. MASS 2007* (pp. 1–12).
5. Cunha, A., Severino, R., Pereira, N., Koubaa, A., & Alves, M. (2008). Zigbee over tinyos: Implementation and experimental challenges. *8th Portuguese conference on automatic control (CONTROLO2008)* (pp. 21–23). Portugal: Vila Real.
6. Demir, A., Demiray, H., & Baydere, S. (2014). Qosmos: Cross-layer QoS architecture for wireless multimedia sensor networks. *Wireless Networks*, *20*(4), 655–670. doi:10.1007/s11276-013-0628-3.
7. Di Francesco, M., Pinotti, C. M., & Das, S. K. (2012). Interference-free scheduling with bounded delay in cluster-tree wireless sensor networks. In *Proceedings of the 15th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems, ACM, New York, NY, USA, MSWiM '12* (pp. 99–106).
8. Evidence. (2014). Erika real-time operating system, online. http://erika.sssup.it/.
9. Gibson, J., Xie, G., & Xiao, Y. (2007). Performance limits of fair-access in sensor networks with linear and selected grid topologies. In *Global telecommunications conference, 2007. GLOBECOM '07. IEEE* (pp. 688–693).
10. Hanzalek, Z., & Jurcik, P. (2010). Energy efficient scheduling for cluster-tree wireless sensor networks with time-bounded data flows: Application to IEEE 802.15.4/ZigBee. *IEEE Transactions on Industrial Informatics*, *6*(3), 438–450.
11. Huang, Y. K., Pang, A. C., Hsiu, P. C., Zhuang, W., & Liu, P. (2012). Distributed throughput optimization for zigbee cluster-tree networks. *IEEE Transactions on Parallel and Distributed Systems*, *23*(3), 513–520.
12. Hyun, T., Lee, D., Ahn, J. H., & Choi, S. (2005). Priority toning strategy for fast emergency notification in IEEE 802.15.4 LR-WPAN. In *Proceedings of the 15th joint conference on communications & information (JCCI)*.
13. IEEE-TG154. (2006). IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements. Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs).
14. Jeon, J., Lee, J. W., Kim, H. S., & Kwon, W. H. (2007). Pecap: Priority-based delay alleviation algorithm for ieee 802.15.4 beacon-enabled networks. *Wirel Pers Commun*, *43*(4), 1625–1631. doi:10.1007/s11277-007-9331-y.
15. Jurcik, P., Koubâa, A., Severino, R., Alves, M., & Tovar, E. (2010). Dimensioning and worst-case analysis of cluster-tree sensor networks. *ACM Transactions on Sensor Networks*, *7*(2), 14:1–14:47.
16. Kim, E. J., Kim, M., Youm, S. K., Choi, S., & Kang, C. H. (2007). Priority-based service differentiation scheme for IEEE 802.15.4 sensor networks. *International Journal of Electronics and Communications*, *61*(2), 69–81. doi:10.1016/j.aeue.2006.02.004.
17. Kim, Y. D., Cho, K. R., Cho, H. S., & Kim, D. (2014). A cross-layer channel access and routing protocol for medical-grade QoS support in wireless sensor networks. *Wireless Personal Communications*, *77*(1), 309–328. doi:10.1007/s11277-013-1507-z.
18. Kipnis, D., Willig, A., Hauer, J. H., & Karowski, N. (2008). The angel IEEE 802.15.4 enhancement layer: Coupling priority queueing and service differentiation. In *Proceedings of 14th European wireless conference, Prague* (pp. 1–7).
19. Koomey, J. (2011). Growth in data center electricity use 2005 to 2010. Technical Report, Analytics Press, Oakland, CA. www.analyticspress.com/datacenters.html.
20. Koubaa, A., Alves, M., Nefzi, B., & Song, Y. Q. (2006). Improving the IEEE 802.15.4 slotted CSMA/CA MAC for time-critical events in wireless sensor networks. In *Proceedings of the workshop of real-time networks (RTN 2006), satellite workshop to (ECRTS 2006)*.
21. Kouba, A., Cunha, A., Alves, M., & Tovar, E. (2008). TDBS: A time division beacon scheduling mechanism for zigbee cluster-tree wireless sensor networks. *Real-Time Systems Journal*, *40*, 321–354.
22. MEMSIC. (2014). Telosb datasheet. www.memsic.com/userfiles/files/Datasheets/WSN/telosb_datasheet.pdf.
23. Mendes, L. D. P., & Rodrigues, J. P. C. (2011). Review: A survey on cross-layer solutions for wireless sensor networks. *Journal of Network and Computer Applications*, *34*(2), 523–534. doi:10.1016/j.jnca.2010.11.009.
24. Muthukumaran, P., de Paz, R., Spinar, R., & Pesch, D. (2010). Meshmac: Enabling mesh networking over IEEE 802.15. 4 through distributed beacon scheduling.
25. Nandi, S., & Yadav, A. (2011). Cross layer adaptation for QoS in WSN. *International Journal of Computer Networks & Communications*, *3*(5), 287.
26. Ndih, E. D. N., Khaled, N., & De Micheli, G. (2009). An analytical model for the contention access period of the slotted IEEE 802.15.4 with service differentiation. In *ICC 2009, international conference on communication, Dresden*.
27. Open-ZB. (2014). Online. http://www.open-zb.net.
28. OPNET Technologies Incorporation. (2014). Modeler wireless suite, online. http://www.opnet.com.
29. Pan, M. S., & Tseng, Y. C. (2008). Quick convergecast in zigbee beacon-enabled tree-based wireless sensor networks. *Computer Communications*, *31*(5), 999–1011.
30. Pereira, N., Tennina, S., Loureiro, J., Severino, R., Saraiva, B., Santos, M., Pacheco, F., & Tovar, E. (2014). A microscope for the data center. In *International journal of sensor networks (IJSNet) inderscience*.
31. Prabh, K. S., & Abdelzaher, T. F. (2007). On scheduling and real-time capacity of hexagonal wireless sensor networks. In *Proceedings of the 19th Euromicro conference on real-time systems,*

IEEE Computer Society, Washington, DC, USA, ECRTS '07 (pp. 136–145).
32. Raman, B., & Chebrolu, K. (2008). Censor networks: A critique of "sensor networks" from a systems perspective. *SIGCOMM Computer Communication Review*, *38*(3), 75–78.
33. Semprebom, T., Montez, C., Moraes, R., Vasques, F., & Custodio, R. (2009). Distributed DBP: A (m, k)-firm based distributed approach for QoS provision in IEEE 802.15.4 networks. In *14th IEEE international conference on emerging technologies and factory automation (ETFA 09)*.
34. SENODS. (2014). Sustainable energy-optimized datacenters. http://www.cister.isep.ipp.pt/projects/senods/.
35. Severino, R., Batsa, M., Alves, M., & Koubaa, A. (2010). A traffic differentiation add-on to the IEEE 802.15.4 protocol: Implementation and experimental validation over a real-time operating system. In *2010 13th Euromicro conference on digital system design: Architectures, methods and tools (DSD)* (pp. 501–508). doi:10.1109/DSD.2010.95.
36. Severino, R., Pereira, N., & Tovar, E. (2014). Dynamic cluster scheduling for cluster-tree WSNS. *SpringerPlus Communication Networks*, *3*(493).
37. Shah, G., Liang, W., & Akan, O. (2012). Cross-layer framework for QoS support in wireless multimedia sensor networks. *IEEE Transactions on Multimedia*, *14*(5), 1442–1455. doi:10.1109/TMM.2012.2196510.
38. Stankovic, J. A., Lee, I., Mok, A., & Rajkumar, R. (2005). Opportunities and obligations for physical computing systems. *Computer*, *38*(11), 23–31.
39. Tennina, S., Koubaa, A., Daidone, R., Alves, M., Jurcik, P., Severino, R., Tiloca, M., Pereira N., Hauer, J.-H., Dini, G., Bouroche, M., & Tovar, E. (2013). *IEEE 802.15.4 and ZigBee as enabling technologies for low-power wireless systems with quality-of-service constraints, lecture notes in electrical engineering*. Berlin: Springer.
40. TinyOS. (2014a). Online. http://www.tinyos.net/.
41. TinyOS. (2014b). Zigbee working group. http://tinyos.stanford.edu/tinyos-wiki/index.php/TinyOS_ZigBee_Working_Group.
42. Yigitel, M. A., Incel, O. D., & Ersoy, C. (2011). Qos-aware mac protocols for wireless sensor networks: A survey. *Computer Networks*, *55*(8), 1982–2004. doi:10.1016/j.comnet.2011.02.007.
43. ZigBee-Alliance. (2005). Zigbee specification. Technical Report.

Ricardo Severino received his Ph.D. degree in Electrical and Computer Engineering from the Faculty of Engineering of the University of Porto, Portugal, in 2015. Since 2006, he has been working in the area of Wireless Sensor Networks, namely on improving quality-of-service (QoS) in WSNs by using standard and commercial-off-the-shelf (COTS) technology, at the CISTER Research Centre in Portugal. In this line, he has been actively participating in several international projects and has been involved in open-source projects such as the Open-ZB (http://www.open-zb.net) research framework. He is also a founding member and contributor of the 15.4 and ZigBee TinyOS Working Groups. He has several publications in reputed conferences (e.g. MASS, RTCSA, ECRTS) and journals (e.g. IEEE TII) and has served as a reviewer for several conferences.

Sana Ullah received his Ph.D. degree in Information and Communication Engineering from Inha University in 2011. He worked as an assistant professor in the College of Computer and Information Science, King Saud University, Riyadh from Dec. 2011 to Mar. 2014. He currently serves as an editor for Springer Journal of Medical Systems, KSII Transaction of Internet and Information Systems (TIIS), Wiley Security and Communication Network (SCN), Journal of Internet Technology and International Journal of Autonomous and Adaptive Communications Systems (IJAACS). He served as a guest editor for many top journals including IEEE Systems Journal, Elsevier Journal of Information Science (INS), Springer Journal of Medical System (JOMS), and Springer Journal of Telecommunication Systems (TS). He also served as a co-chair/TPC member for a number of top-ranked international conferences including BodyNets, IEEE PIMRC, IEEE ICC, IEEE Healthcom, IEEE Globecom, and IEEE WCNC. He is currently working as a Research Scientist at CISTER Research Unit at ISEP/IPP.

Eduardo Tovar heads the CISTER Research Centre, a top ranked (Excellent) research unit of the FCT Portuguese network of research units. Since 1991 he authored or co-authored more than 180 scientific and technical papers in the area of real-time computing systems, wireless sensor networks, distributed embedded systems and industrial computer engineering. Eduardo Tovar has been Program Chair for ECRTS 2005, OPODIS 2007, RTCSA 2010, RTAS 2013 and ARCS 2014.