Journal of Network and Computer Applications 82 (2017) 21–34

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

A power-aware MAC layer protocol for real-time communication in wireless embedded systems

Gianluca Franchino*, Giorgio Buttazzo

Scuola Superiore S. Anna, Pisa, Italy

ARTICLE INFO

Keywords:
Wireless embedded networks
Media access control
Multi-channel
Energy management
Real-time streams

ABSTRACT

The number of battery-operated devices with wireless communication capabilities has increased enormously in the last years and is expected to increase even more in the future. A fundamental need in these systems is to guarantee a minimum system lifetime and timing constraints through a careful management of energy, communication, and computational resources. This paper describes WBuST, a MAC layer protocol designed to bound the maximum delay of real-time messages and guarantee the system lifetime by properly allocating a share of the available bandwidth to each node of the network. The protocol allows multi-hop wireless communication under different network topologies. The proposed approach is assessed through theoretical analysis and experimental results.

1. Introduction

In the recent years, the interest on networked wireless systems has experienced an exponential growth, mainly due to the wide range of applications, including defense systems, health monitoring, domotics, intelligent buildings and industrial control systems.

The delay introduced by the network has a significant impact on the system performance, which can be specified according to different Quality of Service (QoS) levels. For example, time-critical data related to alarms must be delivered within stringent deadlines, and control loops data have to be transmitted periodically with a bounded delay variation (jitter).

When considering the design of a communication stack for real-time systems, a deterministic Medium Access Control (MAC) layer is crucial to guarantee a bounded transmission delay for any packet sent throughout the network. The techniques adopted to handle the channel access can be roughly divided in three categories: contention based, scheduling based, and hybrid approaches. The former makes use of CSMA/CA or ALOHA (Rappaport, 1996) methods, the second one implements scheduling algorithms to rule the channel access and the latter is a combination of both, see for instance IEEE 802.15.4 Std-2011 (2011). Each approach has its own advantages and drawbacks. CSMA/CA is simple, robust, highly scalable and does not need clock synchronization between nodes. The downside is that it suffers access collisions where two or more nodes can access the channel at the same time, causing a delay in the message transmission. Moreover, since carrier sensing does not work for nodes more than one hop away, a handshake mechanism (The IEEE, 1999) is necessary to mitigate the hidden/exposed terminal problem (Tobagi and Kleinrock, 1975). As a consequence of both collisions and handshake, the network throughput can be greatly reduced. On the other hand, scheduling based methods do not suffer hidden/exposed node problems, are collision-free and highly predictable in terms of transmission delay. The main shortcoming is, in many cases, the need of some form of clock synchronization between nodes that increases the protocol overhead; the network scalability is more difficult to achieve and much more infrastructure support is needed with respect to CSMA/CA.

In battery-operated systems, the energy management represents another key issue to be addressed at design time. As highlighted by Ye et al. (2004), four main sources of energy waste can be identified at the MAC level: *Overhearing*, this is the energy wasted by a node when receiving packets directed to other nodes; *Collision*, if a packet is corrupted, it has to be resent, hence both the sender and the receiver have to consume additional energy to exchange the packet; *Control Packet Overhead*, this is the energy consumed by a node to send and receive control packets; and *Idle Listening*, this is the energy dissipated by a node in receiving mode while waiting for incoming messages.

Note that CSMA/CA protocols are particularly prone to collisions and idle listening, whereas scheduling based protocols are mainly affected by control packet overhead.

Radio devices available in the market have different operating modes, each characterized by a different level of power consumption. The most common are: sleep, receiving, and transmitting. In this work, the possibility offered by the sleep mode is exploited to reduce energy

* Corresponding author.
E-mail addresses: g.franchino@sssup.it (G. Franchino), g.buttazzo@sssup.it (G. Buttazzo).

http://dx.doi.org/10.1016/j.jnca.2017.01.006
Received 24 April 2015; Received in revised form 26 April 2016; Accepted 8 January 2017
Available online 10 January 2017
1084-8045/

consumption.

This paper describes the Wireless Budget Sharing Token (WBuST) protocol, which is a MAC layer protocol designed for real-time wireless sensor/actuator networks (WSNs) of embedded devices. WBuST can handle both real-time and best effort traffic in multi-hop networks, while saving energy to guarantee a desired lifetime. The channel access is handled by a mixed approach that adopts a bandwidth-reservation mechanism to guarantee the desired performance, and a contention-based mechanism for the transmission of control and management messages.

Network devices are grouped into clusters of adjacent nodes, and a different radio channel is assigned to each cluster. In this way, the transmissions within adjacent clusters can take place at the same time without interfering with each other. The clusters can be connected to form various network topologies, where each cluster is managed by a coordinator, which is the node with the best link quality to neighbor nodes.

1.1. Contributions and summary

The most relevant contribution of this work is the analysis of the protocol performance, which provides a powerful method for guaranteeing a desired QoS level and lifetime for a given amount of network traffic. This is particularly useful for implementing admission control mechanisms to handle overload conditions. Concerning power management issues, besides of minimizing the energy consumption, as done in most of the related works reported in Section 2, this work also provides a method for selecting the protocol parameters that guarantee a given network lifetime. The properties of WBuST are also validated through experimental results.

This paper extends and completes a preliminary work by Franchino and Buttazzo (2012) in several directions: first, it extends the state of the art to present further related works; it proposes a methodology based on Network Calculus that can be applied to determine the maximum end-to-end inter-cluster communication delays of real-time streams in cluster-tree networks; then, it derives bandwidth and buffering requirements for network routers to guarantee the deadlines of real-time streams; and finally, it includes a new set of experimental results carried out to assess the performance of the WBuST protocol in a multi-hop scenario and show the effectiveness of the proposed energy saving mechanism.

The rest of the paper is organized as follows. Section 2 analyzes the related works, Section 3 describes the proposed protocol in detail, Section 4 introduces the traffic model, the bandwidth allocation schemes and the analysis of the protocol performance for intra-cluster communications. The method used by the protocol to save energy and predicting the network lifetime is shown in Section 5. Section 6 describes how to compute the maximum end-to-end communication delay of real-time streams, bandwidth and buffering requirements of router nodes through a technique based on Network Calculus. Section 7 reports and discusses the experimental results and, finally, Section 8 states the conclusions.

2. Related work

Real-time communication and energy saving issues over wireless networks have received great consideration in the literature during the last years. However, not many authors addressed both problems simultaneously.

Caccamo et al. (2002) proposed a cellular network architecture with a MAC protocol based on the Earliest Deadline First (EDF) algorithm (Liu and Layland, 1973). Implicit prioritization is achieved by exploiting the periodic nature of the traffic in sensor networks. The authors analyzed the capacity of the network using an implicit EDF scheme, where each node locally generates the same EDF schedule and accesses the channel without collisions. The implementation of this scheme requires clock synchronization among nodes contending for a channel. Moreover, a slotted reservation may lead to a waste of bandwidth if nodes either fail or, due to the variable packet size, do not use their reserved slots entirely. To address these problems, Crenshaw et al. (2005) presented the Robust Implicit-EDF (RI-EDF) protocol, which does not require clock synchronization, providing bandwidth reclamation and robustness in the presence of certain classes of node failures. A limitation of the RI-EDF protocol is that it requires full-duplex radio devices to achieve multi-hop communications.

Sobral and Becker (2008) proposed a Hybrid Contention/TDMA-based MAC protocol for ad hoc wireless networks organized into clusters. The proposed protocol can guarantee timely bounded communications both inside and outside the clusters, operating without a central coordinator. Shashi Prabh (2007) considered hexagonal meshes networks and proposed a transmission scheduling algorithm that guarantees a real-time communication. Furthermore, the authors provided an implicit clock synchronization method for supporting message scheduling and derived the real-time capacity (Abdelzaher et al., 2004) of the scheduling algorithm. Bui et al. (2007) introduced a prioritized MAC layer protocol that provides soft real-time communication in multi-hop wireless networks. The proposed protocol guarantees a collision-free channel contention and per packet prioritization by exploiting the Black Burst (Sobrinho and Krishnakumar, 1999) scheme and multiple channels.

The RT-Link protocol, proposed by Rowe et al. (2006), is a time-synchronized link layer protocol that guarantees a predictable lifetime and a bounded end-to-end delay across multiple hops. The authors provided an analytical estimation of the maximum energy consumption, such that it is possible to derive the minimum network lifetime. The maximum end-to-end communication latency is not analytically computed, but is evaluated by simulation experiments.

Koubaa et al. (2007) analyzed the power efficiency and the timeliness of the IEEE 802.15.4 Std-2011 (2011) when the GTSs mechanism is used. They also proposed a method to select the protocol parameters to trade power efficiency and delay bound guarantees. Toscano and Lo Bello presented an algorithm for superframe scheduling in industrial sensor networks based on the IEEE 802.15.4 standard. This algorithm is able to avoid beacon collisions by scheduling cluster superframes over multiple radio channels.

Saifullah et al. (2010) analyzed the problem of real-time transmission scheduling WirelessHART (2013) networks. In particular, they proved that the scheduling problem is NP-hard and provided an optimal branch and bound algorithm that finds a solution whenever a feasible one exists. Later, the same authors extended their work with an analysis of the end-to-end delay for fixed-priority scheduling (Saifullah et al., 2011).

More details on MAC layer protocols designed for Wireless Sensor Networks (WSNs) can be found in a set of survey papers (Demirkol et al., 2006; Huang et al., 2013; Rashid and Rehmani).

A technology that can be used to improve the network performance in ad-hoc networks is the beamforming antennas technique (Vilzmann and Bettstetter, 2005; Bazan and Jaseemuddin, 2012). This last makes use of antenna arrays and Digital Signal Processing (DSP) algorithms to create the so-called “smart antennas”. This technique provides several advantages, such as a higher throughput and a reduced energy consumption.

A further technology that it is worth to be considered is the Cognitive Radio approach, which can be used to better exploit the communication spectrum with the goal of achieving an higher throughput and extending the network coverage (Rehmani, 2010; Rashid et al.).

This paper takes into account several issues that are not fully addressed in the works cited above. In particular, it proposes a channel access protocol that can guarantee both real-time and best-effort traffic in worst-case scenarios; it presents a bandwidth reclaiming mechanism that improves the protocol performance in the average case; it

Table 1
Notation used throughout the paper.

Symbol	Description
C_i	The ith cluster or its coordinator node
R_i	The router node of cluster C_i
CW	Contention Window
T_b	Beacon period
B_c	Contention slot
B_i	Slot assigned to the i-th node
B_S	Sleep slot
S_i	Sporadic message stream
M_i	Maximum message length of stream S_i
T_i	Minimum inter-arrival time (or period) of stream S_i
D_i	Relative deadline of stream S_i
U_i	Bandwidth of stream S_i
U	Total channel bandwidth
U^*	Worst case achievable utilization
T_{BT}	Target beacon time
τ	Protocol overhead
α	Bandwidth lost due to τ
WC_i	Worst case transmission time for a message of S_i
$E_i(t)$	Average energy wasted by node i after t time units
$\phi_i^{data}(t)$	Data flow arrival curve generated by node i
b_i	Maximum data burst size of node i
r_i^{data}	Average data rate of node i
$\gamma_i(t)$	Service curve guaranteed to node i
ρ_i	Average service rate (bandwidth) of node i
T_i^d	Service latency of node i
$\phi_i^*(t)$	Output data flow from node i to router R_i
dp	Maximum depth of a cluster tree
N_c	Maximum number of nodes in each cluster
N_l	Maximum number of leaves per parent node
$\phi_{dp}(t)$	Input flow of each router at depth dp
b_{dp}	Input burst size of each router at depth dp
r_{dp}	Input rate size of each router at depth dp
$\phi^*_{dp-j}(t)$	Output flow of a child router at depth $(dp - j)$
$\rho_{dp-(j+1)}$	Average service rate of a router at depth $(dp - (j + 1))$
$\overline{D}_{dp-j}$	Max. per-hop transmission delay for node at depth $(dp - j)$
D_i^{e2e}	Max. end-to-end transmission delay for node i
D_i^{data}	Max. intra-cluster transmission delay for node i

describes a new energy saving algorithm able to take into account both timing constraints and the network lifetime; it presents the analysis for estimating the maximum end-to-end communication delays, bandwidth and buffers requirements of the network; and finally, it validates the approach by experimental results.

Table 1 summarizes the notation used through the paper.

3. The WBuST protocol

This section describes the proposed protocol in detail. WBuST is a MAC layer protocol that can operate both in single-hop and in multi-hop networks, serving different kinds of communication flows.

3.1. Network model

Network nodes are grouped into n clusters, each denoted by C_i. A node can be of three different types:

- Cluster node. It is a node that may exchange data with other nodes within and outside the cluster.
- Coordinator node. It is a node located at the central area of the cluster in charge of synchronizing and scheduling the cluster nodes to access the wireless medium. Depending on the context, C_i is also used to denote the node coordinator of cluster i.
- Router node. It is a node, denoted by R_i, located in the central area of the cluster in charge of managing the communication with other router nodes.

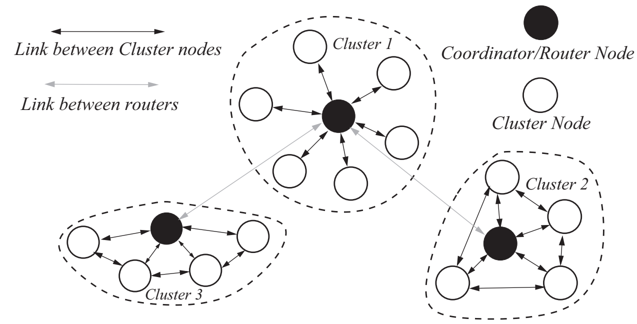


Fig. 1. Example of network structure.

An example of a network is illustrated in Fig. 1. In the rest of the paper, we assume that each cluster contains a set of cluster nodes and a coordinator that also operates as a router. Moreover, nodes are connected either by a peer-to-peer or star topology, depending on the application requirements. In a peer-to-peer topology, any node can communicate with any other node within its communication range, and all nodes are connected to the coordinator. In a star topology, all nodes can only communicate with the coordinator, meaning that any communication between two nodes must pass thorough the coordinator. For instance, in the network illustrated in Fig. 1 the nodes of cluster C_1 are connected using a star topology, whereas the other clusters adopt a peer-to-peer topology.

More complex topologies can be implemented at the network level by connecting the clusters through their coordinator/router nodes. The support offered by WBuST for multi-hop networks is described in Section 3.3.

The network formation problem is not taken into account in this work, since it can be solved using standard techniques available in the literature, see for instance Kumarawadu et al. (2008).

3.2. Intra-cluster communication

The communication among cluster nodes occurs by sharing a periodic Communication Window (*CW*), whose structure is illustrated in Fig. 2. Each *CW* is delimited by a coordination packet, namely the *beacon*, periodically sent by the coordinator. The beacon is used to define the *CW* length, synchronize the nodes and communicate the *CW* schedule.

Each *CW* is divided into slots, whose duration is referred to as *time budget*. Some slots have a specific usage. In particular:

- B_C is the contention slot. It immediately follows the beacon and it is used by cluster nodes to send requests to the coordinator for joining the cluster, reserving a slot, or exchanging control information with the coordinator.
- B_i, with $i = 1, \ldots, n$, is the slot reserved to node i, in which the node can transmit its messages accessing the channel without contention. Its dimension depends on the parameters of message flows.
- B_S is the last slot in the *CW* used by all nodes to enter in sleep mode to save energy.

A slot can be used by a node to transmit both real-time and best-effort traffic. The rules for allocating and managing slots are described in Section 4. Since each node transmits in its own slot, the transmis-

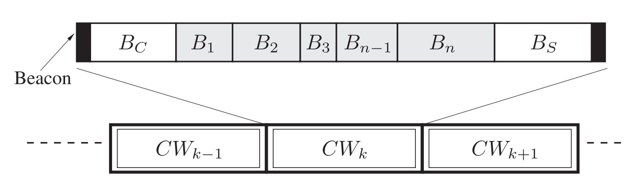


Fig. 2. Intra-cluster communication structure.

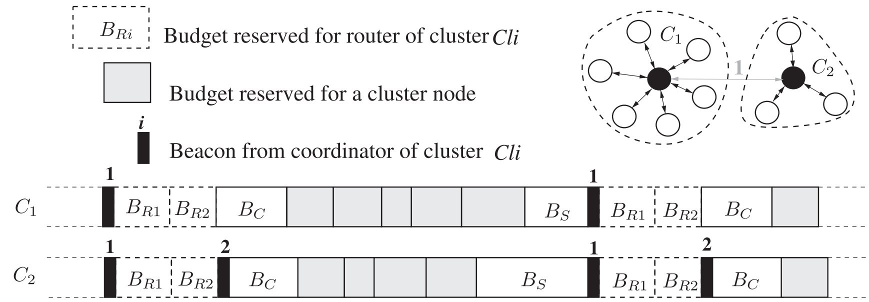


Fig. 3. Example of inter-cluster communication.

sions are collision-free, except within the B_C slot, where nodes willing to communicate with the coordinator contend for accessing the channel using the CSMA/CA scheme available in the IEEE 802.15.4 Std-2011 (2011) standard.

3.3. Inter-cluster communication

Inter-cluster communication is handled by router nodes. Although a single router per cluster is assumed in the following, protocol rules are still valid in the case of multiple routers per cluster.

3.3.1. Cluster-chain topology

The inter-cluster communication is introduced through a simple example consisting of a network composed by two clusters, C_1 and C_2, as shown in Fig. 3. In each cluster C_i, both coordinating and routing functions are carried out by the same node, denoted by R_i.

To achieve a reliable and efficient inter-cluster communication, the following rules must be observed:

- The link between two clusters must be synchronized by the beacon transmitted by one of the cluster coordinators, defined at design time to act as a master. In the example shown in Fig. 3, the inter-cluster synchronization is obtained through the beacon sent by cluster C_1.
- To guarantee a correct inter-cluster communication, both clusters must use the same beacon period, T_b.
- To allow simultaneous communications within different clusters preventing interference, each cluster must use a different radio channel. To use low-cost radio devices, each router is assumed to be equipped with a half-duplex transceiver, which can use one frequency at a time and cannot receive and transmit simultaneously.
- The two routers communicate on the channel allocated to the master coordinator (R_1 in the considered example).
- Each router R_i can transmit real-time and best-effort traffic using a budget B_{R_i} assigned at design time.
- The budget of both routers must be allocated in the *CWs* of both clusters.

The communication between clusters proceeds as follows:

1. At the beginning of each *CW* in C_1, R_2 listens to the channel allocated to R_1 to receive the beacon from it.
2. Once the beacon is received, both routers are synchronized and the inter-cluster communication can take place within the slots B_{R1} and B_{R2} reserved to them in each *CW*. For the master coordinator, the budgets for inter-cluster communication are placed at the beginning of its *CW*, right after the beacon, while for the other router they are placed at the end of the window.

At the end of budget B_{R2}, R_2 switches to the channel assigned to its cluster and sends its beacon to synchronize the inter-cluster communication, that starts with the contention budget B_C (see Fig. 3). Instead, in cluster C_1, the inter-cluster communication starts immediately after B_{R2}, because the beacon in C_1 has already been sent at the beginning of the *CW*. Note that each *CW* of C_2 includes the transmission of two beacons on two different channels: one from R_1 and one from R_2. Instead, the *CW* of the master includes only a beacon transmission. It follows that the bandwidth lost due to the protocol overhead (due to beacon transmissions) is greater in C_2 than in C_1.

The two-cluster topology can easily be extended to n clusters connected as a chain. As before, the link between two adjacent clusters has to be managed by one of the cluster coordinators. For instance, the link between two adjacent clusters could be managed by the coordinator with the smallest index.

3.3.2. Cluster-tree topology

The second topology considered in this work is the cluster-tree structure, an example of which is shown in Fig. 4, including seven clusters forming a binary tree. To guarantee a correct inter-cluster communication, the links between a parent node and its leaves is synchronized by the coordinator of the parent node. For example, the links C_1–C_2 and C_1–C_3 are both coordinated by C_1. In the figure, the number on each link identifies the link coordinator.

The inter-cluster communication scheduling starts from the root of the tree, going downward to the tree leaves. The schedule of the inter-cluster communication, for the left branch, is represented in Fig. 5. The inter-cluster coordination in the right branch of the tree, not shown due to the lack of space, is performed in the same way; for further details, an interested reader can refer to Franchino and Buttazzo (2012). At the beginning of each *CW* of cluster C_1, R_1 sends its beacon and both R_2 and R_3 are listening to its channel. Once the beacon is received, C_1, C_2 and C_3 are synchronized and can start transmitting their inter-cluster traffic within slots B_{R1}, B_{R2} and B_{R3}, in the corresponding *CW*. After transmitting its messages in B_{R2}, R_2 switches on its cluster's channel and transmits the beacon to coordinate the inter-cluster communication with C_4 and C_5, as well as its inter-cluster communication. Note that at the beginning of each *CW* in C_2, R_4 and R_5 are listening to the channel of C_2 to get the corresponding beacon. Once

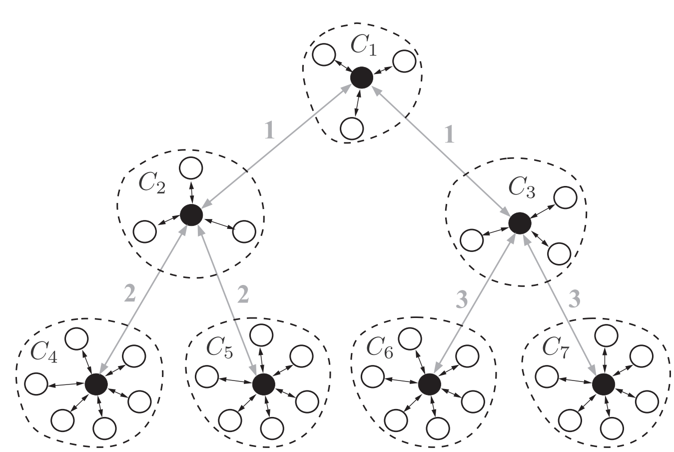


Fig. 4. Cluster-tree topology example.

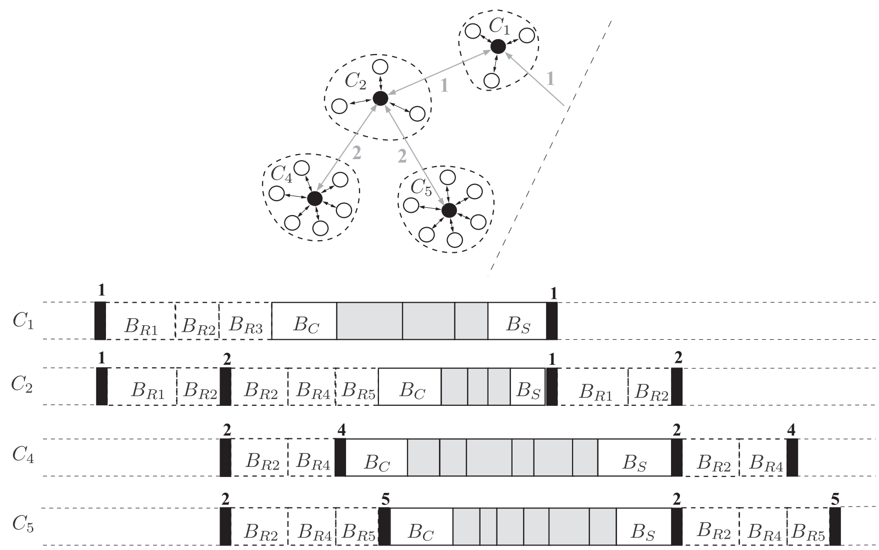


Fig. 5. Communication schedule for the left branch of the cluster-tree.

this is received, R_2, R_4, and R_5 can communicate within slots B_{R2}, B_{R4} and B_{R5}. After the inter-cluster slots, R_4 and R_5 transmit their local beacons to coordinate the inter-cluster communication. Note that, even if it is not connected to router R_4, to keep the coordination with its parent router R_2 and wait for its transmission turn, R_5 must allocate budget B_{R4} in its *CW*.

4. Budget allocation and protocol properties

This section analyzes the timing properties of the protocol in order to perform real-time guarantee tests on message deadlines. In particular, worst-case transmission times are derived for a number of bandwidth allocation schemes.

Referring to the *CW* structure shown in Fig. 1, it can be noted that cluster nodes access the channel, one by one, in a circular fashion, and the access time of node i is limited by the slot budget B_i. In other words, the channel access is regulated by a weighted round robin policy, where each budget B_i is proportional to (weighted with) the length of stream S_i. Since each node uses its budget to transmit both real-time and best-effort traffic, it is not difficult to see that the WBuST scheduling policy is equivalent to that of the BuST (Franchino et al., 2007) protocol (designed for wired networks), hence the results obtained for BuST can be exploited for the analysis of WBuST.

BuST is based on a token-passing scheduler where network nodes form a logical ring by exchanging a control packet, the token, in a circular fashion, and only the node holding the token can access the channel. Once a node gets the token, it can transmit its traffic, real-time and best-effort, for a time no greater than its time budget. Then, the token is passed to the next node along the logical ring. If a node has no traffic to deliver or it finishes its transmission without consuming the entire budget, the token is passed to the next node. In this way, the unused bandwidth is implicitly reused by the other nodes, so reducing the time interval between two consecutive channel accesses by the same node. In WBuST, a similar behavior is obtained by the transmission of the beacon, which can be seen as a token that synchronizes the nodes and defines the structure of the *CWs*. With respect to BuST, however, there are a couple of differences: first, by using a single beacon per *CW*, the protocol overhead is reduced; second, if a node does not use its budget, the following nodes cannot advance their transmission. Although WBuST does not allow the nodes to advance their channel access, as done in BuST, a bandwidth reclaiming method that emulates the token passing mechanism will be described in Section 4.3.

4.1. Traffic model

WBuST manages two types of traffic: real-time sporadic and best-effort traffic. The sporadic traffic of a node i is modeled through a sporadic message stream S_i, characterized by three parameters:

- the maximum length M_i, measured in time units, of the messages generated by the stream;
- the relative deadline D_i associated to each message of the stream;
- the minimum inter-arrival time T_i (equivalent to the period in case of periodic traffic) between the generation of two consecutive messages in the stream.

The time unit corresponds to the time needed to send a packet and receive its associated acknowledgment, hence all stream and protocol parameters are expressed in number of packet transactions.

The ratio $U_i = M_i/T_i$ denotes the bandwidth of the stream S_i. The best effort-traffic is generated by non real-time messages without specific timing requirements. Furthermore, a node can also generate best-effort traffic without considering any particular traffic model. The traffic generated in each cluster is defined by a set of n streams $\Gamma = \{S_1, S_2, \ldots, S_n\}$, and the total channel bandwidth U required by Γ is defined as

$$U = \sum_{i=1}^{n} U_i. \tag{1}$$

In addition to the time budgets B_C, B_i, B_{R_j} and B_S, the following protocol parameters are defined:

- T_b is the beacon period which defines the dimension of each *CW*.
- the Target Beacon Time (T_{BT}) is the greatest value for T_b that guarantees the correct operation of WBuST.τ
- is the protocol overhead, that is, the time in each *CW* that cannot be used by nodes to transmit their messages. It is given by the time needed to transmit the beacon plus other components, such as the Inter-Frame Spacing (IFS) required between consecutive packet transmissions to leave a receiving node the time to process a packet before receiving the next one. Note that, as highlighted later in the paper, in some clusters the *CW* can contain two beacons that have to be accounted in the overhead.
- $\alpha = \tau/T_b$ is the bandwidth lost due to the protocol overhead.

To guarantee the correct operation of the protocol, T_{BT} has to be not greater than the minimum relative deadline $D_{min} = \min_i(D_i)$. This is a

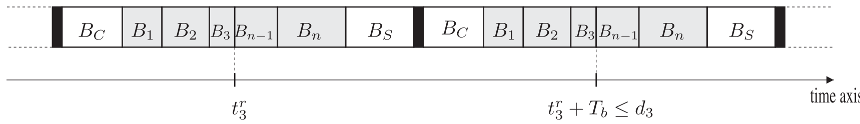


Fig. 6. Example showing the constraint on T_{BT}.

necessary and sufficient condition to guarantee at least one packet transmission for each node i, between the time t_i^r a new message in S_i is produced for transmission and its absolute deadline $d_i = t_i^r + D_i$. Fig. 6 shows the maximum delay between the time t_3^r a new message is ready in stream S_3 and the end of the budget B_3 in the next *CW*. Such a delay is equal to $T_b \le T_{BT}$ and has to be no greater than D_3. Supposing $D_{min} = D_3$ and $T_b = T_{BT}$, the necessary and sufficient condition that guarantees at least one packet transmission becomes: $t_3^r + T_b \le t_3^r + D_3$. Note that a message in S_i experiences the worst-case transmission delay when it becomes ready just after the end of the budget assigned to node i.

To guarantee a correct operation of the protocol, any selection of communication parameters must satisfy the following constraints.

Definition 4.1 (*Bandwidth Constraint***).** For each network cluster, the total channel bandwidth allocated to nodes must not exceed the available bandwidth:

$$\sum_{i=1}^{n} \frac{B_i}{T_{BT}} \le 1 - \alpha \tag{2}$$

where T_{BT} is the maximum beacon transmission period, that is, the maximum dimension of the CW.

The *Bandwidth Constraint* is necessary to guarantee a stable operation of the protocol.

Definition 4.2 (*Deadline Constraint***).** For a stream S_i, let WC_i be the maximum time interval between the generation of a message and the time at which its transmission is completed, namely the worst-case transmission time. Then, the Deadline Constraint requires that for any i:

$$WC_i \le D_i \tag{3}$$

where D_i is the relative deadline of stream S_i.

The *Deadline Constraint* is necessary to guarantee that all messages are sent by their deadline. Note that, while the deadline D_i is imposed by the application, WC_i depends on protocol parameters, such as the budget B_i and the beacon period T_b.

In the rest of the paper, only streams with D_i=T_i are taken into account. In the case of streams with $D_i < T_i$, the results can be extended by replacing T_i with D_i. The case of $D_i > T_i$ is not treated and it will be part of future work.

To guarantee a correct transmission of real-time traffic, time budgets have to be properly dimensioned as shown by the following lemma, which provide the worst- case transmission time WC_i for any message generated by stream S_i.

Lemma 4.3. *Under the* WBuST *protocol, if* $T_i \ge T_{BT}$ *and the network traffic includes both real-time and best-effort streams, it holds that, for* $i = 1, \ldots, n$,

$$WC_i = \left\lceil \frac{M_i}{B_i} \right\rceil T_{BT}.$$

Proof. If $D_i \le T_i$, then when the j-th message of stream S_i is ready, it is the only one in the stream transmission queue. In the worst-case, the node is using the portion δ_i of the budget B_i left by message $(j-1)$ to transmit best-effort traffic. In this case, message j cannot preempt the transmission, and in the worst-case has to wait for the end of the best-effort traffic transmission plus the time necessary to get again the right to access the channel in the next *CW*, that is, it has to wait for $\delta_i + T_{BT} - B_i$ units of time before the node can start transmitting its message. It follows that, in the worst-case, the time to deliver a portion B_i of the message is $\delta_i + T_{BT} - B_i + B_i = T_{BT} + \delta_i$. To transmit an entire message of length M_i, a node needs $\lceil \frac{M_i}{B_i} \rceil$ *CW*s, and $\delta_i = \lceil \frac{M_i}{B_i} \rceil B_i - M_i$. Two cases are possible:

1. If $\frac{M_i}{B_i} \in \mathbb{N}$, then $\lceil \frac{M_i}{B_i} \rceil = \frac{M_i}{B_i}$ and $\delta_i = 0$. Hence, the message is completely delivered after $\lceil \frac{M_i}{B_i} \rceil T_{BT}$ units of time, and the lemma follows.
2. If $\frac{M_i}{B_i} \notin \mathbb{N}$, then after $\lfloor \frac{M_i}{B_i} \rfloor T_{BT} + \delta_i$, the budget B_i is consumed and the node stops transmitting the message. The remaining portion $M_i - \lfloor \frac{M_i}{B_i} \rfloor B_i$ can be delivered after $T_{BT} - B_i$ units of time in the following *CW*. In fact, the transmission of the j-th message can be completed because: $M_i - \lfloor \frac{M_i}{B_i} \rfloor B_i < M_i - (\frac{M_i}{B_i} - 1)B_i = B_i$.
3. Since $\frac{M_i}{B_i} \ne \lceil \frac{M_i}{B_i} \rceil$, then $\lceil \frac{M_i}{B_i} \rceil = \lfloor \frac{M_i}{B_i} \rfloor + 1$. Hence, $\delta_i = \lceil \frac{M_i}{B_i} \rceil B_i - M_i = \lfloor \frac{M_i}{B_i} \rfloor B_i + B_i - M_i$, and the worst-case transmission time WC_i for each message generated by stream S_i can be computed by adding up all these terms:

$$WC_i = \left\lfloor \frac{M_i}{B_i} \right\rfloor T_{BT} + \delta_i + + T_{BT} - B_i + M_i - \left\lfloor \frac{M_i}{B_i} \right\rfloor B_i$$
$$= \left(\left\lfloor \frac{M_i}{B_i} \right\rfloor + 1 \right) T_{BT} + \delta_i - \delta_i = \left\lceil \frac{M_i}{B_i} \right\rceil T_{BT}. \square$$

From the previous lemmas and the *Deadline Constraint*, it is clear that the guarantee of message deadlines depends on the budgets reserved to the nodes. Such an issue is discussed in the following section.

4.2. Budget allocation schemes

The *Deadline Constraint* and the *Protocol Constraint* can be satisfied by a proper allocation of time budgets. Several scheme have been proposed in the literature for timed token protocols (Zhang et al., 2004), which can also be used in this context. In particular, this work will focus on the analysis of Proportional Allocation (PA), Normalized Proportional Allocation (NPA) and Modified Local Allocation (MLA) (Chan et al., 2001) schemes. Such schemes are listed in Table 2 together with their assignment rule, where $\beta_i = T_i/T_{BT}$.

Following the classification proposed by Chen et al. (1998), the first two schemes, PA and NPA, are T_{BT} *– partitioning* schemes in that they assign to each node a portion of T_{BT}, which is the maximum value for the beacon period. The MLA scheme belongs to the set of M_i *– partitioning* schemes, since it assigns each node i a partition of the message length M_i.

The performance of each Budget Allocation Scheme (BAS) has been extensively analyzed for timed token protocols (Zhang et al., 2004;

Table 2
Budget allocation schemes.

Budget alloc. scheme	Assignment rule	U^*
PA	$B_i = U_i(T_{BT} - \tau)$	$\frac{1-3\alpha}{2(1-\alpha)}$
NPA	$B_i = \frac{U_i}{U}(T_{BT} - \tau)$	$\left\lfloor \frac{\beta_{min}}{\beta_{min}+1} \right\rfloor (1-\alpha)$
MLA	$B_i = \frac{M_i}{\lfloor \beta_i \rfloor}$	$\left\lfloor \frac{\beta_{min}}{\beta_{min}+1} \right\rfloor (1-\alpha)$

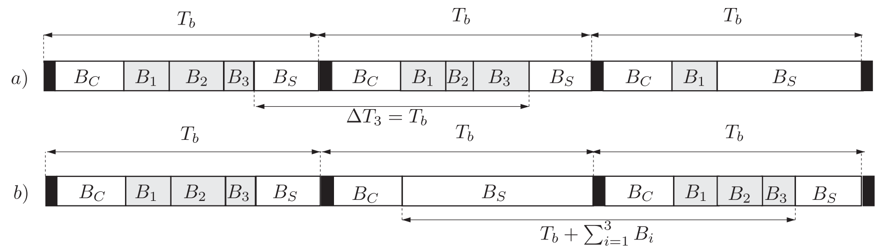


Fig. 7. Example of bandwidth reclaiming mechanism.

Chan et al., 1997, 2001; Chen et al., 1998) and BuST (Franchino et al., 2008, 2010a). Mostly, the metric adopted to compare the allocation schemes is the Worst Case Achievable Utilization (WCAU), which is the maximum channel bandwidth U^* such that, for any stream set having total channel utilization $U \le U^*$, the scheme can guarantee that all message deadlines will be met.

The third column of Table 2 shows the WCAU of each budget allocation scheme considered in this work. Note that, since the BuST and the WBuST protocols implement the same scheduling policy, the U^* derived for BuST in each scheme is still valid for WBuST, hence the formulas shown in the table are taken from the literature (Franchino et al., 2010a). Note that all schemes but PA have the same WCAU, which depends on $\beta_{min} = \min_i(\beta_i) = \min_i(T_i/T_{BT})$. In particular, given the minimum stream period T_{min}, the lower T_{BT} the greater U^*. It follows that, to guarantee more bandwidth for real-time streams, it is necessary to keep T_{BT} as small as possible. Conversely, when decreasing T_{BT}, the protocol overhead, and consequently α, increase; hence, the value of T_{BT} has to be carefully chosen.

Concerning best-effort traffic, BuST and consequently WBuST can guarantee a minimum bandwidth for non-real-time traffic as long as the bandwidth U required by real-time streams is less than $1 - \alpha$, so avoiding the risk of starvation (Franchino et al., 2007, 2008).

Since each *CW* also contains the contention slot (B_C) and sleep slot (B_S), to verify the message schedulability through the WCAU, it is necessary either to add these slots to the overhead τ, or to create two message streams with dummy parameters, namely S_C and S_S, and then assigning B_C and B_S with the same BAS used for the other streams.

All allocation schemes shown in Table 2 work for inter-cluster communication (single-hop networks), whereas for inter-cluster communication only the NPA scheme can be adopted. The reason is that, to guarantee the inter-cluster synchronization as described in Section 3.3, the *CWs* of all clusters must have the same dimension, that is, the beacon period must be the same for all clusters. It means that, given T_{BT}, for a multi-hop network formed by n clusters C_j with $j \in \{1, \ldots, n\}$ it must be:

$$T_b = \tau + B_C + \sum_i B_i + B_S = T_{BT}. \tag{4}$$

To guarantee this requirement, it is necessary to select $T_{BT} \le \min_{i,j}(D_i^j)$, where D_i^j is the relative deadline of stream S_i in cluster C_j, and to use an allocation scheme that satisfies Eq. (4). The only scheme that satisfied such a requirement is NPA. For the other schemes, if $U < 1$ then $T_b < T_{BT}$.

4.3. Bandwidth reclaiming

When a node does not use its reserved slot completely, the left budget can be reclaimed to increase the transmission time of others nodes. Under WBuST, this situation can occur every time a node scheduled to access the channel has not messages to deliver or has less traffic to transmit than expected. To overcome this problem, a bandwidth reclaiming mechanism can be implemented as follows. When node i saves some budget, the unused budget is added to that of node $i + 1$, and so on, until the unused budget by all nodes is added to the sleep budget. In this way, since the sum of node budgets is constant and equal to $\sum_{i=1}^{n} B_i$, the dimension of each *CW* is constant and equal to T_b.

When the reclaiming mechanism is used, the bandwidth available for each node i is:

$$BW_i = \frac{B_i + B_{i-1}^l}{T_b} \tag{5}$$

where B_{i-1}^l is the amount of the budget saved by node $i - 1$ and by all previous nodes. Note that the bandwidth saved in a *CW* can only be reclaimed in the same *CW*. In the worst-case, all the unused bandwidth is reclaimed in the sleep slot, and hence it is entirely used to save energy.

Fig. 7(a) shows an example in which in the second *CW* the budget left by node 2 is used by node 3. In the third *CW*, nodes 2 and 3 do not transmit, so their budgets are added to B_S. Fig. 7(b) shows another example in which all the nodes have not traffic to deliver in the second *CW*, so that all the budgets are added to B_S. Observe that the beacon period, hence the dimension of each *CW*, is constant and equal to T_b.

To implement the bandwidth reclaiming mechanism, each node can add the transmission length to every packet header, such that the following node can derive the amount of budget unused by the preceding one. In practice, each node i starts transmitting as soon as node $i - 1$ finishes its transmission, and continues to transmit until the end of slot B_i. In this case, the transmission time of node i will be $B_i + B_{i-1}^l$, where $B_{i-1}^l > 0$ if node $i - 1$ finished before the beginning of B_i in the current *CW*, otherwise $B_{i-1}^l = 0$. In case a node has no traffic to transmit, it sends a short packet with transmission length equal to 0 to the following node.

The subsequent lemma provides the worst-case transmission time for any message of stream S_i, when the reclaiming mechanism is implemented. The result provided below also holds in the case the nodes have both real-time and best-effort traffic to deliver.

Lemma 4.4. *Under the* WBuST *protocol with the bandwidth reclaiming mechanism, for* $i = (1, \ldots, n)$, *if* $T_i \ge T_{BT} + \sum_{j=1}^{i} B_j$, *then*

$$WC_i = \left\lceil \frac{M_i}{B_i} \right\rceil (T_{BT} - B_i) + M_i + \sum_{j=1}^{i} B_j. \tag{6}$$

Proof. Since for $i = (1, \ldots, n)$, $T_i \le D_i$, when a message is generated by stream S_i it is the only one in the transmission queue. The situation that causes the maximum transmission delay is shown in Fig. 7(b). Suppose that at the end of budget B_C (as shown in the second *CW*), all nodes have no messages to transmit, hence all budgets are added to B_S, so that the total sleep budget in the current *CW* is $B_S + B_n^l = B_S + \sum_{i=1}^{n} B_i$. Suppose that, just after the beginning of the sleep budget in the current *CW*, a new message is ready to be transmitted in all the nodes, such that the transmission of these messages will start at the next channel access of the corresponding node. Because of the delay imposed by the sleep budget plus the reclaimed budgets, added to the sleep slot, the time interval before the next channel access of node 1 will be $B_S + \sum_{i=1}^{n} B_i + \tau + B_C = T_b$, for node 2 will be $B_S + \sum_{i=1}^{n} B_i + \tau + B_C + B_1 = T_b + B_1$, and so forth. Hence, the minimum interval before the next channel access of node i is

$$T_b + \sum_{j=1}^{i-1} B_j \le T_{BT} + \sum_{j=1}^{i-1} B_j = (T_{BT} - B_i) + \sum_{j=1}^{i} B_j.$$

It follows that, in the worst case, node i accesses the channel after $(T_{BT} - B_i) + \sum_{j=1}^{i} B_j$ time units and transmits for a time no greater than B_i. It means that, node i has the possibility to transmit at least one packet if

$$T_i \ge T_{BT} + \sum_{j=1}^{i-1} B_j + B_i = T_{BT} + \sum_{j=1}^{i} B_j.$$

As shown in the proof of Lemma 4.3, without the reclaiming mechanism, the maximum channel access delay for node i is $(T_{BT} - B_i)$. Instead, as just shown above, when the bandwidth reclaiming mechanism is implemented, node i can suffer an additional access delay that, in the worst case, is given by $\sum_{j=1}^{i} B_j$. Hence, to derive the worst-case transmission time WC_i it is sufficient to add such an additional delay to the computation of WC_i provided by Lemma 4.3, that is

$$WC_i = \left\lceil \frac{M_i}{B_i} \right\rceil (T_{BT} - B_i) + M_i + \sum_{j=1}^{i} B_j. \square$$

From the lemma above it follows that, for any node i, the worst-case interval between two consecutive channel accesses depends on the position of B_i within the *CWs*: the smaller the node index (i), the shorter the transmission delay of node i. This property can be taken into account when selecting the node indexes. In general, the index assignment should be based on the message deadline: the shorter the deadline the smaller the node index.

5. Energy saving mechanism

As already mentioned, a sleep slot is allocated at the end of each *CW* to allow cluster nodes to turn off their radio transceiver. This section describes how to calculate the dimension of this slot to guarantee a desired lifetime for each network cluster.

Before describing how to compute the sleep slot, it is worth showing how to identify the sources of energy consumption mentioned in the introduction. Fig. 8 shows the *CW* of cluster C_2 in the example of the cluster-tree structure described in Section 3.3.2. Starting from the beginning of the *CW*, the first energy waste for the nodes in C_2 is due to *Control Packet Overhead* when receiving the beacon from the cluster coordinator. Then, the energy due to *Overhearing* is wasted during slots B_{R2}, B_{R4} and B_{R5}, which are only used by the coordinator to exchange messages with clusters C_4 and C_5. Within slot B_C, the cluster nodes exchange control messages with the coordinator, hence they consume further energy due to *Control Packet Overhead* or due to *Overhearing* because a control message could be addressed to a specific node. Moreover, since in slot B_C the channel is accessed by a CSMA/CA algorithm and sometimes the nodes and the coordinator do not need to exchange messages, nodes can also waste energy due to *Collision* and *Idle Listening*. A similar reasoning can be applied to identify the energy wasted during the slots reserved to cluster nodes to transmit data messages. The only difference is that in the reserved slots transmissions are collision-free, thus, energy cannot be wasted due to *Collision*. In the last part of the *CW*, defined by the transmission of the beacon from C_1 and budgets B_{R1} and B_{R2} reserved to the inter-cluster communications between C_2 and C_1, nodes waste energy due to *Idle Listening* because transmissions occur on the channel of cluster C_1, while nodes in C_2 are listening on a different channel. Note that, since cluster nodes are not involved in the inter-cluster communication, they could save more energy going on sleep mode during the budgets allocated to inter-cluster traffic.

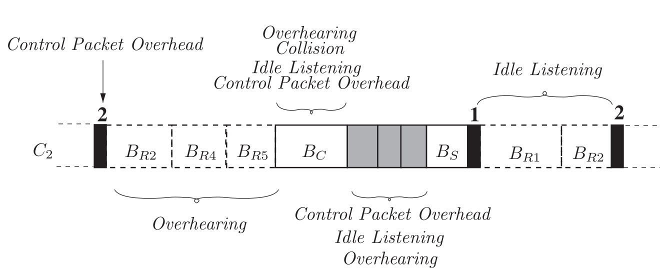


Fig. 8. Sources of energy consumption with WBuST.

To calculate the average energy consumption of a node, observe that, in each *CW*, a node i transmits for a time no greater than B_i, it is in receiving mode for a time no greater than $T_b - B_i - B_S$, and in sleep mode for B_S time units. If P^{tx} is the power dissipated by a node in transmission mode, P^{rx} in receiving mode, and P^{sl} in sleep mode, then the average energy wasted by node i after t units of time is:

$$E_i(t) = [P^{tx}B_i + P^{rx}(T_b - B_i - B_S) + P^{sl}B_S]\frac{t}{T_b}. \tag{7}$$

Since the time needed to join a cluster is usually negligible with respect to the time a node operates in the cluster, the energy wasted during the joining phase is not considered in the equation above. Moreover, the coordinator node is assumed to be mains powered, thus, its energy consumption is not a concern.

In the following, we show how to guarantee a minimum lifetime L_j^m for each cluster C_j by properly dimensioning the sleep budget. The cluster lifetime is defined as the time instant at which k nodes of the cluster run out of energy. The value of k depends on the application. For instance, if cluster C_j includes a set of n=10 sensor nodes that sample the same physical quantity, e.g. the temperature, and the application requires that for each sampling period at least r=3 samples (from r different nodes) are needed for an accurate measure of the temperature, then the cluster lifetime L_j can be defined as the time at which $k = n - r + 1 = 8$ nodes exhaust their energy.

Given a desired lifetime L_j^m for each cluster C_j, Eq. (7) allows calculating the minimum sleep slot that can guarantee L_j^m. For the sake of simplicity, it is assumed that at the system startup all nodes have the same amount of available energy E^0.

To guarantee that in cluster C_j all nodes will operate at least for L_j^m time units, it is sufficient to impose that for any i, $E_i(L_j^m) \le E^0$, that is:

$$[P^{tx}B_i + P^{rx}(T_b - B_i - B_S) + P^{sl}B_S]\frac{L_j^m}{T_b} \le E^0. \tag{8}$$

Rearranging the terms, it is possible to derive the sleep budget B_S as a function of B_i:

$$\forall\ i\ B_S(B_i) \ge \frac{(P^{tx} - P^{rx})B_i + \left(P^{rx} - \frac{E^0}{L_j^m}\right)T_b}{P^{rx} - P^{sl}} \tag{9}$$

Considering $(P^{tx} - P^{rx}) \ge 0$, if k=1, since $B_S(B_i)$ is a straight line growing with B_i, given the greatest node budget $B_{max} = \max_i(B_i)$, to guarantee the desired lifetime is sufficient to select B_S as follows:

$$B_S \ge \frac{(P^{tx} - P^{rx})B_{max} + \left(P^{rx} - \frac{E^0}{L_j^m}\right)T_b}{P^{rx} - P^{sl}} \tag{10}$$

If k=2, B_S is computed considering the second greatest node budget; if k=3, it is necessary to consider the third one, and so on. Instead, if $(P^{tx} - P^{rx}) < 0$, since $B_S(B_i)$ is a straight line decreasing with B_i, the k-th smallest budget should be considered to compute B_S.

After computing the sleep budget, the next step is to verify the stream set feasibility. Considering B_S as a budget assigned to a dummy stream $S_S(M_S, D_S, T_S)$, where parameters M_S, $D_S = T_S$, and $U_S = M_S/T_S$

depend on the allocation scheme, it is possible to assess the message schedulability by the methods shown in Section 4.2. In particular, to exploit the worst-case achievable utilization tests, for both the PA and the NPA schemes it is necessary to derive U_S such that, if $U + U_S \le U^*$, then all message deadlines will be met.

From the assignment rules of Table 2, it is possible to derive the parameters of stream S_S for the PA and NPA schemes. To derive U_S with the PA scheme it is sufficient to impose $B_S = U_S(T_{BT} - \tau)$, that is:

$$U_S = \frac{B_S}{T_{BT} - \tau}. \tag{11}$$

For NPA, $B_S = \frac{U_{SI}(T_{BT} - \tau)}{U + U_{SI}}$, thus, it turns out:

$$U_S = \frac{UB_S}{T_{BT} - \tau - B_S}. \tag{12}$$

For M_i-partitioning schemes, such as MLA, the Deadline Constraint (Inequality (3)) is always met for any stream set with $U \le 1$ (Franchino et al., 2008). Hence, it is not necessary to derive U_S, but to guarantee the stream set schedulability it is sufficient to verify that the Bandwidth Constraint (Inequality (2)) holds, that is:

$$\sum_{i=1}^{n} \frac{B_i}{T_{BT}} + \frac{B_S}{T_{BT}} \le 1 - \alpha. \tag{13}$$

Finally, if message deadlines cannot be met for a given value of B_S, it is possible to adopt an elastic approach (Franchino et al., 2010b), where the stream utilization U_i is not fixed, but can range in an interval $[U_i^{min}, U_i^{max}]$, varying the slot B_i in the range $[B_i^{min}, B_i^{max}]$, selected such that both the message deadlines and lifetime are met. The development of this is idea is part of future work.

6. Multi-hop delay analysis

This section considers the inter-cluster communication and describes a method to derive the maximum end-to-end communication delay of message streams, the bandwidth and buffering requirements for the router nodes of a cluster-tree network.

Data traffic in cluster-tree networks can be formed by both upstream and downstream flows. Usually, downstream messages carry queries or control information from root to cluster nodes and upstream flows carry critical messages, e.g. sensor data, from cluster nodes to the root node. This work focuses on upstream flows leaving the analysis of downstream flows to future work.

The method presented in the following is based on Network Calculus (Leboudec and Thiran, 2001) and is derived from the general methodology proposed by Koubaa et al. (2006).

Using the Network Calculus representation, in each cluster, the maximum individual data flow that a cluster node i can generate is constrained by the arrival curve (Leboudec and Thiran, 2001):

$$\phi_i^{data}(t) = b_i + r_i^{data} \cdot t, \tag{14}$$

where b_i is the maximum data burst size, r_i^{data} is the average data rate of the flow (measured in bits per seconds), and t is the current time. Note that, ϕ_i^{data} represents an upper-bound on the cumulative arrival function of the flow and can model both real-time and best-effort traffic. The relation between the arrival curve model expressed in Eq. (14) and the stream models defined in Section 4.1 has been derived by Koubaa and Song (2004). In particular, the relationship between the parameters of the two models is:

$$r_i^{data} = U_i b_i = U_i(T_i + J_i), \tag{15}$$

where J_i is the jitter on message inter-arrival time. Note that, for sporadic streams $U_i = M_i/T_i$ and, if J_i=0, then b_i=M_i.

The service curve guaranteed by the protocol to each cluster node is (Leboudec and Thiran, 2001):

$$\gamma_i(t) = \rho_i^{data}(t - T_i^d)^+, \tag{16}$$

where ρ_i is the average service rate (bandwidth), $T^d{}_i$ is the service latency and $(x)^+ = \max(0, x)$.

According to Eqs. (14) and (16), as shown in Koubaa et al. (2006), the output data flow from a cluster node i to the cluster router is defined by:

$$\phi_i^*(t) = \phi_i^{data}(t) + r_i^{data} \cdot T_i^d. \tag{17}$$

For the purpose of our analysis, a cluster-tree network is modeled by three parameters:

- dp is the maximum depth of the tree, which defines the maximum number of hops between any cluster node and the tree root (i.e., the cluster router/coordinator located at depth 0).
- N_c is the maximum number of nodes in each cluster.
- N_l is the maximum number of leaves per parent node, namely the maximum number of child routers/clusters that can be associated with a parent router/cluster.

Fig. 9 shows an example of a cluster-tree network with dp=3, N_c=3, N_l=2 and where the tree root is represented by the router R_1. Note that this representation of a cluster-tree is similar to that used in Koubaa et al. (2006) and slightly different from that of Fig. 4.

Without loss of generality, assuming that all clusters have the same number of nodes equal to N_c, at depth dp the total input flow of each cluster router is (Koubaa et al., 2006):

$$\begin{aligned}\phi_{dp}(t) &= \sum_{i=1}^{N_c} \phi_i^*(t) \\ &= \sum_{i=1}^{N_c} (\phi_i^{data}(t) + r_i^{data} \cdot T_i^d) \\ &= \sum_{i=1}^{N_c} (b_i + r_i^{data} \cdot t + r_i^{data} \cdot T_i^d) \\ &= \sum_{i=1}^{N_c} (b_i + r_i^{data} \cdot T_i^d) + \sum_{i=1}^{N_c} r_i^{data} \cdot t,\end{aligned} \tag{18}$$

where the resulting burst size is:

$$b_{dp} \triangleq \sum_{i=1}^{N_c} b_i + r_i^{data} \cdot T_i^d, \tag{19}$$

and the resulting rate of the aggregated N_c cluster nodes is:

$$r_{dp} \triangleq \sum_{i=1}^{N_c} r_i^{data}. \tag{20}$$

The input flow $\phi_{dp}(t)$ of a router at depth dp is forwarded to its parent router at depth $dp - 1$. The service curve allocated by this last to its child router is (Koubaa et al., 2006):

$$\gamma_{dp-1}(t) = \rho_{dp-1}(t - T_{dp-1}^d)^+. \tag{21}$$

Hence, the output flow of the router at depth $dp - 1$ is:

$$\phi_{dp-1}^*(t) = \phi_{dp}(t) + r_{dp} \cdot T_{dp-1}^d. \tag{22}$$

Starting from Eqs. (18) and (22), by recurrence, it turns out that the input flow of each router at depth $(dp - j)$ is (Koubaa et al., 2006):

$$\phi_{dp-j}(t) = b_{dp-j} + r_{dp-j}t \tag{23}$$

where

$$b_{dp-j} \triangleq \sum_{i=0}^{j} N_l^i \cdot b_{dp} + \sum_{i=0}^{j-1} N_l^{j-i} \sigma_{dp-(i+1)}, \tag{24}$$

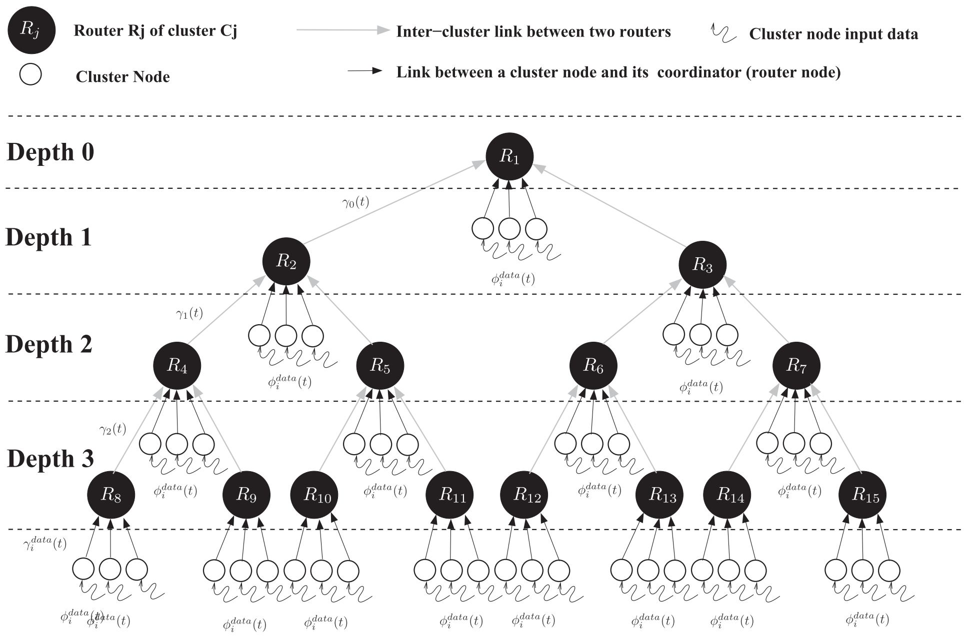


Fig. 9. Cluster tree network example

is the input burst of a each router at depth $(dp - j)$;

$$r_{dp-j} \triangleq \sum_{i=0}^{j} N_l^i \cdot r_{dp} \tag{25}$$

is the input rate of the router at depth $(dp - j)$; and σ_{dp-k} is defined as follows: $\sigma_{dp-k} = (\sum_{i=0}^{k-1} N_l^i) \cdot r_{dp} \cdot T_{dp-k}^d$.

In the same way, the output flow of a child router at depth $(dp - j)$ with a service curve $\gamma_{dp-(j+1)}(t)$ is defined by (Koubaa et al., 2006):

$$\phi_{dp-j}^*(t) = \phi_{dp-j}(t) + \sigma_{dp-(j+1)}. \tag{26}$$

6.1. Buffering and bandwidth requirements

To guarantee a bounded delay in inter-cluster communication, the bandwidth allocated to each router should be no less than the corresponding input rate. In particular, for a router at depth $dp - (j + 1)$, it must be:

$$\rho_{dp-(j+1)} \geq r_{dp-j}. \tag{27}$$

From Eqs. (23) and (26), it turns out:

$$r_{dp-j} = r_{dp-j}^* = \sum_{k=0}^{j} N_l^k \cdot r_{dp}. \tag{28}$$

Moreover, each router needs a minimum buffer size to store all incoming data messages to avoid buffer overflow. Considering a router at depth $dp - j$, its minimum buffer size must be greater than the burst size b_{dp-j} of the input flow $\phi_{dp-j}(t)$. Notice that, from Eq. (24), the first term of b_{dp-j} depends on the input burst at depth $dp - j$ and is a function of both the number of child routers N_l and the depth at which the router is placed; the second term depends on the service curve guaranteed to child routers and expresses the cumulative effect of the service latency at each depth.

6.2. End-to-end delay analysis

From Network Calculus theory, through the equations used to compute the input and output flows at each router it is possible to derive the maximum per-hop transmission delay $\overline{D}_{dp-j}$ for a node in a cluster at depth $dp - j$ (Koubaa et al., 2006):

$$\overline{D}_{dp-j} = \frac{b_{dp-j}}{\rho_{dp-(j+1)}} + T_{dp-(j+1)}^d, \tag{29}$$

where b_{dp-j} is the burst size of $\phi_{dp-i}(t)$, as defined in Eq. (23), $\rho_{dp-(j+1)}$ is the service rate, and $T_{dp-(j+1)}^d$ is the service latency of a router at depth $dp - (j + 1)$.

The maximum end-to-end transmission delay D_i^{e2e} for a message sent by a node i in a cluster at dp is obtained by adding up all maximum per-hop transmission delays (Koubaa et al., 2006):

$$D_i^{e2e} = D_i^{data} + \sum_{k=0}^{dp-1} \overline{D}_{dp-k} \tag{30}$$

$$D_i^{data} = \frac{b_i}{\rho_i^{data}} + T_i^d \tag{31}$$

where D_i^{data} is the maximum inter-cluster transmission delay guaranteed to a cluster node i.

6.3. Illustrative example

The following example shows how to apply the methodology described above to compute the maximum end-to-end communication delays and resource dimensioning under the WBuST protocol.

Let us consider the cluster-tree network represented in Fig. 9, where dp=3, N_c=3, and N_l=2. Assume that each cluster node generates the same amount of data traffic defined by a stream $S^{RT}(M, D = T)$ without jitter (J=0), corresponding to an arrival curve $\phi^{data} = b + r^{data} \cdot t = M + (M/T) \cdot t$. Moreover, assuming that the budgets are assigned by the NPA scheme, it follows that $T_b = T_{BT}$ and for all i $B_i = B = (MT_b/TU)(1 - \alpha)$. Considering the channel utilization in each cluster equal to the WCAU, that is, $U = (1 - \alpha)/2$, and selecting $T_b = T$ it turns out that $B = 2M$. For the sake of simplicity the energy saving is not considered, hence the sleep budgets are not allocated.

Because of the cumulative effect of upstream flows, the maximum bandwidth requirement concerns routers R_2 and R_3, which are connected to router $R1$ in the root cluster. Due to the symmetry of the cluster tree and since all cluster nodes transmit the same amount of traffic, both routers require the same bandwidth, that is, the same budget: $B_{R2} = B_{R3} = B_R$. Observe that both R_2 and R_3 are located at depth 1 and R_1 is located at depth 0. Since the bandwidth allocated to the child routers $R2$ and $R3$ is B_R/T_b, it turns out that the average service rate of R_1 provided to each child router is $\rho_0 = B_R/T_b$. Moreover, according to Eqs. (28), for $R1$ the maximum input rate from each child router is:

$$r_1 = \sum_{k=0}^{2} N_l^k \cdot r_{dp}. \tag{32}$$

From the bandwidth bound expressed by Eq. (27), it is possible to derive the maximum data rate r_{max}^{data} for message streams:

$$\sum_{k=0}^{2} N_l^k \cdot r_{dp} \le \frac{B_R}{T_b}, \sum_{k=0}^{2} N_l^k \cdot \sum_{i=1}^{N_c} r^{data} \le \frac{B_R}{T_b}, r_{max}^{data} = \frac{B_R}{T_b N_c \sum_{k=0}^{2} N_l^k}. \tag{33}$$

The maximum value that can be assigned to router budgets can be calculated by subtracting the budgets and the overhead from the *CW* dimension T_b:

$$2B_R = T_b - B_C - N_c B - \tau = T_b - B_C - 2N_c M - \tau. \tag{34}$$

Substituting the expression (34) in (33) it follows that:

$$r_{max}^{data} = \frac{T_b - B_C - 2N_c M - \tau}{2T_b N_c \sum_{k=0}^{2} N_l^k}. \tag{35}$$

Note that in the equation above the maximum transmission rate depends on the message length M and the beacon period T_b.

In each cluster, any budget B_x provides a service curve γ_x with an average service bandwidth $\rho_x = B_x/T_b$ and a service latency $T_x^d = T_b - B_x$. It follows that all routers at depth $dp - j$ need a minimum budget $B_{dp-j} = T_b \cdot r_{dp-j}$. Thus, Eqs. (27) and (28) allow deriving the minimum budget that must be allocated to routers at each depth so that the associated bandwidth is sufficient to service the input rate. For instance, the minimum budget assigned to routers R_2 and R_3 in the *CW* of root cluster is $B_R = T_b \cdot r_1$. Assuming $T_b = T$, M=1 and observing that $r_i^{data} = r^{data} = M/T$, through Eq. (19), the resulting budget is B_R=21. Thus, the minimum feasible value for the beacon period can be computed as the sum of the elements that form the contention window in the root cluster, i.e., the sum of budgets plus the overhead: $T_b = 2B_R + B_C + \tau + N_c B$. Remembering that $B = 2M$, assuming B_C=1 and $\tau = 1$, it follows that $T_b = 2B_R + B_C + \tau N_c B = 50$. Substituting this value in Eq. (33) and considering a radio transceiver with a transmission rate of 250 kbps, it turns out that the maximum bandwidth that the protocol can guarantee to data streams is: $250\text{ kbps} \cdot r_{max}^{data} = 250\text{ kbps} \cdot 0.02 = 5\text{ kbps}$.

Table 3(a) shows the buffer requirements, the budgets and the input rates (bandwidth requirements) for the routers at each depth of the tree. Note that, as expected, since the routers closer to the root have a higher data input rate, the budget dimension increases as the tree depth decreases.

Table 3(b) shows, in the first column, the maximum transmission end-to-end delay for messages transmitted from a node in a cluster leaf to the root cluster. The second column shows the maximum transmission delay from a node to its cluster router. The rest of the columns show the maximum per-hop delay at each depth. These values are computed through Eqs. (29)–(31).

It worth remembering that all values reported in the tables, except the input rate r_{dp-j}, are expressed in number of packet transactions, as defined in Section 4.1. Thus, for instance, considering a packet dimension of 50 bytes (400 bits), an acknowledgment packet of 10 bytes (80 bits), a maximum interval between a packet and its acknowledgment of 0.2 ms, and radio devices transmitting at 250 kbps, the resulting maximum end-to-end transmission delay is $561.58 \cdot (480/250 + 0.2) = 1190.55$ ms.

Table 3

(a) Bandwidth and buffer requirements and router budgets at each depth

$dp - j$	b_{dp-j}	r_{dp-j} (kbps)	B_{dp-j}
1	61.44	105	21
2	22.56	45	9
3	5.88	15	3

(b) Maximum end-to-end and per-hop delay

D^{e2e}	D^{data}	$\overline{D}_1$	$\overline{D}_2$	$\overline{D}_3$
561.58	75	175.28	166.3	145

7. Experimental results

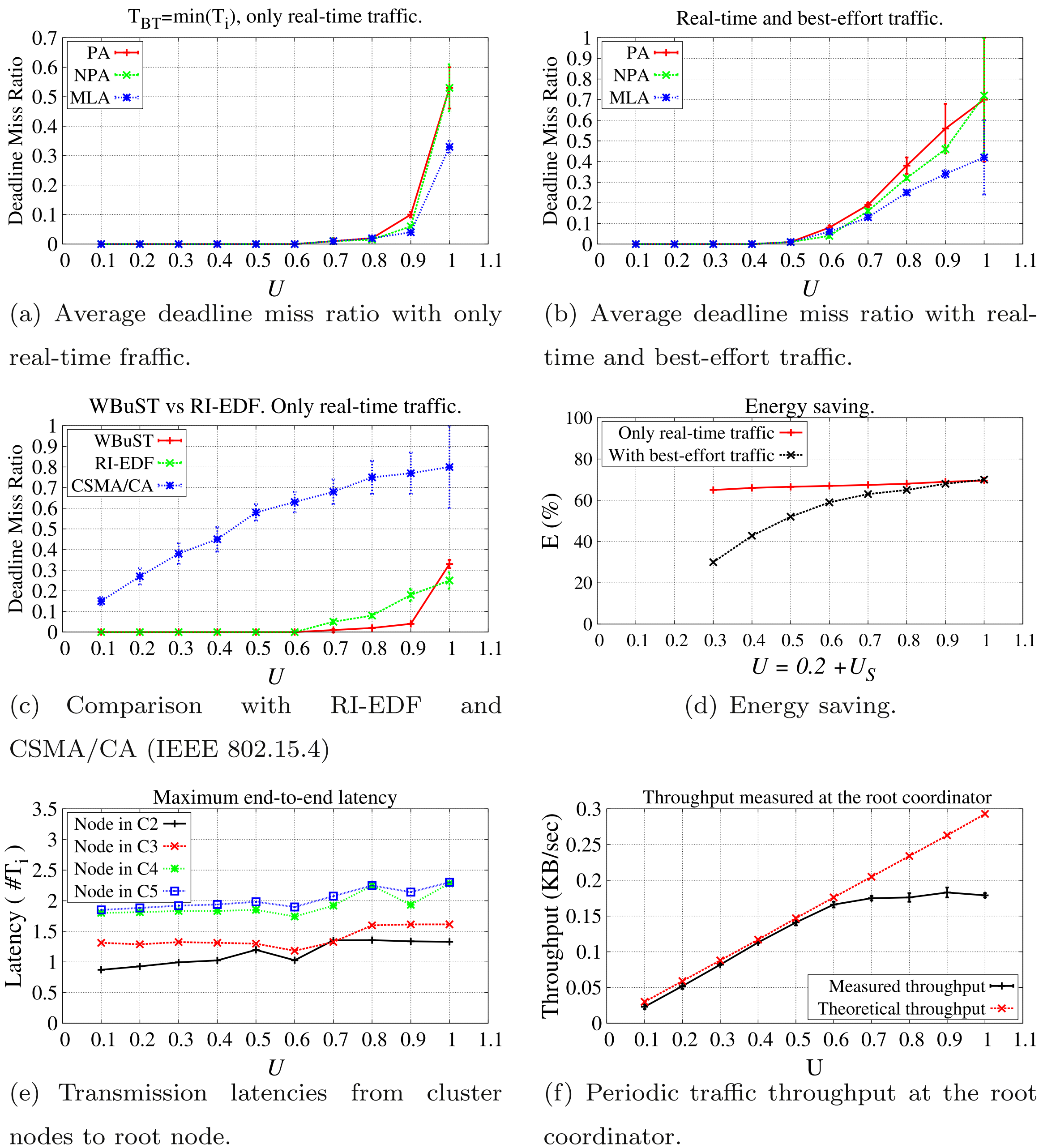
The effectiveness of the WBuST protocol has been evaluated by a set of experiments carried out on a network of 10 FLEX boards (FlexBoards, 2013) equipped with a 16-bit microcontroller and a IEEE 802.15.4 compliant transceiver. The firmware has been written in C under the ERIKA Enterprise real-time kernel (ErikaEnterprise, 2013).

All experiments refer to a cluster of 9 nodes plus the coordinator. Two message streams are assigned to each node, having a total amount of 18 real-time streams. The metric used to assess the protocol performance is the Average Deadline Miss Ratio (ADMS), defined as the average ratio of messages that miss their deadline and the total number of messages generated by all network streams. ADMS is measured by varying the total channel utilization U of real-time traffic, as defined by Eq. (1), from 0.1 to 1 with a step of 0.1. A total amount of 50 experiments have been carried out for each value of U. Furthermore, in each test, the deadline miss ratio was computed by monitoring the network for 10 min and generating the stream parameters as explained below. First, the stream utilizations were randomly generated within a uniform distribution. Then, for each value U_i, a relative deadline D_i was randomly generated with uniform distribution in the interval [300; 900] with resolution of 5 units. Message length M_i was computed by multiplying U_i by D_i. Note that in this evaluation all stream periods were considered equal to deadlines. Finally, node budgets were assigned by the allocation schemes of Table 2 and T_{BT} was set equal to $\min(D_i)$.

Since for these experiments, the number of node/streams is fixed during the tests, the control budget B_C is not necessary and so it was not allocated in the *CWs*. Similarly, since energy consumption is not taken into account, the sleep budget was not considered as well.

Fig. 10 shows the ADMS of budget allocation schemes when nodes transmit and receive only real-time traffic. The results show that, as long as U is not greater than 0.6, every message is delivered within its deadline, under all schemes. When $U \ge 0.7$, the number of messages missing their deadline starts increasing. However, for $U \le 0.9$ the ADMS is still less than 0.1, meaning that more than the 90% of the messages meet their deadline. In particular, the best performing scheme, MLA, presents an average deadline miss ratio not greater than 0.05. For U=1 the cluster channel becomes overloaded, because the available bandwidth is $1 - \alpha = 0.9$, and the ADMS is quite high. Note that, since nodes transmit only real-time traffic, the results obtained in this experiment are better than those predicted by the WCAU of each scheme, as shown in Table 2.

Fig. 10(b) reports the results of the second set of experiments, where nodes also transmit best-effort traffic. In particular, it is assumed that a node has an infinite amount of best-effort traffic to deliver, thus the entire budget is always used. In other words, for any value of U the channel is fully loaded, that is, the sum of U and the best-effort traffic utilization is always equal to $1 - \alpha$. The results reported in the figure show that, for all schemes, the ADMS is null as long as $U \le 0.5$ and increases for $U > 0.5$. The MLA scheme presents the lowest ADMS for all values of U. Using the formulas reported in Table 2, with $T_{BT} = \min(D_i)$ and $1 - \alpha = 0.9$, the WCAU results to be slightly lower than 0.5 for all schemes, hence, the experimental results are consistent with the theory.



(a) Average deadline miss ratio with only real-time fraffic.

(b) Average deadline miss ratio with real-time and best-effort traffic.

(c) Comparison with RI-EDF and CSMA/CA (IEEE 802.15.4)

(d) Energy saving.

(e) Transmission latencies from cluster nodes to root node.

(f) Periodic traffic throughput at the root coordinator.

Fig. 10. Experimental results.

In another experiment, WBuST has also been compared with the RI-EDF real-time protocol and a CSMA/CA approach, as defined by the un-slotted IEEE 802.15.4 standard. In this test, the node budgets are allocated through the MLAscheme and the nodes transmit only real-time traffic. The results are shown in Fig. 10(c). As reported in the graph, both WBuST and RI-EDF do not present any deadline miss as long as $U \leq 0.6$. For greater U both protocols show a non-null ADMS. In particular, WBuST performs better than RI-EDF for $0.7 \leq U < 1$. RI-EDF performs slightly better than WBuST only when the channel is overloaded (U=1). Notice that, although based on EDF, RI-EDF can only guarantee a channel bandwidth no larger than 0.6 for real-time streams. This is due to the protocol overhead, which is mainly given by the packet recovery mechanism. Finally, since the CSMA/CA approach is not designed to support real-time communications, its ADMS is always non-null and increases rapidly with U.

Figs. 10(a)–(c) also show the 95% confidence interval of the ADMS. However, note that for values of ADMS smaller than 0.3 the confidence interval is not appreciable in the plots.

Another set of experiments has been carried out to test the effectiveness of WBuST on saving energy through the sleep budget mechanism. The goal was to measure the energy E_S consumed by a node when the sleep budget is used, in comparison with the energy E consumed by the node without this mechanism. The results are reported in Fig. 10(d). The first experiment considers a scenario with only real-time traffic with utilization 0.2. Each node has only one stream and the utilization of the sleep stream, U_S, ranges from 0.1 to 0.8, so that the total stream utilization U ranges between 0.3 and 1. The results show that, with $U_S = 0.1$ (U=0.3), the percentage of energy saving, computed as $E\% = 100(1 - E_S/E)$, is already equal to 65%. However, increasing U_S beyond 0.1 does not produce a significant growth of $E\%$. The reason is that, since the real time traffic utilization is low, 0.2, the budgets are not fully exploited and the nodes are in sleep mode most of the time. Conversely, when nodes also have to deliver best-effort traffic, U_S has a bigger impact on the percentage of the saved energy. More specifically, as also shown in Fig. 10(d), for $U_S = 0.1$ (U=0.3) the percentage of saved energy is 30% and increases with U_S. For $U_S = 0.8$ (U=1), $E\%$ is equal to 70%.

A final set of experiments has been executed to analyze the performance of WBuST in a multi-hop scenario. In this case, the 10 nodes form a tree of 5 clusters, each composed by a coordinator and a cluster node. The network topology is that of Fig. 4, without clusters C_6 and C_7. A periodic stream is allocated to each node and the stream parameters are generated as in the previous experiments. The node budgets are assigned by the NPA scheme and the total channel bandwidth U is the same for all clusters. The nodes have also best-effort traffic to deliver. Two metrics have been used to evaluate the protocol. The first one considers the maximum latency, defined as the maximum time taken by a message from a cluster node to reach the coordinator of root cluster C_1. For each stream S_i, the latency is expressed as the number of stream periods T_i taken by every message to reach the root. In this way, latencies are normalized and can be compared with each other in a consistent manner. The second metric considers the average amount of periodic traffic received by the coordinator of the root cluster. Both metrics have been evaluated by varying the total channel bandwidth U, which is also the total bandwidth of the root cluster. For each value of U, the transmission latency and the amount of periodic traffic have been averaged over a total amount of 20 experiments.

Fig. 10(e) reports the results on latency measurement for the cluster nodes located in all clusters, except root C_1. As it can be noted, the latency is constant or slightly increasing with the bandwidth and, even when the cluster channel is fully loaded, a message sent to the root takes an amount of time bounded to the number of hops between the sending node and the root coordinator.

Fig. 10(f) shows the average amount with the 95% confidence interval of real-time traffic throughput measured at the root coordinator in comparison with the theoretical amount. This last value is computed by multiplying U by the transceiver transmission rate (250 kbps). As it can be seen, the throughput increases linearly till $U \leq 0.6$ and then drastically reduces the rising speed. The reason is that, as previously seen in Fig. 10(b), for $U \geq 0.6$ the number of messages that miss their deadline increases proportionally to U. Note that, as long as $U \leq 0.6$, the difference between the throughput and the theoretical amount is mainly due to the overhead introduced by the physical level of the radio transceiver.

8. Conclusions and future work

This paper presented WBuST, a MAC layer protocol for time sensitive communication in wireless embedded systems. WBuST supports both real-time and best-effort traffic in multi-hop networks, grouping the network devices into clusters managed by a coordinator node. A different radio channel is assigned to each cluster, where the nodes are synchronized by the coordinator through the transmission of a periodic beacon. The channel access is regulated by a budget reservation mechanism that guarantees a predictable transmission time, both for intra-cluster and inter-cluster communications. An energy saving mechanism is also provided to reduce energy consumption and guarantee a desired network lifetime.

The protocol has been formally analyzed and a guarantee test has been provided to verify whether a given amount of real-time traffic, described by a stream set, can be delivered by the protocol within predefined timing constraints. A method, base on Network Calculus, has been also presented to compute both bandwidth and buffering requirements needed by cluster routers to guarantee bounded end-to-end delays and avoiding buffer overflows.

The experimental evaluation showed the ability of WBuST in managing real-time traffic while reducing the energy consumption, as well as the consistency between theoretical and experimental results. A comparison with the RI-EDF protocol showed that WBuST outperform this last in terms of deadline miss ratio.

Considered the positive results achieved in the experiments, further work is planned to extend the worst case end-to-end delay analysis to the case of downstream flows. A further research direction consists of investigating adaptive energy management mechanisms to deal with streams with variable parameters.

References

Abdelzaher, T.F., Prabh, K.S., Kiran, R, 2004. On real-time capacity limits of multihop wireless sensor networks. In: Proceedings of the 25th IEEE International Real-Time Systems Symposium (RTSS 2004).

Bazan, O., Jaseemuddin, M., 2012. A survey on mac protocols for wireless adhoc networks with beamforming antennas. IEEE Commun. Surv. Tutor. 14 (2), 216–239.

Bui, B.D., Pellizzoni, R., Caccamo, M., Cheah, C.F., Tzakis, A., 2007. Soft real-time chains for multi-hop wireless ad-hoc networks. In: Proceedings of the 13th IEEE Real Time and Embedded Technology and Applications Symposium (RTAS'07).

Caccamo, M., Zhang, L.Y., Sha, L., Buttazzo, G., 2002. An implicit prioritized access protocol for wireless sensor networks. In: Proceedings of the IEEE Real-Time Systems Symposium (RTSS 2002).

Chan, E., Chen, D., Cao, J., Lee, C.-H., 1997. Time properties of the FFDI-M medium access protocol. Comput. J. 82 (1), 96–102.

Chan, E., Chen, D., Lee, V.C., 2001. Effectiveness of the FDDI-M protocol in supporting synchronous traffic. J. Syst. Softw. 56 (1), 51–62.

Chen, D., Chan, E., Lee, C.-H., 1998. Timing properties of the FDDI-M medium access protocol for a class of synchronous bandwidth allocation schemes. In: Proceedings of Seventh International Conference on Computer Communications and Networks (ICCCN 1998).

Crenshaw, T.L. Tirumala, A., Hoke, S., Caccamo, M., 2005. A robust implicit access protocol for real-time wireless collaboration. In: Proceedings of the 17th IEEE Euromicro Conference on Real-Time Systems (ECRTS 2005).

Demirkol, I., Ersoy, C., Alagoz, F., et al., 2006. Mac protocols for wireless sensor networks: a survey. IEEE Commun. Mag. 44 (4), 115–121.

ErikaEnterprise, 2013. Rtos Erika Enterprise. URL ⟨http://erika.tuxfamily.org/drupal⟩.

FlexBoards, 2013. Flex Boards. URL ⟨http://www.erika.tuxfamily.org/drupal/flex-board.html⟩.

Franchino, G., Buttazzo, G.C., 2012. WBuST: a real-time energy-aware MAC layer protocol for wireless embedded systems. In: Proceedings of the 17th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2012).

Franchino, G., Buttazzo, G.C., Facchinetti, T., 2007. BuST: budget sharing protocol for hard real-time communication. In: Proceedings of the 12th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2007).

Franchino, G., Buttazzo, G.C., Facchinetti, T., 2008. Properties of BuST and timed token protocols in managing hard real-time traffic. In: Proceedings of the 13th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2008).

Franchino, G., Buttazzo, G.C., Facchinetti, T., 2010a. Token Passing Techniques for Hard Real-Time Communication. Rijeka, Croatia: IN-TECH, March 2010.

Franchino, G., Buttazzo, G.C., Marinoni, M., 2010b. An energy-aware algorithm for TDMA MAC protocols in real-time wireless networks. In: Proceedings of the 5th IEEE Symposium on Industrial Embedded Systems (SIES2010).

Huang, P., Xiao, L., Soltani, S., Mutka, M.W., Xi, N., 2013. The evolution of mac protocols in wireless sensor networks: a survey. IEEE Commun. Surv. Tutor. 15 (1), 101–120.

IEEE 802.15.4 Std-2011, 2011. Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs). Technical Report, IEEE-SA Standards Board (June).

Koubaa, A., Song, Y.-Q., 2004. Evaluation and improvement of response time bounds for real-time applications under non-pre-emptive fixed priority scheduling. Int. J. Prod. Res. 42 (14), 2899–2913.

Koubaa, A. Alves, M., Tovar, E., 2006. Modeling and worst-case dimensioning of cluster-tree wireless sensor networks. In: Proceedings of the 27th IEEE Real-Time Systems Symposium (RTSS 2006).

Koubaa, A., Alves, M., Tovar, E., 2007. Energy/delay trade-off of the GTS allocation mechanism in IEEE 802.15.4 for wireless sensor network. Int. J. Commun. Syst. 20 (7), 791–808.

Kumarawadu, P., Dechene, D., Luccini, M., Sauer, A., 2008. Algorithms for node clustering in wireless sensor networks: a survey. In: Proceedings of the 4th International Conference on Information and Automation for Sustainability (ICIAFS 2008).

Leboudec, J.-Y., Thiran, P., 2001. Network Calculus. Springer Verlag, New York.

Liu, C.L., Layland, J.W., 1973. Scheduling algorithms for multiprogramming in hard real-time environment. J. ACM 20 (1), 40–61.

Rappaport, T.S., 1996. Principles and Practice, Wireless Communication. Prentice-Hall, New Jersey.

Rashid, B., Rehmani, M.H. Applications of wireless sensor networks for urban areas: a survey. J. Netw. Comput. Appl.

Rashid, B., Rehmani, M.H., Ahmad, A. Broadcasting strategies for cognitive radio networks: taxonomy, issues, and open challenges. Comput. Electr. Eng.

Rehmani, M.H., 2010. Channel assortment strategy for reliable communication in multi-hop cognitive radio networks. In: 2010 IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM). Montreal, QC Canada: IEEE, pp. 1–2.

Rowe, A., Mangharam, R., Rajkumar, R., 2006. RT-Link: a time-synchronized link protocol for energy-constrained multi-hop wireless networks. In: Third IEEE International Conference on Sensors, Mesh and Ad Hoc Communications and Networks (IEEE SECON06).

Saifullah, A., Xu, Y., Lu, C., Chen, Y., 2010. Real-time scheduling for WirelessHART networks. In: Proceedings of the 31st IEEE Real-Time Systems Symposium (RTSS 2010).

Saifullah, A., Xu, Y., Lu, C., Chen, Y., 2011. End-to-end delay analysis for fixed priority scheduling in wirelessHART networks. In: IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS11).

Shashi Prabh, T.F.A.K., 2007. On Scheduling and real-time capacity of hexagonal wireless sensor networks. In: Proceedings of the 19th IEEE Euromicro Conference on Real-Time Systems (ECRTS 2007).

Sobral, M.M., Becker, L.B., 2008. A wireless hybrid contention/tdma based mac for real-time mobile application. In: Proceedings of the ACM Symposium on Applied Computing (SAC 2008), Fortaleza, Brazil.

Sobrinho, J.L., Krishnakumar, A.S., 1999. Quality-of-service in ad hoc carrier sense multiple access wireless networks. IEEE J. Sel. Areas Commun. 17 (8), 1353–1368.

The IEEE, 1999. Wireless LAN Medium access Control and Physical Layer (PHY) Specification, Technical Report, ANSI/IEEE Std. 802.11 ⟨http://standards.ieee.org⟩.

Tobagi, F.A., Kleinrock, L., 1975. Packet switching in radio channels: Part II – the hidden terminal problem in carrier sense multiple-access modes and the busy-tone solution. IEEE Trans. Commun. 23 (12), 1417–1433.

Toscano, E., Lo Bello, L., 2012. Multiplechannel superframe scheduling for IEEE 802.15.4 industrial wireless sensor networks. IEEE Trans. Ind. Inform. 8 (2).

Vilzmann, R., Bettstetter, C., 2005. A survey on mac protocols for ad hoc networks with directional antennas. In: EUNICE 2005: Networks and Applications Towards a Ubiquitously Connected World. Madrid, Spain: Springer, pp. 187–200.

WirelessHART, 2013. WirelessHART Specification. URL ⟨http://www.hartcomm.org⟩.

Ye, W., Heidemann, J., Estrin, D., 2004. Medium access control with coordinated, adaptive sleeping for wireless sensor networks. ACM/IEEE Trans. Netw. 12 (3), 493–506.

Zhang, S., Burns, A., Chen, J., Lee, E., 2004. Hard real-time communication with the timed token protocol: current state and challenging problems. Real-Time Syst. 27 (3), 271–295.