Published in IET Wireless Sensor Systems
Received on 5th June 2013
Revised on 12th February 2014
Accepted on 13th February 2014
doi: 10.1049/iet-wss.2013.0065

ISSN 2043-6386

Efficient scheduling of sporadic tasks for real-time wireless sensor networks

Mukhtar Azeem[1], Majid Iqbal Khan[1], Samee Ullah Khan[2], Wilfried Gansterer[3]
[1]Department of Computer Science, COMSATS Institute of Information Technology, Islamabad 22000, Pakistan
[2]Department of Electrical and Computer Engineering, North Dakota State University, Fargo, ND 58104, USA
[3]Department of Computer Science, University of Vienna, 1090 Vienna, Austria
E-mail: samee.khan@ndsu.edu

Abstract: Industrial automation requires hard real-time delivery of data that can be of periodic or sporadic in nature. It is challenging to ensure hard real-time delivery of periodic and sporadic data to a multi-hop away destination in a bandwidth constrained environment, such as wireless sensor networks. In this regard, research has been done for joint scheduling of periodic and sporadic data delivery using the IEEE 802.15.4 standard. However, delivery to a destination that is multiple hops away has not been handled. Moreover, the IEEE 802.15.4 standard compliance is not fully taken care of. This study proposes a novel communication protocol that is IEEE 802.15.4 standard compliant, for real-time delivery of sporadic and periodic data to a destination that is multiple hops away. Implicit decisions at various nodes result in the minimal increase in the control traffic and reduced energy dissipation. The proposed protocol is tested using the OPNET simulator and the correctness is proven through a multitude of simulation results.

1 Introduction

There are many applications of wireless sensors in real-time critical environments. These environments include industrial, entertainment, military and medicine to name a few. Wireless sensors can be usefully applied to these time critical environments for sensing, processing and propagation of information. Real-time propagation of data is of the utmost importance in critical industrial environments [1]. Infrastructure-based wireless sensor networks (WSNs) can be deployed in situations where sensors are statically deployed at predetermined positions. This topology is suitable for industrial WSNs. Sensory information in the industries is generated by 'two' types of 'events' [2, 3]: (i) time triggered or periodic events, and (ii) sporadic events. Communication 'tasks' are used for the propagation of sensory information. 'Periodic tasks' are used to propagate sensory information because of 'periodic events'. These 'periodic events' are predictable. Therefore, the periods and the execution times of the 'periodic tasks' can be estimated and schedules for such tasks can be generated offline. 'Sporadic events' are random and unpredictable. These events are usually very important and require urgent attention. It is challenging to handle 'sporadic events' because neither the time of occurrence, nor the reoccurrence of such events is known. Moreover, scheduling sporadic tasks along with 'periodic tasks' requires online modification of the schedule that is computationally infeasible. In the industrial setup, the destination of the sporadic information is usually a fully functional device (FFD) that is also the coordinator of the network. The network is deployed in such a way that the coordinator is at the root of the network topology. The coordinator is the node where the critical decisions based on the 'sporadic events' can be made. Therefore, all of the nodes in the network must send sporadic information that they sense themselves and/or receive from their child nodes to the root of the tree. Totally random and unpredictable events are extremely difficult to handle optimally. However, as defined by Isovic and Fohler [4], sporadic tasks are 'the tasks that can arrive at the system at arbitrary points in time, but with defined minimum inter-arrival times between 'two' consecutive invocations. A sporadic task T_s is characterised by its relative deadline, minimum inter-arrival time and worst-case execution time'. With the help of these parameters, the problem of finding a schedule of 'sporadic tasks' along with 'periodic tasks' becomes possible. To the best of our knowledge, no research has addressed the scheduling of sporadic tasks within an existing periodic schedule in WSNs using the IEEE 802.15.4 standard for end-to-end sporadic packet delivery. Allocating resources for periodic events proactively is useful because such events are to occur regularly. However, such a strategy is wasteful for sporadic events because sporadic events are not regular.

The major contributions of this paper are:

1. An offline resource reservation strategy for scheduling the sporadic tasks using IEEE 802.15.4 standard, such that these resources, do not go wasted and can be used for other purposes in the absence of the sporadic events.
2. A reactive acceptability test for the sporadic events that is used to ensure that the sporadic events are accepted only if

Table 1 Table of symbols

Symbol	Definition
CH	cluster head
GTS	guaranteed time slot
SC	spare capacity
BI	beacon interval
CAP	contention access period
CFP	contention free period
FFD	fully functional device
SD	superframe duration
BI	beacon interval
SO	superframe order
BO	beacon order
d_{e2e}	end-to-end deadline of the sporadic task
$delay_{e2e}$	calculated end-to-end delay for sporadic tasks
PAN	personal area network
t_e	time of occurrence of sporadic event
CSMA/CA	carrier sense multiple access/collision avoidance

these sporadic events meet the criteria set aside to guarantee deadlines.
3. An online guaranteed time slot (GTS) (de)allocation protocol that is used to allocate required resource once a sporadic event is accepted after passing the acceptability test. The event is rejected if it fails the acceptability test.

Table 1 provides the list of abbreviations and their definitions used throughout this paper.

The paper is organised as follows. Section 2 gives an overview of relevant sections of the IEEE 802.15.4 standard. Section 3 describes the proposed work. The IEEE 802.15.4 standard compliance is discussed in Section 4. Section 5 explores the factors affecting the energy dissipation. Simulation studies are summarised in Section 6. Section 7 contains related work and Section 8 concludes the paper.

2 IEEE 802.15.4 wireless personal area network for time-sensitive applications

IEEE 802.15.4/Zigbee is the de facto standard for WSNs, which provides a mechanism for accessing the wireless channel for communication. It is of particular interest for the real-time community because it provides a mode that supports time-sensitive applications for real-time communication. In the synchronous mode of IEEE 802.15.4 standard, nodes use special periodic packets, known as beacons, for synchronisation among themselves. The time period between 'two' beacons is called beacon interval (BI). The BI can be divided into an active part and an inactive part. All of the communication among the nodes is done during the active part (also called *SuperFrame*), whereas nodes go into sleep mode during the inactive part, to save energy. The active part of the BI is further divided into 'two' subperiods: (i) contention access period (CAP) that is for applications requiring best effort services and (ii) contention free period (CFP) that is for time-sensitive applications.

The CFP is composed of the GTS, allocated to individual nodes of the cluster, prior to their use. Owing to being pre-allocated, it can support time-sensitive applications and time guarantees are possible. This coordinated mode is available in the IEEE 802.15.4 standard in a star topology where the cluster head (CH) is the coordinator of cluster. It is the originator of beacons and all of the communications among nodes is through the CH.

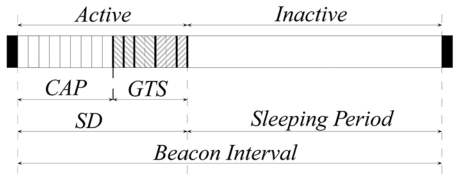

Fig. 1 *Subdivision of BI*

The frequency of beacon frames depends on the value of the beacon order (BO). The BI and BO are related according the (1).

$$\text{BI} = aBaseSuperframeDuration \times 2^{\text{BO}} \quad (1)$$

With BO equal to zero, the beacon frames are transmitted every *aBaseSuperframeDuration* time units. It is also clear that as the BO increases, the size of the BI doubles. Fig. 1 depicts the various parts of the BI.

During the CAP, nodes contend for the channel using carrier sense multiple access/collision avoidance (CSMA/CA) before they can communicate. This period supports the best effort service. The CFP is for time-sensitive applications where nodes acquire a GTS before they can communicate. The duration of the active period [superframe duration (SD)], within a BI, is determined by the superframe order (SO). The SD and SO are related according to (2).

$$\text{SD} = aBaseSuperframeDuration \times 2^{\text{SO}} \quad (2)$$

where $0 \leq \text{SO} \leq \text{BO} \leq 14$ holds. It is also clear that as the SO increases, the size of the SD doubles. It is noteworthy to mention that the BI and the SD are not interdependent except through the *aBaseSperframeDuration*. Therefore, a change in the BO does not affect the SD and modification in the SO does not affect the BI. The SD is divided into 16 equal sized slots. The size of 'one' slot is denoted as *slotSize*.

The slot allocation within the CFP is performed by the CH at the request of a node that intends to use it. At most seven GTS allocations are possible (including the transmit and receive GTSs) and all of these allocations can be in multiples of the basic slot sizes (*slotSize*). Once allocated to a particular node, only the node that requested the GTS can communicate with the CH during that slot. Each node can have 'two' GTSs at the most, one for transmission (from node to the CH) and the other for reception (from CH to the node). If a node needs to communicate via the GTS, the slot must first be allocated by the CH in response to the 'GTS request' from the requesting node. The GTS is allocated from the CAP. However, according to the IEEE 802.15.4 standard, when allocating GTS, CAP should not become zero. A CAP of sufficient size, such that at least 'one' 'GTS request' can be sent to the CH, must be maintained.

The IEEE 802.15.4 standard supports real-time communication between 'two' peers, through the GTS allocation mechanism in the star topology. Koubaa *et al.* [5] have extended this basic provision through the use of overlapped clusters to provide end-to-end guaranteed communication between a source and a destination that are more than 'one' hop apart. The beacon periods of the overlapping clusters are scheduled in such a way that there is no overlapping of active periods among neighbouring clusters having a shared collision domain. In the resultant

IET Wirel. Sens. Syst., pp. 1–10
doi: 10.1049/iet-wss.2013.0065

schedule, the GTSs are scheduled such that the deadlines of the periodic flows are met. All of the aforementioned is done through the offline scheduling tools. Once the size of the BO, SO and schedule of the GTSs of various clusters is determined, the network nodes allocate GTSs accordingly. Consequently, a guaranteed end-to-end communication is possible.

The determination of the SO and the BO depends on the number and parameters (release time, period, deadline and size) of the periodic events. Once the SO is determined and the GTS parameters are calculated offline, the IEEE 802.15.4 standard 'GTS request' primitive is used by the requesting child nodes to receive the GTSs allocated in their parent cluster. After this allocation is done, the remaining part of the superframe that is at least *aMinCAPLength* is used as CAP for the best effort services.

3 Scheduling for sporadic and periodic events

Generating a schedule of the periodic events is easy because the release time of every event is known. Similarly, their periods, deadlines and execution times are also known. With this information in hand, offline schedules (also regarded as static schedules) can be generated using any of the well-known scheduling techniques. Since deterministic scheduling is possible offline, no scheduling decision is left until runtime. Offline scheduling of sporadic events is impossible because their release times and periods are unknown. Unless an event is actually released, we cannot schedule it. However, if the complete load of scheduling is left until release time of sporadic event, then the scheduling itself becomes computationally complex (because of complicated scheduling decisions).

By combining the offline and online scheduling techniques, sporadic events can be handled. The idea is to generate an offline GTS schedule of periodic events (for IEEE 802.15.4 standard), whereby extra space for the sporadic events is kept in the CAP for sporadic events proactively. In the absence of any sporadic events, the extra space can be used for best effort services. When sporadic event occurs, the GTSs can be reactively allocated from this extra space of the CAP.

3.1 Offline scheduling for periodic and sporadic tasks

Scheduling of the completely random sporadic events is impossible because there is no associated release time and period. However, as suggested by Isovic and Fohler [4, 6], periodic reservation tasks with parameters equal to worst-case parameters of sporadic tasks, can be used to represent the sporadic tasks and can be used for scheduling. Space requirement of these proxy tasks can be calculated. In the context of the IEEE 802.15.4 standard, the space is kept in CAP. While calculating CAP of any cluster, IEEE 802.15.4 standard imposes the condition that $CAPlength \geq aMinCAPLength$. To handle sporadic events, there will be enough spare capacity (SC) in the CAP for guaranteed, reactive/instantaneous allocation of desired GTS slots. The condition for CAP calculation is therefore modified to (3).

$$NewCAPlength \geq aMinCAPLength + \lambda_j \quad (3)$$

Where λ_j is the SC requirement of cluster j calculated on the basis of all of the sporadic events that are expected to occur on any of the downstream nodes of the node which is the head of cluster j. Fig. 2*a* represents the CAP inflation. In Fig. 2*a*, 'New CAP Length' meets the condition of (3). Consequently, the online 'GTS request' for sporadic data will never fail.

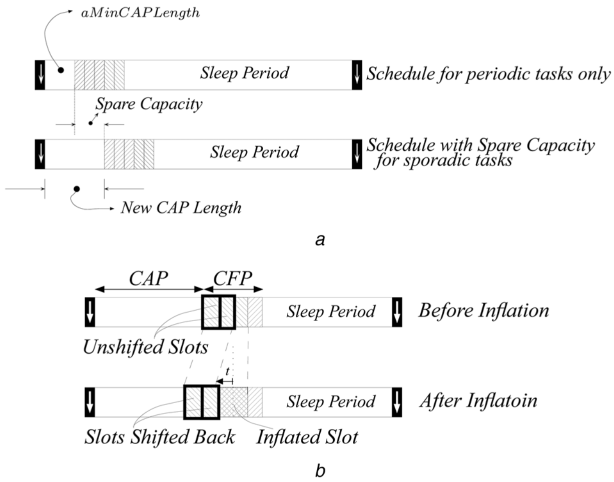

Fig. 2 *CAP adjustment*

a Owing to offline rescheduling and
b Owing to online inflation

During offline scheduling, the required size of the CAP for a particular cluster is therefore variable depending on the SC requirement of the cluster. The SC calculation results in the SC requirement matrix ***S***. In matrix ***S***, row headings represent all nodes within the network and column headings represent routers. The value of the element $S_{i,j}$ of the matrix is the slot requirement of router j because of the traffic from node i. To get the total bandwidth requirement of a particular router, all of the elements of a column under that router number are added together.

The selection of a node as the source of sporadic events adds to the space requirement of all the nodes of the branch from itself to the root node (the sink). Therefore, (4) gives us the total SC requirement in the number of time slots, $S_{i,j}$, of any cluster R_j (column headers of matrix) because of a sporadic event which is expected to occur at node i (row headers of matrix).

$$\lambda_j = \left\lceil \left(\sum_{i=0}^{\text{no of routers}} S_{i,j}/slotSize \right) \right\rceil \times slotSize \quad (4)$$

where $slotSize = \text{SD}/16$. Therefore, using the sporadic events as shown in Fig. 3, the value of the matrix $S_{i,j}$ is as shown in the matrix given in Table 2. Summing up the columns of the matrix ***S*** and taking *slotSize* equal to 'one', gives us the value of λ_j for a particular router, as given in (5).

$$\lambda_j = \begin{pmatrix} 4 & 1 & 0 & 1 & 0 & 0 \end{pmatrix} \quad (5)$$

While calculating the BO for the network and the SO for the individual clusters, the variable *CAPlength* values based on (3) are used. The result is a multicluster schedule with each of the cluster having adequate capacity in the CAPs to accommodate sporadic tasks. The generated schedule is such that if each of the nodes goes on transmitting the sporadic data to its parent, then it will reach the destination, the root, at or before the deadline.

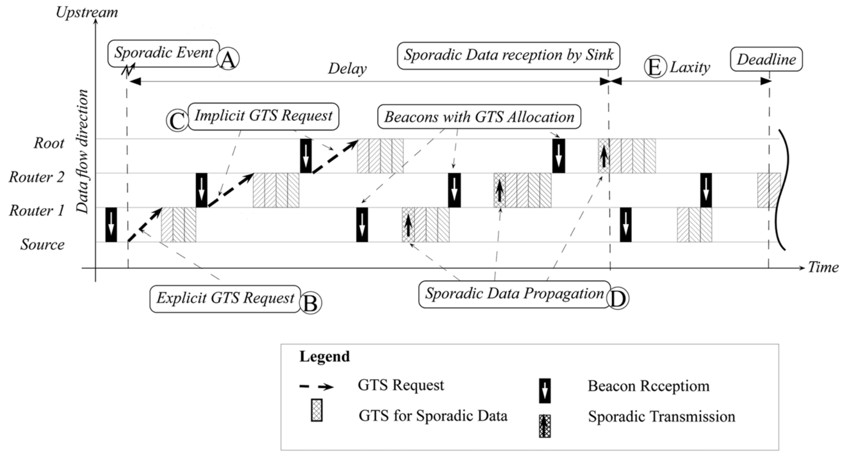

Fig. 3 *GTS request and sporadic data propagation*

3.1.1 Offline laxity calculation: The deadline (d_{e2e}) is the value that is provided to the offline scheduler along with the sporadic task parameters. This is the time before which the event data must reach the root node (R1). If the set of tasks (periodic and sporadic) is feasible, then the scheduler succeeds in generating a schedule. When the schedule is followed by the nodes the relevant data can reach the destination node on or before the deadline. The delay_{e2e} denotes effective time at which the event data will actually arrive at the destination, calculated as a result of the offline scheduling. Therefore, for a feasible schedule, the difference between d_{e2e} and delay_{e2e}, as given in (6), must be greater than or equal to zero. The value of this difference is a measure of how much can the schedule of a sporadic event be delayed still meeting the deadline and is termed as the 'laxity' of the schedule.

$$\text{Laxity} = d_{e\,2\,e} - \text{delay}_{e\,2\,e} \quad (6)$$

The value of the 'laxity' is provided to the online scheduler along with other parameters. When the sporadic event occurs, the value of 'laxity' is used (by online scheduler) to perform acceptability test to determine whether the event should be accepted or not, as can be seen in Section 3.2.2.

3.2 Online GTS scheduling for sporadic tasks using CAP

The offline schedule, generated proactively, guarantees the deadlines for periodic events. It will also guarantee the deadline for sporadic events if the sporadic events do not violate the parameters, such as maximum frequency which was provided while generating the offline schedule. For periodic schedule, the GTSs are allocated before the start time of the periodic flows and remain allocated until the end time of the flows. For sporadic events, GTSs are required only instantaneously. Therefore, we allocate GTSs only for a single BI. The CAP is sufficiently large to accommodate as many sporadic tasks as were scheduled during the offline schedule generation. The benefit of keeping it in the CAP is the improvement of the response times of the best effort service of the network in the absence of sporadic events.

To ensure that the implicit guarantee offered by offline scheduler remains intact, a synchronised allocation of GTSs throughout the intervening cluster is needed. This requires a GTS allocation protocol that is invoked after the sporadic event is accepted. Therefore, a novel protocol is proposed as the major contribution of this paper as reported in Section 3.2.1.

Table 2 Sporadic requirements matrix, ***S***

	R1	R2	R3	R4	R5	R6
R1	0	0	0	0	0	0
R2	0	0	0	0	0	0
R3	1	0	0	0	0	0
R4	1	0	0	0	0	0
R5	0	0	0	0	0	0
R6	0	0	0	0	0	0
N7	0	0	0	0	0	0
N8	0	0	0	0	0	0
N9	1	1	0	0	0	0
N10	0	0	0	0	0	0
N11	0	0	0	0	0	0
N12	1	0	0	1	0	0
N13	0	0	0	0	0	0
N14	0	0	0	0	0	0

3.2.1 Dynamic inflation implicit deflation protocol: Consider the tree topologies as shown in Fig. 3. This is a personal area network (PAN) of overlapping clusters. Every node acts as a child as well as a CH. The leaf nodes act as children only and the root acts as a CH only. The root of the tree that synchronises the overall activities of the network, is called the PAN coordinator (PC). When a sporadic event occurs at a node, the data associated must reach PC (also the sink for sporadic data). According to this topology, if every node of all of the intervening clusters goes on delivering data to its parent CH, the data will ultimately reach PC, the root node. However, according to the IEEE 802.15.4 standard, prior to sending data to its parent, a node has to allocate a slot in the CFP by sending 'GTS request' to its CH. The CH checks to see the following conditions:

1. Sufficient space is available in the CAP slot to allocate the desired GTS.
2. Requesting node is not already allocated to a GTS.
3. The total number of slots allocated in the cluster is less than seven.

If CH finds that any of the conditions cannot be met, then it denies the 'GTS request'. In case, if the SC has been allocated according to (3), then Condition 1 will always be met because the CAP of adequate size was allocated to ensure that the GTS

doi: 10.1049/iet-wss.2013.0065

Algorithm 1

```
if (CAPlength ≥ aMinCAPLength + n) then
    if no GTS is already allocated to requesting node in direction 'd' then
        Allocate GTS
    else
        Inflate GTS as per requirement
    end if
else
    Insufficient Resources
end if
```

Fig. 4 *Allocate 'n' GTS slots in 'd' direction*

allocation is success. For the case when the standard denies the allocation because of the non-fulfillment of Condition 2, we propose a GTS inflation protocol. The proposal helps to handle the sporadic events that require instantaneous space in the GTS slot to carry the sporadic data. This proposed protocol is defined in Algorithm 1 (see Fig. 4).

Accordingly, if a CH receives a 'GTS request' from a child node and the requesting node is already allocated a slot in the BI (Condition 2 is not met), then instead of denying the request, the existing GTS is inflated according to the space requested. The new adjusted schedule is sent in the forthcoming beacon frame. This fresh GTS allocation takes space from the SC of the CAP ensured during offline scheduling. If the sporadic traffic complies with the specifications that were provided to the offline scheduler, then it is an implicit guarantee that the CH will always find the required space in the CAP slot and Condition 1 will always be met.

In case of inflation, if there are any slots prior to the slot being inflated, then such slots will be shifted back in time as can be seen in Fig. 2*b* taking space from the spare capacity of the CAP, so that the sleep period length is not affected. Therefore, the transmission that was taking place during these shifted slots will complete earlier in time after inflation.

For the end-to-end sporadic information propagation, the protocol needs to handle the traversal of the overlapping clusters. Since the PC is considered as the sink of sporadic data; therefore, when an intermediate CH receives a 'GTS request' for a sporadic GTS, it allocates a GTS to the requesting node and in addition it enques a 'GTS request', implicitly, for its parent node (CH_p) (also the CH of another cluster). This happens at a time when CH_p is sleeping. When CH_p wakes up, it immediately receives the enqued 'GTS request'.

Fig. 5 shows the occurrence of a sporadic event at tag Ⓐ. An explicit 'GTS request' is generated at tag Ⓑ. Every receiver of the 'GTS request' sends an implicit 'GTS request' to the parent, as shown at tag ©. This happens until the receiver of the 'GTS request' is the root (the sink of sporadic data) that stops the implicit 'GTS request' generation. In this way, every intermediate node that sends the 'GTS request', gets a GTS in the forthcoming beacon for the transmission of sporadic data. Every sporadic 'GTS request' is guaranteed to get the required GTS because enough resources (in the form of SC) were allocated when the offline schedule was generated. As a result of this guaranteed GTS allocation, nodes can send sporadic data during, the desired, allocated slots. This upstream data propagation ends at PC as is shown via tag ©. We must recall that the PC is the sink of the sporadic data.

Reactive inflation and implicit deflation of GTS: If a CH receives a GTS request from a node that already owns a GTS (slot with double hatching can be seen in Fig. 6), then the CH will 'inflate' that slot as described earlier. If a request is received from a node that does not already own a GTS, then the CH will 'allocate' a new GTS for it. This sporadic allocation/inflation is done for a single BI only. It

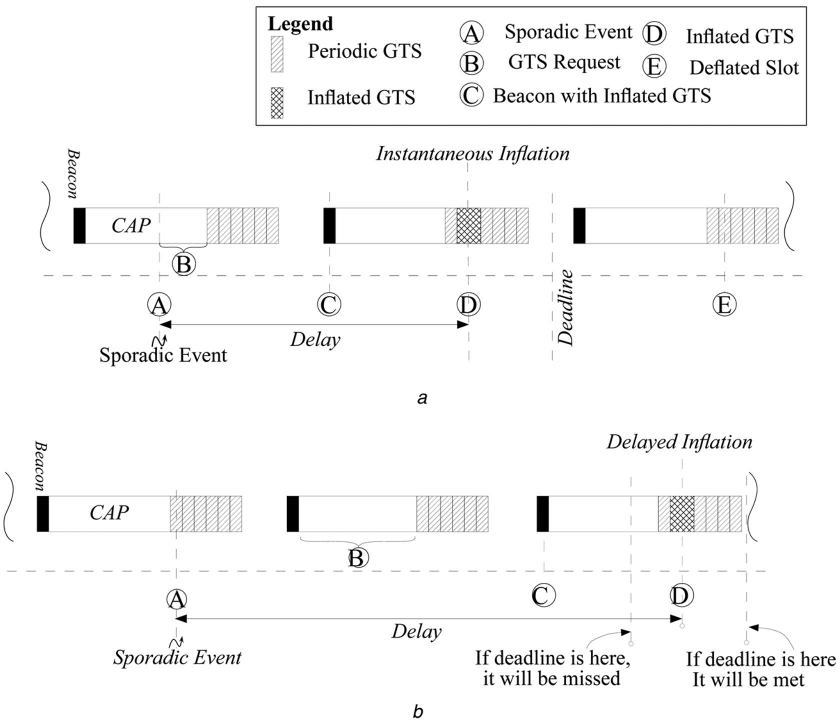

Fig. 5 *Effect of event release time on GTS inflation*

a Allocation received in the forthcoming beacon and
b Allocation received after one beacon

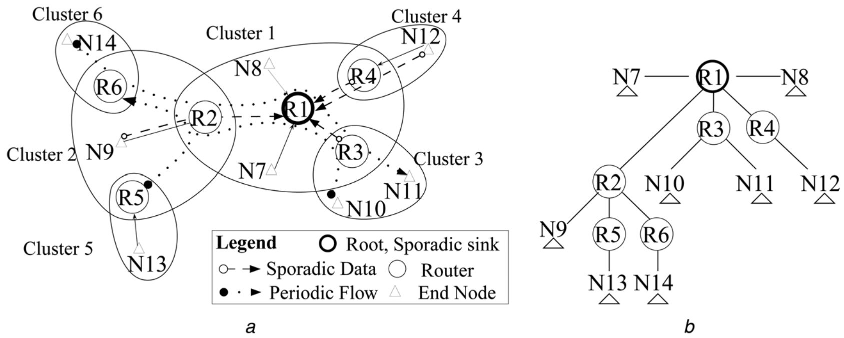

Fig. 6 *Overlapping clusters*

a Clusters with periodic and sporadic paths and
b Tree topology

is implicitly deallocated/deflated thereafter. This necessitates that the sporadic data, for which this request was generated, will be forwarded to the CH using this allocated/inflated GTS in the forthcoming BI. According to our proposal, for the deallocation/deflation of this slot, the CH does not wait for the deallocation request from the child node and the CH deallocates/deflates the inflated slot implicitly. The child node does not need to send the deallocation request and it implicitly considers the GTS deallocated/deflated in the next beacon. Owing to these implicit decisions, the control traffic is kept limited only to the 'GTS request'. Figs. 6*a* and *b* show this elastic allocation where at tag Ⓐ events occur. At tag Ⓑ, a *Request* is sent to the parent. The inflation/allocation response is received through a beacon frame at tag © and finally, sporadic data are transmitted through allocated/inflated slot at tag Ⓓ. Tag Ⓔ represents the implicitly deallocated/deflated slot. Whenever another sporadic event occurs, this reactive allocation/implicit deallocation mechanism is repeated.

3.2.2 Online acceptance test: When a sporadic event occurs, it is to be determined whether it must be accepted or not. From [7], we already know that the command frames (e.g. GTS requests) are transmitted between the 'two' nodes during the CAP period. Therefore, if the event occurs during the CAP period, then an associated 'GTS request' will be sent to the CH during the CAP of the ongoing BI see (tag Ⓐ in Fig. 5*a*). However, if the event occurs at a time when the CAP of the period has already passed (that means the event occurred at any time other than the CAP), then the relevant 'GTS request' can only be sent in the CAP of the forthcoming BI (see tag Ⓑ Fig. 5*b*). The offline calculation of the schedule parameters is such that if a 'GTS request' is sent in the same BI in which the sporadic event occurs, then the sporadic task is guaranteed to meet the deadline. In the other case (when the 'GTS request' can only be sent during the CAP of forthcoming beacon period), 'laxity' as defined in (6), is used to determine whether the deadline will still be met or otherwise. In light of the aforementioned, to make that decision, we have the following information in hand: (i) the current state (CAP, GTS or sleep period) of the superframe of IEEE 802.15.4 standard and (ii) the 'laxity' of the relevant cluster. With the help of this information we define an acceptance test as follows

$$x = \begin{cases} 0, \text{if} (StartOfCurrentGTS - t_e) \geq l_{\text{GTSRequest}} \text{OR laxity} >= \text{BI} \\ 1, \qquad \text{otherwise} \end{cases}$$

where t_e is the time at which sporadic event occurred and x is a binary decision variable that equals 'zero' if there is enough time in the current CAP to transmit the 'GTS request' and 'one', otherwise.

The value of the x tells us whether the delay of 'one' BI is acceptable or not. If the x is equal to 'one', the deadline can still be met and the sporadic task is accepted, otherwise it is rejected.

4 Compatibility with IEEE 802.15.4 standard and implementation issues

The implementation of the dynamic inflation implicit deflation (DIID) protocol requires modifications in the data structures of IEEE 802.15.4 standard. While suggesting these modifications, care has been taken not to compromise the IEEE 802.15.4 standard compatibility. In this respect, the GTS request format and the Beacon format must be modified.

The GTS request for a sporadic allocation has to be differentiated from the regular GTS request. For this purpose, a technique similar to the one used by Koubaa *et al*. [8] is used that keeps the IEEE 802.15.4 standard compatibility intact. The sixth bit of the GTS request is adapted as the *Allocation Type* bit to indicate that it is a sporadic GTS allocation request, as shown in Fig. 8*a*. A 'one' in the bit indicates a sporadic GTS allocation request and as a result it can invoke the DIID allocation protocol that can inflate the existing GTSs if these are already allocated.

Similarly, the beacon frame that informs the cluster children about the GTS allocations must carry the information about which of the GTS slots are inflated so that the implicit deallocation can be invoked at the child nodes. For this purpose, the reserved bit 3 (as shown in Fig. 8*b*) in the standard GTS specification field of the beacon frame is adapted as *Inflation Flag* to indicate that one or more of the GTSs in the *GTS Descriptor List* is inflated and also that there is an additional element at the tail of GTS list as shown in Fig. 8*c*. This element does not contain the *GTS Descriptor* but it contains the bit map. Each bit of the bit map indicates which of the GTSs in the *GTS Descriptor List* are inflated. The number of the valid bits in this bit map and the number of *GTS Descriptors* in the *GTS Descriptor List* is specified by the *GTS Descriptor Count*. Therefore, the *GTS Descriptor Count* does not include the extra element at the tail of the *GTS Descriptor List* if the *Inflation Flag* is set.

doi: 10.1049/iet-wss.2013.0065

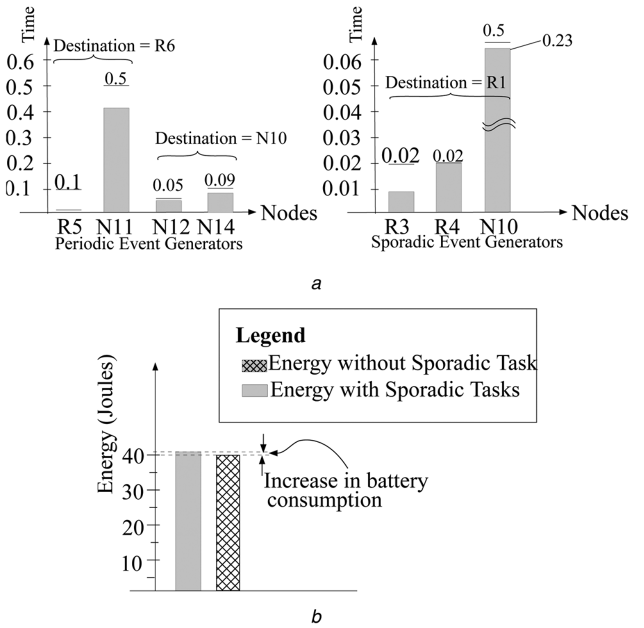

Fig. 7 *Joint scheduling of periodic flows and sporadic events, BO = 5*

a Delay and deadlines of periodic flows and sporadic events and
b Effect of sporadic scheduling on energy consumption

With these modifications, the node implementing DIID can accept the Beacon and 'GTS request' frames generated by non-DIID nodes to ensure compliance.

5 Quantification of energy dissipation

Network topology, depth of tree, the number of children at each node (that is limited to 'seven'), the number of periodic events, the number of sporadic events, the positions where these events occur are the factors affecting energy dissipation. Based on these factors, the parameters that affect the energy dissipation of the WSN can be divided into two categories: (i) energy increase because of change in the offline schedule; (ii) energy increase because of increase in reactive control traffic. The GTS mechanism of the IEEE 802.15.4 standard is supported only in a cluster tree topology that is a special form of peer-to-peer topology. Therefore, only cluster tree topology will be considered while discussing energy dissipation.

5.1 Energy increase because of change in offline schedule

Equation (4) gives the sporadic slots required at a particular node because of sporadic events. Closer to the root node, the sporadic slot requirement and the 'CAP' size is the largest. This is because all of the sporadic packets are targeted to the root. Therefore, the energy dissipation of the 'Root' is the highest. It was supposed that a sporadic packet can be accommodated within one time *slotSize*. Therefore, (7) gives the energy requirement at a particular node *j* because of the sporadic slots.

$$E_{s_j} = \lambda_j \times E_{\text{slot}} \tag{7}$$

where E_s represents the sporadic energy at node *j* and E_{slot} is the energy dissipated in one slot. Koubaa *et al.* [9] have modelled a cluster tree and those interested in the energy calculation for sporadic events can revert to the paper for further guidance.

5.2 Energy increase because of increase in reactive control traffic

Owing to implicit deallocation, no deallocation request is generated. Therefore, the energy consumption per sporadic event is limited by the number of clusters traversed as:

$E_c = E_{\text{GTSReq}} * N_{\text{cl}}$, where E_c is energy because of control traffic, N_{cl} is the number of clusters traversed and E_{GTSReq} is the energy required to transmit 'one' GTS request. Therefore, the difference of energy between the run of simulation with no sporadic task and another with sporadic tasks occurring is very small as: $E_{\text{ts}} = E_c \times N_{\text{SE}}$, where E_{ts} is the extra energy dissipation because of the increase in the control traffic for sporadic events, and N_{SE} is the number of sporadic events.

6 Comparative study and simulation results

The proposals given in the paper is implemented in OPNET and results are collected to verify correctness of proposed protocol. A network of '14' nodes is simulated. It contains 'six' routers (R1–R6) and 'eight' end nodes (N7–N14). The nodes are arranged in six clusters. The clusters overlap with each other as shown in Fig. 6*a* and are arranged in a tree as shown in Fig. 6*b*. Each cluster is headed by a router. Clusters are named after their heads (clusters 1–6). Periodic events flow from any node to any other node. Using the offline scheduler, a 'feasible' set, $f = \{p, s\}$, of 'two' types of events *p* and *s* are chosen for simulation. Therefore, *p* is a set of 'two' periodic flows {Flow1, Flow2}. Events from node N14 constitute Flow 1, having a destination of node N10. Flow 2 consists of periodic packets from nodes N11, and R5 towards R6. The parameters of periodic flows are shown in Table 3(a). On the other hand, *s* is a set of 'four' nodes (R2, R3, R4 and N10) that generate sporadic events. The sporadic data are destined for R1 that is the root of the corresponding tree. Parameters of the sporadic event used for offline scheduling purpose are shown in Table 3(b).

The offline scheduler generates another set of parameters that is used for simulation. The new set of parameters is shown in Table 4. For the feasible set, *f*, of events, Table 4 shows the values of BO and SO for each of the cluster. The CAP length is such that it has enough spare capacity for the sporadic events whenever these occur. The 'Length' column has two comma separated values. The first is the GTS

Table 3 Parameters of periodic and sporadic events used for offline scheduling

(a) Periodic task parameters

Flow	Sources	Sink	Deadline	Period	Size
1	12,14	10	0.05,0.09	0.5	64
2	5,11	6	0.1,0.5	1.0	16

(b) Sporadic task parameters where R1 is the sink and sample size is 16 bits

Sources	Freq_{max}	Deadline
3	0.5	0.02
4	0.5	0.02
10	0.5	0.5

Table 4 Simulation parameters generated by offline scheduler and used for simulation

Cluster	BO	SO	Starting time	CAP length	Device	Length	Direction	Starting slot
1	4	1	0.0	19.198	R2	1*p*, 1*s*	transmit	10
					R3	1*p*, 2*s*	transmit	11
					R4	1*p*, 1*s*	transmit	12
					R2	1*p*	receive	13
					R3	2*p*	receive	14
2	4	0	0.21504	7.678	R5	2*p*	transmit	8
					R6	2*p*	transmit	10
					R6	4*p*	receive	12
3	4	0	0.03072	9.598	R11	2*s*	transmit	8
					R10	4*p*	transmit	10
					N10	4*p*	receive	12
4	4	0	0.23040	11.518	N12	2*p*, 2*s*	transmit	14
6	4	0	0.23040	11.518	N14	2*p*, 2*s*	transmit	14

allocated for a particular node (given under 'device' column) for the periodic flow propagation. These are the GTSs allocated during convergence period before start of the periodic data communication.

The second value, after comma, is the GTS that can be allocated at the occurrence of a sporadic event. The GTS will take space from the CAP.

During simulations, the time taken by each type of data (because of periodic or sporadic events) to reach the destination was recorded and compared with the respective deadline. Fig. 7*a* shows that all of the periodic tasks and all of the sporadic tasks completed before the deadlines. A small horizontal line at the top of each of the vertical bar represents the deadline and the height of each of the bar represents the effective delay. The left side of the figure shows the periodic tasks and the right side of the figure shows the sporadic tasks. It is to be noted that the effective delay remains unaltered in the presence and absence of sporadic tasks. This is because the sporadic tasks are scheduled in the SC and the periodic schedule is not disturbed as expected. In another study, the increase in the battery consumption between the two 'runs' (with and without the sporadic events) was computed. Bars in Fig. 7*b* show that with the increase of about 1 J, 44 sporadic events reached the destination. This happened in the presence of about 120 periodic events that did not faulter on the deadlines.

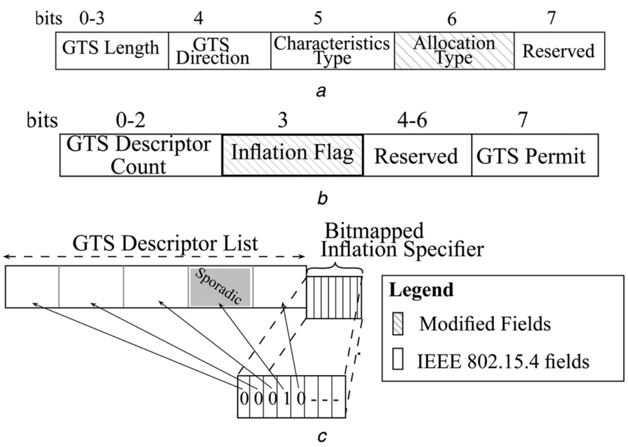

Fig. 8 *Modified data structures for DIID use*

a Modified sporadic GTS characteristics field format
b Modified GTS specification format and
c Modified GTS list format

The increase in the battery consumption is due to two reasons: (i) transmission of control packets that were generated online for sporadic allocation, and (ii) propagation of sporadic data packets. Owing to the implicit decision making, transmission of the control packets was limited to one packet per sporadic event per hop. Consequently, the increase in energy consumption because of handling of sporadic packets was minimal. Table 5 gives an overview of the features of various protocols that are studied for the comparison purpose.

From Table 5, it can be seen that no other protocol provides end-to-end protocol that can handle periodic and sporadic event propagation with hard real-time guarantees in WSN using the IEEE 802.15.4 standard across multiple hops. Simulation results prove that DIID provides such guarantees and meets the deadlines for accepted sporadic tasks along with periodic tasks that was the initial claim of this paper.

7 Related work

The IEEE 802.15.4 standard [7] provides a contention-free GTS allocation mechanism for intra-cluster real-time data propagation. For the end-to-end periodic data transmission (where source and destination are many hops away) schemes, such as proposed by Koubaa *et al*. [5], have been proposed. Similarly, for periodic scheduling, Hanzalek and Jurcik [10] have presented a mechanism for the cluster-based cyclic scheduling of WSNs. In the iGAME, Koubaa *et al.* [8] have proposed techniques for the implicit scheduling of resources to share the allocated resources (communication bandwidth) among various nodes. All of these techniques deal with the scheduling of the periodic events.

In the real-time community, a number of research works have dealt with the sporadic tasks by treating these as periodic tasks. Isovic and Fohler [4, 6, 12] have worked on the combined offline and online ways for efficient scheduling of real-time tasks. Theis and Fohler [15] have also developed a technique to convert the sporadic task parameters into periodic reservation tasks. Audsley *et al.* [16] have proposed the allocation of SC, while generating an offline schedule. This SC is used for the online scheduling of sporadic tasks when they actually occur. In [17], Kim *et al.* have proposed a generic method for combined scheduling of periodic, sporadic and best effort services. All of these research works are non-IEEE 802.15.4 standard compliant. Koubaa *et al.* [5] have suggested an improvement in the CSMA/CA mechanism of IEEE 802.15.4 standard for urgent packet, such as 'GTS request' delivery during the contention period. This is not an

doi: 10.1049/iet-wss.2013.0065

Table 5 Properties of IEEE 802.15.4 standard-based proposals

	Multi cluster scheduling	Periodic flows	Sporadic events	IEEE 802.15.4 compliance	Energy efficiency
Koubaa *et al.* [8]	✓	✓	×	✓	✓
Hanzalek and Jurcik [10]	×	✓	×	✓	✓
Kim *et al.* [11]	×	×	✓	×	×
Isovic and Fohler [12]	×	✓	✓	×	×
Choi and Kim [13]	×	✓	×	×	✓
Antasti *et al.* [14]	×	×	×	✓	✓
DIID	✓	✓	✓	✓	✓

end-to-end solution but will help in online end-to-end scheduling of sporadic tasks that handle sporadic events in IEEE 802.15.4 standard.

In [13], Choi and Kim proposed a scheduling of the sporadic along with periodic tasks for wireless fieldbus. Sporadic tasks have been scheduled in the CAP period. The sleeping period has been reduced to zero by taking SO equal to 'zero' to reduce delays for real-time tasks. Only a single cluster event propagation is taken care of. The scheduling decisions are made online that makes the online scheduler heavy in terms of processing. In [18], Kim *et al*. use the IEEE 802.15.4 standard to handle three ways to handle sporadic data. The proposal is to transmit sporadic data in CAP, CFP and sleep periods. It also does not talk about multicluster issues. Moreover, because of the GTS tailing, it cannot be applied to the clusters which partially overlap. Kim *et al*. [11] and Kim and Choi [19] propose a non-standard protocol for channel access that cannot be used in a multicluster tree network.

Other works, such as Koubaa *et al*. [5] and Yoo *et al*. [20] have proposed a prioritised channel access mechanism with careful modifications in the CSMA/CA parameters not to lose compatibility with IEEE 802.15.4 standard. Anastasi *et al*. [21] have analysed that by the careful adjustment of the CSMA/CA parameters guarantees of up to 100% are possible for the prioritised packet delivery. Anastasi *et al*. [14] also shows the impact on the reliable delivery using static CSMA/CA parameters and suggest that the dynamic tuning of such parameters will further improve the reliability.

The WSN nodes are battery operated. Therefore energy preservation must be of concern while proposing any strategy for a WSN [22–24]. For periodic scheduling, such minimisation techniques have been employed by Hanzalek and Jurcik [10] by developing the time division cluster scheduling (TDCS) tool. The TDCS is developed in Matlab by posing the scheduling problem as an optimisation problem to a mathematical tool GNU liner programing kit [25], whereby the sleep period is maximised to save energy, and the time to reach the destination is minimised to meet the deadline.

By the combination of offline and online scheduling techniques, we can guarantee the timely delivery of the sporadic along with the periodic events. The offline schedule generates the system parameters, employing TDCS [10], such that CAP of appropriate size is available, to be utilised at the time of need by the online scheduler. When a sporadic event occurs, the online scheduler on the end node requests the CH to allocate resources and make room for it in the GTS block of the forthcoming beacon period. The allocated resources are then used to carry the data in a real-time manner.

8 Conclusions and future work

We have presented a new GTS allocation protocol, DIID, where the network can be configured to handle sporadic as well as periodic events. With the use of an implicit decision used in deallocation/deflation of dynamically allocated GTS slots, the control traffic is kept low resulting in low battery consumption. By offline scheduling of sporadic events, the sensor nodes are offloaded from the dynamic scheduling. Similarly, with the offline laxity calculation, the online acceptance test is reduced to a very simple test on a Boolean variable.

A further improvement of our protocol to be investigated is the cyclic sporadic flow handling, where the sporadic data are large enough that it cannot be accommodated within GTS of a single BI. We also envision an extension of this work, which will include a WSN with non-root destinations.

9 References

1 Paavola, M., Leiviska, K.: 'Wireless sensor networks in industrial automation', in Silvestre-Blanes, J. (Ed.): 'Factory automation' (InTech, 2010), pp. 202–220

2 Flammini, A., Ferrari, P., Marioli, D., Sisinni, E., Taroni, A.: 'Wired and wireless sensor networks for industrial applications', *Microelectron. J.*, 2009, **40**, (9), pp. 1322–1336

3 Bonastre, A., Capella, J., Ors, R., Peris, M.: 'In-line monitoring of chemical-analysis processes using wireless sensor networks', *TrAC Trends Anal. Chem.*, 2012, **34**, pp. 111–125

4 Isovic, D., Fohler, G.: 'Efficient scheduling of sporadic, aperiodic, and periodic tasks with complex constraints'. Proc. Twenty-first IEEE Real-Time Systems Symp., 2000, pp. 207–216

5 Koubaa, A., Alves, M., Nefzi, B., *et al*.: 'Improving the IEEE 802.15. 4 slotted CSMA/CA MAC for time-critical events in wireless sensor networks'. Proc. Workshop of Real-Time Networks (RTN 2006), Satellite Workshop to (ECRTS 2006), 2006

6 Isovic, D., Fohler, G.: 'Handling mixed sets of tasks in combined offline and online scheduled real-time systems', *Real-Time Syst.*, 2009, **43**, (3), pp. 296–325

7 IEEE Standards: 'Standard for Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs)' ANSI/IEEE 80215, 2003, vol. 4.

8 Koubaa, A., Alves, M., Tovar, E.: 'i-GAME: an implicit GTS allocation mechanism in IEEE 802.15. 4 for time-sensitive wireless sensor networks'. Eighteenth Euromicro Conf. Real-Time Systems, 2006, pp. 183–192

9 Koubaa, A., Alves, M., Tovar, E.: 'Modeling and worst-case dimensioning of cluster-tree wireless sensor networks'. Twenty-seventh IEEE Int. Real-Time Systems Symp., 2006 (RTSS '06), 2006, pp. 412–421

10 Hanzalek, Z., Jurcik, P.: 'Energy efficient scheduling for cluster-tree wireless sensor networks with time-bounded data flows: application to IEEE 802.15. 4/ZigBee', *IEEE Trans. Ind. Inform.*, 2010, **6**, (3), pp. 438–450

11 Kim, T., Lee, D., Ahn, J., Choi, S.: 'Priority toning strategy for fast emergency notification in IEEE 802.15.4 LR-WPAN'. Proc. 15th Joint Conf. Communications and Information (JCCI), Citeseer, 2005, pp. 18–30

12 Isovic, D., Fohler, G.: 'Handling sporadic tasks in off-line scheduled distributed realtime systems'. Proc. 11th Euromicro Conf. Real-Time Systems, 1999, pp. 60–67

13 Choi, D., Kim, D.: 'Wireless fieldbus for networked control systems using LR-WPAN', *Int. J. Control Autom. Syst.*, 2008, **6**, (1), pp. 119

14 Anastasi, G., Conti, M., Di Francesco, M., Neri, V.: 'Reliability and energy efficiency in multi-hop IEEE 802.15.4/ZigBee wireless sensor

networks'. 2010 IEEE Symp. Computers and Communications (ISCC)',, 2010, pp. 336–341
15 Theis, J., Fohler, G.: 'Transformation of sporadic tasks for off-line scheduling with utilization and response time trade-offs.' (RTNS, 2011), pp. 119–128
16 Audsley, N.C., Burns, A., Richardson, M.F., Wellings, A.J.: 'Hard real-time scheduling: the deadline monotonic approach'. Proc. Eighth IEEE Workshop on Real-Time Operating Systems and Software, 1991, pp. 127–132
17 Kim, D.S., Choi, D.H., Mohapatra, P.: 'Real-time scheduling method for networked discrete control systems', *Control Eng. Pract.*, 2009, **17**, (5), pp. 564–570
18 Kim, D.S., Jeon, J., Mohapatra, P.: 'Scheduling of wireless control networks based on IEEE 802.15. 4 networks: mixed traffic environment', *Control Eng. Pract.*, 2011, **19**, (10), pp. 1223–1230
19 Kim, T.H., Choi, S.: 'Priority-based delay mitigation for event-monitoring IEEE 802.15.4 LR-WPANs', *IEEE Commun. Lett.*, 2006, **10**, (3), pp. 213–215
20 Yoo, S.E., Chong, P.K., Kim, D., *et al.*: 'Guaranteeing real-time services for industrial wireless sensor networks with IEEE 802.15.4', *IEEE Trans. Ind. Electron.*, 2010, **57**, (11), pp. 3868–3876
21 Anastasi, G., Conti, M., Di Francesco, M.: 'A comprehensive analysis of the MAC unreliability problem in IEEE 802.15.4 wireless sensor networks', *IEEE Trans. Ind. Inform.*, 2011, **7**, (1), pp. 52–65
22 Wu, Y., Liu, W.: 'Routing protocol based on genetic algorithm for energy harvesting-wireless sensor networks', *IET Wirel. Sens. Syst.*, 2013, **3**, (2), pp. 112–118
23 Hoang, D.C., Kumar, R., Panda, S.K.: 'Optimal data aggregation tree in wireless sensor networks based on intelligent water drops algorithm', *IET Wirel. Sens. Syst.*, 2012, **2**, (3), pp. 282–292
24 Azevedo, J., Santos, F., Rodrigues, M., Aguiar, L.: 'Sleeping ZigBee networks at the application layer', *IET Wirel. Sens. Syst.*, 2013, **4**, (1), pp. 35–41
25 Makhorin, A.: GLPK (GNU linear programming kit); 2006. http://www.gnu.org/software/glpk/, [Online; accessed 24 January 2014]

doi: 10.1049/iet-wss.2013.0065