IPP Hurray!
www.hurray.isep.ipp.pt

Technical Report

i-GAME: An Implicit GTS Allocation Mechanism in IEEE 802.15.4 for Time-Sensitive Wireless Sensor Networks

Anis Koubaa

Mário Alves

Eduardo Tovar

TR-060706
Version: 1.0
Date: Jul 2006

i-GAME: An Implicit GTS Allocation Mechanism in IEEE 802.15.4 for Time-Sensitive Wireless Sensor Networks

Anis KOUBAA, Mário ALVES, Eduardo TOVAR

IPP-HURRAY!
Polytechnic Institute of Porto (ISEP-IPP)
Rua Dr. António Bernardino de Almeida, 431
4200-072 Porto
Portugal
Tel.: +351.22.8340509, Fax: +351.22.8340509
E-mail: { akoubaa@dei ,mjf@, emt@dei }.isep.ipp.pt
http://www.hurray.isep.ipp.pt

Abstract

The IEEE 802.15.4 Medium Access Control (MAC) protocol is an enabling technology for time sensitive wireless sensor networks thanks to its Guaranteed-Time Slot (GTS) mechanism in the beacon-enabled mode. However, the protocol only supports explicit GTS allocation, i.e. a node allocates a number of time slots in each superframe for exclusive use. The limitation of this explicit GTS allocation is that GTS resources may quickly disappear, since a maximum of seven GTSs can be allocated in each superframe, preventing other nodes to benefit from guaranteed service. Moreover, the GTSs may be only partially used, resulting in wasted bandwidth. To overcome these limitations, this paper proposes i-GAME, an implicit GTS Allocation Mechanism in beacon-enabled IEEE 802.15.4 networks. The allocation is based on implicit GTS allocation requests, taking into account the traffic specifications and the delay requirements of the flows. The i-GAME approach enables the use of a GTS by multiple nodes, while all their (delay, bandwidth) requirements are still satisfied. For that purpose, we propose an admission control algorithm that enables to decide whether to accept a new GTS allocation request or not, based not only on the remaining time slots, but also on the traffic specifications of the flows, their delay requirements and the available bandwidth resources. We show that our proposal improves the bandwidth utilization compared to the explicit allocation used in the IEEE 802.15.4 protocol standard. We also present some practical considerations for the implementation of i-GAME, ensuring backward compatibility with the IEEE 801.5.4 standard with only minor add-ons.

i-GAME: An Implicit GTS Allocation Mechanism in IEEE 802.15.4 for Time-Sensitive Wireless Sensor Networks

Anis KOUBAA, Mário ALVES, Eduardo TOVAR

IPP-HURRAY! Research Group, Polytechnic Institute of Porto
Rua Dr. Antonio Bernardino de Almeida, 431, 4200-072 Porto, PORTUGAL
{akoubaa, emt}@dei.isep.ipp.pt , mjf@isep.ipp.pt

Abstract

The IEEE 802.15.4 Medium Access Control (MAC) protocol is an enabling technology for time sensitive wireless sensor networks thanks to its Guaranteed-Time Slot (GTS) mechanism in the beacon-enabled mode. However, the protocol only supports explicit GTS allocation, i.e. a node allocates a number of time slots in each superframe for exclusive use. The limitation of this explicit GTS allocation is that GTS resources may quickly disappear, since a maximum of seven GTSs can be allocated in each superframe, preventing other nodes to benefit from guaranteed service. Moreover, the GTSs may be only partially used, resulting in wasted bandwidth. To overcome these limitations, this paper proposes ***i-GAME****, an* ***i****mplicit* ***G****TS* ***A****llocation* ***Me****chanism in beacon-enabled IEEE 802.15.4 networks. The allocation is based on implicit GTS allocation requests, taking into account the traffic specifications and the delay requirements of the flows. The i-GAME approach enables the use of a GTS by multiple nodes, while all their (delay, bandwidth) requirements are still satisfied. For that purpose, we propose an admission control algorithm that enables to decide whether to accept a new GTS allocation request or not, based not only on the remaining time slots, but also on the traffic specifications of the flows, their delay requirements and the available bandwidth resources. We show that our proposal improves the bandwidth utilization compared to the explicit allocation used in the IEEE 802.15.4 protocol standard. We also present some practical considerations for the implementation of i-GAME, ensuring backward compatibility with the IEEE 801.5.4 standard with only minor add-ons.*

1. Introduction

The IEEE 802.15.4 protocol [1] has been recently adopted as a communication standard for Low-Rate Wireless Local Area Networks (LR-WPANs). It presents the advantage to be flexible enough for fitting different requirements of potential applications by adequately tuning its parameters. Even though the IEEE 802.15.4 protocol was not specifically designed for Wireless Sensor Networks (WSNs), it is intended to be suitable for them. In fact, low data rate, low power consumption and low cost wireless networking are the key features of the IEEE 802.15.4 protocol, which typically fit the requirements of WSNs.

More specifically, the IEEE 802.15.4 Medium Access Control (MAC) protocol has the ability to provide very low duty cycles (up to 0.1 %). This feature is particularly interesting for WSN applications, where energy consumption and network lifetime are main concerns. Additionally, the IEEE 802.15.4 protocol also provides real-time guarantees by using the Guaranteed-Time Slot (GTS) mechanism. This feature is quite attractive for time-sensitive WSNs. In fact, when operating in beacon-enabled mode, i.e. beacon frames are transmitted periodically by a central node called *PAN coordinator* for synchronizing the network, the IEEE 802.15.4 protocol allows the allocation/deallocation of GTSs in a superframe for nodes that require real-time guarantees. Hence, the GTS mechanism provides a minimum service guarantee for the corresponding nodes and enables the prediction of the worst-case performance for each node's application.

However, the GTS mechanism, as proposed in the standard [1], presents some limitations in terms of efficiency and deployment in WSNs with a large number of nodes. In fact, during each superframe (divided into sixteen time slots) only up to seven GTSs (from 1 up to 15 time slots per GTS) can be allocated, forming the *Contention-Free Period* (CFP) (see Fig. 1). The remaining time slots in the superframe compose the *Contention Access Period* (CAP) using Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) as a MAC protocol.

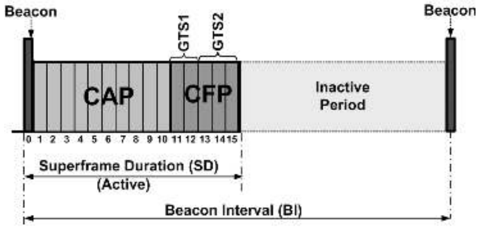


Fig. 1. Beacon Interval and Superframe Structure

Since each GTS is exclusively assigned to one node, the number of nodes involved in the CFP is limited to seven or less. This is because the IEEE 802.15.4 standard [1] assumes that a node performs an **explicit GTS allocation request** by asking the PAN coordinator for a certain number of time slots. A node is admitted to transmit during the CFP, if the number of available time slots in the superframe is higher than requested, and the minimum CAP length will not be violated after the allocation [1]. Two negative impacts may result from this explicit allocation scheme.

This work was partially funded by FCT under CISTER research unit (UI608).

1. The GTSs can be quickly consumed by a few number of nodes, preventing the others from having a guaranteed service.
2. A node with a low arrival rate that has allocated a GTS, may only partially use it (when the amount of guaranteed bandwidth is higher than its arrival rate). This leads to underutilization of the GTS bandwidth resources. Due to the pre-fixed time slot duration in a superframe, it is practically impossible to balance the arrival rate of a node and its guaranteed GTS bandwidth. The amount of wasted bandwidth increases with the variance between the guaranteed bandwidth and the arrival rate.

This paper proposes a simple and effective solution to overcome the previously described limitations of the explicit GTS allocation in the IEEE 802.15.4 protocol. Basically, the idea consists in sharing the same GTS between multiple nodes, instead of being exclusively dedicated to one node, if a certain schedule that satisfies the requirements of all requesting nodes exists. Sharing a GTS by several nodes means that the time slots of this GTS are dynamically allocated to different nodes in each superframe, according to a given schedule. In contrast, an explicit allocation statically devotes a GTS to only one node in all subsequent superframes. Hence, the GTS allocation mechanism proposed in this paper is based on the traffic specification of the requesting nodes, their delay requirements, and the available GTS resources. Instead of asking for a fixed number a time slots, a node that wants to have a guaranteed service sends its traffic specification and delay requirement to the PAN coordinator. The latter runs an admission control algorithm based on this information and the amount of available GTS resources. The new allocation request will be accepted if there is a schedule that satisfies its requirements and those of all other previously accepted allocation requests; otherwise, the new allocation request is rejected. We refer to this as the **implicit GTS allocation mechanism** (i-GAME). We show that i-GAME has the advantage to accept multiple flows sharing the same GTS while still meeting their delay requirements. It also improves the utilization of the CFP by reducing the amount of wasted bandwidth of GTSs and maximizes the duration of the CAP, since the CFP length is reduced to a minimum.

Related Work. The performance of the explicit GTS allocation in IEEE 802.15.4 has been recently evaluated in [2]. That work proposes a delay bound analysis of an explicit GTS allocation. It also analyzes the impact of the beacon and superframe orders on the throughput, delay and power efficiency of a GTS allocation. In this paper, we extend the work in [2] by considering implicit GTS allocations. We also prove the improvement as compared to the explicit GTS allocation approach, in terms of bandwidth utilization.

Basically, the problem that we are addressing in this paper can be regarded as analyzing the schedulability of a given number of flows sharing a certain number of time slots. This problem has already been addressed by some works in the literature, but with completely different contexts and assumptions, as briefly outlined next.

In [3-4], the authors have addressed multicycle polling scheduling in fieldbus networks. These papers have contributed to the schedulability analysis of a set of periodic tasks with deadlines equal to periods under Rate Monotonic (RM) and Earliest Deadline First (EDF) scheduling policies, where the nodes polled are different from one cycle to another. In both approaches, the idea consists in finding the minimum cycle, called primary cycle, which corresponds to the greatest common divisor of all task periods, and computing the number of time slots needed to transmit periodic traffic inside each cycle, if the task set is schedulable. The last step consists in executing tasks according to their priorities (using RM or EDF) in each primary cycle.

Our work differs from these approaches in two aspects. First, we don't consider periodic message arrivals, but we adopt a more general representation of the traffic using the (b,r)-curve model where b is the burst size of the flow and r is the average rate. This traffic model also incorporates the classical representation of the periodic arrival model with or without jitter [5]. For that reason, our analysis is based on the Network Calculus theory. Second, the durations of the cycles in the pre-cited approaches are fixed and related to the periods of the flows. This does not match with our case, since in the IEEE 802.15.4 protocol one cycle is represented by a Beacon Interval (BI) (see Fig. 1), whose duration depends on the beacon order parameter as it will be shown in Section 2. Moreover, since the IEEE 802.15.4 protocol does not allow more than seven GTS allocations, this may restrict the number of time slots in each cycle in contrast with the approaches in [3-4], where the number of time slots is only limited by the durations of the primary cycle and the time slot.

Contributions of this paper. The contributions of this paper are the following.

- First, we present the motivation for an implicit GTS allocation mechanism for the IEEE 802.15.4 protocol, showing that the explicit allocation mechanism proposed by the standard lacks bandwidth efficiency, particularly for low rate WSN applications (Section 3). We also introduce the implicit allocation mechanism, i-GAME, through a practical example.
- Second, we evaluate the schedulability analysis of an implicit GTS allocation of k time slots shared by N nodes, where $k < N$, under round robin scheduling (Section 4). For that purpose we derive the service curve and the delay bound guaranteed by such an allocation, defined by the tuple (b, r, D) where b is the burst size, r is the arrival rate, and D is the delay requirement.
- Finally, we present the i-GAME admission control mechanism, based on our analysis and we provide some guidelines for its implementation, with minor add-ons to the IEEE 802.15.4 standard protocol defined in [1] (Section 5).

2. Background

2.1 Overview of the IEEE 802.15.4 MAC protocol

The IEEE 802.15.4 MAC protocol supports two operational modes that may be selected by a central node called *PAN coordinator*: (1) the *non beacon-enabled mode* where the MAC is ruled by non-slotted CSMA/CA; (2) the *beacon-enabled mode* where beacons are periodically sent by the PAN coordinator to identify its PAN and synchronize nodes that are associated with it. The most relevant MAC features are outlined next.

In this paper, we only consider the beacon-enabled mode, since it enables GTS allocations. In beacon-enabled mode, the *Beacon Interval* (BI) defines the time between two consecutive beacons, and includes an active period and, optionally, an inactive period. The active period, called *superframe*, is divided into 16 equally-sized time slots, during which data frame transmissions are allowed. During the inactive period (if it exists), all nodes may enter into a sleep mode, thus saving energy. Fig. 1 illustrates the beacon interval and the superframe structure.

The Beacon Interval and the *Superframe Duration* (SD) are determined by two parameters, the *Beacon Order* (BO) and the *Superframe Order* (SO), respectively.

The Beacon Interval is defined as follows:

$$\begin{aligned} BI &= aBaseSuperframeDuration \cdot 2^{BO}, \\ &for\ \ 0 \le BO \le 14 \end{aligned} \qquad (1)$$

The Superframe Duration, which determines the length of the active period, is defined as follows:

$$\begin{aligned} SD &= aBaseSuperframeDuration \cdot 2^{SO}, \\ &for\ \ 0 \le SO \le BO \le 14 \end{aligned} \qquad (2)$$

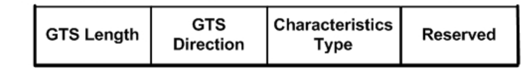
In Eqs.(1) and (2), *aBaseSuperframeDuration* denotes the minimum duration of the superframe, corresponding to $SO = 0$. This value corresponds to 15.36 ms, assuming 250 kbps in the 2.4 GHz frequency band, which will be considered throughout the rest of this paper.

By default, the nodes compete for medium access using slotted CSMA/CA during the *Contention Access Period* (CAP). More details can be found in [1].

2.2 Explicit GTS allocation in IEEE 802.15.4

The IEEE 802.15.4 protocol also offers the possibility of having a *Contention-Free Period* (CFP) within the superframe (Fig. 1). The CFP, being optional, is activated upon request from a node to the PAN coordinator for allocating a certain number of time slots.

Fig. 2 shows the GTS characteristics field format sent within an allocation request command frame [1] by a node to the PAN coordinator.

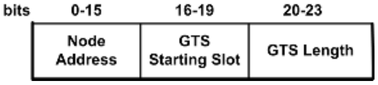


bits	0-3	4	5	6-7
	GTS Length	GTS Direction	Characteristics Type	Reserved

Fig. 2. GTS characteristics field format in IEEE 802.15.4

The node explicitly expresses the number of time slots that it wants to allocate in the *GTS Length* field. Note that the GTS length can be up to 15 time slots. The *GTS Direction* field specifies if the GTS is in receive-only mode (value = 1), i.e. data is transmitted from the PAN coordinator to the requesting node, or in transmit-only mode (value = 0), i.e. data is transmitted from the requesting node to the PAN coordinator. The *Characteristics Type* field refers to a GTS allocation if it is set to one or a GTS deallocation if it is set to zero.

Upon receiving this request, the PAN coordinator checks whether there are sufficient time slots available in the superframe for this request. If the number of available time slots in the superframe is smaller than the number requested, the GTS allocation request is rejected, otherwise it is accepted. The PAN coordinator must ensure that the CAP length remains always greater than *aMinCAPLength* equal to 7.04 ms [1]. In the former case, the corresponding node may send its data frames during the CAP, but with no guarantee. If the GTS allocation request is accepted, the admitted node must keep track of beacon frames for checking which time slots have been allocated in the current superframe. This information is located in the GTS descriptor field (Fig. 3), which is embedded in each beacon frame. A beacon frame cannot have more than seven GTS descriptors, limiting the number of GTSs to seven.



bits	0-15	16-19	20-23
	Node Address	GTS Starting Slot	GTS Length

Fig. 3. GTS Descriptor Field Format in IEEE 802.15.4

The explicit GTS allocation adopted by the standard has the advantage of being simple. However, it may be not efficient enough in terms of bandwidth utilization for flows with low arrival rates, which is typically the case in wireless sensor networks, since the guaranteed bandwidth of a GTS can be much higher than the arrival rates (see Section 3.2).

2.3 Delay bound analysis using Network Calculus

In Network Calculus theory [7], the delay bound analysis for a given data flow with a cumulative arrival function $R(t)$ assumes the following.

1. It exists an arrival curve $\alpha(t)$ that upper bounds $R(t)$ such that $\forall s$, $0 \le s \le t$, $R(t) - R(s) \le \alpha(t-s)$. This inequality means that the amount of traffic that arrives to receive service in any interval $[s,t]$ never exceeds $\alpha(t-s)$.
2. It exists a minimum service curve $\beta(t)$ guaranteed to $R(t)$.

Then, the delay bound, D_{max}, for a data flow with an arrival curve $\alpha(t)$ that receives the service $\beta(t)$ is the maximum horizontal distance between $\alpha(t)$ and $\beta(t)$:

$$D_{\max} = h(\alpha, \beta) = \sup_{s \ge 0} \left\{ \inf \left\{ \tau \ge 0 : \alpha(s) \le \beta(s+\tau) \right\} \right\} \qquad (3)$$

Fig. 4 presents an example of the delay bound for a linear arrival curve $\alpha(t)=b+r\cdot t$ that receives a *rate-latency* service curve $\beta_{R,T}(t)=R\cdot(t-T)^{+}$, where $R\geq r$ is the guaranteed bandwidth, T is the maximum latency of the service and $(x)^{+}=\max(0,x)$. This service curve is typically used for servers that provide a bandwidth guarantee with a certain latency. The latency T refers to the deviation of the service (e.g. blocking factor of non-preemptive transmissions).

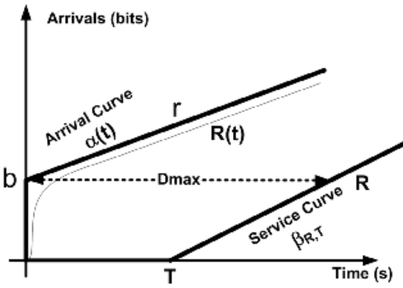


Fig. 4. Arrival Curve, Service Curve and Delay Bound

The delay bound D_{max} (presented in Fig. 4) guaranteed for the data flow with the arrival curve $\alpha(t)=b+r\cdot t$ (also called (*b*, *r*) curve) by the service curve $\beta_{R,T}(t)=R\cdot(t-T)^{+}$ is computed as follows [7]:

$$D_{\max}=\frac{b}{R}+T \tag{4}$$

2.4 Delay bound of an explicit GTS Allocation

In [2], the authors have derived the delay bound for flows with an arrival curve $\alpha(t)=b+r\cdot t$ using Network Calculus. It has been shown that the service curve offered by a GTS allocation of *n* time slots is approximated by a rate-latency service curve $\beta_{R_n,T_n}(t)=R_n\cdot(t-T_n)$, where R_n is the guaranteed bandwidth of a GTS defined as:

$$R_n=n\cdot\left(\frac{T_{data}}{BI}\cdot C\right) \tag{5}$$

and T_n is the latency of the service expressed as:

$$T_n=BI-n\cdot TS \tag{6}$$

T_{data} defines the maximum duration used for data frame transmission inside a GTS, without taking the control overheads (inter-frame spacing (IFS) and acknowledgement) into account [2]. *C* denotes the data rate equal to 250 kpbs.

As a result, it is shown that the delay bound guaranteed by the service curve $\beta_{R_n,T_n}(t)$ for a data flow bounded by a (b,r) curve is:

$$D_{n,\max}=\frac{b}{n\cdot\left(\left(T_{data}\cdot C\right)/BI\right)}+\left(BI-n\cdot Ts\right) \tag{7}$$

Another service curve in the form of a stair function was also derived in [2]. However, the analysis presented in that paper considers the rate-latency service curve $\beta_{R_n,T_n}(t)$ of one GTS.

3. i-GAME: An Implicit GTS Allocation Mechanism

3.1 System model and assumptions

We consider an IEEE 802.15.4 cluster composed of a set of sensor nodes in the range of a particular node considered as the PAN coordinator. Note that this star topology may be particularly interesting for large-scale sensor networks when using clustering and/or two-tiered architectures [6]. Moreover, the IEEE 802.15.4 supports cluster-tree topologies, which extend the star network by means of child coordinators [1] that synchronize the nodes out of the range of the PAN coordinator.

We assume that the PAN coordinator sets up the network with a superframe structure defined by the beacon order *BO* and the superframe order *SO*. The beacon interval (*BI*) and the superframe duration (*SD*) are computed using Eqs. (1) and (2), respectively.

Each node *i* generates a flow F_i bounded by the arrival curve $\alpha_i(t)=b_i+r_i\cdot t$, where b_i is the maximum burst size, r_i is the average arrival rate and D_i denotes the delay requirement of flow F_i. We represent flow F_i by the tuple $F_{spec,i}=(b_i,r_i,D_i)$.

Let R_{TS} denote the guaranteed bandwidth per one time slot. Observe that R_{TS} can be computed using Eq. (5) for $n=1$. For a GTS with a length of *k* allocated time slots (*k*<15), we denote as R_{kTS} the bandwidth guaranteed by *k* time slots expressed as:

$$R_{kTS}=k\cdot R_{TS} \tag{8}$$

The main problem addressed in this paper is how to **fairly share** the allocation of *k* time slots in the CFP between *N* requesting nodes, with respect to their $F_{spec,i}=(b_i,r_i,D_i)$.

Intuitively, *N* flows are allowed to share a GTS allocation of *k* time slots, if two necessary conditions (C1) and (C2) hold:

$$\begin{aligned}&\textbf{(C1)}\quad \sum\nolimits_{i=1}^{N} r_i\leq R_{kTS}\\&\textbf{(C2)}\quad D_{i,\max}\leq D_i,\forall 1\leq i\leq N\end{aligned} \tag{9}$$

(C1) states that the sum of all arrival rates does not exceed the entire bandwidth of *k* time slots. (C2) states that the delay bound guaranteed by the allocation does not exceed the delay requirement, for each flow F_i.

3.2 Bandwidth utilization of explicit GTS allocations

This section defines the bandwidth utilization of a GTS allocation. It also presents the limitations of an explicit allocation in terms of bandwidth utilization efficiency.

Consider a flow $F_i=(b_i,r_i,D_i)$ that has an explicit GTS allocation of k_i time slots. Then, the bandwidth utilization of this GTS allocation is defined as:

$$U_{k_iTS}=r_i/R_{k_iTS}=r_i/\left(k_i\cdot R_{TS}\right) \tag{10}$$

Now, for a CFP of a length k time slots, $k \leq 15$, containing all allocated GTSs ($k = \sum_{i=1}^{N} k_i$) and corresponding to N allocating nodes, the average bandwidth utilization of the CFP is defined as:

$$U_{CFP} = U_{kTS} = \frac{1}{N}\sum_{i=1}^{N} U_{k_i TS} = \frac{1}{N \cdot R_{TS}}\sum_{i=1}^{N}\frac{r_i}{k_i} \quad (11)$$

Observe that the minimum bandwidth that can be allocated is R_{TS} (it is not divisible). It is logical to assume that, with an explicit allocation, a node i that requests a GTS allocation of k_i time slots has an arrival rate r_i that satisfies:

$$\left(k_i - 1\right) \cdot R_{TS} < r_i \leq k_i \cdot R_{TS} \quad (12)$$

From Eqs. (8), (10) and (12), we obtain:

$$\frac{\left(k_i - 1\right)}{k_i} < U_{k_i TS} \leq 1 \quad (13)$$

Then, the minimum utilization limit is defined as:

$$U_{\min}^{k} = \frac{\left(k - 1\right)}{k} \quad \forall k, 1 \leq k \leq 15 \quad (14)$$

Fig. 5 presents the minimum utilization limits for different GTS lengths values, for one node.

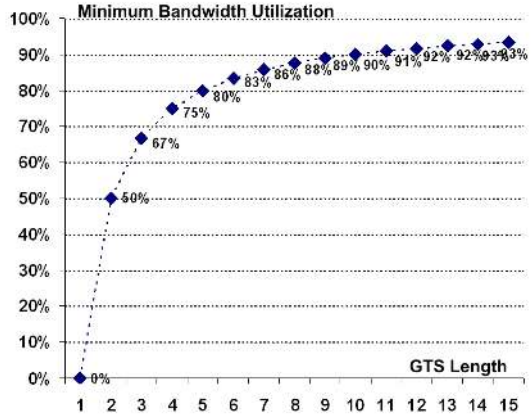


Fig. 5. Minimum Utilization Limits of an Explicit Allocation

From Fig. 5, it can be understood that the lowest utilizations can be experimented for GTSs with one time slot allocation. This is because the arrival rates of the flows can be low fractions of the indivisible R_{TS}, which triggers the motivation for sharing the time slot with other nodes, if the delay requirements of the flows can still be satisfied. This case is most likely to happen in sensor networks since their arrival rates may be particularly low.

3.3 Improving bandwidth utilization via implicit GTS allocations: i-GAME

According to condition (C1) in Eq. (9), N flows may share one GTS if the sum of their arrival rates is smaller or equal to the guaranteed bandwidth of the GTS. The main problem in this case is to find the adequate time slot allocation schedule in each beacon interval that respects a per-flow guaranteed bandwidth greater or equal to its arrival rate. The complexity of finding the adequate schedule depends on the number of GTS allocation requests and on the per-flow utilization of the GTS. A particular simple form of sharing the GTS is by using round robin scheduling, thus providing a fair share. However, round robin offers the same amount of guaranteed bandwidth to all flows without any differentiation. Hence, round robin is adequate when the arrival rate of each flow sharing the GTS is smaller than the bandwidth guaranteed by a fair share of the GTS. More formally, for a GTS allocation of k time slots fairly shared by N flows $F_{spec,i} = \left(b_i, r_i, D_i\right), i = 1..N$, then:

$$r_i \leq \frac{k \cdot R_{Ts}}{N} \qquad \forall i = 1..N \quad (15)$$

Note that the fair sharing of a GTS is effective when the arrival rates of the flows are similar. For instance, a flow with an arrival rate of 20 kbps cannot fairly share the same resource with a flow with an arrival rate of 1 kbps. Hence, we assume that Eq. (12) must hold for flows that are candidates for sharing the same GTS with other flows. This assumption is relevant for WSN applications, since flows generated by sensor nodes have similar behaviors.

For the sake of simplicity and without loss of generality, in this paper we analyze flows with (at most) one time slot allocation. We make this assumption for two reasons.

1. It is common in WSNs that flows are generated at low rates. It has been shown in [2] that the guaranteed bandwidth per one time slot allocation for a full duty cycle ($BO = SO$), is comprised between 9.38 kbps and 13.50 kbps, depending on the superframe order (SO). The traffic pattern of most WSN applications should have arrival rates much lower than these values, since in WSNs it is most likely to have a large number of nodes with low rates rather than a small number of nodes with high rates.
2. According to Fig. 5, the case of one time slot allocation is the most interesting for the i-GAME approach since the utilization (without i-GAME) can be very low (less than 50%), particularly for flows with low rates.

Note that the methodology presented next can be easily extended based on the same principles in order to merge flows requesting the same number of k time slots (satisfying Eq. (12), for $k > 1$) into one GTS with reduced size.

Based on the definition of implicit GTS allocation, the utilization of the GTS of k time slots shared by N flows $F_{spec,i} = \left(b_i, r_i, D_i\right), i = 1..N$, is defined as:

$$U_{kTS}^{N} = \frac{1}{k \cdot R_{TS}}\sum_{i=1}^{N} r_i \quad (16)$$

The bandwidth utilization of each flow in the GTS is defined as:

$$U_{i,kTS} = \frac{r_i}{k \cdot R_{TS}} \tag{17}$$

In summary, this paper considers flows requesting an implicit GTS allocation with arrival rates $r_i \le R_{Ts}$, which corresponds to one time slot allocation in case of an explicit allocation. Our problem is then reduced to find a fair share of implicit GTS allocations into a CFP with a length of *k* time slots for *N* requesting nodes, where $k \le N$. Note that in this case, the CFP length (corresponding to implicit allocations) does not exceed seven time slots (*k*<7) since only seven GTSs, each of one time slot length, can be allocated in a given superframe.

3.4 A practical intuition on the i-GAME approach

To give a practical intuition on the implicit GTS allocation approach, we present the following illustrative example.

Assume an IEEE 802.15.4 cluster where the PAN coordinator sets up the superframe structure with $BO = 0$ and $SO = 0$. This configuration corresponds to $BI = SD = 15.36\text{ ms}$, $Ts = 0.96\text{ ms}$ and $R_{TS} = 9.38\text{ kbps}$ [2].

A first request. Now, let a node *A* generate a flow F_A bounded by the arrival curve $\alpha_A(t) = 0.2 + 3 \cdot t$ kbits (a burst size with b_A = 200 bits and an arrival rate of r_A = 3 kbps) and with a delay requirement $D_A = 150\text{ ms}$. Then, $F_{spec,A} = (200\text{ bits}, 3\text{ kbps}, 150\text{ ms})$. When node *A* requests a GTS allocation, it must send $F_{spec,A}$ to the PAN coordinator, which has to decide whether to accept the flow or not. Based on the results in [2], with $BO = 0$ and $SO = 0$ (Eqs. (5) and (6)), the service curve offered by one time slot allocation is $\beta_{1node,1TS}(t) = 9.38 \cdot (t - 14.40)^+$ kbits. Figs. 6.a and 7.a present the allocation of the GTS by node *A* and its service curve, respectively. Using Eq. (7), the PAN coordinator can compute the delay bound guaranteed by one time slot allocation based on $F_{spec,A}$. This delay bound is $D_{A,\max} = 35.72\text{ ms}$. Observe also that the guaranteed bandwidth by one time slot allocation (9.39 kbps) is higher than the arrival rate (3 kbps). As a result, both conditions (C1) and (C2) in Eq. (9) are satisfied; hence, the flow is accepted for one time slot allocation. The GTS will be partially used by node *A* with an utilization $r_A / R_{TS} = 32\%$ (see Eq. (16)).

A second request. Assume that a second node *B* generating a flow F_B with a traffic specification $F_{spec,B} = (400\text{ bits, } 2\text{ kbps, } 150\text{ ms})$ wants to allocate a GTS. The traditional explicit mechanism would require the allocation of a new time slot exclusively for node *B*. This would lead to an additional wasted bandwidth, as for node *A*, since the arrival rate of r_B is lower than R_{TS}. We propose a different approach that is to share the previous GTS allocation with node *A*, if it would be possible to respect

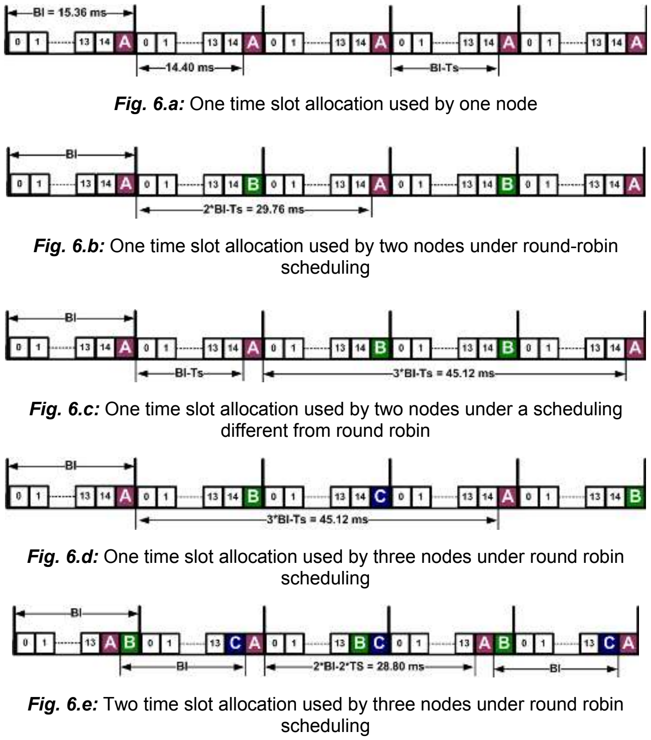


Fig. 6.a: One time slot allocation used by one node

Fig. 6.b: One time slot allocation used by two nodes under round-robin scheduling

Fig. 6.c: One time slot allocation used by two nodes under a scheduling different from round robin

Fig. 6.d: One time slot allocation used by three nodes under round robin scheduling

Fig. 6.e: Two time slot allocation used by three nodes under round robin scheduling

Fig. 6. Different Implicit GTS Allocations

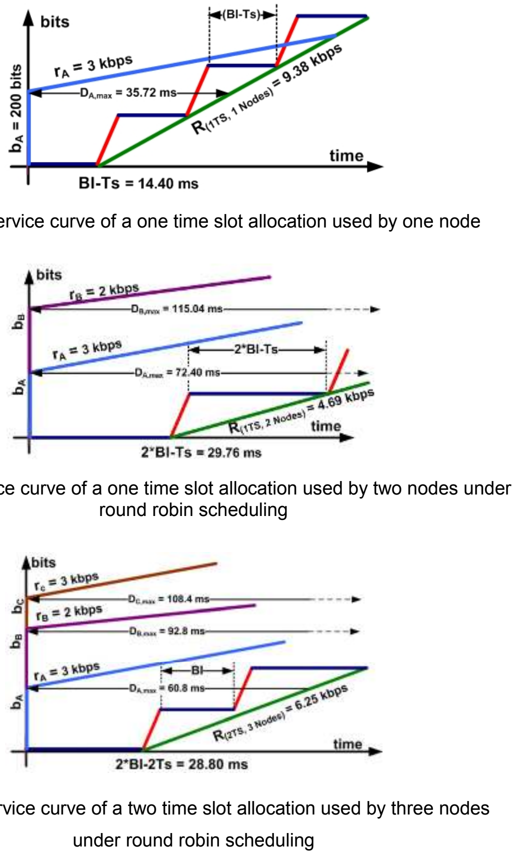


Fig. 7.a: Service curve of a one time slot allocation used by one node

Fig. 7.b: Service curve of a one time slot allocation used by two nodes under round robin scheduling

Fig. 7.c: Service curve of a two time slot allocation used by three nodes under round robin scheduling

Fig. 7. Service Curves of Implicit GTS Allocations

both $F_{spec,A}$ and $F_{spec,B}$. The problem is to determine the service curve offered by the same time slot for each flow.

Assuming that the sharing of this time slot is based on round robin scheduling, the time slot alternates between both flows in each beacon interval (refer to Figs. 6.b). Fig. 7.b shows the corresponding service curve for each flow. Since the time slot is shared between two nodes, the bandwidth guaranteed for each flow is equal to $R_{TS}/2$, and the latency is equal to $2\cdot BI - Ts$ (see Figs. 6.b and 7.b). As a result, the service curve granted for each flow using round robin is $\beta_{2nodes,1TS}(t) = 4.69\cdot(t-29.76)^{+}$. Now, applying Eq. (7) to each flow F_A and F_B using the per-flow service curve $\beta_{2node,1TS}(t)$, we have: $D_{A,\max} = 72.40\text{ ms}$ and $D_{B,\max} = 115.04\text{ ms}$, and thus condition (C2) is satisfied. Observe that the guaranteed rate of one time slot is higher than the sum of the arrival rates of both flows, i.e. condition (C1) is satisfied. As a result, both flows can be accepted to share the same GTS allocation under round robin scheduling. In this case, the utilization of the GTS is equal to $(r_A + r_B)/R_{TS} = 53\%$ (see Eq. (16)), obviously higher than that in the previous case.

Observe in Fig. 6.c that changing the scheduling policy results in a change of the service curve, even if the guaranteed bandwidth is the same. In Fig. 6.c, the maximum latency is higher than the one with round robin scheduling.

A third request. Now, assume that a third node *C* generating a flow F_C with a traffic specification $F_{spec,C} = (500\text{ bits, 3 kbps, 150 ms})$ wants to allocate a GTS. Like in the previous requests, we compute the per-flow service curve for each node while sharing one time slot using round robin policy $\beta_{3nodes,1TS}(t) = 3.12\cdot(t-45.12)^{+}$ (see Fig. 6.d). The corresponding delay bounds for each of the three flows are: $D_{A,\max} = 109.22\text{ ms} \le 150\text{ ms}$, $D_{B,\max} = 173.32\text{ ms} \ge 150\text{ ms}$, $D_{C,\max} = 205.4\text{ ms} \ge 150\text{ ms}$.

As a consequence, node *C* cannot be admitted to share the same time slot with *A* and *B*, even though the sum of all arrival rates is still lower than the guaranteed bandwidth $(3+2+3 < 9.38)$. Since there are still available resources in the superframe, it is possible to extend the CFP to two time slots and apply the same admission control algorithm to node *C*, but with a service curve $\beta_{3nodes,2TS}(t) = 6.25\cdot(t-28.80)^{+}$ (see Fig. 6.e, Fig. 7.c). The corresponding delay bounds for each of the three flows are: $D_{A,\max} = 60.8\text{ ms} \le 150\text{ ms}$; $D_{B,\max} = 92.8\text{ ms} \le 150\text{ ms}$; $D_{C,\max} = 108.4\text{ ms} \le 150\text{ ms}$. As a consequence, it is possible to meet the delay requirements of the three flows with only two time slots.

Impact of the delay on the utilization. In this latter case, the implicit allocation mechanism saves one time slot compared to an explicit GTS allocation. The bandwidth utilization of the CFP with implicit GTS allocation is then $(r_A + r_B + r_C)/R_{2TS} = 42\%$ (Eq. (16)), whereas in case of an explicit GTS allocation the bandwidth utilization is $(r_A + r_B + r_C)/3R_{TS} = 28\%$ (Eq. (11)). The improvement in terms of utilization depends on the delay requirement. For more relaxed delay requirements, the improvement on utilization is more significant. For example, if the three flows had a delay requirement of 250 ms, it would be possible to allocate only one time slot, resulting in an utilization of $(r_A + r_B + r_C)/R_{1TS} = 85\%$.

What if the guaranteed bandwidth is lower than the arrival rate? Observe that in the previous scenarios, the guaranteed bandwidths offered by a shared GTS using round robin scheduling are higher than the arrival rates of the three flows. Now, imagine that node *C* has an arrival rate equal to 7 kbps. In this case, round robin is not sufficient for flow *C* since the guaranteed rate 6.25 kbps is lower than flow *C*'s arrival rate. A first option is to extend the length of the CFP to have higher bandwidth and compute the corresponding service curve, while still applying round robin. This technique is simple, but it tends to an explicit allocation. Another technique consists in using weighted round robin, by assigning time slots proportionally to the arrival rates, and thus providing differentiated services inside one shared GTS with respect to the arrival rates. Each flow will then have its own service curve with respect to its arrival rate, and the corresponding delay bound would be compared to the delay requirement of the flow, as made previously. This technique is more efficient in terms of utilization, but introduces additional complexity to determine the weights, the schedule and then the corresponding service curves for each flow.

For the sake of simplicity, we consider in this paper the first alternative of extending the CFP length.

4. Schedulability analysis of an implicit GTS allocation under round robin

This section presents a generalization of the practical intuition presented in Section 3.4. Our purpose is to find a general expression of the service curve for *N* flows that share *k* time slots, where $k \le N$ using round robin scheduling, assuming that flows F_i have arrival rates $r_i \le R_{Ts}$ (the most relevant for WSN applications). Note that in this particular case, *k* < 7, since the maximum number of GTSs per superframe is limited to 7. Since we are considering a fair share of a GTS using round robin policy, $\beta_i(t)$ is equal to a rate-latency service curve $\beta_{R,T}(t)$ common to all flows sharing the same GTS. Two distinct cases need to be addressed.

1. **Case of *k* = *N*.** This case is equivalent to an explicit allocation. Each node has its own time slot since round robin is deployed. The delay bound is then computed based on Eq. (7) and compared to D_i.
2. **Case of *k* < *N*.** this case is more interesting because the number of nodes is higher than the allocated time slots, as presented in the example (Section 3.4). Obviously, it can be understood from the motivating example that the guaranteed rate for each flow is:

$$R = \frac{R_{kTS}}{N} = \frac{k}{N} \cdot R_{TS} \qquad (18)$$

The main problem is to compute the service latency related to the service curve. Observe, through the examples of Fig. 6, that the latency can be expressed as:

$$T = p \cdot BI + q \cdot Ts \qquad (19)$$

where $p \in \mathbb{N}$ denotes the number of beacon intervals that contributes to the service latency, and $q \in \mathbb{Z}^{-}$ represents the number of time slots to be subtracted from the latency. For $k \leq 7$ and $N \geq 1$, we have:

$$p = \left\lceil \frac{N}{k} \right\rceil > 0 \text{ and} \qquad (20)$$
$$q = N - p \cdot k - 1 < 0$$

Eq. (20) can be verified with the examples in Fig. 6 as well as with all other combinations of *N* and *k*. As a result, the service curve corresponding to a fair share of *k* time slots between *N* nodes for *k* < *N* under round robin scheduling is:

$$\beta_{R,T}\left(t\right) = \frac{k}{N} \cdot R_{TS} \cdot \left(t - \left(p \cdot BI + q \cdot Ts\right)\right)^{+} \qquad (21)$$

with *p* and *q* are defined in Eq. (20).

So, for a flow F_i with $r_i \leq R$, the corresponding delay bound guaranteed by the fair share, based on Eq. (4), is:

$$D_{i,\max} = N \cdot \frac{b_i}{k \cdot R_{TS}} + \left(p \cdot BI + q \cdot Ts\right) \qquad (22)$$

Observe that Eqs. (21) and (22) are general expressions that are also valid in case of $k = N$ (in this case, *p* = 1 and *q* = -1 and Eqs. (21) and (22) are equivalent to Eqs. (5-6) and (7) (with *n* = 1), respectively).

In summary, a set of *N* flows (F_i, *i* = 1..*N*) sharing a number of *k* time slots where $k \leq N$ are schedulable under round robin, if, for each flow with $F_{spec,i} = \left(b_i, r_i, D_i\right)$ we have $r_i \leq k \cdot R_{Ts} / N$ (Eq. (15) implies condition (C1) in Eq. (9)), and $D_{i,\max} \leq D_i$ (equivalent to condition (C2) in Eq. (9)) where $D_{i,\max}$ is obtained from Eq. (22).

5. i-GAME implementation approach

5.1 i-GAME Admission Control Algorithm

This section presents an admission control algorithm for the implicit GTS allocation mechanism (i-GAME) presented in Sections 3 and 4. We define the admission control algorithm in case of $r_i \leq R_{Ts}, \forall i = 1..N$ under round robin scheduling (results of Section 4). We assume that flows with $r_i \geq R_{Ts}$ explicitly request a number of time slots based on their arrival rates, using Eq. (12).

```
    i-GAME Management Algorithm
1   type Flow = (id, b, r, D) //traffic specification and delay requirement
2   type FlowSetType = (Fi , where Fi requests a time slot in the CFP)
3   int N = 0; // the number of flow sharing a GTS
4   int k = 1; // the number of shared time slot
5   FlowSetType FlowSet; Flow F;
6   On (arrival of a new flow F) do {
7     N = N + 1;
8       if (admission_control (k, N, FlowSet, F) == false) {
9           if (k == 7) { //the maximum number of GTSs is reached
10              reject_request(F);
11              N = N - 1;   break;
12          }
13          else {    // k < 7
14              k = k + 1; //increase the length of the CFP
15              goto line 8;
16           }
17      }
18      else {
19          accept_request(F); //accept the new flow to share the GTS
20          FlowSet_Add(FlowSet, F); //add the new flow to the GTSset
21      }
```

Fig. 8. i-GAME Management Algorithm

Fig. 8 presents the i-GAME *management algorithm* for implicit GTS allocations under round robin scheduling.

Fig. 9 presents the *admission control function* used to decide whether to accept or not a node requesting an implicit allocation of a GTS based on its $F_{spec} = \left(b, r, D\right)$.

GTS Management Algorithm. When a new implicit GTS allocation request initiated by a flow *F* = (*b*, *r*, *D*) is received by the PAN coordinator, the admission control algorithm increments *N*, i.e. the number of flows sharing the same GTS. Then, the admission control function is called taking as inputs the number of allocated time slots (*k*), the number of flows sharing this GTS of *k* time slots (*N*), the set of flows sharing the GTS (*FlowSet*), and the new flow *F* requesting the GTS allocation.

If the admission control function returns *false*, then the PAN coordinator tries to extend the CFP length by adding a new time slot, if the maximum length of seven time slots has not been reached (since, in this case, each node allocates at most one time slot in a superframe); otherwise the new request is rejected. If a new time slot can be added to the CFP, then *k* is incremented by one and the admission control algorithm is called again with the new *k* value.

If the admission control returns *true* , the request of the new flow *F* will be accepted and the latter is added to the *FlowSet* list.

The admission control function. This function returns a Boolean value stating whether to accept or not the new flow requesting a GTS. As we have mentioned before, we assume that flows requesting implicit allocation satisfies $r_i \leq R_{Ts}, \forall i = 1..N$ (for the validity of Eqs. (16) to (22)). This decision is based on the shared GTS length (*k*), the number of flows sharing the GTS (*N*), the specification of the new flow *F* and the existing flows in the *FlowSet*. The *adm_crt* flag is set to *true* at the start of the algorithm, and will be set to *false* if the delay requirement cannot be met or if the guaranteed bandwidth is higher than the arrival rate. The delay requirement of each flow will be compared to the guaranteed delay expressed in Eq. (22), which is shown to be valid for both cases $k < N$ and $k = N$. Actually, the case $k > N$ is not considered in the i-GAME admission control function, since it is considered as an explicit GTS allocation request, as we have previously mentioned.

```
i-GAME Admission Control Function
$R_{TS}$ = guaranteed bandwidth by one time slot
Ts = time slot duration
boolean admission_control (int k, int N, FlowSetType FlowSet,
                                                      Flow F)
{
    boolean adm_crt = true;
    if (k <= N) {
        p = ceil (N / k);
        q = N – p * k – 1;
        for (int i = 1, i++; i<=N)
            if (( Di < (( bi / (k * $R_{TS}$ / N)) + ( p * Bi –  q * Ts))) or
.                                           (ri>k*$R_{TS}$/N))
                adm_crt = false;
    } else //the case (k>N) is considered as explicit allocation
        adm_crt = false;
}
```

Fig. 9. i-GAME Admission Control Function

5.2 i-GAME implementation guidelines

This section presents some practical considerations for the implementation of the i-GAME mechanism in IEEE 802.15.4. An interesting feature of i-GAME is that its implementation only requires minor add-ons to the standard protocol, i.e. it does not impose any changes to the existing protocol.

The idea consists in using the reserved 6th bit in the GTS characteristics frame, embedded in a GTS allocation request command field (compare Fig. 10 and Fig. 2). This bit is referred to as *Allocation Type*.

bits	0-3	4	5	6	7
	GTS Length	GTS Direction	Characteristics Type	Allocation Type	Reserved

Fig. 10. GTS Characteristics Extension Field Format for Implicit Request Allocation

If the *Allocation Type* bit is set to 0 it refers to an explicit GTS allocation. In this case, the allocation process will follow the standard recommendations. If it is set to 1, it refers to the i-GAME implicit allocation mechanism proposed in this paper. In this case, to keep the IEEE 802.15.4 with no changes, the flow specification information $F_{spec} = (b, r, D)$ should be embedded in the higher layer packets, as presented in Fig. 11.

bits	0-3	4-7	8-12	12-15
	Burst size	Arrival Rate	Delay Requirement	Reserved

Fig. 11. Flow Specification Field Format for i-GAME

The admission control algorithm should be implemented at a higher layer (e.g. Network Layer) and should return the decision to the MAC sublayer (Fig. 12).

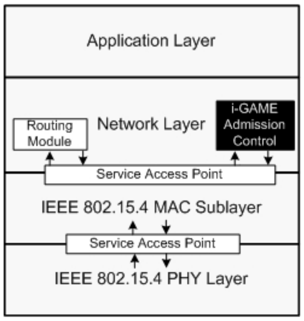


Fig. 12. Protocol Layer Architecture for i-GAME

Hence, upon reception of an implicit GTS allocation request (*Allocation Type* = 1), the MAC sublayer of the PAN coordinator should forward the traffic specification field (shown in Fig. 11) to the higher layer for processing by the admission control module. The *burst size* and the *arrival rate* fields should be expressed by four bits each (16 classes for each field). The *Delay Requirement* field is expressed by five bits (32 classes). Using this frame format, the PAN coordinator should define a fixed range for each value (class) of the corresponding field. These patterns should be known in advance by all nodes associated to the PAN before initiating an implicit allocation. The specification of these classes and ranges is out of the scope of this paper.

When the flow specification is received by the admission control module, it evaluates the acceptance of the new flow based on the algorithm defined in Section 5.1. The decision should be notified to the MAC sublayer through the service access point. In case of acceptance, the MAC sublayer allocates the time slots in the CFP in round robin order to all accepted nodes. For that purpose, the MAC sublayer should establish a certain order to allocate the time slots according to round robin scheduling. Each beacon frame of a new beacon interval should indicate which nodes are allowed to use the GTS in the current superframe, with respect to the established order.

6. Performance evaluation

The purpose of this section is to illustrate the advantage of i-GAME in improving the bandwidth utilization efficiency as compared to the explicit GTS allocation mechanism.

Consider a set of 14 flows $F_i, \forall 1 \le i \le 14$ with the arrival rates as presented in Table 1.

Table 1. Arrival rates of the flows F1 – F 14

Flow	Arrival rates r_i (kbits/sec)
F1, F12	0.5
F2, F5, F6	1
F3, F4, F8	1.25
F7, F9, F11	0.25
F10	0.1
F13	0.3
F14	0.2

Since the guaranteed delay bound typically depends on the burst size, we assume (without loss of generality) that all flows have the same burst size $b_i = 200 \text{ bits}, \forall 1 \le i \le 14$.

We consider a PAN with the same parameters as in Section 3.4 ($BO = SO = 0$).

Consider also the following three cases:

1. Explicit GTS allocations for 7 flows (F1 to F7)
2. Implicit GTS allocations for 7 flows (F1 to F7)
3. Implicit GTS allocations for 14 flows (F1 to F14)

Note that condition (C1) is satisfied in all cases ($\sum_{i=1}^{7} r_i = 6.25 \text{ kbps}, \sum_{i=1}^{14} r_i = 9.1 \text{ kbps}$). Figs. 13 and 14 present the bandwidth utilization and the guaranteed delay bound, respectively, as a function of the number of allocated time slots for the implicit GTS allocations. The bandwidth utilization of the explicit allocation (9.5%) is obtained from Eq. (11) and represented by a dotted line for a comparison purpose. The bandwidth utilization of implicit allocations is obtained from Eq. (16).

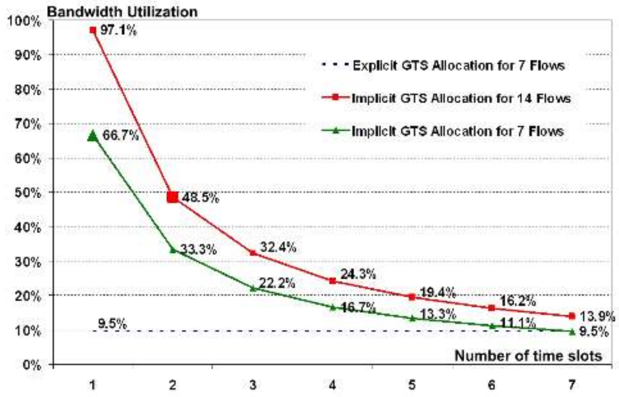


Fig. 13. Bandwidth Utilization Improvement with i-GAME

It can be understood from Fig. 13 that the i-GAME approach significantly improves the bandwidth utilization compared to the explicit allocation. However, the degree of improvement depends on the delay requirements of the flows as it is observed in Fig. 14.

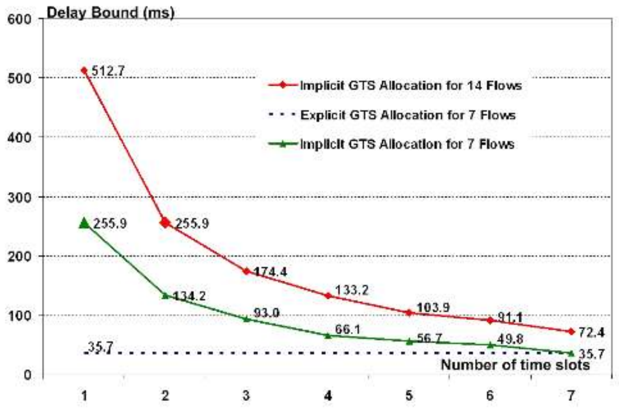


Fig. 14. Delay Bounds Guaranteed by the i-GAME Approach

For example, assume that all flows have a delay requirement D_i = 300 ms. It is possible to meet this requirement for the seven flows (F1 to F7) with only one allocated time slot ($D_{i,\max}$ = 255.9 ms), since $r_i \le R_{TS}/7, \forall 1 \le i \le 7$. In this case, the bandwidth utilization is 66.7%, which is much higher than 9.5% in case of the explicit GTS allocation. It is also possible to meet this delay requirement for the fourteen flows with only two allocated time slots resulting in a utilization of 48.5%. In this case, all the flows take advantage of a guaranteed service with only two allocated time slots, which is not possible using the explicit GTS allocation mechanism. Moreover, by using the implicit GTS allocation, the length of the CFP is significantly reduced, thus increasing the CAP period.

7. Conclusions

This paper improves on the state-of-the-art with the definition of i-GAME, a new approach to allocate GTS in the IEEE 802.15.4 protocol for WSNs. This proposal is motivated by the bandwidth utilization inefficiency of the explicit GTS allocation mechanism supported by the IEEE 802.15.4 protocol for flows with low rates. i-GAME overcomes this problem by allowing to share the same GTS between multiple flows based on their traffic specifications and delay requirements. The performance evaluation study clearly showed the improvement of i-GAME compared with the explicit GTS allocation mechanism in terms of bandwidth utilization efficiency.

Moreover, the implementation of i-GAME only requires minor add-ons to the IEEE 80.15.4 protocol and ensures backward compatibility with the standard, making our approach easily implementable in Commercial-Off-The-Shell (COTS) platforms.

References

[1] IEEE 802.15.4 Standard-2003, "Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)", IEEE-SA Standards Board, 2003.

[2] A. Koubâa, M. Alves, E. Tovar, "GTS Allocation Analysis in IEEE 802.15.4 for Real-Time Wireless Sensor Networks", submitted to WPDRTS 2006, available for download at http://www.dei.isep.ipp.pt/~akoubaa/submissions.htm

[3] S. Cavalieri, S. Monforte, A. Corsaro, G. Scapellato, "Multicycle Polling Scheduling Algorithms for FieldBus Networks" *Journal of Real-Time Systems*, V. 25 (2-3), pp. 157 – 185, Sept.-Oct. 2003.

[4] P. Raja, G. Noubir, "Static and Dynamic Polling Mechanisms for FieldBus Networks", *ACM SIGOPS Operating Systems Review*, V.27 (3), pp. 34-45, July 1993.

[5] A. Koubâa, Y. Q. Song, "Evaluation and Improvement of Response Time Bounds for Real-Time Applications under Non-Preemptive Fixed Priority Scheduling", *International Journal of Production and Research* (IJPR), (Taylor & Francis), V. 42 (14), pp. 2899-2913, July 2004.

[6] A. Koubaa, M. Alves, "A Two-Tiered Architecture for Real-Time Communications in Large-Scale Wireless Sensor Networks: Research Challenges " In Proc. of 17th Euromicro Conference on Real-Time Systems (ECRTS'05), WiP Session, Mallorca (Spain), July, 2005.

[7] J-Y. Leboudec, P. Thiran, *A Theory of Deterministic Queuing Systems for the Internet*, LNCS, Vol. 2050, 2001.