Scheduling Channel Time Allocations in 802.15.3 WPANs for Supporting Multimedia Applications

G. Boggia, P. Camarda, and L. A. Grieco

Abstract

The 802.15.3 standard is a reference point for high data-rate Wireless Personal Area Networks, being able to support also multimedia applications with specific Quality of Service (QoS) requirements. In a 802.15.3 piconet, a single piconet coordinator (PNC) distributes Channel Time Allocations (CTAs) to devices associated to it, in order to provide the expected QoS. The 802.15.3 standard does not specify how this allocation should be done. To bridge this gap, we propose a dynamic CTA scheduling scheme that provides bounded average delays to multimedia applications. The algorithm is based on a control theoretic approach and acts in two steps: (1) each single DEV computes its transmission needs by applying a control algorithm that targets empty transmission queues; (2) the PNC collects requests of devices and allocates CTAs by taking into account the capacity constraints of the wireless channel. The main properties and the tuning rules of the algorithm have been theoretically investigated. Moreover, in order to provide a comprehensive performance evaluation of the proposed approach, realistic scenarios with video, voice, and FTP flows have been simulated using ns-2. Results confirm that our scheduler is able to satisfy the expected delay bounds in a wide range of traffic loads.

A preliminary version of this work appeared as ”‘A Dynamic Bandwidth Allocation Algorithm for Supporting Real-time Applications in 802.15.3 WPANs”’ in Proc. of the Q2SWinet 2007 Workshop, Chania, Greece.

Authors are with the Dipartimento di Elettrotecnica ed Elettronica - Politecnico di Bari. Via Orabona, 4 – 70125 BARI, Italy. (e-mail: {g.boggia, camarda, a.grieco}@poliba.it).

Index Terms

QoS, Wireless PAN, Scheduling

I. INTRODUCTION AND RELATED WORK

The emerging 802.15.3 standard represents a reference point for high speed Wireless Personal Area Networks (WPANs) [1], [2]. It can support multimedia applications with specific Quality of Service (QoS) requirements in small geographic areas [2]. The basic set of a 802.15.3 WPAN is the so called piconet, which is made by a single PicoNet Coordinator (PNC) and one or more devices (DEVs) associated to it. Among other tasks, the PNC has to distribute the Channel Time Allocations (CTAs) to devices (DEVs), i.e., the time interval in which DEVs have the right to transmit. Such a distribution is made in order to fulfill the flow QoS requirements, subject to capacity constraints of the wireless channel, but the 802.15.3 standard does not specify how this allocation should be done. Therefore, the literature has recently proposed many scheduling algorithms and Call Admission Control (CAC) schemes for supporting multimedia applications in high-speed WPANs. It follows a review of the related work.

In [3], a CAC algorithm for high data rate WPANs is proposed for supporting multimedia applications. The admission scheme, which has been referred to as Reservation Based and Revenue Test CAC with Bandwidth Reallocation, is based on bandwidth satisfaction, revenue rate, and bandwidth reallocation cost functions. Main advantages of this scheme include the maximization of the overall bandwidth satisfaction and the admission with guaranteed priority level of multiple class of services with different bandwidth requirements.

In [4], a scheduling mechanism that simultaneously utilizes the multiple channels available in UWB based networks for IEEE 802.15.3 WPANs has been presented. It also employs a distributed dynamic channel allocation algorithm to efficiently allocate channels to neighboring, interfering piconets. Assuming Poisson traffic, it is shown that the simultaneous use of multiple channels increases the throughput and reduces the average packet delay.

In [5], the VBR-MCTA scheduling algorithm is proposed and two optimizations are also discussed, namely VBR-Blind and VBR-TokenBus. The rationale of VBR-MCTA is to allow many flows to share the same CTAs, exploiting the StreamGroupID feature and the new Relinquish command that gives the opportunity to a CTA owner to pass the control of its CTA to another device. By allowing streams belonging to a common StreamGroupID to multiplex packets in each other CTAs, a PNC only needs to allocate CTAs by taking into account the mean rates of the streams. In order to arbitrate the access to unused portions of allocated CTAs, VBR-MCTA collects traffic requirements of devices using MCTAs. With VBR-Blind, devices gain ownership of a unused portion of a CTA in a round-robin manner. VBR-TokenBus, instead, sorts devices using a ranking criterion, such as queue length, and the device with the highest rank is then given ownership. After this device has finished, the remaining time in the CTA is passed to the next device in the sorted list. Using computer simulations, the paper demonstrates that the proposed algorithms offer a real-time service in scenarios with less than 10 nodes.

In [6], a novel MAC protocol for high data-rate WPANs has been proposed to maximize the throughput and minimize delays in presence of multimedia traffic. The key feature of the approach is to choose in an adaptive way the amount of time that should be assigned to each type of traffic. The effectiveness of the proposed algorithm has been demonstrated in network scenarios composed by 4 nodes hosting CBR and WWW traffic sources.

In [7], a dynamic channel time allocation algorithm, called feedback-assisted channel time allocation (FACTA), is reported with the goal of providing delay guarantees to MPEG video streams in the high-rate WPANs. FACTA requires modifications to the standard superframe structure in order to transmit at the end of a superframe feedback control packets of short duration that deliver dynamic parameters for channel time requests from DEVs to the PNC. FACTA outperforms an analogous algorithm proposed in [8] in network scenarios with MPEG-4 video flows for several load levels.

In [9], assuming that a number c of CTAs are allocated for current transmissions, a queuing model $M/M/c$ is used to design a scheduling scheme for both reducing the average waiting for new devices and meeting the QoS requirements of multimedia flows. No simulations including multimedia traffic are reported to assess the effectiveness of the scheme in realistic conditions.

In [10] a link-layer scheduling algorithm has been designed using a control theoretic approach. The algorithm distributes CTAs exploiting transmission queue lengths of DEVs as feedback signal. Network induced delays in the control system have not been considered. The performance of the algorithm have been evaluated only in presence of MPEG Video traffic.

The works listed above have some drawbacks that justify our proposal and the research about scheduling in 802.15.3 networks. In particular, the main lacks of existing schemes are that: no explicit mention to expected delay bounds is provided [3]-[6], [10]; unrealistic traffic models have been used [4], [9]; modifications to the 802.15.3 standard are required [7], [8]; only network scenarios with homogeneous multimedia traffic sources have been considered [4], [7], [9], [10].

To overcome these drawbacks, the present work proposes a new dynamic CTA scheduling scheme, fully compliant with the 802.15.3 standard, that aims to provide bounded average delays to multimedia applications. The algorithm is based on a control theoretic approach and acts in two steps: (1) each single DEV computes its transmission needs (i.e., CTAs) by taking into account its transmission queue length; (2) the PNC collects requests of DEVs and allocates CTAs by taking into account the capacity constraints of the wireless channel. The main properties and the tuning rules of the proposed algorithm have been theoretically investigated. Moreover, in order to provide a comprehensive performance evaluation of the proposed approach, realistic scenarios with video, voice, and FTP flows have been simulated using *ns*-2, confirming that our scheduler is able to provide services with bounded average delays to multimedia applications in a wide range of traffic conditions.

The rest of the paper is organized as follows: Section II gives an overview of the 802.15.3 Medium Access Control (MAC); in Section III, our CTA scheduling algorithm is presented; Section IV shows simulation results; finally, last Section draws the conclusions.

II. Overview of the 802.15.3 MAC

IEEE 802.15.3 defines physical layer and MAC specifications for a high data rate WPAN [2]. The basic set of a 802.15.3 network is the so called piconet, which is a wireless ad hoc data communication system that allows a number of independent data devices (DEVs) to communicate to each other, by following a peer-to-peer paradigm. Both isochronous and asynchronous data transfer are supported. Isochronous data transfer is used to support flows with specific QoS requirements, whereas asynchronous transfer is used for flows without QoS requirements. One DEV of the piconet is required to assume the role of piconet coordinator (i.e., the PNC) that is responsible for timing, traffic scheduling, and QoS management. Time is seen as an endless sequence of superframes (Fig. 1), each one consisting of three parts: a beacon, an optional Contention Access Period (CAP) and a Channel Time Allocation period (CTAP). The beacon frame, transmitted by the PNC at the beginning of each superframe, is used to broadcast management and scheduling information. The CAP is used for transmission of commands and asynchronous data (i.e., traffic with no specific QoS requirements); during this interval channel is accessed by using the well-known Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) scheme [1]. The CTAP is used for both asynchronous and isochronous data; it is made of Channel Time Allocations (CTAs) used by DEVs for transmitting their data and of optional Management Channel Time Allocations (MCTA) for command frames. In each superframe, during the CAP, devices request CTAs for the next superframe to the PNC which is responsible for their allocation.

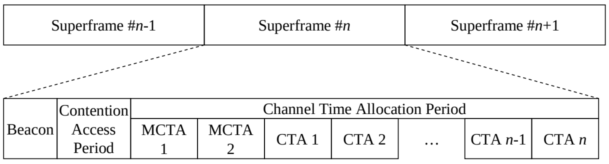


Fig. 1. 802.15.3 superframe structure.

All data in a 802.15.3 piconet are exchanged following a peer-to-peer paradigm. There are two methods to transfer data among DEVs: sending small amounts of asynchronous data in the CAP, if available, or using CTAs in the CTAP for isochronous or asynchronous data. If a DEV needs to transmit on a regular time basis, an isochronous channel time is requested (Fig. 2); note that, in particular, this kind of channel time allocation is well suited when real-time bounded delay services are required. In details, for CTA creation the DEV sends a Channel Time Request command (CTRq) to the PNC during the CAP, declaring the minimum and the desired channel time in term of number of time units (TUs). The TU length is calculated according to the frame length, the channel bandwidth and the ACK policy adopted. The PNC responds with a Channel Time Response command and, if channel time is available, it assigns to the DEV a number of TUs for the stream. Finally, the PNC broadcasts all the CTA allocations in the next beacon frame, specifying their starting time (an offset from the start of the beacon) and duration. If the bandwidth requirements change, the DEV sends a new CTRq message. Isochronous streams do not expire until they are terminated by either the source DEV, or the destination DEV, or the PNC.

Unlike isochronous streams, for an asynchronous data transfer the channel time allocation is requested for the total amount of time needed by the data transfer itself. This kind of request will be scheduled whenever the PNC is able to fit it in a superframe, according to a best effort approach.

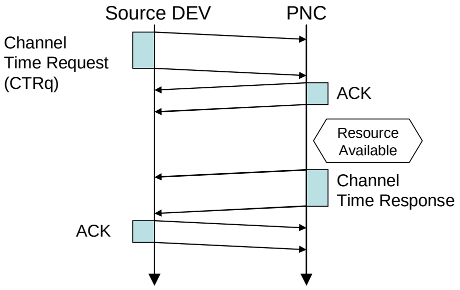


Fig. 2. Message sequence chart for creating and modifying an isochronous stream CTA.

III. SCHEDULING CTAS

In this section, our new scheduling algorithm for providing services with bounded average delays to isochronous streams is designed. Similarly to [11], the classic discrete-time feedback control theory has been employed. The algorithm is based on on the possibility of DEVs to explicitly request CTAs to the PNC. In particular, during each CAP, the PNC: (1) collects CTRq packets from DEVs, containing CTA requests for the next superframe; (2) computes the CTA allocations for the next superframe by taking into account capacity constraints. These allocations are then advertised to all DEVs at the beginning of the next superframe.

A. Superframe formation

We refer to a piconet system made of a PNC and a set of DEVs, each one with a queue fed by a stream (isochronous or asynchronous). The superframe structure is composed by the beacon, the CAP, and a set of CTAs. At the beginning of each CAP, devices hosting an application with QoS requirements send CTRq commands to the PNC, asking for the CTA duration they need. If a DEV is not able to send the CTRq frame due to the collisions occurred during the CAP (remember that during the CAP there is a contention based access), the PNC will consider the last request successfully received from the considered DEV.

In any case, at the end of the CAP, the PNC has information to estimate the total amount of time requested by all the isochronous streams. If this value does not exceed the established maximum duration for the CTAP (which is a system variable), the PNC satisfies all requests and can allocate CTAs for asynchronous flows in the remaining time. Otherwise (i.e., if the maximum CTAP duration is not enough for all the requests), the PNC reduces channel time requests to fit them in the CTAP. As a consequence, there will be no resources for asynchronous traffic in the channel time allocation period.

Finally, at the beginning of the next superframe, the PNC broadcasts the beacon frame announcing the CAP duration and every CTA offset and length.

For the above mentioned procedure, the whole superframe duration depends mainly on the network load and traffic type handled by DEVs. In particular, in the presence of asynchronous traffic the superframe length corresponds to its maximum value (Fig. 3a). If there are only real-time flows, the superframe length depends on the CTA requirements of the DEVs (Fig. 3b); also in this case, when the traffic load increases, the superframe duration tends to its maximum (Fig. 3c).

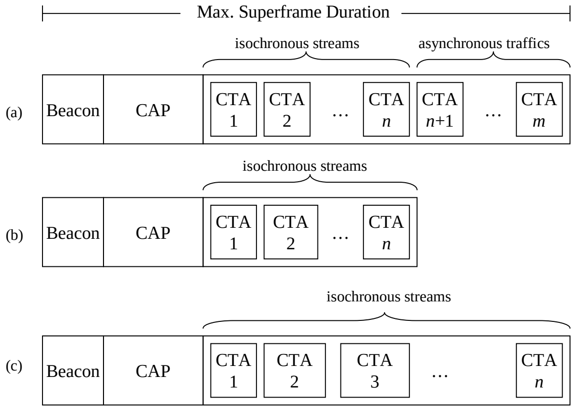


Fig. 3. CTA structure and superframe length.

B. The queue model

In the following, we assume that the time between two consecutive CTAs allocated to the same queue is equal to the superframe duration T_{SF}. The following discrete time linear model can be used to describe the dynamics on the i^{th} queue:

$$q_i(n+1) = q_i(n) + d_i(n) \cdot T_{SF} + u_i(n) \cdot T_{SF} \tag{1}$$

where, referring to the n^{th} superframe, $q_i(n) \geq 0$ is the queue level at the beginning; $u_i(n) \leq 0$ is the average depletion rate (i.e. its absolute value $|u_i(n)|$ represents the bandwidth assigned to the queue); $d_i(n) \geq 0$ is the average input rate at the queue. Note that the input $d_i(n)$ is unpredictable since it depends on the behavior of the source that feeds the i^{th} queue. So that, from a control theoretic perspective, $d_i(n)$ can be modeled as a disturbance [11]. Without loss of generality, the following piece-wise constant model for the disturbance $d_i(n)$ can be assumed:

$$d_i(n) = \sum_{j=0}^{+\infty} d_{0j} \cdot 1(n - t_j) \tag{2}$$

where $1(n)$ is the unitary step function, $d_{0j} \in \mathbb{R}$, and t_j is a time lag [11].

Due to the last assumption given by eq. (2), the linearity of the system described by eq. (1), and the superposition principle that holds for linear systems, we can design the feedback control law by considering only a step disturbance: $d_i(n) = d_0 \cdot 1(n)$. In particular, our design is articulated in three phases: (1) a bound on the steady state queuing delay τ_i^T is imposed, assuming that a step disturbance is applied to the system; (2) a proof that such a bound can be applied to the average queuing delay in a general case is provided; (3) the bound on average delay is increased by T_{SF} to take into account the burstiness of data sources.

C. The closed loop control scheme

In order to satisfy our goal, we consider the closed loop control system shown in Fig. 4, where the set point q_i^T is equal to zero (i.e., we would ideally target empty queues). Regarding the transfer function $G_i(z)$ of the controller, we focus on a very simple proportional controller $G_i(z) = k_{pi}$.

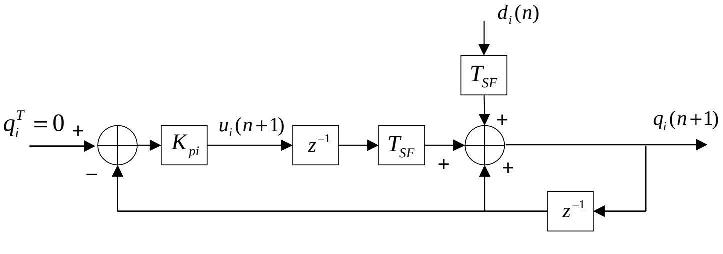


Fig. 4. Closed-loop control scheme.

In this scheme, each DEV computes its bandwidth needs $|u_i(n+1)|$ in the $(n+1)^{th}$ superframe, proportional to the queue level at the beginning of the CAP of the n^{th} superframe. The bandwidth

$|u_i(n+1)|$ is translated in a CTA request that is delivered to the PNC during the CAP (details about the CTA computation will be provided below).

Proposition 1: The system reported in Fig. 4, where $G_i(z) = k_{pi}$, is asymptomatically stable if and only if the following inequality holds:

$$0 < k_{pi} < 1/T_{FS}. \tag{3}$$

Proof: By considering the control scheme in Fig. 4, it is straightforward to find the $\mathcal{Z}$-transforms of $q_i(n)$ and $u_i(n)$:

$$\begin{aligned} Q_i(z) &= \frac{z \cdot T_{SF}}{z^2 - z + k_{pi} \cdot T_{SF}} \cdot D_i(z); \\ U_i(z) &= -\frac{k_{pi} \cdot T_{SF}}{z^2 - z + k_{pi} \cdot T_{SF}} \cdot D_i(z) \end{aligned} \tag{4}$$

where $D_i(z) = \mathcal{Z}[d_i(n)]$.

From eqs. (4), it results that the system poles are $z_p = \frac{1 \pm \sqrt{1-4k_{pi} \cdot T_{SF}}}{2}$; thus, the system is asymptotically stable if and only if $|z_p| < 1$, that is: $0 < k_{pi} < 1/T_{SF} \quad \square$.

.

Proposition 2: By considering the system reported in Fig. 4, where $d_i(n) = d_0 \cdot 1(n)$ and $G_i(z) = k_{p_i}$, the following inequality has to be satisfied in order to achieve a steady-state delay smaller than the target delay τ_i^T:

$$k_{pi} \geq 1/\tau_i^T. \tag{5}$$

Proof: By considering the $\mathcal{Z}$-transform $D_i(z) = d_0 \cdot \frac{z}{z-1}$ of the step function $d_i(n) = d_0 \cdot 1(n)$, if we apply the final value theorem [12] to Eqs. (4), it results:

$$u_i(\infty) = \lim_{n \to +\infty} u_i(n) = \lim_{z \to 1} (z-1) U_i(z) = -d_0;$$

$$q_i(\infty) = d_0 / k_{pi},$$

which implies that the steady state queueing delay is:

$$\tau_i(\infty) = |q_i(\infty)/u_i(\infty)| = 1/k_{pi}$$

The proof is derived by imposing $\tau_i(\infty) \leq \tau_i^T \quad \Box$.

.

Remark 1: It is worth noting that $q(\infty) > 0$ even if $q_i^T = 0$, which means that the proportional controller is not able to fully reject the step disturbance $d_0 \cdot 1(n)$.

Remark 2: From inequalities (3) and (5), the T_{SF} parameter must satisfy the following constraint:

$$T_{SF} < \min_{i=1..M} \tau_i^T. \tag{6}$$

Remark 3: From propositions 1 and 2 it turns out that the gain k_{pi} can vary in the range $\left[\frac{1}{\tau_i^T}, \frac{1}{T_{SF}}\right[$.

In the following, we will set k_{pi} at its lowest admissible value $1/\tau_i^T$, thus allocating the lowest bandwidth that guarantees the target delay. In this way, a cautious usage of the WPAN channel is achieved.

D. Average delays

Our control loop allows to obtain a steady-state delay less that τ_i^T when a step disturbance is applied to the system. In the general case, using the Little law [13] the average delay is given by

$$\tau_{avg} = E[q_i]/E[d_i] \tag{7}$$

where $E[q_i]$ and $E[d_i]$ are the average queue length and the average delay, respectively.

That is, assuming the processes as ergodic,

$$\tau_{avg} = \frac{\lim_{N\to\infty} \frac{\sum_{k=0}^{N} q_i(k)}{N}}{\lim_{N\to\infty} \frac{\sum_{k=0}^{N} d_i(k)}{N}} = \frac{Q_i(1)}{D_i(1)} = \frac{1}{k_{pi}} \leq \tau_i^T \tag{8}$$

The bound given by eq. (8) has to be increased by T_{SF} to take into account the behavior of bursty data sources. In fact, for each burst of data, there is a transient before the algorithm can allocate CTAs. This transient has not been taken into account in the proposed fluid model. In the best case, the burst is transmitted as soon as it arrives because a previous CTA allocation is available. In the worst case (see Fig. 5), the transient lasts up to $2T_{SF}$. Thus, the average transient duration can be estimated as T_{SF} and the resulting bound on average queuing delay is $\tau_i^T + T_{SF}$

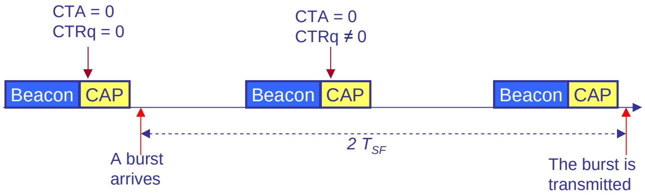


Fig. 5. Transient in the worst case of a CTA allocation.

Therefore, the proposed bandwidth allocation algorithm provides a strict bound on average delays. Note that this bound is derived assuming that there are enough resources to allocate CTAs to all flows in the WPAN. In the next section, the impact of the network load on network performance will be investigated using ns-2 simulations.

E. Assignment of Isochronous CTAs

As seen before, a DEV sends a CTRq command to the PNC during the CAP to communicate its channel time needed in the next superframe. Now, we show how to transform the bandwidth $|u_i|$ calculated with the proposed algorithm into a CTA. Let us suppose that our piconet is composed by a PNC and a set of M_I devices, each one involved with the transmission of one isochronous flow (i.e., one real-time flow). During the CAP of the n^{th} superframe, the i^{th} DEV sends a CTRq requiring the number of time units $N_{TU_i}(n+1)$ needed (up to 256 according to the standard [2]) and specifying their duration $T_{U_i}(n+1)$, up to 65535 μs [2]. Hence, the duration of the requested CTA for the i^{th} real-time flow in the $(n+1)^{th}$ is:

$$CTA_{rt_i}(n+1) = N_{TU_i}(n+1) \cdot T_{U_i}(n+1). \tag{9}$$

As the 802.15.3 standard suggests, $T_{U_i}(n)$ may be evaluated by considering the sum of the frame transmission time, the ACK transmission time (if the acknowledgment is sent), and the appropriate inter frame spaces. Hence, $N_{TU_i}(n)$ should be equal to the number of frames that

DEV wants to send in the next superframe. Nevertheless, the DEV is free to exploit each assigned CTA as a continuous channel time allocation. In our scheme, the DEV estimation of the number of frames to be transmitted during the $(n+1)^{th}$ superframe is given by [1]:

$$N_{TU_i}(n+1) = \left\lceil \frac{|u_i(n+1)| \cdot \overline{T_{SF}}(n)}{\overline{MSDU_i}} \right\rceil \tag{10}$$

where $\overline{MSDU_i}$ is the mean length of the MAC payload (i.e., the Mac Service Data Unit) of the i^{th} source and $\overline{T_{SF}}(n)$ is the average superframe length.

$\overline{T_{SF}}(n)$ is estimated by exploiting the following moving average filter:

$$\overline{T_{SF}}(n) = \alpha \cdot \overline{T_{SF}}(n-1) + (1-\alpha) \cdot T_{SF}(n), 0 < \alpha < 1 \tag{11}$$

where $T_{SF}(n)$ is the duration of the n^{th} superframe, measured by each DEV as the interarrival time between two consecutive beacon frame.

Finally, the duration of the CAP in terms of time units for the i^{th} stream is calculated considering the average value $\overline{MSDU_i}$ which is invariant with respect to n:

$$T_{U_i} = \left(\overline{MSDU_i} + FCS\right)/C_i + H \tag{12}$$

where FCS is the length of the frame check sequence; C_i is the transmission data rate; H is the time overhead that takes into account the time spent to send the frame headers (independent on the data rate) and the ACKs according to the policy adopted by the stream.

If the estimated CTA duration is not enough to send the longest frame that the source generates, only one time unit is requested by the DEV. In this case, its duration is calculated from eq. (12) considering the maximum length of the frame payload expected from the source itself, instead of its average value. This ensures to DEVs the ability to send at least one frame of maximum duration.

---

[1] $\lceil x \rceil$ is the rounding up of x.

F. Handling channel saturation

At the beginning of the n^{th} superframe, the PNC checks if the total amount of channel time requested by the M_I DEVs exceeds the maximum CTAP duration $CTAP_{max}$, i.e., if $\sum_{i=1}^{M_I} CTA_{rt_i}(n) \ > \ CTAP_{max}$

If this happens for a given n_0, then the PNC recalculates CTA assignments reducing each CTA_{rt_i} proportionally. To this aim, first of all the exceeding time $\Delta(n_0)$ is evaluated as:

$$\Delta(n_0) = \sum_{i=1}^{M_I} CTA_{rt_i}(n_0) - CTAP_{max}, \tag{13}$$

and then it is divided among the M_I real-time streams proportionally to their channel time requests:

$$\Delta_i(n_0) = \Delta(n_0) \cdot CTA_{rt_i}(n_0) \Bigg/ \sum_{i=1}^{M_I} CTA_i^{rt}(n_0) \ . \tag{14}$$

The value $\Delta_i(n_0)$ is converted into a number of time units:

$$\Delta TU_i(n_0) = \left\lceil \frac{\Delta_i(n_0)}{T_{U_i}} \right\rceil \tag{15}$$

and the effective number of time units $N'_{TU_i}(n_0)$ reassigned to the i^{th} source is reduced by the last quantity, that is:

$$N'_{TU_i}(n_0) = N_{TU_i}(n_0) - \Delta TU_i(n_0). \tag{16}$$

G. Scheduling of Asynchronous Streams

Herein, we describe how to schedule asynchronous streams. Let M_A be the number of DEVs involved with a non real-time traffic, i.e., with an asynchronous stream which requires only a best effort service.

If the total amount of CTA assignment for real-time flows in the n^{th} superframe is less than $CTAP_{max}$, i.e., the channel is not saturated, asynchronous traffic can be scheduled by the PNC in the remaining time $T_{nrt}(n)$ of the CTAP. Obviously, the remaining time $T_{nrt}(n)$ is given by:

$$T_{nrt}(n) = CTAP_{max} - \sum_{i=1}^{M_I} CTA_{rt_i}(n) \tag{17}$$

The PNC evaluates CTA duration for each non real-time traffic splitting uniformly $T_{nrt}(n)$ among all the M_A non real-time flows [2]:

$$CTA_{nrt_i} = \left\lfloor \frac{T_{nrt}(n)}{M_A \cdot T_{U_{nrt}}} \right\rfloor \cdot TU_{nrt}, \quad 1 < i < M_A \tag{18}$$

where $T_{U_{nrt}}$ is the duration in time units evaluated, for all asynchronous flows, as the time needed to send a frame with the maximum payload allowed by 802.15.3.

The minimum CTA_{nrt_i} assignment could be one time unit, therefore it is possible that $T_{nrt}(n)$ evaluated with eq. (17) is not enough to allocate at least one CTA for each asynchronous stream. When this happens, a sufficient number of streams have to be excluded from evaluation in eq. (18), starting from the latest flows registered at the PNC.

IV. PERFORMANCE EVALUATION

In order to evaluate performance of the proposed scheduling algorithm, computer simulations have been carried out considering realistic scenarios with voice, video, and FTP flows. To implement our scheduling algorithm, we have deeply modified the ns-2 [14] basic MAC modules for 802.15.3 developed by Intel [15].

We pay attention to queuing delays experienced by frames in comparison with the target delay τ_i^T imposed by the feedback control law. In particular, the main objective of this section is to demonstrate the validity of the proposed scheduling algorithm in realistic conditions, i.e., also when the restrictive hypotheses we made for ease the theoretical analysis do not hold. These hypothesis refer to: bandwidth limitations; loss of CTA requests due to contention during the CAP; time-varying superframe duration.

Fig. 6 shows the topology of the piconet considered for our tests. We have chosen the data rate of each DEV as 55 Mbps, i.e., the maximum data rate of 802.15.3 standard.

We have considered several scenarios where a 802.15.3 piconet is used for the simultaneous transmission of different kinds of streams, arranged into the CTAP in the following order: a

---

[2] $\lfloor x \rfloor$ is the rounding down of x.

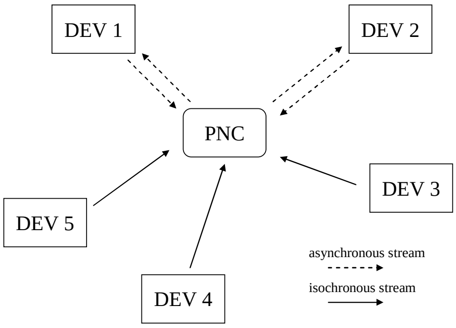


Fig. 6. Piconet topology

number of α DVD-video streams encoded with MPEG2 [16] and AC3 standard [17], α MPEG-4 encoded video flows [18], α H.263 video flows [19], α voice flows encoded with the G.729 standard [20], and 5 FTP asynchronous flows. Therefore, in such a scenario, the traffic load due to multimedia flow is proportional to the parameter α. Unless otherwise specified, CTAs are allocated to the different flows with the following order: DVD, MPEG-4, H.263, G.729, FTP.

MPEG-4 and H.263 video flows are traffic traces available from the video trace library [21]. DVD streams have been extracted directly from real DVD movies. G.729 sources are modeled using Markov ON/OFF sources as in [11].

Each simulation lasts 5 minutes. The target delay τ_i^T has been set equal to 30 ms for voice flows and 40 ms for the other flows; the proportional gain k_{pi} is set equal to $1 \left/ \tau_i^T \right.$. It is important to note that the target delay τ_i^T strongly affects the behavior of the scheduling strategy. In fact, the larger is τ_i^T and the smaller becomes k_{pi}, which, in turn, is equal to the reciprocal of the expected average delay as demonstrated in Eq. (8). Maximum superframe length and CAP duration have been set to 30 ms and 2 ms, respectively. The main characteristics of considered multimedia flows are summarized in Table I.

In the following sections two simulations sets are discussed. The first set involves only multimedia flows, the second one considers also 5 FTP asynchronous traffic sources.

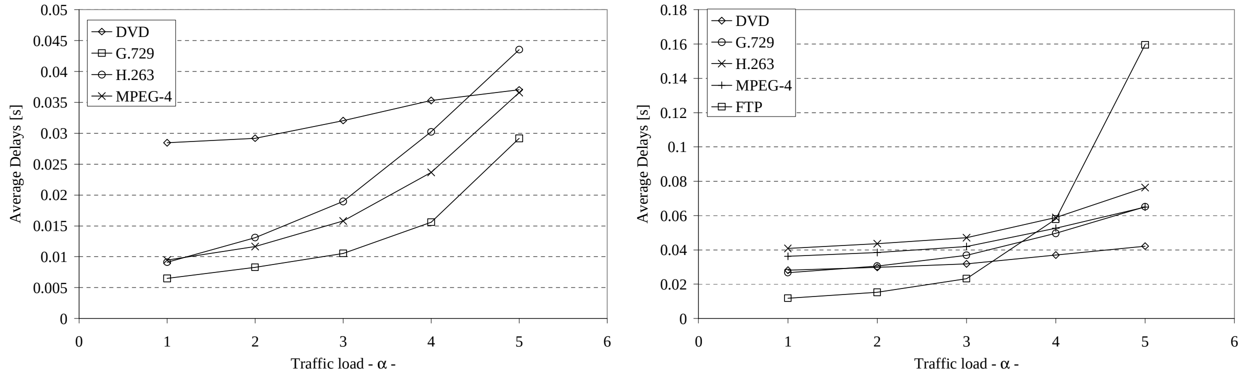
Fig. 7 shows the average one-way packet delays experienced by DVD, MPEG4, H.263, G.729,

TABLE I

MAIN FEATURES OF MULTIMEDIA FLOWS.

Flow Type	*Nominal (Maximum) MSDU Size* [Byte]	*Mean (Maximum) Data Rate* [kbps]
DVD	1336 (1336)	5865 (10080)
MPEG-4	1616 (2044)	770 (3300)
H.263	1634 (2044)	450 (3400)
G.729	60 (60)	13.76 (24)

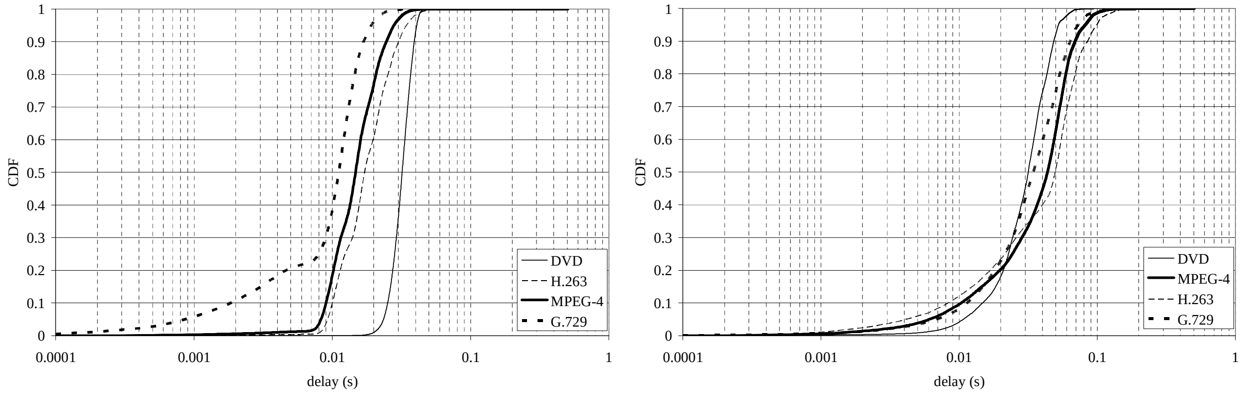
and FTP flows when α varies in the range $[1 \div 5]$. As we can see, average delays increase with the load parameter α, but remain below the bound $\tau_i^T + T_{SF}$, which is equal to 70 ms for DVD and video flows, and 60 ms for voice flows. This clearly demonstrates the effectiveness of the proposed scheduling strategy, also in the presence of a high network load. Moreover, it is straightforward to observe that the average packet delay of DVD flows exhibit a different trend with respect the other kind of flows, when the asyncronous traffic is turned off. This effect disappears when the asynchronous traffic is turned on. This is due to the time-varying superframe size. In particular, DVD flows have the largest bandwidth requirements (see Tab. I). Thus, when the bandwidth ask of DVD flows increases, the superframe size increases too and, viceversa. In this way, when the bandwidth ask of DVD flows is low, the superframe size is small and the all kind of flows are served with a very small packet delay. But in these circumstances, DVD flows has a small number of packet to be transmitted and cannot exploit the performance gain due to a small superframe size. Obviously, when the asynchronous traffic is turned on, these considerations are no more valid, because the superframe size is constant (see Fig. 3). On the other hand, the presence of asynchronous FTP flows worsen the performance of real-time streams. This effect is due to the superframe duration that, now, reaches its maximum value. Anyway, our scheme limits average delays of real-time flows at the expense of FTP ones. This behavior is particularly evident for $\alpha \geq 4$.



(a) (b)

Fig. 7. Average one-way packet delay: (a) scenario without asynchronous flows; (b) scenario with asynchronous flows.

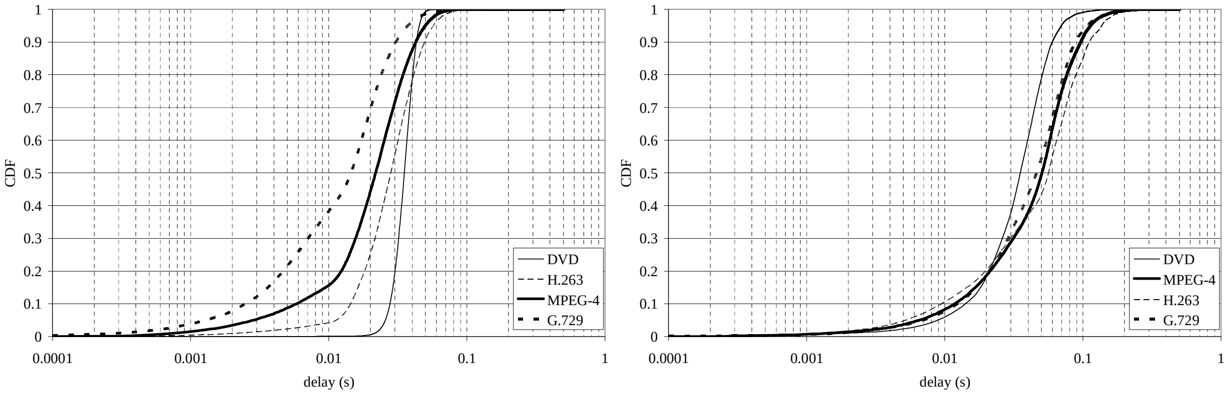
To gain an insight into the behavior of our allocation algorithm, Figs. 8-10 show the cumulative distribution functions (CDFs) of the one-way packet delay. We can observe that, when the asynchronous traffic is absent, also in the scenario with the highest load (i.e., $\alpha = 5$), about 90% of packets experience a delay smaller than 80 ms, which represents a very sharp performance bound. The same bounds grows up tp 150 ms when FTP flows are turned on.



(a) (b)

Fig. 8. CDFs of one-way packet delays with $\alpha = 3$: (a) scenario without asynchronous flows; (b) scenario with asynchronous flows.

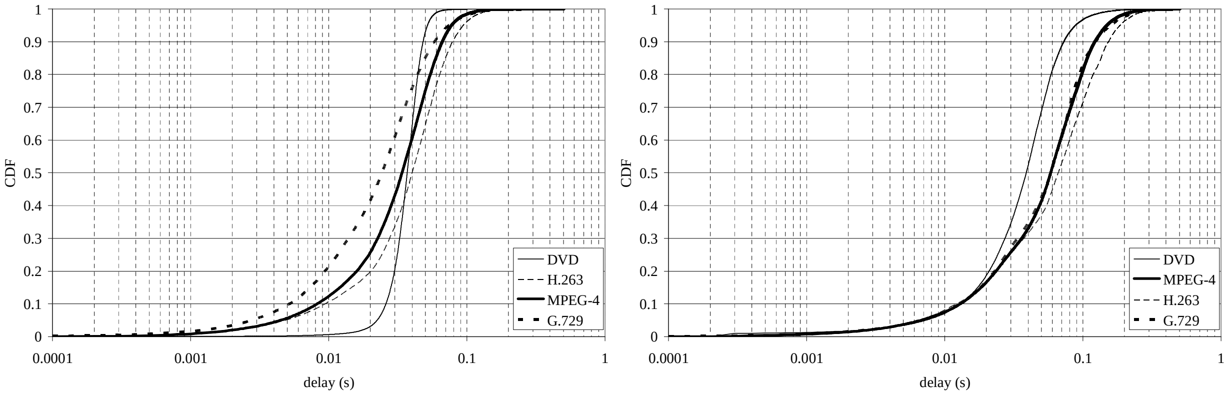
Figs. 11-13 show the CTAP utilization (i.e., the percentage of CTAP used by CTAs) during



(a)

(b)

Fig. 9. CDFs of one-way packet delays with $\alpha = 4$: (a) scenario without asynchronous flows; (b) scenario with asynchronous flows.

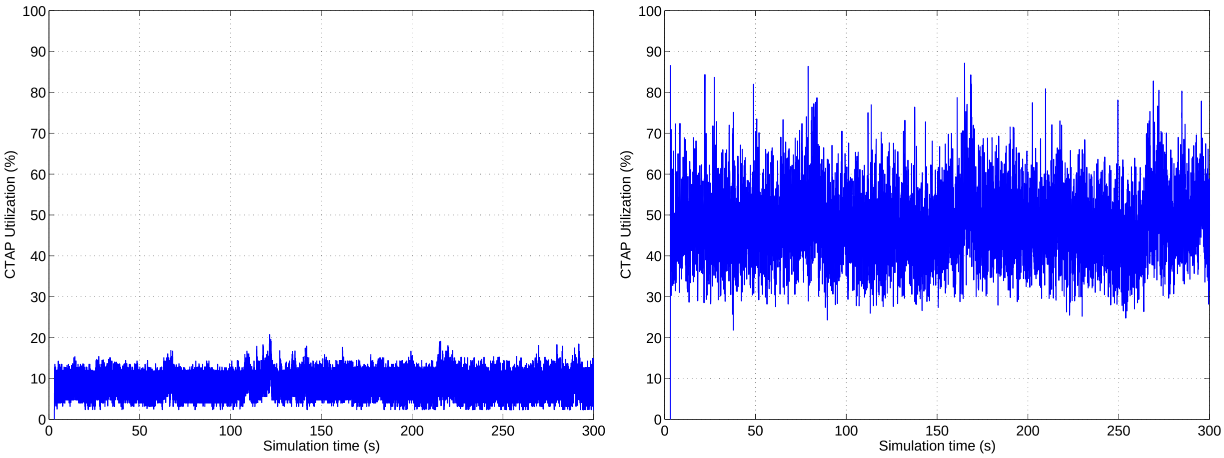


(a)

(b)

Fig. 10. CDFs of one-way packet delays with $\alpha = 5$: (a) scenario without asynchronous flows; (b) scenario with asynchronous flows.

the simulation, defined as the percentage of the channel capacity reserved to isochronous streams and used for serving multimedia flows. When the utilization is very close to 100%, it means that all the bandwidth available for isochronous streams has been allocated. In these conditions, the superframe size grows up to its maximum value as pictured in Fig. 3. Note that, when the asynchronous traffic is turned off, the whole channel time required by DEVs is less than the available one, i.e., utilization is less than 100%, thus, the average superframe duration is quite smaller than its maximum value 30 ms. Such a behavior highlights that to fulfill on average the requirement imposed by τ_i^T, it is not necessary to use all the WPAN bandwidth, i.e., the proposed scheme does not waste channel bandwidth. On the other hand, when FTP flows are active, for $\alpha > 3$, i.e., at high network load, CTAP utilization is often very close to 100%. This effect is due to the superframe duration, which now reachs its maximal value (see Fig. 3), thus making less frequent transmissions from DEVs that host real-time flows. As a consequence, when a DEV obtain its CTA, it has more data to transmit with respect to the case of the scenarios without asyncronous flows. Analyzing simulation results, we noticed that, in these circumnstancies, the channel saturation happens quite often, thus, the PNC has to frequently reduce the CTAs requested by DEVs in order to keep the superframe duration below 30 ms. This, in turn, affect packet delays.



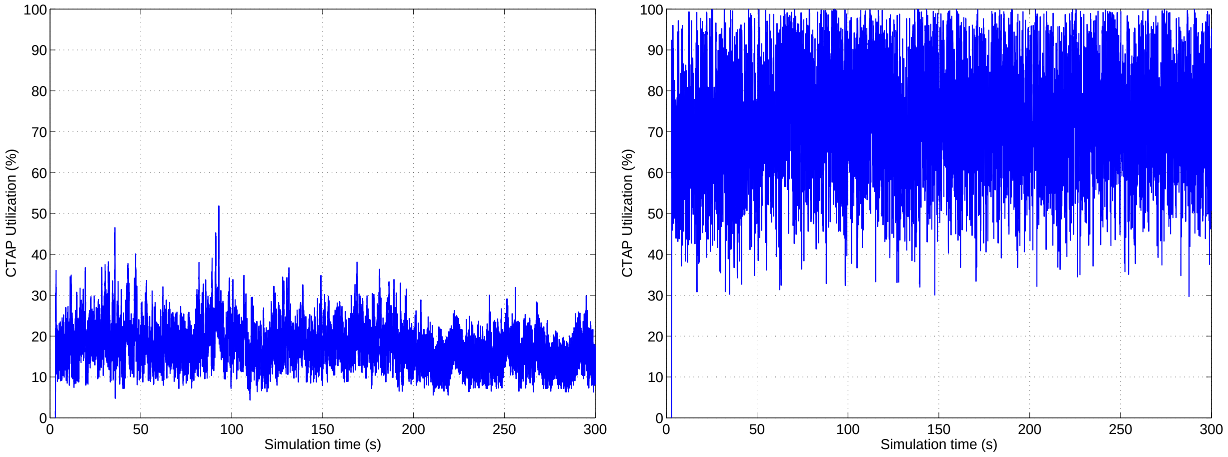
(a)

(b)

Fig. 11. CTAP Utilization when $\alpha = 3$: (a) scenario without asynchronous flows; (b) scenario with asynchronous flows.

A. CAP efficiency

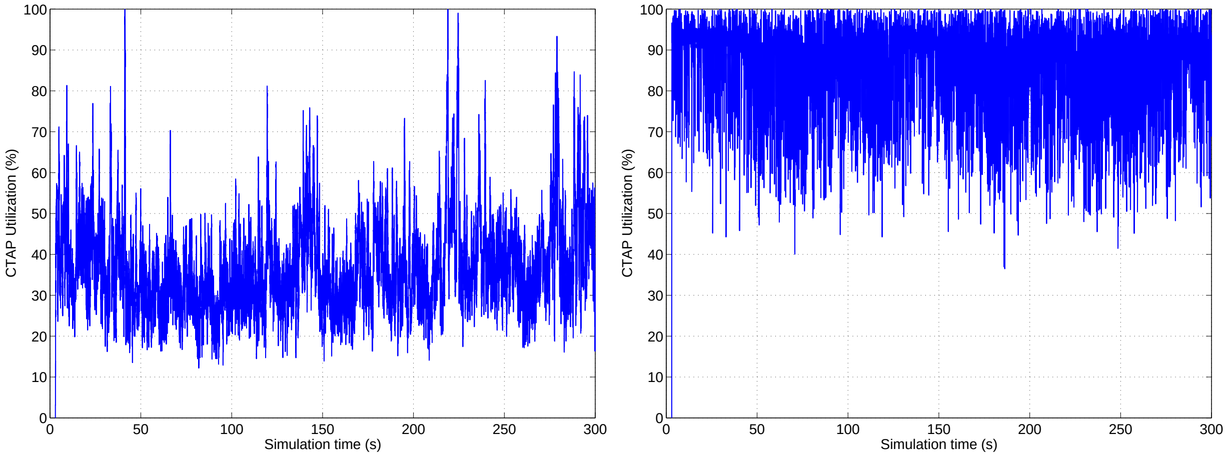
As seen before (Sec. II), channel time request commands are sent by DEVs during the CAP, according to a CSMA/CA mechanism. When the traffic load grows, the number of CTRq increases and, consequently, collisions among them become more frequent. Herein, we analyze



(a)

(b)

Fig. 12. CTAP Utilization when $\alpha = 4$: (a) scenario without asynchronous flows; (b) scenario with asynchronous flows.



(a)

(b)

Fig. 13. CTAP Utilization when $\alpha = 5$: (a) scenario without asynchronous flows; (b) scenario with asynchronous flows.

the behavior of CAP efficiency defined as the ratio between the number of CTRq successfully received by the PNC and the number of new CTRq generated by DEVs. Results are plotted in Fig. 14 for both the scenarios with and without asynchronous flows. It is evident that results are quite similar. This means that the number of CTRq processed by PNC does not depend on the presence of asynchronous streams.

As expected, the efficiency decreases as traffic load increase, given that there are more requests and then more collisions in the CAP. Nevertheless, our scheme is still able to exploit the percentage of CTRq successfully received for properly tuning CTA assignments to DEVs. This is an important point given that the scheduling algorithm has been designed assuming that DEVs send CTRq packets on regularly basis to the PNC. From a control theoretic perspective, this means that the control signal reaches the plant with a constant sampling time. Thus, the demonstration that the algorithm provides the expected performance also in the presence of CTRq collisions is a proof of its robustness to signal under-sampling in the control loop.

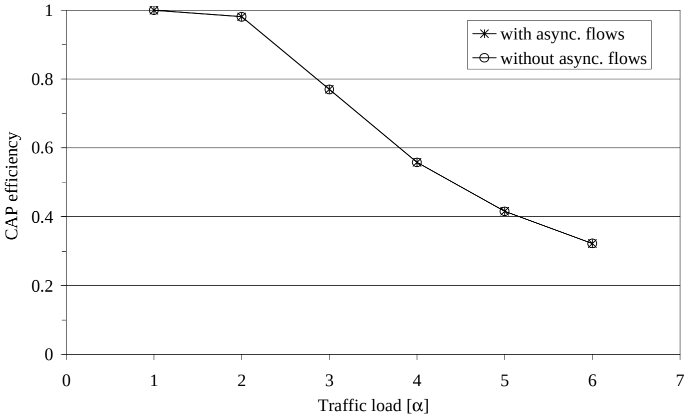


Fig. 14. CAP efficiency.

B. The Impact of the service order

Herein, we evaluate how the order of CTA assignments to different flows affect the performance of the algorithm. For that purpose, we have considered a scenario with 7 DVD flows. We have compared the performance obtained with a fixed order of CTA assignments with respect to the case of a random assignment order, selected at each superframe. Fig. 15 reports the CDFs of packet delays obtained using a fixed (random) CTA allocation order. In both cases, the average packet delay is equal to 43 ms, which is smaller than 70 ms as expected. This means that the service order does not affect the packet delay. The performance improvement we obtain using a random order is a smaller delay variance; in fact, CDFs are closer to each other using a random service order with respect to the case of a fixed one.

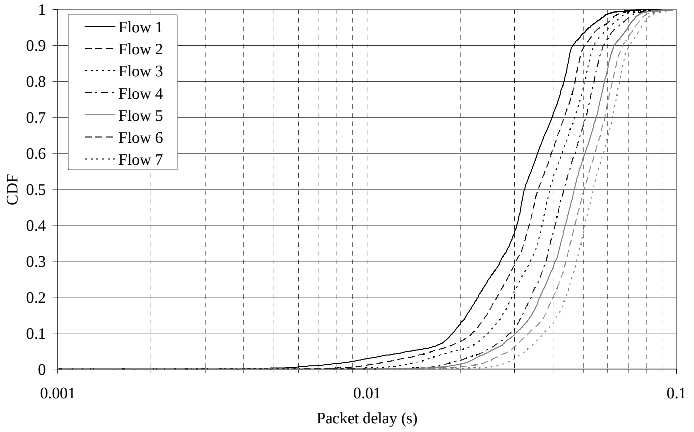


Fig. 15. CDFs of the packet delay of 7 DVD flows served using a fixed order.

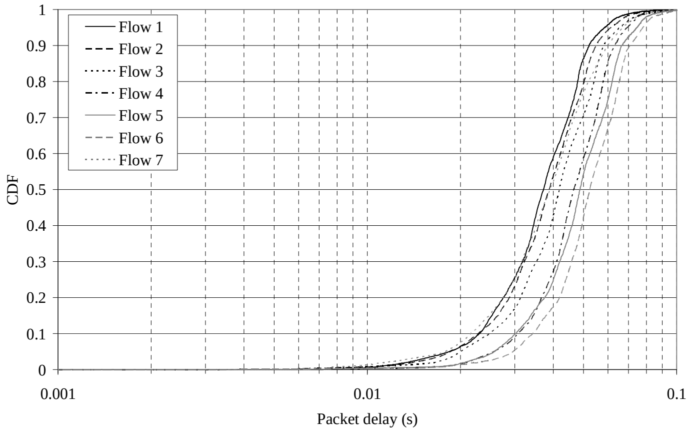


Fig. 16. CDFs of the packet delay of 7 DVD flows served using a random order.

V. CONCLUSION AND FURTHER RESEARCH

In this paper, we have proposed a CTA scheduling algorithm for 802.15.3 WPAN in order to provide a bounded average packet delay to real-time flows. A DEV requests to the PNC the desired number of time units it needs to support its stream; such a value is calculated using classic discrete-time feedback control theory. The PNC allocates CTAs to DEVs dynamically, according to channel condition. We have reported the theoretical analysis of the proposed scheme illustrating how to tune the algorithm. Moreover, the effectiveness of our approach has been validated by extensive computer simulations in several scenarios with different real-time and FTP flows. Further research will compare the performance of the proposed algorithm with respect to other recent proposal of the scientific community.

VI. ACKNOWLEDGMENTS

Authors want to thank G. Tomasicchio and P. Mallen for their valuable contribution in the development of the simulator.

This work was partially funded by Apulia Region's Projects PE_082 and PS_121.

REFERENCES

[1] B. H. Walke, S. Mangold, and L. Berlemann, *IEEE 802 Wireless Systems*. John Wiley and Sons, 2006.

[2] IEEE 802.15.3, *Part 15.3: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Personal Area Networks (WPANs)*, IEEE Std., 2003.

[3] X. Zhai, G. Bi, and P. Xu, "Call admission control with bandwidth reallocation for adaptive multimedia in high-rate WPAN," *Proceedings of the IEEE 6th Circuits and Systems Symposium on Emerging Technologies: Frontiers of Mobile and Wireless Communication, 2004.*, vol. 2, pp. 557–560, 31 May-2 Jun. 2004.

[4] A. Rangnekar and K. M. Sivalingam, "Multiple channel scheduling in uwb based ieee 802.15.3 networks," *First International Conference on Broadband Networks, 2004. BroadNets 2004. Proceedings.*, pp. 406–415, 25-29 Oct. 2004.

[5] K. W. Chin and D. Lowe, "A novel ieee 802.15.3 cta sharing protocol for supporting vbr streams," *Computer Communications and Networks, 2005. ICCCN 2005. Proceedings. 14th International Conference on*, pp. 107–112, 17-19 Oct. 2005.

[6] R. Fantacci, D. Tarchi, and G. Izzo, "Multimedia traffic management at MAC layer in IEEE 802.15.3a personal area networks," *2005 International Conference on Wireless Networks, Communications and Mobile Computing*, vol. 2, pp. 923–928, 13-16 Jun. 2005.

[7] S.-M. Kim and Y.-J. Cho, "Scheduling scheme for providing QoS to real-time multimedia traffics in high-rate wireless PANs," *IEEE Transactions on Consumer Electronics*, vol. 51, no. 4, pp. 1159–1166, Nov. 2005.

[8] R. Mangharam, M. Demirhan, R. Rajkumar, and D. Raychaudhuri, "Size matters: Size-based scheduling for mpeg-4 over wireless channel," *SPIE Conference on Multimedia Computing and Networking*, pp. 110–122, Jan. 2004.

[9] R. Zeng and G.-S. Kuo, “A novel scheduling scheme and mac enhancements for IEEE 802.15.3 high-rate WPAN,” *IEEE Wireless Communications and Networking Conference (WCNC 2005)*, pp. 2478–2483, Mar. 2005.

[10] X. Liu, Q. Dai, and Q. Wu, “Link-level scheduling for providing qos in wpan,” *International Conference on Communications, Circuits and Systems 2006*, pp. 1372–1376, Jun. 2006.

[11] G. Boggia, P. Camarda, L. A. Grieco, and S. Mascolo, “Feedback-based control for providing real-time services with the 802.11e MAC,” *IEEE/ACM Trans. on Networking*, vol. 15, no. 2, pp. 323–333, Apr. 2007.

[12] K. J. Astrom and B. Wittenmark, *Computer controlled systems: theory and design*, 3rd ed. Prentice Hall, 1995.

[13] L. Kleinrock, *Queueing systems, Volume I: Theory*. Wiley & Sons, 1975.

[14] Ns-2, “Network simulator,” http://www.isi.edu/nsnam/ns, 2007.

[15] I. Lab, “IEEE 802.15.3 MAC Model,” available at http://www.macroangel.com/nslast.zip, 2002.

[16] ISO/IEC International Standard, *Generic coding of moving pictures and associated audio information*, ISO/IEC International Standard 13818, 1994.

[17] Advanced Television Systems Committee, *Digital Audio Compression Standard (AC-3, E-AC-3) Revision B, A/52B*, Advanced Television Systems Committee, Jun. 2005.

[18] MPEG-4 Video Group, “Mpeg-4 overview,” available at http://www.mpeg.org/, Mar. 2006.

[19] International Telecommunication Union, *Video coding for low bit rate communication*, ITU-T Recomm. H.263, Feb. 1998.

[20] International Telecommunication Union (ITU), *A silence compression scheme for G.729 optimized for terminals conforming to Recommendation V.70*, ITU-T Recommendation G.729 Annex B, Nov. 1996.

[21] P. Seeling, F. H. Fitzek, and M. Reisslein, *Video Traces for Network Performance Evaluation. A Comprehensive Overview and Guide on Video Traces and Their Utilization in Networking Research*. Springer, 2006.