Proceedings of the 18th World Congress
The International Federation of Automatic Control
Milano (Italy) August 28 - September 2, 2011

Self-triggered Control of Multiple Loops over IEEE 802.15.4 Networks

U. Tiberi * **C. Fischione** ** **K.H. Johansson** ** **M.D. Di Benedetto** *

* *Department of Electrical and Information Engineering, and Center of Excellence DEWS University of L'Aquila, 67040 Poggio di Roio, L'Aquila, Italy (e-mail: {ubaldo.tiberi,mariadomenica.dibenedetto}@univaq.it).*
** *ACCESS Linnaeus Center, KTH Royal Institute of Technology, Stockholm, Sweden (e-mail:{carlofi,kallej}@ee.kth.se)*

Abstract: Given the communication savings offered by self-triggered sampling, it is becoming an essential paradigm for closed-loop control over energy-constrained wireless sensor networks (WSNs). The understanding of the performance of self-triggered control systems when the feedback loops are closed over IEEE 802.15.4 WSNs is of major interest, since the communication standard IEEE 802.15.4 is the de-facto the reference protocol for energy-efficient WSNs. In this paper, a new approach to control several processes over a shared IEEE 802.15.4 network by self-triggered sampling is proposed. It is shown that the sampling time of the processes, the protocol parameters, and the scheduling of the transmissions must be jointly selected to ensure stability of the processes and energy efficiency of the network. The challenging part of the proposed analysis is ensuring stability and making an energy efficient scheduling of the state transmissions. These transmissions over IEEE 802.15.4 are allowed only at certain time slots, which are difficult to schedule when multiple control loops share the network. The approach establishes that the joint design of self-triggered samplers and the network protocol 1) ensures the stability of each loop, 2) increases the network capacity, 3) reduces the number of transmissions of the nodes, and 4) increases the sleep time of the nodes. A new dynamic scheduling problem is proposed to control each process, adapt the protocol parameters, and reduce the energy consumption. An algorithm is then derived, which adapts to any choice of the self-triggered samplers of every control loop. Numerical examples illustrate the analysis and show the benefits of the new approach.

Keywords: Wireless Sensor Network, NCS, Self–Triggered Control, IEEE 802.15.4.

1. INTRODUCTION

Wireless sensor networks are making it possible to embed controllers and actuators everywhere in the physical world, where the state of processes can be sampled by sensors connected via wireless communications to controllers. The communication standard IEEE 802.15.4 is becoming the reference protocol for low data rate and energy-efficient WSNs, see IEEE 802.15.4 (2006). In industrial automation, it has been adopted with minor variations by other protocols such as WirelessHART and ISA100, Willig (2008). Although there are several studies on design and applications of IEEE 802.15.4 WSNs, there are not yet satisfactorily methods on how to close control loops over these networks.

When nodes of a WSN are battery powered, one of the main goal is to make the network life as long as possible. This can be obtained by reducing the energy consumption of nodes, which is mainly due to the communications. While in classical wireless networks a main goal is providing a high bandwidth, in WSNs the driving constraint is the energy consumption. This new constraint may have a significant impact on networked control systems (NCS) over WSNs and cannot be neglected.

In several results available in the NCS literature, the communication channel is often abstracted only in terms of packet losses and time delays, see e.g., Hespanha et al. (2007) and the references therein. However, the essential aspect of energy consumption and the typical dynamics of network protocols have not been considered, with the consequence that the controlled NCSs may be energy-inefficient. The network protocols do not allow the sensor nodes attached to the processes to send information to the controller at desired times. As a result, the controller is updated only at certain time instants and, between two consecutive transmissions, the process runs in open loop. Notice that this is exactly what happens nowadays in industry, where digital controllers are used to control continuous time processes. On the other hand, WSNs can adapt themselves to the requirements of the control applications, as proposed by the system level design method in Bonivento et al. (2007). Based on this approach, entirely new protocol stacks have been developed for control over WSNs, such as Breath and TREnD Park et al. (2010); Di Marco et al. (2010), but concerning the popular IEEE 802.15.4 protocol, this approach has not yet been fully investigated.

NCS over WSNs could be based on a periodic sampling of the state. Once a periodic controller is available, one could adapt the network and schedule the transmission according to these periods. However, in the specific case of IEEE 802.15.4 networks, this traduces to an inefficient usage of the network resources, thus wasting energy. The self–trigger control paradigm Velasco et al. (2003) – Anta and Tabuada (2009) has been recently proposed to save communication energy by

[1] This work was supported by the FeedNetBack and Hycon2 EU projects.

dynamically adapting the sampling time and transmitting only when it is needed. However, the communication protocols do not allow to transmit information at any given instants, but only during certain times. This means that the self-triggering sampling must be co-designed with the network protocol, to avoid the situation in which a sample must be transmitted but the protocol does not give access to the wireless channel. To overcome this problem, in Tiberi et al. (2010) we showed that by a system level design it is possible to design a controller based on the self–triggered sampling paradigm, and, simultaneously, adjust the protocol parameter to ensure a practical stability of the process and a parsimonious usage of the network energy consumption.

In this paper, we extend our earlier work to the case of several loops that share the same IEEE 802.15.4 network. Compared to the single loop control problem, the control of multiple loops is much more complex: first we have to decide how to adapt the network parameters to meet the control requirements, and then we have to determine an energy–efficient scheduling for the transmission of parallel process states. Based on a self–triggered control policy, we provide a new decentralized adaptive algorithm to achieve stability of the entire NCS while reducing the energy consumption of the nodes. Moreover, we establish that our proposed control and scheduling mechanism leads to an increase of the network capacity. To the best of our knowledge, this is the first paper that extends the self-triggered sampling paradigm to the control of multiple loops over WSNs while ensuring energy-efficiency.

The rest of the paper is organized as follows: In Section 2, the networked control system architecture is introduced. The problem we tackle in this paper is posed in Section 3. Section 4 gives the core contribution of the paper. In Section 5, we illustrate the analysis by simulation results.

1.1 Notation

We indicate by $\|\cdot\|$ the usual Euclidean norm. The notation $\mathbb{N}^+$ indicates the set of the natural numbers $\mathbb{N}^+ = \mathbb{N}\backslash\{0\}$. The error due to the sampling is denoted by $e_{i,k} = x_i(t_{i,k}) - x_i(t)$. Coherently with the IEEE 802.15.4 communication standard, we denote by *superframe* the duration of the time over which multiple nodes attempt to transmit in a time division multiple access fashion. The duration of the portion of the frame allocated to a node is called time slot. We indicate with Δ_{slot} the time slot duration, with $T_{0,k}$ the time in which the k-th superframe begins and with $T_{i,k}$ the time in which the allocated time slot for the i-th node begins in the k-th superframe.

2. IEEE 802.15.4 NETWORKED CONTROL SYSTEM

In this section we describe the IEEE 802.15.4 NCS architecture we address in this paper. We consider n independent control loops that share the same wireless network, and we limit our attention to one-direction channel feedback, in which there is wireless communication only between the processes and the controllers. We consider star topology networks, where each sensor node is attached to a process and transmits to the unique central node of the network. This central node is directly connected to the controllers. We assume that each node can measure the full state of the associated process. The overall NCS scheme is depicted in Fig. 1.

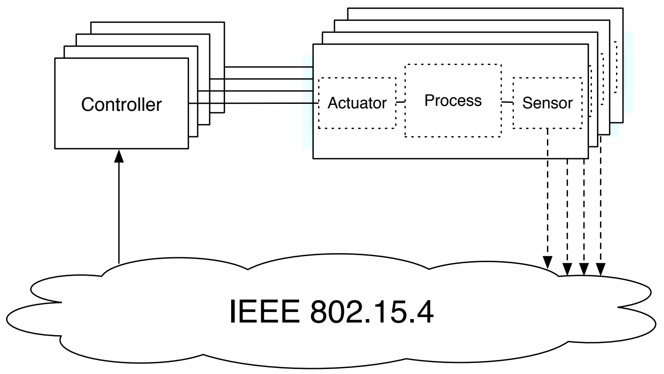

Fig. 1. A networked control system where a number of independent control loops transmit over a shared IEEE 802.15.4 network.

2.1 Processes and controllers

We assume that every process $i = 1, \ldots, n$ follows linear dynamics of the form

$$\dot{x}_i = A_i x_i + B_i u_i \,, \tag{1}$$

where $x_i \in \mathbb{R}^{n_i}$ and $u_i \in \mathbb{R}^{m_i}$, where the index i is associated to the loop of the i-th process. We assume that the state of the process i is sampled at time $T_{i,k}$, where k refers to the superframe number of the IEEE 802.15.4 network. We explain in the following subsection the superframe organization of IEEE 802.15.4. The state is then transmitted at time $T_{i,k}$ to the node that is connected to the controller. When the controller receives the state measurement, it holds it and the resulting control is piecewise constant, with

$$u_i(t) = K_i x_i(T_{i,k}) \,, \quad t \in [T_{i,k}, T_{i,k+1}) \,. \tag{2}$$

By using the controller (2), the closed loop dynamics of the i-th loop, for $t \geq T_{i,k}$, can be rewritten as

$$\dot{x}_i = (A_i + B_i K_i) x_i + B_i K_i e_{i,k} \,, \tag{3}$$

Notice how the perturbation term $e_{i,k}$ leads the system to instability. In the following subsection, we introduce the network over which the state information is transmitted to the controller.

2.2 IEEE 802.15.4 Protocol Model

The IEEE 802.15.4 standard specifies the physical and medium access control layers of the protocol stack of WSNs composed by low cost and low powered nodes, IEEE 802.15.4 (2006). In each 802.15.4 network there is a special node, the PAN coordinator (PANC), that manages the operations of the entire network. We assume that the controllers are connected to this coordinator.

In the unslotted modality networks the nodes attempt to transmit packets according to the Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) algorithm for all the time, while in slotted modality networks, the nodes transmit packets in a time division multiple access (TDMA) fashion, see IEEE 802.15.4 (2006). In slotted modality networks, the time frame of the protocol is denoted as *superframe*, which is bounded by special signalling packets sent by the PANC called *network beacons* to manage the network. Every node of the network must follow this superframe when transmitting packets. The superframe length is denoted as *Beacon Interval* (BI) and satisfies

$$\mathrm{BI} = aBaseSuperFrameDuration \times 2^{\mathrm{BO}} \,, \tag{4}$$

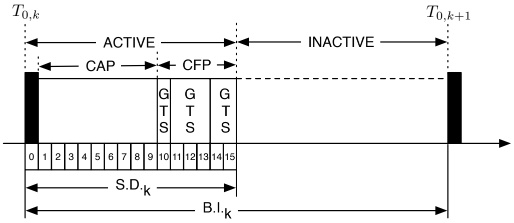

Fig. 2. Slotted IEEE 802.15.4 superframe time organization. The index $k \geq 0$ denotes the k superframe. BD_k denotes the superframe duration and BI_k denotes the beacon interval. $T_{0,k}$ is the time in which the superframe begins. During the inactive period, nodes sleep to save energy. IEEE 802.15.4 allows us to adapt the protocol parameters SD and BI to the needs of the NCS.

with $0 \leq \mathrm{BO} \leq 14$, where the exponent is called *Beacon Order* and $aBaseSuperFrameDuration$ is a parameter of the protocol, which specifies the shortest duration of a superframe. The superframe is split into an active portion and an inactive portion. The active portion is the time interval where there can be transmissions of packets. In the inactive period no communication is allowed and the nodes go in a sleep state to save energy. The time interval of the active period is called *Superframe Duration* (SD). It is divided into 16 equally sized time slots and satisfies

$$\mathrm{SD} = aBaseSuperFrameDuration \times 2^{\mathrm{SO}}\,, \tag{5}$$

with $0 \leq \mathrm{SO} \leq 14$ and where the exponent SO is called *Superframe Order*. It must be SO≤BO, according to the IEEE standard. The parameter $aBaseSlotDuration = \mathrm{SD}/16$ indicates the length of each time slot. For notation simplicity, we define $\Delta_{slot} \triangleq aBaseSlotDuration$. The active portion is further divided in two parts: the *Contention Access Period* (CAP) and the *Contention Free Period* (CFP). During the CAP nodes contend to access the medium with the CSMA/CA algorithm, whereas in the CFP period the PANC reserves dedicated time slots to nodes so that they do not have to contend for transmitting packets. During the current superframe, a node can ask to the PANC a number of dedicated time slots (up to 7 time slots per superframe). Whenever possible, the PANC allocates the required time slots for the next superframe. This mechanism is called *Guarantee Time Slots* allocation. A time slot is called GTS. During a GTS, a node can send and receive more than one packet. The scheduling is decided before the starting of the superframe by the PANC and is communicated to all nodes by the beacon message. The IEEE 802.15.4 MAC superframe is depicted in Fig. 2.

At the beginning of each superframe, all the nodes of the network must be awake to receive the beacon packet from the PANC. This beacon packet contains all the settings of the incoming superframe, such as which GTS is reserved to which node (control loop), the length of the incoming beacon interval, and the superframe duration. During the inactive period, the nodes are in a sleep state. They wake up to receive the next beacon packet from the PANC at the end of the inactive period. The IEEE 802.15.4 standard allows us to adapt the superframe to the need of the controller by tuning the protocol parameters SD and BI, and it allow us to decide if allocate or not a GTS to a certain node and in which time slot it must be allocated. These decisions are taken at the PANC during superframe k, and they will take effect in the next superframe $k+1$.

A common measure of the energy efficiency of the network is given by the *duty cycle* of the nodes, which is defined as

$$\mathrm{DC}_k\% = \frac{\mathrm{SD}_k}{\mathrm{BI}_k} \times 100\,.$$

The *network capacity* indicates how many nodes are allowed to transmit on the network during a superframe. We define the network capacity of the k-th superframe as the ratio of the available time slots in the k-th superframe to the used time slots in that superframe:

$$\mathrm{C}_k = \frac{16}{Allocated\,GTS\;in\;the\;k-th\;superframe}\,.$$

During CAP, since there is no control on the delay encountered by the packets before being transmitted, and there is no guarantee that the packets can be received successfully due to possible collisions, in this paper we limit our attention to the CFP. We assume that a node attached to a process is scheduled for transmission to one GTS, and, whenever a GTS is allocated, the associated node sends the full measurement to the PANC within a time slot duration Δ_{slot}. Because of the simple network topology (star topology) and the utilization of the GTSs, we assume full reliability and bounded time delays. We also assume that a beacon is sent and received by all the nodes within a time slot duration. In the sequel we will show how to control the network by considering a fixed SD, a dynamic adaptation of the parameter BO_k and a dynamic allocation of the GTSs to reduce the duty cycle and to increase the network capacity while ensuring the stability of the entire NCS.

3. PROBLEM FORMULATION

We aim at designing an algorithm to adapt dynamically BO and the schedule the GTS with a threefold goal: achieve stability of each loop, reduce the duty-cycle, and increase the network capacity. Since we are assuming that whenever a GTS is allocated the associated node performs a transmission, a reduction of the number of allocated GTS leads to a reduction of the number of transmissions in addition to a network capacity increasing.

4. SYSTEM LEVEL DESIGN OF IEEE 802.15.4 SELF–TRIGGERED NCS

In this section we present the core contribution of the paper, namely the system level design of multiple control loops over a shared IEEE 802.15.4 network. We use a self–triggered sampler for each loop to dynamically adapt the superframe by the beacon interval, and to allocate the time slots GTSs. We show that the duty cycle is reduced by varying the beacon interval BI_k parameter at each superframe, by considering a fixed superframe duration SD_k, $\forall k$, and by allocating the GTS only when a node actually needs to transmit the state information.

4.1 The self–triggered control

Sampled–data systems have often been studied by assuming periodic implementations of the controller, where the sampling period is chosen by following some rule of thumb (see Åström (1997) or similar textbooks). This often leads to a conservative utilization of resources, because next sample is taken disregarding the current measure of the state. The self–triggered sampling aims at reducing the conservativeness of periodic implementations by using the current measure to determine the next sampling instant, see Mazo and Tabuada (2008) and Wang and Lemmon (2009). More in detail, a self–triggered sampler has the general expression

$$t_{i,k+1} = t_{i,k} + \gamma_i(\|x_i(t_{i,k})\|)\,, \tag{6}$$

where the function $\gamma_i(\cdot)$ increases as $\|x_i(t_{i,k})\|$ decreases. For networked applications, self-triggered sampler must have a bound as imposed by the communication protocol. In general, the next sampling time given by a self-triggered sampler is a sampling sufficient condition. Hence, if $\gamma_i(\|x_i(t_{i,k})\|)$ becomes too small for some $x_i(t_{i,k})$, one can always go back and use a periodic implementation of the associated controller. It is then useful the following definition:

Definition 4.1. Let n be the number of loops over the same shared network and consider the set $\mathcal{H} = \{h_1, \ldots, h_n\}$, where h_i is the period of a time periodic implementation of the controller associated to the i-th process. We define the *fastest loop* of the network the j-th loop that satisfies

$$j = \arg\min_i \{h_i\}\,, \quad h_i \in \mathcal{H}\,,$$

with *fastest time* $h_{\text{fastest}} = h_j$. ⋄

By considering that the IEEE 802.15.4 standard allows us to pick the measurements at time $T_{i,k}$, and by including periodic implementations, we modify (6) as

$$t_{i,k+1} = T_{i,k} + \max\{h_i, \gamma_i(\|x_i(T_{i,k})\|)\}\,. \tag{7}$$

Note that $t_{i,k+1} > T_{i,k}, \forall i, k$. The value of $t_{i,k+1}$ can be viewed as a deadline by which a measurement must be picked to ensure stability of the i-th loop.

4.2 Dynamic adaptation of the Superframe length

In this section we describe the dynamic adaptation of the beacon order, and thus the beacon interval that measures the frame length, to achieve an increase the duty cycle of the network. With these goals in mind, we need some definitions:

Definition 4.2. Let n be the number of nodes of the network. The *scheduling set* of node i is a set $\vartheta_i = \{\vartheta_{i,k}\}_{k\in\mathbb{N}^+}, \vartheta_{i,k} \in \{0, 1, \ldots, n\}$ such that if $\vartheta_{i,k} = j \neq 0$ node i transmits at slot j in the k-th superframe. If $\vartheta_{i,k} = 0$, node i does not perform any transmission during the k-th superframe. ⋄

Definition 4.3. Let n be the number of the nodes of the network. A *well posed GTS* scheduling in the k-th superframe, is a set $\Theta_k = \{\vartheta_{1,k}, \ldots, \vartheta_{n,k}\}$ such that $\vartheta_{i,k} \neq \vartheta_{j,k}\, \forall i, j, k$ and $\min \Theta_k \geq 1$, $\forall k$. ⋄

In the previous definition, *well posed* denotes that two nodes cannot be scheduled in the same time slot, and that in each superframe at least one node is allocated in a GTS.

Definition 4.4. We define *last scheduled GTS* of node i the quantity $l_{i,k} = \max_{1\leq j\leq k} j : \vartheta_{i,j} \neq 0, \vartheta_{i,j} \in \vartheta_i$. ⋄

Given the k-th superframe, the value of $l_{i,k}$ tells us when node i performed the last transmission.

According to the previous definitions, note that the time in which a node performs a transmission during the k-th superframe is $T_{i,k} = T_{0,k} + \vartheta_{i,k}\Delta_{slot}$.

Finally, consider node i and let $0 < j < k$ such that $\vartheta_{i,j} \neq 0, \vartheta_{i,k} \neq 0$ and $\vartheta_{i,r} = 0, \forall j < r < k$. We define the time between a transmission of loop i during superframe j and its first next transmission during some superframe k as $D(\vartheta_{i,k}|j) = T_{0,k} - T_{i,j} + \vartheta_{i,k}\Delta_{slot}$, see Fig.3.

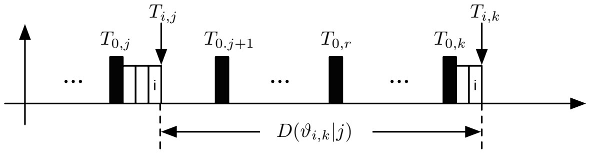

Fig. 3. $D(\vartheta_{i,k}|j)$ is the time between a transmission of loop i during superframe j and its first next transmission during some superframe k.

We are now ready to show how to adapt dynamically the superframe length. As we have shown in Section 2.2, recall that during a superframe k, the PANC must decide the length of the next superframe, namely it must set the value of $T_{0,k+2}$ of the next superframe interval $[T_{0,k+1}, T_{0,k+2})$. The idea is then to set the ending of next superframe $T_{0,k+2}$ in such a way that the processes can be sampled as sporadically as possible. To set the best $T_{0,k+2}$, we need the state $x_i(t_{i,k+1})$ during the superframe $k+1$, because this state is used in the self–triggered sampler (7) to compute next deadline $t_{i,k+2}$. Clearly, $T_{0,k+2}$ should be placed before such a deadline. However, the decision of $T_{0,k+2}$ can be taken by the PANC only during the superframe k, when no state information to be taken during superframe $k+1$ is already available. Therefore, we use last measurement of process i $x_i(T_{i,l_{i,k}})$ to obtain an upper bound prediction of the next measurement in superfame $k+1$:

$$\begin{aligned}\|\hat{x}_i(T_{i,k+1}|x_i(T_{i,l_{i,k}})\| = \|&\Phi_i(D(\vartheta_{i,k+1}|l_{i,k})) \\ &+ \Gamma_i(D(\vartheta_{i,k+1}|l_{i,k}))\|\|x_i(T_{i,l_{i,k}})\|\,,\end{aligned} \tag{8}$$

where

$$\Phi_i(D(\vartheta_{i,k+1}|l_{i,k})) = e^{A_i D(\vartheta_{i,k+1}|l_{i,k})}\,,$$

and

$$\Gamma_i(D(\vartheta_{i,k+1}|l_{i,k})) = \int_0^{D(\vartheta_{i,k+1}|l_{i,k})} e^{A_i s} B_i K_i ds\,.$$

Note that estimate $\|\hat{x}_i(T_{i,k+1}|x_i(T_{i,l_{i,k}})\|$ depends on the length of the time horizon $D(\vartheta_{i,k+1}|l_{i,k})$, and on $\vartheta_{i,k+1}$. Thus, we use the self–triggered samplers in the following predictive form

$$\hat{t}_{i,k+2|l_{i,k}} = T_{i,k+1} + \max\{h_i, \gamma_i(\|\hat{x}_i(T_{i,k+1}|x_i(T_{i,l_{i,k}})\|)\}\,.$$

By using the self–triggered in a predictor form, we are now in the position to establish an adaptation policy for the superframes:

Theorem 4.1. Let a NCS over a slotted IEEE 802.15.4 network be composed by n nodes and suppose that n self–triggered samplers in the form (7) are used. Assume that every node can measure the full state of the associated process and that $\vartheta_{i,k} \neq 0$, $\forall i, k$, i.e., all the nodes send a packet in every superframe. By selecting the superframe order as

$$\text{SO}_k = \left\lfloor \log_2 \frac{h_{\text{fastest}}}{aBaseSuperFrameDuration} \right\rfloor \forall k\,, \tag{9}$$

and by adapting the the beacon order with

$$\text{BO}_{k+1} = \text{BO}_k + \left\lfloor \log_2 \frac{\hat{t}_{k+2} - T_{0,k+1} - (n+1)\Delta_{slot}}{\text{BI}_k} \right\rfloor, \tag{10}$$

where $\hat{t}_{k+2} = \min_{\vartheta_{i,k+1}}\{\hat{t}_{i,k+2|l_{i,k}}\}$, then the NCS is stable. ◁

This theorem says that the adaptation of the superframes is done on the base of the minimum response time among all self–triggered samplers, and that in each superframe all nodes will transmit.

4.3 GTS Scheduling

In the previous subsection, we have established how to adapt the superframes length to control multiple loops while reducing the duty cycle. Now, the next step is to determine how time slots can be scheduled in each superframe to the various processes. We show that there is no need for slow control loops to wake-up at each superframe, but they can be deallocated and then re-allocated later on, thus saving the energy and increasing the network capacity.

The idea of the scheduling that we would like to propose is based on that if a node i is not allocated to any GTS of a superframe, it does not perform a transmission and the network has an additional free time slot that can be included in the CAP. This results in an increase of the network capacity. We have the following result:

Corollary 4.1. Consider the assumptions of Theorem (4.1). Assume to set the superframe order as in Eq. (9) and to adapt the beacon order as in Eq. (10). Finally, let $\mathcal{T}_{k+1} = \{t_{i,l_{i,k}+1} : T_{0,k+1} \le t_{i,l_{i,k}+1} \le T_{0,k+2}\}$. Then, the GTS scheduling Θ_{k+1} are well posed for all k and the NCS is stable if

- $\vartheta_{i,k+1} \neq 0$ for all nodes $i : t_{i,l_{i,k}+1} \in \mathcal{T}_{k+1}$,
- $\vartheta_{j,k+1} \neq 0$ for the node j such that $j : \hat{t}_{j,k+2|l_{i,j}} = \hat{t}_{k+2}$,
- $\vartheta_{r,k+1} = 0$ for all other nodes.

◁

This corollary, along with Theorem 4.1, allow us to obtain Algorithm 1, which adapts the superframe duration and schedule the GTS to the control loops so as to reduce energy consumption and allocate transmissions only if needed. We remark that to compute $t_{i,l_{i,k}+1}$ and $\hat{t}_{i,k+2|l_{i,k}}$ in the algorithm, node i must compute: 1) $\|x_i(T_{i,l_{i,k}})\|$, 2) $D(\vartheta_{i,k+1}|l_{i,k})$ and 3) $\|\hat{x}_i(T_{i,k+1}|x_i(T_{i,l_{i,k}})\|$. If nodes do not have sufficient computation capabilities, these computation can be devolved to the PANC. An example of application of Algorithm 1 is given in Fig. 4.

Algorithm 1 Dynamic network protocol adaption

```
init
The PANC sets SO_1 with (9);
The PANC sets BI_1 = BI_min;
The PANC sets ϑ_{i,1} ≠ 0, ∀1 ≤ i ≤ n;
end init
for all k do
  The PANC sends a beacon;
  for all the nodes i s.t. ϑ_{i,k} ≠ 0 do
    l_{i,k} ← k;
    The node picks the measurement x_i(T_{i,k});
    The node computes t_{i,l_{i,k}+1} and t̂_{i,k+2|l_{i,k}};
    The node sends x_i(T_{i,k}), t_{i,l_{i,k}+1} and t̂_{i,k+2|l_{i,k}} to the
    PANC;
    The PANC updates the control laws;
  end for
  for all the nodes j s.t. ϑ_{j,k} = 0 do
    The PANC computes t̂_{j,k+2|l_{i,k}};
    The PANC sets BO_{k+1} with (10);
    The PANC sets Θ_{k+1} according to Corollary 4.1;
  end for
end for
```

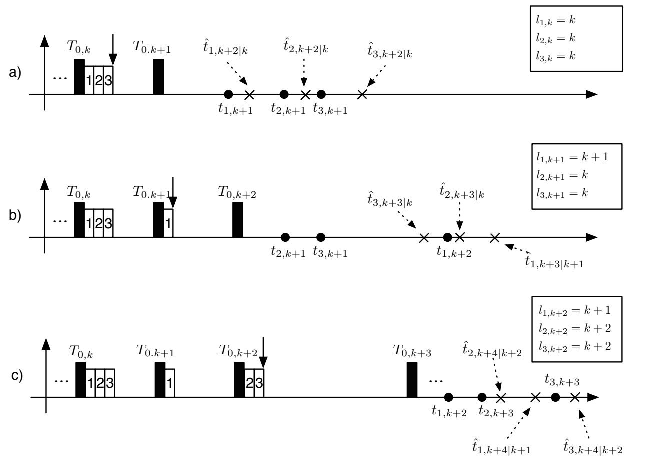

Fig. 4. Example of application of Algorithm 1: a) At the k-th superframe, BO_{k+1} is adapted according to $\hat{t}_{1,k+2|k}$ and node 1 is scheduled in the $(k+1)$-th superframe. b) At the $(k+1)$-th superframe, the only node scheduled for transmission is node 1. BO_{k+2} is adapted according to $\hat{t}_{3,k+3|k}$ and nodes 2 and 3 are scheduled in the $(k+2)$-th superframe. c) The resulting superframe after the $(k+1)$-th adaptation. Note that all the deadlines are met.

5. SIMULATION RESULTS

In this section we present simulation results that illustrate the analysis we developed in the previous sections. We consider the problem of stabilizing three control loops that share same IEEE 802.15.4 network. The three loops have the form of Eq. (1) and the controllers are have the form of Eq. (2). For every loop, we used the self–triggered sampler developed in Tiberi et al. (2010). Simulations include time delays in the self-triggered samples of Eq. (7) as described in Tiberi et al. (2010). The three loops are specified in the following:

Loop 1. The first loop is

$$A_1 = \begin{bmatrix} -0.1 & 0.05 \\ 0.2 & 0.1 \end{bmatrix}, \qquad B_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

We set $\varepsilon_1 = 1.2$ and and $h_1 = 300\,\mathrm{ms}$. The controller is designed to put the closed loop system eigenvalues in $\lambda_{1,1}^{(A_1+B_1K_1)} = -0.32, \lambda_{1,2}^{(A_1+B_1K_1)} = -0.15$. The initial condition are $x_{1,1}(0) = 40, x_{1,2}(0) = 40$.

Loop 2. The second loop is

$$A_2 = \begin{bmatrix} 0.2 & 0 \\ 0.2 & 0.2 \end{bmatrix}, \qquad B_2 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

We set $\varepsilon_2 = 0.25$ and $h_2 = 310\,\mathrm{ms}$. The controller sets the closed loop system eigenvalues in $\lambda_{2,1}^{(A_2+B_2K_2)} = -0.4, \lambda_{2,2}^{(A_2+B_2K_2)} = -0.6$. The initial conditions for this loop are $x_{2,1}(0) = -40, x_{2,2}(0) = 20$.

Loop 3. The third loop is

$$A_3 = \begin{bmatrix} -0.4 & 0 \\ -0.2 & 0.2 \end{bmatrix}, \qquad B_3 = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

with $\varepsilon_3 = 0.22$ and $h_3 = 330\,\mathrm{ms}$. The closed loop eigenvalues are in $\lambda_{3,1}^{(A_3+B_3K_3)} = -0.46, \lambda_{3,2}^{(A_3+B_3K_3)} = -0.6$. The initial conditions are $x_{3,1}(0) = 30, x_{3,2}(0) = 40$.

Protocol parameters. We set the IEEE 802.15.4 network parameters as SO$= 4$ (and then $\mathrm{BO}_{\min} = 4$), for which we had a time slot duration $\Delta_{slot} = aBaseSlotDuration = 15.32$ ms and we considered a maximum network time delay of 10 ms.

The results are reported in Figs. 5, 6, and 7. All the loops are correctly controlled. The adaptation of the network parameters gives a reduction of the duty cycle and increases the network capacity. Specifically, after 150 superframes we had an average duty cycle $\simeq$ 75.2% and an average network capacity of 10.56. In addiction to a reduction of the duty cycle, we had 68 transmissions for the first loop, 132 for the second loop and 82 for the third loop resulting in a reduction of the number of transmissions and then to a further energy saving. For instance, it is interesting to see how the network experiences an increase of the network capacity but a fixed duty cycle around 11 s. This is because there are some loops that require a larger sampling and their associated GTS are deallocated, while some other loop requires a fast sampling that enforces a short BO and then a high duty cycle.

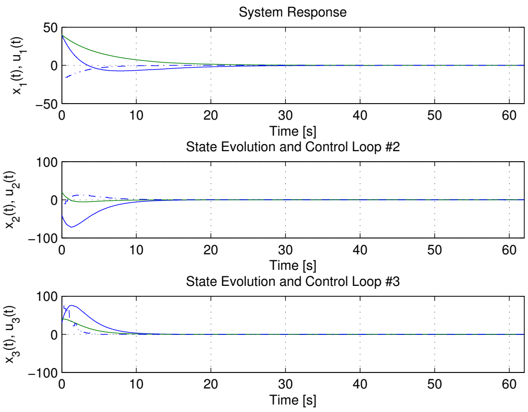

Fig. 5. Systems state evolution (continuous line) and control (dashed–dot line).

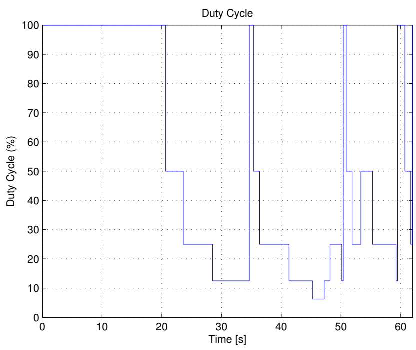

Fig. 6. Duty cycle of the network for every superframe.

6. CONCLUSIONS

In this paper we investigated how to control multiple loops over a shared IEEE 802.15.4 wireless sensor network. We used self–triggered control and proposed a decentralized algorithm to ensure stability of every control loop by dynamically adapting the protocol parameters. We showed that the proposed adaptation policy has three benefits: it increases the sleep time of the nodes, it reduces the number of transmissions, and increases the network capacity.

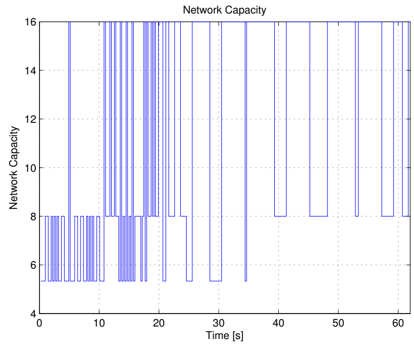

Fig. 7. Network capacity for every superframe.

Future works include the study of the fundamental network capacity of NCS over IEEE 802.15.4 when self/event–triggered strategies are used over hybrid MAC protocols and packet losses are present.

REFERENCES

Anta, A. and Tabuada, P. (2009). On the benefits of relaxing the periodicity assumption for networked control systems over can. In 30^{th} *Real-Time System Symposium*.

Åström, K.J. (1997). *Computer Controlled Systems*. Prentice Hall, 3^{rd} edition.

Bonivento, A., Fischione, C., Necchi, L., Pianegiani, F., and Sangiovanni-Vincentelli, A. (2007). System level design for clustered wireless sensor networks. *IEEE Transactions on Industrial Informatics*, 202–204.

Di Marco, P., Park, P., Fischione, C., and Johansson, K.H. (2010). TREnD: a timely, reliable, energy-efficient and dynamic wsn protocol for control applications. In *Proceedings of the Information Communicatoin Conference*.

Hespanha, J.P., Naghshtabrizi, P., and Xu, Y. (2007). A survey of recent results in networked control systems. *Proceedings of the IEEE*, 95(1).

IEEE 802.15.4 (2006). Wireless medium access control (mac) and physical layer (phy) specifications for low rate wireless personal area networks (lr-wpans).

Mazo, M. and Tabuada, P. (2008). On event-triggered and self-triggered control over sensor/actuator networks. In 47^{th} *IEEE Conference on Decision and Control*.

Park, P., Fischione, C., Bonivento, A., Johansson, K.H., and Sangiovanni-Vincentelli, A. (2010). Breath: an adaptive protocol for industrial control applications using wireless sensor networks. *IEEE Transactions on Mobile Computing*, To appear.

Tiberi, U., Fischione, C., Johansson, K.H., and Di Benedetto, M.D. (2010). Adaptive self triggered control over 802.15.4 networks. In 49^{th} *IEEE Conference on Decision and Control*. Atlanta, Georgia - USA.

Velasco, M., Marti, P., and Fuertes, J. (2003). The self triggered task model for real-time control systems. In *Work-in-Progress Session of the* 24^{th} *IEEE Real-Time Systems Symposium (RTSS03)*.

Wang, X. and Lemmon, M. (2009). Self-triggered feedback control systems with finite-gain $\mathcal{L}_2$ stability. *IEEE Transactions on Automatic Control*, 54(3), 452–467. doi: 10.1109/TAC.2009.2012973.

Willig, A. (2008). Recent and emerging topics in wireless industrial communication. *IEEE Transactions on Industrial Informatics*, 4(2), 102–124.