Discrete-time Event-triggered Control for Wireless Networks: Design and Network Calculus Analysis

M. Hossein Roohi[1], Jafar Ghaisari[2], Iman Izadi[3] and Hossein Saidi[4]
Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, Iran
Email[1]: mh.roohi@ec.iut.ac.ir
Email[2]: ghaisari@cc.iut.ac.ir
Email[3]: iman.izadi@cc.iut.ac.ir
Email[4]: hsaidi@cc.iut.ac.ir

***Abstract*—In the past few years, as an alternative for time-triggered control, event-triggered control (ETC) has been investigated for networked control systems (NCS). ETC is a control strategy where the control algorithm is executed only when a special condition is satisfied. In this paper, a solution to find an appropriate bandwidth for event-based control systems over wireless networks is proposed. Moreover, it is shown how the delay bound of network can be affected by control system. These two results are achieved by applying network calculus theorem to a control system. For efficient use of network resources, a new triggering algorithm is proposed. An advantage of proposed algorithm is that the minimum time between two consecutive events is considered as a designable control parameter and it can guarantee a lower bound on input traffic to the network. An illustrative case study of a NCS based on IEEE 802.15.4 protocol is presented to show the effectiveness of the proposed event-triggered algorithm and investigate the effect of bandwidth allocation on NCS.**

I. Introduction

Nowadays, according to the development of wireless communication technology we have witnessed an increasing trend to design control systems in which the control loops are closed over a wireless network. From the other side, for some application, industries need a kind of low priced control system which results to cheap digital system with pure resources for computation and communication. This led to a new control strategy called event-triggered control (ETC).

The idea behind ETC is to send data just when it is necessary to send. The criterion of necessity of sending data is determined so as to guarantee a specified performance or just stability of the control system. So an event-triggered control system needs a mechanism that detects whether or not to send data. This mechanism is called event detector or triggering mechanism.

There are many contributions have been given in the literature for event-triggered control systems. To mention a few, [1] addressed the problem of scheduling control tasks on embedded microprocessors to guarantee the asymptomatically stability of the overall system. [2] proposed an event-triggered strategies for control of discrete-time systems and extended the results for self-triggered strategy. Similarly, [3] analyzed discrete-time systems where the authors proposed a Lyapunov based triggering mechanism, in which the value of some Lyapunov function is calculated at each sampling instant and based on model of the system the value of the Lyapunov function for the next sampling instant is determined. So the triggering mechanism can verify whether the Lyapunov function is decreasing or not, the results are based on state feedback as well as output feedback scheme. [4], [5] Introduced periodic event-triggered control (PETC) which deals with continuous-time model of the system. A significant contribution of PETC is that the triggering condition is verified periodically and at sampling times.

From the other side, some paper studied the problem of scheduling event-triggered controller for networked control system. for instance, [6] investigated a scenario where a number of control loops are using a shared network for various medium access protocols. Furthermore in [7], the authors showed how event-triggered controller can be implemented over the well known IEEE 802.15.4 protocol.

Event-based control systems are supposed to decrease the usage of bandwidth. For this purpose numerous research are done which minimize the total number of event during a period of time. From the other side, according to our studies on communication networks, it is more important to decrease the input traffic to the network which means the input data to a network in a short period of time. An important result of this kind of reduction is that the network can simply guarantee a predictable bandwidth to a control loop and furthermore it simplifies the analysis of network delay which affect the control loop. In this paper we propose a procedure to design controller so that the time between to consecutive event (inter-event time) could be considered as a parameter of designing. Indeed the minimum inter-event time is guaranteed in the absence of disturbance which results an upper bound on input traffic to the network.

From the network point of view, we use network calculus as our analyzing tool. Network calculus is set of theorem that yields a deterministic model of the network. The model in network calculus differs from other models of network in the assumption about the traffic entering the network [8], [9]. The most of other, model the traffic entering as a stochastic process. There are many papers devoted to analyzing the

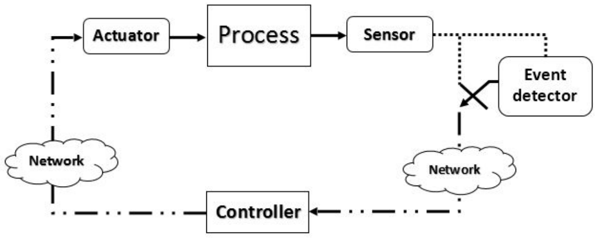

Fig. 1: Networked event-triggered control system

network behaviors responding to various arrival traffics using network calculus [10]–[12]. Network calculus is used by [13] to investigate the guaranteed time slot (GTS) allocation in IEEE 802.15.4, and we will use the results of this paper for our case study.

II. Event-triggered Control

A typical event-triggered control system is shown in Fig. 1, which consists of a process, a sensor which discretizes the output of the process, an event detector, a controller, and an actuator which contains a Zero-Order Hold (ZOH) that holds the discrete-time control input for the process. In this system the event detector measures a discrete-time signal periodically and verifies a triggering condition. As soon as the triggering condition is satisfied, the event detector sends the state of the system to the controller.

A. Linear Discrete-Time System

Consider a discrete time linear time invariant system

$$x(k+1) = Ax(k) + Bu(k), \tag{1}$$

where $x : \mathbb{Z}^+ \to \mathbb{R}^n$ is the state vector and $u : \mathbb{Z}^+ \to \mathbb{R}^m$ is the control input. Now consider an event based state feedback controller which is given by

$$u(k) = F\hat{x}(k). \tag{2}$$

Here, $\hat{x}(k)$ changes just at the triggering instants. So, let us define it as

$$\hat{x}(k) = \begin{cases} x(k), & \text{when } C(x(k), \hat{x}(k)) > 0 \\ \hat{x}(k-1), & \text{when } C(x(k), \hat{x}(k)) \leq 0 \end{cases}, \tag{3}$$

in which $C(x(k), \hat{x}(k))$ denotes triggering function. The closed loop state space equation of the system is

$$x(k+1) = Ax(k) + BF\hat{x}(k). \tag{4}$$

Now, assume the triggering function

$$C(x(k), \hat{x}(k)) = \sigma\|x(k)\| - \|\hat{x}(k)\|, \tag{5}$$

and the triggering condition

$$\sigma\|x(k)\| - \|\hat{x}(k)\| > 0, \tag{6}$$

with $0 < \sigma < 1$. The triggering mechanism checks (6) at sampling instant $k-1$. In fact, it predicts $x(k)$ and computes $C(x(k), \hat{x}(k))$. This procedure can be expressed as Algorithm 1.

Algorithm 1 Event Trigger Protocol

Let $k = 0$
Compute the control value $u(k) = Fx(k)$ and send to the plant.
In the event detector compute

$$x(k+1) = Ax(k) + BF\hat{x}(k)$$

if $C(x(k+1), \hat{x}(k+1)) > 0$, **then**
  Update $\hat{x}(k+1)$ with $x(k+1)$
  Compute $u(k+1) = F\hat{x}(k+1)$ and send the control value to the plant.
Do the procedure for the next sampling instant.

B. Stability Analysis

By definition, the discrete time switched system

$$x(k+1) = A_{\sigma(k)}x(k) + B_{\sigma(k)}w(k), \qquad x(k) = x_0. \tag{7}$$

is exponentially stable (with stability degree $\lambda < 1$) if $\|x(k)\| \leq c\lambda^k\|x_0\|$ for all $k \geq 0$ and a constant c.

Theorem 1. *The proposed event based state feedback of Algorithm 1 guarantees exponential stability of system* (1) *with* $\lambda = \sigma^{\frac{1}{max(m_i)}}$, *where* m_i *denotes the maximum inter-event samples.*

Proof: Let $x(k_i)$ denotes the state of system (1) at sampling instant i. Algorithm 1 implies that

$$\|x(k_{i+1})\| \leq \sigma\|x(k_i)\| \leq ... \leq \sigma^{i+1}\|x(k_0)\|. \tag{8}$$

Furthermore, according to (1) and (2) for the sampling instant after each triggering instant

$$x(k_i+1) = Ax(k_i) + BFx(k_i). \tag{9}$$

So for $k_i < k \leq k_{i+1}$, (between two consecutive events) we have

$$x(k) = \Big(A^{k-k_i} + \big(\sum_{j=0}^{k-k_i-1} A^j\big)BF\Big)x(k_i). \tag{10}$$

It is clear that for a matrix P and a vector v, $\|Pv\| \leq \|P\|\|v\|$. Now assume that the state feedback controller F is designed to exponentially stabilize the system. So

$$\|A^{\tilde{k}} + \big(\sum_{j=0}^{\tilde{k}-1} A^j\big)BF\| < \sigma, \tag{11}$$

in which $\tilde{k} = k - k_i$ and $\sigma < 1$. So

$$\|x(k)\| \leq \|x(k_i)\| \qquad \forall k \ st.\ k_i < k \leq k_{i+1}, \tag{12}$$

which means

$$\begin{aligned} \sigma^{-1}\|x(k_{i+1})\| &\leq \|x(k_{i+1}-1)\| \leq ... \\ &\leq \|x(k_{i+1}-m_i)\| \leq \|x(k_i)\|. \end{aligned} \tag{13}$$

Now combining (8) and (12) we can conclude that the system is exponentially stable with stability degree of $\lambda = \sigma^{\frac{1}{max(m_i)}}$. ∎

Remark 1. *For a given (A, B) and F, it is possible to numerically compute the minimum m_i, which will be used for evaluating minimum arrival rate of the corresponding system.*

III. Controller Design

This section provides a method to design an exponentially stabilizing controller for the plant (1). Before that we present a lemma which has an important role in the design of controller.

Lemma 1. *(Schur complement [14]) For some constant symmetric matrix M and Q and constant matrix L, all of appropriate dimension, $Q > 0$ and $M + L^T Q^{-1} L < 0$ if and only if*

$$\begin{bmatrix} M & L^T \\ L & -Q \end{bmatrix} < 0 \tag{14}$$

Now assume that want to find a state-feedback controller F which guarantees the system to not trigger before some specific number of sampling instants. Let us introduce this maximum number of instants as $k^* = max(m_i)$, then (11) must be held for $\tilde{k} \in \{1, ..., k^*\}$. This set of inequalities are inappropriate for controller design procedure. It is possible to convert these inequalities to some linear matrix inequalities (LMI). As we know, $\|P\| \leq \alpha$ is equivalent to $P^T P \leq \alpha^2 I$ which is equivalent to $\lambda_{max}(P^T P) \leq \alpha^2$ (note that $P^T P$ is symmetric). Similarly, (11) can be expressed as follows

$$\lambda_{max}\Big((A^{\tilde{k}} + (\sum_{j=0}^{\tilde{k}-1} A^j)BF)^T (A^{\tilde{k}} + (\sum_{j=0}^{\tilde{k}-1} A^j)BF)\Big) \leq \sigma^2 I. \tag{15}$$

Using Schur lemma this inequality can be expressed as the following LMI

$$\begin{bmatrix} \sigma^2 I & (A^{\tilde{k}} + \big(\sum_{j=0}^{\tilde{k}-1} A^j\big)BF) \\ (A^{\tilde{k}} + \big(\sum_{j=0}^{\tilde{k}-1} A^j\big)BF)^T & I \end{bmatrix} \leq 0. \tag{16}$$

This LMI must be held for $\tilde{k} \in \{1, .., k^*\}$. Thus for a given (A, B), σ, and k^* it is possible to numerically compute F by solving k^* LMIs. Using these LMIs, it is possible to determine the minimum inter-event samples in the absence of disturbance. In the procedure of controller design it may seem that the proposed mechanism is conservative but in practice the triggering condition is implemented as (6) which is less conservative.

IV. Network Analysis

In this section, the so-called network calculus method is utilized to compute appropriate and less conservative parameters for a class of networks called guaranteed service networks. We propose a method to calculate the maximum delay induced to control system by the network. Furthermore, we will show that this induced delay is related to the control strategy because it is related to the arrival traffic of the control system.

At the end of this section we propose an algorithmic to co-design control system together with network.

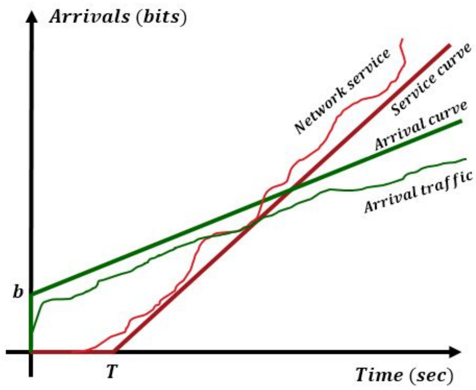

Fig. 2: Concept of arrival curve and service curve

A. Overview of Network Calculus

Network calculus is a theory of analyzing performance which is guaranteed in computer based systems such as communication networks. Two fundamental tools in network calculus are arrival curve and service curve as shown in Fig. 2 and are defined as following.

Definition 1 (Arrival curve)**.** An arrival curve $\alpha(t) = b + r.t$ is defined as an upper bound for arrival traffic $R(t)$ of a given node in which, b and r denote burst size and arrival rate respectively. $\alpha(t)$ is said to be arrival curve of $R(t)$ if and only if

$$\forall t \geq 0, s \geq 0 \quad and \quad s \leq t : R(t) - R(t-s) \leq \alpha(s). \tag{17}$$

This inequality means that the arrival traffic in any interval $[s, t]$ is less than $\alpha(t - s)$.

Definition 2 (Service curve)**.** A service curve $\beta(t) = r(t - D_l)^+$ and $(x)^+ = max(x, 0)$ is defined as a lower bound for the amount of arrival traffic that a network guarantees to transfer. R and D_l are know as the guaranteed bandwidth and network latency respectively.

Now consider a system S, with input function $R(t)$ and output function $R^*(t)$. Output data of S is input data affected by variable delay $d(t)$. $R(t)$ indicates an upper bound for arrival traffic in time interval $[0,\ t]$. At time t, $R(t) - R^*(t)$ is the amount of bits of data that arrive to the system but had not yet delivered to the output. Let $d(t)$ denote the delay at time t experienced by a bit of data which arrive at time t, then $R^*\big(t + d(t)\big) = R(t)$ and

$$d(t) = \inf\{T : T \geq 0 \text{ and } R(t) \leq R^*(t + T)\} \tag{18}$$

Network calculus provides a method for computation of maximum delay for an arbitrary system [10]. A network could be modeled as an interconnection of such systems and the data which is generated by sensors, controller or other parts of control system could be considered as input flow to a system S.

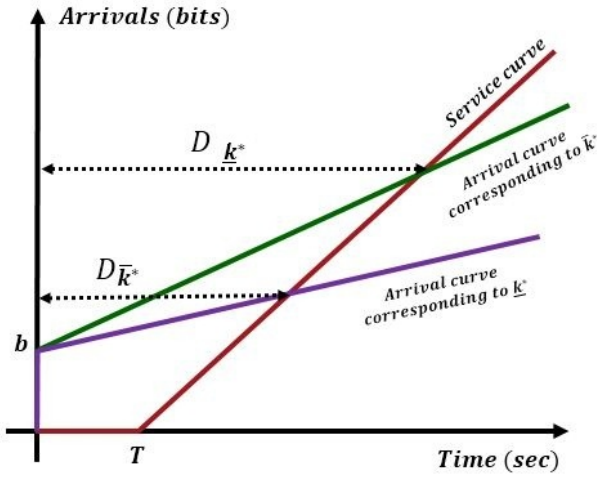

Fig. 3: Arrival and service curves and maximum delay

B. Maximum induced delay

Consider a flow $R(t)$ coming from the sampler connected to a sensor which may be limited by an event detector. $R(t)$ is constrained by $\alpha(t)$ which satisfies (17). Consider a system that provides at least a service $\beta(t)$. In control system design, most of the time a maximum limit on the delay is required for verification of system stability in the presence of variable delay. In this case, an upper bound on the network induced delay denoted by $h(\alpha, \beta)$ can be considered as

$$d(t) \leq h(\alpha, \beta), \tag{19}$$

where $h(\alpha, \beta)$ is the horizontal deviation between arrival curve and service curve [10], namely,

$$h(\alpha, \beta) = \sup_{s \geq 0}[\inf\{T : T \geq \ 0 \text{ and } \alpha(s) \leq \beta(s+T)\}]. \tag{20}$$

The maximum delay for two arrival curve is shown in Fig. 3.

V. Bandwidth allocation

This section consists of two subsections which corresponds to two different GTS allocation problems in NCSs. Both schemes assume the same network topology but their difference is in control side of view. In the first scheme, it is assumed that there is no other client in the network which means the network shall serve only the arrival traffic of the control loop. In this scheme our aim is to find the minimum number of GTSs that meets our demand from control performance view. We find the lower bound of GTSs so as to determine the amount of bandwidth which is available for future demand of the network.

In the second scheme, we have many other clients on the network and our aim is to design a controller based on Algorithm 1 in order to guarantee the stability of the control system and determine the bound of network induced delay to the control system. Then using this delay bounds it is possible to evaluate the stability of the control system in the presence of network delay.

A. Scheme 1: Free network

Assume a network that guarantees to serve a coming flow at a constant rate C_g and an input flow $R(t)$ from the event detector constrained by an arrival curve $\alpha(t)$ (satisfies (17)). Our aim is to find minimum guaranteed rate C_g which is required to serve arrival curve $\alpha(t)$ with a predetermined constraint on the upper bound of induced delay $D > 0$ to the control system. It is proved that the condition on guaranteed rate C_g is $C_g \geq C_D$ [10]. According to the nature of network calculus C_D is a deterministic value and is called effective bandwidth. As an example, in Section VI it will be investigated how C_D depends on parameters of IEEE 802.15.4 protocol.

The problem is to find minimum guaranteed rate C_g to achieve an upper limit for network induced delay D for a given arrival curve $\alpha(s)$ and service curve $\beta(s) = C_g(s-T)^+$ and $x^+ = max(x, 0)$ in which T denotes network latency. According to the nature of service curve, D couldn't be considered lower than T. Let us define D_e as

$$D_e = D - T \tag{21}$$

in which $D_e \geq 0$. So we wish to find a rate C_g such that $h(\alpha, \beta) \leq D$ which is equivalent to $\alpha(s) \leq \beta(s + D)$ for all $s \geq 0$. Furthermore, argument of β is positive because we assumed $D \geq T$ so

$$C_g \geq \sup_{s \geq 0} \frac{\alpha(s)}{s + D_e}. \tag{22}$$

Consider a linear discrete time system which is sampled with an appropriate sample rate T_s. Then evaluate the packet size generated by the sensor. For instance, each state variable occupies four bytes of data, thus a n-dimensional system occupies $4n$ bytes. Assume a guaranteed service network latency with maximum latency T. Algorithm 2 guarantees an upper bound for delay if there exist a stabilizing controller for at least $k^* = 1$. Furthermore effective bandwidth C_D shows the amount of bandwidth usage, and Thus, the amount of available bandwidth for future usage is known. To compensate the effect of delay it is possible to estimate the triggering time D second sooner in the event detector. This method is investigated in [7].

Algorithm 2 Computation of GTS Allocation

Step 1: Put $k^* = 1$
Step 2: Design the controller base on section III
Step 3: Compute arrival curve $\alpha(s) = T_s.(K^*)^{-1}.s + b$
Step 4: Calculate the effective bandwidth as $C_D \geq \sup_{s \geq 0} \frac{\alpha(s)}{s+D_e}$.
Step 5: Compute service curve $\beta(s) = C_D(s-T)^+$ and the network induced delay $h(\alpha, \beta)$ using (20)
Verify the required performance of the system by simulation, if it's not satisfied, goto 6.
Step 5: Increase k^* and repeat from step 2.
Step 6: The controller which was designed corresponding to $k^* - 1$ is the solution.

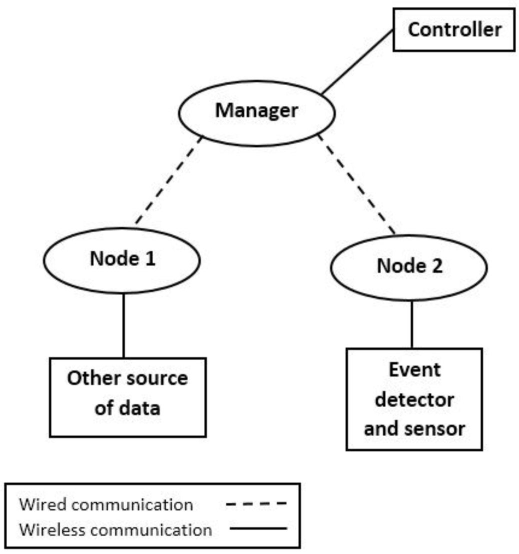

Fig. 4: Network architecture of scheme 2

B. Scheme 2: Partially used network

Without loss of generality, in this scheme we consider a network architecture consisting of two nodes. As we mentioned, network calculus models the input traffic as arrival curves and at each point of the network, the input traffic can be modeled as a single source.

Now consider network N shown in Fig. 4 in which node 1 aggregates data from various source and delivers them to the network. The problem is to find an event-based controller which can stabilize a plant and communicate thorough a shared network N from the access point node 2. For this objective we convert network N to a multiplexer. The multiplexer has two input and one output links. In network calculus terminology, multiplexer is a block of model which merges the arriving streams on the input links onto the output link [8]. Multiplexer is described by arrival curves α_1 into link 1 and α_1 into link 2 and service curve β. This service curve guarantees service of the multiplexer for one of the inputs while the other input is not generating data. The following Lemma is a direct result of [15] which is proved in [16]

Lemma 2. *Assume a multiplexer with arrival curve $\alpha_2 = b_2 + r_2.s$ into link 2 and service curve $\beta = C_g(s-T)^+$, $(x)^+ = max(0,x)$. The guaranteed service curve of the multiplexer for link 1 is*

$$B^1_{MUX} = R_{MUX}(s - T_{MUX})^+, \tag{23}$$

in which

$$R_{MUX} = R - r_2, \tag{24}$$

and

$$T_{MUX} = \frac{b_2 + r_2.(b_2/R)}{R_{MUX}} + T. \tag{25}$$

Using results of Lemma 2 and (20) it is possible to compute the induced delay in the presence of other client in a shared communication link. Let $\hat{C}_D$ denote the available bandwidth of the network. The arrival curve should satisfy

$$\hat{C}_D \geq \sup_{s\geq 0} \frac{\alpha(s)}{s + D_e}. \tag{26}$$

Assuming arrival curve as $\alpha(s) = T_s.(k^*)^{-1}.s + b$ the lower bound of k^* is $\underline{k}^* = \left\lceil \frac{T_s}{C_D} \right\rceil$. A similar procedure as in Algorithm 2 could be imagined to design the stabilizing controller for this scheme with respect to minimum allowable inter-event time $\underline{k}^*$.

VI. Case Study

In this section we are going to illustrate the proposed method by simulation. We start by introducing IEEE 802.15.4 standard [17] as an appropriate guaranteed service protocol for networked control system. The we apply the result of this paper together with [13] to this protocol. Finally we will compare our method with two other discrete time event-based method of [3] and [2].

A. Overview of 802.15.4 Standard

The IEEE 802.15.4 standard is known as a multiple access control (MAC) and physical layer (PHY) standard for low cost, low power, and low data rate wireless personal area networks (WPANs). In this paper We assume beacon-enabled mode of IEEE 802.15.4 standard.

Figure 5 illustrates the concept of beacon interval and superframe. A superframe consists of two parts, contention free period (CFP) and contention access period (CAP). In the CAP nodes are content to access the network, so it has a stochastic nature. In contrast to CAP, CFP contains guaranteed time slots which are allocated to different nodes. So the MAC layer guarantees a specified time interval for each node to access the network and during this time the other nodes can not send their data over the network. In this mode, it is necessary to synchronize all nodes with each other, this is led to a synchronization frame called beacon frame which is shown in Fig. 5. Beacon Interval is determined by personal area network (PAN) coordinator by sending beacon frames at the start and the end of beacon interval. The duration of beacon interval (BI) and superframe can be modified using two parameters, the beacon order (BO) and the superframe order (SO). The beacon interval is defined as

$$\begin{aligned} BI =& aBaseSuperframeDuration * 2^{BO}, \\ &\text{for } \; 0 \leq BO \leq 14, \end{aligned} \tag{27}$$

and superframe duration is defined as

$$\begin{aligned} SD =& aBaseSuperframeDuration * 2^{SO}, \\ &\text{for } \; 0 \leq SO \leq BO \leq 14, \end{aligned} \tag{28}$$

in which $aBaseSuperframeDuration$ corresponds to minimum length of the superframe which is fixed to 960 symbols by IEEE 802.15.4 standard. A symbol is equal to 4 bits, so a superframe at least consists of 430 Bytes, shared with 16 GTSs, so in this case the length of a GTS is 30 Bytes.

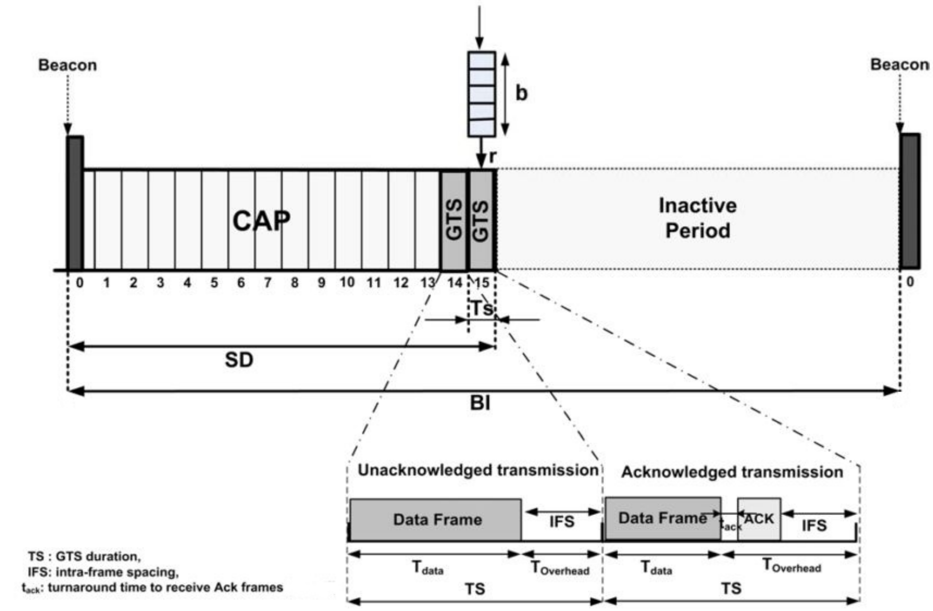

Fig. 5: Concept of superframe and GTS [13]

B. The survive curve of a GTS

Let T_s denote the time slot duration in the superframe. A superframe consists of sixteen GTSs, so assuming $SO = 0$, for allocation of one GTS,

$$T_s = aBaseSuperframeDuration * \frac{1}{16}. \tag{29}$$

As illustrated in Fig. 5, T_s is divided into two parts

$$T_s = T_{data} + T_{idle}, \tag{30}$$

in which T_{idle} corresponds to the time when the node does not send data due to lack of data or acknowledgment (ACK) frame or etc, which is

$$T_{idle} = T_{overhead} + T_{wasted}, \tag{31}$$

Now we define the maximum latency T that a node may wait to transmit its data over the network. This is worst case for a data frame which occurs when the node needs to send data just after the end of its allocated GTS. So

$$T = BI - T_s. \tag{32}$$

and R is the guaranteed bandwidth of one GTS, which is the rate of bits sent in each beacon interval. So

$$R = \frac{T_{data}.C_g}{BI}. \tag{33}$$

Now using these two parameters we can determine the service curve of a GTS as $\beta_{R,T}(t) = R(t-T)^+$, such that $(x)^+ = max(0, x)$. This result can be extended for more than one GTS allocation as

$$R_n = n.\frac{T_{data}.C_g}{BI}, \tag{34}$$

and

$$T = BI - n.T_s. \tag{35}$$

So it is possible to find an appropriate number of GTSs, according to maximum acceptable delay.

C. Simulation Results

Consider a linear, unstable, discrete time system described by

$$x(k+1) = Ax(k) + Bu(k), \tag{36}$$

in which

$$A = \begin{bmatrix} 0.1 & 1.2 \\ 0.007 & 1.05 \end{bmatrix}$$

and

$$B = \begin{bmatrix} 300 & 200 \\ 0.5 & 0.001 \end{bmatrix}$$

We assume that system is discretized with non-pathological sampling rate $T_s = 0.05$Sec. We also assume that all of the states are observable. This system is also investigated in [2] and [18]. In order to design a stabilizing controller for the system (36) we use Algorithm 2 and the result for $k^* = 3$ is

$$K = \begin{bmatrix} 0.365 & -0.549 \\ -0.549 & 0.819 \end{bmatrix}. \tag{37}$$

Simulation has been performed starting from the initial state $x_0 = [0.2, \ -0.5]$ shown in Fig. 6. Fig. 7 shows the event time and event intervals for $k \in [0, \ 100]$. As shown in th figure the minimum inter event time is 4 samples. It does not conflict with our assumption, because we wanted to design a system with lower bound on the minimum inter-event time which was corresponding to $k^* = 3$. It is possible to calculate maximum

TABLE I: IEEE 802.15.4 parameters

Parameter	Value
SO	1
BO	2
Maximum MAC frame size	150 bits

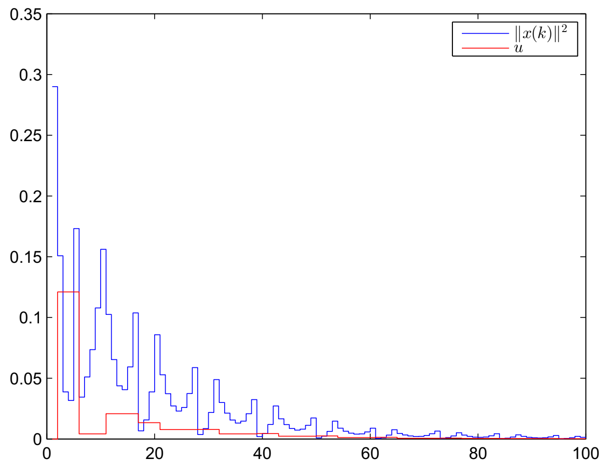

Fig. 6: Sum of square error and manipulated value

delay and allocated bandwidth for a given arrival curve using a MATLAB toolbox introduced [19]. Assume IEEE 802.15.4 parameters of Table I. So

$$SD = 30.72 \ (ms),$$

$$BI = 61.44 \ (ms).$$

System of (36) has two states and each states as a float number occupies 4 Bytes. So the burst size of the arrival curve coming from event detector is 64 bits. We consider $b = 150$ bits and let the other 86 bits be overhead of control system in each packet of data. For proposed $k^* = 3$, $r = 1$ kbps. It is important to note that minimum inter-event time of proposed algorithm is 4 samples but we use $k^* = 3$ which is guaranteed in the controller design procedure.

Using Algorithm 2 for system (36) and aforementioned setup of controller and network yield a less conservative bandwidth allocation is guaranteed with allocation of 2 GTSs which is equal to $C_D = 2.34$ kbps and an upper bound for induced delay is $D = 0.12$ (s).

Triggering times of the methods proposed in [2], [18] and our method is compared in table II. According to this table, the methods proposed in [2], and [18] need more guaranteed bandwidth of the network. Furthermore an upper bound for inter-event time is not considered during the controller design in their method.

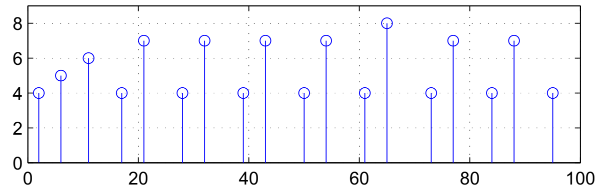

Fig. 7: Event instants and inter-event time

TABLE II: Inter-event comparison between various approaches

	Inter event time		
Approach	Minimum	Maximum	Sent sample
[2]	3	4	33%
[18]	1	11	13%
Proposed	4	8	20%

Conclusion

In this paper, Event-triggered Control (ETC) design for wireless network have been investigated. A method was proposed to design the controller so as to guarantee a designable minimum time between two consecutive event. Network calculus is used as analyzing tool on wireless networks and a theorem is proved to guarantee exponentially stability of the control system. Furthermore, a method is provided to compute the required bandwidth for the control system and the bound of delay is evaluated. The result of this paper is investigated through a case study which consists of a discrete-time process and an event based controller for the process over IEEE 802.15.4 standard.

Acknowledgment

This research was supported by National Iranian Gas Company.

References

[1] P. Tabuada, "Event-Triggered Real-Time Scheduling of Stabilizing Control Tasks," *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1680–1685, 2007.

[2] A. Eqtami, D. V. Dimarogonas, and K. J. Kyriakopoulos, "Event-triggered control for discrete-time systems," *Proceedings of the 2010 American Control Conference*, pp. 4719–4724, Jun. 2010.

[3] X. Meng and T. Chen, "Event based agreement protocols for multi-agent networks," *Automatica*, vol. 49, no. 7, pp. 2125–2132, Jul. 2013.

[4] W. P. M. H. Heemels, M. C. F. Donkers, and A. R. Teel, "Periodic event-triggered control based on state feedback," *IEEE Conference on Decision and Control and European Control Conference*, pp. 2571–2576, Dec. 2011.

[5] W. P. M. H. Heemels and M. C. F. Donkers, "Model-based periodic event-triggered control for linear systems," *Automatica*, vol. 49, no. 3, pp. 698–711, Mar. 2013.

[6] A. Cervin and T. Henningsson, "Scheduling of event-triggered controllers on a shared network," *47th IEEE Conference on Decision and Control*, no. 2, pp. 3601–3606, 2008.

[7] J. Araújo, M. Mazo, A. Anta, P. Tabuada, S. Member, and K. H. Johansson, "System Architectures , Protocols and Algorithms for Aperiodic Wireless Control Systems," *IEEE Transactions on Industrial Informatics.*, vol. 10, no. 1, pp. 175–184, 2014.

[8] R. L. Cruz, "A calculus for network delay, Part I; Network element in isolation," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 114–131, 1991.

[9] R. L. cruz, "A calculus for network delay, Part II; Network analysis," *IEEE Transactions on Information Theory*, vol. 37, no. January, pp. 132–141, 1991.

[10] J.-Y. L. Boudec, "Application of network calculus to guaranteed service networks," *IEEE Transactions on Information Theory*, vol. 44, no. 3, pp. 1087–1096, 1998.

[11] M. Fidler, "A Survey of Deterministic and Stochastic Service Curve Models in the Network Calculus," *IEEE Communications Surveys and Tutorials*, vol. 12, no. 1, pp. 59–86, 2010.

[12] C. Li and A. Burchard, "A Network Calculus With Effective Bandwidth," *IEEE/ACM Transactions on Networking*, vol. 15, no. 6, pp. 1442–1453, 2007.

[13] A. Koubaa, M. Alves, E. Tovar, R. Antonio, and B. D. Almeida, "Technical Report: GTS Allocation Analysis in IEEE 802 . 15 . 4 for Real-Time Wireless Sensor Networks."

[14] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, "Matrix Inequalities in System and Control Theory," *Applied Mathematics, Philadelphia, PA, USA*, vol. 15, 1994.

[15] A. Koubaa, M. Alves, and E. Tavor, "Technical Report: Modelling and Worst-Case Dimensioning of Cluster-Tree Wireless Sensor Networks," no. October, 2006.

[16] L. Lenzini, E. Mingozzi, and G. Stea, "Delay bounds for FIFO aggregates : a case study," *Computer Communications*, vol. 28, no. 3, pp. 287–299, 2005.

[17] IEEE802.15.4, *IEEE Standard for Information technology: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs)*, 2006.

[18] X. Meng and T. Chen, "Event-driven communication for sampled-data control systems," *American Control Conference*, pp. 3002–3007, Jun. 2013.

[19] P. Jurpík, A. Koubâa, M. Alves, E. Tovar, and Z. Hanzálek, "A Simulation Model for the IEEE 802 . 15 . 4 Protocol : Delay / Throughput Evaluation of the GTS Mechanism," *15th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, pp. 109–116, 2007.

References

[1] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems,"*PROCEEDINGS-IEEE*, Vol. 95, 2007.

[2] G. Pin, T. Parisini, "Networked Predictive Control of Uncertain Constrained Nonlinear Systems: Recursive Feasibility and Input-to-State Stability Analysis," *IEEE Transactions on Automatic Control*, Vol. 56, No. 1, pp. 72-87, 2011.

[3] W. Zhang, M. S. Branicky, and S. M. Philips, "Stability of Networked Control System," *IEEE Control Systems*, Vol. 21, pp. 84-99, 2001.

[4] J. H. Lee, "Model predictive control: review of the three decades of development," *International Journal of Control, Automation and Systems*, Vol. 9, pp. 415-424, 2011.

[5] E. F. Camacho, and C. Bordons, "Model Predictive Control," Springer-Verlag GmbH, 2004.

[6] A.Pytel, S. Kozak, "Modelling and effective predictive control of gas turbine process," *IEEE Control Conference (ICCC)*, pp. 469-474, 2014.

[7] G.P. Liu, J. X. Mu, D. Rees, and C. Chai, "Design and stability analysis of networked control systems with random communication time delay using the modified MPC," *International Journal of Control*, Vol. 79, No. 4, pp. 288-297, 2006.

[8] S. Chai, G. P. Liu, D. Rees, and Y. Xia, "Design and practical implementation of internet-based predictive control of a servo system," *IEEE Transaction on Control Systems Technology*, Vol. 16, No. 1, pp. 158-168, 2008.

[9] P. G. Liu, Y. Xia, J. Chen, "Networked Predictive Control of Systems with Random Network Delays in Both Forward and Feedback Channels," *IEEE Transaction on Industrial Electronics*, Vol. 54, pp. 1282-1297, 2007.

[10] P. L. Tang, C. W. De Silva, "Compensation for transmission delays in an Ethernet-based control network using variable-horizon predictive control," *IEEE Transaction on Control Systems Technology*, Vol. 14, No. 4, pp. 707718, 2006.

[11] I. Jurado, P. Milln, D. Quevedo, F. R. Rubio, "Stochastic MPC with applications to process control," *International Journal of Control*, Vol. 0, No. 0, pp. 1-16, 2014.

[12] H. Li, Y. Shi, "Network-Based Predictive Control for Constrained Nonlinear Systems with Two-Channel Data dropouts," *IEEE Transaction on Industrial Electronics*, Vol. 61, No. 3, 2014.

[13] M. Farina, A. Gaugliardi, C. Sandroni, and R. Scattolini, "Model predictive control of voltage profiles in MV networks with distributed generation," *Control Engineering Practice*, Vol. 34, pp. 18-29, 2015.

[14] M. V. Kothare, V. Balakrishnan, and M. Morari, "Robust constrained model predictive control using linear matrix inequalities," *Automatica*, Vol. 32, pp. 1361-1379, 1996.

[15] S. Boyd, E. Elghaoui, E. Feron, and V. Balakrishnan, "Linear matrix inequalities in system and control theory," SIAM, Philadelphia, 1994.