Distributed event-triggered control
with network delays and packet losses

M. Guinaldo, D. Lehmann, J. Sánchez, S. Dormido, K. H. Johansson

Abstract— This paper presents a distributed event-based control approach to cope with communication delays and packet losses affecting a networked dynamical system consisting of N linear time-invariant coupled systems. Two communication protocols are proposed to deal with these communication effects. It is shown that both protocols preserve the system stability in the sense that the state of every subsystem converges to a small region around the origin if the delay and the number of packet losses are bounded. Analytical expressions for the delay bound and the maximum number of consecutive packet losses are derived. Simulations illustrate the results.

I. INTRODUCTION

Event-triggered control has been developed to reduce the need for feedback while guaranteeing certain levels of performance and it has been proposed in Networked Control Systems (NCS) for allowing a more efficient usage of the limited network bandwidth [1]-[5]. There is a natural interest in applying these techniques to decentralized NCS since the design of a centralized controller is inappropiate for a large number of subsystems due to the overload of the network by requesting and sending information from/to each node.

There are some recent contributions on distributed event-triggered control, see [6]-[12]. The basic idea is that each subsystem (also called agent or node) decides when to transmit the measurements based only on local information. Different approaches can be found in the literature such as deadband control [2], Lyapunov approaches to event-based control [3],[13] or self-triggered control [14], [15].

Even though the problem of limited bandwidth can be faced by reducing the communication due to event-based control, network delays and packet losses cannot be avoided [16]. However, up to now, only few papers have considered the effect of these issues on event-based control and even less on its decentralized implementation. Early papers [1], [4] study simple stochastic systems and investigate the event-based control performance in dependence upon the medium access mechanism applied.

In [17], delays are compensated by a control input generator and an event generator both of which emulate the

continuous-time state feedback controller, and by measuring the delay. However, this scheme is difficult to implement in a distributed scheme since measuring transmission delays between two nodes requires clock synchronization in the entire network. Another paper that studies the delay effect combining model-based and event-triggered control in a single loop is [18].

In [8], a thorough study of event triggering in distributed NCS subject to delays and packet losses is carried out. The design of the event triggering threshold is based on Lyapunov methods. The main drawback of this work is that a lower bound for the broadcasting period, i.e., the difference between successive broadcasting times, cannot be guaranteed when the system approaches the origin. This might cause severe problems since it would require the detection of events and transmission of data infinitely fast.

In this paper the effect of network delays and packet losses is analyzed by considering a distributed event-based control design for interconnected linear systems. It extends the previous paper [10] in which the stability of the proposed trigger mechanism in an ideal network is investigated. The analysis is focussed on determining bounds for the delay and the number of successive packet dropouts that ensure stability and guarantee a certain level of performance. Moreover, a lower bound for the inter-event times is derived. Two different transmission protocols are proposed. The first one preserves *state consistency* [8], which means that a broadcasted state is updated synchronously in each neighboring agent. This restriction is relaxed by the second protocol allowing the neighbors to use different versions of the broadcasted states.

The remainder of the paper is organized as follows. Section II provides the system desciption. The proposed communication protocols are described in Section III. Section IV analyzes the system performance and provides the main results of the paper. The simulation results are presented in Section V, and finally, the conclusions and the future works conclude the paper.

II. PRELIMINARIES

A. System description

Consider a system of N linear time-invariant subsystems. The dynamics of each subsystem are given by:

$$\dot{x}_i(t) = A_i x_i(t) + B_i u_i(t) + \sum_{j \in N_i} H_{ij} x_j(t), \qquad (1)$$

$\forall i = 1, ..., N$, where N_i is the set of neighbors of subsystem i, i.e., subsystems with which it can communicate, and H_{ij} is the interaction term between agent i and agent j. We assume

that neighboring is a symmetric relation, and therefore $H_{ij} = H_{ji}$. The state x_i of the ith agent has dimension n_i, u_i is the m_i-dimensional local control signal of agent i, and A_i, B_i and H_{ij} are matrices of appropriate dimensions.

Let us assume that the state x_i is measurable. Each agent i sends its state through the network to its neighbors when an event is triggered. The time instances at which the agent i broadcast its state are denoted by $\{t_k^i\}_{k=0}^{\infty}$, where $t_k^i < t_{k+1}^i$ for all k, and $\tilde{x}_i$ is the broadcasted state.

B. Ideal network

Considering an ideal network scenario, the time instances at which the event is generated, the state $\tilde{x}_i$ is sent in one node and received in all neighboring nodes are the same. Thus, the control given by

$$u_i(t) = K_i \tilde{x}_i(t) + \sum_{j \in N_i} L_{ij} \tilde{x}_j(t), \quad \forall i = 1, ..., N \quad (2)$$

where K_i is the feedback gain for the nominal subsystem i and L_{ij} is a set of decoupling gains, $\tilde{x}_i$ is a piecewise constant function in the inter-event times since $\tilde{x}_i(t) = x_i(t_k^i)$, $\forall t \in [t_k^i, t_{k+1}^i]$ and $\forall i = 1, \ldots, N$.

By defining the error between the current state x_i and the latest broadcasted state $\tilde{x}_i$ as $e_i(t) = \tilde{x}_i(t) - x_i(t)$, denoting $A_{K,i} = A_i + B_i K_i$ and assuming that the *matching condition* $(B_i L_{ij} = H_{ij})$ is satisfied, (1) can be rewritten according to $\dot{x}_i(t) = A_{K,i} x_i(t) + B_i K_i e_i(t) + \sum_{j \in N_i} B_i L_{ij} e_j(t)$. Because $\tilde{x}_i(t)$ is piecewise constant between events, it holds

$$\dot{e}_i(t) = -A_{K,i} x_i(t) - B_i K_i e_i(t) - \sum_{j \in N_i} B_i L_{ij} e_j(t) \quad (3)$$

in each interval $t \in [t_k^i, t_{k+1}^i)$.

Assumption 1: We assume that $A_{K,i}$, $i = 1, \ldots, N$ is diagonalizable so that the Jordan form of $A_{K,i}$ is diagonal and its elements are the eigenvalues of $A_{K,i}$, $\lambda_k(A_{K,i})$ $k = 1, \ldots, n$. This assumption facilitates the calculations, but the extension to general Jordan blocks is straightforward.

Finally, let us denote by $\lambda_{max}(A_{K,i})$ the maximum real part of the eigenvalues of $A_{K,i}$, and let $k_{0,i}$ be the positive constant $k_{0,i} = \|V_i\| \|V_i^{-1}\|$, where V_i is the matrix of the eigenvectors of $A_{K,i}$ and $\| \cdot \|$ is the Euclidian norm.

C. Trigger function

The occurrence of an event is defined by trigger functions f_i which only depend on local information of agent i and take values in $\mathbb{R}$. The sequence of broadcasting times t_k^i are determined recursively by the event trigger function as $t_{k+1}^i = \inf\{t : t > t_k^i, f_i(t) > 0\}$.

We consider static trigger functions defined by

$$f_i(e_i(t)) = \|e_i(t)\| - c, \ c > 0. \quad (4)$$

III. TRANSMISSION PROTOCOL

This section discusses the protocol requirements that preserve a certain level of performance when network delays and packet losses occur in the transmission of data. Assuming an ideal network and according to (4), the error in the subsystem i is upper bounded by $\|e_i(t)\| \leq c$ in each subsystem

$i \in N_i$ with $j \in N_i$. As this property cannot be guaranteed when dealing with delays or packet losses, it forces the consideration of reasonable assumptions for the analysis in the non-ideal network scenario.

Let us first introduce some notation.

Definition 2: We denote by $\tau_k^{i \to j}$ the delay in the tranmission of the state $x_i(t_k^i)$ of agent i to its neighbor j, $j \in N_i$, at time t_k^i, and $\bar{\tau}_k^i = \max\{\tau_k^{i \to j}, j \in N_i\}$.

Definition 3: We denote by $P_{i \to j}^k$ the number of sucessive packet losses in the tranmission of the state $x_i(t_k^i)$ of agent i to its neighbor j, $j \in N_i$, at time t_k^i, and by P_i^k the maximum of $P_{i \to j}^k$ for all $j \in N_i$.

We next introduce the basic assumption that imposes constraints on delays and packet dropouts.

Assumption 4: We assume that the maximum delay and the number of successive packet dropouts which occur in the tranmission of information from subsystem i to its neighbors $j \in N_i$, denoted by $(\tau^\star)^i$ and $P_i^\star$ respectively, are such that no event is generated before all the neighbors have successfully received the broadcasted state $\tilde{x}_i$.

The second important consideration is that the sender i knows that the data has been successfully received by j by getting an acknowledgment signal (ACK). If an ACK is not received before a *waiting time* denoted by T_W^i, the packet is discarded. How to determine T_W^i is analyzed later, but it seems logical to set this value larger than the maximum delay. If agent i has not received an ACK of the reception of all the neighbors after the waiting time T_W^i, we propose two alternatives that we denote *Protocol 1* and *Protocol 2*.

1) Protocol 1: The state at $t_k^i + T_W^i$ is broadcasted again to all the neighbors. If after waiting T_W^i an ACK is not received from all $j \in N_i$, the retranmission takes place again, and so on. This process can occur at most $P_i^\star + 1$ times. Once all the neighbors have successfully received the data, agent i sends a *permission signal* (PERM) so that the previously transmitted data can be used to update the control law (2). Both signals ACK and PERM are assumed to be delivered with a delay approximated by zero over a reliable channel. A very similar protocol is presented in [8]. As stated there, the reason to the use a PERM signal and to retransmit the state to all the neighbors instead of only retransmitting to those from which an ACK signal has not been received, is to preserve the *state consistency* mentioned in the introduction. Since the broadcasted data is not valid until a PERM signal is received from agent i, all the neighboring agents update the value at the same time and therefore, the value of the error e_i is the same in all nodes. This allows to define stack vectors for the state and the error signal so that the stability of the overall system can be studied as in the ideal network case [10].

2) Protocol 2: The previous protocol simplifies the analysis but it has some drawbacks. First, all nodes in the neighborhood have to wait for the slower connection (larger delay) to process the received data. Secondly, *Protocol 1* may involve unnecessary transmission, since even if an agent j received a measurement, if other neighbor did not, the broadcast takes place with an updated measurement $\forall j \in N_i$. Finally, the ACK signal is vastly used in network protocols

to guarantee reliability of packet transfers, but the PERM demands a more involved communication protocol. In order to overcome these drawbacks, in the new protocol:

- Agent i waits T_W^i to get acknowledgements from the neighbors. To those agents $j \in N_i$ from which it did not receive the ACK signal, it sends the state $x_i(t_k^i + T_W^i)$ at time $t_k^i + T_W^i$. Agent i may transmit before the next event at most $P_i^\star + 1$ times.
- Let us denote by $\mathcal{N}_i(t) \subseteq N_i$ the agents to which the subsystem i transmits information at time t. In contrast to *Protocol 1*, agent i only transmit a new measurement to those agents from which it has not received the ACK signal. If the last event occurred at time t_k^i and $t \in [t_k^i, t_{k+1}^i)$, thus

$$\forall j \in N_i, \notin \mathcal{N}_i(t) \quad \exists t_k^{i \to j} : t_k^i \le t_k^{i \to j} < t,$$

where $t_k^{i \to j}$ is the time of the successful broadcast to agent j. Hence if at time t the node j is not in $\mathcal{N}_i(t)$ it means that it has correctly received a broadcasted state after the occurrence of the last event and it has confirmed this reception with an ACK signal.
- The number of consecutive packet losses and the network delay are upper-bounded for each agent i, according to Assumption 4. Thus, it must hold that $\mathcal{N}_i((t_{k+1}^i)^-) = \emptyset$, where $(t_{k+1}^i)^-$ refers to the instant time before t_{k+1}^i. I.e., all neighbors have successfully received x_i before the next event occurance.

In order to clarify the difference between both protocols, a simple example is given in Fig. 1. A system with two agents is depicted. Assume that Agent 1 detects an event at time t_k^1 and broadcast its state $x_1(t_k^1)$ to its neighbor Agent 2. The tranmission is delayed by τ_k^1 and Agent 2 sends then the ACK signal. In the scenario of Protocol 1 (Fig. 1a), the PERM signal is sent after receiving ACK (both assumed to be sent and received instantaneously) and both agents update the broadcasted state in the control law at the same time $t_k^1 + \tau_k^1$. Thus, $\tilde{x}_1$ takes the same value at any time in both agents and, hence, $e_1(t)$ is the same in the dynamics of Agent 1 and 2. For the Protocol 2 (Fig. 1b), the update in Agent 1 is applied inmediately at time t_k^1, whereas the receiver updates the state information at time $t_k^1 + \tau_k^{1 \to 2}$ (τ_k^1 and $\tau_k^{1 \to 2}$ are the same). Thus, in the interval $[t_k^1, t_k^1 + \tau_k^{1 \to 2})$ the broadcasted state $\tilde{x}_1$ has different values in the two nodes and consequently the error e_1 considered in Agent 1 differs from the error affecting the dynamics of Agent 2. Note that Agent 2 does not monitor e_1 since it only knows the state of Agent 1 at event times. It is drawn in the figure to clarify the difference between the two protocols.

IV. PERFORMANCE ANALYSIS

We firstly investigate the performance of the event-based control obtained by using *Protocol 1*. After that, we extend the results to the situation which uses *Protocol 2*. Finally, we discuss an extension to time-dependent trigger functions.
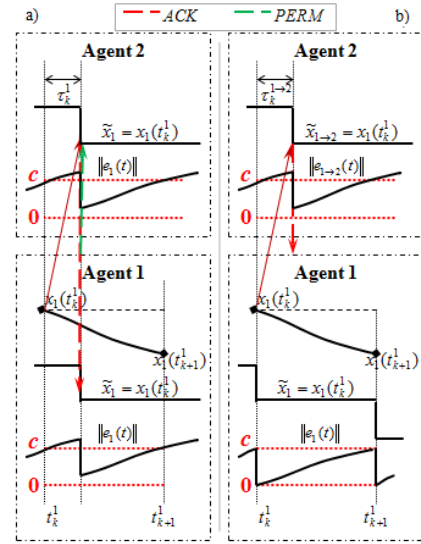

Fig. 1. Update mechanism of a) Protocol 1 and b) Protocol 2

A. Properties of event-triggered control using Protocol 1

Let us first assume that the communication can only experience delays but no packet dropouts.

1) Communication with delays:

Proposition 5: Let us consider trigger functions of the form (4) and *Protocol 1*. If Assumption 4 holds, the error of any subsystem i is upper bounded by $\|e_i(t)\| < 2c$.

Proof: Assume that the last event occurred at time t_k^i and that the maximum transmission delay to its neighbors is $\bar{\tau}_k^i$. From Assumption 4, it follows that

$$\left\| \int_{t_k^i}^{t_k^i + \bar{\tau}_k^i} \dot{e}_i(s)ds \right\| = \|e_i(t_k^i + \bar{\tau}_k^i) - e_i(t_k^i)\| < c, \quad (5)$$

has to be satisfied (see (4)) because no event is generated in the time interval $[t_k^i, t_{k+1}^i)$. Since an event has occurred at time t_k^i $\|e_i(t_k^i)\| = c$ holds and, thus, $\|e_i(t_k^i + \bar{\tau}_k^i)\| < 2c$ has to hold, which is valid for any time t. ∎

Theorem 6: If the network delay is upper bounded by

$$(\tau^\star)^i = \frac{c}{\|A_{K,i}\|\|k_{0,i}\|\|x_i(0)\| + \mu_i\left(1 + \frac{\|A_{K,i}\|\|k_{0,i}\|}{|\lambda_{max}(A_{K,i})|}\right)2c}, \quad (6)$$

where $\mu_i = \|B_iK_i\| + \sum_{j \in N_i} \|B_iL_{ij}\|$, then any broadcasted state $\tilde{x}_i$ of any subsystem $i \in 1, \ldots, N$ is successfully received by the neighbors $j \in N_i$ before a new event occurs. Hence, the inter-event times are lower bounded $t_{k+1}^i - t_k^i \ge (\tau^\star)^i$. Moreover, for all initial conditions $x_i(0)$ and $t > 0$ it holds

$$\|x_i(t)\| \le k_{0,i}\left(\frac{\mu_i 2c}{|\lambda_{max}(A_{K,i})|} + e^{-|\lambda_{max}(A_{K,i})|t}\left(\|x_i(0)\| - \frac{\mu_i 2c}{|\lambda_{max}(A_{K,i})|}\right)\right). \quad (7)$$

Proof: In order to prove the theorem, let us assume that Assumption 4 holds.

The analysis will derive an upper bound for the delay which preserves this assumption. The error in the time interval $[t_k^i, t_k^i + \bar{\tau}_k^i)$ is given by $e_i(t_k^i + \bar{\tau}_k^i) - e_i(t_k^i) = x_i(t_k^i) -$

$x_i(t_k^i + \bar\tau_k^i)$, since the broadcasted state $\tilde x_i$ is not updated in any agent before the time instance $t_k^i + \bar\tau_k^i$ according to *Protocol 1*, so that $\tilde x_i(t_k^i + \bar\tau_k^i) = \tilde x_i(t_k^i) = x_i(t_{k-1}^i)$ holds. This yields $e_i(t_k^i + \bar\tau_k^i) - e_i(t_k^i) = \big(I - e^{A_{K,i}\bar\tau_k^i}\big)x_i(t_k^i) + \int_0^{\bar\tau_k^i} e^{A_{K,i}s}\big(B_i K_i e_i(s) + B_i \sum_{j\in N_i} L_{ij}e_j(s)\big)ds$, based on which the upper bound for the delay $\bar\tau_k^i$ can be derived as $(\tau^\star)_k^i = \arg\min_{\bar\tau_k^i \geq 0}\{\|(I - e^{A_{K,i}\bar\tau_k^i})x_i(t_k^i) +$

$\int_0^{\bar\tau_k^i} e^{A_{K,i}s}\big(B_i K_i e_i(s) + B_i \sum_{j\in N_i} L_{ij}e_j(s)\big)ds\| = c\}$.

Note that this bound depends on $x_i(t_k^i)$. In order to guarantee the existence of the bound for the delay, we need to prove that the state is bounded for any t_k^i. The state at any time is given by $x_i(t) = e^{A_{K,i}t}x_i(0) + \int_0^t e^{A_{K,i}(t-s)}\big(B_i K_i e_i(s) + B_i \sum_{j\in N_i} L_{ij}e_j(s)\big)ds$. The error is bounded by $\|e_i(t)\| < 2c, \forall i$ by Proposition 5. Thus, a bound on $x_i(t)$ can be calculated following the methodology of [10] as (7).

Note that (7) is upper bounded by $\|x_i(t)\| \leq k_{0,i}\big(\frac{\|B_i K_i\|2c + (\sum_{j\in N_i}\|B_i L_{ij}\|)2c}{|\lambda_{max}(A_{K,i})|} + \|x_i(0)\|\big)$, for any t.

In order to derive an upper bound for the delay for any t, we recall that (3) holds in the interval $t \in [t_{k-1}^i + \bar\tau_{k-1}^i, t_k^i + \bar\tau_k^i)$ for any two consecutive events t_{k-1}^i, t_k^i, and, hence, it particular holds in the subinterval $[t_k^i, t_k^i + \bar\tau_k^i) \subset [t_{k-1}^i + \bar\tau_{k-1}^i, t_k^i + \bar\tau_k^i)$. With $\|\dot e_i(t)\| = \|A_{K,i}x_i(t) + B_i K_i e_i(t) + \sum_{j\in N_i} B_i L_{ij}e_j\| \leq \|A_{K,i}\|\|x_i(t)\| + \|B_i K_i\|\|e_i(t)\| + \sum_{j\in N_i}\|B_i L_{ij}\|\|e_j\|$, the bound on the state $x_i(t)$ given by (7) and Proposition 5, it follows from (5) that

$$\|e_i(t_k^i + \bar\tau_k^i) - e_i(t_k^i)\| \leq \Big(\|A_{K,i}\|k_{0,i}\big(\|x_i(0)\| +$$
$$\frac{(\|B_i K_i\| + \sum_{j\in N_i}\|B_i L_{ij}\|)2c}{|\lambda_{max}(A_{K,i})|}\big) + \big(\|B_i K_i\| + \sum_{j\in N_i}\|B_i L_{ij}\|)2c\Big)\bar\tau_k^i.$$

If the delay for any event k in the agent i is upper bounded by (6), Assumption 4 holds, which proves the theorem. ∎

Remark 7: Note that the lower bound on the inter-event times is always smaller than in the case of ideal communication derived in [10], which is given by

$$\tau^i = \frac{c}{\|A_{K,i}\|k_{0,i}\|x_i(0)\| + \mu_i\big(1 + \frac{\|A_{K,i}\|k_{0,i}}{|\lambda_{max}(A_{K,i})|}\big)c}.$$

2) Communication with delays and packet losses: The previous paragraph was exclusively focussed on the effect of delays. However, in practice, delays and packet losses generally occur simultaneously.

Corollary 8: Assume that the maximum number of consecutive packet losses is upper bounded by $P_i^\star$, that the transmission delay τ_k^i of the state x_i of the agent i, $i \in 1,\ldots,N$ to its neighbors $j \in N_i$ is upper bounded by a constant $\bar\tau^i$ according to

$$\bar\tau^i = \frac{(\tau^\star)^i}{P_i^\star + 1}, \tag{8}$$

where $(\tau^\star)^i$ is given by (6), and that the waiting time T_W^i is set to $\bar\tau^i$. Then, there is a successful broadcast before the occurrence of a new event, and the state of each agent i is bounded by (7).

Proof: The accumulated error after $P_i^\star$ consecutive packet losses and a transmission delay $\bar\tau_k^i \leq \bar\tau^i$, assuming

that an event was triggered at time t_k^i, is

$$\underbrace{(e_i(t_k^i + T_W^i) - e_i(t_k^i)) + (e_i(t_k^i + 2T_W^i) - e_i(t_k^i + T_W^i)) + \ldots}_{P_i^\star \text{ times}}$$
$$+ (e_i(t_k^i + P_i^\star T_W^i + \bar\tau_k^i) - e_i(t_k^i + P_i^\star T_W^i))$$
$$= e_i(t_k^i + P_i^\star T_W^i + \bar\tau_k^i) - e_i(t_k^i). \tag{9}$$

Since $P_i^\star T_W^i + \bar\tau_k^i \leq P_i^\star T_W^i + \bar\tau^i = (P_i^\star + 1)\bar\tau^i = (\tau^\star)^i$, and $(\tau^\star)^i$ is also the minimum inter-event time for the system, an event cannot occur before $(\tau^\star)^i$, which implies that $\|e_i(t_k^i + P_i^\star T_W^i + \bar\tau_k^i) - e_i(t_k^i)\| < c$. Hence, $\|e_i(t)\| < 2c$ holds and so does the bound (7). ∎

Remark 9: Note that the maximum number of consecutive packet dropouts $P_i^\star$ and the maximum tolerable delay $\bar\tau^i$ are correlated. A large value of $P_i^\star$ implies small values of $\bar\tau^i$ and vice versa. So there is a trade-off between both parameters.

B. Properties of event-triggered control using Protocol 2

Using *Protocol 2*, the main issue is to keep track of the different versions of the broadcasted states. Therefore, some definitions are firstly introduced to adapt the previous analysis to this new scenario.

Definition 10: We denote $\{t_k^{i\to j}\}$ the successful broadcasting times from agent i to agent j, and the error

$$e_{i\to j}(t) = \tilde x_{i\to j}(t_k^{i\to j}) - x_i(t), \quad t \in [t_k^{i\to j}, t_{k+1}^{i\to j}), \tag{10}$$

where $\tilde x_{i\to j}(t_k^{i\to j})$ is the last broadcasted state from agent i to agent j, $j \in N_i$.

With this definition, the dynamics of agent i are given by

$$\dot x_i(t) = A_{K,i}x_i(t) + B_i K_i e_i(t) + \sum_{j\in N_i} B_i L_{ij}e_{j\to i}(t) \tag{11}$$

with $e_i(t)$ the version of the error that agent i has of itself. We assume that agent i automatically updates its broadcasted state in its control law and does not need to wait to receive an acknowledgment of successful reception from its neighbors.

With these prerequisites the following theorem is obtained.

Theorem 11: If *Protocol 2* is used, the network delay is upper bounded by

$$\bar\tau^i = \frac{(\tau^\star)^i}{P_i^\star + 1}, \tag{12}$$

where $P_i^\star$ is the maximum number of consecutive packet losses and

$$(\tau^\star)^i = \frac{c}{\|A_{K,i}\|k_{0,i}\|x_i(0)\| + \bar\mu_i\big(1 + \frac{\|A_{K,i}\|k_{0,i}}{|\lambda_{max}(A_{K,i})|}\big)2c}, \tag{13}$$

with $\bar\mu_i = \frac{1}{2}\|B_i K_i\| + \sum_{j\in N_i}\|B_i L_{ij}\|$, and the *waiting time* T_W^i is set to $\bar\tau^i$, then any broadcasted state $\tilde x_i$ of any subsystem i, $i \in 1,\ldots,N$ to the neighbors $j \in N_i$ is successfully received before a new event occurs, the local inter-event times $t_{k+1}^i - t_k^i$ are lower bounded by (13), and for any initial condition $x_i(0)$ and for any $t > 0$, it holds

$$\|x_i(t)\| \leq k_{0,i}\Big(\frac{\bar\mu_i 2c}{|\lambda_{max}(A_{K,i})|} + e^{-|\lambda_{max}(A_{K,i})|t}\big(\|x_i(0)\|$$
$$- \frac{\bar\mu_i 2c}{|\lambda_{max}(A_{K,i})|}\big)\Big). \tag{14}$$

Proof: According to *Protocol 2*, $\|e_i(t)\| \leq c$ holds and $e_i(t) \neq e_{i \to j}(t)$, in general. However, since Assumption 4 applies, $\|e_{i \to j}(t_k^{i \to j}) - e_i(t_k^i)\| < c$ yields $\|e_{i \to j}(t)\| < 2c$.

According to that, a bound on the state can be derived from (11) in a similar way as in Theorem 6 and (14) holds. The proof of the first part of the theorem can be obtained by following the proof of Theorem 6, since in the interval $[t_k^i, t_k^{i \to j})$ the state information $\tilde{x}_{i \to j}$ remains constant in the agent j, so that $\dot{e}_{i \to j}(t) = -\dot{x}_i(t)$ holds. If th error $e_{i \to j}(t)$ is integrated in the interval $[t_k^i, t_k^{i \to j})$, the state is bounded by (14), and the error is bounded as discussed above, (13) is derived. Finally, (12) can be derived as in Corollary 8. ∎

Remark 12: Note that the delay bound for *Protocol 1* and *Protocol 2* (see (6), (13)) are different. Since $\bar{\mu}_i < \mu_i$, under the same initial conditions *Protocol 2* allows larger delays.

C. Discussion of time-dependent trigger functions

Let us define trigger functions of the form

$$f_i(e_i(t)) = \|e_i(t)\| - ce^{-\alpha t} \tag{15}$$

where c is a positive constant and $\alpha < 0$.

The motivation of trigger functions of the form (15) was discussed in [10]. There is trade-off between the performance and the minimum inter-event in the choice of the parameter c for static trigger functions. A small c yields the generation of a lot of events, and large value of c may give a bad performance when the system approaches the origin. Time-dependent trigger functions provide larger inter-event times for small t of time and good performance for large values of t, if the parameter α is adequately selected.

In [10], it has been proved graphically that the inter-event time is lower bounded if $\alpha < |\lambda_{max}(A_K)|$, where $A_K = diag(A_{K,1}, \cdots, A_{K,N})$. Hence, under Assumption 4, similar conclusions could be derived to upper bound the delay allowing less conservative results. The simulation results in the next section will illustrate the benefits of using trigger functions of the form (15).

V. SIMULATION RESULTS

A. System description

The system considered is a collection of N inverted pendulums of mass m and length l coupled by springs with rate k. Each subsystem can be described by

$$\dot{x}_i = \begin{pmatrix} 0 & 1 \\ \frac{g}{l} - \frac{a_i k}{ml^2} & 0 \end{pmatrix} x_i + \begin{pmatrix} 0 \\ \frac{1}{ml^2} \end{pmatrix} u_i + \sum_{j \in N_i} \begin{pmatrix} 0 & 0 \\ \frac{h_{ij} k}{ml^2} & 0 \end{pmatrix} x_j$$

where $x_i = \begin{pmatrix} x_{i_1} & x_{i_2} \end{pmatrix}^T$, a_i is the number of springs connected to the ith pendulum and $h_{ij} = 1, \forall j \in N_i$ and 0 otherwise. K_i and L_{ij} gains are designed to decouple the system and place the poles at -1, -2. Therefore, $K_i = \begin{pmatrix} -3ml^2 & a_i k - \frac{ml^2}{4}(8 + \frac{4g}{l}) \end{pmatrix}$ and $L_{ij} = \begin{pmatrix} -k & 0 \end{pmatrix}$.

The same system has been used in [10] to demonstrate the event-based control strategy assuming an ideal network.

TABLE I
DELAYS FOR DIFFERENT VALUES OF C

c	0.01	0.02	0.05	0.1
$(\tau^\star)_{(6)}^i (ms)$	0.362	0.636	1.170	1.624
$(\tau^\star)_{(13)}^i (ms)$	0.379	0.693	1.378	2.054

B. Performance

To illustrate the theoretical results, the system behavior is investigated in three situations: 1) Ideal communication channel 2) Non-ideal network using *Protocol 1* 3) Non-ideal network using *Protocol 2*.

Consider that the number of subsystems is $N = 4$ and the initial conditions are $x(0) = \begin{pmatrix} -0.9425 & 0 & 1.0472 & 0 & 0.6283 & 0 & -1.4137 & 0 \end{pmatrix}^T$. The upper bounds on the delay are computed for *Protocol 1* (6) and *Protocol 2* (13) and for different values of the parameter c of the trigger function (4). The results illustrated in Table I correspond to one of the inner pendulums ($i = 2$). Note that the difference between the value of $(\tau^\star)^i$ given by the two protocols increases with c and that *Protocol 2* always allows larger (less conservative) values on the delay. Consider now $c = 0.05$ and a delay generated randomly between zero and the corresponding upper bound specified in Table I. The state of subsystem 2, the events time and the control input $u(t)$ are depicted in Fig. 2 for the three situations stated above. The behavior of the subsystem is similar in the three cases as the effect of delays in the performance of the system is mitigated by means of the two protocols proposed. Note that even though the delay does not significantly affect in the performance, it has an impact on the sequence of events. This is an interesting property of event-based control, because the delay in one tranmission will affect the occurrence of future events.
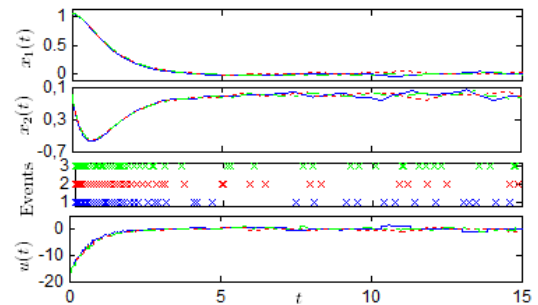

Fig. 2. Behavior of the system with Protocol 1 (dotted line), Protocol 2 (dashed line) and a ideal network (solid line)

C. Protocol 1 vs. Protocol 2

In order to illustrate the difference between *Protocol 1* and *Protocol 2* in more detail, Fig. 3 extracts a short time interval showing how the broadcasted state $\tilde{x}$ of Agent 2 is used in the system. Since Agent 2 is an inner pendulum, it has two neighbors. For *Protocol 1* the three copies of $\tilde{x}$ (one in Agent 2, one in Agent 1 and the third in Agent 3) are

identical since all the neighbors wait for the last reception (here $\tilde{x}_{2\to3}$) at time $t = 1.668$ s to update the value of $\tilde{x}$ (Fig. 3a), which is depicted by the solid line. In contrast, using *Protocol 2* (Fig. 3b), whenever an event is triggered in Agent 2, its state is broadcasted and inmediatelly updated in Agent 2.The neighbors also update as soon as they received the broadcasted state. Note that the events times are not the same in the two protocols because the time of one update affects the generation of future events, as mentioned before.
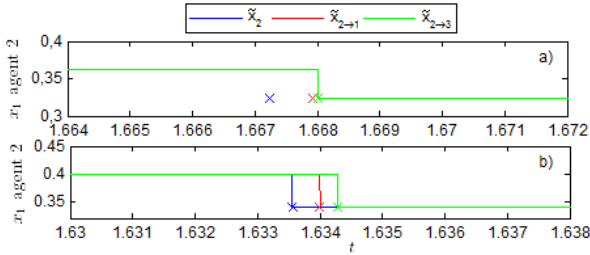
Fig. 3. Difference between a) Protocol 1 and b) Protocol 2 in updating the state information. Only the first component of the state is depicted.

D. Time-dependent trigger function

If we consider time-dependent trigger functions of the form (15) with parameters $c = 0.5$ and $\alpha = 0.8$ and if we compute a lower bound for the minimum inter-event time τ as discussed in [10] assuming an ideal network scenario, we obtain $\tau \in [0.0072, 0.0109)$ s.

The performance of the system under the time-dependent trigger functions is compareed with the behavior using the static-trigger functions for $(\tau^\star)^i = 3.6$ ms. The results are shown in Fig. 4. The state of agent 2 (x_{2_1}, x_{2_2}) is depicted in Fig. 4a, and Fig. 4b shows the broadcasted states $(\tilde{x}_{2_1}, \tilde{x}_{2_2})$. Note that the number of updates in the broadcasted state
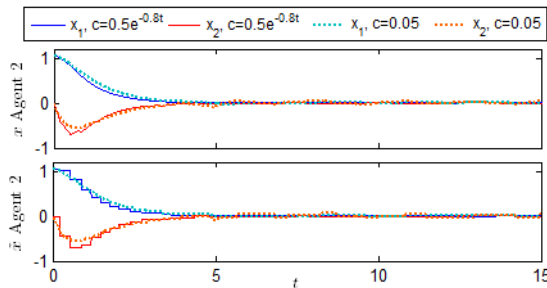
Fig. 4. Behavior of the system with trigger functions (4) ($c = 0.05$) and (15) ($c = 0.5$, $\alpha = 0.8$), and delay bound set to 3.6 ms

(number of events) decreases with trigger functions (15) and the performance around the equilibria is better with respect to (4). Moreover, the minimum and mean inter-event times have been computed according for these simulation results, resulting in 3.9 ms and 353 ms, respectively, for (15), and 1.2 ms, which agrees with the results of Table I, and 215 ms for (4). Hence, the time-dependent trigger functions seem an interesting alternative in non-ideal networks.

VI. CONCLUSIONS AND FUTURE WORK

The performance of distributed event-triggered control of interconnected linear systems for a non-ideal communication network inducing delays and packet losses has been analyzed in this paper. Upper bounds on the delay and the maximum number of consecutive packet dropouts are derived considering two different communication protocols both of which guarantee a stable behavior. The first protocol preserves *state consistency* which is relaxed by the second protocol leading to less conservative upper bound on the delay. The theoretical results are illustrated through simulations, showing the stability of the system under the proposed schemes.

Future works will include the analytical extension of the scheme to deal with time-dependent trigger functions showing promising results in the simulations and the extension to considering exogenous disturbances.

REFERENCES

[1] A. Cervin, T. Henningsson. Scheduling of event-triggered controllers on a shared network. *47th IEEE Conference on Decision and Control*, Cancun, 2008.

[2] W.P.M.H. Heemels, J. Sandee, P. van den Bosch. Analysis of event-driven controllers for linear systems. *International Journal of Control*, 81(4):571-590, 2008.

[3] P. Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Transactions on Automatic Control*, 52(9), 2007.

[4] M. Rabi, K. H. Johansson. Scheduling packets for event-triggered control. *European Control Conference*, Budapest, 2009.

[5] J. Lunze, D. Lehmann. A state-feedback approach to event-based control. *Automatica*, 46(1):211-215, 2010.

[6] D.V. Dimarogonas, K.H. Johansson. Event-triggered control for multi-agent systems. *48th IEEE Conference on Decision and Control*, Shanghai, 2009.

[7] M. Mazo,P. Tabuada. Decentralized event-triggered control over wireless sensor/actuator networks. *IEEE Transactions on Automatic Control*, 56(10):2456-2461, 2011.

[8] X. Wang, M.D. Lemmon. Event-triggering in distributed networked control systems. *IEEE Transactions on Automatic Control*, 56(3):586-601, 2011.

[9] G.S. Seyboth, D.V. Dimarogonas, K.H. Johansson. Control of Multi-Agent Systems via Event-based Communication. *18th IFAC World Congress*, Milano, 2011.

[10] M. Guinaldo, D.V. Dimarogonas, K.H. Johansson, J. Sánchez, S. Dormido. Distributed Event-Based Control for Interconnected Linear Systems. *50th IEEE Conference on Decision and Control*, Orlando, 2011.

[11] R. Postoyan, T. Tabuada, D. Nešìc̀, A. Anta. Event-triggered and self-triggered stabilization of distributed networked control systems. *50th IEEE Conference on Decision and Control*, Orlando, 2011.

[12] A. Molin, S. Hirche. Optimal design of decentralized event-triggered controllers for large-scale systems with contention-based communication. *50th IEEE Conference on Decision and Control*, Orlando, 2011.

[13] F.Y. Wang, D. Liu. *Networked Control Systems: Theory and Applications*. London: Springer-Verlag, 2008.

[14] M. Mazo, A. Anta, P. Tabuada. An ISS self-triggered implementation for linear controllers. *Automatica*, 46(8):1310-1314, 2010.

[15] X. Wang, M.D. Lemmon. Self-triggering under state-independent disturbances. *IEEE Transactions on Automatic Control*, 55(6):1494-1500, 2010.

[16] R. Blind, F. Allgöwer. On the optimal sending rate for networked control systems with a shared communication medium. *50th IEEE Conference on Decision and Control*, Orlando, 2011.

[17] D. Lehmann, J. Lunze. Event-based control with communication delays and packet losses. *International Journal of Control*, 85(5):563-577, 2012.

[18] E. Garcia P.J. Antsaklis. Model-based event-triggered control with time-varying network delays. *50th IEEE Conference on Decision and Control*, Orlando, 2011.