2021 60th IEEE Conference on Decision and Control (CDC)
December 13-15, 2021. Austin, Texas

Reducing Attack Vulnerabilities Through Decentralized Event-Triggered Control

Paul Griffioen, Raffaele Romagnoli, Bruce H. Krogh, and Bruno Sinopoli

2021 60th IEEE Conference on Decision and Control (CDC) | 978-1-6654-3659-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/CDC45484.2021.9682870

Abstract— **Decentralized control systems are widely used in a number of situations and applications. In order for these systems to function properly and achieve their desired goals, information must be propagated between agents, which requires connecting to a network. To reduce vulnerabilities to attacks that may be carried out through the network, we design an event-triggered mechanism for network connection and communication that minimizes the amount of time agents must be connected to the network, in turn decreasing communication costs. This mechanism is a function of only local information and ensures stability for the overall system in attack-free scenarios. Our approach distinguishes itself from current decentralized event-triggered control strategies by including measurements in the system model, by not needing to implement any reachability analysis, and by considering scenarios where agents are not always connected to the network to receive critical information from other agents. Algorithms describing these network connection and communication protocols are provided, and our approach is illustrated via simulation.**

I. Introduction

Cyber-physical systems, engineered systems which include sensing, communication, and control in physical spaces, are essential to secure and protect in today's society. Cyber-physical systems are ubiquitous in modern critical infrastructures including the smart grid, transportation systems, health care, sewage/water management, energy delivery, and manufacturing. These large scale, highly connected systems may be deployed in insecure public spaces and may contain heterogeneous components and devices, thus creating numerous attack surfaces. Consequently, these systems are attractive targets for adversaries, especially safety critical systems [1]–[4]. Many of these systems are distributed over a wide area and are comprised of many different agents which interact with one another in a decentralized manner. It is important, therefore, to guarantee the safety and security of these decentralized control systems which rely heavily on network communication to achieve their goals.

In decentralized control systems, individual agents have access to differing amounts of information. In order to maintain the stability of the overall system and achieve a global objective, agents must occasionally communicate some subset of their local information with other agents, for instance in car platoons [5], [6]. However, communicating with other agents requires connecting to the network, opening up the possibility for adversaries to corrupt information that is sent over the network and corrupt an agent's control software by using the network connection to inject malicious code. These facts, in addition to the desire for minimizing communication costs, motivate the need for intermittent network connections as opposed to holding a constant network connection all the time. Due to the fact that different sets of local information are available to each agent, the decision about when network connection and communication are necessary for a particular agent must be triggered locally. While intermittent network connections alone will not ensure resiliency against attacks, they reduce an adversary's window of opportunity for attack while also providing a framework in which a resilience strategy may be implemented.

Existing approaches to decentralized event-triggered control have mainly been concerned with minimizing communication costs, not with reducing the overall system's vulnerabilities to attacks from the network. Different approaches to event-triggered control are summarized well in [7]. A variety of decentralized event-triggered control mechanisms for linear, nonlinear, continuous time, and discrete time systems are presented in [8]–[14] and provide conditions under which global asymptotic stability, global exponential stability, $\mathcal{L}_\infty$ gain performance, or $\mathcal{L}_p$ gain performance are achieved. All of these approaches assume that each agent is always connected to the network and is always available to receive any information that is sent to it, even though the agent might not always be broadcasting information to other agents. However, this assumption does not hold in contexts where safety and security is important since the attack window is minimized when each agent disconnects from the network for as long as possible. While [15] presents an approach which does not assume that all agents are always connected to the network, it does not include sensor measurements in the system model, and consequently it presents a trigger condition that cannot be applied to systems where the state cannot be accessed directly. Furthermore, [15] implements reachability analysis in order to evaluate the trigger condition, which can be computationally costly.

In contrast to these previous approaches, we present a decentralized event-triggered network connection and communication protocol that does not assume that all agents are always connected to the network, that includes sensor measurements in the system model, and that does not require any reachability analysis to be implemented. The network connection and communication protocol ensures the stability of the overall system in attack-free scenarios when agents periodically connect and disconnect from the network (as

P. Griffioen, R. Romagnoli, and B. H. Krogh are with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, USA 15213. B. Sinopoli is with the Department of Electrical and Systems Engineering, Washington University in St. Louis, St. Louis, MO, USA 63130. Email: {pgriffi1|rromagno|krogh}@andrew.cmu.edu, bsinopoli@wustl.edu

opposed to periodically broadcasting information to other agents). This protocol uses a trigger condition based only on local information to determine when a particular agent must connect to the network to send and receive information from other agents. This trigger condition is designed to guarantee system stability by having an agent connect to the network when the magnitude of the state estimation error grows too large.

The remainder of this paper is organized as follows. Section II introduces the system model and estimation procedure that is used by each agent. Section III presents the triggering mechanism, network connection and communication protocol, and conditions under which stability of the overall system is achieved. Simulation results are presented in Section IV, and Section V concludes the paper.

II. Problem Formulation

A. System Model

We model the plant as a discrete time linear time invariant system composed of N agents. The overall system dynamics are given by

$$x_{k+1} = Ax_k + Bu_k + w_k, \tag{1}$$

$$y_k = Cx_k + v_k, \tag{2}$$

where $x_k \in \mathbb{R}^n$ represents the state, $u_k \in \mathbb{R}^p$ denotes the control input, $y_k \in \mathbb{R}^m$ represents the sensor measurements, and $w_k \in \mathbb{R}^n$ and $v_k \in \mathbb{R}^m$ are bounded disturbances which lie in the compact sets W and V, respectively, given by

$$W \triangleq \left\{w_k \middle| w_k^T Q w_k \leq 1\right\}, \tag{3}$$

$$V \triangleq \left\{v_k \middle| v_k^T R v_k \leq 1\right\}. \tag{4}$$

We let $y_k^i \in \mathbb{R}^{m_i}$ represent the sensor measurements that are locally available to agent i so that $y_k = \begin{bmatrix} y_k^{1^T} & \cdots & y_k^{N^T} \end{bmatrix}^T$, $m = \sum_{i=1}^N m_i$, $C \triangleq \begin{bmatrix} C_1^T & \cdots & C_N^T \end{bmatrix}^T$, and $C_i \in \mathbb{R}^{m_i \times n}$. Similarly, we let $u_k^i \in \mathbb{R}^{p_i}$ denote agent i's control inputs so that $u_k = \begin{bmatrix} u_k^{1^T} & \cdots & u_k^{N^T} \end{bmatrix}^T$, $p = \sum_{i=1}^N p_i$, $B \triangleq \begin{bmatrix} B_1 & \cdots & B_N \end{bmatrix}$, and $B_i \in \mathbb{R}^{n \times p_i}$.

Each agent i is able to directly access its own local sensor measurements y_k^i and control inputs u_k^i but must rely on communication from other agents to access y_k^j and u_k^j $\forall j \neq i$, the sensor measurements and control inputs locally available to other agents. We assume that the communication graph is a complete graph so that agents are able to directly send information to one another when connected to the network.

B. State Estimation

The control input for agent i is given by

$$u_k^i = K_i \hat{x}_{k|k}^i, \tag{5}$$

where $\hat{x}_{k|k}^i \in \mathbb{R}^n$ is agent i's a posteriori estimate of the overall state. Before agent i decides whether or not to connect to the network, it uses $\hat{x}_{k|k}^i$ and its local sensor measurements to compute $\hat{x}_{k+1|k}^i \in \mathbb{R}^n$, agent i's a priori estimate of the overall state, according to

$$\hat{x}_{k+1|k}^i = A_{bk} \hat{x}_{k|k}^i + L_i(y_k^i - C_i \hat{x}_{k|k}^i), \tag{6}$$

where $A_{bk} \triangleq A + BK$ and $K \triangleq \begin{bmatrix} K_1^T & \cdots & K_N^T \end{bmatrix}^T$. Here L_i is agent i's observer gain matrix, given by

$$L_i \triangleq T_i \begin{bmatrix} L_i^o \\ 0 \end{bmatrix},\ T_i^{-1} A T_i = \begin{bmatrix} A_i^o & 0 \\ A_i^{21} & A_i^{\bar{o}} \end{bmatrix},\ C_i T_i = \begin{bmatrix} C_i^o & 0 \end{bmatrix}, \tag{7}$$

where L_i^o is designed so that $A_i^o - L_i^o C_i^o$ is Schur stable, and T_i is a similarity transformation matrix used to carry out the observability decomposition so that (A_i^o, C_i^o) is observable. The observer presented in (6) is simply a Luenberger observer that uses only local sensor measurements y_k^i and approximates $u_k^j \approx K_j \hat{x}_{k|k}^i$ $\forall j \neq i$ since these inputs are not locally available to agent i. As a result, $\hat{x}_{k+1|k}^i$ is computed without connecting to the network.

If agent i connects to the network, it uses the information it receives from other agents to compute its a posteriori state estimate according to

$$\begin{aligned} \hat{x}_{k+1|k+1}^i = A\hat{x}_{k|k}^i + \sum_{j=1}^N B_j(\delta_k^{ij} u_k^j + (1-\delta_k^{ij}) K_j \hat{x}_{k|k}^i) \\ + \sum_{j=1}^N L_j(\delta_k^{ij})(y_k^j - C_j \hat{x}_{k|k}^i) \end{aligned} \tag{8}$$

$$= \left(\prod_{\ell=k_i'}^{k} \hat{A}_{k+k_i'-\ell}^i \right) \hat{x}_{k_i'|k_i'}^i + \sum_{\ell=k_i'}^{k} \left(\prod_{\zeta=\ell+1}^{k} \hat{A}_{k+\ell+1-\zeta}^i \right) \psi_\ell^i, \tag{9}$$

where $\hat{A}_k^i \triangleq (A + \sum_{j=1}^N (1-\delta_k^{ij}) B_j K_j - L_j(\delta_k^{ij}) C_j)$, $\psi_k^i \triangleq \sum_{j=1}^N \delta_k^{ij} B_j u_k^j + L_j(\delta_k^{ij}) y_k^j$,

$$\delta_k^{ij} = \begin{cases} 1 & \text{if agent } i \text{ possesses } \{u_k^j, y_k^j\} \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

and k_i' represents the most recent time step where agent i possesses $\{u_k^j, y_k^j\}$ for all agents at all time steps up to k_i', given by

$$k_i' \triangleq \max_\ell \ell \text{ s.t. } \delta_{0:\ell-1}^{ij} = 1\ \forall j \in \{1, \cdots, N\}. \tag{11}$$

If $\delta_k^{ij} = 0$, then $L_j(\delta_k^{ij}) \triangleq 0_{n \times m_j}$. If $\delta_k^{ij} = 1$, then $L_j(\delta_k^{ij})$ is designed in the following manner. Let $\delta_k^i \triangleq \{\delta_k^{i1}, \cdots, \delta_k^{iN}\}$, and let $\hat{C}(\delta_k^i) \in \mathbb{R}^{(\sum_{j=1}^N \delta_k^{ij} m_j) \times n}$ be a matrix composed of the rows of C corresponding to the sensor measurements y_k^j that agent i possesses. In other words, $\hat{C}(\delta_k^i)$ is composed by stacking the blocks C_j on top of each other $\forall j \in \{1, \cdots, N\}$ such that $\delta_k^{ij} = 1$. Similarly, let $\hat{L}(\delta_k^i) \in \mathbb{R}^{n \times (\sum_{j=1}^N \delta_k^{ij} m_j)}$ represent agent i's observer gain matrix for all the sensor measurements y_k^j that it possesses. This matrix is designed in the same manner as L_i in (7), where L_i and C_i are replaced with $\hat{L}(\delta_k^i)$ and $\hat{C}(\delta_k^i)$, respectively. Then the set of matrices $\{L_j(\delta_k^{ij}) \in \mathbb{R}^{n \times m_j} | \delta_k^{ij} = 1\}$ are defined as the appropriate blocks of $\hat{L}(\delta_k^i)$. The design of $\hat{L}(\delta_k^i)$ can be carried out offline for all possible values of δ_k^i, resulting in a set of

$2^N - 1$ observer gain matrices that are used by all agents.

Note that since agent i always has access to its own local control inputs and sensor measurements, $\delta_k^{ii} = 1$ $\forall i \in \{1, \cdots, N\}$. Also note that in the case where agent i does not connect to the network at time step k, $L_i(\delta_k^{ii}) = L_i$ and $L_j(\delta_k^{ij}) = 0_{n \times m_j}$ $\forall j \neq i$. This results in (8) reducing to (6) so that $\hat{x}^i_{k+1|k+1} = \hat{x}^i_{k+1|k}$. When agent i does connect to the network, it obtains the data $\{u_k^j, y_k^j\}$, $j \neq i$ from other agents according to the procedure presented in Algorithm 1, and it uses that data to compute $\hat{x}^i_{k+1|k+1}$ according to (9). By having agents periodically connect and disconnect from the network, the data that is sent over the network is sent in periodic bursts as opposed to being spread out uniformly over a large period of time. Note that agents do not need to have the same initial state estimate $\hat{x}^i_{0|0}$.

Algorithm 1 Data Communication Procedure for Agent i

1: Initialize $\theta_k^i = \bar{\theta}_k^i = \emptyset$
2: **parfor** $j \in \{1, \cdots, N\}$, $j \neq i$
3: Send $\kappa_k^{i\ell}$ $\forall \ell \in \{1, \cdots, N\}$, $\ell \neq i$ to agent j and listen for information from agent j
4: **if** Information is received
5: $\theta_k^i = \{\theta_k^i, j\}$
6: Send $\{u^i_{\kappa_k^{ji}+1:k}, y^i_{\kappa_k^{ji}+1:k}\}$ to agent j
7: **else**
8: $\bar{\theta}_k^i = \{\bar{\theta}_k^i, j\}$
9: **end if**
10: **end parfor**
11: **for** $j \in \bar{\theta}_k^i$
12: **if** $\kappa_k^{ij} \geq \kappa_k^{\ell j}$ $\forall \ell \in \theta_k^i$
13: **parfor** $\ell \in \theta_k^i$
14: Send $\{u^j_{\kappa_k^{\ell j}+1:\kappa_k^{ij}}, y^j_{\kappa_k^{\ell j}+1:\kappa_k^{ij}}\}$ to agent ℓ
15: **end parfor**
16: **end if**
17: **end for**

In Algorithm 1, θ_k^i and $\bar{\theta}_k^i$ represent the sets of agents connected and disconnected from the network at time step k, respectively, not including agent i in either set. κ_k^{ij} represents the most recent time step agent i possesses information about agent j's inputs and outputs, given by

$$\kappa_k^{ij} \triangleq \max_{\ell} \ell \text{ s.t. } \delta_\ell^{ij} = 1, \ \ell \in \{0, \cdots, k\}. \tag{12}$$

If agent i connects to the network, it first sends κ_k^{ij} to every agent (line 3), letting every agent know the most recent time step agent i possesses information about each agent's inputs and outputs. Agent i also receives this information from each agent connected to the network and proceeds to send each of these agents all of its own local data $\{u_k^i, y_k^i\}$ that these agents do not currently possess (line 6). Of the agents currently connected to the network, if agent i possesses the most recent information about an agent j not currently connected to the network, it sends that data $\{u_k^j, y_k^j\}$ to all the other agents currently connected to the network (line 14). This results in a maximum of 3 messages sent to each agent connected to the network (lines 3, 6, and 14) and 1 message sent to each agent disconnected from the network (line 3).

In this way, Algorithm 1 ensures that all agents currently connected to the network possess the exact same set of data $\{u_k^j, y_k^j\}$ about all agents. By possessing this exact same set of data, each agent i is able to use the maximum amount of information available to compute its a posteriori state estimate $\hat{x}^i_{k+1|k+1}$ according to (9). Note that when $\delta^{ij}_{0:k} = 1$ $\forall j \in \{1, \cdots, N\}$, $L \triangleq \hat{L}(\delta_k^i)$ is the Luenberger observer gain matrix and (8) reduces to

$$\hat{x}^i_{k+1|k+1} = A\hat{x}^i_{k|k} + Bu_k + L(y_k - C\hat{x}^i_{k|k}), \tag{13}$$

implying that $\hat{x}_{k_i'|k_i'}$ is equivalent to the state estimate obtained by a centralized Luenberger observer with access to all the inputs and the outputs of the system. As a result, each agent i only needs to store the data $\{u_k^j, y_k^j\}$ from agents for all time steps at and after $\bar{k} \triangleq \min_j k_j'$ s.t. $j \in \{1, \cdots, N\}$.

Remark 1: Note that cases may exist where agent i's local memory capacity places a non-negligible limit on the amount of data $\{u_k^j, y_k^j\}$ it can store, which may occur in circumstances where agents do not connect to the network very often. In these cases, agent i can simply connect to the network whenever its local memory has been filled, allowing it to share its data with other agents. If all agents know every agent's local memory capacity, then each agent can compute the minimum number of time steps it will take before another agent's local memory is filled. With this information, each agent can make sure that it always connects to the network when another agent's local memory is filled, allowing agents to share data with one another, in turn increasing the value of $\bar{k}$, the most distant time step from which agents need to store data.

C. Problem Formulation

Given the controller in (5), the system dynamics in (1) can be written as

$$x_{k+1} = A_{bk}x_k - Ee_{k|k} + w_k, \tag{14}$$

where $E \triangleq \begin{bmatrix} B_1K_1 & \cdots & B_NK_N \end{bmatrix}$, $e_{k|k} \triangleq \begin{bmatrix} e_{k|k}^{1^T} & \cdots & e_{k|k}^{N^T} \end{bmatrix}^T$, and $e^i_{k|k} \triangleq x_k - \hat{x}^i_{k|k}$ so that $e_{k|k} \in \mathbb{R}^{Nn}$ and $e^i_{k|k} \in \mathbb{R}^n$. Here $e^i_{k|k}$ represents the error between the overall state and agent i's a posteriori estimate of the overall state. We next introduce a network connection protocol which decides when it is necessary for each agent to connect to the network and communicate with other agents to ensure the stability of the overall system.

III. Network Connection Protocol

A. Quadratic Boundedness

In order to maintain the stability and safety of the overall system, each agent i must occasionally share $\{u_k^i, y_k^i\}$ with other agents. We would like to design a network connection protocol that ensures the stability of the overall system by properly coordinating communication between different agents while also minimizing the number of times each agent connects to the network. The mechanism that triggers this network connection is only able to use locally available information. The stability which we design the network connection protocol to achieve is quadratic γ-boundedness

which is described in Definition 1, Definition 2, and Lemma 1. These definitions and lemmas have been uniquely modified and adapted from [16] by adding the parameter γ to not only specify *when* the Lyapunov function decreases or increases but also *how fast* it does so.

Definition 1 ([16]): Let z_k represent a state vector, let d_k^1 and d_k^2 represent disturbance vectors, and let D_1 and D_2 be compact sets. A system of the form

$$z_{k+1} = \mathcal{A}z_k + \mathcal{B}_1 d_k^1 + \mathcal{B}_2 d_k^2, \quad d_k^1 \in D_1,\ d_k^2 \in D_2 \tag{15}$$

is quadratically γ-bounded with $\gamma \geq 0$ and symmetric positive definite Lyapunov matrix $\mathcal{P}$ if and only if $\forall d_k^1 \in D_1$ and $\forall d_k^2 \in D_2$,

$$z_k^T \mathcal{P} z_k \geq 1 \implies z_{k+1}^T \mathcal{P} z_{k+1} < \gamma z_k^T \mathcal{P} z_k. \tag{16}$$

Note that when $\gamma \in [0,1]$, γ specifies how quickly the Lyapunov function decreases over time, and when $\gamma > 1$, γ sets an upper bound on how quickly the Lyapunov function can increase over time.

Definition 2 ([16]): The set Z is a robustly positively invariant set for (15) if and only if $z_0 \in Z$ implies that $z_k \in Z\ \forall k \geq 0$, $\forall d_k^1 \in D_1$, and $\forall d_k^2 \in D_2$.

Lemma 1 ([16]): The following two statements are equivalent:

1) System (15) is quadratically γ-bounded with $\gamma \in [0,1]$ and symmetric positive definite Lyapunov matrix $\mathcal{P}$.
2) The set $Z \triangleq \{z_k^T \mathcal{P} z_k \leq 1,\ \mathcal{P} \succ 0\}$ is a robustly positively invariant set for (15).

Given these definitions, Lemma 2 provides a sufficient condition for evaluating the quadratic γ-boundedness of (15).

Lemma 2: Let $D_1 \triangleq \{d_k^1 | {d_k^1}^T \mathcal{D}_1 d_k^1 \leq 1,\ \mathcal{D}_1 \succ 0\}$ and $D_2 \triangleq \{d_k^2 | {d_k^2}^T \mathcal{D}_2 d_k^2 \leq 1,\ \mathcal{D}_2 \succ 0\}$ for the system in (15). If $\exists \alpha \geq 0$ such that

$$\begin{bmatrix} (\gamma - 2\alpha)\mathcal{P} - \mathcal{A}^T\mathcal{P}\mathcal{A} & -\mathcal{A}^T\mathcal{P}\mathcal{B}_1 & -\mathcal{A}^T\mathcal{P}\mathcal{B}_2 \\ -\mathcal{B}_1^T\mathcal{P}\mathcal{A} & \alpha\mathcal{D}_1 - \mathcal{B}_1^T\mathcal{P}\mathcal{B}_1 & -\mathcal{B}_1^T\mathcal{P}\mathcal{B}_2 \\ -\mathcal{B}_2^T\mathcal{P}\mathcal{A} & -\mathcal{B}_2^T\mathcal{P}\mathcal{B}_1 & \alpha\mathcal{D}_2 - \mathcal{B}_2^T\mathcal{P}\mathcal{B}_2 \end{bmatrix} \succ 0, \tag{17}$$

then the system in (15) is quadratically γ-bounded with $\gamma \geq 0$ and symmetric positive definite Lyapunov matrix $\mathcal{P}$.

Proof: By using the S-procedure [17], (17) is equivalent to

$$\begin{aligned} &\begin{bmatrix} z_k \\ d_k^1 \\ d_k^2 \end{bmatrix}^T \begin{bmatrix} -2\mathcal{P} & 0 & 0 \\ 0 & \mathcal{D}_1 & 0 \\ 0 & 0 & \mathcal{D}_2 \end{bmatrix} \begin{bmatrix} z_k \\ d_k^1 \\ d_k^2 \end{bmatrix} \leq 0 \implies \\ &\begin{bmatrix} z_k \\ d_k^1 \\ d_k^2 \end{bmatrix}^T \begin{bmatrix} \mathcal{A}^T\mathcal{P}\mathcal{A} - \gamma\mathcal{P} & \mathcal{A}^T\mathcal{P}\mathcal{B}_1 & \mathcal{A}^T\mathcal{P}\mathcal{B}_2 \\ \mathcal{B}_1^T\mathcal{P}\mathcal{A} & \mathcal{B}_1^T\mathcal{P}\mathcal{B}_1 & \mathcal{B}_1^T\mathcal{P}\mathcal{B}_2 \\ \mathcal{B}_2^T\mathcal{P}\mathcal{A} & \mathcal{B}_2^T\mathcal{P}\mathcal{B}_1 & \mathcal{B}_2^T\mathcal{P}\mathcal{B}_2 \end{bmatrix} \begin{bmatrix} z_k \\ d_k^1 \\ d_k^2 \end{bmatrix} < 0, \end{aligned} \tag{18}$$

which in turn is equivalent to

$$\begin{aligned} -2z_k^T\mathcal{P}z_k + {d_k^1}^T\mathcal{D}_1 d_k^1 + {d_k^2}^T\mathcal{D}_2 d_k^2 \leq 0 \\ \implies z_{k+1}^T\mathcal{P}z_{k+1} < \gamma z_k^T\mathcal{P}z_k. \end{aligned} \tag{19}$$

Note that

$$\left.\begin{matrix} z_k^T\mathcal{P}z_k \geq 1 \\ {d_k^1}^T\mathcal{D}_1 d_k^1 \leq 1 \\ {d_k^2}^T\mathcal{D}_2 d_k^2 \leq 1 \end{matrix}\right\} \implies -2z_k^T\mathcal{P}z_k + {d_k^1}^T\mathcal{D}_1 d_k^1 + {d_k^2}^T\mathcal{D}_2 d_k^2 \leq 0. \tag{20}$$

Taking (19) and (20) in conjunction with one another yields that $\forall d_k^1 \in D_1$ and $\forall d_k^2 \in D_2$,

$$z_k^T\mathcal{P}z_k \geq 1 \implies z_{k+1}^T\mathcal{P}z_{k+1} < \gamma z_k^T\mathcal{P}z_k, \tag{21}$$

implying that the system in (15) is quadratically γ-bounded with $\gamma \geq 0$ and symmetric positive definite Lyapunov matrix $\mathcal{P}$. ∎

According to (14) and (8), the error dynamics for the overall system are given by

$$e_{k+1|k+1} = \bar{A}(\delta_k)e_{k|k} + \mathcal{I}w_k + \bar{L}(\delta_k)v_k, \tag{22}$$

where $\delta_k \triangleq \{\delta_k^1, \cdots, \delta_k^N\}$, $\mathcal{I} \triangleq \begin{bmatrix} I_n & \cdots & I_n \end{bmatrix}^T$,

$$\bar{A}(\delta_k) \triangleq \begin{bmatrix} A_{bk} - F(\delta_k^1) & \cdots & (\delta_k^{1N} - 1)B_N K_N \\ \vdots & \ddots & \vdots \\ (\delta_k^{N1} - 1)B_1 K_1 & \cdots & A_{bk} - F(\delta_k^N) \end{bmatrix},$$

$$\bar{L}(\delta_k) \triangleq \begin{bmatrix} -L_1(\delta_k^{11}) & \cdots & -L_N(\delta_k^{1N}) \\ \vdots & \ddots & \vdots \\ -L_1(\delta_k^{N1}) & \cdots & -L_N(\delta_k^{NN}) \end{bmatrix},$$

and $F(\delta_k^i) \triangleq \sum_{j=1}^N \delta_k^{ij} B_j K_j + L_j(\delta_k^{ij}) C_j$. Lemma 3 provides a sufficient condition under which the error is quadratically 1-bounded and remains in the robust positive invariant set $\mathcal{E}_e$ given by

$$\mathcal{E}_e \triangleq \left\{ e_{k|k} \middle| e_{k|k}^T \bar{P} e_{k|k} \leq 1,\ \bar{P} \succ 0 \right\} \tag{23}$$

when all agents are connected to the network. In other words, when $\delta_k^{ij} = 1\ \forall i, j \in \{1, \cdots, N\}$.

Lemma 3: If $\exists \alpha_1 \geq 0$ such that

$$\begin{bmatrix} (\gamma - 2\alpha_1)\bar{P} - \bar{A}(\delta_k)^T\bar{P}\bar{A}(\delta_k) & -\bar{A}(\delta_k)^T\bar{P}\mathcal{I} & -\bar{A}(\delta_k)^T\bar{P}\bar{L}(\delta_k) \\ -\mathcal{I}^T\bar{P}\bar{A}(\delta_k) & \alpha_1 Q - \mathcal{I}^T\bar{P}\mathcal{I} & -\mathcal{I}^T\bar{P}\bar{L}(\delta_k) \\ -\bar{L}(\delta_k)^T\bar{P}\bar{A}(\delta_k) & -\bar{L}(\delta_k)^T\bar{P}\mathcal{I} & \alpha_1 R - \bar{L}(\delta_k)^T\bar{P}\bar{L}(\delta_k) \end{bmatrix} \succ 0 \tag{24}$$

when $\gamma = 1$ and $\delta_k^{ij} = 1\ \forall i, j \in \{1, \cdots, N\}$, then the error in (22) is quadratically 1-bounded with symmetric positive definite Lyapunov matrix $\bar{P}$. Furthermore, if $e_{0|0} \in \mathcal{E}_e$, then $e_{k|k} \in \mathcal{E}_e\ \forall k \geq 0$.

Proof: Applying Lemma 2 to the system in (22) implies that if (24) is satisfied with $\gamma = 1$, then the error in (22) is quadratically 1-bounded. Lemma 1 and Definition 2 imply that if $e_{0|0} \in \mathcal{E}_e$, then $e_{k|k} \in \mathcal{E}_e\ \forall k \geq 0$. ∎

Lemma 4 provides a sufficient condition under which the overall system is quadratically 1-bounded when all agents are connected to the network.

Lemma 4: If $\exists \alpha_2 \geq 0$ such that

$$\begin{bmatrix} (1 - 2\alpha_2)P - A_{bk}^T P A_{bk} & A_{bk}^T P E & -A_{bk}^T P \\ E^T P A_{bk} & \alpha_2 \bar{P} - E^T P E & E^T P \\ -P A_{bk} & P E & \alpha_2 Q - P \end{bmatrix} \succ 0, \tag{25}$$

then the system in (14) is quadratically 1-bounded with symmetric positive definite Lyapunov matrix P when $e_{k|k} \in$

$\mathcal{E}_e$.

Proof: Applying Lemma 2 to the system in (14) implies that if (25) is satisfied, then the system in (14) is quadratically 1-bounded. ■

B. Stability Conditions

We want to ensure that the overall system in (14) is quadratically 1-bounded by creating a network connection protocol which guarantees that when the Lyapunov function $V(x_k) \triangleq x_k^T P x_k$ is greater than or equal to 1, it decreases at every time step and converges to the robust positive invariant set $\mathcal{E}_x$ given by

$$\mathcal{E}_x \triangleq \left\{ x_k \middle| x_k^T P x_k \leq 1,\ P \succ 0 \right\}. \tag{26}$$

The invariance of $\mathcal{E}_x$ is shown in Lemma 1 to be equivalent to quadratic 1-boundedness. Consequently, the network connection protocol should guarantee that when $V(x_k) \geq 1$, $V(x_{k+1}) < V(x_k)\ \forall k$. The following theorem, motivated by [8] and [9], sets forth sufficient conditions under which $V(x_{k+1}) < V(x_k)\ \forall k$.

Theorem 1: If the network connection protocol ensures that for some $i \in \{1, \cdots, N\}$,

$$-y_k^{i^T} Y_i y_k^i + e_{k|k}^T \bar{P} e_{k|k} + w_k^T Q w_k + v_k^T R v_k < 0, \tag{27}$$

where $Y_i \succ 0$, and if $\forall i \in \{1, \cdots, N\}$,

$$\begin{bmatrix} P - A_{bk}^T P A_{bk} - C_i^T Y_i C_i & A_{bk}^T P E & -A_{bk}^T P & -C_i^T Y_i \Gamma_i \\ E^T P A_{bk} & \bar{P} - E^T P E & E^T P & 0 \\ -P A_{bk} & P E & Q - P & 0 \\ -\Gamma_i^T Y_i C_i & 0 & 0 & R - \Gamma_i^T Y_i \Gamma_i \end{bmatrix} \succeq 0, \tag{28}$$

where $\Gamma_i \triangleq \begin{bmatrix} 0_{m_i \times \sum_{j=1}^{i-1} m_j} & I_{m_i} & 0_{m_i \times \sum_{j=i+1}^{N} m_j} \end{bmatrix}$, then $V(x_{k+1}) < V(x_k)\ \forall k$ for the system in (14).

Proof: The condition in (28) is equivalent to

$$\begin{aligned} &x_k^T P x_k - (A_{bk} x_k - E e_{k|k} + w_k)^T P (A_{bk} x_k - E e_{k|k} + w_k) \\ &- y_k^{i^T} Y_i y_k^i + e_{k|k}^T \bar{P} e_{k|k} + w_k^T Q w_k + v_k^T R v_k \geq 0, \end{aligned} \tag{29}$$

which is equivalent to

$$V(x_{k+1}) - V(x_k) \leq - y_k^{i^T} Y_i y_k^i + e_{k|k}^T \bar{P} e_{k|k} + w_k^T Q w_k + v_k^T R v_k. \tag{30}$$

Taking (30) in conjunction with the condition in (27) ensures that $V(x_{k+1}) - V(x_k) < 0$. Consequently, $V(x_{k+1}) < V(x_k)\ \forall k$ for the system in (14) when (28) is satisfied $\forall i \in \{1, \cdots, N\}$ and (27) is satisfied for some $i \in \{1, \cdots, N\}$. ■

Note that $\bar{P}$ determines the size of the invariant set $\mathcal{E}_e$ in which the error lies when all agents are connected to the network, P determines the size of the invariant set $\mathcal{E}_x$ to which the state converges, and Y_i will have a direct impact on the frequency at which agent i connects to the network as will be seen in (32). Maximizing $\log\det \bar{P}$ is proportional to minimizing the volume of $\mathcal{E}_e$, compressing the size of the invariant set in which the error lies when all agents are connected to the network. Maximizing $\log\det P$ is proportional to minimizing the volume of $\mathcal{E}_x$, compressing the size of the invariant set to which the state converges. Maximizing $\log\det Y_i$ is proportional to maximizing $y_k^{i^T} Y_i y_k^i\ \forall y_k^i$, which minimizes the number of times agent i connects to the network as will be seen in (32). Consequently, the desired values for $\bar{P}$, P, and Y_i are obtained according to the following optimization problem

$$\begin{aligned} &\underset{\alpha_1, \alpha_2, \bar{P}, P, Y_1, \cdots, Y_N}{\operatorname{argmax}} \quad \omega_e \log\det \bar{P} + \omega_x \log\det P + \sum_{i=1}^{N} \omega_i \log\det Y_i \\ &\text{s.t. } \gamma = 1,\ \alpha_1, \alpha_2 \geq 0,\ \bar{P} \succ 0,\ P \succ 0,\ Y_i \succ 0\ \forall i \in \{1, \cdots, N\}, \\ &(24),\ (25),\ \text{and } (28) \text{ are satisfied with } \delta_k^{ij} = 1\ \forall i, j \in \{1, \cdots, N\}, \end{aligned} \tag{31}$$

where ω_e, ω_x, and ω_i, $i \in \{1, \cdots, N\}$ are nonnegative constants chosen by the designer to weight the importance of minimizing $\mathcal{E}_e$, $\mathcal{E}_x$, and the communication frequency of agent i, respectively. Because this optimization problem is not convex, a suboptimal solution may be obtained by restricting the possible values of α_1 and α_2 to a finite set lying within $[0, \frac{1}{2}]$ and carrying out the optimization problem in (31) over that finite set, since (31) is convex for set values of α_1 and α_2.

C. Triggering Conditions

The following theorem leverages the results of Lemma 3, Lemma 4, and Theorem 1, providing a network connection triggering condition for each agent based on (27) to ensure that the system in (14) is quadratically 1-bounded.

Theorem 2: If $\exists \alpha_1, \alpha_2 \geq 0$ such that (24) and (25) are satisfied with $\gamma = 1$ when $\delta_k^{ij} = 1\ \forall i, j \in \{1, \cdots, N\}$, if $e_{0|0} \in \mathcal{E}_e$, if (28) is satisfied $\forall i \in \{1, \cdots, N\}$, and if agent i connects to the network, communicates with all the other agents on the network according to Algorithm 1, and updates its a posteriori state estimate according to (9) when

$$y_k^{i^T} Y_i y_k^i \leq 2 + \max\left(1, \prod_{\ell=0}^{k-1} \gamma(\delta_\ell)\right), \tag{32}$$

where

$$\gamma(\delta_k) \triangleq \min_{\alpha_1, \gamma} \gamma \text{ s.t. } \alpha_1, \gamma \geq 0,\ (24) \text{ is satisfied}, \tag{33}$$

then the system in (14) is quadratically 1-bounded.

Proof: By applying Lemma 2 and Definition 1 to the system in (22), the definition of $\gamma(\delta_k)$ in (33) implies that $\forall w_k \in W$ and $\forall v_k \in V$,

$$e_{k|k}^T \bar{P} e_{k|k} \geq 1 \implies e_{k+1|k+1}^T \bar{P} e_{k+1|k+1} < \gamma(\delta_k) e_{k|k}^T \bar{P} e_{k|k}. \tag{34}$$

Consequently,

$$e_{k|k}^T \bar{P} e_{k|k} < \max\left(1, \left(\prod_{\ell=0}^{k-1} \gamma(\delta_\ell)\right) e_{0|0}^T \bar{P} e_{0|0}\right). \tag{35}$$

If $e_{0|0} \in \mathcal{E}_e$, then $\max_{e_{0|0}} e_{0|0}^T \bar{P} e_{0|0} = 1$, implying that

$$e_{k|k}^T \bar{P} e_{k|k} < \max\left(1, \prod_{\ell=0}^{k-1} \gamma(\delta_\ell)\right). \tag{36}$$

The trigger condition in (32) can be written equivalently as

$$-y_k^{i^T} Y_i y_k^i + \max\left(1, \prod_{\ell=0}^{k-1} \gamma(\delta_\ell)\right) + 2 \geq 0. \tag{37}$$

Note that $\forall w_k \in W$ and $\forall v_k \in V$,

$$\begin{aligned} -y_k^{i^T} Y_i y_k^i + \max\left(1, \prod_{\ell=0}^{k-1} \gamma(\delta_\ell)\right) + 2 > \\ -y_k^{i^T} Y_i y_k^i + e_{k|k}^T \bar{P} e_{k|k} + w_k^T Q w_k + v_k^T R v_k \end{aligned} \tag{38}$$

according to (36). Consequently, (32) functions as an upper bound on the condition in (27) so that whenever (27) is not satisfied, the condition in (32) will be triggered.

If (28) is satisfied $\forall i \in \{1, \cdots, N\}$, Theorem 1 states that $V(x_{k+1}) < V(x_k)\ \forall k$ when (32) is not triggered for at least one agent, implying that the system in (14) is quadratically 1-bounded when (32) is not triggered for some $i \in \{1, \cdots, N\}$. In the case where (32) is triggered $\forall i \in \{1, \cdots, N\}$, all agents will be connected to the network. In this situation, Lemma 3 and Lemma 4 state that if $\exists \alpha_1, \alpha_2 \geq 0$ such that (24) and (25) are satisfied with $\gamma = 1$ and if $e_{0|0} \in \mathcal{E}_e$, the system in (14) will be quadratically 1-bounded. ■

Theorem 2 ensures that when the magnitude of the error $e_{k|k}^T \bar{P} e_{k|k}$ grows too large, (32) will be triggered and agent i will connect to the network to communicate data with other agents according to Algorithm 1. Note that in (32), y_k^i is locally available to agent i, but $\delta_{0:k-1}$ contains some information that is not locally available to agent i. Let $\bar{\delta}_\ell^i$ and $\hat{\delta}_\ell^i$ represent the portions of δ_ℓ whose values are known and unknown by agent i, respectively, so that $\delta_\ell = \{\bar{\delta}_\ell^i, \hat{\delta}_\ell^i\}\ \forall i \in \{1, \cdots, N\}$. For agent i to evaluate this trigger condition, it first solves for $\gamma(\delta_\ell)$ offline for all possible values of δ_ℓ such that $\delta_\ell^{ii} = 1\ \forall i \in \{1, \cdots, N\}$. Agent i then plugs in $\bar{\gamma}_i(\delta_\ell)$ for $\gamma(\delta_\ell)$ in (32) to evaluate it online, where $\bar{\gamma}_i(\delta_\ell)$ is given by

$$\bar{\gamma}_i(\delta_\ell) \triangleq \max_{\hat{\delta}_\ell^i} \gamma(\delta_\ell) \text{ s.t. } \delta_\ell = \{\bar{\delta}_\ell^i, \hat{\delta}_\ell^i\}. \tag{39}$$

In this way, agent i always evaluates the trigger condition with values that result in the right side of the inequality in (32) being greater than or equal to its actual value. This then functions as an upper bound on the actual value of the condition in (32) which is itself an upper bound on the condition in (27), implying that whenever (27) is not satisfied, the condition in (32) evaluated with $\bar{\gamma}_i(\delta_\ell)$ will be triggered.

Remark 2: Note that computing $\gamma(\delta_k)$ in (33) for all possible values of δ_k such that $\delta_k^{ii} = 1\ \forall i \in \{1, \cdots, N\}$ requires evaluating between $2^N - N$ and $2^{N(N-1)}$ linear matrix inequalities (LMIs) since these are lower and upper bounds, respectively, on the number of possible permutations for the values in δ_k subject to the data communication procedure of Algorithm 1. To ensure that every permutation is covered, all $2^{N(N-1)}$ LMIs can be evaluated, but this will include permutations of δ_k that cannot possibly occur according to the data communication procedure of Algorithm 1. For instance, if $\delta_k^{ij} = 1$ for some $i \neq j$, then there must exist an $\ell \neq i$ such that $\delta_k^{i\ell} = \delta_k^{\ell i} = 1$. This is due to the fact that in order for agent i to possess information about agent j where $i \neq j$, it must receive that information from some agent $\ell \neq i$. Consequently, $\delta_k^{i\ell} = \delta_k^{\ell i} = 1$ since agents i and ℓ share their own data with each other when connected to the network. If lines 11-17 were eliminated from Algorithm 1, then agents would only share data about themselves and never about other agents, causing $\delta_k^{ij} = \delta_k^{ji}$ $\forall i, j \in \{1, \cdots, N\}$ and decreasing overall performance. This would result in a total of $2^N - N$ possible permutations for the values in δ_k so that exactly $2^N - N$ LMIs would need to be evaluated. However, all of this computation is completed offline ahead of time and grows with the number of agents N, not the number of control inputs p or sensor measurements m. Furthermore, (33) does not need to be evaluated at each time step since the set of possible values for δ_k is time-invariant. Future work includes addressing cases where this offline calculation becomes computationally intractable with large N.

D. Network Connection Procedure

Algorithm 2 describes a procedure which ensures the quadratic 1-boundedness of the overall system as guaranteed by Theorems 1 and 2 when (24), (25), and (28) are satisfied. The information available to agent i at time step k is given by

$$\begin{aligned} \mathcal{J}_k^i \triangleq \big\{ & A, B, C, Q, R, K, \hat{L}(\delta_k^i), \bar{P}, P, Y_1, \cdots, Y_N, N, \hat{x}_{k_i'|k_i'}^i \\ & \hat{x}_{k|k-1}^i, \hat{x}_{k|k}^i, \bar{\delta}_{0:k}^i, \{u_{\bar{k}:\kappa_k^{ij}}^j, y_{\bar{k}:\kappa_k^{ij}}^j\}\ \forall j \in \{1, \cdots, N\} \big\}. \end{aligned}$$

The event-triggered communication procedure presented in Algorithm 2 uses this local information $\mathcal{J}_k^i$ to indicate when agent i needs to connect to the network and communicate with other agents. At each time step, agent i first computes its a priori state estimate according to (6). It then uses its local sensor measurements y_k^i as well as $\bar{\gamma}_i(\delta_{0:k})$ to evaluate the trigger condition in (32). If (32) is not satisfied, then agent i sets its a posteriori state estimate equal to its a

Algorithm 2 Network Connection Procedure for Agent i

1: Initialize $\hat{x}_{0|0}^i\ \forall i \in \{1, \cdots, N\}$ so that $e_{0|0} \in \mathcal{E}_e$
2: $k = 0$
3: **while** $k \geq 0$
4: $\quad \hat{x}_{k+1|k}^i = A_{bk} \hat{x}_{k|k}^i + L_i(y_k^i - C_i \hat{x}_{k|k}^i)$
5: $\quad$ Update $\bar{\delta}_k^i$ with $\delta_k^{ij} = 1\ \forall j \in \{1, \cdots, N\}$
6: $\quad$ **if** (32) is satisfied
7: $\quad\quad$ Open network connection
8: $\quad\quad$ Share data with other agents according to Algorithm 1
9: $\quad\quad$ Update $\hat{x}_{k+1|k+1}^i$ according to (9)
10: $\quad\quad$ Update $\bar{\delta}_{0:k}^i$ with information from θ_k^i, $\bar{\theta}_k^i$, and $\bar{\delta}_{0:k}^j$ $\forall j \in \theta_k^i$
11: $\quad\quad$ Close network connection
12: $\quad$ **else**
13: $\quad\quad \hat{x}_{k+1|k+1}^i = \hat{x}_{k+1|k}^i$
14: $\quad\quad$ Update $\bar{\delta}_k^i$ with $\delta_k^{ij} = \delta_k^{ji} = 0\ \forall j \in \{1, \cdots, N\}, j \neq i$
15: $\quad$ **end if**
16: $\quad k = k + 1$
17: **end while**

priori state estimate and updates its local information about δ_k according to the fact that it is not connected to the network. If (32) is satisfied, then agent i connects to the network, communicates with all the other agents on the network according to Algorithm 1, updates its a posteriori state estimate according to (9), and uses information from other agents to update its local information about δ_k before proceeding to disconnect from the network.

Remark 3: Note that because (32) functions as an upper bound on the condition in (27), agents will connect to the network more often than is necessary. Consequently, there may be instances where agent i attempts to send information to other agents who are not connected to the network to receive that information. However, as long as (28) is satisfied $\forall i \in \{1, \cdots, N\}$, Theorems 1 and 2 imply that each one of these instances will only occur when the information being transmitted is not necessary for maintaining the stability of the overall system.

Remark 4: Note that Algorithm 2 presents a procedure where an agent's sending and receiving capabilities are simultaneously triggered by the condition in (32). However, an attack on an agent is initiated through data that is incoming to that agent, not outgoing from that agent. Consequently, data could constantly be broadcast to agents all the time, while (32) would only be used for deciding when to receive information from other agents. In this case, agent i would send $\{u^i_{k:k}, y^i_{k:k}\}$ to each agent $j \in \{1, \cdots, N\}$, $j \neq i$ at every time step. By doing so, an agent receiving information would possess the full set of inputs and outputs for the overall system, reducing that agent's state estimation error compared to the current scenario where only a subset of the inputs and outputs may be received. This in turn would decrease the number of times (32) is triggered since (32) is a function of the estimation error, further reducing an adversary's window of opportunity to carry out an attack. However, this approach would increase communication costs considerably since all agents would always be broadcasting information at every time step. The implementation of this approach, along with an investigation of the tradeoff between overall performance and communication costs, is left for future work.

E. Ensuring Resiliency Against Attacks

The network connection procedure presented in Algorithm 2 is sufficient for ensuring that agents connect to the network when necessary to maintain the stability of the overall system in attack-free scenarios. However, during those brief periods of time when various agents are connected to the network, the safety of the overall system against attacks is not guaranteed. Since resilience against attacks is the ultimate goal, a variety of mechanisms and strategies may be used during these brief periods of network connection to guarantee safety and security. For example, software rejuvenation [18], [19] is one mechanism that has been introduced to guarantee the safety of agents when connecting to the network to maintain stability or recover from a disturbance. The detailed implementation of such a mechanism within the context of the network connection protocol in Algorithm 2 is beyond the scope of this paper and is left for future work. However, to guarantee the safety of the overall system in the presence of attacks, some such resiliency mechanism will need to be implemented during those brief periods of time when various agents connect to the network and share critical information.

IV. Simulation

To illustrate the effectiveness of the network connection and communication protocol, we consider a smart water distribution system used at a four-hectare wine estate in the south of England [13]. The goal of the water distribution system is to stabilize the water levels of three district meter area tanks at predesigned constant reference levels. The system state is given by the difference between the reference levels and the current water levels, the control inputs are the open levels of the valves, and the sensors measure the current water levels of the tanks. The system model is linearized at a reference level of 3 m as presented in [13] and is given by

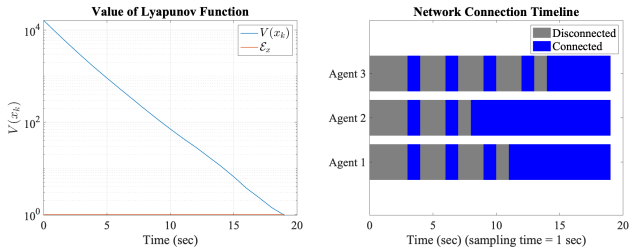
$$x_{k+1} = \begin{bmatrix} 0.9992 & 0 & 0 \\ 0 & 0.9994 & 0 \\ 0 & 0 & 0.9995 \end{bmatrix} x_k + \begin{bmatrix} 0.1068 & -0.0371 & -0.0371 \\ -0.0279 & 0.0801 & -0.0279 \\ -0.0223 & -0.0223 & 0.0641 \end{bmatrix} u_k + w_k, \tag{40}$$

$$y_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} x_k + v_k, \tag{41}$$

where the system in [13] is discretized using a zero-order hold with a sampling time of 1 sec. We let $Q = \frac{10000}{3} I$ and $R = \frac{10000}{3} I$ so that the process and measurement disturbances are less than 1 cm for each tank.

A discrete-time controller K with poles at 0.7, -0.7, and 0.8 is designed to stabilize the system when all agents are connected to the network, and discrete-time observers $\hat{L}(\delta^i_k)$ are designed for all possible values of δ^i_k according to (7) so that the estimation error for the observable states is stabilized. The water distribution system is comprised of $N = 3$ agents, where each agent has access to one local control input and one local sensor measurement. We solve for $\bar{P}$, P, and Y_i $\forall i \in \{1, \cdots, N\}$ according to (31) with $\omega_e = \omega_x = 1$ and $\omega_i = 100$ $\forall i \in \{1, \cdots, N\}$. The network connection protocol in Algorithm 2 is executed for each agent from the initial state $x_0 = \begin{bmatrix} 10 & 10 & 10 \end{bmatrix}^T$.

Figures 1a and 1b depict the Lyapunov function convergence and network connection timeline, respectively, for a particular simulation. As seen in Figure 1a, the Lyapunov function continually decreases until it is less than 1, equivalent to the state converging to the invariant set $\mathcal{E}_x$, demonstrating the quadratic 1-boundedness of the system. Figure 1b depicts the detailed connection timeline for each agent, showing that agents are able to disconnect from the network for approximately 42% of the time before converging to $\mathcal{E}_x$. This provides less time for adversaries to attack different agents while also ensuring the stability of the overall system when there is no attack.



(a) Value of Lyapunov Function (b) Network Connection Timeline

Fig. 1. (a) Convergence of the Lyapunov function to the invariant set $\mathcal{E}_x$ (b) Timeline of agents' network connections (agents remain disconnected approximately 42% of the time before converging to $\mathcal{E}_x$)

Note that the network connection times may vary for each agent since each agent has different sets of local sensors and since the dynamics of some agents may be more tightly coupled to one another than the dynamics of other agents, requiring some agents to connect to the network more than others to ensure the stability of the overall system. In addition, no performance is lost in using the decentralized event-triggered network connection protocol. Over 1000 trials, the time taken to converge to the invariant set $\mathcal{E}_x$ remains the same regardless of whether communication between agents occurs all the time (average convergence time of 19.973 sec) or whether agents disconnect from the network for periods of time (average convergence time of 19.982 sec).

V. Conclusion

This paper has investigated using decentralized event-triggered control to reduce vulnerabilities to attacks. An event-triggered mechanism for network connection and communication is designed based on only local information. This mechanism ensures the stability of the overall system in the sense of quadratic boundedness for attack-free scenarios. It also allows agents to disconnect from the network for periods of time, minimizing an adversary's window of opportunity when attacking different agents. A network connection protocol is designed which uses this event-triggered mechanism, and its effectiveness is illustrated in the context of a smart water distribution system. To ensure safety and security against attacks, future work should introduce resiliency mechanisms for those times when agents are connected to the network and are vulnerable to attacks. Future work also includes considering cases where the communication graph is not complete so that each agent can only directly send information to a subset of agents. Lastly, future work includes considering scenarios where non-negligible communication delays exist when sending data over the network.

Acknowledgments

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.
Carnegie Mellon® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University. DM21-0268

References

[1] A. Cardenas, S. Amin, and S. Sastry, "Secure control: Towards survivable cyber-physical systems," in *28th International Conf. on Distributed Computing Systems Workshops, 2008. ICDCS'08.*, 2008.
[2] J. Slay and M. Miller, "Lessons learned from the maroochy water breach," in *International Conference on Critical Infrastructure Protection*. Springer, 2007, pp. 73–82.
[3] D. U. Case, "Analysis of the cyber attack on the Ukrainian power grid," *Electricity Information Sharing and Analysis Center*, 2016.
[4] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Security & Privacy*, vol. 9, no. 3, pp. 49–51, 2011.
[5] X. Sun, R. Horowitz, and C.-W. Tan, "An efficient lane change maneuver for platoons of vehicles in an automated highway system," in *ASME International Mechanical Engineering Congress and Exposition*, vol. 37130, 2003, pp. 355–362.
[6] J. Huang, Q. Huang, Y. Deng, and Y.-H. Chen, "Toward robust vehicle platooning with bounded spacing error," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 4, pp. 562–572, 2016.
[7] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, 2012, pp. 3270–3285.
[8] M. Donkers, "Networked and event-triggered control systems," *PhD diss., Eindhoven: Technische Universiteit Eindhoven*, 2011.
[9] W. P. M. H. Heemels, M. C. F. Donkers, and A. R. Teel, "Periodic event-triggered control for linear systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 4, pp. 847–861, April 2013.
[10] M. Donkers and W. Heemels, "Output-based event-triggered control with guaranteed $\mathcal{L}_\infty$-gain and improved and decentralized event-triggering," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1362–1376, 2011.
[11] V. S. Dolk, D. P. Borgers, and W. P. M. H. Heemels, "Output-based and decentralized dynamic event-triggered control with guaranteed $\mathcal{L}_p$-gain performance and zeno-freeness," *IEEE Transactions on Automatic Control*, vol. 62, no. 1, pp. 34–49, 2017.
[12] W. Heemels and M. Donkers, "Model-based periodic event-triggered control for linear systems," *Automatica*, vol. 49, no. 3, pp. 698–711, 2013.
[13] A. Fu and J. A. McCann, "Dynamic decentralized periodic event-triggered control for wireless cyber-physical systems," *IEEE Transactions on Control Systems Technology*, 2020.
[14] A. Fu and M. Mazo Jr, "Decentralized periodic event-triggered control with quantization and asynchronous communication," *Automatica*, vol. 94, pp. 294–299, 2018.
[15] P. Griffioen, R. Romagnoli, B. H. Krogh, and B. Sinopoli, "Decentralized event-triggered control in the presence of adversaries," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 3236–3242.
[16] A. Alessandri, M. Baglietto, and G. Battistelli, "On estimation error bounds for receding-horizon filters using quadratic boundedness," *IEEE Transactions on Automatic Control*, vol. 49, no. 8, pp. 1350–1355, Aug 2004.
[17] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. SIAM, 1994.
[18] P. Griffioen, R. Romagnoli, B. H. Krogh, and B. Sinopoli, "Secure networked control via software rejuvenation," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 3878–3884.
[19] ——, "Secure networked control for decentralized systems via software rejuvenation," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 1266–1273.