Reducing Attack Opportunities Through Decentralized Event-Triggered Control

Paul Griffioen ⊕, Member, IEEE, Raffaele Romagnoli ⊕, Member, IEEE, Bruce H. Krogh ⊕, Life Fellow, IEEE, and Bruno Sinopoli ⊕, Fellow, IEEE

Abstract—Cyber-physical systems are prevalent in many critical infrastructures and are vulnerable to a variety of attacks from the network. Decentralized control systems are particularly vulnerable since they rely heavily on network communication to achieve their goals. To address network vulnerabilities, we present an event-triggered network connection and communication protocol that minimizes the amount of time agents are connected to the network, reducing the window of opportunity for attacks. 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 considering an undirected and connected communication graph and by not assuming that agents are always connected to the network to receive critical information from other agents. An algorithm describing this network connection and communication protocol is provided, and our approach is illustrated via simulation.

Index Terms—Communication networks, cyber-physical systems, decision/estimation theory, networked control systems.

I. INTRODUCTION

Cyber-physical systems, engineered systems that include sensing, communication, and control in physical spaces, are essential to secure and protect in today’s society. The cyber-physical systems are ubiquitous in modern critical infrastructures, including the smart grid, transportation systems, healthcare, 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], [2], [3], [4].

One of the primary ways these systems are vulnerable is through their connections to the network. Adversaries can corrupt information that is sent over the network and corrupt the system’s control software by using the network connection to inject malicious code [5]. If the control system is centralized, then it can protect itself from these kinds of attacks by disconnecting from the network [6]. However, this is not an option for decentralized control systems since they rely heavily on network communication to achieve their goals. Since individual agents have access to differing amounts of information, agents must communicate some subset of their local information with other agents to maintain the stability of the overall system and achieve a global objective, for instance, in car platoons [7], [8].

While approaches, such as decentralized event-triggered control, are useful for minimizing the amount of communication, they do not address the issue of minimizing the vulnerability to attacks from the network. This is due to the fact that decentralized control schemes simply reduce the amount of outgoing communication from each agent, assuming that all the agents are always connected to the network and are always able to receive any information that is broadcast to them. Consequently, this constant incoming network connection is a continual source of vulnerability to attacks from the network. Different approaches to event-triggered control are summarized well in [9]. A variety of decentralized event-triggered control mechanisms for linear, nonlinear, continuous-time, and discrete time systems is presented in [10], [11], [12], [13], [14], [15], and [16] and provides conditions under which global asymptotic stability, global exponential stability, $L_\infty$ gain performance, and $L_p$ gain performance are achieved.

In contrast to these schemes, this article addresses the constant network connection vulnerability present in decentralized event-triggered control. Rather than holding a constant network connection all the time, we present a scheme where agents intermittently connect and disconnect from the network. This reduces the window of opportunity for attacks while also providing a framework in which a resilience strategy may be implemented. We note that intermittent network connection is necessary in certain marine robotics applications, where underwater vehicles must surface in order to share information with one another [17]. Our scheme is carried out through event-based network connection and communication, not simply event-based communication as is the case with decentralized event-triggered control.
a decentralized self-triggered scheme for intermittent network connection is presented in [18], it assumes that each agent has direct access to the state, whereas our scheme includes sensor measurements and output feedback in the system model. We present a protocol that uses trigger conditions 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 in attack-free scenarios by having an agent connect to the network when the magnitude of the state estimation error grows too large.

The main contributions of this article relative to our previous work in [19] and [20] are as follows.

1) In contrast to [19], we present a decentralized event-triggered network connection protocol that includes sensor measurements in the system model and does not require any reachability analysis to be implemented, which can be computationally costly.

2) In this article, only two observer gain matrices need to be designed for each agent, and only two linear matrix inequalities (LMIs) need to be evaluated for the error dynamics. In contrast, $2^N - 1$ observer gain matrices and $2^N - N$ LMIs are needed in [20], where $N$ is the number of agents, so this becomes computationally intractable with a large number of agents.

3) This article is presented in continuous time with extensions to discrete time, unlike [19] and [20], which are only presented in discrete time.

4) The restrictive assumption of a complete communication graph that is required in [19] and [20] is relaxed in this article so that only the assumption of an undirected and connected communication graph is needed.

The rest of this article 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 the stability of the overall system is achieved. Simulation results are presented in Sections IV. Finally, Section V concludes this article.

II. Problem Formulation

A. Preliminaries

A communication graph at time $t$ is denoted by $G(t) = (\mathcal{N}, A(t))$, where $\mathcal{N} = \{1, \ldots, N\}$ is a finite nonempty set of agents and $A(t) \in \mathbb{R}^{N \times N}$ represents the adjacency matrix at time $t$, whose $(i, j)$th element is given by $a_{ij}(t) \in \{0, 1\}$. If $a_{ij}(t) = 1$, agent $j$ is able to send information to agent $i$ at time $t$, and if $a_{ij}(t) = 0$, agent $j$ is unable to send information to agent $i$ at time $t$. A communication graph is undirected at time $t$ if and only if $a_{ij}(t) = a_{ji}(t)$ for all $i, j \in \mathcal{N}$. Self-connection is excluded so that $a_{ii}(t) = 0 \forall i \in \mathcal{N}, \forall t.$

A path of length $\ell$ from agent $i$ to agent $j$ at time $t$ is a sequence $(i_0, \ldots, i_{\ell})$ of agents such that $i_0 = i, i_{\ell} = j,$ and $a_{i_k,i_k-1}(t) = 1$ for all $k = 1, \ldots, \ell$. An undirected communication graph is connected at time $t$ if there is a path from $i$ to $j$ at time $t$ for two arbitrary distinct agents $i, j \in \mathcal{N}$.

The Laplacian of $G(t)$ is the matrix $L(t)$, where the $(i, j)$th element is given by $l_{ij}(t)$. The Laplacian is defined as $l_{ij}(t) = \sum_{j=1}^{N} a_{ij}(t)$ and $l_{ij}(t) = -a_{ij}(t), i \neq j.$ Note that $L(t)1_N = 0 \forall t$, where $1_N$ represents an $N \times 1$ column vector comprised of all ones. By its construction, $L(t)$ contains a zero eigenvalue with a corresponding eigenvector $1_N$, and all the other eigenvalues lie in the closed right-half complex plane. For undirected communication graphs at time $t$, both $A(t)$ and $L(t)$ are symmetric and have real eigenvalues.

Let $\tilde{L}$ be the Laplacian for an undirected and connected communication graph. It is well known that $\tilde{L}$ has a simple zero eigenvalue, or, in other words, all the other eigenvalues are real positive. In this case, by Schur’s lemma, there exists a normal (real unitary) matrix $U \in \mathbb{R}^{N \times N}$ such that $U\tilde{L}U^T = \text{Diag}(0, \Lambda^+)$, where $\Lambda^+ \in \mathbb{R}^{(N-1) \times (N-1)}$ is the diagonal matrix with entries corresponding to the positive eigenvalues of $\tilde{L}$. Given the properties of the Laplacian matrix, the first row of $U$ will be $\frac{1}{\sqrt{N}} 1_N^T$. Then, there is a matrix $S \in \mathbb{R}^{N \times (N-1)}$ such that

$$U = \left[ \frac{1}{\sqrt{N}} 1_N^T \right], \left[ \frac{1}{\sqrt{N}} 1_N^T \right] \tilde{L} \left[ \frac{1}{\sqrt{N}} 1_N \right] S = \begin{bmatrix} 0 & 0 \\ 0 & \Lambda^+ \end{bmatrix}. \quad (1)$$

In other words, there is a matrix $S \in \mathbb{R}^{N \times (N-1)}$ such that

$$1_N^T S = 0, \quad S^T S = I_{N-1}, \quad S^T \tilde{L} S = \Lambda^+. \quad (2)$$

B. System Model

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

$$\dot{x}(t) = Ax(t) + Bu(t) + w(t) \quad (3)$$

$$y(t) = Cx(t) + v(t) \quad (4)$$

where $x(t) \in \mathbb{R}^n$ represents the state at time $t$, $u(t) \in \mathbb{R}^p$ denotes the control input, $y(t) \in \mathbb{R}^m$ represents the sensor measurements, and $w(t) \in \mathbb{R}^n$ and $v(t) \in \mathbb{R}^m$ are bounded disturbances that lie in the compact sets $W$ and $V$, respectively, given by

$$W \triangleq \{ w(t) | w(t)^T Q w(t) \leq 1 \} \quad (5)$$

$$V \triangleq \{ v(t) | v(t)^T R v(t) \leq 1 \}. \quad (6)$$

We let $y_i(t) \in \mathbb{R}^{m_i}$ represent the sensor measurements that are locally available to agent $i$ so that $y(t) = [y_1(t)^T \cdots y_N(t)^T]^T$, $m = \sum_{i=1}^{N} m_i$, $C \triangleq [C_1^T \cdots C_N^T]^T$, and $C_i \in \mathbb{R}^{m_i \times n}$. Similarly, we let $u_i(t) \in \mathbb{R}^{p_i}$ denote agent $i$’s control inputs so that $u(t) = [u_1(t)^T \cdots u_N(t)^T]^T$, $p = \sum_{i=1}^{N} p_i$, $B \triangleq [B_1 \cdots B_N]$, and $B_i \in \mathbb{R}^{n \times p_i}$.

Each agent $i$ is able to directly access its own local sensor measurements $y_i(t)$ and control inputs $u_i(t)$ but must rely on communication from other agents to access any information locally available to other agents. We assume that the same underlying undirected and connected communication graph is always present, given by $G = (\mathcal{N}, A)$ with Laplacian matrix $\tilde{L}$ and with $\mathcal{N}$ denoting the set of agent $i$’s neighbors. When all the agents...
are connected to the network, they are able to communicate with each other according to this underlying communication graph $\mathcal{G}$. Consequently, if all the agents are connected to the network at time $t$, $\mathcal{G}(t) = \mathcal{G}$ so that $\hat{A}(t) = \hat{A}$ and $\mathcal{L}(t) = \hat{\mathcal{L}}$. If agent $i$ disconnects from the network at time $t$, it can no longer communicate with its neighbors, so $a_{ij}(t) = a_{ji}(t) = 0$ $\forall j \in \mathcal{N}_i$. When agent $i$ connects to the network at time $t$, it is able to communicate with its neighbors who are also connected to the network at time $t$. Consequently, $a_{ij}(t) = a_{ji}(t) = 1$ for all the agents $j \in \mathcal{N}_i$ connected to the network at time $t$.

### C. State Estimation

The control input for agent $i$ is given by

$$u_i(t) = K_i \hat{x}_i(t)$$

where $\hat{x}_i(t) \in \mathbb{R}^n$ is agent $i$’s estimate of the overall state. Motivated by the distributed observer in [21], this estimate is computed according to the following scheme. Let $K \triangleq \begin{bmatrix} K_1^T & \cdots & K_N^T \end{bmatrix}^T$, $A_{bk} \triangleq A + BK$, and $L \triangleq \begin{bmatrix} L_1 & \cdots & L_N \end{bmatrix}$ be designed so that $A - LC$ is Hurwitz, where $L_i \in \mathbb{R}^{n \times m_i}$. Then, let $L_i$ be agent $i$’s observer gain matrix, given by

$$\hat{L}_i \triangleq T_i \begin{bmatrix} L_i^0 & 0 \\ 0 & 0 \end{bmatrix}, \quad T_i^{-1}AT_i = \begin{bmatrix} A_i^0 & 0 \\ A_i^{21} & A_i^o \end{bmatrix}, \quad C_iT_i = \begin{bmatrix} C_i^0 & 0 \end{bmatrix}$$

where $L_i^0$ is designed so that $A_i^o - L_i^o C_i^o$ is Hurwitz, 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. Finally, we let

$$\hat{L}_i(t) \triangleq \begin{cases} \hat{L}_i, & \ell = 0 \\ NL_i, & \forall \ell \in \mathbb{Z}^+ \end{cases}$$

Then, agent $i$’s estimate of the overall state is computed according to

$$\hat{x}_i(t) = A_{bk}\hat{x}_i(t) + \hat{L}_i(l_i(t))y_i(t) - C_i\hat{x}_i(t)$$

$$+ \sum_{j=1}^{N} a_{ij}(t)(\hat{x}_j(t) - \hat{x}_i(t)).$$

The observer presented in (10) is simply a Luenberger observer that uses only local sensor measurements $y_i(t)$ when agent $i$ is not connected to the network. When agent $i$ is connected to the network, it broadcasts the current value of its state estimate to its neighbors, receives the current values of the state estimates of its neighbors who are connected to the network, and incorporates them into the coupling term of the observer, where $\eta \geq 0$ is the coupling gain. By having agents intermittently connect and disconnect from the network, data are only sent over the network during intermittent intervals as opposed to being sent in a continuous stream over time. Note that agents do not need to have the same initial state estimate $\hat{x}_i(0)$.

**Remark 1:** Note that one of the benefits of this state estimation scheme compared to the one presented in [20] is that each agent only uses two observer gain matrices, i.e., $\hat{L}_i(t) = \hat{L}_i$ and $\hat{L}_i(t) = NL_i \forall \ell \in \mathbb{Z}^+$. In contrast, $2^N - 1$ observer gain matrices are needed in [20].

### D. Problem Formulation

Given the controller in (7), the system dynamics in (3) can be written as

$$\dot{x}(t) = A_{bk}x(t) - Ec(t) + w(t)$$

where $E \triangleq [B_1^\top \cdots B_N^\top]$, $e(t) \triangleq [e_1(t)^T \cdots e_N(t)^T]^T$, and $e_i(t) \triangleq x(t) - \hat{x}_i(t)$ so that $e(t) \in \mathbb{R}^{Nn}$ and $e_i(t) \in \mathbb{R}^n$. Here, $e_i(t)$ represents the error between the overall state and agent $i$’s 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 Boundness

In order to maintain the stability and safety of the overall system, each agent $i$ must occasionally connect to the network and share information with its neighboring 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, which includes information that has been received from other agents through past communications. The stability that we design the network connection protocol to achieve is quadratic $\gamma$-boundedness, which is described in Definitions 1, 2, and Lemma 1. These definitions and lemmas have been uniquely modified and adapted from [22] by adding the parameter $\gamma$ to not only specify when the Lyapunov function decreases or increases but also how fast it does so.

**Definition 1 (see [22]):** Let $z(t)$ represent a state vector, let $d_1(t)$ and $d_2(t)$ represent disturbance vectors, and let $\mathcal{D}_1$ and $\mathcal{D}_2$ be compact sets. A system of the form

$$\dot{z}(t) = Az(t) + B_1d_1(t) + B_2d_2(t), \quad d_1(t) \in \mathcal{D}_1, \quad d_2(t) \in \mathcal{D}_2$$

is quadratically $\gamma$-bounded with $\gamma \in \mathbb{R}$ and symmetric positive-definite Lyapunov matrix $P$ if and only if $\|d_1(t)\|_P \leq 1$ and $\|d_2(t)\|_P \leq 1$.

**Lemma 1 (see [22]):** The following two statements are equivalent:

1. System (12) is quadratically $\gamma$-bounded with $\gamma \leq 0$ and symmetric positive-definite Lyapunov matrix $P$.
2. The set $Z \triangleq \{z(t)^TPz(t) \leq 1, \quad P > 0\}$ is a robustly positively invariant set for (12).
Given these definitions, Lemma 2 provides a sufficient condition for evaluating the quadratic γ-boundedness of (12).

Lemma 2: Let $D_1 \triangleq \{d_1(t) \mid d_1(t)^T D_1 d_1(t) \leq 1, D_1 \succ 0\}$ and $D_2 \triangleq \{d_2(t) \mid d_2(t)^T D_2 d_2(t) \leq 1, D_2 \succ 0\}$ for the system in (12). If $\exists \alpha \geq 0$ such that

$$
\begin{bmatrix}
(\gamma - 2\alpha)P - A^T P - PA - PB_1 & -PB_2 \\
-B_1^T P & \alpha D_1 \\
-B_2^T P & \alpha D_2
\end{bmatrix} > 0
$$

(14)

then the system in (12) is quadratically γ-bounded with symmetric positive-definite Lyapunov matrix $P$.

Proof: By using the $S$-procedure [23], (14) is equivalent to

$$
\begin{bmatrix}
z(t) \\
d_1(t) \\
d_2(t)
\end{bmatrix}^T
\begin{bmatrix}
-2P & 0 & 0 \\
0 & D_1 & 0 \\
0 & 0 & D_2
\end{bmatrix}
\begin{bmatrix}
z(t) \\
d_1(t) \\
d_2(t)
\end{bmatrix} \leq 0 \implies
$$

$$
\begin{bmatrix}
z(t) \\
d_1(t) \\
d_2(t)
\end{bmatrix}^T
\begin{bmatrix}
AT^T + PA - \gamma P & PB_1 & PB_2 \\
B_1^T P & 0 & 0 \\
B_2^T P & 0 & 0
\end{bmatrix}
\begin{bmatrix}
z(t) \\
d_1(t) \\
d_2(t)
\end{bmatrix} < 0
$$

(15)

which in turn is equivalent to

$$
-2z(t)^T P z(t) + d_1(t)^T D_1 d_1(t) + d_2(t)^T D_2 d_2(t) \leq 0
$$

implies

(16)

Note that

$$
\begin{bmatrix}
z(t)^T P z(t) \\
d_1(t)^T D_1 d_1(t) \\
d_2(t)^T D_2 d_2(t)
\end{bmatrix} \geq \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \implies
$$

$$
-2z(t)^T P z(t) + d_1(t)^T D_1 d_1(t) + d_2(t)^T D_2 d_2(t) \leq 0.
$$

(17)

Taking (16) and (17) in conjunction with one another yields that $\forall d_1(t) \in D_1$ and $\forall d_2(t) \in D_2$

$$
z(t)^T P z(t) \geq 1 \implies \frac{d}{dt} (z(t)^T P z(t)) < \gamma z(t)^T P z(t)
$$

(18)

implying that the system in (12) is quadratically γ-bounded with symmetric positive-definite Lyapunov matrix $P$.

Lemma 3 provides a sufficient condition under which the overall system is quadratically 0-bounded and the state remains in the robust positive invariant set $E_x$ given by

$$
E_x \triangleq \{x(0) \mid x(0)^T P x(0) \leq 1, P \succ 0\}.
$$

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

$$
\begin{bmatrix}
-2\alpha_1 P & -A^T b_k & -P & P & 0 \\
P & \alpha_1 P & 0 & 0 & Q
\end{bmatrix} > 0
$$

(20)

then the system in (11) is quadratically 0-bounded with symmetric positive-definite Lyapunov matrix $P$ when $e(t) \in E_x$, where $E_x$ is given by

$$
E_x \triangleq \{x(0) \mid x(0)^T P x(0) \leq 1, P \succ 0\}.
$$

(21)

Furthermore, if $x(0) \in E_x$, then $x(t) \in E_x \forall t \geq 0$.

Proof: Applying Lemma 2 to the system in (11) implies that if (20) is satisfied, then the system in (11) is quadratically 0-bounded. Lemma 1 and Definition 2 imply that if $x(0) \in E_x$, then $x(t) \in E_x \forall t \geq 0$.

According to (3), (4), (7), and (10), the error dynamics for agent $i$ are given by

$$
\dot{e}_i(t) = (A_{bk} - \tilde{L}_i(l_{ii}(t))C_i)e_i(t) + w(t)
$$

(22)

The error dynamics for the overall system are then given by

$$
\dot{e}(t) = (F(L(t)) - \eta(L(t) \otimes I_n))e(t) + \mathcal{W}(t) - J(L(t))v(t)
$$

(23)

where $\mathcal{W} \triangleq [I_n \cdots I_n]^T$, $J(L(t)) \triangleq \text{BlkDiag}(L_1(l_{11}(t)), \ldots, L_N(l_{NN}(t)))$, and $F(L(t)) \triangleq \Theta(L(t)) - (1_N \otimes E)$,

$$
\Theta(L(t)) \triangleq \begin{bmatrix}
A_{bk} - \tilde{L}_1(l_{11}(t))C_1 & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & A_{bk} - \tilde{L}_N(l_{NN}(t))C_N
\end{bmatrix},
$$

with $1_N$ denoting a column vector of $N$ ones.

Lemma 4 provides a sufficient condition under which the error is quadratically γ($\mathcal{L}(t)$)-bounded, where γ is a function of the specific configuration of the communication graph at time $t$.

Lemma 4: If $\exists \alpha_2 \geq 0$ such that (24) is satisfied, then the error in (23) is quadratically γ($\mathcal{L}(t)$)-bounded with symmetric positive-definite Lyapunov matrix $\tilde{P}$.

Proof: Applying Lemma 2 to the system in (23) implies that if (24) (shown at the bottom of the page) is satisfied, then the error in (23) is quadratically γ($\mathcal{L}(t)$)-bounded.

As shown in [21], in the case where all the agents are connected to the network, we can decompose the error $e(t)$ into its average $\bar{e}(t) \triangleq \frac{1}{N} \sum_{i=1}^{N} e_i(t)$ and the rest $\bar{e}(t) \triangleq (S^T \otimes I_n)e(t)$ so that

$$
e(t) = \begin{bmatrix}
\mathcal{I} & \mathcal{S} & \otimes & I_n & \mathcal{E}  \\
\end{bmatrix}
\begin{bmatrix}
\bar{e}(t) \\
\bar{e}(t)
\end{bmatrix}
$$

(25)

where $S$ is defined in (1) and (2). When all the agents connect to the network, the error dynamics for the overall system are given

$$
\begin{bmatrix}
(\gamma(\mathcal{L}(t)) - 2\alpha_2)\tilde{P} - A_{e}(\mathcal{L}(t))^T \tilde{P} - \tilde{P} A_{e}(\mathcal{L}(t)) & -\tilde{P} I & \tilde{P} J(\mathcal{L}(t)) \\
J(\mathcal{L}(t))^T \tilde{P} & \alpha_2 Q & 0 \\
0 & 0 & \alpha_2 R
\end{bmatrix} > 0
$$

(24)
Applying Lemma 2, if the network connection protocol ensures that
\[ \tilde{e}(t) = \begin{bmatrix} A - LC - \hat{H}(S \otimes I_n) S \otimes I_n \end{bmatrix}_n \tilde{e}(t) + \begin{bmatrix} I_n \end{bmatrix}_n w(t) - \begin{bmatrix} L \end{bmatrix}_n \tilde{v}(t) \] 

where \( \tilde{J} \equiv N \text{BlkDiag}(L_1, \ldots, L_N) \)

\[ H \equiv \begin{bmatrix} \frac{1}{N} A_{bb} - B_1 K_1 - L_1 C_1 & \cdots & \frac{1}{N} A_{bb} - B_N K_N - L_N C_N \end{bmatrix} \]

\[ F \equiv \begin{bmatrix} A_{bb} - N L_1 C_1 - B_1 K_1 & \cdots & -B_N K_N \end{bmatrix} \]

Since \( \Lambda^+ \) is positive definite, the coupling gain \( \eta \) can be designed with a sufficiently large value to ensure that the lower right block of \( \Lambda_n \) is negative definite. This ensures that the diagonal blocks of \( \Lambda_n \) only contain eigenvalues that lie in the left-half plane since \( L \) has been designed to make \( A - LC \) Hurwitz.

Lemma 5 provides a sufficient condition under which the error is quadratically \( \gamma(\tilde{L}) \)-bounded when all the agents connect to the network.

**Lemma 5:** If \( \exists \alpha_3 \geq 0 \) such that (27) is satisfied, then the error in (23) is quadratically \( \gamma(\tilde{L}) \)-bounded with symmetric positive-definite Lyapunov matrix \( \tilde{P} \) when all the agents are connected to the network.

**Proof:** Applying Lemma 2 to the system in (26) implies that if (27) (shown at the bottom of the page) is satisfied, then the error \( \tilde{e}(t) \) in (26) is quadratically \( \gamma(\tilde{L}) \)-bounded with symmetric positive-definite Lyapunov matrix \( \tilde{P} \) \( \tilde{P} \). Since \( \tilde{e}(t)^T \tilde{P} \tilde{e}(t) = (e(t))^T P e(t) \), this is equivalent to the error in (23) being quadratically \( \gamma(\tilde{L}) \)-bounded with symmetric positive-definite Lyapunov matrix \( \tilde{P} \) when all the agents are connected to the network.

**B. Stability Conditions**

We want to ensure that \( x(t) \) converges to \( E_x \) by creating a network connection protocol, which guarantees that the Lyapunov function \( V(x(t)) \equiv x(t)^T P x(t) \) is greater than or equal to 1, it decreases over time and converges to the robust positive invariant set \( E_x \). The invariance of \( E_x \) is shown in Lemma 1 to be equivalent to quadratic 0-boundedness. Consequently, the network connection protocol should guarantee that when \( V(x(t)) \geq 1, \tilde{V}(x(t)) < 0 \forall t \). The following theorem, motivated by Donkers [10] and Heemels et al. [11], sets forth sufficient conditions under which \( \tilde{V}(x(t)) < 0 \forall t \).

\[
\begin{bmatrix}
(\gamma(\tilde{L}) - 2\alpha_3) \tilde{P} \tilde{P} - \tilde{A}_p^T \tilde{P} \tilde{P} - \tilde{E}_p^T \tilde{P} \tilde{P} \tilde{A}_e - \tilde{E}_p^T \tilde{P} \tilde{E} \tilde{W} - \tilde{W}^T \tilde{P} \tilde{P} \\
- \tilde{W}^T \tilde{P} \tilde{P} \\
\tilde{V}_p^T \tilde{P} \tilde{P} \\
\end{bmatrix} > 0
\]

### Theorem 1
If the network connection protocol ensures that for some \( i \in \{1, \ldots, N\} \)

\[ -y_i(t)^T Y_i y_i(t) + e(t)^T \tilde{P} e(t) + w(t)^T Q w(t) + v(t)^T R v(t) < 0 \]

where \( Y_i > 0 \), and if \( \forall i \in \{1, \ldots, N\} \)

\[
\begin{bmatrix}
-A_{bb}^T P - PA_{bb} - C_i^T Y_i C_i \\
E^T P - P E \\
E^T P - P E \\
-\Gamma_i Y_i C_i \\
-\Gamma_i Y_i C_i \\
R - \Gamma_i^T Y_i \Gamma_i \\
\end{bmatrix} \geq 0
\]

where \( \Gamma_i \equiv \begin{bmatrix} 0_{m_i \times \sum_{j=1}^{i-1} m_j} & I_m \end{bmatrix} \), then

\[ \tilde{V}(x(t)) < 0 \forall t \] for the system in (11).

**Proof:** The condition in (29) is equivalent to

\[
w(t)^T Q w(t) - (A_{bb} x(t) - E e(t) + w(t))^T P x(t)
+ v(t)^T R v(t) - x(t)^T P (A_{bb} x(t) - E e(t) + w(t))
+ e(t)^T P e(t) - y_i(t)^T Y_i y_i(t) \geq 0
\]

which is equivalent to

\[
\tilde{V}(x(t)) \leq -y_i(t)^T Y_i y_i(t) + e(t)^T \tilde{P} e(t)
+ w(t)^T Q w(t) + v(t)^T R v(t).
\]

Taking (31) in conjunction with the condition in (28) ensures that \( \tilde{V}(x(t)) < 0 \forall t \) for the system in (11) when (29) is satisfied \( \forall i \in \{1, \ldots, N\} \) and (28) is satisfied for some \( i \in \{1, \ldots, N\} \).

Note that \( P \) determines the size of the invariant set \( E_x \) to which the state converges, \( \tilde{P} \) determines the size of the invariant set \( E_e \) to which the error 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 (33). Maximizing \( \log \det P \) is proportional to minimizing the volume of \( E_x \), compressing the size of the invariant set to which the state converges. Maximizing \( \log \det \tilde{P} \) is proportional to minimizing the volume of \( E_e \), compressing the size of the invariant set to which the error converges. Maximizing \( \log \det Y_i \) is proportional to maximizing \( y_i(t)^T Y_i y_i(t) \forall y_i(t) \), which minimizes the number of times agent \( i \) connects to the network, as will be seen in (33). Consequently, the desired values for \( P, \tilde{P}, \) and \( Y_i \) are obtained according to the following optimization problem:

\[
\arg \max_{\alpha_1, \alpha_3, \gamma(\tilde{L}), \bar{P}, P, Y_i, \ldots, Y_N} \omega_x \log \det \bar{P} + \omega_e \log \det \tilde{P} + \sum_{i=1}^{N} \omega_i \log \det Y_i
\]

s.t. \( \gamma(\tilde{L}) \leq 0, \alpha_1, \alpha_3 \geq 0, \bar{P} > 0, \tilde{P} > 0 \)

\[ Y_i > 0 \forall i \in \{1, \ldots, N\} \]

(20), (27), and (29) are satisfied

\[
(\gamma(\tilde{L}) - 2\alpha_3) \tilde{E}_p^T \tilde{P} \tilde{E} - \tilde{A}_p^T \tilde{P} \tilde{P} - \tilde{E}_p^T \tilde{P} \tilde{E} \tilde{A}_e - \tilde{E}_p^T \tilde{P} \tilde{W} - \tilde{W}^T \tilde{P} \tilde{P} \tilde{E} \tilde{W} - \tilde{V}^T \tilde{P} \tilde{E} \tilde{V} > 0
\]

(27)
where $\omega_x, \omega_z, \text{ and } \omega_i, i \in \{1, \ldots, N\}$, are nonnegative constants chosen by the designer to weight the importance of minimizing $E_x, E_{zt}$, and the communication frequency of agent $i$, respectively. The constraint $\gamma(\bar{L}) \leq 0$ is included in this optimization problem because we would like the error to decrease when all the agents are connected to the network, as will be seen in Theorem 2. Because this optimization problem is not convex, a suboptimal solution may be obtained by restricting the possible values of $\alpha_1$ and $\alpha_3$ to a finite set and carrying out the optimization problem in (32) over that finite set, since (32) is convex for set values of $\alpha_1$ and $\alpha_3$.

\section{C. Triggering Conditions}

The following theorem leverages the results of Lemmas 3–5 and Theorem 1, providing a set of network connection triggering conditions for each agent based on (28) to ensure that $x(t)$ converges to $E_x$.

\textbf{Theorem 2:} Let $\{\bar{t}_k\}, k \in \mathbb{Z}^+$, represent the sequence of time instants, where the configuration of the communication graph changes due to agents connecting or disconnecting from the network. We let $\bar{t}_0 \triangleq 0$. If $\exists \alpha_1, \alpha_3 \geq 0$ such that (20) and (27) are satisfied with $\gamma(\bar{L}) \leq 0$, then (29) is satisfied $\forall i \in \{1, \ldots, N\}$, and if agent $i$ connects to the network when $y_i(t)\bar{Y}_i(t) \leq 2 + 2 \left(1, e^{\gamma(\bar{t}_k)(t-\bar{t}_k)+\sum_{j=0}^{k-1} \gamma(\bar{t}_j)(\bar{t}_{j+1}-\bar{t}_j)} \right)$ for all $t \in [\bar{t}_k, \bar{t}_{k+1}) \forall k \in \mathbb{Z}^+$, where

$$\gamma(\bar{t}_k) \triangleq \min_{\alpha_2, \gamma(\bar{t}_k)} \gamma(\bar{L}(t)) \quad \text{s.t. } \alpha_2 \geq 0,$$

then $x(t)$ will converge to $E_x$ as long as agent $i$ stays connected to the network while

\begin{align}
\int_0^t \left(2 + \max \left(1, e^{\gamma(\bar{t}_k)(t-\bar{t}_k)+\sum_{j=0}^{k-1} \gamma(\bar{t}_j)(\bar{t}_{j+1}-\bar{t}_j)} \right) - y_i(\varphi)^T \bar{Y}_i(t) y_i(\varphi) \right) d\varphi > f(t) \quad (35)
\end{align}

for all $t \in [\bar{t}_k, \bar{t}_{k+1}) \forall k \in \mathbb{Z}^+$ when agent $i$ and all its neighbors are connected to the network, where $f(t) : \mathbb{R} \rightarrow \mathbb{R}$ is any strictly decreasing function such that $f(t) \leq 0 \forall t \geq 0$.

\textbf{Proof:} By applying Lemma 2 and Definition 1 to the system in (23), the definition of $\gamma(\bar{t}_k)$ in (34) implies that $\forall w(t) \in W$ and $\forall \bar{t}_k, \bar{t}_{k+1} \in \mathbb{Z}^+$

$$V_e(e(t)) \geq 1 \Rightarrow \dot{V}_e(e(t)) < \gamma(\bar{t}_k) V_e(e(t)) \quad \forall t \in [\bar{t}_k, \bar{t}_{k+1})$$

where $V_e(e(t)) \triangleq e(t)^T \bar{P} e(t)$. This is equivalent to

$$V_e(e(t)) \geq 1 \forall t \in [\bar{t}_k, \bar{t}_{k+1})$$

$$\Rightarrow V_e(e(\bar{t}_{k+1})) < e^{\gamma(\bar{t}_k)(\bar{t}_{k+1}-\bar{t}_k)} V_e(e(\bar{t}_k)). \quad (37)$$

Consequently, $\forall t \in [\bar{t}_k, \bar{t}_{k+1})$

$$V_e(e(t)) \leq \max \left(1, e^{\gamma(\bar{t}_k)(t-\bar{t}_k)+\sum_{j=0}^{k-1} \gamma(\bar{t}_j)(\bar{t}_{j+1}-\bar{t}_j)} V_e(e(0)) \right). \quad (38)$$

If $e(0) \in E_x$, then $\max_{e(0)} V_e(e(0)) = 1$, implying that $\forall t \in [\bar{t}_k, \bar{t}_{k+1})$

$$V_e(e(t)) \leq \max \left(1, e^{\gamma(\bar{t}_k)(t-\bar{t}_k)+\sum_{j=0}^{k-1} \gamma(\bar{t}_j)(\bar{t}_{j+1}-\bar{t}_j)} \right). \quad (39)$$

The trigger condition in (33) can be written equivalently as

$$2 - y_i(t)^T \bar{Y}_i(t) y_i(t) + \max \left(1, e^{\gamma(\bar{t}_k)(t-\bar{t}_k)+\sum_{j=0}^{k-1} \gamma(\bar{t}_j)(\bar{t}_{j+1}-\bar{t}_j)} \right) \geq 0. \quad (40)$$

Note that $\forall w(t) \in W$ and $\forall \bar{t}_k \in V$

$$2 - y_i(t)^T \bar{Y}_i(t) y_i(t) + \max \left(1, e^{\gamma(\bar{t}_k)(t-\bar{t}_k)+\sum_{j=0}^{k-1} \gamma(\bar{t}_j)(\bar{t}_{j+1}-\bar{t}_j)} \right) \geq 0 \quad (41)$$

for all $t \in [\bar{t}_k, \bar{t}_{k+1}) \forall k \in \mathbb{Z}^+$ according to (39). Consequently, (33) functions as an upper bound on the condition in (28) so that whenever (28) is not satisfied, the condition in (33) will be triggered.

If $\bar{V}(t)$ is satisfied $\forall i \in \{1, \ldots, N\}$, Theorem 1 states that $\bar{V}(t) < 0 \forall t$ when (33) is not satisfied for at least one agent, implying that $x(t)$ converges to $E_x$ when (33) is not triggered for some $i \in \{1, \ldots, N\}$. In the case where (33) is triggered $\forall i \in \{1, \ldots, N\}$, all the agents will be connected to the network, and either $e(t) \in E_x$ or $e(t) \notin E_x$. If $e(t) \in E_x$, then Lemma 3 states that if $\exists \alpha_1 \geq 0$ such that (20) is satisfied, then $x(t)$ will converge to $E_x$. If $e(t) \notin E_x$, Lemma 5 states that if $\exists \alpha_3 \geq 0$ such that (27) is satisfied with $\gamma(\bar{L}) \leq 0$, then $e(t)$ will converge to $E_x$. Once $e(t) \in E_x$, we know from Lemma 3 that $x(t)$ will converge to $E_x$.

If agent $i$ stays connected to the network when (35) is satisfied and all of agent $i$’s neighbors are connected to the network, then $\bar{V}(x(t))$ will not increase enough to destabilize the system when $e(t)$ converges to $E_x$. The integrand in (35) functions as an upper bound on the left-hand side of the inequality in (28), which in turn is an upper bound on $\bar{V}(x(t))$. Consequently, the condition in (35) ensures that when all the agents connect to the network, they will stay connected until $\bar{V}(x(t)) - \bar{V}(0) < f(t)$, guaranteeing that $x(t)$ eventually converges to $E_x$, even when $e(t) \notin E_x$. $\blacksquare$

Theorem 2 ensures that when the magnitude of the error $V_e(e(t))$ grows too large, (33) will be triggered, and agent $i$ will connect to the network to communicate with other agents. Note that by minimizing $\gamma(\bar{L}(t))$ in (34), $\gamma(\bar{t}_k)$ is defined as the smallest upper bound on the rate at which the magnitude of the error increases during the period $[\bar{t}_k, \bar{t}_{k+1})$. In this way, the maximization term in (33) is the smallest upper bound on the magnitude of the error $V_e(e(t))$, as seen in (39). Also note that in (33) and (35), $y_i(t)$ is locally available to agent $i$, but $\gamma(\bar{t}_j)$ contains some information that is not locally available to agent $i$ since $\gamma(\bar{t}_j)$ is a function of the configuration of the overall communication graph $\bar{G}(t) \forall t \in [\bar{t}_j, \bar{t}_{j+1})$. In the next
section, we describe how to compute $\gamma(\bar{t}_j)$ without possessing complete knowledge of the history of communication graph configurations.

D. Evaluating the Triggering Conditions

Since each agent is unable to compute $\gamma(\bar{t}_j) \forall j \in \{0, \ldots, k\}$ when evaluating (33) and (35), each agent assumes a history of communication graph configurations that results in the worst case value of $\gamma(\bar{t}_j)$. As each agent acquires more information about the history of communication graph configurations, it updates its worst case value of $\gamma(\bar{t}_j)$ and evaluates (33) and (35) accordingly. This process can be described more formally as follows.

Let $\bar{L}_i(t)$ and $\hat{L}_i(t)$ represent the portions of $\mathcal{L}(t)$ whose values are known and unknown by agent $i$, respectively, so that $\mathcal{L}(t) = \bar{L}_i(t) + \hat{L}_i(t) \forall i \in \{1, \ldots, N\}$. For agent $i$ to evaluate (33) and (35), it first solves for $\gamma(\bar{t}_j)$ offline in (34) for all the possible configurations of the communication graph $\bar{G}$. This produces a lookup table where each possible value of the Laplacian $\bar{L}(t)$ is associated with a particular value of $\gamma(\bar{t}_j)$. The length of this lookup table is not dependent on $t$ or $\bar{t}_j$ since there are only a finite number of possible communication graph configurations that are consistent with the underlying and unchanging communication graph $\bar{G}$. Agent $i$ then evaluates (33) and (35) online using $\bar{\gamma}_i(\bar{t}_j)$ instead of $\gamma(\bar{t}_j)$, where $\bar{\gamma}_i(\bar{t}_j)$ is given by

$$\bar{\gamma}_i(\bar{t}_j) = \max_{\bar{L}(t)} \gamma(\bar{t}_j) \text{ s.t. } \mathcal{L}(t) = \bar{L}_i(t) + \hat{L}_i(t) \forall t \in [\bar{t}_j, \bar{t}_{j+1}).$$

In this way, agent $i$ always evaluates the trigger condition with values that result in the right-hand side of the inequality in (33) being greater than or equal to its actual value. This then functions as an upper bound on the actual value of the condition in (33), which is itself an upper bound on the condition in (28), implying that whenever (28) is not satisfied, the condition in (33) evaluated with $\bar{\gamma}_i(\bar{t}_j)$ will be triggered. Similarly, agent $i$ always evaluates (35) with values that result in the left-hand side of the inequality in (35) being greater than or equal to its actual value, ensuring that whenever (35) is satisfied, the condition in (35) evaluated with $\bar{\gamma}_i(\bar{t}_j)$ will also be satisfied.

Remark 2: Note that computing $\bar{\gamma}_i(\bar{t}_k)$ in (34) for all the possible values of $\mathcal{L}(t)$ requires evaluating $2^N - N$ LMIs since this is the number of possible configurations of the communication graph $\bar{G}(t)$ for different sets of agents connected and disconnected from the network. However, all of this computation is completed offline ahead of time and grows with the number of agents $N$, the number of control inputs $p$ or sensor measurements $m$. Furthermore, (34) does not need to be evaluated for each $\bar{t}_k$ since the set of possible values for $\mathcal{L}(t)$ is dependent on the configuration of the underlying communication graph $\bar{G}$ and is, therefore, time invariant.

Theorem 3 addresses the case where the offline calculation in (34) becomes computationally intractable with large $N$. It states that $\gamma(\bar{t}_k)$ will always be the largest when all the agents are disconnected from the network, or in other words, $V_c(e(t))$ will grow the fastest when all the agents are disconnected from the network. Consequently, each agent can always use this worst case value for $\gamma(\bar{t}_k)$ when evaluating (33) and (35), and therefore, only two LMIs need to be evaluated in (34) (the cases where all the agents are either connected or disconnected from the network) instead of $2^N - N$ LMIs, which are needed in [20].

**Theorem 3:** Let $\hat{V}_c(e(t))$ represent the value of $V_c(e(t))$ when all the agents are disconnected from the network. Then

$$\hat{V}_c(e(t)) \leq \bar{V}_c(e(t))$$

implying that $V_c(e(t))$ increases the most when all the agents are disconnected from the network.

**Proof:** Given the error dynamics for the overall system in (23)

$$\dot{V}_c(e(t)) = -\eta e(t)^T \left( (\mathcal{L}(t) \otimes I_n) \bar{P} + \bar{P} (\mathcal{L}(t) \otimes I_n) \right) e(t) + 2e(t)^T \bar{P} \left( F(\mathcal{L}(t))e(t) + \bar{w}(t) - J(\mathcal{L}(t))w(t) \right)$$

$$\leq 2e(t)^T \bar{P} \left( F(\mathcal{L}(t))e(t) + \bar{w}(t) - J(\mathcal{L}(t))w(t) \right) = \hat{V}_c(e(t))$$

where the inequality follows from the fact that $\eta \geq 0$ and that $(\mathcal{L}(t) \otimes I_n)^T \bar{P} + \bar{P} (\mathcal{L}(t) \otimes I_n)$ is positive semidefinite since the eigenvalues of the Laplacian $\mathcal{L}(t)$ are always nonnegative.

E. Network Connection Procedure

Algorithm 1 describes a procedure that ensures that $x(t)$ converges to $E_x$ as guaranteed by Theorems 1 and 2 when (20), (27), and (29) are satisfied. The event-triggered communication procedure presented in Algorithm 1 uses only local information to indicate when agent $i$ needs to connect to the network and communicate with other agents. In line 3, agent $i$ uses its local sensor measurements $y_i(t)$ as well as $\bar{\gamma}_i(\bar{t}_k) \forall \bar{t}_k \leq t$ to evaluate the trigger condition in (33). If (33) is satisfied, agent $i$ connects to the network (line 4) and broadcasts $\tilde{x}_i(t)$ to its neighboring agents (line 5). Agent $i$ also broadcasts $\tau_j$ to its neighboring agents, where $\tau_j$ represents the most recent time instant where agent $i$ possesses full information about changes in the configuration of the communication graph. After receiving $\tau_j$ from each neighboring agent $j$ currently connected to the network (line 6), agent $i$ sends the information it possesses about communication graph configuration changes from time $\tau_j$ to the current time to each of these agents (line 8). Agent $i$ then uses the information it receives from these agents to update its information about communication graph configuration changes (line 9) as well as to update $\tau_i$ (line 11). If all the agent $i$’s neighbors are connected to the network, agent $i$ uses its local sensor measurements $y_i(t)$ as well as $\bar{\gamma}_i(\bar{t}_k) \forall \bar{t}_k \leq t$ to evaluate the condition in (35). If (35) is satisfied, then agent $i$ stays connected to the network and continues to communicate with its neighboring agents. Otherwise, agent $i$ disconnects from the network (line 13).

If (33) is not satisfied, then agent $i$ updates its information about communication graph configuration changes with the fact that it is not connected to the network at time $t$ (line 16). Any state estimates agent $i$ receives from its neighboring agents are
Algorithm 1: Network Connection Procedure for Agent $i$.

1: Initialize $\bar{x}_i(0)$, $\forall i \in \{1, \ldots, N\}$ so that $e(0) \in \mathcal{E}_e$
2: while 1
3: if (33) is satisfied
4: Open network connection
5: Broadcast $\bar{x}_i(t)$ and $\tau_i$ to agents $j \in \mathcal{N}_i$
6: Receive $\tau_j$ from agents $j \in \mathcal{N}_i$ connected to the network
7: parfor $j \in \mathcal{N}_i$ connected to the network
8: Send $\{\bar{L}_i(\bar{t}_k), \bar{t}_k\}$ $\forall \bar{t}_k \in (\tau_j, t]$ to agent $j$
9: Update $\bar{L}_i(\bar{t}_k)$ with information received from $\bar{L}_j(\bar{t}_k)$ $\forall \bar{t}_k \in (\tau_j, t]$.
10: end parfor
11: Update $\tau_i$
12: if Some agent $j \in \mathcal{N}_i$ is not connected to the network or (35) is not satisfied
13: Close network connection
14: end if
15: else
16: Update $\bar{L}_i(t)$ with $a_{ij}(t) = a_{ji}(t) = 0$ $\forall j \in \{1, \ldots, N\}$
17: end if
18: $\hat{x}_i(t) = A_{bb} \hat{x}_i(t) + \bar{L}_i(\bar{t}_i(t))(y_i(t) - C_i \hat{x}_i(t)) + \eta \sum_{j=1}^{N} a_{ij}(t)(\bar{x}_j(t) - \hat{x}_i(t))$
19: end while

used to compute its state estimate $\bar{x}_i(t)$ according to (10) (line 18). Note that because (33) functions as an upper bound on the condition in (28), agents will connect to the network more often than is necessary.

Remark 3: Note that Algorithm 1 presents a procedure where an agent’s sending and receiving capabilities are simultaneously triggered by the condition in (33). However, an attack on an agent is initiated through data that are incoming to that agent, not outgoing from that agent. Consequently, data could constantly be broadcast to agents all the time, while (33) would only be used for deciding when to receive information from other agents. In this case, agent $i$ would continuously broadcast $\bar{x}_i(t)$, $\tau_i$, and $\{\bar{L}_i(\bar{t}_k), \bar{t}_k\}$ $\forall \bar{t}_k \in (\min_{j \in \mathcal{N}_i} \tau_j, t]$ to all its neighbors. By doing so, an agent receiving information would possess the state estimates from all its neighbors, reducing that agent’s state estimation error compared to the current scenario where only a subset of the neighbors’ state estimates may be received. This in turn would decrease the number of times (33) is triggered since (33) 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 the agents would always be broadcasting information. The implementation of this approach, along with an investigation of the tradeoff between overall performance and communication costs, is left for future work.

Remark 4: For the discrete time case, (20), (24), (27), (29), (33), and (35) would be replaced by

$$\begin{bmatrix} (1 - 2\alpha_1)P - A_{bb}^T A_{bb} & A_{bb}^T P \bar{E} & -A_{bb}^T P \\ E^T A_{bb} & \alpha_1 P - E^T P \bar{E} & \alpha_1 Q - P \end{bmatrix} > 0$$

(46), (47), (48) (shown at the bottom of the page),

$$\sum_{i=0}^{k} 2 + \max\left(1, \prod_{j=0}^{\ell-1} \gamma(\mathcal{L}_j)\right) y_k^T \bar{Y}_i y_k^i \leq 2 + \max\left(1, \prod_{j=0}^{\ell-1} \gamma(\mathcal{L}_j)\right) - y_k^T \bar{Y}_i y_k^i > f(k + 1)$$

(50)

respectively, where $\mathcal{L}_k$ represents the Laplacian at time step $k$, and $y_k^i$ denotes agent $i$’s sensor measurements at time step $k$. However, (47) may not be satisfied with $\gamma(\mathcal{L}) \in [0, 1]$ if it is not possible to design the coupling gain $\eta$ so that the eigenvalues of the lower right block of $A_e$ lie within the unit circle.

F. Ensuring Resiliency Against Attacks

The network connection procedure presented in Algorithm 1 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 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 1 is beyond the scope of this article 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 [15]. 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 of the three tanks, the control inputs are the open levels of the valves, and the sensors measure the current water levels of the tanks. The global system model is linearized at a reference level of 3 m, as presented in [15], and is given by

\[
\dot{x}(t) = \begin{bmatrix} -8.367 & 0 & 0 \\ 0 & -6.276 & 0 \\ 0 & 0 & -5.020 \end{bmatrix} x(t) \times 10^{-4} + \begin{bmatrix} 0.1068 & -0.0371 & -0.0371 \\ -0.0279 & 0.0801 & -0.0279 \\ -0.0223 & -0.0223 & 0.0641 \end{bmatrix} u(t) + w(t)
\]

\[
y(t) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} x(t) + v(t).
\]

We let \( Q = \frac{10000}{3} I_n \) and \( R = \frac{10000}{3} I_m \) so that the process and measurement disturbances are less than 1 cm for each tank.

A continuous-time controller \( K \) with three poles at \(-1.5\) is designed to stabilize the system when all the agents are connected to the network, and continuous-time observers \( L \) and \( \hat{L} \) \( \forall i \in \{1, \ldots, N\} \) are designed according to (8) so that the estimation error for the observable states is stabilized. \( L \) is designed with three poles at \(-100\), \( L'_i \) is designed with a pole at \(-15 \forall i \in \{1, \ldots, N\} \), and the coupling gain is chosen as \( \eta = 100000 \). The water distribution system is comprised of

\[
N = 3 \text{ agents, where each agent has access to one local control input and one local sensor measurement, and the adjacency matrix for the underlying communication graph is given by}
\]

\[
\bar{A} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.
\]

We solve for \( P, \bar{P} \), and \( Y_i \forall i \in \{1, \ldots, N\} \) so that (20), (27), and (29) are satisfied with \( \gamma(\bar{L}) \leq 0 \). The network connection protocol in Algorithm 1 is executed for each agent from the initial state \( x(0) = [10 \ 10 \ 10]^T \). To model changes in the set point as well as impulsive disturbances, we periodically update the state so that it jumps to a new value, simulating the behavior of a real-world system.

Figs. 1 and 2 depict the Lyapunov function convergence and the network connection time line, respectively, for a particular
simulation. As seen in Fig. 1, the Lyapunov function continually decreases between set points, which each causes a jump in the value of the Lyapunov function. Fig. 2 depicts the detailed connection time line for each agent, showing that agents are able to disconnect from the network approximately 49% of the time. This provides less time for adversaries to attack different agents while also ensuring the stability of the overall system when there is no attack.

Note that the network connection times may vary for each agent since each agent has different sets of local sensors, since the dynamics of some agents may be more tightly coupled to one another than the dynamics of other agents, and since some agents may have more neighbors than those of other agents. These factors cause some agents to connect to the network more than others to ensure the stability of the overall system. Since agent 2 is the only agent that has two neighbors, it receives more information each time it connects to the network, and consequently, it connects to the network far less than agents 1 and 3, as seen in Fig. 2.

In addition, no performance is lost in using the decentralized event-triggered network connection protocol. Over 1000 trials with no set point updates, the time taken to converge to the invariant set $E_c$ remains the same regardless of whether communication between agents occurs all the time (average convergence time of 2.6728 s) or whether agents disconnect from the network for intermittent periods of time (average convergence time of 2.6749 s). This average convergence time is significantly less than the average convergence time of 19.982 s in [20] due to a more efficient state estimation scheme.

V. Conclusion

This article investigated using decentralized event-triggered control to reduce attack opportunities. An event-triggered mechanism for network connection and communication was designed based on only local information. This mechanism ensured the stability of the overall system for attack-free scenarios. It also allowed 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 was designed that used the event-triggered mechanism, and its effectiveness was 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 scenarios where nonnegligible communication delays exist when sending data over the network.

References

[1] A. A. Cardenas, S. Amin, and S. Sastry, “Secure control: Towards survivable cyber-physical systems,” in Proc. 28th Int. Conf. Distrib. Comput. Syst. Workshops, 2008, pp. 495–500.

[2] J. Slay and M. Miller, “Lessons learned from the Maroochy water breach,” in Proc. Int. Conf. Crit. Infrastruct. Protection, 2007, pp. 73–82.

[3] Defense Use Case, “Analysis of the cyber attack on the Ukrainian power grid,” Elect. Inf. Sharing Anal. Center, Washington, DC, USA, vol. 388, 2016, pp. 1–29.

[4] R. Langner, “Stuxnet: Dissecting a cyberwarfare weapon,” IEEE Secur. Privacy, vol. 9, no. 3, pp. 49–51, May/Jun. 2011.

[5] L. Szekeres, M. Payer, T. Wei, and D. Song, “SoK: Eternal war in memory,” in Proc. IEEE Symp. Secur. Privacy, 2013, pp. 48–62.

[6] R. Romagnoli, B. H. Krogh, and B. Sinopoli, “Design of software rejuvenation for CPS security using invariant sets,” in Proc. Amer. Control Conf., 2019, pp. 3740–3745.

[7] X.-S. Sun, P. Horiwitz, and C.-W. Tan, “An efficient lane change maneuver for platoons of vehicles in an automated highway system,” in Proc. ASME Int. Mech. Eng. Congr. Expo., 2003, pp. 355–362.

[8] J. Huang, Q. Huang, Y. Deng, and Y.-H. Chen, “Toward robust vehicle platooning with bounded spacing error,” IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 36, no. 4, pp. 562–572, Apr. 2017.

[9] W. P. Heemels, K. J. Johansson, and P. Tabuada, “An introduction to event-triggered and self-triggered control,” in Proc. IEEE 51st Conf. Decis. Control, 2012, pp. 3270–3285.

[10] M. Donkers, “Networked and event-triggered control systems,” Ph.D. dissertation, Dept. Mech. Eng., Tech. Univ. Eindhoven, Eindhoven, The Netherlands, 2011.

[11] W. H. Heemels, M. Donkers, and A. R. Teel, “Periodic event-triggered control for linear systems,” IEEE Trans. Autom. Control, vol. 58, no. 4, pp. 847–861, Apr. 2013.

[12] M. Donkers and W. Heemels, “Output-based event-triggered control with guaranteed $L_2$-gain and improved and decentralized event-triggering,” IEEE Trans. Automat. Control, vol. 57, no. 6, pp. 1362–1376, Jun. 2012.

[13] V. Dolk, D. P. Borgers, and W. Heemels, “Output-based and decentralized dynamic event-triggered control with guaranteed $L_p$-gain performance and Zeno-freeness,” IEEE Trans. Automat. Control, vol. 62, no. 1, pp. 34–49, Jan. 2017.

[14] W. Heemels and M. Donkers, “Model-based periodic event-triggered control for linear systems,” Automatica, vol. 49, no. 3, pp. 698–711, 2013.

[15] A. Fu and J. A. McCann, “Dynamic decentralized periodic event-triggered control for wireless cyber-physical systems,” IEEE Trans. Control Syst. Technol., vol. 29, no. 4, pp. 1783–1790, Jul. 2021.

[16] A. Fu and M. Mazo Jr., “Decentralized periodic event-triggered control with quantization and asynchronous communication,” Automatica, vol. 94, pp. 294–299, 2018.

[17] A. Zolich et al., “Survey on communication and networks for autonomous marine systems,” J. Intell. Robot. Syst., vol. 95, no. 3, pp. 789–813, 2019.

[18] H. Cui, G. Zhao, S. Liu, and Z. Li, “A decentralized dynamic self-triggered control approach to consensus of multiagent systems,” IEEE Trans. Syst., Man, Cybern. Syst., vol. 53, no. 9, pp. 5772–5783, Sep. 2023.

[19] P. Griffioen, R. Romagnoli, B. H. Krogh, and B. Sinopoli, “Decentralized event-triggered control in the presence of adversaries,” in Proc. IEEE 59th Conf. Decis. Control, 2020, pp. 3236–3242.

[20] P. Griffioen, R. Romagnoli, B. H. Krogh, and B. Sinopoli, “Reducing attack vulnerabilities through decentralized event-triggered control,” in Proc. IEEE 60th Conf. Decis. Control, 2021, pp. 5715–5722.

[21] T. Kim, D. Lee, and H. Shim, “Decentralized design and plug-and-play distributed control for linear multi-channel systems,” IEEE Trans. Autom. Control, early access, Jul. 6, 2023, doi: 10.1109/TAC.2023.3290306.

[22] M. L. Brockman and M. Corless, “Quadratic boundedness of nominal linear systems,” Int. J. Control, vol. 71, no. 6, pp. 1105–1117, 1998.

[23] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, Linear Matrix Inequalities in System and Control Theory. Philadelphia, PA, USA: SIAM, 1994.

[24] P. Griffioen, R. Romagnoli, B. H. Krogh, and B. Sinopoli, “Secure networked control via software rejuvenation,” in Proc. IEEE 58th Conf. Decis. Control, 2019, pp. 3878–3884.

[25] P. Griffioen, R. Romagnoli, B. H. Krogh, and B. Sinopoli, “Secure networked control for decentralized systems via software rejuvenation,” in Proc. Amer. Control Conf., 2020, pp. 1266–1273.
Paul Griffioen (Member, IEEE) received the B.S. degree in electrical and computer engineering from Calvin College, Grand Rapids, MI, USA, in 2016, and the M.S. and Ph.D. degrees in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2018 and 2022, respectively.

He is currently a Postdoctoral Researcher (working with Murat Arcak) with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, USA. His research interests include the modeling, analysis, and design of high-performance cyber-physical systems that ensure safety while operating under computational constraints. His research interests also include the modeling, analysis, and design of active detection techniques and response mechanisms for ensuring resilient and secure cyber-physical systems.

Raffaele Romagnoli (Member, IEEE) received the Ph.D. degree in control system and automation specializing in optimal and robust control system theory from the Università Politecnica delle Marche, Ancona, Italy, in 2015.

From 2015 to 2017, he was a Postdoctoral Researcher with the Department of Control Engineering and System Analysis, Université Libre de Bruxelles, Brussels, Belgium. He was a Postdoctoral Researcher with the Department of Electrical and Computer Engineering, Carnegie Mellon University (CMU), Pittsburgh, PA, USA, where he is currently a Research Scientist working on safe and secure control of AI robotics applications and edge computing. He is also with the Software Engineering Institute, CMU, where he works on secure control for cyber-physical systems. His research interests include nonlinear control, control of biological systems, energy storage (Li-ion batteries), space applications in microgravity conditions, and model inversion.

Bruce H. Krogh (Life Fellow, IEEE) received the B.S. degree in mathematics and physics from Wheaton College, Wheaton, IL, USA, in 1975, and the M.S. and Ph.D. degrees in electrical engineering from the University of Illinois Urbana-Champaign, Champaign, IL, USA, in 1979 and 1982, respectively.

From 1980 to 1981, he was a Systems Engineer with Westinghouse Corporation involved in the development of algorithms and control software for utility energy control centers. He was the Founding Director of Carnegie Mellon University-Africa, Kigali, Rwanda, where he lived from 2012 to 2017 and was a Member of the Technical Staff of Carnegie Mellon's Software Research Institute from 2018 to 2022. He is currently a Professor Emeritus of electrical and computer engineering with Carnegie Mellon University, Pittsburgh, PA, USA. His research focuses on the theory and application of control systems in a number of application areas, with a current focus on methods for guaranteeing safety and security of cyberphysical systems.

Dr. Krogh was the recipient of the Presidential Young Investigator Award from the U.S. National Science Foundation and Senior U.S. Scientist Award from the Alexander von Humboldt Foundation in Germany. From 2021 to 2023, he was Co-Lead of the IEEE Continuous Education initiative to develop IEEE’s continuing education resources for technical professionals in Africa. He currently chairs the board of the Kigali Collaborative Research Centre (KCRC) in Rwanda. He was the founding Editor-in-Chief of the IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY and is a Distinguished Member of the IEEE Control Systems Society.

Bruno Sinopoli (Fellow, IEEE) received the B.S. degree in electrical engineering from the University of Padova, Padua, Italy, in 1998, and the M.S. and Ph.D. degrees from the University of California at Berkeley, Berkeley, CA, USA, in 2003 and 2005, respectively, all in electrical engineering.

He is currently the Das Family Distinguished Professor with Washington University in St. Louis, St. Louis, MO, USA, where he is also the founding Director of the Center for Trustworthy AI in Cyber-Physical Systems and the Chair of the Department of Electrical and Systems Engineering. He was a Postdoctoral Researcher with Stanford University, Stanford, CA, USA. From 2007 to 2019, he was a Member of the faculty with Carnegie Mellon University, Pittsburgh, PA, USA, where he was a Professor with the Department of Electrical and Computer Engineering with courtesy appointments with the Department of Mechanical Engineering and with the Robotics Institute and the Co-director of the Smart Infrastructure Institute. His research interests include modeling, analysis, and design of resilient cyber-physical systems with applications to smart interdependent infrastructures systems, such as energy and transportation, Internet of Things, and control of computing systems.