Information Control in Networked Multi-User Discrete-Event Systems Using State Estimates

Fei Wang, *Member, IEEE*, Feng Lin, *Life Fellow, IEEE*, and Jun Chen, *Senior Member, IEEE*

***Abstract*—This paper investigates the problem of information control in networked multi-user systems, where agents such as robots, sensors, and software entities interact via a communication network to achieve individual or shared goals. Information control involves deciding which state estimates to share or broadcast, balancing cooperation among friends and privacy from adversaries. Since each user has only partial knowledge of the system, efficient protocols for sharing relevant data to balance privacy, security, and transparency is needed. This study models multi-user systems as discrete-event systems where agents need to distinguish certain state pairs in order to perform their tasks. We systematically study and solve critical problems to address the key aspects of information control: determining the necessity of shared information, minimizing communication for security, and maximizing public information release when required. A framework that addresses private communications, public broadcasting, and adversarial dynamics, offering strategies to meet both security and transparency requirements is introduced. Solutions and algorithms are proposed to solve these problems.**

***Note to Practitioners*—Engineering systems like connected vehicles, smart grids, and robotic teams face a critical challenge: controlling information in a decentralized environment where users have control over information flow. Traditional information theory, focused on reliability, does not address the strategic question of what information to release and when. This paper introduces a rigorous framework based on discrete-event systems and state estimates to solve five key information control problems. A major advantage over event-based methods is our approach's significantly lower computational complexity, enabling real-time execution. The approach also reduces the risk of cyberattacks by mitigating data poisoning attacks, preserving privacy, and limiting an attacker's ability to infer sensitive system states. Our findings provide engineers with a practical toolkit to enhance the security and resilience of distributed systems.**

***Index Terms*—Multi-user systems, information control, discrete-event systems, security, privacy, transparency.**

Received 16 September 2025; revised 18 December 2025 and 7 April 2026; accepted 29 May 2026. Date of publication 8 June 2026; date of current version 17 June 2026. This article was recommended for publication by Associate Editor Z. Ma and Editor C. Seatzu upon evaluation of the reviewers' comments. This work was supported in part by the National Science Foundation of USA under Grant 2432098 and Grant 2432099. *(Corresponding author: Fei Wang.)*

Fei Wang is with the School of Electrical and Electronic Engineering, Shanghai Institute of Technology, Shanghai 201418, China (e-mail: feiwang@sit.edu.cn).

Feng Lin is with the Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI 48202 USA (e-mail: flin@wayne.edu).

Jun Chen is with the Department of Electrical and Computer Engineering, Oakland University, Rochester, MI 48309 USA (e-mail: junchen@oakland.edu).

Digital Object Identifier 10.1109/TASE.2026.3701324

I. Introduction

NETWORKED multi-user systems consist of a collection of autonomous users or agents such as robots, sensors, drones, or software entities that interact and collaborate over a communication network to achieve individual or shared objectives. These systems are increasingly prevalent in applications like autonomous vehicle fleets, smart grids, distributed robotics, sensor networks, and collaborative decision-making systems.

In networked multi-user systems, information control, which manages how information is gathered, processed, shared, and utilized across the network to achieve desired system-level behavior, plays a crucial role in ensuring effective communication, coordination, and cooperation among users. The distributed and decentralized nature of multi-user systems introduces unique challenges for information control. Unlike centralized systems, where a single entity has access to all information, users in a multi-user system have only partial and local information of the system. This necessitates efficient protocols and algorithms that enable users to share relevant information optimally to ensure privacy, security, and/or transparency.

Intuitively, the information control problem explored in this paper can be summarized as follows. In a networked system with multiple users or agents, each has its own objectives (individual or shared). To achieve their objectives, users often require information from others. There are two methods for sharing information: (1) *communicating* it privately to specific users (for example, via encrypted messaging) and (2) *broadcasting* it publicly to all users. Each user has control authority over its own information and must decide whether or not to communicate or broadcast it. Due to security concerns, users may want to communicate as little information as possible. Conversely, transparency requirements may compel users to broadcast as much information as possible. Hence, the questions related to information control include the following. (1) What information should a user communicate privately to others? (2) What information shall a user broadcast? (3) How to minimize information communicated to ensure privacy and security in some applications? (4) How to maximize information broadcast to ensure transparency in other applications? This paper aims to develop a systematic approach to address these and other questions within the framework of networked multi-user systems.

Traditional information theory [1], [2], [3], [4], [5] primarily deals with challenges related to reliable and efficient communication, such as channel coding, data compression, and information encryption. In contrast, this paper addresses problems that assume reliable communication as a given. The focus of this paper is on the optimal control of information release, including the mechanisms by which information is shared and the concepts of minimum and maximum information disclosure. Traditional information theory does not directly address these critical aspects of information release and control, necessitating new approaches to tackle these problems. This is the first main difference of this paper from the existing literature and hence the innovations of this paper.

The second main difference of this paper from the existing literature is that we model multi-user systems as discrete-event systems (DES) using automata (also called finite-state machines) with discrete states and discrete events. To make the approach general, we model a user's task or goal as distinguishing certain pairs of states. For example, if a user wants to know whether the smart phone he/she ordered has been shipped or not, he/she needs to distinguish the states before the shipment from the states after the shipment.

Defining a task in terms of distinguishing certain pairs of states is a very general approach, and most common tasks can be specified in this manner [6], [7], [8]. For instance, in supervisory control, a supervisor must distinguish between legal and illegal states [9], [10], [11]. Similarly, in diagnosis, a diagnoser needs to distinguish normal states from faulty ones [12], [13], [14], [15].

If a user can distinguish the required pairs of states using only the information directly available to it, then it does not need any information from other users, making this a trivial case. Typically, however, a user needs information that is communicated or broadcast by others in order to perform its own task of distinguishing the required pairs of states. Additionally, a user can control the release of its own information by deciding what to share with others, either through direct communication or public broadcasting. The user's information control objectives may include one or more of the following: (1) assisting certain users in performing their tasks, (2) preventing other users from completing their tasks, (3) minimizing the amount of information communicated to others for security purposes, and (4) maximizing the amount of information broadcast to ensure transparency.

The information directly available to a user is the occurrences of some events local to the user. These events are called (locally) observable events of the user. Based on the observation of its observable events, a user can calculate the set of all possible states the system may be in. This set is called state estimate [6], [16]. The third main difference of this paper from the existing literature is that the information control of a user is to decide whether to communicate its state estimates to other users and/or to broadcast its state estimates to the public. Here the public refers to all users in the system. On the receiving end, if a user receives state estimates from other users, either from communication or from broadcasting, then it will have a better idea as which states the system may be in by taking the intersections of all available state estimates of other users with its own state estimate. Suppose that a user's goal is to distinguish a set of states Q_1 from another set of states Q_2. Then its goal can be achieved if and only if states in Q_1 are not mixed with states in Q_2 in the intersection. Clearly, the more state estimates it receives, the smaller is the intersection (meaning it is more certain about the actual state of the system), and the more likely its goal can be achieved.

The problem of information control becomes challenging in complex systems with many users who have different goals. We assume that users are divided into two or more groups: users within the same group are considered friends, while users in different group are viewed as adversaries, where the group divisions are specific to the problem being addressed. For simplicity, only two groups are considered, but the results presented can be extended to scenarios involving more than two groups.

Let's assume the initial information control objectives are: (1) to assist friends in achieving their goals and (2) to prevent adversarial users from reaching their goals. Accordingly, the initial control strategy is to share all information with friends, withhold all information from adversaries, and avoid broadcasting any information. If a user can achieve their goal under this initial control strategy, there is nothing their adversaries can do to prevent it from succeeding. Conversely, if a user's goal cannot be achieved under this initial control strategy, their friends cannot help them achieve it either. From this starting point, we will explore how to further refine the control based on additional requirements.

For reasons such as privacy and security, it is often necessary to minimize communication among users. The initial control can be refined by requiring minimum communication among friends without compromising the goals that these friends can achieve under the initial control strategy.

Intuitively, this involves removing some state estimates from being communicated, which reduces the amount of information available to some users. Consequently, this alters their intersections of state estimates. In general, this reduction in communication will cause the intersections to become larger (meaning the users are less certain about the actual state of the system). If removing a state estimate from being communicated causes a friend's intersection to grow to the extent that it includes both states in Q_1 and Q_2, then the user's goal can no longer be achieved, and the state estimate must be communicated. Otherwise, the state estimate can be removed from communication.

Minimum communication is achieved when no further state estimates can be removed. The specific order in which state estimates are examined may result in different minimum communication strategies. A minimum communication strategy is optimal in the sense that no other strategy can communicate strictly less.

For certain users, such as government agencies, there is a requirement to release (that is, broadcast) as much information as possible.[1] We also investigate the problem of maximizing information release in this paper. Starting again with the initial control strategy described earlier, we aim to refine it by requiring certain users to maximize the amount of information they broadcast while ensuring that their adversaries are not assisted in achieving any goals that were unattainable under

[1]For example, in the USA, the Freedom of Information Act (FOIA) mandates that certain information and records held by government agencies must be made available to the public upon request, except when the release could harm national security or falls under one of nine specific exemptions.

the initial control. Such a maximum broadcasting strategy is optimal in the sense that no other strategy can broadcast strictly more.

Therefore, we solve the following five information control problems in this paper: (1) Can a user perform its task based on its own local state estimates without state estimates from other users, including its friends? (2) Can a user perform its task based on its own local state estimates and all its friends' state estimates? (3) How can a user minimize communications to its friends without compromising the goals of its friends? (4) How can a user maximize its broadcasting without helping its adversaries to achieve their goals? (5) If users are required to broadcast some minimum information to the public, can we still solve the above four information control problems?

Minimizing communication in centralized and distributed discrete-event systems has been investigated in [17], [18], [19], and [20]. Maximize information release for a single user has been investigated in [21]. Compared with existing results in the literature, the contributions of this paper are as follows. (1) State-estimate-based DES framework: we develop a state-estimate-based DES approach for modeling information control in networked multi-user systems and propose a formal framework for systems where agents operate with limited local information. (2) Dual-focus information control: we address both privacy preservation (minimizing information exchange) and transparency requirements (maximizing public release) as well as identify and categorize five critical information control challenges in multi-agent systems. (3) Objective-based information control: we propose strategies for selective information release based on specific objectives and enable agents to control state-estimate-based information flows to either assist friends in achieving their objectives or prevent adversaries from reaching their objectives. (4) Beyond traditional information theory: we introduce state-estimate-based optimal control of information flows tailored to agent's objectives and emphasize objective-driven information control rather than efficient transmission. (5) Protocol design for multi-agent systems: we develop detailed protocols for balancing security and transparency concerns and provided frameworks for agents to navigate complex tasks involving both private communications and public broadcasting.

This paper is organized as follows. In Section II, we introduce our model of networked multi-user systems, which is a discrete-event system built from its components. In Section III, two mechanisms of information exchange among different users are proposed: private communication and public broadcasting. The information to be exchanged is state estimates. In Section IV, five information control problems are introduced. The task of a user is specified as a set of state pairs that the user needs to distinguish. Solutions and algorithms are proposed to solve the five problems. In Section V, information control methods on communicating events and state estimates are compared. In Section VI, an illustrative example of a distribution system is given to illustrate the results of the paper.

II. Networked Multi-User Systems

Deterministic finite automata, also known as deterministic finite-state machines [9], [10], [16], is used to model the networked multi-user systems under study:

$$G = (Q, \Sigma, \delta, q_0),$$

where Q is the set of finite states; Σ is the set of finite events; q_0 is the initial state; and $\delta : Q \times \Sigma \to Q$ is the transition function which describes the dynamics of the system. With a slight abuse of notation, the set of all possible transitions is also denoted by δ, that is, $\delta = \{(q, \sigma, q') \in Q \times \Sigma \times Q : \delta(q, \sigma) = q'\}$. The transition function is extended to $\delta : Q \times \Sigma^* \to Q$ in the usual way [16], where Σ^* denotes the set of all strings over Σ, including the empty string ε.

Networked systems is modeled as discrete-event systems for several reasons. First, this model is both general and flexible, making it suitable for representing most networked systems at some level of abstraction. Second, it enables the construction of a networked system model in a modular fashion, where individual components are modeled as small automata and then integrated using parallel composition, a process that can be automated by computers [16]. Third, it effectively captures system properties and information flows.

A trajectory s of G is a string that starts at q_0 and is defined by δ. We use $\delta(q_0, s)!$ to mean that $\delta(q_0, s)$ is defined. The set of all possible trajectories describing the behavior of G is called the language generated by G:

$$L(G) = \{s : s \in \Sigma^* : \delta(q_0, s)!\}.$$

Flexibility and scalability are crucial when modeling networked systems, as the components within these systems often change frequently. One benefit of using an automaton G is that the model can be constructed modularly: Assume that the system consists of M components. Each component of a networked system can be represented by a small automaton

$$G_i = (Q_i, \Sigma_i, \delta_i, q_{i,0}), \quad i = 1, 2, \cdots, M.$$

The overall system model can then be derived by applying parallel composition [16]:

$$\begin{aligned} G &= (Q, \Sigma, \delta, q_0) \\ &= G_1 || G_2 || \cdots || G_M \\ &= (Q_1 \times \cdots \times Q_M, \Sigma_1 \cup \cdots \cup \Sigma_M, \\ &\quad \delta_1 \times \cdots \times \delta_M, (q_{1,0}, \cdots, q_{M,0})). \end{aligned}$$

Therefore, our approach offers both flexibility and scalability.

We assume that there are N users. A user in a networked system is denoted by U_i, $i = 1, 2, \cdots, N$. Each user observes local observable events denoted by $\Sigma_{o,i}$. To describe the local observation, we use the projection $P_i : \Sigma^* \to \Sigma_{o,i}^*$ that erases all unobservable events from a string. Formally, $P_i(s)$ is defined recursively as

$$P_i(\varepsilon) = \varepsilon, \qquad P_i(s\sigma) = \begin{cases} P_i(s)\sigma & \text{if } \sigma \in \Sigma_{o,i} \\ P_i(s) & \text{otherwise} \end{cases}.$$

In other words, if a string of events $s \in L(G)$ occurred in the networked system G, User[2] U_i will observe $P_i(s)$. In this paper, we assume that U_i communicates its state estimate information to other users

[2]In the remainder of this paper, for simplicity, U_i is used to refer to User U_i.

based on its own local observation $w_i \in P_i(L(G))$ only, where $P_i(L(G))$ is the projection of $L(G)$, representing all possible local observations by U_i.

III. Information Exchange via State Estimates

We assume that the information contents to be exchanged/released by U_i are the estimates of states by U_i, that is, the set of possible states that G may be in after observing $w_i \in P_i(L(G))$. Formally, the local state estimate of U_i after observing w_i is defined by

$$E_i(w_i) = \{q \in Q : (\exists s \in L(G))P_i(s) = w_i \wedge \delta(q_0, s) = q\}.$$

Two types of information flows/exchanges among users are investigated: (1) Private communication from U_i to $U_j, j = 1, 2, \cdots, N$ and (2) Public broadcasting by U_i as described below.

(1) Private communication from U_i to U_j, based on U_i's local observation, is given by the following mapping

$$\theta_{ij} : P_i(L(G)) \rightarrow 2^Q,$$

such that, for all $w_i \in P_i(L(G))$,

$$\theta_{ij}(w_i) = \begin{cases} E_i(w_i) & \text{if } U_i \text{ communicates } E_i(w_i) \text{ to } U_j \\ Q & \text{otherwise} \end{cases}.$$

Note that, since $Q' \cap Q = Q'$ for any $Q' \subseteq Q$, $\theta_{ij}(w_i) = Q$ means that no information is communicated from U_i to U_j. The reason to use $\theta_{ij}(w_i) = Q$ rather than $\theta_{ij}(w_i) = \emptyset$ or $\theta_{ij}(w_i)$ is undefined is notation convenience (Q is not actually communicated) because we will take the intersection to combine state estimates later. The same for $\phi_i(w_i)$ below.

(2) Public broadcasting by U_i is given by the following mapping

$$\phi_i : P_i(L(G)) \rightarrow 2^Q,$$

such that, for all $w_i \in P_i(L(G))$,

$$\phi_i(w_i) = \begin{cases} E_i(w_i) & \text{if } U_i \text{ broadcasts } E_i(w_i) \text{ to the public} \\ Q & \text{otherwise} \end{cases}.$$

To calculate state estimate $E_i(w_i)$ and to obtain a finite implementation of θ_{ij} and ϕ_i, we construct the automaton generating $P_i(L(G))$, called observer and denoted by H_i as follows.

Step 1: Replace all transitions in G whose events are not in $\Sigma_{o,i}$ by ε-transitions and denote the resulting automaton as

$$G_{i,\varepsilon} = (Q, \Sigma_{o,i}, \delta_{i,\varepsilon}, q_0),$$

where $\delta_{i,\varepsilon} = \{(q, \sigma, q') : (q, \sigma, q') \in \delta \wedge \sigma \in \Sigma_{o,i}\} \cup \{(q, \varepsilon, q') : (q, \sigma, q') \in \delta \wedge \sigma \notin \Sigma_{o,i}\}$.

Step 2: Convert $G_{i,\varepsilon}$ to observer H_i as follows.

$$H_i = (X_i, \Sigma_{o,i}, \xi_i, x_{i,0}) = Ac(2^Q, \Sigma_{o,i}, \xi_i, UR(\{q_0\})),$$

where $Ac(\cdot)$ denotes the accessible part; $UR(\cdot)$ is the unobservable reach defined, for $x_i \subseteq Q$, as

$$UR(x_i) = \{q \in Q : (\exists q' \in x_i)q \in \delta_{i,\varepsilon}(q', \varepsilon)\}.$$

The transition function ξ is defined, for $x_i \in X_i$ and $\sigma \in \Sigma_{o,i}$ as

$$\xi(x_i, \sigma) = UR(\{q \in Q : (\exists q' \in x_i)q \in \delta_{i,\varepsilon}(q', \sigma)\}).$$

It is well-known [16], [22] that $L(H_i) = P_i(L(G))$ and $x_i = \xi_i(x_{i,0}, w_i)$ is the state estimate after observing w_i, that is,

$$E_i(w_i) = \xi_i(x_{i,0}, w_i).$$

Therefore, $x_i = \xi_i(x_{i,0}, w_i)$ represents two things: the state H_i is in and the state estimate at x_i (after observing w_i). Note that the state estimate $E_i(w_i)$ is calculated based on U_i's own observation (not the information communicated or broadcast by other users). The reason for this are: (1) minimize the mutual influences among users, (2) reduce the computational complexity of on-line computation, and (3) minimize the errors due to communication delays, loses, and attacks.

How to control information communicated or broadcast is the key to information control in networked systems. Information communicated or broadcast by $U_i, i = 1, 2, \ldots, N$ is controlled by a controller

$$\pi_i = (\varphi_i, \vartheta_{i1}, \cdots, \vartheta_{iN}),$$

where φ_i and ϑ_{ij} are control mappings

$$\varphi_i : X_i \rightarrow \{0, 1\}$$
$$\vartheta_{ij} : X_i \rightarrow \{0, 1\}, \ \ j = 1, 2, \cdots, N$$

defined as, for $x_i \in X_i$

$$\varphi_i(x_i) = \begin{cases} 1 & \text{if } U_i \text{ broadcasts } x_i \text{ to the public at } x_i \\ 0 & \text{otherwise} \end{cases}$$

$$\vartheta_{ij}(x_i) = \begin{cases} 1 & \text{if } U_i \text{ communicates } x_i \text{ to } U_j \text{ at } x_i \\ 0 & \text{otherwise} \end{cases}.$$

Obviously, for all $j = i$, $\vartheta_{ij}(x_i) = \vartheta_{jj}(x_j) = 1$.

Therefore,

$$\phi_i(w_i) = \begin{cases} \xi_i(x_{i,0}, w_i) & \text{if } \varphi_i(\xi_i(x_{i,0}, w_i)) = 1 \\ Q & \text{otherwise} \end{cases}$$

$$\theta_{ij}(w_i) = \begin{cases} \xi_i(x_{i,0}, w_i) & \text{if } \vartheta_{ij}(\xi_i(x_{i,0}, w_i)) = 1 \\ Q & \text{otherwise} \end{cases}. \quad (1)$$

We investigate how to design information control strategy $\pi = (\pi_1, \pi_2, \cdots, \pi_N)$. Intuitively, if U_i wants to help U_j to perform its tasks, then U_i shall send *all* of its state estimate to U_j. However, due to privacy, security, communication costs, and other factors, U_i may want to minimize the information it sends to U_j while still helping U_j to perform its task. On the other hand, if U_i wants to prevent U_j from performing its tasks, then U_i shall not send any of its state estimate to U_j and shall avoid broadcasting information that may help U_j to perform its tasks.

After the occurrence of $s \in L(G)$, $U_j, j = 1, 2, \cdots, N$ observes $P_j(s)$. Under control $\pi = (\pi_1, \pi_2, \cdots, \pi_N)$, its local state estimate $E_j(P_j(s))$ is enhanced by state estimates $\phi_i(P_i(s))$ and $\theta_{ij}(P_i(s))$ if they are broadcast and communicated by U_i, respectively. Hence, the enhanced state estimate of U_j after the occurrence of s is given by

$$\rho_j(s) = E_j(P_j(s)) \cap \phi_1(P_1(s)) \cap \cdots \cap \phi_N(P_N(s)) \cap \theta_{1j}(P_1(s)) \cap \cdots \cap \theta_{Nj}(P_N(s)). \quad (2)$$

Note that if nothing is broadcast and/or communicated by U_i, then $\phi_i(P_i(s)) = Q$ and/or $\theta_{ij}(P_i(s)) = Q$, meaning that there is no help from U_i.

We assume that, in order for U_j to perform its tasks, U_j needs to distinguish some states in G from some other states in G. Formally, let $T = Q \times Q$ be the set of all state pairs and let

$$T^j_{spec} \subseteq T$$

be the task specification for U_j. We say that U_j can perform its task if it can always distinguish all state pairs in T^j_{spec}, that is,

$$(\forall s \in L(G))(\rho_j(s) \times \rho_j(s)) \cap T^j_{spec} = \emptyset. \tag{3}$$

Remark 1: Specifying a task using T_{spec} is a broad approach, and it is applicable to most tasks. The following examples demonstrate that tasks related to supervisory control, diagnosability, and detectability can all be described using T_{spec}. In supervisory control [9], [10], a common objective is to prevent a system from entering illegal or unsafe states. To achieve this, a supervisor must distinguish between legal states $Q_l \subseteq Q$ and illegal states $Q_{il} \subseteq Q$. Therefore, $T_{spec} = (Q_l \times Q_{il}) \cup (Q_{il} \times Q_l)$. In diagnosability [12], a diagnoser needs to differentiate between normal states $Q_n \subseteq Q$ and fault states $Q_f \subseteq Q$. Hence, $T_{spec} = (Q_n \times Q_f) \cup (Q_f \times Q_n)$. The objective of detectability can also be specified using T_{spec} [23]. To calculate $\rho_j(s)$ and to check Equation (3), we take the parallel composition [16], [22] of G and H_i as

$$\begin{aligned}\tilde{G} &= (\tilde{Q}, \Sigma, \tilde{\delta}, \tilde{q}_0)\\ &= G \,\|H_1\,\| \cdots \|\, H_N\\ &= (Q \times X_1 \times \cdots \times X_N, \Sigma, \delta \times \xi_1 \times \cdots \times \xi_N,\\ &\quad (q_0, x_{1,0}, \cdots, x_{N,0})).\end{aligned}$$

The difference between G and $\tilde{G}$ is that G models the system without users while $\tilde{G}$ models the system with users. Since information control is among users, $\tilde{G}$ will be used in the investigation of information control.

We have the following theorem, which states that the enhanced state estimate of U_j after information exchange is the intersection of all state estimates communicated or broadcast to U_j.

Theorem 1: The enhanced state estimate of U_j after information exchange can be calculated as follows. For $s \in L(G)$, denote $\tilde{\delta}(\tilde{q}_0, s) = (q, x_1, \cdots, x_N)$. Then, the enhanced state estimate of U_j after the occurrence of s is given by

$$\rho_j(s) = x_j \cap \bigcap_{\varphi_i(x_i)=1 \vee \vartheta_{ij}(x_i)=1} x_i. \tag{4}$$

Proof: The language generated by $\tilde{G}$ is given by

$$\begin{aligned}&L(\tilde{G})\\ &= L(G) \cap P_1^{-1}(H_1) \cap \cdots \cap P_N^{-1}(H_N)\\ &= L(G) \cap P_1^{-1}(P_1(L(G))) \cap \cdots \cap P_N^{-1}(P_N(L(G)))\\ &= L(G).\end{aligned}$$

For $s \in L(\tilde{G}) = L(G)$, let $\tilde{\delta}(\tilde{q}_0, s) = \tilde{q} = (q, x_1, \cdots, x_N) \in Q \times X_1 \times \cdots \times X_N$. Then

$$x_i = \xi_i(x_{i,0}, P_i(s)) = E_i(P_i(s)),\ \ i = 1, 2, \cdots, N. \tag{5}$$

Therefore,

$$\begin{aligned}\rho_j(s) &= E_j(P_j(s)) \cap \phi_1(P_1(s)) \cap \cdots \cap \phi_N(P_N(s))\\ &\quad \cap \theta_{1j}(P_1(s)) \cap \cdots \cap \theta_{Nj}(P_N(s))\\ &= E_j(P_j(s)) \cap \bigcap_i \phi_i(P_i(s)) \cap \bigcap_i \theta_{ij}(P_i(s))\\ &= E_j(P_j(s)) \cap \bigcap_{\varphi_i(x_i)=1} x_i \cap \bigcap_{\vartheta_{ij}(x_i)=1} x_i\end{aligned}$$

(by Equations (1) and (5))

$$= x_j \cap \bigcap_{\varphi_i(x_i)=1 \vee \vartheta_{ij}(x_i)=1} x_i.$$

■

IV. Information Control Problems and Solutions

In this section, we propose five information control problems and provide their solutions. We assume that users are divided into two groups:

$$\begin{aligned}&Group\ 1 = \{1, 2, \cdots, N_1\},\\ &Group\ 2 = \{N_1 + 1, N_1 + 2, \cdots, N\}.\end{aligned}$$

Users in the same group are friends, and users in the other group are adversaries. Five information control problems are investigated. The first problem is stated as follows.

Information Control Problem 1: Can U_j perform its task based on its own local state estimates without state estimates from other users, including its friends?

Since U_j uses only its own local observation, we have

$$\rho_j(s) = x_j = \xi_j(x_{j,0}, P_j(s)).$$

Therefore, U_j can perform its task if and only if

$$\begin{aligned}&(\forall s \in L(G))(\rho_j(s) \times \rho_j(s)) \cap T^j_{spec} = \emptyset\\ &\Leftrightarrow (\forall x_j \in X_j)(x_j \times x_j) \cap T^j_{spec} = \emptyset.\end{aligned}$$

Note that the above condition can be checked based on H_j, without the need of constructing $\tilde{G}$. This can be proved as follows.

$$\begin{aligned}&(\forall s \in L(G))(\rho_j(s) \times \rho_j(s)) \cap T^j_{spec} = \emptyset\\ \Leftrightarrow&(\forall s \in L(G))(\xi_j(x_{j,0}, P_j(s)) \times \xi_j(x_{j,0}, P_j(s))) \cap T^j_{spec} = \emptyset\\ \Leftrightarrow&(\forall w_j \in L(H_j))(\xi_j(x_{j,0}, w_j)) \times \xi_j(x_{j,0}, w_j)) \cap T^j_{spec} = \emptyset\\ &(\text{let } P_j(s) = w_j)\\ \Leftrightarrow&(\forall x_j \in X_j)(x_j \times x_j) \cap T^j_{spec} = \emptyset.\end{aligned}$$

A. Information Control Among Friends

If the answer to Problem 1 is "no", then $U_j, j = 1, \ldots, N_1$ needs helps from other users. Hence, we investigate the second problem, stated as follows.

Information Control Problem 2: Can U_j perform its task based on its own local state estimates and all its friends' state estimates? In other words, assume that all its friends communicate their state estimates to U_j, can U_j perform its task?

If all its friends communicate all state estimates to U_j, then

$$\rho_j(s) = \bigcap_{i=1,2,\cdots,N_1} x_i = \bigcap_{i=1,2,\cdots,N_1} \xi_i(x_{i,0}, P_i(s)).$$

Therefore, U_j can perform its task if and only if

$$\begin{aligned}&(\forall s \in L(G))(\rho_j(s) \times \rho_j(s)) \cap T^j_{spec} = \emptyset\\ &\Leftrightarrow (\forall s \in L(\tilde{G}))(\rho_j(s) \times \rho_j(s)) \cap T^j_{spec} = \emptyset\\ &\Leftrightarrow (\forall s \in L(\tilde{G}))(\bigcap_{i=1,2,\cdots,N_1} \xi_i(x_{i,0}, P_i(s))\\ &\quad \times \bigcap_{i=1,2,\cdots,N_1} \xi_i(x_{i,0}, P_i(s))) \cap T^j_{spec} = \emptyset\\ &\Leftrightarrow (\forall \tilde{q} = (q, x_1, \cdots, x_N) \in Q \times X_1 \times \cdots \times X_N)\\ &(\bigcap_{i=1,2,\cdots,N_1} x_i \times \bigcap_{i=1,2,\cdots,N_1} x_i) \cap T^j_{spec} = \emptyset.\end{aligned}$$

If the answer to Problem 2 is "yes", then we investigate the following problem.

Information Control Problem 3: How to minimize communications from its friends to U_j, without affecting U_j's task?

Without loss of generality, let $U_j = U_1$. To minimize the communication, we proceed as follows. Since the answer to the second problem is "yes", we know that

$$\begin{aligned}&(\forall \tilde{q} = (q, x_1, \cdots, x_N) \in Q \times X_1 \times \cdots \times X_N)\\ &(\bigcap_{i=1,2,\cdots,N_1} x_i \times \bigcap_{i=1,2,\cdots,N_1} x_i) \cap T^1_{spec} = \emptyset.\end{aligned}$$

Let us remove $x_i, i = 2, \cdots, N_1$ one by one from the above equation and check if the equation is still satisfied. If it is satisfied, then we remove the x_i permanently (that is, U_i does not need to communicate its state estimates x_i to U_1); otherwise, we put the x_i back (that is, U_i needs to communicate its state estimates x_i to U_1).

Note that the above checking needs to be done for all $\tilde{q} = (q, x_1, \cdots, x_N) \in Q \times X_1 \times \cdots \times X_N$ and the answer may be different for different $\tilde{q}$. However, when implement control ϑ_{i1}, U_i does not know combined state $\tilde{q}$, its control is based on local state x_i. Therefore, when design control ϑ_{i1} (note that the designer does know the combined state during design), if x_i cannot be permanently removed in some combined state $\tilde{q} = (q, x_1, \cdots, x_N) \in Q \times X_1 \times \cdots \times X_N$, it should not be permanently removed at x_i. In other words, for $x_i \in X_i, i = 2, \cdots, N_1$,

$$\begin{aligned}&\vartheta_{i1}(x_i) = 0\\ &\Leftrightarrow (\forall \tilde{q} = (q, x_1, \cdots, x_N) \in Q \times X_1 \times \cdots \times X_N)\\ &(x_1 \cap \bigcap_{k=2,\cdots,N_1 \wedge k \neq i \wedge \vartheta_{k1}(x_k)=1} x_k \times x_1\\ &\cap \bigcap_{k=2,\cdots,N_1 \wedge k \neq i \wedge \vartheta_{k1}(x_k)=1} x_k) \cap T^1_{spec} = \emptyset\\ &\Leftrightarrow (\forall \tilde{q} = (q, x_1, \cdots, x_N) \in Q \times X_1 \times \cdots \times X_N)\\ &(\bigcap_{k=1,\cdots,N_1 \wedge k \neq i \wedge \vartheta_{k1}(x_k)=1} x_k\\ &\quad \times \bigcap_{k=1,\cdots,N_1 \wedge k \neq i \wedge \vartheta_{k1}(x_k)=1} x_k) \cap T^1_{spec} = \emptyset.\\ &(\text{because } \vartheta_{11}(x_1) = 1).\end{aligned}$$

The above procedure is summarized in Algorithm 1.

Algorithm 1 Design information control ϑ_{i1}

Input: $\tilde{G} = G \,||H_1\, || \,\cdots\, ||\, H_N$
Output: ϑ_{i1}

for $i = 1, 2, \cdots, N_1$ **do**
  for *all* $x \in X_i$ **do**
    $\vartheta_{i1}(x) = 1$;
for $i = N_1 + 1, N_1 + 2, \cdots, N$ **do**
  for *all* $x \in X_i$ **do**
    $\vartheta_{i1}(x) = 0$;
for $i = 2, \cdots, N_1$ **do**
  for *all* $x \in X_i$ **do**
    $\vartheta_{i1}(x) = 0$;
    for *all* $(q, x_1, \cdots, x_N) \in Q \times X_1 \times \cdots \times X_N \wedge x_i = x$ **do**
      $Ex = x_1$;
      for *all* $k = 2, \cdots, N_1 \wedge k \neq i \wedge \vartheta_{k1}(x_k) = 1$ **do**
        $Ex = Ex \cap x_k$;
      if $(Ex \times Ex) \cap T^1_{spec} \neq \emptyset$ **then**
        $\vartheta_{i1}(x) = 1$;
End.

To design information control ϑ_{i1} to minimize communication from U_i to U_1, Algorithm 1 initiates $\vartheta_{i1}(x) = 1$ for friends and $\vartheta_{i1}(x) = 0$ for adversaries. Algorithm 1 then checks if communication can be removed at some states x of some friends i. This check needs to be done for $N_1 - 1$ friends and $|\tilde{Q}|$ states. Therefore, the computational complexity of Algorithm 1 is $O((N_1 - 1)\ |\tilde{Q}|)$, given that $\tilde{G}$ has been constructed.

B. Information Control Among Adversaries

If the answer to Problem 2 is "no", then U_1 cannot perform its task unless some adversaries make some mistakes and release information that shall not be released. Therefore, the problem is how an adversary can avoid making such mistakes. If an adversary, say $U_j, j = N_1 + 1, \cdots, N$, has no obligation to broadcast any information, then its information control is simple: It shall only communicate with its friends to help them to perform their tasks. It shall not communicate anything to its adversaries, that is, $(\forall j = N_1 + 1, \cdots, N)(\forall i = 1, \cdots, N_1)(\forall x_j \in X_j)\vartheta_{ij}(x_i) = 0$, and it shall not broadcast any information to the public, that is, $(\forall j = N_1 + 1, \cdots, N)(\forall x_j \in X_j)\varphi_j(x_j) = 0$. On the other hand, if U_j has obligation to release as much information as it could to the public, then we investigate the following problem.

Information Control Problem 4: How should U_j broadcast maximum information to the public without helping U_1 to perform its task?

To maximize the broadcasting, we proceed as follows. Since the answer to the second problem is "no", we know that

$$\begin{aligned}&(\exists \tilde{q} = (q, x_1, \cdots, x_N) \in Q \times X_1 \times \cdots \times X_N)\\ &(\bigcap_{i=1,\cdots,N_1} x_i \times \bigcap_{i=1,\cdots,N_1} x_i) \cap T^1_{spec} \neq \emptyset.\end{aligned}$$

Let us add $x_j, j = N_1 + 1, \cdots, N$ one by one to the above equation and check if the equation is still satisfied. If it is satisfied, then we add the x_i permanently (that is, U_j can broadcast its state estimates to the public); otherwise, we do not add the x_j (that is, U_j cannot broadcast its state estimates to the public).

As in the previous problem, the above checking needs to be done for all $\tilde{q} = (q, x_1, \cdots, x_N) \in Q \times X_1 \times \cdots \times X_N$ and the answer may be different for different $\tilde{q}$. If x_j can be permanently added in some combined state $\tilde{q} = (q, x_1, \cdots, x_N) \in Q \times X_1 \times \cdots \times X_N$, then it can be permanently added. In other words, for $x_j \in X_j$,

$$
\begin{aligned}
&\varphi_j(x_j) = 1 \\
\Leftrightarrow &(\exists \tilde{q} = (q, x_1, \cdots, x_N) \in Q \times X_1 \times \cdots \times X_N) \\
&(x_j \cap \bigcap_{i=1,\cdots,N_1} x_i \cap \bigcap_{k=N_1+1,\cdots,N \wedge k \neq j \wedge \varphi_k(x_k)=1} x_k \\
&\quad \times x_j \cap \bigcap_{i=1,\cdots,N_1} x_i \cap \bigcap_{k=N_1+1,\cdots,N \wedge k \neq j \wedge \varphi_k(x_k)=1} x_k) \\
&\cap T^1_{spec} \neq \emptyset.
\end{aligned}
$$

Or, equivalently,

$$
\begin{aligned}
&\varphi_j(x_j) = 0 \\
&\Leftrightarrow (\forall \tilde{q} = (q, x_1, \cdots, x_N) \in Q \times X_1 \times \cdots \times X_N) \\
&(x_j \cap \bigcap_{i=1,\cdots,N_1} x_i \cap \bigcap_{k=N_1+1,\cdots,N \wedge k \neq j \wedge \varphi_k(x_k)=1} x_k \\
&\quad \times x_j \cap \bigcap_{i=1,\cdots,N_1} x_i \cap \bigcap_{k=N_1+1,\cdots,N \wedge k \neq j \wedge \varphi_k(x_k)=1} x_k) \\
&\cap T^1_{spec} = \emptyset.
\end{aligned}
$$

Algorithm 2 Design information control φ_j

```
Input: G~ = G ||H_1 || ··· || H_N
Output: φ_j
for j = N_1 + 1, ··· , N do
    for all x ∈ X_j do
        φ_j(x) = 0;
for j = N_1 + 1, ··· , N do
    for all x ∈ X_j do
        for all (q, x_1, ··· , x_N) ∈ Q × X_1 × ··· × X_N ∧ x_j = x do
            Ex = x_j ∩ ⋂_{i=1,··· ,N_1} x_i;
            for all k = N_1 + 1, ··· , N ∧ k ≠ j ∧ φ_k(x_k) = 1 do
                Ex = Ex ∩ x_k;
            if (Ex × Ex) ∩ T^1_spec ≠ ∅ then
                φ_j(x) = 1;
End.
```

The above procedure is summarized in Algorithm 2.

To design information control φ_j to maximize broadcasting by U_j, $j = N_1 + 1, \cdots, N$, Algorithm 2 initiates $\varphi_j(x) = 0$. Algorithm 2 then checks if broadcasting can be added at some states x of some users j. This check needs to be done for $N - N_1$ users and $|\tilde{Q}|$ states. Therefore, the computational complexity of Algorithm 2 is $O((N - N_1)\ |\tilde{Q}|)$, given that G has been constructed.

C. Information Control Under Minimum Information Release Requirement

In some cases, for transparency, fairness, and/or other reasons, the system operator may request each user to broadcast some minimum information to the public. This minimum requirement is given by

$$
\varphi_{j,min}(x_j), \text{ for } x_j \in X_j, j = 1, 2, \cdots, N,
$$

where $\varphi_{j,min} : X_j \to \{0, 1\}$ is a particular control map $\varphi_j : X_j \to \{0, 1\}$ specifying the minimally required broadcasting.

If minimal broadcasting is required, then we need to solve the following problem.

Information Control Problem 5: What are the impacts of minimally required broadcasting $\varphi_{j,min}$ on information control?

To solve this fifth information control problem, we discuss the impacts on each of the information control problem discussed above as follows.

Information Control Problem 1 becomes: Can U_j perform its task based on its own local state estimates and minimally required broadcasting from other users?

The answer is "yes" if the following is satisfied.

$$
\begin{aligned}
&(\forall \tilde{q} = (q, x_1, \cdots, x_N) \in Q \times X_1 \times \cdots \times X_N) \\
&(x_j \cap \bigcap_{i=1,\cdots,N \wedge \varphi_{i,min}(x_i)=1} x_i \\
&\quad \times x_j \cap \bigcap_{i=1,\cdots,N \wedge \varphi_{i,min}(x_i)=1} x_i) \cap T^j_{spec} = \emptyset.
\end{aligned}
$$

Information Control Problem 2 becomes: Can U_j perform its task based on its own local state estimates, all its friends' state estimates, and minimally required broadcasting from other users?

The answer is "yes" if the following is satisfied.

$$
\begin{aligned}
&(\forall \tilde{q} = (q, x_1, \cdots, x_N) \in Q \times X_1 \times \cdots \times X_N) \\
&(\bigcap_{k=1,\cdots,N_1} x_k \cap \bigcap_{i=1,\cdots,N \wedge \varphi_{i,min}(x_i)=1} x_i \\
&\quad \times \bigcap_{k=1,\cdots,N_1} x_k \cap \bigcap_{i=1,\cdots,N \wedge \varphi_{i,min}(x_i)=1} x_i) \cap T^j_{spec} = \emptyset.
\end{aligned}
$$

Information Control Problem 3 becomes: How to minimize communications from its friends to U_1 knowing the minimally required broadcasting from other users?

The solution is given by: for $x_i \in X_i$,

$$
\begin{aligned}
&\vartheta_{i1}(x_i) = 0 \\
\Leftrightarrow &(\forall \tilde{q} = (q, x_1, \cdots, x_N) \in Q \times X_1 \times \cdots \times X_N) \\
&(\bigcap_{k=1,\cdots,N_1 \wedge k \neq i \wedge \vartheta_{k1}(x_k)=1} x_k \cap \bigcap_{i=1,\cdots,N \wedge \varphi_{i,min}(x_i)=1} x_i \\
&\quad \times \bigcap_{k=1,\cdots,N_1 \wedge k \neq i \wedge \vartheta_{k1}(x_k)=1} x_k \cap \bigcap_{i=1,\cdots,N \wedge \varphi_{i,min}(x_i)=1} x_i) \\
&\cap T^1_{spec} = \emptyset.
\end{aligned}
$$

Algorithm 1 is then modified as Algorithm 3.

Algorithm 3 Design information control ϑ_{i1} with minimally required broadcasting

Input: $\tilde{G} = G \,||H_1\,||\,\cdots\,||\,H_N$
Output: ϑ_{i1}

for $i = 1, \cdots, N_1$ **do**
  for *all* $x \in X_i$ **do**
    $\vartheta_{i1}(x) = 1$;
for $i = N_1 + 1, \cdots, N$ **do**
  for *all* $x \in X_i$ **do**
    $\vartheta_{i1}(x) = 0$;
for $i = 2, \cdots, N_1$ **do**
  for *all* $x \in X_i$ **do**
    $\vartheta_{i1}(x) = 0$;
    for *all* $(q, x_1, \cdots, x_N) \in Q \times X_1 \times \cdots \times X_N \wedge x_i = x$ **do**
      $Ex = x_1$;
      for *all* $k = 2, \cdots, N_1 \wedge k \neq i \wedge \vartheta_{k1}(x_k) = 1$ **do**
        $Ex = Ex \cap x_k$;
      for *all* $k = 1, \cdots, N \wedge \varphi_{i,min}(x_k) = 1$ **do**
        $Ex = Ex \cap x_k$;
      if $(Ex \times Ex) \cap T^1_{spec} \neq \emptyset$ **then**
        $\vartheta_{i1}(x) = 1$;
End.

Algorithm 4 Design control φ_j with minimally required broadcasting

Input: $\tilde{G} = G \,||H_1\,||\,\cdots\,||\,H_N$
Output: φ_j

for $j = N_1 + 1, \cdots, N$ **do**
  for *all* $x \in X_j$ **do**
    $\varphi_j(x) = 0$;
for $j = N_1 + 1, \cdots, N$ **do**
  for *all* $x \in X_j$ **do**
    for *all* $(q, x_1, \cdots, x_N) \in Q \times X_1 \times \cdots \times X_N \wedge x_j = x$ **do**
      $Ex = x_j \cap \bigcap_{i=1,\cdots,N_1} x_i$;
      for *all* $k = N_1 + 1, \cdots, N \wedge k \neq j \wedge \varphi_k(x_k) = 1$ **do**
        $Ex = Ex \cap x_k$;
      for *all* $k = 1, \cdots, N \wedge \varphi_{i,min}(x_k) = 1$ **do**
        $Ex = Ex \cap x_k$;
      if $(Ex \times Ex) \cap T^1_{spec} \neq \emptyset$ **then**
        $\varphi_j(x) = 1$;
End.

Algorithm 3 initiates $\vartheta_{i1}(x) = 1$ for friends and $\vartheta_{i1}(x) = 0$ for adversaries. It then checks if communication can be removed at some states x of some friends i in a way similar to that of Algorithm 1. This check needs to be done for $N_1 - 1$ friends and $|\tilde{Q}|$ states. Therefore, the computational complexity of Algorithm 3 is also $O((N_1 - 1)\,|\tilde{Q}|)$.

Information Control Problem 4 becomes: How to broadcast maximum information to the public without helping U_1 to perform its task knowing the minimally required broadcasting from other users?

The solution is given by: for $x_i \in X_i$,

$$\begin{aligned}
&\varphi_j(x_j) = 1\\
\Leftrightarrow &(\exists \tilde{q} = (q, x_1, \cdots, x_N) \in Q \times X_1 \times \cdots \times X_N)\\
&\Big(\bigcap_{i=1,\cdots,N_1} x_i \cap \bigcap_{k=N_1+1,\cdots,N \wedge k\neq j \wedge \varphi_k(x_k)=1} x_k\\
&\cap \bigcap_{i=1,\cdots,N \wedge \varphi_{i,min}(x_i)=1} x_i\\
&\quad \times \bigcap_{i=1,\cdots,N_1} x_i \cap \bigcap_{k=N_1+1,\cdots,N \wedge k\neq j \wedge \varphi_k(x_k)=1} x_k\\
&\cap \bigcap_{i=1,\cdots,N \wedge \varphi_{i,min}(x_i)=1} x_i\Big) \cap T^1_{spec} \neq \emptyset.
\end{aligned}$$

Algorithm 2 is then modified as Algorithm 4.

Algorithm 4 initiates $\varphi_j(x) = 0$ and then checks if broadcasting can be added at some states x of some users j in a way similar to that of Algorithm 2. This check needs to be done for $N - N_1$ users and $|\tilde{Q}|$ states. Therefore, the computational complexity of Algorithm 4 is also $O((N - N_1)\,|\tilde{Q}|)$.

V. Communicating/Broadcasting Events Versus State Estimates

In [24], information control problems are investigated in the framework of communicating/broadcasting events. In comparison, this paper uses the framework of communicating/broadcasting state estimates. We discuss the advantages of each framework.

The advantage of communicating/broadcasting state estimates is that the computational complexity is low. While the computational complexity of Algorithm 1 is $O((N_1 - 1)\,|\tilde{Q}|)$, the correspond algorithm in [24] has computational complexity $O((N_1 - 1)\,|\tilde{Q}|\,|\Sigma|^2\,|2^{\tilde{Q}}|)$. Clearly, the computational complexity is significantly reduced, because there is no need to construct observer $\tilde{G}^i_{obs}$ from $\tilde{G}$.

While the computational complexity of Algorithm 2 is $O((N - N_1)\,|\tilde{Q}|)$, the correspond algorithm in [24] has computational complexity $O((N - N_1)\,|\tilde{Q}|\,|\Sigma|^2\,|2^{\tilde{Q}}|)$. Similar analysis can be done for Algorithms 3 and 4. In all cases, the computational complexity is significantly reduced. It is well known in DES that the minimal communication problem (corresponding to Algorithm 2) has high computational complexities. So, the computational complexities of Algorithms 1-4 are the best that one can expect.

Although communicating/broadcasting state estimates has much low computational complexity, it is less accurate than communicating/broadcasting events. To show this, consider two projections $P_1 : \Sigma^* \to \Sigma^*_{o,1}$, $P_2 : \Sigma^* \to \Sigma^*_{o,2}$, and the joint projection $P_{12} : \Sigma^* \to (\Sigma_{o,1} \cup \Sigma_{o,2})^*$. After the occurrence of $s \in L(G)$, the state estimates with respect to these projections are

$$\begin{aligned}
&E_1(P_1(s))\\
&\ = \{q \in Q : (\exists s' \in L(G))P_1(s') = P_1(s) \wedge \delta(q_0, s') = q\}\\
&E_2(P_2(s))\\
&\ = \{q \in Q : (\exists s' \in L(G))P_2(s') = P_2(s) \wedge \delta(q_0, s') = q\}
\end{aligned}$$

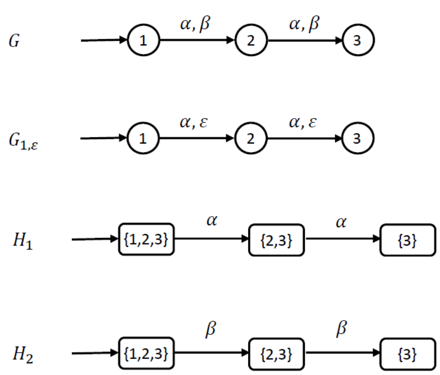

Fig. 1. A counter example of the state estimates, where $'' \rightarrow ''$ without event label denotes the initial state.

Fig. 2. A distribution system covering 18 cities in USA.

$$E_{12}(P_{12}(s)) \\ = \{q \in Q : (\exists s' \in L(G))P_{12}(s') = P_{12}(s) \wedge \delta(q_0, s') = q\}.$$

If U_2 sends the its state estimate $E_2(P_2(s))$ to U_1, then U_1 concludes that the state of G is in $E_1(P_1(s)) \cap E_2(P_2(s))$. If U_2 sends its observable events to U_1, then U_1 concludes that the state of G is in $E_{12}(P_{12}(s))$. The following theorem says that $E_{12}(P_{12}(s))$ is smaller (and hence better) than $E_1(P_1(s)) \cap E_2(P_2(s))$.

Theorem 2: The relation among three state estimates above is as follows. After the occurrence of any $s \in L(G)$, the state estimates satisfy the following condition

$$E_{12}(P_{12}(s)) \subseteq E_1(P_1(s)) \cap E_2(P_2(s)).$$

Proof: We prove that, for all $q \in Q$,

$$q \in E_{12}(P_{12}(s)) \;\Rightarrow\; q \in E_1(P_1(s)) \cap E_2(P_2(s)).$$

as follows.

$$\begin{aligned} & q \in E_{12}(P_{12}(s)) \\ \Leftrightarrow\; & (\exists s' \in L(G))P_{12}(s') = P_{12}(s) \wedge \delta(q_0, s') = q \\ \Rightarrow\; & (\exists s' \in L(G))P_1(s') = P_1(s) \wedge P_2(s') = P_2(s) \\ & \wedge \delta(q_0, s') = q \\ & (\text{because } P_i(s) = P_i(P_{12}(s)),\ i = 1, 2) \\ \Leftrightarrow\; & (\exists s' \in L(G))P_1(s') = P_1(s) \wedge \delta(q_0, s') = q \\ & \wedge P_2(s') = P_2(s) \wedge \delta(q_0, s') = q \\ \Rightarrow\; & (\exists s' \in L(G))P_1(s') = P_1(s) \wedge \delta(q_0, s') = q \\ & \wedge (\exists s' \in L(G))P_2(s') = P_2(s) \wedge \delta(q_0, s') = q \\ & (\text{because } (\exists x)(A(x) \wedge B(x)) \Rightarrow ((\exists x)A(x) \\ & \wedge ((\exists x)B(x))) \\ \Leftrightarrow\; & q \in E_1(P_1(s)) \cap E_2(P_2(s)). \end{aligned}$$

■

Note that, in general,

$$E_{12}(P_{12}(s)) \supseteq E_1(P_1(s)) \cap E_2(P_2(s))$$

is not true as shown in the following counter-example.

Example 1: Consider the system G shown in Figure 1. Assume that there are two users with

$$\Sigma_{o,1} = \{\alpha\}, \quad \Sigma_{o,2} = \{\beta\}.$$

Then $G_{1,\varepsilon}$, H_1, and H_2 are constructed as shown in Figure 1. Clearly,

$$E_{12}(P_{12}(s)) \supseteq E_1(P_1(s)) \cap E_2(P_2(s))$$

is not true, because for $s = \alpha$,

$$E_{12}(P_{12}(s)) = \{2\} \not\supseteq E_1(P_1(s)) \cap E_2(P_2(s)) = \{2, 3\}$$

Remark 2: According to Theorem 2 and Example 1, if U_1 can perform its task based on the state estimates communicated by U_2, then it can perform its task based on the (observed) events communicated by U_2. However, the vice versa is not true. That is, if U_1 can perform its task based on the event communication mechanism, it may not perform its task based on the state estimate communication.

VI. Illustrative Example

In this section, an example is used to illustrate the results of the previous sections. In order to draw the automata, the example is simple and is for illustration only.

Let us consider a distribution system shown in Figure 2. The system consists of 18 cities in USA. These cities are linked by railways as shown in the automaton G of Figure 3. In G, states represent cities as follows.

q_1 : Seattle	q_2 : Portland
q_3 : San Francisco	q_4 : Los Angeles
q_5 : San Diego	q_6 : Salt Lake City
q_7 : Phoenix	q_8 : Denver
q_9 : Minneapolis	q_{10} : Chicago
q_{11} : Detroit	q_{12} : New York City
q_{13} : Baltimore	q_{14} : Washington D.C.
q_{15} : Miami	q_{16} : Houston
q_{17} : Austin	q_{18} : Dallas.

If there is a direct railway link between city q_i and city q_j, then two events are defined as follows.

$$\alpha_{i,j}: \text{ a train moves from } q_i \text{ to } q_j.$$

Hence, there is a direct railway links cities q_i and q_j if and only if there are transitions between state q_i and state q_j.

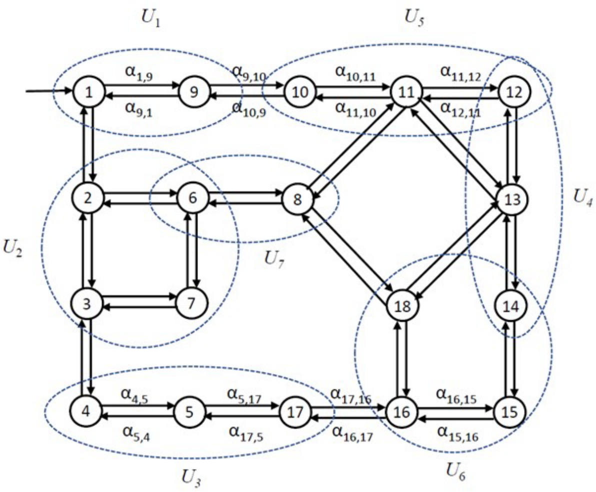

Fig. 3. Automaton G of the distribution system.

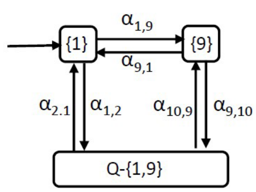

Fig. 4. The observer H_1 of G with respect to P_1.

Without loss of generality, we assume that the initial state is q_1, which means that there is initially a train in the proximity of Seattle.

Note that for the clarity of the figure, state q_i is denoted by i and not all events are labeled in Figure 3, because these labels are obvious. Note also that for this illustrative example, there is no need to use parallel composition to obtain G.

The distribution system is managed by $N = 7$ distributors/users. The cities covered by each distributor are also shown in Figure 3. For example, U_1 covers Seattle and Minneapolis, while U_4 covers New York City, Baltimore, and Washington D.C. Note that a city may be covered by more than one distributor.

The local events $\Sigma_{o,i}$ for $U_i, i = 1, 2, 3, 4, 5, 6, 7$ are movements of a train from or to a city covered by U_i. For example,

$$\Sigma_{o,1} = \{\alpha_{1,9}, \alpha_{9,1}, \alpha_{1,2}, \alpha_{2,1}, \alpha_{9,10}, \alpha_{10,9}\},$$
$$\Sigma_{o,4} = \{\alpha_{11,12}, \alpha_{12,11}, \alpha_{11,13}, \alpha_{13,11}, \alpha_{12,13}, \alpha_{13,12}, \alpha_{13,18}, \alpha_{18,13}, \alpha_{13,14}, \alpha_{14,13}, \alpha_{14,15}, \alpha_{15,14}\}.$$

The corresponding observers

$$H_1 = (X_1, \Sigma_{o,1}, \xi_1, x_{1,0}), H_4 = (X_4, \Sigma_{o,4}, \xi_4, x_{4,0})$$

are shown in Figure 3 and Figure 4, respectively.

The users are divided into two groups:

Group 1 = {1, 2, 3, 4 }, Group 2 = {5, 6, 7 },

that is, $N_1 = 4$.

To perform its tasks, U_1 needs to know if the train has arrived in Baltimore, that is, state q_{13} must be distinguishable from other states. Thus, the specification for U_1 is given by

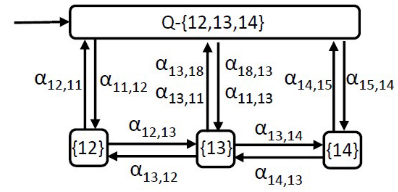

Fig. 5. The observer H_4 of G with respect to P_4.

TABLE I

State Relabeling From q to $\tilde{q} = (q, x_1, x_2, \cdots, x_7)$

q	x_1	x_2	x_3	x_4	x_5	x_6	x_7
1	{1}	Q-{2,3,6,7}	Q-{4,5,17}	Q-{12,13,14}	Q-{10,11,12}	Q-{14,15,16,18}	Q-{6,8}
2	Q-{1,9}	{2}	Q-{4,5,17}	Q-{12,13,14}	Q-{10,11,12}	Q-{14,15,16,18}	Q-{6,8}
3	Q-{1,9}	{3}	Q-{4,5,17}	Q-{12,13,14}	Q-{10,11,12}	Q-{14,15,16,18}	Q-{6,8}
4	Q-{1,9}	Q-{2,3,6,7}	{4}	Q-{12,13,14}	Q-{10,11,12}	Q-{14,15,16,18}	Q-{6,8}
5	Q-{1,9}	Q-{2,3,6,7}	{5}	Q-{12,13,14}	Q-{10,11,12}	Q-{14,15,16,18}	Q-{6,8}
6	Q-{1,9}	{6}	Q-{4,5,17}	Q-{12,13,14}	Q-{10,11,12}	Q-{14,15,16,18}	{6}
7	Q-{1,9}	{7}	Q-{4,5,17}	Q-{12,13,14}	Q-{10,11,12}	Q-{14,15,16,18}	Q-{6,8}
8	Q-{1,9}	Q-{2,3,6,7}	Q-{4,5,17}	Q-{12,13,14}	Q-{10,11,12}	Q-{14,15,16,18}	{8}
9	{9}	Q-{2,3,6,7}	Q-{4,5,17}	Q-{12,13,14}	Q-{10,11,12}	Q-{14,15,16,18}	Q-{6,8}
10	Q-{1,9}	Q-{2,3,6,7}	Q-{4,5,17}	Q-{12,13,14}	{10}	Q-{14,15,16,18}	Q-{6,8}
11	Q-{1,9}	Q-{2,3,6,7}	Q-{4,5,17}	Q-{12,13,14}	{11}	Q-{14,15,16,18}	Q-{6,8}
12	Q-{1,9}	Q-{2,3,6,7}	Q-{4,5,17}	{12}	{12}	Q-{14,15,16,18}	Q-{6,8}
13	Q-{1,9}	Q-{2,3,6,7}	Q-{4,5,17}	{13}	Q-{10,11,12}	Q-{14,15,16,18}	Q-{6,8}
14	Q-{1,9}	Q-{2,3,6,7}	Q-{4,5,17}	{14}	Q-{10,11,12}	{14}	Q-{6,8}
15	Q-{1,9}	Q-{2,3,6,7}	Q-{4,5,17}	Q-{12,13,14}	Q-{10,11,12}	{15}	Q-{6,8}
16	Q-{1,9}	Q-{2,3,6,7}	Q-{4,5,17}	Q-{12,13,14}	Q-{10,11,12}	{16}	Q-{6,8}
17	Q-{1,9}	Q-{2,3,6,7}	{17}	Q-{12,13,14}	Q-{10,11,12}	Q-{14,15,16,18}	Q-{6,8}
18	Q-{1,9}	Q-{2,3,6,7}	Q-{4,5,17}	Q-{12,13,14}	Q-{10,11,12}	{18}	Q-{6,8}

$$T^1_{spec} = \{(q_{13}, q_i) : q_i \in Q - \{q_{13}\}\}. \tag{6}$$

The specifications for other users can be defined similarly. Let us now solve the information control problems investigated in the previous section as follows.

Information Control Problem 1: Can U_1 perform its task based on its own local state estimates without state estimates from other users, including its friends?

To solve this problem, let us check if

$$(\forall x_1 \in X_1)(x_1 \times x_1) \cap T^1_{spec} = \emptyset$$

is satisfied or not. From Figure 3, we can see that the above condition is not satisfied because state q_{13} is mixed with other states in $Q - \{q_1, q_9\}$. Therefore, U_1 cannot perform its task solely based on its own observations and state estimates.

Information Control Problem 2: Can U_1 perform its task based on its own local state estimates and all its friends' state estimates?

To solve this problem, let us construct

$$\tilde{G} = G \,\|H_1\, \| \,\cdots\, \|\, H_7.$$

$\tilde{G}$ is isomorphic to G with state relabeling from q to $\tilde{q} = (q, x_1, x_2, \cdots, x_7)$ as shown in Table I. For clarity, q_i is also denoted by i in the tables.

If all its friends communicate all state estimates to U_1, then U_1 can perform its task if and only if

$$(\forall \tilde{q} = (q, x_1, \cdots, x_7) \in Q \times X_1 \times \cdots \times X_7)$$
$$(\bigcap_{i=1,2,3,4} x_i \times \bigcap_{i=1,2,3,4} x_i) \cap T^1_{spec} = \emptyset.$$

To see if the above condition is satisfied, we calculate $\bigcap_{i=1,2,3,4} x_i$ for all $\tilde{q} = (q, x_1, \cdots, x_7)$ as shown in Table II, Column 2. As can be seen, the condition is satisfied, because

TABLE II
INTERSECTIONS OF STATE ESTIMATES

q	Intersection of Users 1,2,3,4	Intersection of Users 1,2,3	Intersections of Users 1,2,4,5,7
1	{1}	{1}	{1}
2	{2}	{2}	{2}
3	{3}	{3}	{3}
4	{4}	{4}	{4,5,15,16,17,18}
5	{5}	{5}	{4,5,15,16,17,18}
6	{6}	{6}	{6}
7	{7}	{7}	{7}
8	{8,10,11,15,16,18}	{8,10,11,12,13,14,15,16,18}	{8}
9	{9}	{9}	{9}
10	{8,10,11,15,16,18}	{8,10,11,12,13,14,15,16,18}	{10}
11	{8,10,11,15,16,18}	{8,10,11,12,13,14,15,16,18}	{11}
12	{12}	{8,10,11,12,13,14,15,16,18}	{12}
13	{13}	{8,10,11,12,13,14,15,16,18}	{13}
14	{14}	{8,10,11,12,13,14,15,16,18}	{14}
15	{8,10,11,15,16,18}	{8,10,11,12,13,14,15,16,18}	{4,5,15,16,17,18}
16	{8,10,11,15,16,18}	{8,10,11,12,13,14,15,16,18}	{4,5,15,16,17,18}
17	{17}	{17}	{4,5,15,16,17,18}
18	{8,10,11,15,16,18}	{8,10,11,12,13,14,15,16,18}	{4,5,15,16,17,18}

state q_{13} is not mixed with any other states. Therefore, U_1 can perform its task based on its own local state estimates and all its friends' state estimates.

Information Control Problem 3

Since the answer to the second problem is "yes", the third problem is how to minimize communications from its friends to U_1. Algorithm 1 is used to find the minimum communication. Let us illustrate the execution of Algorithm 1 as follows.

Steps 1-6 initiate $\vartheta_{i1}(x)$ as $\vartheta_{i1}(x) = 1$ for all $x \in X_i$ and $i = 1, 2, 3, 4$ and $\vartheta_{i1}(x) = 0$ for all $x \in X_i$ and $i = 5, 6, 7$.

Steps 7-15 minimize the communications $\vartheta_{i1}(x)$ when possible for $i = 2, 3, 4$ and $x \in X_i$.

For example, let $i = 2$ and $x = Q - \{2, 3, 6, 7\}$. Steps 10-15 check all $x = Q - \{2, 3, 6, 7\}$ in Table I, Column 3 and see if

$$\begin{aligned}
&(\forall \tilde{q} = (q, x_1, \cdots, x_y) \in Q \times X_1 \times \cdots \times X_y) \\
&(\bigcap_{k=1,3,4 \wedge \vartheta_{k1}(x_k)=1} x_k \times \bigcap_{k=1,3,4 \wedge \vartheta_{k1}(x_k)=1} x_k) \cap T^1_{spec} = \emptyset \\
\Leftrightarrow &(\forall \tilde{q} = (q, x_1, \cdots, x_y) \in Q \times X_1 \times \cdots \times X_y) \\
&(\bigcap_{k=1,3,4} x_k \times \bigcap_{k=1,3,4} x_k) \cap T^1_{spec} = \emptyset \\
&(\text{because } \vartheta_{k1}(x_k) = 1 \text{ at this point}) \qquad (7)
\end{aligned}$$

is satisfied or not.

The largest $\bigcap_{k=1,3,4} x_k$ corresponding to $x = Q - \{2, 3, 6, 7\}$ in Table I, Column 3 is calculated as

$$\begin{aligned}
&\bigcap_{k=1,3,4} x_k \\
&= (Q - \{1, 9\}) \cap (Q - \{4, 5, 17\}) \cap (Q - \{12, 13, 14\}) \\
&= \{2, 3, 6, 7, 8, 10, 11, 15, 16, 18\}.
\end{aligned}$$

Hence, Equation (7) is satisfied. Therefore, $\vartheta_{21}(x) = 0$, that is, U_2 does not need to communicate $x = Q - \{2, 3, 6, 7\}$ to U_1 at $x = Q - \{2, 3, 6, 7\}$.

Similarly, Algorithm 1 determines that U_2 does not need to communicate x to U_1 at all other states $x = \{2\}, \{3\}, \{6\}, \{7\}$. For $i = 3$, Algorithm 1 determines that U_3 does not need to communicate x to U_1 at any $x \in X_3$. For $i = 4$, however, Algorithm 1 determines that U_4 needs to communicate x to U_1 at all $x \in X_4$.

Information Control Problem 4: How can a user broadcast maximum information to the public without helping its adversaries?

To illustrate this problem, let us move U_4 from *Group* 1 to *Group* 2, that is,

$$Group\ 1 = \{1, 2, 3\}, \quad Group\ 2 = \{4, 5, 6, 7\}.$$

Since U_4 is now an adversary of U_1, it shall not communicate anything to U_1. Without communication from U_4, let us calculate $\bigcap_{i=1,2,3} x_i$ for all $\tilde{q} = (q, x_1, \cdots, x_7)$ as shown in Table II, Column 3.

Since state q_{13} is mixed with other states in some elements of Table II, Column 3, the condition

$$\begin{aligned}
&(\forall \tilde{q} = (q, x_1, \cdots, x_7) \in Q \times X_1 \times \cdots \times X_7) \\
&(\bigcap_{i=1,2,3} x_i \times \bigcap_{i=1,2,3} x_i) \cap T^j_{spec} = \emptyset
\end{aligned}$$

is not satisfied. Thus, U_1 cannot perform its task based on its own local state estimates and all its friends' state estimates. So, the problem is: How can U_4, U_5, U_6, U_7 broadcast maximum information to the public without helping U_1? Algorithm 2 is used to find the maximum broadcasting. Let us illustrate the execution of Algorithm 2 as follows.

Steps 1-3 initiate $\varphi_j(x)$ as $\varphi_j(x) = 0$ for all $x \in X_j$ and $j = 4, 5, 6, 7$. Note that $\varphi_j(x)$ is irrelevant in Information Control Problem 4 for $U_j, j = 1, 2, 3$ as they are in the same group and can communicate among themselves if needed.

Steps 4-11 maximize the broadcasting $\varphi_j(x)$ when possible for $j = 4, 5, 6, 7$ and $x \in X_j$.

For example, let $j = 4$ and $x = Q - \{12, 13, 14\}$. Step 7-11 check all $x = Q - \{12, 13, 14\}$ in Table I, Column 5 and see if

$$\begin{aligned}
&(\forall \tilde{q} = (q, x_1, \cdots, x_7) \in Q \times X_1 \times \cdots \times X_7) \\
&(x_4 \cap \bigcap_{i=1,2,3} x_i \cap \bigcap_{k=5,6,7 \wedge \varphi_k(x_k)=1} x_k \\
&\quad \times x_4 \cap \bigcap_{i=1,2,3} x_i \cap \bigcap_{k=5,6,7 \wedge \varphi_k(x_k)=1} x_k) \cap T^1_{spec} = \emptyset \\
\Leftrightarrow &(\forall \tilde{q} = (q, x_1, \cdots, x_7) \in Q \times X_1 \times \cdots \times X_7) \\
&(x_4 \cap \bigcap_{i=1,2,3} x_i \times x_4 \cap \bigcap_{i=1,2,3} x_i) \cap T^1_{spec} = \emptyset \\
&(\text{because } \varphi_k(x_k) = 0 \text{ at this point}) \qquad (8)
\end{aligned}$$

is satisfied or not.

The largest $x_4 \cap \bigcap_{i=1,2,3} x_i$ corresponding to $x = Q - \{12, 13, 14\}$ in Table I, Column 5 is calculated as

$$\begin{aligned}
&x_4 \cap \bigcap_{i=1,2,3} x_i \\
&= (Q - \{1, 9\}) \cap (Q - \{2, 3, 6, 7\}) \cap (Q - \{4, 5, 17\}) \\
&\cap (Q - \{12, 13, 14\}) \\
&= \{8, 10, 11, 15, 16, 18\}.
\end{aligned}$$

Hence, Equation (8) is satisfied. Therefore, $\varphi_j(x_j) = 0$, that is, U_4 cannot broadcast $x = Q - \{12, 13, 14\}$ to the public at $x = Q - \{12, 13, 14\}$.

Similarly, Algorithm 4 determines that U_4 cannot broadcast x to the public at all other states ($x = \{12\}, \{13\}$, and $\{14\}$). For $j = 5, 6, 7$, Algorithm 2 determines that U_j can broadcast x

to the public at all $x \in X_j$. Intuitively, this is because only U_4 knows if the train has arrived in Baltimore or not. As long as U_4 does not broadcast its state estimates, the other users can broadcast their state estimates.

Information Control Problem 5

In solving the above four information control problems, we assume that there is no minimum information release required by the system operator, that is,

$$(\forall x_i \in X_i)\varphi_{i,min}(x_i) = \emptyset.$$

We now relax this assumption. We consider the following minimally required information release.

$$\begin{aligned}&(\forall x_1 \in X_1)\varphi_{1,min}(x_1) = 0, (\forall x_2 \in X_2)\varphi_{2,min}(x_2) = 1\\&(\forall x_3 \in X_3)\varphi_{3,min}(x_3) = 0, (\forall x_4 \in X_4)\varphi_{4,min}(x_4) = 1\\&(\forall x_5 \in X_5)\varphi_{5,min}(x_5) = 1, (\forall x_6 \in X_6)\varphi_{6,min}(x_6) = 0\\&(\forall x_7 \in X_7)\varphi_{7,min}(x_7) = 1.\end{aligned} \tag{9}$$

In other words, U_2, U_4, U_5, and U_7 are required to broadcast their state estimates. For the above $\varphi_{i,min}$, Information Control Problem 1 becomes: Can U_1 perform its task based on its own local state estimates and minimally required broadcasting from other users?

The answer is "yes" if the following is satisfied.

$$\begin{aligned}&(\forall \tilde{q} = (q, x_1, \cdots, x_N) \in Q \times X_1 \times \cdots \times X_N)\\&(x_1 \cap \bigcap_{i=1,\cdots,N \wedge \varphi_{i,min}(x_i)=1} x_i\\&\quad \times x_1 \cap \bigcap_{i=1,\cdots,N \wedge \varphi_{i,min}(x_i)=1} x_i) \cap T^j_{spec} = \emptyset\\\Leftrightarrow&(\forall \tilde{q} = (q, x_1, \cdots, x_N) \in Q \times X_1 \times \cdots \times X_N)\\&(x_1 \cap x_2 \cap x_4 \cap x_5 \cap x_7 \times x_1 \cap x_2 \cap x_4 \cap x_5 \cap x_7)\\&\cap T^j_{spec} = \emptyset.\end{aligned}$$

$x_1 \cap x_2 \cap x_4 \cap x_5 \cap x_7$ is calculated as shown in Table II, Column 4. Since state q_{13} is not mixed with other states in all elements of Table II, Column 4, the above condition is satisfied. Hence, U_1 can perform its task based on its own local state estimates and minimally required broadcasting from other users.

Similarly, we can update the solutions to other information control problems with the minimally required information release given in Equation (9).

VII. Conclusion

This paper presents a systematic approach to information control within networked multi-user systems. By modeling these systems as discrete-event systems, we explored how individual agents, with limited local information, navigate complex tasks involving both private communications and public broadcasting. Our focus was on addressing five critical information control challenges, from minimizing information exchange to ensure privacy, to maximizing publicly release to ensure transparency.

Our findings emphasize the need for nuanced protocols in decentralized systems where agents must balance security and transparency concerns. The proposed strategies enable users to control the release of information selectively—either to assist friends in achieving shared goals or to prevent adversaries from reaching their goals. These approaches go beyond traditional information theory by emphasizing not only reliable communication but also the optimal regulation of information flows based on specific objectives.

The contributions of this work lay the groundwork for more secure, efficient, and transparent multi-agent systems. As networked systems grow in complexity and applicability, this framework provides a structured way to manage information exchange, ultimately supporting more robust and reliable multi-user collaborations.

References

[1] C. Arndt, *Information Measures: Information and Its Description in Science and Engineering*. New York, NY, USA: Springer, 2003.
[2] R. McEliece, *The Theory of Information and Coding*, 2nd ed., Cambridge, U.K.: Cambridge Univ. Press, 2009.
[3] F. M. Reza, *An Introduction to Information Theory*. New York, NY, USA: Dover, 1994.
[4] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul. 1948.
[5] R. W. Yeung, *A First Course in Information Theory*. New York, NY, USA: Springer, 2002.
[6] S. Shu, F. Lin, and H. Ying, "Detectability of discrete event systems," *IEEE Trans. Autom. Control*, vol. 52, no. 12, pp. 2356–2359, Dec. 2007.
[7] C. Keroglou and C. N. Hadjicostis, "Detectability in stochastic discrete event systems," *Syst. Control Lett.*, vol. 84, pp. 21–26, Oct. 2015.
[8] Y. Liu, Z. Liu, X. Yin, and S. Li, "An improved approach for verifying delayed detectability of discrete-event systems," *Automatica*, vol. 124, Feb. 2021, Art. no. 109291.
[9] P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes," *SIAM J. Control Optim.*, vol. 25, no. 1, pp. 206–230, Jan. 1987.
[10] F. Lin and W. M. Wonham, "On observability of discrete-event systems," *Inf. Sci.*, vol. 44, no. 3, pp. 173–198, Apr. 1988.
[11] K. Rudie and W. M. Wonham, "Think globally, act locally: Decentralized supervisory control," *IEEE Trans. Autom. Control*, vol. 37, no. 11, pp. 1692–1708, Nov. 1992.
[12] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Diagnosability of discrete-event systems," *IEEE Trans. Autom. Control*, vol. 40, no. 9, pp. 1555–1575, Sep. 1995.
[13] F. Lin, "Diagnosability of discrete event systems and its applications," *Discrete Event Dyn. Syst., Theory Appl.*, vol. 4, no. 2, pp. 197–212, May 1994.
[14] L. K. Carvalho, M. V. Moreira, and J. C. Basilio, "Generalized robust diagnosability of discrete event systems," *IFAC Proc. Volumes*, vol. 44, no. 1, pp. 8737–8742, Jan. 2011.
[15] M. P. Cabasino, A. Giua, and C. Seatzu, "Diagnosability of discrete-event systems using labeled Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 1, pp. 144–153, Jan. 2014.
[16] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, 3rd ed., New York, NY, USA: Springer, 2021.
[17] K. Rudie, S. Lafortune, and F. Lin, "Minimal communication in a distributed discrete-event system," *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 957–975, Jun. 2003.
[18] W. Wang, S. Lafortune, and F. Lin, "Minimization of communication of event occurrences in acyclic discrete event systems," *IEEE Trans. Autom. Control*, vol. 53, no. 9, pp. 2197–2202, Oct. 2008.
[19] X. Yin and S. Lafortune, "A general approach for optimizing dynamic sensor activation for discrete event systems," *Automatica*, vol. 105, pp. 376–383, Jul. 2019.
[20] D. Sears and K. Rudie, "Minimal sensor activation and minimal communication in discrete-event systems," *Discrete Event Dyn. Syst.*, vol. 26, no. 2, pp. 295–349, Jun. 2016.
[21] B. Behinaein, F. Lin, and K. Rudie, "Optimal information release for mixed opacity in discrete-event systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 4, pp. 1960–1970, Oct. 2019.
[22] W. M. Wonham and K. Cai, *Supervisory Control of Discrete-Event Systems*. New York, NY, USA: Springer, 2019.
[23] S. Shu and F. Lin, "Detectability of discrete event systems with dynamic event observation," *Syst. Control Lett.*, vol. 59, no. 1, pp. 9–17, Jan. 2010.
[24] F. Wang and F. Lin, "Information control in networked discrete event systems," *Inf. Sci.*, vol. 669, pp. 1206–1217, May 2024.

Fei Wang (Member, IEEE) received the M.Sc. degree in control science and engineering from Shanghai University, Shanghai, China, in 2010, and the Ph.D. degree in control science and engineering from Tongji University, Shanghai, in 2017. From 2011 to 2012, he was a Visiting Scholar with Wayne State University, Detroit, MI, USA. Since 2018, he has been with the School of Electrical and Electronic Engineering, Shanghai Institute of Technology, Shanghai. His research interests include discrete event systems, robust control, and bioinformatics, with applications in automated manufacturing, robotics, and genomic sequence analysis.

Feng Lin (Life Fellow, IEEE) received the B.Eng. degree in electrical engineering from Shanghai Jiao Tong University, Shanghai, China, in 1982, and the M.A.Sc. and Ph.D. degrees in electrical engineering from the University of Toronto, Toronto, ON, Canada, in 1984 and 1988, respectively. He was a Post-Doctoral Fellow with Harvard University, Cambridge, MA, USA, from 1987 to 1988. Since 1988, he has been with the Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI, USA, where he is currently a Professor. He authored a book entitled *Robust Control Design: An Optimal Control Approach* and co-authored an article that received a George Axel by Outstanding Paper Award from the IEEE Control Systems Society. His current research interests include discrete event systems, hybrid systems, neural networks, robust control, and their applications in alternative energy, biomedical systems, machine learning, and automotive control.

Jun Chen (Senior Member, IEEE) received the bachelor's degree in automation from Zhejiang University, Hangzhou, China, in 2009, and the Ph.D. degree in electrical engineering from Iowa State University, Ames, IA, USA, in 2014.

He was with Idaho National Laboratory from 2014 to 2016 and also with General Motors from 2017 to 2020. In 2020, he joined Oakland University, where he is currently an Associate Professor with the ECE Department. His research interests include advanced control and optimization, model predictive control, artificial intelligence, and stochastic hybrid systems, with applications in intelligent vehicles, robotics, and energy systems. He was a recipient of the NSF CAREER Award, the Best Paper Award from IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, the Best Paper Award from the IEEE International Conference on Electro Information Technology, the Best Paper Award from the IEEE Cyber Awareness and Research Symposium, the New Investigator Research Excellence Award and Outstanding Graduate Mentor Award from Oakland University, the Publication Achievement Award from Idaho National Laboratory, the Research Excellence Award from Iowa State University, and the Outstanding Student Award from Zhejiang University.