Information Control in Networked Discrete Event Systems☆

Fei Wang[a], Feng Lin[b,*]

[a] *School of Electrical and Electronic Engineering, Shanghai Institute of Technology, Shanghai 201416, China.*
[b] *Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI 48202, USA.*

Abstract

How to control information exchange among different users is an important problem in networked systems with many users/agents. Generally speaking, there are several considerations in control of information exchange in a networked system, including (1) to ensure a friend user has sufficient information to perform its tasks, (2) to deprive an adversary user its information to perform its tasks, (3) to minimize information exchange among friend users so that the risk of information leaking is minimized, and (4) to maximize information broadcasted to all users to achieve maximum transparency. In this paper, we investigate the information control problems in the framework of discrete event systems. Based on the problem at hand, we divide users in a networked system into two or more groups. Users in the same group are consider as friends and users in a different group are consider

☆This work is supported in part by the National Natural Science Foundation of China under Grant 62073242 and by the National Science Foundation of USA under grant 2146615.

*Corresponding author. Address: Wayne State University, 5050 Anthony Wayne Dr. Detroit, MI 48202, USA. Tel.: +1 313-577-3428.

Email addresses: feiwang@sit.edu.cn (Fei Wang), flin@wayne.edu (Feng Lin)

as adversaries. Several information control problems are investigated and solved using a systematic and rigorous approach. Methods are developed to design controllers that send minimum information to its friends to help them to perform their tasks and broadcast maximum information without helping its adversaries.
Keywords: Networked systems, discrete event systems, multi-agent systems, information flow, information control

---

1. Introduction

The Internet revolution has led to information explosion. Wide use of Internet and cyberspace has made information that was previously difficult to obtain now readily available. This information revolution has greatly improved productivity, enriched people's life, and brought the world closer. While the information revolution has many significant positive impacts on the society, it also has some negative impacts, especially in terms of information security and information abuses. To enhance positive impacts and to reduce negative impacts of information explosion, it is important to control information flow in cyberspace.

Intuitively, the information control problem to be investigated in this paper can be briefly described as follows. There are many users/agents in a networked system, each has its own goals. To reach its goals, one user needs information from other users. At the same time, each user can control its own information by deciding whether or not to exchange its information. There are two ways to exchange information: (1) communicating the information to other users privately (say, via encrypted messaging), or (2) broadcasting the information to all users publicly. For reasons of security, some users may want to communicate as little information as possible. For reasons of transparency, some users may be required to broadcast as much information as possible. Therefore, the questions related to information control include the following. (1) What information shall one user communicate to others? (2) What information shall a user broadcast? (3) How to minimize information communicated? (4) How to maximize information broadcasted? We plan to develop a systematic approach to answer these and other questions in the framework of discrete event systems.

The traditional information theory [1, 12, 14, 18, 21] focuses on issues related to reliable and efficient communication, such as channel coding, data compression, and information encryption. The issues addressed in this paper are based on reliable communication, that is, we assume that the communication is reliable. We focus on the optimal control of information release. Specifically, we will investigate the mechanism of information release, as well as the concepts of minimum and maximum release. The traditional information theory cannot be directly applied to address these important issues related to information release and information control.

For example, consider the discrete event system shown in Fig. 1(a). Assume that two users, a boss and his/her subordinate, know the system model. For the subordinate to perform his/her task, he/she needs to know if the system is in state 2 or not (for example, the subordinate needs to call an ambulance if the patient described by the system in Fig. 1(a) is in State 2, but there are no needs to call an ambulance if the patient is in States 1, 3, and 4). The boss wants to communicate the occurrences of some events to the subordinate so that he/she knows whether the system is in State 2 or not. The question is: which event shall the boss communicate to the subordinate? Shall it be α? β? γ? or some combinations of them? In this example, the answer is α, because if the number of occurrences of α is an odd number, then the system is in State 2 and if the number of occurrences of α is an even number, then the system is in States 1, 3, or 4.

Since this example is simple, the answer is unique and can be obtained by intuition. If the system is complex, consisting of hundreds of states and many events, or the problem is not to identify one state, but rather, to distinguish one subset of states from another subset of states, then the answer will not be unique and intuitive. For example, consider the system shown in Fig. 1(b). If a user needs to distinguish states 0 and 5 from states 2 and 4, then the answer to the question of which events shall be communicated to him/her is not as intuitive and straightforward as the answer to the system in Fig. 1(a). Hence, a systematic approach that can be implemented using computers is highly desirable.

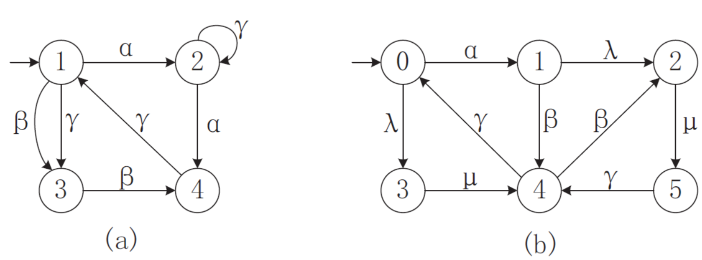


Figure 1: Information control using discrete event system model.

To make the approach general, we model a user's task as to distinguish certain pairs of states. Specifying a task as distinguishing certain pairs of states is very general and most common tasks can be specified in this way [20, 8, 11]. For example, in supervisory control, a supervisor needs to distinguish legal states from illegal states [13, 10, 16]. In diagnosis, a diagnoser needs to distinguish normal states from fault states [17, 9, 4, 3]

If a user can distinguish these pairs of states based on information directly available to the user, then it does not need any information from other users. This will be a trivial case. In general, a user needs information communicated or broadcasted by other users in order to perform its task. A user can also control/release its information by deciding what information to communicate or broadcast to other users. Its information control objective may include one or two or more of the the following. (1) Help some users to perform their tasks. (2) Prevent some other users from performing their tasks. (3) Minimize information communicated to other users for security reasons. (4) Maximize information broadcasted to others to ensure transparency.

The information directly available to a user is the occurrences of some events local to the user. We call these events (locally) observable events of the user. A user can decide how to communicate or broadcast the occurrences of its observable events. The total information available to a user is its own observation of (locally) observable events, event occurrences communicated by other users and event occurrences broadcasted by other users. Since each user has only partial information of the system, no user knows the exact state of the system. Based on the information available, a user can calculate the set of all possible states the system may be in. We call this set "state estimate". Suppose that a user's goal is to distinguish a set of states Q_1 from another set of states Q_2. If its state estimate contains states in Q_1 but no states in Q_2, or contains states in Q_2 but no states in Q_1, then its goal can be reached.

The information control problem is challenging when the system is complex with many users of different goals. We assume that users are divided into two or more groups: users in the same group are friends and users in a different group are adversaries. The division of groups depends on the problems to be solved and is problem specific. For notational simplicity, we consider two groups. It is not difficult to extend the results of the paper from two groups to several groups. Suppose that the initial control objectives are: (1) to help friend users to reach their goals, (2) to prevent adversary users from reaching their goals. Then the initial control strategy is to communicate all information to friends and not to communicate anything to adversaries, and not to broadcast anything. If a user's goal can be reached under this initial control, then nothing his adversaries can do to prevent him from reach his goal. If a user's goal cannot be reached under this initial control, then nothing his friends can to to help him to reach his goal. From this initial control, we will investigate how to further improve the control based on additional requirements as follows.

For privacy, security, and other reasons, it is often required that the communication among users be minimized. We can improve the initial control by requiring minimal communication among the friends without jeopardizing the goals that can be reached by the friends under the initial control. Intuitively, what we can do is removing some events from communication. Hence the information available to some users are reduced. This will change the state estimates E of these users. Generally speaking, this will make the state estimates E bigger (that is, less certain). If removing an event σ enlarges E of a friend user to the point that it contains both states in Q_1 and Q_2, then the goal of the user can no longer be reached. Hence σ must be communicated. Otherwise, σ can be removed from communication. Minimal communication is achieved when no more events can be removed. Depending on the order of events being examined, the minimal communication is not unique. Minimizing communication in distributed discrete-event systems has been investigated in the literature [15, 22]. This paper extends the existing results to multiple users with different grouping in a systematic and comprehensive way. Minimizing information diffusion is a topic also discussed in the context of continuous-time diffusion networks [7]. However, our approach is different and uses the framework of discrete event systems.

For some users such as government agencies, it is required that they release (broadcast) as much information as possible[1]. We studied the maximum information release problem for single user in [2]. We extend this to multiple users in this paper. Again, we start with the initial control described early. We can improve the initial control by requiring some users to maximize the information broadcasted without helping their adversaries to reach goals that cannot be reached under the initial control. The maximizing privacy is discussed in continuous-time diffusion networks [6], which is different than our approach.

[1]For example, in USA, the Freedom of Information Act (FOIA) requires that certain information and records of government agencies to be released to the public upon request, unless such release will harm national security or be covered under other nine specific exemptions.

Compared with the results in the literature, the novelty and contributions of this paper are as follows. (1) We consider multiple users with multiple groups. Some users are friends, and some other users are adversaries. (2) We consider both private communications among users and public broadcasting to all users. (3) We systematically investigate five information control problems and provide solutions to the problems. These problems capture the essence of information exchange, security, and transparency in large networked systems.

This paper is organized as follows. In Section 2, we introduce our model of networked systems, which is a discrete event system built from its components. In Section 3, two mechanisms of information exchange among different users are proposed: private communication and public broadcasting. State estimates for all users are introduced and a procedure is proposed to obtain them. In Section 4, controllers are introduced to control information flow in a networked system. The task of a user is specified as a set of state pairs that the user needs to distinguish based on its own local observation and communication from its friends and broadcasting from other users. Necessary and sufficient condition is derived for a user to perform its task. Five information control problems are then solved. In Section 5, an illustrative example of a distribution system is given to illustrate the results of the paper.

2. Networked Systems

We model networked systems as discrete event systems. The reasons for using discrete event system model are as follows. (1) The model is general and flexible. Most networked systems can be modeled as discrete event systems at some level of abstraction. (2) It allows us to build a networked system model in a modular way where components are modeled by small automata and then combined using parallel composition (automatically using computers). (3) It can describe system properties and information flows very well. We use automaton (also called finite state machine) to model a discrete event system [13, 10, 5]:

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

where Q is the set of finite states; Σ is the set of finite events; q_o is the initial state; and $\delta : Q \times \Sigma \to Q$ is the transition function which describes the dynamics of the system. The transition function is extended to $\delta : Q \times \Sigma^* \to Q$ in the usual way [5].

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

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

One advantage of using automaton G is that the model can be built in a modular way: Each component of a networked system can be modeled by a small automaton G_i. Then the model for the overall system can be obtained using the parallel composition [5]:

$$G = G_1 || G_2 || ... || G_M.$$

Flexibility and scalability are important in modeling networked systems, as components in a networked systems change frequently. Our model is flexible and scalable.

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

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

where ε is the empty string. In other words, if a string of events $s = \sigma_1\sigma_2...\sigma_k \in L(G)$ occurred in the networked system G, User U_i will directly observe $w = P_i(s)$. In the paper, we assume that User U_i communicates to other users based on its own local observation $w \in P_i(L(G))$, where $P_i(L(G))$ is the projection of $L(G)$, representing all possible local observations by User U_i.

3. Information Exchanges among Users

We assume that the information contents to be exchanged/released are occurrences of (locally observed) events. We investigate two types of information flows/exchanges among users: (1) Private communication from User U_i to User U_j and (2) Public broadcasting by User U_i.

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

$$\theta_{ij} : P_i(L(G)) \to 2^{\Sigma_{o,i}}.$$

In other words, if the current local observation of User U_i is $w \in P_i(L(G))$, then if any event $\sigma \in \theta_{ij}(w)$ occurs, User U_i will let User U_j know, that is, User U_i will communicate this information to User U_j.

Without loss of generality, we use state-base mapping in the rest of the paper. In other words, we assume that there exists a deterministic automaton

$$H_i = (X_i, \Sigma_{o,i}, \xi_i, x_{i,o})$$

with $P_i(L(G)) \subseteq L(H_i)$ and a mapping

$$\vartheta_{ij} : X_i \rightarrow 2^{\Sigma_{o,i}}$$

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

$$\theta_{ij}(w) = \vartheta_{ij}(\xi_i(x_{i,o}, w)).$$

We denote this state-based mapping by $\theta_{ij} = (H_i, \vartheta_{ij})$. Note that θ_{ij} has two subscripts, where i is the user sending the communication and j is the user receiving the communication. θ_{ij} also specifies who can communicate with whom. If $\theta_{ij}(w) = \emptyset$ for all $w \in P_i(L(G))$, then user U_i cannot communicate with user U_j.

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

$$\phi_i : P_i(L(G)) \rightarrow 2^{\Sigma_{o,i}}.$$

In other words, if the current local observation of User U_i is $w \in P_i(L(G))$, then if any event $\sigma \in \phi_i(w)$ occurs, User U_i will let all users know, that is, User U_i will broadcast this information.

We again use state-base mapping based on H_i in the rest of the paper and assume that there exists a mapping

$$\varphi_i : X_i \rightarrow 2^{\Sigma_{o,i}}$$

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

$$\phi_i(w) = \varphi_i(\xi_i(x_{i,o}, w)).$$

We denote this state-based mapping by $\phi_i = (H_i, \varphi_i)$. Note that ϕ_i has only one subscript, where i is the user sending the communication. There is no need to specify which user receives the communication as it is broadcasted to all users. Note further that we use the same H_i for both θ_{ij} and ϕ_i without loss of generality, because we can always refine the state space X_i to make it suitable for both θ_{ij} and ϕ_i.

Therefore, information (that is, occurrences of events) received by user U_j is given by

$$\rho_j = P_j \cup \phi_1 \cup ... \cup \phi_N \cup \theta_{1j} \cup ... \cup \theta_{Nj}. \tag{1}$$

where the union $\cup$ is interpreted as follows. User U_j knows the occurrence of an event if (1) it is observed by itself; (2) it is broadcasted by some User U_i; or (3) it is communicated to User U_j by some User U_i. Hence, if a string of events $s \in L(G)$ occurred in the networked system G, User U_j will see $w = \rho_j(s)$. Formally, $\rho_j(s)$ is defined recursively as

$$\rho_j(\varepsilon) = \varepsilon, \quad \rho_j(s\sigma) = \begin{cases} \rho_j(s)\sigma & \text{if } \sigma \in \Sigma_{o,j} \cup \phi_1(P_1(s)) \cup ... \cup \ \phi_N(P_N(s)) \\ & \cup\ \theta_{1j}(P_1(s)) \cup ... \cup \ \theta_{Nj}(P_N(s)) \\ \rho_j(s) & \text{otherwise} \end{cases}.$$

Given an information mapping $\rho_j : L(G) \rightarrow \Sigma^*$, define state estimate of User U_j after observing a string $w \in \rho_j(L(G))$ as the set of all possible states G may be in from the view point of U_j:

$$E^{\rho_j}(w) = \{q \in Q : (\exists s \in L(G))\rho_j(s) = w \wedge \delta(q_0, s) = q\}.$$

We propose the following procedure to calculate state estimate $E^{\rho_j}(w)$.

Step 1. Take the parallel composition of G and H_i, $i = 1, 2, ..., N$:

$$\begin{aligned}\tilde{G} &= (\tilde{Q}, \Sigma, \tilde{\delta}, \tilde{q}_o) = G||H_1||...||H_N \\ &= (Q \times X_1 \times ... \times X_N, \Sigma, \tilde{\delta}, (q_o, x_{1,o}, ..., x_{N,o})).\end{aligned}$$

Since $P_i(L(G)) \subseteq L(H_i)$, it is clear that

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

Step 2. For each User U_j, $j = 1, 2, ..., N$, replace the transitions in $\tilde{G}$ that cannot be observed by U_j with ε-transitions to obtain

$$\tilde{G}_\varepsilon^j = (\tilde{Q}, \Sigma, \tilde{\delta}_\varepsilon^j, \tilde{q}_o),$$

where $\tilde{\delta}_\varepsilon^j$ is defined as follows. With a slight abuse of notation, denote the set of all transitions of $\tilde{G}$ also by $\tilde{\delta}$, that is, $\tilde{\delta} = \{(\tilde{q}, \sigma, \tilde{\delta}(\tilde{q}, \sigma)) : \tilde{q} \in \tilde{Q} \wedge \sigma \in \Sigma \wedge \tilde{\delta}(\tilde{q}, \sigma))!\}$. For $\tilde{q} = (q, x_1, ..., x_N)$ and $\sigma \in \Sigma$, transition $(\tilde{q}, \sigma, \tilde{\delta}(\tilde{q}, \sigma))$ is replaced with ε-transition $(\tilde{q}, \varepsilon, \tilde{\delta}(\tilde{q}, \sigma))$ if σ cannot be observed by User U_j, that is, $\sigma \notin \Sigma_{o,j} \cup (\cup_{i=1}^N \vartheta_{ij}(x_i)) \cup (\cup_{i=1}^N \varphi_i(x_i))$. In other words,

$$\begin{aligned}\tilde{\delta}_\varepsilon^j =& \{(\tilde{q}, \sigma, \tilde{\delta}(\tilde{q}, \sigma)) = ((q, x_1, ..., x_N), \sigma, \tilde{\delta}(\tilde{q}, \sigma)) : \\ &((q, x_1, ..., x_N), \sigma, \tilde{\delta}(\tilde{q}, \sigma)) \in \tilde{\delta} \wedge \sigma \in \Sigma_{o,j} \cup (\cup_{i=1}^N \vartheta_{ij}(x_i)) \cup (\cup_{i=1}^N \varphi_i(x_i))\} \\ &\cup \{(\tilde{q}, \varepsilon, \tilde{\delta}(\tilde{q}, \sigma)) = ((q, x_1, ..., x_N), \varepsilon, \tilde{\delta}(\tilde{q}, \sigma)) : \\ &((q, x_1, ..., x_N), \sigma, \tilde{\delta}(\tilde{q}, \sigma)) \in \tilde{\delta} \wedge \sigma \notin \Sigma_{o,j} \cup (\cup_{i=1}^N \vartheta_{ij}(x_i)) \cup (\cup_{i=1}^N \varphi_i(x_i))\}.\end{aligned}$$

Note that $\tilde{G}_\varepsilon^j$ is a nondeterministic automaton. By the above definition of $\tilde{\delta}_\varepsilon^j$, it is clear that

$$L(\tilde{G}_\varepsilon^j) = \rho_j(L(\tilde{G})) = \rho_j(L(G)).$$

Step 3. For each User U_j, $j = 1, 2, ..., N$, convert $\tilde{G}_{\varepsilon}^{j}$ to a deterministic automaton $\tilde{G}_{obs}^{j}$, called observer, as

$$\tilde{G}_{obs}^{j} = (Y_j, \Sigma, \zeta_j, y_{j,o}) = Ac(2^{\tilde{Q}}, \Sigma, \zeta_j, UR(\{\tilde{q}_o\})),$$

where $Ac(.)$ denotes the accessible part; $UR(.)$ is the unobservable reach defined, for $y \subseteq \tilde{Q}$, as

$$UR(y) = \{\tilde{q} \in \tilde{Q} : (\exists \tilde{q}' \in y)\tilde{q} \in \tilde{\delta}_{\varepsilon}^{j}(\tilde{q}', \varepsilon)\}.$$

The transition function ζ_j is defined, for $y \in Y_j$ and $\sigma \in \Sigma$, as

$$\zeta_j(y, \sigma) = UR(\{\tilde{q} \in \tilde{Q} : (\exists \tilde{q}' \in y)\tilde{q} \in \tilde{\delta}_{\varepsilon}^{j}(\tilde{q}', \sigma)\}).$$

It is well-known (see, for example, in [5]) that (1) $L(\tilde{G}_{obs}^{j}) = L(\tilde{G}_{\varepsilon}^{j}) = \rho_j(L(G))$ and (2) for all $w \in \rho_j(L(G)) = L(\tilde{G}_{obs}^{j})$,

$$\zeta_j(y_{j,o}, w) = \{\tilde{q} \in \tilde{Q} : (\exists s \in L(\tilde{G}))\rho_j(s) = w \wedge \tilde{\delta}(\tilde{q}_0, s) = \tilde{q}\}.$$

Define

$$\tilde{E}^{\rho_j}(w) = \{\tilde{q} \in \tilde{Q} : (\exists s \in L(\tilde{G}))\rho_j(s) = w \wedge \tilde{\delta}(\tilde{q}_0, s) = \tilde{q}\},$$

then $\zeta_j(y_{j,o}, w) = \tilde{E}^{\rho_j}(w)$.

For any $y \in Y_j$ (hence $y \subseteq Q$), define

$$y|_Q = \{q \in Q : (\exists \tilde{q} \in y)\tilde{q} = (q, x_1, ..., x_N)\}.$$

In particular,

$$\tilde{E}^{\rho_j}(w)|_Q = \{q \in Q : (\exists \tilde{q} \in \tilde{E}^{\rho_j}(w))\tilde{q} = (q, x_1, ..., x_N)\}.$$

We have the following theorem.

Theorem 1. *The state estimate of User U_j after observing a string $w \in \rho_j(L(G))$ is given by*

$$E^{\rho_j}(w) = \tilde{E}^{\rho_j}(w)|_Q = \zeta_j(y_{j,o}, w)|_Q. \tag{2}$$

Proof:

By the definitions,

$$\begin{aligned}\tilde{E}^{\rho_j}(w)|_Q =& \{q \in Q : (\exists \tilde{q} \in \tilde{E}^{\rho_j}(w))\tilde{q} = (q, x_1, ..., x_N)\} \\ =& \{q \in Q : (\exists s \in L(\tilde{G}))\rho_j(s) = w \wedge \tilde{\delta}(\tilde{q}_0, s) = (q, x_1, ..., x_N)\} \\ & \text{(by the definition of } \tilde{E}^{\rho_j}(w)) \\ =& \{q \in Q : (\exists s \in L(G))\rho_j(s) = w \wedge \delta(q_0, s) = q\} \\ & \text{(because } L(G) = L(\tilde{G})) \\ =& E^{\rho_j}(w).\end{aligned}$$

■

4. Information Control

How to control information communicated or broadcasted is the key to information control in networked systems. Information communicated or broadcasted by User $U_i, i = 1, 2, ..., N$ is controlled by a controller π_i, which determines φ_i and ϑ_{ij}. In other words,

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

We investigate how to design $\pi = (\pi_1, ..., \pi_N)$. Intuitively, if User U_i wants to help User U_j to perform its tasks, then User U_i shall send its observation to User U_j. User U_i may want to minimize the information it sends to User U_j while still helps User U_j to perform its task. If User U_i wants to prevent User U_j from performing its tasks, then User U_i shall not send information to User U_j and shall avoid broadcasting information that may help User U_j to perform its tasks.

We assume that in order for User $U_j, j = 1, 2, ..., N$ 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 User U_j. We say that User U_j can perform its task if it can always distinguish all state pairs in T^j_{spec}, that is, for all $w \in \rho_j(L(G))$,

$$(E^{\rho_j}(w) \times E^{\rho_j}(w)) \cap T^j_{spec} = \emptyset.$$

Remark 1. *Specifying a task using T_{spec} is very general and most common tasks can be specified in this way. The following examples show that tasks in supervisory control, diagnosability, and detectability can all be specified by T_{spec}. In supervisory control, a common task is to prevent a system from entering some illegal/unsafe states. In order to do so, a supervisor needs to distinguish legal states $Q_l \subseteq Q$ from illegal states $Q_{il} \subseteq Q$. Hence, $T_{spec} = (Q_l \times Q_{il}) \cup (Q_{il} \times Q_l)$. For diagnosability, a diagnoser needs to distinguish normal states $Q_n \subseteq Q$ from fault states $Q_f \subseteq Q$. Hence, $T_{spec} = (Q_n \times Q_f) \cup (Q_f \times Q_n)$. The goal of detectability is also specified by T_{spec} [19].*

We have the following theorem.

Theorem 2. *User $U_j, j = 1, 2, ..., N$ can perform its task if and only if in the observer $\tilde{G}^j_{obs}$,*

$$(\forall y \in Y_j)(y|_Q \times y|_Q) \cap T^j_{spec} = \emptyset. \tag{3}$$

Proof:

We need to prove

$$\begin{aligned} &(\forall w \in \rho_j(L(G)))(E^{\rho_j}(w) \times E^{\rho_j}(w)) \cap T^j_{spec} = \emptyset \\ \Leftrightarrow &(\forall y \in Y_j)(y|_Q \times y|_Q) \cap T^j_{spec} = \emptyset. \end{aligned}$$

Or, equivalently,

$$\begin{aligned} &(\exists w \in \rho_j(L(G)))(E^{\rho_j}(w) \times E^{\rho_j}(w)) \cap T^j_{spec} \neq \emptyset \\ \Leftrightarrow &(\exists y \in Y_j)(y|_Q \times y|_Q) \cap T^j_{spec} \neq \emptyset. \end{aligned}$$

($\Rightarrow$): If $(\exists w \in \rho_j(L(G)))(E^{\rho_j}(w) \times E^{\rho_j}(w)) \cap T^j_{spec} \neq \emptyset$ is true, then let $y = \zeta_j(y_{j,o}, w)$. By Theorem 1, $E^{\rho_j}(w) = y|_Q$. Therefore, $(\exists y \in Y_j)(y|_Q \times y|_Q) \cap T^j_{spec} \neq \emptyset$.

($\Leftarrow$): If $(\exists y \in Y_j)(y|_Q \times y|_Q) \cap T^j_{spec} \neq \emptyset$ is true, then let w be any string from $y_{j,o}$ to y, that is, $y = \zeta_j(y_{j,o}, w)$. By Theorem 1, $E^{\rho_j}(w) = y|_Q$. Therefore, $(\exists w \in \rho_j(L(G)))(E^{\rho_j}(w) \times E^{\rho_j}(w)) \cap T^j_{spec} \neq \emptyset$.

■

We assume that users are divided into two groups:

$$Group\ 1 = \{1, ..., N_1\}, \quad Group\ 2 = \{N_1 + 1, ..., N\}.$$

Users in the same group are friends and users in the other group are adversaries. We investigate the following information control problems.

Information Control Problem 1

The first problem that we investigate is: Can User U_j perform its task based on its own local observation without information from other users, including its friends?

To solve this problem, we let $\rho_j = P_j$ and check if the condition of Theorem 2 is satisfied or not. Note that, since $\rho_j = P_j$, the procedure is simpler than outlined in the previous section. In fact, we do not need to take the parallel composition $\tilde{G} = G||H_1||...||H_N$. We can simply let $\tilde{G} = G$ and construct the observer of G with respect to P_j. If the condition of Theorem 2 is satisfied, then User U_j can perform its task based on its own local observation without information from other users.

Information Control Problem 2

If the answer to the first problem is "no", then User U_j needs helps from other users. Hence, we investigate the second problem: Can User U_j perform its task based on its own local observation and all its friends' observation? In other words, assume that all its friends will communicate all information to User U_j, can User U_j perform its task?

Without loss of generality, let $U_j = U_1$. If all its friends communicate all information to User U_1, then $\rho_1 = P_1 \cup P_2 \cup ... \cup P_{N_1}$. To solve the second problem, again, there is no need to take the parallel composition and we can let $\tilde{G} = G$. We construct the observer of G with respect to $P_1 \cup P_2 \cup ... \cup P_{N_1}$ and check if the condition of Theorem 2 is satisfied or not. If it is satisfied, then User U_1 can perform its task based on its own local observation and all its friends' observation.

Information Control Problem 3

If the answer to the second problem is "yes", then the third problem is how to minimize communications from its friends to User U_1.

To minimize the communication, we proceed as follows. We partition the transitions in $\tilde{G}$ into three groups: (1) transitions belonging to $\Sigma_{o,1}$ (observable by U_1 itself), (2) transitions belonging to $\Sigma_{o,2} \cup ... \cup \Sigma_{o,N_1} - \Sigma_{o,1}$ (observable by its friends), and (3) other transitions. In other words, $\tilde{\delta} = \tilde{\delta}_1^1 \cup \tilde{\delta}_2^1 \cup \tilde{\delta}_3^1$ with

$$\begin{aligned}
\tilde{\delta}_1^1 &= \{(\tilde{q}, \sigma, \tilde{\delta}(\tilde{q}, \sigma)) \in \tilde{\delta} : \sigma \in \Sigma_{o,1}\} \\
\tilde{\delta}_2^1 &= \{(\tilde{q}, \sigma, \tilde{\delta}(\tilde{q}, \sigma)) \in \tilde{\delta} : \sigma \in \Sigma_{o,2} \cup ... \cup \Sigma_{o,N_1} - \Sigma_{o,1}\} \\
\tilde{\delta}_3^1 &= \{(\tilde{q}, \sigma, \tilde{\delta}(\tilde{q}, \sigma)) \in \tilde{\delta} : \sigma \in \Sigma - \Sigma_{o,1} \cup ... \cup \Sigma_{o,N_1}\}.
\end{aligned} \tag{4}$$

Since the answer to the second problem is "yes", we know that by replacing all transitions in $\tilde{\delta}_3^1$ by ε-transitions, the resulting observer of U_1 satisfies the condition of Theorem 2. Transitions in $\tilde{\delta}_2^1$ require communications from Users $U_i, i = 2, ..., N_1$. To minimize such communications, let us find a minimum set $\tilde{\delta}_{2,min}^1 \subseteq \tilde{\delta}_2^1$ under which the resulting observer of U_1 satisfies the condition of Theorem 2 using the following algorithm.

Algorithm 1. *Calculation of a minimum set* $\tilde{\delta}_{2,min}^1 \subseteq \tilde{\delta}_2^1$

Input: $\tilde{G}$

Output: $\tilde{\delta}_{2,min}^1$

1: *Partition the transitions in* $\tilde{G}$ *as* $\tilde{\delta} = \tilde{\delta}_1^1 \cup \tilde{\delta}_2^1 \cup \tilde{\delta}_3^1$*;*

2: Initially, let

$$\tilde{\delta}_o^1 = \tilde{\delta}_1^1 \cup \tilde{\delta}_2^1,\ \tilde{\delta}_{uo}^1 = \tilde{\delta}_3^1;$$

3: For all $(\tilde{q}, \sigma, \tilde{\delta}(\tilde{q}, \sigma)) \in \tilde{\delta}_2^1$ *do*

$$\tilde{\delta}_o^1 = \tilde{\delta}_o^1 - \{(\tilde{q}, \sigma, \tilde{\delta}(\tilde{q}, \sigma))\};$$

$$\tilde{\delta}_{uo}^1 = \tilde{\delta}_{uo}^1 \cup \{(\tilde{q}, \sigma, \tilde{\delta}(\tilde{q}, \sigma))\};$$

$$\tilde{\delta}_\varepsilon^1 = \{(\tilde{q}, \sigma, \tilde{\delta}(\tilde{q}, \sigma)) \in \tilde{\delta} : (\tilde{q}, \sigma, \tilde{\delta}(\tilde{q}, \sigma)) \in \tilde{\delta}_o\}$$
$$\cup \{(\tilde{q}, \varepsilon, \tilde{\delta}(\tilde{q}, \sigma)) \in \tilde{\delta} : (\tilde{q}, \sigma, \tilde{\delta}(\tilde{q}, \sigma)) \in \tilde{\delta}_{uo}\};$$

$$\tilde{G}_\varepsilon^1 = (\tilde{Q}, \Sigma, \tilde{\delta}_\varepsilon^1, \tilde{q}_o);$$

$$\tilde{G}_{obs}^1 = (Y_1, \Sigma, \zeta_1, y_{1,o});$$

If $(\forall y \in Y_1)(y|_Q \times y|_Q) \cap T_{spec}^1 = \emptyset$ *is not true, then*

$$\tilde{\delta}_o^1 = \tilde{\delta}_o^1 \cup \{(\tilde{q}, \sigma, \tilde{\delta}(\tilde{q}, \sigma))\};$$

$$\tilde{\delta}_{uo}^1 = \tilde{\delta}_{uo}^1 - \{(\tilde{q}, \sigma, \tilde{\delta}(\tilde{q}, \sigma))\};$$

4: Let

$$\tilde{\delta}_{2,min}^1 = \tilde{\delta}_o^1 - \tilde{\delta}_1^1;$$

5: End.

To calculate a minimum set $\tilde{\delta}_{2,min}^1$, Algorithm 1 checks transitions in $\tilde{\delta}_2^1$ one by one to see if it is needed for User U_1 to perform its task. If it is not needed, it will be removed. Note that minimum set $\tilde{\delta}_{2,min}^1$ is not unique, depending on the order in which transitions in $\tilde{\delta}_2^1$ are checked. It is not difficult to see that the computational complexity of Algorithm 1 is determined by Step 3. In Step 3, constructing observer $\tilde{G}^1_{obs}$ has complexity $|\Sigma|\ |2^{\tilde{Q}}|$. Step 3 may be repeated at most $|\tilde{Q}|\ |\Sigma|$ times. Therefore, the computational complexity of Algorithm 1 is $O(|\tilde{Q}|\ |\Sigma|^2\ |2^{\tilde{Q}}|)$.

Clearly, transitions in the resulting $\tilde{\delta}^1_{2,min}$ need to be communicated to U_1 by one of $U_i, i = 2, ..., N_1$. In order for a transition

$$(\tilde{q}, \sigma, \tilde{\delta}(\tilde{q}, \sigma)) = ((q, x_1, ..., x_N), \sigma, \tilde{\delta}(\tilde{q}, \sigma)) \in \tilde{\delta}^1_{2,min}$$

to be communicated to U_1, it is requires that

$$\sigma \in (\cup_{i=2}^{N_1} \vartheta_{i1}(x_i)) \cup (\cup_{i=2}^{N_1} \varphi_i(x_i)).$$

Therefore, we need to find a set of minimum controls

$$\pi_i = (\varphi_i, \vartheta_{i1}, ..., \vartheta_{iN}),\ i = 2, ..., N_1.$$

satisfying

$$\begin{aligned}&(\forall (\tilde{q}, \sigma, \tilde{\delta}(\tilde{q}, \sigma)) = ((q, x_1, ..., x_N), \sigma, \tilde{\delta}(\tilde{q}, \sigma)) \in \tilde{\delta}^1_{2,min})\\ &\sigma \in (\cup_{i=2}^{N_1} \vartheta_{i1}(x_i)) \cup (\cup_{i=2}^{N_1} \varphi_i(x_i)).\end{aligned} \tag{5}$$

Obviously, set of minimum controls is not unique. The following algorithm finds one set of minimum controls.

Algorithm 2. *Calculation of a set of minimum controls* $\pi_i = (\varphi_i, \vartheta_{i1}, ..., \vartheta_{iN}),\ i = 2, ..., N_1$

Input: $\tilde{\delta}^1_{2,min}$

Output: $\pi_i = (\varphi_i, \vartheta_{i1},\ ..., \vartheta_{iN}),\ i = 2, ..., N_1$

1: *For* $i = 2, ..., N_1$ *and* $x_i \in X_i$ *do* $\varphi_i(x_i) = \emptyset$*;*

2: *For* $i = 2, ..., N_1$ *and* $x_i \in X_i$ *do* $\vartheta_{i1}(x_i) = \emptyset$*;*

3: *For* $((q, x_1, ..., x_N), \sigma, \tilde{\delta}(\tilde{q}, \sigma)) \in \tilde{\delta}^1_{2,min}$ *do*

If $\sigma \notin (\cup_{2=1}^{N_1} \vartheta_{i1}(x_i)) \cup (\cup_{i=2}^{N_1} \varphi_i(x_i))$ *is true, then*

$$\textit{pick } i = 2, ..., N_1 \textit{ such that } \sigma \in \Sigma_{o,i}; \vartheta_{i1}(x_i) = \vartheta_{i1}(x_i) \cup \{\sigma\};$$

4: *End.*

In Step 3, the choice of which $\vartheta_{i1}(x_i)$ to add σ is arbitrary, but can be made based in the other considerations such as sharing the communication burden among friends or choosing neighboring friends. Algorithm 2 has a computational complexity of $O(N\ |\tilde{Q}|\ |\Sigma|)$.

Information Control Problem 4

If the answer to the second problem is "no", then User U_1 cannot perform its task unless some adversaries make some mistakes and release information that they shall not release. Therefore, the problem is how an adversary can avoid making such mistakes. If an adversary, say User $U_j, j = N_1 + 1, ..., N$, 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, and it shall not broadcast any information to the public. On the other hand, if User U_j has obligation to release as much information as possible to the public, then the fourth problem is how to broadcast maximal information to the public without helping User U_1 to perform its task.

To maximize the broadcasting, we proceed as follows. We consider again the partition $\tilde{\delta} = \tilde{\delta}^1_1 \cup \tilde{\delta}^1_2 \cup \tilde{\delta}^1_3$. In order to maximize the broadcasting without helping U_1, we use the following algorithm to find a maximum set $\tilde{\delta}^1_{3,max} \subseteq \tilde{\delta}^1_3$ such that the condition of Theorem 2 is not satisfied if U_1 observes transitions in $\tilde{\delta}^1_1 \cup \tilde{\delta}^1_2 \cup \tilde{\delta}^1_{3,max}$.

Algorithm 3. *Calculation of a maximum set* $\tilde{\delta}^1_{3,max} \subseteq \tilde{\delta}^1_3$

Input: $\tilde{G}$

Output: $\tilde{\delta}^1_{3,max}$

1: *Partition the transitions in* $\tilde{G}$ *as* $\tilde{\delta} = \tilde{\delta}^1_1 \cup \tilde{\delta}^1_2 \cup \tilde{\delta}^1_3$*;*

2: *Initially, let*

$$\tilde{\delta}^1_o = \tilde{\delta}^1_1 \cup \tilde{\delta}^1_2,\ \tilde{\delta}^1_{uo} = \tilde{\delta}^1_3;$$

3: *For all* $(\tilde{q}, \sigma, \tilde{\delta}(\tilde{q}, \sigma)) \in \tilde{\delta}^1_3$ *do*

$$\tilde{\delta}^1_o = \tilde{\delta}^1_o \cup \{(\tilde{q}, \sigma, \tilde{\delta}(\tilde{q}, \sigma))\};$$

$$\tilde{\delta}^1_{uo} = \tilde{\delta}^1_{uo} - \{(\tilde{q}, \sigma, \tilde{\delta}(\tilde{q}, \sigma))\};$$

$$\tilde{\delta}^1_\varepsilon = \{(\tilde{q}, \sigma, \tilde{\delta}(\tilde{q}, \sigma)) \in \tilde{\delta} : (\tilde{q}, \sigma, \tilde{\delta}(\tilde{q}, \sigma)) \in \tilde{\delta}_o\}$$
$$\cup \{(\tilde{q}, \varepsilon, \tilde{\delta}(\tilde{q}, \sigma)) \in \tilde{\delta} : (\tilde{q}, \sigma, \tilde{\delta}(\tilde{q}, \sigma)) \in \tilde{\delta}_{uo}\};$$

$$\tilde{G}^1_\varepsilon = (\tilde{Q}, \Sigma, \tilde{\delta}^1_\varepsilon, \tilde{q}_o);$$

$$\tilde{G}^1_{obs} = (Y_1, \Sigma, \zeta_1, y_{1,o});$$

If $(\forall y \in Y_1)(y|_Q \times y|_Q) \cap T^1_{spec} = \emptyset$ *is true, then*

$$\tilde{\delta}^1_o = \tilde{\delta}^1_o - \{(\tilde{q}, \sigma, \tilde{\delta}(\tilde{q}, \sigma))\};$$

$$\tilde{\delta}^1_{uo} = \tilde{\delta}^1_{uo} \cup \{(\tilde{q}, \sigma, \tilde{\delta}(\tilde{q}, \sigma))\};$$

4: *Let*

$$\tilde{\delta}^1_{3,max} = \tilde{\delta}^1_o - (\tilde{\delta}^1_1 \cup \tilde{\delta}^1_2);$$

5: End.

To calculate a maximum set $\tilde{\delta}^1_{3,max}$, Algorithm 3 checks transitions in $\tilde{\delta}^1_3$ one by one to see if it will help User U_1 to perform its task. If it will help User U_1, then it will be removed. Note that maximum set $\tilde{\delta}^1_{3,max}$ is not unique, depending on the order in which transitions in $\tilde{\delta}^1_3$ is checked. User U_1 is able to observe transitions in $\tilde{\delta}^1_{3,max}$. Similar to Algorithm 1, the computational complexity of Algorithm 3 is $O(|\tilde{Q}|\ |\Sigma|^2\ |2^{\tilde{Q}}|)$.

Therefore, we need to find a set of maximum controls (on broadcasting)

$$\varphi_j,\ j = N_1 + 1, ..., N$$

such that only transitions in $\tilde{\delta}^1_{3,max}$ are observable to U_1. In order words, transitions in $\tilde{\delta}^1_3 - \tilde{\delta}^1_{3,max}$ are not observable to U_1:

$$\begin{aligned}&(\forall(\tilde{q}, \sigma, \tilde{\delta}(\tilde{q}, \sigma)) = ((q, x_1, ..., x_N), \sigma, \tilde{\delta}(\tilde{q}, \sigma))\\&\in \tilde{\delta}^1_3 - \tilde{\delta}^1_{3,max})\sigma \notin \cup_{j=N_1+1}^{N} \varphi_j(x_j).\end{aligned} \tag{6}$$

Obviously, set of maximum controls is not unique. The following algorithm finds one such set.

Algorithm 4. *Calculation of a set of maximum controls* $\varphi_j,\ j = N_1 + 1, ..., N$.

Input: $\tilde{\delta}^1_{3,max}$

Output: $\varphi_j,\ j = N_1 + 1, ..., N$

1: For $j = N_1 + 1, ..., N$ *and* $x_j \in X_j$ *do* $\varphi_j(x_j) = \emptyset$;

2: For $j = N_1 + 1, ..., N$, $x_j \in X_j$, *and* $\sigma \in \Sigma_{o,i}$ *do* $\varphi_j(x_j) = \varphi_j(x_j) \cup \{\sigma\}$;

If $(\exists((q,x_1,...,x_N),\sigma,\tilde{\delta}(\tilde{q},\sigma))\in\tilde{\delta}_3^1-\tilde{\delta}_{3,max}^1)$
$\sigma\in\cup_{j=N_1+1}^N\varphi_j(x_j)$ *is true, then* $\varphi_j(x_j)=\varphi_j(x_j)-\{\sigma\}$*;*

3: End.

Algorithm 4 starts with empty set, that is, no broadcasting, and add events one by one unless such the addition will help U_1 to perform its task. Algorithm 4 has a computational complexity of $O(N\ |\tilde{Q}|\ |\Sigma|)$.

Information Control Problem 5

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

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

When this is the case, we need to solve the following problem: What are the impacts of minimally required broadcasting $\varphi_{j,min}$ on information control?

To solve this problem, we partition the transitions in $\tilde{G}$ into three groups by taking $\varphi_{j,min}$ into account: (1) transitions observable by U_1 itself plus minimally required broadcasting by other users, (2) additional transitions observable by its friends, and (3) the remaining transitions. In other words, $\tilde{\delta}=\tilde{\delta}_1^1\cup\tilde{\delta}_2^1\cup\tilde{\delta}_3^1$ with

$$\begin{aligned}\tilde{\delta}_1^1=&\{(\tilde{q},\sigma,\tilde{\delta}(\tilde{q},\sigma))\in\tilde{\delta}:\sigma\in\Sigma_{o,1}\vee(\tilde{q}=(q,x_1,...,x_N)\\&\wedge\sigma\in\cup_{j=1}^N\varphi_{j,min}(x_j))\}\\\tilde{\delta}_2^1=&\{(\tilde{q},\sigma,\tilde{\delta}(\tilde{q},\sigma))\in\tilde{\delta}:\sigma\in\Sigma_{o,2}\cup...\cup\Sigma_{o,N_1}\}-\tilde{\delta}_1^1\\\tilde{\delta}_3^1=&\{(\tilde{q},\sigma,\tilde{\delta}(\tilde{q},\sigma))\in\tilde{\delta}:\sigma\in\Sigma\}-(\tilde{\delta}_1^1\cup\tilde{\delta}_2^1).\end{aligned}\tag{7}$$

We first check if User U_1 can perform its task with User $U_j, j = 1, 2, ..., N$ broadcasting the minimally required broadcasting information $\varphi_{j,min}$ as follows. Let

$$\tilde{\delta}_o^1 = \tilde{\delta}_1^1$$
$$\tilde{\delta}_{uo}^1 = \tilde{\delta}_2^1 \cup \tilde{\delta}_3^1$$
$$\tilde{\delta}_\varepsilon^1 = \{(\tilde{q}, \sigma, \tilde{\delta}(\tilde{q}, \sigma)) \in \tilde{\delta} : (\tilde{q}, \sigma, \tilde{\delta}(\tilde{q}, \sigma)) \in \tilde{\delta}_o\}$$
$$\cup \{(\tilde{q}, \varepsilon, \tilde{\delta}(\tilde{q}, \sigma)) \in \tilde{\delta} : (\tilde{q}, \sigma, \tilde{\delta}(\tilde{q}, \sigma)) \in \tilde{\delta}_{uo}\}$$
$$\tilde{G}_\varepsilon^1 = (\tilde{Q}, \Sigma, \tilde{\delta}_\varepsilon^1, \tilde{q}_o)$$
$$\tilde{G}_{obs}^1 = (Y_1, \Sigma, \zeta_1, y_{1,o})$$

If $(\forall y \in Y_1)(y|_Q \times y|_Q) \cap T_{spec}^1 = \emptyset$ is true, then User U_1 can perform its task with all users broadcasting $\varphi_{j,min}$.

If User U_1 cannot perform its task with all users broadcasting $\varphi_{j,min}$, we then check if User U_1 can perform its task with the help of its friends as follows. Let

$$\tilde{\delta}_o^1 = \tilde{\delta}_1^1 \cup \tilde{\delta}_2^1$$
$$\tilde{\delta}_{uo}^1 = \tilde{\delta}_3^1$$
$$\tilde{\delta}_\varepsilon^1 = \{(\tilde{q}, \sigma, \tilde{\delta}(\tilde{q}, \sigma)) \in \tilde{\delta} : (\tilde{q}, \sigma, \tilde{\delta}(\tilde{q}, \sigma)) \in \tilde{\delta}_o\}$$
$$\cup \{(\tilde{q}, \varepsilon, \tilde{\delta}(\tilde{q}, \sigma)) \in \tilde{\delta} : (\tilde{q}, \sigma, \tilde{\delta}(\tilde{q}, \sigma)) \in \tilde{\delta}_{uo}\}$$
$$\tilde{G}_\varepsilon^1 = (\tilde{Q}, \Sigma, \tilde{\delta}_\varepsilon^1, \tilde{q}_o)$$
$$\tilde{G}_{obs}^1 = (Y_1, \Sigma, \zeta_1, y_{1,o})$$

If $(\forall y \in Y_1)(y|_Q \times y|_Q) \cap T_{spec}^1 = \emptyset$ is true, then User U_1 can perform its task with the help of its friends.

We can then minimize the communications from its friends to User U_1 by using Algorithm 1. Algorithm 1 remains unchanged, but the partition of $\tilde{\delta} = \tilde{\delta}_1^1 \cup \tilde{\delta}_2^1 \cup \tilde{\delta}_3^1$ is modified as in Equation (7) to take into account of minimal required broadcasting $\varphi_{j,min}$.

The corresponding minimum controls can be calculated using Algorithm 2. To take into account of minimal required broadcasting $\varphi_{j,min}$, Step 1 in the Algorithm 2 needs to be modified as follows.

For $i = 2, ..., N_1$ and $x_i \in X_i$ do

$$\varphi_i(x_i) = \varphi_{i,min}(x_i);$$

If users have obligation to release as much information as possible to the public, then users can maximize the broadcasting without helping User U_1 by using Algorithm 3 with the modified partition of $\tilde{\delta} = \tilde{\delta}_1^1 \cup \tilde{\delta}_2^1 \cup \tilde{\delta}_3^1$ described in Equation (7).

The corresponding maximum controls can be calculated using Algorithm 4 with Step 1 in the Algorithm 4 modified as follows.

For $j = N_1 + 1, ..., N$ and $x_j \in X_j$ do $\varphi_j(x_j) = \varphi_{j,min}(x_j)$.

5. Illustrative Example

In this section, we use an example 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 Fig. 2. The system consists of 18 cities in USA.

Figure 2: A distribution system covering 18 cities in USA.

These cities are linked by railways as shown in the automaton G of Fig. 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

Without loss of generality, we assume that the initial state is q_1. If there is a railway link between city q_i and city q_j, then two events are defined as follows.

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

Note that for the clarity of the figure, state q_i is denoted by i and not all events are labeled in Fig. 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 7 distributors/users. The cities covered by each distributor are also shown in Fig. 3. For example, User U_1 covers Seattle and Minneapolis, while User U_6 covers Dallas, Washington D.C., Houston, and Miami. Note that a city may be covered by more than one distributors.

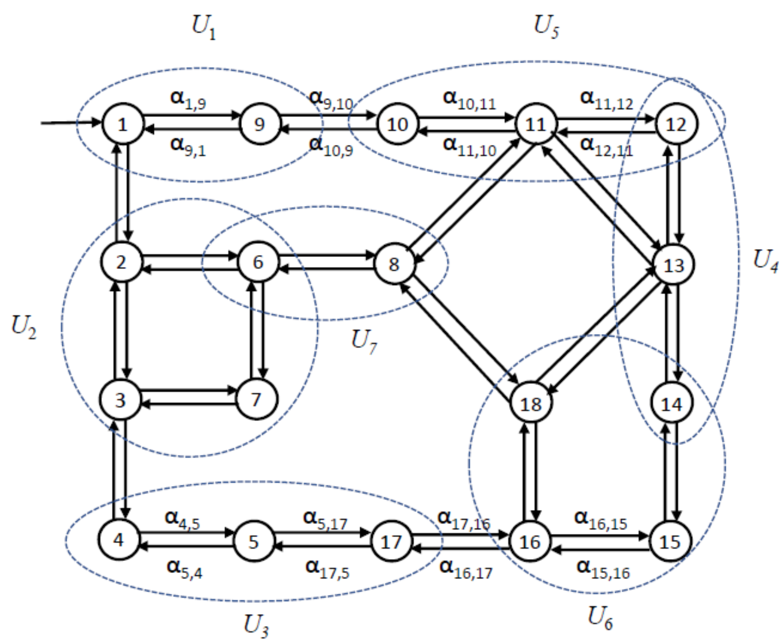
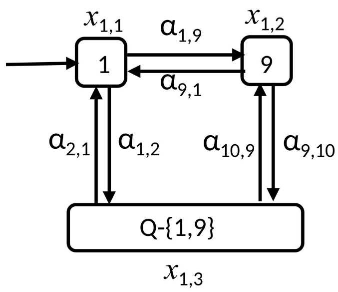


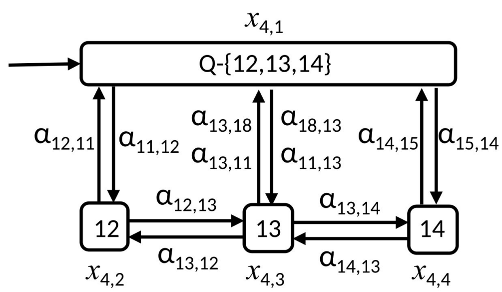
Figure 3: Automaton G of the distribution system.

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,

$$\Sigma_{o,1} = \{\alpha_{1,9}, \alpha_{9,1}, \alpha_{1,2}, \alpha_{2,1}, \alpha_{9,10}, \alpha_{10,9}\},$$



(a) H_1



(b) H_4

Figure 4: The observers of G with respect to P_1 and P_4.

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

The corresponding deterministic automata H_1 and H_4 can be obtained by constructing the corresponding observers [5] as

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

where $X_1 = \{x_{1,1}, x_{1,2}, x_{1,3}\}$, $X_4 = \{x_{4,1}, x_{4,2}, x_{4,3}, x_{4,4}\}$, $x_{1,o} = x_{1,1}$, $x_{4,o} = x_{4,1}$. The transition functions ξ_1 and ξ_1 are shown in Fig. 4. It is well-known that $P_i(L(G)) = L(H_i)$.

The users are divided into two groups:

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

To perform its tasks, User U_1 needs to know if the train has arrived in Baltimore. Thus, the specification for User U_1 is given by

$$T^1_{spec} = \{(q_{13}, q_i) : q_i \in Q - \{q_{13}\}\}. \tag{8}$$

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 User U_1 perform its task based on its own local observation without information from other users, including its friends?

To solve this problem, we let $\tilde{G} = G$ and construct the observer of G with respect to P_1, which is isomorphic to H_1 shown in Fig. 4. Since state q_{13} is mixed with other states in $x_{1,3}$ $(= Q - \{q_1, q_9\})$, the condition of Theorem 2 is not satisfied. Thus, User U_1 cannot perform its task based on its own local observation without information from other users.

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

To solve this problem, we again let $\tilde{G} = G$ and construct the observer $G^1_{obs} = (Y_1, \Sigma, \zeta_1, y_{1,o})$ of G with respect to $P_1 \cup P_2 \cup P_3 \cup P_4$ as shown in Fig. 5. Since state q_{13} only appears alone in y_7, the condition of Theorem 2 is satisfied. Thus, User U_1 can perform its task based on its own local observation and all its friends' observation.

Information Control Problem 3: How can communications from its friends to User U_1 be minimized?

To minimize communications from Users U_2, U_3, U_4 to User U_1, we construct $\tilde{G} = G||H_1||...||H_7$, which is isomorphic to G. We then partition the

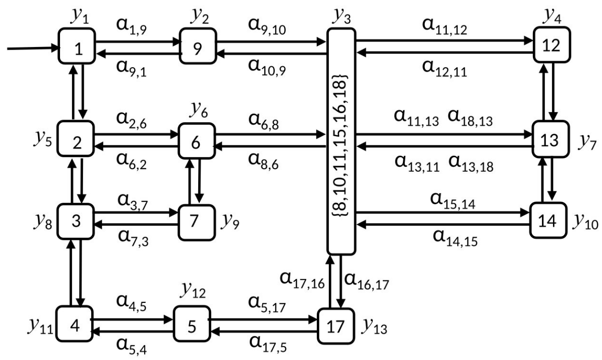


Figure 5: The observer of G with respect to $P_1 \cup P_2 \cup P_3 \cup P_4$.

transitions in $\tilde{G}$ into three groups as shown in Fig. 6: (1) transitions with events in $\Sigma_{o,1}$, denoted by $\tilde{\delta}_1^1$ and represented by bold lines in Fig. 6, (2) transitions with events in $\Sigma_{o,2} \cup \Sigma_{o,3} \cup \Sigma_{o,4} - \Sigma_{o,1}$, denoted by $\tilde{\delta}_2^1$ and represented with normal lines in Fig. 6, and (3) other transitions, denoted by δ_3^1 and represented by dashed lines in Fig. 6.

Using Algorithm 1, we can find a minimum set $\tilde{\delta}_{2,min}^1 \subseteq \tilde{\delta}_2^1$ under which the resulting observer of U_1 satisfies the condition of Theorem 2, which is given by

$$\tilde{\delta}_{2,min}^1 = \{(\tilde{q}, \sigma, \tilde{\delta}(\tilde{q}, \sigma)) : \sigma = \alpha_{13,11}, \alpha_{11,13}, \alpha_{13,12}, \alpha_{12,13}, \alpha_{13,14}, \alpha_{14,13}, \alpha_{13,18}, \alpha_{18,13}, \}.$$

The transitions in $\tilde{\delta}_{2,min}^1$ must be communicated to User U_1. Since all transitions in $\tilde{\delta}_{2,min}^1$ are related to state q_{13}, they are observed by User U_4. Therefore, User U_4 needs to communicate these transitions to User U_1, that is, the communication mapping

$$\vartheta_{4,1} : X_4 \rightarrow 2^{\Sigma_{o,4}}$$

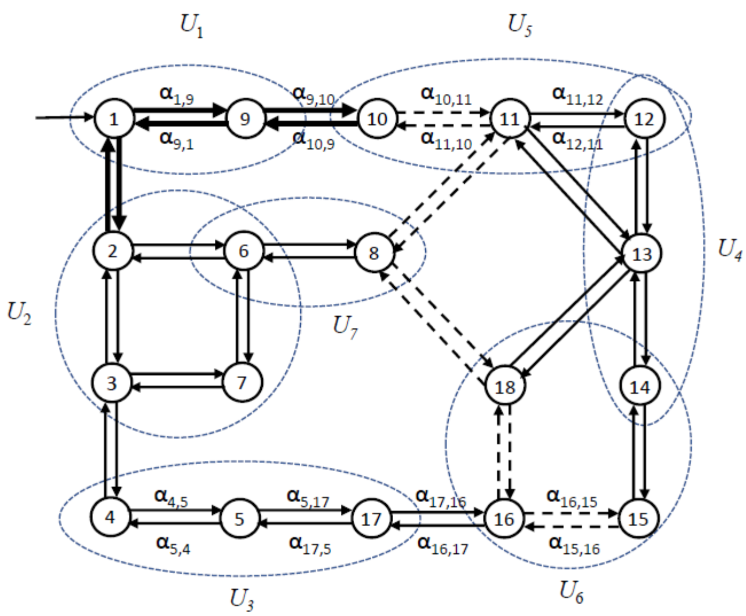


Figure 6: Partition of transitions into three groups $\tilde{\delta} = \tilde{\delta}_1^1 \cup \tilde{\delta}_2^1 \cup \tilde{\delta}_3^1$ with respect to $P_1 \cup P_2 \cup P_3 \cup P_4$: $\tilde{\delta}_1^1$ - bold lines, $\tilde{\delta}_2^1$ - normal lines, and $\tilde{\delta}_3^1$ - dashed lines.

from U_4 to U_1 is given by

$$(\forall x_4 \in X_4)\vartheta_{4,1}(x_4) = \{\alpha_{13,11}, \alpha_{11,13}, \alpha_{13,12}, \alpha_{12,13}, \alpha_{13,14}, \alpha_{14,13}, \alpha_{13,18}, \alpha_{18,13}\}.$$

Hence, for all $w \in P_4(L(G))$,

$$\theta_{4,1}(w) = \vartheta_{4,1}(\xi_4(x_{4,o}, w)) = \{\alpha_{13,11}, \alpha_{11,13}, \alpha_{13,12}, \alpha_{12,13}, \alpha_{13,14}, \alpha_{14,13}, \alpha_{13,18}, \alpha_{18,13}\}.$$

In other words, User U_4 will communicate $\alpha_{13,11}, \alpha_{11,13}, \alpha_{13,12}, \alpha_{12,13}, \alpha_{13,14}, \alpha_{14,13}, \alpha_{13,18}, \alpha_{18,13}$ to User U_1 whenever it occurs.

The communication mapping

$$\vartheta_{i,1} : X_i \to 2^{\Sigma_{o,i}}$$

from $U_i, i = 2, 3, 5, 6, 7$ to U_1 is given by

$$(\forall x_i \in X_i)\vartheta_{4,1}(x_i) = \emptyset.$$

Hence, for all $w \in P_i(L(G)), i = 2, 3, 5, 6, 7$,

$$\theta_{i,1}(w) = \vartheta_{i,1}(\xi_i(x_{i,o}, w)) = \emptyset.$$

In other words, Users $U_i, i = 2, 3, 5, 6, 7$ will communicate nothing to User U_1.

The broadcasting mapping

$$\varphi_i : X_i \to 2^{\Sigma_{o,i}}$$

from $U_i, i = 2, 3, 4, 5, 6, 7$ is given by $(\forall x_i \in X_i)\varphi_i(x_i) = \emptyset$.

Hence, for all $w \in P_i(L(G)), i = 2, 3, 4, 5, 6, 7$,

$$\phi_i(w) = \varphi_i(\xi_i(x_{i,o}, w)) = \emptyset.$$

In other words, Users $U_i, i = 2, 3, 4, 5, 6, 7$ will broadcast nothing.

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

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

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

Since User U_4 is now an adversary of User U_1, it shall not communicate anything to User U_1. Without communication from U_4, the observer $G^1_{obs} = (Y_1, \Sigma, \zeta_1, y_{1,o})$ of G with respect to $P_1 \cup P_2 \cup P_3$ is shown in Fig. 7.

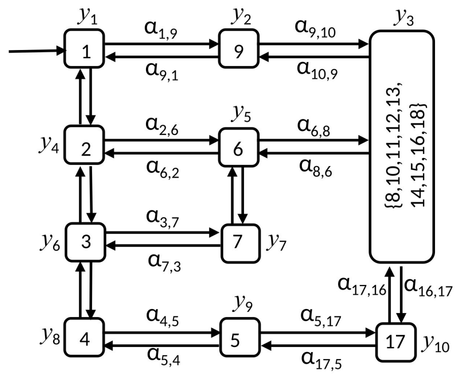


Figure 7: The observer of G with respect to $P_1 \cup P_2 \cup P_3$.

Since state q_{13} is mixed with other states in y_3 (=$\{q_8, q_{10}, q_{11}, q_{12}, q_{13}, q_{14}, q_{15}, q_{16}, q_{18}\}$), the condition of Theorem 2 is not satisfied. Thus, User U_1 cannot perform its task based on its own local observation and the observations from its friends. So, the problem is: How can U_4, U_5, U_6, U_7 broadcast maximal information to the public without helping U_1?

To solve the problem, we consider the new partition $\tilde{\delta} = \tilde{\delta}_1^1 \cup \tilde{\delta}_2^1 \cup \tilde{\delta}_3^1$ as shown in Fig. 8. Under the new grouping, $\tilde{\delta}_1^1$ is represented by bold lines in Fig. 8 and is same as in Fig. 6; $\tilde{\delta}_2^1$ contains transitions with events in $\Sigma_{o,2} \cup \Sigma_{o,3} - \Sigma_{o,1}$, which is represented by normal lines in Fig. 8; and $\tilde{\delta}_3^1$ contains the remaining transitions (with events in $\Sigma - \Sigma_{o,1} \cup \Sigma_{o,2} \cup \Sigma_{o,3}$), which is represented by dashed lines in Fig. 8.

In order to maximize the broadcasting without helping U_1, we use Algorithm 3 to find a maximum set $\tilde{\delta}_{3,max}^1 \subseteq \tilde{\delta}_3^1$ such that the condition of Theorem 2 is not satisfied if U_1 knows the occurrences of transitions in $\tilde{\delta}_1^1 \cup \tilde{\delta}_2^1 \cup \tilde{\delta}_{3,max}^1$.

$\tilde{\delta}_{3,max}^1$ is not unique. One such $\tilde{\delta}_{3,max}^1$ is given by

$$\tilde{\delta}_{3,max}^1 = \tilde{\delta}_3^1 - \{(\tilde{q}_{13}, \alpha_{13,11}, \tilde{q}_{11})\}.$$

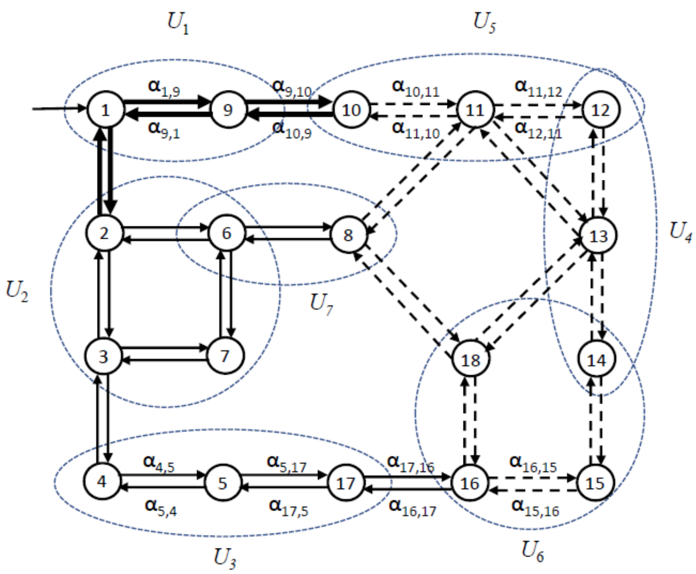


Figure 8: Partition of transitions into three groups $\tilde{\delta} = \tilde{\delta}_1^1 \cup \tilde{\delta}_2^1 \cup \tilde{\delta}_3^1$ with respect to $P_1 \cup P_2 \cup P_3$: $\tilde{\delta}_1^1$ - bold lines, $\tilde{\delta}_2^1$ - normal lines, and $\tilde{\delta}_3^1$ - dashed lines.

The corresponding broadcasting mapping

$$\varphi_i : X_i \rightarrow 2^{\Sigma_{o,i}}$$

from $U_i, i = 2, 3, 4, 5, 6, 7$ can be calculated using Algorithm 4 as

$$\varphi_4(x_4) = \Sigma_{o,4} - \{\alpha_{13,11}\}, \quad \varphi_5(x_5) = \Sigma_{o,5} - \{\alpha_{13,11}\}$$

and for all other $x_i \in X_i, i = 2, 3, 4, 5, 6, 7$,

$$\varphi_i(x_i) = \Sigma_{o,i}.$$

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 minimum required information release.

$$\begin{aligned}
&(\forall x_1 \in X_1)\varphi_{1,min}(x_1) = \emptyset \\
&(\forall x_2 \in X_2)\varphi_{2,min}(x_2) = \{\alpha_{2,6}, \alpha_{6,2}\} \\
&(\forall x_3 \in X_3)\varphi_{3,min}(x_3) = \emptyset \\
&(\forall x_4 \in X_4)\varphi_{4,min}(x_4) = \{\alpha_{12,13}, \alpha_{13,12}, \alpha_{14,13}, \alpha_{13,14}, \} \\
&(\forall x_5 \in X_5)\varphi_{5,min}(x_5) = \{\alpha_{11,13}, \alpha_{13,11}\} \\
&(\forall x_6 \in X_6)\varphi_{6,min}(x_6) = \{\alpha_{18,13}, \alpha_{13,18}\} \\
&(\forall x_7 \in X_7)\varphi_{7,min}(x_7) = \emptyset.
\end{aligned} \tag{9}$$

For the above $\varphi_{i,min}$, we re-partition $\tilde{\delta}$ into three groups $\tilde{\delta} = \tilde{\delta}_1^1 \cup \tilde{\delta}_2^1 \cup \tilde{\delta}_3^1$ as shown in Fig. 9: $\tilde{\delta}_1^1$ are locally transitions observable by U_1 itself plus minimally required transitions broadcasted by other users, represented by bold lines in Fig. 9, $\tilde{\delta}_2^1$ are additional transitions observable by its friends, represented by normal lines in Fig. 9, and $\tilde{\delta}_3^1$ are the remaining transitions, represented by dashed lines in Fig. 9.

Let us check if User U_1 can perform its task by observing transitions in $\tilde{\delta}_1^1$ only. To do so, we construct the observer of G with respect to $\tilde{\delta}_1^1$, which is shown in Fig. 10. Since in the observer, state q_{13} is not mixed with other states, the specification (8) is satisfied. Therefore, User U_1 can perform its task by observing its local events and minimally required transitions broadcasted by other users.

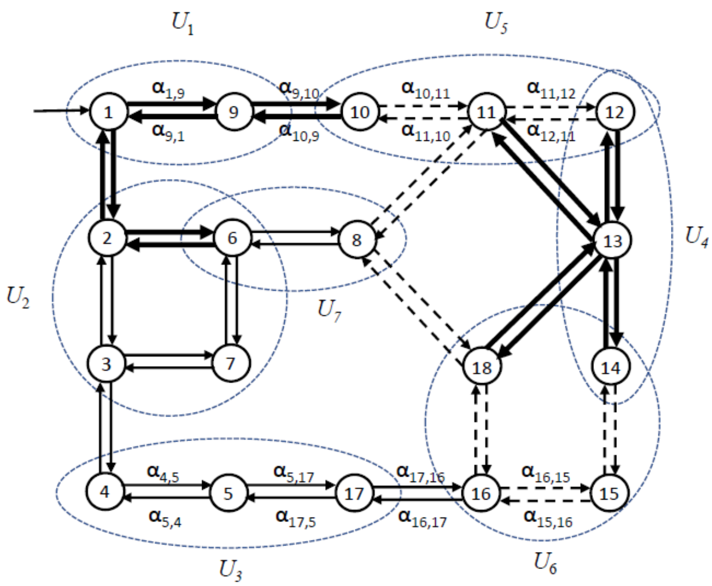


Figure 9: Partition of transitions into three groups $\tilde{\delta} = \tilde{\delta}_1^1 \cup \tilde{\delta}_2^1 \cup \tilde{\delta}_3^1$ with respect to $P_1 \cup P_2 \cup P_3$ and $\varphi_{i,min}$ given by Equation (9): $\tilde{\delta}_1^1$ - bold lines, $\tilde{\delta}_2^1$ - normal lines, and $\tilde{\delta}_3^1$ - dashed lines.

6. Conclusion

We investigate information flow and information control in a large networked systems in the framework of discrete event systems. The main contributions of the paper are summarized as follows. (1) A discrete event system model of a large networked system is proposed with different users, each observes a set of locally observable events. (2) Control of information exchange among users by private communications from one user to another and by public broadcasting to all users is introduced. (3) Five information control problems are investigated and solved for information exchanges among friends and adversaries. (4) Controllers are designed to communicate minimum information to friends to enhance security. (5) Controllers are designed

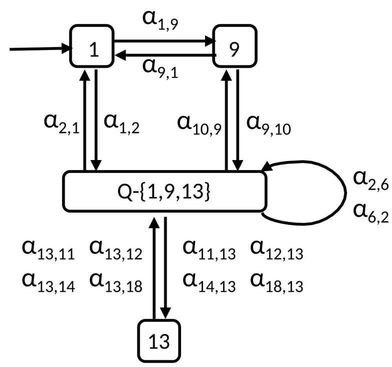


Figure 10: The observer of G with respect to $\tilde{\delta}_1^1$.

to broadcast maximum information to ensure transparency.

[1] C. Arndt. *Information measures: information and its description in science and engineering.* Springer Science & Business Media, 2003.

[2] B. Behinaein, F. Lin, and K. Rudie. Optimal information release for mixed opacity in discrete-event systems. *IEEE Transactions on Automation Science and Engineering*, 16(4):1960–1970, 2019.

[3] M. P. Cabasino, A. Giua, and C. Seatzu. Diagnosability of discrete-event systems using labeled petri nets. *IEEE Transactions on Automation Science and Engineering*, 11(1):144–153, 2013.

[4] L. K. Carvalho, M. V. Moreira, and J. C. Basilio. Generalized robust diagnosability of discrete event systems. *IFAC Proceedings Volumes*, 44(1):8737–8742, 2011.

[5] C. G. Cassandras and S. Lafortune. *Introduction to discrete event systems.* Springer, 2008.

[6] M. Gomez-Rodriguez and B. Schölkopf. Influence maximization in continuous time diffusion networks. In *Proceedings of the 29th International Conference on Machine Learning*, pages 313–320. Omnipress, 2012.

[7] F. Granese, D. Gorla, and C. Palamidessi. Enhanced models for privacy and utility in continuous-time diffusion networks. *International Journal of Information Security*, 20(5):763–782, 2021.

[8] C. Keroglou and C. N. Hadjicostis. Detectability in stochastic discrete event systems. *Systems & Control Letters*, 84:21–26, 2015.

[9] F. Lin. Diagnosability of discrete event systems and its applications. *Discrete Event Dynamic Systems*, 4(2):197–212, 1994.

[10] F. Lin and W. M. Wonham. On observability of discrete-event systems. *Information sciences*, 44(3):173–198, 1988.

[11] Y. Liu, Z. Liu, X. Yin, and S. Li. An improved approach for verifying delayed detectability of discrete-event systems. *Automatica*, 124:109291, 2021.

[12] R. McEliece and R. J. Mac Eliece. *The theory of information and coding*, volume 86. Cambridge University Press, 2002.

[13] P. J. Ramadge and W. M. Wonham. Supervisory control of a class of discrete event processes. *SIAM journal on control and optimization*, 25(1):206–230, 1987.

[14] F. M. Reza. *An introduction to information theory.* Courier Corporation, 1994.

[15] K. Rudie, S. Lafortune, and F. Lin. Minimal communication in a distributed discrete-event system. *IEEE transactions on automatic control*, 48(6):957–975, 2003.

[16] K. Rudie and W. M. Wonham. Think globally, act locally: Decentralized supervisory control. In *1991 American Control Conference*, pages 898–903. IEEE, 1991.

[17] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis. Diagnosability of discrete-event systems. *IEEE Transactions on automatic control*, 40(9):1555–1575, 1995.

[18] C. E. Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[19] S. Shu and F. Lin. Detectability of discrete event systems with dynamic event observation. *Systems & control letters*, 59(1):9–17, 2010.

[20] S. Shu, F. Lin, and H. Ying. Detectability of discrete event systems. *IEEE Transactions on Automatic Control*, 52(12):2356–2359, 2007.

[21] R. W. Yeung. *A first course in information theory*. Springer Science & Business Media, 2002.

[22] X. Yin and S. Lafortune. A general approach for optimizing dynamic sensor activation for discrete event systems. *Automatica*, 105:376–383, 2019.