Networked State Estimation over a Shared Communication Medium

M. Xia, V. Gupta and P. J. Antsaklis

Abstract—We consider state estimation for multiple plants across a shared communication network. Each linear time-invariant plant transmits information through the common network according to either a time-triggered or an event-triggered rule. For an event-triggered algorithm with CSMA (carrier sense multiple access), each plant is assumed to access the network based on a priority mechanism. For a time-triggered algorithm combined with TDMA (time division multiple access), each plant uses the network according to a schedule that is decided a priori. Performance in terms of the communication frequency and the estimation error covariance is analytically characterized for event-triggered schemes. Using these characterizations, we show that event-triggered schemes may perform worse than time-triggered schemes when the effect of the communication network is explicitly considered.

I. INTRODUCTION

In a networked control system (NCS), the communication (between the sensors, the actuators and the controllers) is supported by a shared communication network [2]–[4]. The use of the shared network results in flexible architectures and low costs for installation and maintenance [5]–[7]. Applications of NCS can be found in unmanned aerial vehicles, mobile sensor networks, power systems and automobile systems [7]–[9]. The field of NCS has grown too large to be summarized in a few lines. We thus point to special issues such as [5], [7], [9]–[11] for a general introduction.

In an NCS, the information for control or estimation is often sent across the network in a periodic manner, i.e. using a time-triggered method. However, this may lead to inefficient use of the network [12], [13]. For instance, if the changes in the plant output measurements are very small, then periodic transmission of such measurements for control purposes is not necessary [14], [15]. In contrast, using an event-triggered method can be more efficient since information transmission occurs only if there is some significant change in the measurements. Under many circumstances event-triggered method can achieve comparable performance while reducing the use of the computational and communication resources [8], [12], [16]. Thus, event-triggered sampling and transmission algorithms have emerged as exciting alternatives to periodic, or time-triggered, sampling and transmission traditionally used in estimation and control. To compare a time-triggered method and an event-triggered method for design purposes, two performance metrics are often considered, i.e. the communication rate and the control/estimation performance [12], [13], [17]. For the control and estimation of a scalar Weiner process, works such as [18], [19] showed that the average number of transmissions required for the same state variance could be reduced significantly with event-triggered schemes. Such reductions were also noted experimentally in works such as [20]. Motivated by these results, event-triggered schemes to ensure stability or passivity for arbitrary nonlinear systems were designed in works such as [12], [14], [16]. Event-triggered schemes have also been used in distributed systems [21] and model-based control [22].

For system design, it is of interest to characterize analytically the performance in terms of estimation/control performance as a function of the communication rate. Some such results are available in the literature. For example, a relationship between the communication rate and the estimation quality using event-triggered state estimation was obtained in [23] by using an approximation technique from nonlinear filtering. In [24], a stochastic event-triggered rule for state estimation was used so that the bounds of the error covariance were obtained with no need of the approximation technique used in [23]. The problem of jointly optimal sampling and control was studied in [25], where an optimal controller was characterized for a simple Brownian motion dynamics. Event-triggered schemes were designed in [26]–[29] for state estimation in order to minimize the error covariance while satisfying certain constraints on the communication rate. However, an analytic trade-off between the covariance and communication rate by imposing a hard constraint on the number of communications typically leads to complicated triggering events that are computationally difficult to compute and implement [30], [31]. We would also like to note the related work in [32] that uses generalized geometric programming optimization techniques for an event-triggered approach for the sensor scheduling problem. Analytical expressions for either the communication rate or the estimation error covariance control cost function for general system models remain hard to obtain.

Another direction in which the idea of event-triggered communication is being extended is by moving beyond the assumption that a single process needs to be estimated or controlled. If multiple processes are present, then transmissions corresponding to more than one process may be triggered at the same time. If the communication medium is shared, this can lead to congestion, and in turn, delays and packet losses. Realizing this fact, recent work has considered the interaction of control architecture and communication strategies in the
setting of event triggered control. For instance, [4] considered a communication network being shared by a number of independent control loops and used numerical methods to compare the control performance under various multiple access schemes such as TDMA (time division multiple access), FDMA (frequency division multiple access) and CSMA (carrier sense multiple access). It was shown in [4] that CSMA with event-triggered sampling is a superior scheme in the particular numerical examples presented. The work in [33] considered scalar noisy integrator models and provided the stationary distribution of the process state of such integrator models under the CSMA scheduling policy. A key assumption for their results to hold is that the network is collision free, which is a restrictive assumption if transmission is across wireless networks. Event-triggered loops sharing a network were treated as event-triggered loops subject to packet losses in [34], where it was assumed that packet losses happen when the loops contend for the use of the network. However, the analysis was based on the assumption that the losses for different loops are independent, which is an assumption that does not hold in general as shown in [35]. Moreover, the analysis was limited to processes described by a single integrator driven by white noise. In [36], the same setup was considered and the communication network was modeled using the ALOHA protocol. The correlation among different loops was removed through a particular triggering rule and performance was characterized. A more sophisticated strategy for conflict resolution when two plants wish to transmit simultaneously was proposed in [35], [37]. A Markov chain based model was introduced to characterize the probability of successful transmission for each plant in steady state. The key assumption (originating from [38]) was that the conditional probability of a busy channel for the node attempting to transmit is independent for each node, which is again restrictive. Sufficient conditions were provided in [37] to guarantee mean squared stability of the closed-loop system and specific event-triggered policies were designed that satisfied these conditions. The problem of analytically characterizing the performance of event-triggered loops with contention based medium access schemes also remains open in general [13].

In this paper, we consider multiple plants transmitting information through a common network according to either a time-triggered rule or an event-triggered rule. To avoid collisions when multiple plants wish to transmit in the event-triggered setting, we use CSMA based on a priority mechanism as was proposed in [4]. For the case when the plants transmit according to a time-triggered rule, no collisions are possible and we use a TDMA (round-robin) transmission schedule that has been designed a priori. We derive analytical expressions for estimation performance in terms of the communication rate and the estimation error covariance for the TDMA based schedule and CSMA based schedule with static priorities. For CSMA based schedules with random and dynamic priorities, we provide approximate expressions that agree very well with numerical results. As an example, we use our results to demonstrate that the simple time-triggered implementation can outperform the event-triggered implementation when multiple loops share access to the network. This result may be of interest to designers while moving from implementing event-triggered schemes for a single plant to a wider array of applications.

The main contributions of this paper are summarized as follows. (i) Estimation performance in terms of both the communication rate and the estimation error covariance for a single-plant using threshold based event-triggered rules is analytically characterized. (ii) We also provide performance analysis for state estimation when multiple event-triggered loops share the network based on static, random and dynamic schedulers. For static schedulers, and in the limit when the network utilization is high, our results are exact. For random and dynamic schedulers, our results are approximate but agree very well with numerical results. (iii) When multiple plants share the network, we show that time-triggered scheme combined with TDMA may lead to a smaller error covariance than event-triggered scheme combined with static and random scheduling policies for the same communication frequency.

The rest of the paper is organized as follows. Section II presents the problem formulation. The analysis for event-triggered estimation of a single plant over a dedicated network is provided in Section III and the analysis is extended to NCS with multiple plants sharing the communication network in Section IV. Analytical results in the limit when the triggering level approaches zero for multiple plants are provided in Section V. This paper concludes with some avenues for future work in Section VI.

**Notation:** The $n$-dimensional real space is denoted by $\mathbb{R}^n$. The set of positive integers is denoted by $\mathbb{Z}$. Denote the vector of all zeros by $0$ and the vector of all ones by $1$. The infinity norm of a vector $x$ is denoted by $\|x\|$. For a matrix $M$, the $(i,j)$-th element is denoted by $M_{ij}$. The transpose of a vector $x$ (or a matrix $M$) is denoted by $x^\top$ (or $M^\top$). For a random variable $X$, we denote the covariance as $\text{var}(X)$. For an $m$-dimensional multivariate Gaussian random variable $X$ with mean vector $\mu$ and covariance $R$, we denote the cumulative distribution function $F$ function as $\Pr(|X| < x) \triangleq F(m, \mu, R, x)$, where the inequality is interpreted element-wise. For the truncated multivariate Gaussian random variable obtained by truncating $X$ between the vectors $t_1$ and $t_2$, define the variance by $\Sigma(X, t_1, t_2)$. As with the standard $F$ functions and truncated Gaussian distributions, evaluation of these generalizations is done through Gaussian integrals (see, e.g., [39, Equation (16)] for formulas for the variance of truncated Gaussian distributions) and is a standard feature in most statistics packages. Solutions that are in terms of such expressions are, thus, considered analytical solutions.

**II. PROBLEM FORMULATION**

**A. Problem Setup**

Consider the problem setup as shown in Fig. 1 where $N$ plants transmit information over a shared network with the following associated assumptions.

**Plant and Sensor:** The $i$-th ($1 \leq i \leq N$) plant is described by the following discrete linear time-invariant evolution

\begin{align}
    x_i(k+1) &= A_i x_i(k) + w_i(k), \\
    y_i(k) &= x_i(k) 
\end{align}

---

0018-9286 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
where $x_i(k) \in \mathbb{R}^n$ denotes the state vector, $y_i(k) \in \mathbb{R}^m$ is the output vector, $u_i(k)$ is the process noise assumed to be white Gaussian with zero mean and covariance $R_{w_i} > 0$. The initial condition of the process $x_i(0)$ is assumed to be a Gaussian random vector with zero mean and covariance $R_{e_i}(0)$. The process noises $\{w_i(k)\}$ and the initial condition $x_i(0)$ are assumed to be mutually independent among all processes.

**Remark 1:** We focus on the case when the state is observed solely for pedagogical ease. When the process state is not observed, the development will be similar if we use, e.g., a Kalman Filter at the sensor end as proposed in [40], [41].

**Estimator:** At every time $k$, the $i$-th estimator generates a minimum mean squared error (MMSE) estimate for the state $x_i(k)$ based on whatever information is available to it. Denote the estimate for state $x_i(k)$ held by the $i$-th estimator as $\hat{x}_i^{dec}(k)$. Since we assume that the sensors can observe the states directly, at the $i$-th estimator, we have

$$\hat{x}_i^{dec}(k) = \begin{cases} x_i(k), & \text{if the $i$-th packet received} \\ A_i \hat{x}_i^{dec}(k-1), & \text{otherwise} \end{cases}$$

(2)

where $A_i \hat{x}_i^{dec}(k-1)$ is the optimal estimate at the estimator if the estimator did not receive any information at time $k$ [40].

**Comparator:** The event-triggered algorithm is implemented at the comparator. We consider a level or threshold based scheme as proposed e.g. in [18], [19], [36]. The local event for the $i$-th plant is defined as

$$|e_i^{comp}(k)| > \varepsilon_i$$

(3)

where $e_i^{comp}(k) = x_i(k) - A_i \hat{x}_i^{dec}(k-1)$ and the threshold $\varepsilon_i$ is a given constant. Note that the comparator does not need information from the estimator of $\hat{x}_i^{dec}(k)$ since it is known by the comparator that the estimate $\hat{x}_i^{dec}(k)$ is calculated according to (2).

**Communication Network:** We denote by $\{s_k\}_{k=0,1,2...}$ the packet sequence that is being transmitted across the network. We denote by $s_k = i$ the event that the $i$-th ($1 \leq i \leq N$) plant is allowed to use the network at time $k$ ($k \geq 0$) and by $s_k = \emptyset$ the event the network does not attempt to transmit packet from any plant at time $k$. The communication network is modeled as satisfying the following assumptions.

- **A1:** The network does not permit simultaneous transmissions. Hence, only one plant can transmit information at any time step.
- **A2:** The transmission delay is less than one time step according to the clock by which the process evolves [35], [36]. Further, every packet that is transmitted is received successfully.
- **A3:** Each plant attempts to transmit information whenever an event is generated in an event-triggered implementation. When two or more plants send information simultaneously, the network transmits the packet received from the plant with highest priority [4] and the rest of the packets are discarded. The priority orders of the plants can be decided according to a collision resolution mechanism (CRM) as was proposed in [4]. Let $p_i(k) \in \mathbb{Z}$ be the priority of the $i$-th plant ($1 \leq i \leq N$) at time $k$ ($k \geq 0$). We denote by $p_i(k) = \mu_i \in \{1,2,\ldots,N\}$ the event that the $i$-th plant has the $\mu_i$-th priority among all the plants at time $k$. To illustrate the priority assignment mechanisms, we consider the case when the $i$-th plant and the $\ell$-th plant ($i \neq \ell$) contend to use the network at time $k$, i.e. $|e_i^{comp}(k)| > \varepsilon_i$ and $|e_{\ell}^{comp}(k)| > \varepsilon_{\ell}$.
  1) With a static priority mechanism, $p_i(k) \equiv \mu_i$ and $p_{\ell}(k) \equiv \mu_{\ell}$, where $\mu_i$ and $\mu_{\ell}$ ($\mu_i \neq \mu_{\ell}$) have been decided in advance.
  2) With a random priority mechanism, $\mu_i < \mu_{\ell}$ happens with probability $P_{\alpha}$ at any time $k$.
  3) With a dynamic priority mechanism, $\mu_i > \mu_{\ell}$ if $|e_i^{comp}(k)| > |e_{\ell}^{comp}(k)|$.

- **A4:** In a time-triggered setting, TDMA (time division multiple access) is used to determine which plant accesses the network at any time step using a periodic access schedule that has been decided in advance. As an example, a possible sequence for $\{s_k\}_{k=0,1,2...}$ under TDMA scheduling could be $\{1,2,\ldots,N,\emptyset,1,2,\ldots,N,\emptyset,\ldots\}$.
- **A5:** In both time-triggered and event-triggered setups, the network allows each plant to transmit at least once every $T$ time steps ($T$ is assumed to be large but bounded) to guard against the practical concerns of synchronization, malfunctioning sensors and so on.

**Remark 2:** The medium access schemes described above have been used in many applications. For instance, TDMA is used in mobile communications and WirelessHART [42]. Static and dynamic schedulers are used in Control Area Network (CAN) and random schedulers are used in Ethernet or wireless local area network (WLAN), see e.g. [43], [44].

**Performance Metrics:** We are interested in analyzing the performance of the system as measured by the following two metrics.

1) The communication rate $P$, which is defined as the average probability for the network to transmit information at any time step. Also of interest is to evaluate the average probability at which each plant attempts to transmit.

2) For the $i$-th plant, the estimation performance is defined as the average expected estimation error covariance [26]–[28], [32]. Since there are $N$ plants in the NCS, the
quality of estimate is measured by the aggregate error covariance
\[
J \triangleq \sum_{i=1}^{N} \lim_{t \to \infty} \frac{1}{t} \sum_{k=0}^{t} \mathbb{E} \left[ e_i^{\text{dec}}(k) [e_i^{\text{dec}}(k)]^\top \right]
\] (4)
with \( e_i^{\text{dec}}(k) \triangleq x_i(k) - \hat{x}_i^{\text{dec}}(k) \) as the estimation error for the \( i \)-th \((1 \leq i \leq N)\) plant.

B. An Illustrative Example

We present a simple example to illustrate the formulation. The example also shows that time-triggered method may outperform event-triggered method with associated medium access schemes (i.e. CRMs) described earlier.

Example 1: Suppose the dynamics for the \( i \)-th plant \((i = 1, 2)\) are given by equation (1) with \( A_1 = 1 \) and \( A_2 = 0.9 \). The process noise \( \{w_i(k)\} \) is white, zero mean, Gaussian with covariance unity and the initial condition \( x_i(0) \) is a normal Gaussian random variable.

1) Event-triggered Method: The event for the \( i \)-th plant that triggers information transmission is given by equation (3). We assume that the network allows each plant to transmit at least once every 10 time steps; thus \( T = 10 \). The simulation results are given in Fig. 2 by conducting 10,000 Monte Carlo experiments [45], [46] and by setting \( \varepsilon_1 = \varepsilon_2 \). As shown in the top plot of Fig. 2, the communication rates under various CRM are close to each other for various values of \( \varepsilon_i \). However, it can be seen from the bottom plot of Fig. 2 that the qualities of the estimates under various CRM are different from each other, especially when the level \( \varepsilon_i \) is small, e.g. when \( \varepsilon_i \leq 1 \). As the level \( \varepsilon_i \) gets larger (i.e. when the information transmission becomes less frequently), the communication rate will converge to the value of 0.2 and the quality of estimate will become progressively worse.

2) Time-triggered Method: We assume that each plant uses the network periodically and that the two plants use the network asynchronously. Since we do not consider a cost associated with the utilization of the communication network, we focus on the case when the communication rate (denoted by \( P \)) is close to one, e.g. \( P = 0.98 \). The packets are transmitted through the network according to the following sequence,

\[
\{1, 2, 1, 2, \ldots, 1, 2, \emptyset, 1, 2, 1, 2, \ldots, 1, 2, \emptyset, \ldots\}
\]

3) Comparison of Event-triggered and Time-triggered Methods: The comparison between time-triggered method and event-triggered method is summarized in Table I. From Table I and Fig. 2, we can make the following observations. For the same communication rate \( P = 0.98 \) (i.e. the same utilization of the network), the corresponding thresholds for static, random and dynamic schedulers are different. Further, the event-triggered method with static and random schedulers lead to larger estimation error covariance than the time-triggered method with TDMA. In other words, the time-triggered method with TDMA outperforms the event-triggered method with static and random schedulers. The event-triggered method with dynamic scheduler leads to smaller estimation error covariance than time-triggered method only if \( \varepsilon_i \leq 1 \). □

III. PRELIMINARY RESULTS - SINGLE PLANT ACROSS A DEDICATED NETWORK

We begin with some preliminary results that outline our method of deriving the communication rate and error covariance analytically for event-triggered estimation by considering the simple case of a single plant, see Fig. 3. These results may be of independent interest and were presented first in [1]. The subscript \( i \) for the different plants is dropped in this section. Following (2), the trigger for information transmission is given by the comparator

\[
|e^{\text{comp}}(k)| \triangleq |x(k) - Ax^{\text{dec}}(k-1)| > \varepsilon.
\]

Note that there is no contention for access to the network; thus, whenever an event is triggered, the transmission is successful.

To analyze the performance, we will find it convenient to define a discrete-time discrete-state Markov chain \( M \) with \( T \) + 1 nodes as shown in Fig. 4. We note that a similar structure was used to represent the transition graph of the backward recurrence chain in [46]. The state of the chain is denoted by \( \{X(k)\}_{k \geq 0} \) such that \( X(k) = j \) implies that at time \( k \), the last transmission occurred at time \( k - j \). The transition probabilities are defined as

\[
p_{ij} \triangleq Pr(X(k + 1) = j | X(k) = i)
\]

where \( i, j \in \{0, 1, 2, \ldots, T\} \).

The communication frequency and the estimation error covariance are characterized by this Markov chain. To this
end, define the random variables

\[
Z_i(k) = \sum_{j=0}^{i} A^j w(k+j), \quad 0 \leq i \leq T. \tag{5}
\]

Since the noise \(w(k)\) is white, the probability density function of the variables \(Z_i(k)\) is independent of \(k\). In the sequel, we will simply write \(Z_i\) to denote these random variables. For any \(i\), the vector random variable \(M_i = [Z_0^T, Z_1^T, \ldots, Z_i^T]^T\) has a multi-variate normal distribution with mean \(0\) and covariance matrix \(R_i\) given by (6).

Now, for \(1 \leq i \leq T\), define the events

\[
N_i = \{ |Z_0| < \varepsilon \} \cap \{ |Z_1| < \varepsilon \} \cap \cdots \cap \{ |Z_{i-1}| < \varepsilon \} \tag{7}
\]

with the convention that \(N_0\) is the sure event with probability \(Pr(N_0) = 1\). For \(1 \leq i \leq T\), we have

\[
Pr(N_i) = F(ni, 0, R_i, \varepsilon 1). \tag{8}
\]

**Lemma 1:** Consider the Markov chain as defined above. The transition probabilities \(p_{ij}\) are given by

\[
p_{ij} = \begin{cases} 
1 - \frac{F(n(i+1), 0, R_{i+1}, \varepsilon 1)}{F(ni, 0, R_i, \varepsilon 1)}, & 0 \leq i \leq T - 1, j = 0 \\
1, & i = T, j = 0 \\
1 - p_{i0}, & 0 \leq i \leq T - 1, j = i + 1 \\
0, & \text{otherwise} 
\end{cases} \tag{9}
\]

**Proof:** We concentrate on the case when \(0 \leq i \leq T - 1, j = 0\) since the other expressions are obvious from the structure of the Markov chain shown in Fig. 4. Consider the transition probability \(p_{00}\). Since \(X(k) = 0\) is equivalent to \(e^{dec}(k) = 0\), we have

\[
p_{00} = Pr(X(k+1) = 0 | X(k) = 0) = Pr(|w(k)| > \varepsilon | e^{dec}(k) = 0)
\]

\[
\leq Pr(|w(k)| > \varepsilon)
\]

\[
= Pr(|Z_0| > \varepsilon),
\]

where \((a)\) holds because \(e^{dec}(k)\) is independent of the process noise at time step \(k\). Similarly, for any \(i\) such that \(0 \leq i \leq T - 1\), the probability

\[
p_{i0} = Pr(X(k+1) = 0 | X(k) = i)
\]

\[
\leq Pr(|Z_i| > \varepsilon | N_i, e^{dec}(k-1) = 0)
\]

\[
\leq Pr(|Z_i| > \varepsilon, Z_{i-1} < \varepsilon, \ldots, Z_0 < \varepsilon)
\]

\[
= Pr(|Z_i| > \varepsilon, N_i)
\]

\[
= 1 - Pr(N_{i+1})/Pr(N_i)
\]

where \((b)\) follows the Markovian property and the definitions in (5), and \((c)\) holds because \(e^{dec}(k-1)\) is independent of the process noise after time step \(k - i\), and hence \(Z_i\). Now the result follows from (8) and the fact that \(p_{T0} = 1\). □

**Remark 3:** Note that the transition probabilities are in terms of Gaussian integrals. Calculation of such integrals is a standard feature in most statistics packages. Similar to the solution in terms of a Riccati equation, the expression (9) is also usually considered to be an analytical solution.

**Theorem 1:** The average communication rate for the event-triggered algorithm described above for the single plant case is given by

\[
\lim_{k \to \infty} \frac{1}{k} \sum_{i=0}^{k-1} (1 - p_{00}) = 1 - \frac{1}{1 + \sum_{j=0}^{T-1} (1 - p_{0j})}
\]

where the transition probabilities \(p_{ij}\) are given by (9).

**Proof:** The average communication rate for the system is given by

\[
\lim_{k \to \infty} \frac{1}{k} \sum_{i=0}^{k-1} Pr(X(k) = 0).
\]

From the fact that \(p_{00}\)’s are time-invariant and using the structure of the Markov chain from Fig. 4, the probability for each mode \(j \geq 1\) can be computed as

\[
Pr(X(k) = j) = (1 - p_{j-1,0}) Pr(X(k) = j - 1)
\]

\[
= \prod_{i=0}^{j-1} (1 - p_{i0}) Pr(X(k) = 0).
\]

\[
\tag{10}
\]

Thus, the balance equation for the Markov chain yields

\[
1 = \sum_{j=0}^{T} Pr(X(k) = j)
\]

\[
= Pr(X(k) = 0) + \sum_{j=1}^{T-1} \prod_{i=0}^{j-1} (1 - p_{i0}) Pr(X(k) = 0)
\]

\[
= (1 + \sum_{j=1}^{T} (1 - p_{0j})) Pr(X(k) = 0).
\]

The required probability \(Pr(X(k) = 0)\) can now be calculated as

\[
Pr(X(k) = 0) = \frac{1}{1 + \sum_{j=1}^{T} (1 - p_{0j})}.
\]

\[
\tag{11}
\]
This completes the proof.

The other performance metric is the covariance of estimation error \( \Pi(k) = \mathbb{E}[e^{dec}(k)(e^{dec}(k))'] \) which is given by the following relation.

**Theorem 2:** The steady state average error covariance \( \Pi = \lim_{k \to \infty} \Pi(k) \) for the event-triggered algorithm described above for the single-plant case is given by

\[
\Pi = \sum_{j=1}^{T-1} \prod_{i=0}^{j} (1 - p_{0j}) \Pr(X(k) = 0) \Sigma_{M,j}(j,j) \tag{12}
\]

where \( \Sigma_{M,j} = \Sigma(M_j, -\epsilon 1, \epsilon 1) \), the transition probabilities \( p_{0j} \) are given by (9), and the communication rate \( \Pr(X(k) = 0) \) is given by (11).

**Proof:** We use the relation \( \Pi(k) = \sum_{j=0}^{T-1} \Pr(X(k) = j) \mathbb{E}[e^{dec}(k)(e^{dec}(k))' | X(k) = j] \). For \( j = 0 \), since the estimation error \( e^{dec}(k) \) is 0, we obtain \( \mathbb{E}[e^{dec}(k)(e^{dec}(k))' | X(k) = j] = 0 \). For \( j > 0 \), we use the fact that the error \( e^{dec}(k) \) under the event \( X(k) = j \) is simply \( \sum_{i=0}^{j} A_i w(k-i) \). However, since the process noise \( w(j) \) is white and has a time-invariant probability distribution function, we can alternatively write

\[
\mathbb{E}[e^{dec}(k)(e^{dec}(k))' | X(k) = j] = \text{var}[Z_{j-1} | N_j]
\]

where \( N_j \) was defined in (7). The variance of \( Z_{j-1} \) is given by the \((j,j)\)-th element of the variance matrix of \( M_j \); however, as calculated under the truncation imposed by \( N_j \), i.e., all the elements \( Z_0, \cdots, Z_{j-1} \) being bounded between \(-\epsilon 1, \epsilon 1\). This variance is given by \( \Sigma_{M,j}(j,j) \). Together with (10), this yields the desired expression.

Together, these two results provide analytic expressions for the communication frequency and average error covariance given any level \( \epsilon \).

**Example 2:** To examine the results, consider the following system

\[
x(k+1) = \begin{bmatrix} 0.5 & 0.2 \\ 0.2 & 0.9 \end{bmatrix} x(k) + w(k),
\]

\[
y(k) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} x(k)
\]

where \( w(k) = \begin{bmatrix} w_1(k) \\ w_2(k) \end{bmatrix} \) and the noise \( w_1(k) \) and \( w_2(k) \) are independent white, zero mean, Gaussian with covariance unity.

We set that automatic transmission happens after \( T = 3 \) time steps. For various values of \( \epsilon \) from 0 to 4, we evaluated the communication rate and error covariance as predicted by Theorems 1 and 2 respectively. We compared the analytic results to Monte Carlo simulations of the system. For each value of \( \epsilon \), we conducted 10,000 simulations and obtained the mean communication rate and error covariance. The comparison between analytical results and Monte Carlo experiments are shown in Fig. 5 and Fig. 6. Fig. 5 illustrates the comparison for communication rate and Fig. 6 presents the error covariance for each state. It can be seen that the analytic results match the Monte Carlo simulations very closely. We also see from Fig. 5 that the communication frequency will converge to the value of 0.25. Intuitively, for large threshold values, the states are cycling through the entire Markov chain and transmissions happen once every \( T \) steps.

\[
R_i \triangleq \begin{bmatrix} R_w & R_w A_0' & \cdots & R_w (A_0')^T \\ A R_w & A R_w A_0' + R_w & \cdots & A R_w (A_0')^2 + R_w A_0' \\ \vdots & \vdots & \ddots & \vdots \\ A^i R_w & \cdots & & \cdots 
\end{bmatrix}.
\tag{6}
\]
IV. MULTIPLE PLANTS SHARING THE NETWORK

Having characterized the performance with the event-triggered scheme for a single plant, we now consider the case of multiple plants that share a common network, as shown in Fig. 1.

A. The Discrete-time Discrete-state Model

Similar to the single plant case, we can define a discrete-time discrete-state model $\mathcal{M}$ for the $N$ plants case as follows. It should be noted that the following model for the case when multiple plants share the network may not be a Markov chain as for the single plant case in Section III. This is, roughly speaking, because transitions between the states in the model depend on not only local measurements but also on the contention for the use of the common network.

- The total number of nodes for the model is given by

$$N_s \triangleq (T + 1)T \cdots (T - N + 2)$$

and the states of this model are given by $\{X(k)\}_{k \geq 0} = \mathbf{m}$, where

$$\mathbf{m} \triangleq [m_1, m_2, \ldots, m_N]^\top \in \mathbb{R}^N$$

and $m_i \in \{0, 1, 2, \ldots, T\}$. In particular, if $m_i = j$, where $i \in \{1, 2, \ldots, N\}$ and $j \in \{0, 1, 2, \ldots, T\}$, then the last transmission for plant $i$ through the shared network occurred $j$ time steps ago. Thus, note that $m_i \neq m_j$ for all $i \neq j$ since we assume that the network does not permit simultaneous transmissions for any two plants.

- Define the probabilities

$$p_{mn} \triangleq \Pr[X(k + 1) = \mathbf{m}|X(k) = \mathbf{n}].$$

- The probabilities $p_{mn}$ of the model are determined by the specific CRM used. However, for any CRM, we can define the probability vector for the corresponding model as $\mathcal{P} \triangleq [P_1, P_2, \ldots, P_N]^\top \in \mathbb{R}^{N_s}$, where $P_i$ for $1 \leq i \leq N_s$ represents the probability of the state of the model being in the $i$-th node of the model described above. Further, we can define a matrix $\Delta \in \mathbb{R}^{N_s \times N_s}$ such that the relation

$$\Delta \mathcal{P} = \mathbf{b} \tag{13}$$

holds with a constant vector $\mathbf{b} = [1, 0, \ldots, 0]^\top \in \mathbb{R}^{N_s}$. For any CRM, the first row of $\Delta$ is given by $[1, 1, \ldots, 1]$ to indicate the balance equation (i.e. the sum of the probability of state of the model being in each node should be equal to one). The rest of the matrix $\Delta$ is determined by the specific CRM used and will be given later.

Example 3: We consider a simple example. Assume that $N = 2$ and $T = 1$. The model has $N_s = 2$ nodes as shown in Fig. 7. For notational convenience, we denote the left node as node 1 and the right node as node 2. The probability of the state of the model being in node $i$ ($1 \leq i \leq 2$) is denoted by $P_i$ such that $P_1 + P_2 = 1$. Note that $T = 1$ implies that each plant must transmit at every other time step. Thus, the probabilities $P_{12}$ and $P_{21}$ are both equal to 1, and we have

$$P_1 - P_2 = 0.$$ 

Thus, the relation (13) in this case reduces to the relation

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

from which we can obtain the unique solution that given by $P_1 = 0.5$ and $P_2 = 0.5$.

Remark 4: Although the solution of (13) in terms of $P$ is unique, the matrix $\Delta$ may not be unique since the relations between the nodes can be expressed in multiple ways. For instance, in Example 3, the second row of the matrix $\Delta$ can be chosen to be $[-1, 1]$. Further, the matrix $\Delta$ in (13) characterizes the relationship between the probabilities of the state of the model being in each node; however, it is different from the transition probability matrix since the elements of the matrix $\Delta$ are allowed to be negative, see e.g. [46].

For $T > 2$, a similar discrete-time discrete-state model can be defined by considering two more variables for each node indicating how long each plant has signaled that it wants to transmit.

B. Main Results

For pedagogical case, we concentrate on the case when $N = 2$, i.e. there are two plants that share the network. The general case is a straightforward extension at the cost of more notation. At every time step, the shared network may transmit information for plant 1 (denoted by $S_1$), or plant 2 (denoted by $S_2$), or not transmit information for any plant. The communication rate $P_i$ of the network to transmit information for plant $i$ is given by the probability of the state of the associated model being in the nodes with $X_i(k) = 0$, where $i = 1, 2$. The communication rate of the network denoted by $P$ is given by $P_0 = P_1 + P_2$, i.e. the probability of the state being in the nodes with $X_1(k) = 0$ and $X_2(k) = 0$. To clarify this, consider the following example.

Example 4: Consider a NCS with $N = 2$ plants over a shared medium. Assume the maximum delay that each plant

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

from which we can obtain the unique solution that given by $P_1 = 0.5$ and $P_2 = 0.5$.

Remark 4: Although the solution of (13) in terms of $P$ is unique, the matrix $\Delta$ may not be unique since the relations between the nodes can be expressed in multiple ways. For instance, in Example 3, the second row of the matrix $\Delta$ can be chosen to be $[-1, 1]$. Further, the matrix $\Delta$ in (13) characterizes the relationship between the probabilities of the state of the model being in each node; however, it is different from the transition probability matrix since the elements of the matrix $\Delta$ are allowed to be negative, see e.g. [46]. For $T > 2$, a similar discrete-time discrete-state model can be defined by considering two more variables for each node indicating how long each plant has signaled that it wants to transmit.
can tolerate is \( T = 2 \). We can define a discrete-time discrete-state model with 6 nodes as shown in Fig. 8. For notational convenience, we denote the nodes as nodes 1, 2, and so on starting from the left to the right. The probability of the state being in the node \( \ell \) (\( 1 \leq \ell \leq 6 \)) is denoted by \( P_{\ell} \) as in (13). Thus, the communication rate of the network to transmit information for plant \( S_1 \), i.e., \( P_1 \) is given by

\[
P_1 = P \begin{pmatrix} 0 \\ 1 \end{pmatrix} + P \begin{pmatrix} 0 \\ 2 \end{pmatrix} \triangleq P_1 + P_2. \tag{14}
\]

Finally, the communication rate of the network to transmit information for plant \( S_2 \), i.e. \( P_2 \) is given by

\[
P_2 = P \begin{pmatrix} 1 \\ 0 \end{pmatrix} + P \begin{pmatrix} 2 \\ 0 \end{pmatrix} \triangleq P_3 + P_5. \tag{15}
\]

Thus, the communication rate for the network is given by

\[
P_0 = P_1 + P_2 \triangleq P_1 + P_2 + P_3 + P_5. \tag{16}
\]

Note that \( P_{\ell} \) (\( 1 \leq \ell \leq 6 \)) can be obtained by the solution of equation (13).

We next characterize the matrix \( \Delta \) and evaluate the performance of event-triggered algorithms with static, random and dynamic schedulers through the model defined above. Let \( \mu_i(k) \) be the priority of the \( \ell \)-th plant at time \( k \).

**Static Scheduler:** We begin our analysis with the case when the plants have been given a fixed priority to access the network. Without loss of generality, we assume that \( \mu_1 < \mu_2 \) for any time \( k \), i.e. plant \( S_1 \) has higher priority than plant \( S_2 \).

**Theorem 3:** Consider the model defined above with a static scheduler for resolving contentions between two plants. The average communication rate for the network is given by \( P_1 + P_2 + P_3 + P_5 \), where \( P_i \) are determined by solving the equation \( \Delta P = b \), where the matrix \( \Delta \) is given by (17), where \( P^{(i)} \) represent the probabilities for the \( \ell \)-th (\( \ell = 1, 2 \)) plant and are given by

\[
\begin{align*}
P^{(1)}_{1} &= P \begin{pmatrix} 1 \\ 0 \end{pmatrix} + P \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \quad P^{(1)}_{2} = 1 - P^{(1)}_{10} - P^{(1)}_{00}, \quad \ell = 1, 2, \\
P^{(1)}_{3} &= P \begin{pmatrix} 1 \\ 0 \end{pmatrix} + P \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \quad P^{(1)}_{5} = 1 - P^{(1)}_{10} - P^{(1)}_{00}, \quad \ell = 1, 2.
\end{align*}
\]

**Proof:** We denote by \( P_i \) the probability of the state of the model being in the \( i \)-th (\( 1 \leq i \leq 6 \)) node in Fig. 8, where \( i \) is increasing from the left to the right. Based on the structure of the model in Fig. 8, we have the following relation

\[
P_3 = P_1 \times P^{(1)}_{10} + P_2 \tag{19}
\]

where \( p^{(1)}_{01} \) and \( p^{(2)}_{10} \) are given by (18). Consider the following two-step transition,

\[
\begin{align*}
0 &\rightarrow 1 \\
0 &\rightarrow 2
\end{align*}
\]

from which we have the following relation

\[
P_6 = P_1 \times P^{(1)}_{10} \times P^{(2)}_{12} P^{(2)}_{10} + P_2 \times P^{(1)}_{12} P^{(2)}_{01}. \tag{20}
\]

We can obtain the following relations in a similar manner,

\[
\begin{align*}
P_4 &= P_1 \times P^{(1)}_{01} P^{(2)}_{12} , \\
P_5 &= P_1 \times P^{(1)}_{01} P^{(2)}_{12} P^{(2)}_{00} + P_1 \times P^{(1)}_{01} P^{(2)}_{2} + P_2 \times P^{(1)}_{12} P^{(2)}_{00} , \\
P_7 &= P_1 \times P^{(1)}_{01} P^{(2)}_{12} P^{(2)}_{00} + P_1 \times P^{(1)}_{01} P^{(2)}_{2} + P_2 \times P^{(1)}_{12} P^{(2)}_{00} , \\
P_8 &= P_1 \times P^{(1)}_{01} P^{(2)}_{12} P^{(2)}_{00} + P_1 \times P^{(1)}_{01} P^{(2)}_{2} + P_2 \times P^{(1)}_{12} P^{(2)}_{00} .
\end{align*}
\]

(21)

In this way, we can represent the probability of the state of the model being in every node through that of node 1 and node 2 as in (19)-(21). This yields the matrix \( \Delta \) in (17), where \( p^{(i)}_{ij} \) are given by (18).

The performance metric is the covariance of the estimation error \( \Pi(k) = \Pi_1(k) + \Pi_2(k) \), where \( \Pi_1(k) = \mathbb{E}[e^{(dec)}_1(k) ]^\top \) denotes the covariance of the estimation error for plant \( S_1 \), where \( i = 1, 2 \).

**Theorem 4:** Consider the model defined in Fig. 8. Define \( \Xi_i \) (\( 1 \leq i \leq 4 \)) as

\[
\begin{align*}
\Xi_1 &= \Sigma(1) A_1 \Xi_1(k - 1) + w_1(k) - \bar{\varepsilon}, \\
\Xi_2 &= \Sigma(1) A_1 \Xi_2(k - 1) + 0, \\
\Xi_3 &= \Sigma(2) A_1 \Xi_3(k - 1) + w_2(k) - \bar{\varepsilon}, \\
\Xi_4 &= \Sigma(2) A_1 \Xi_4(k - 1) + 0.
\end{align*}
\]

(22)

Note that \( \Xi_i \) (\( 1 \leq i \leq 4 \)) are given by (18).

Theorem 4: To calculate the estimation error covariance for plant \( S_1 \), we use the relation

\[
\Pi_1(k) = \sum_{j=0}^{6} P_j \mathbb{E}[e^{(dec)}_1(k) ]^\top \mid X(k) = j. \tag{23}
\]

- For \( j = 1 \) and \( j = 2 \) (i.e. when the state of the model is in the node \( \{0, 1\} \) and \( \{0, 2\} \)), since the estimation error for the plant \( S_1 \), \( e^{(dec)}_1(k) \), is 0, we obtain \( \mathbb{E}[e^{(dec)}_1(k) ]^\top \mid X(k) = 1 = 0 \) and \( \mathbb{E}[e^{(dec)}_1(k) ]^\top \mid X(k) = 2 = 0 \).
- For \( j = 3 \), equation (17) implies the relation that \( \Pi_3 = \Pi_1 \Delta_{31} + \Pi_2 \). Further, the estimation error when the state of model is in node 3 is given by

\[
\begin{align*}
P_3 \mathbb{E}[e^{(dec)}_1(k) ]^\top \mid X(k) = 3 &= P_1 \Delta_{31} \text{var}\{w_1(k)\} + P_2 \text{var}\{w_1(k)\} \\
&= P_1 \Delta_{31} \Sigma_{M,1}^{(1)}(1, 1) + P_2 \text{var}\{w_1(k)\}.
\end{align*}
\]

(24)
\[ \Delta = \begin{bmatrix}
\frac{1}{2} + p_{12}^2 & 0 & 0 & 0 & 0 \\
1 - 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
p_{12}^2 & 1 & 0 & 0 & 0 \\
p_{12}^2 & 0 & 0 & 0 & -1 \\
\end{bmatrix}, \quad (17) \]

- For \( j = 4 \), the error covariance \( \mathbb{E}[e_{1 \text{dec}}^2(k)(e_{1 \text{dec}}^2(k))]^T | X(k) = 4 \) is \( \Sigma_{M,2}^{(1)}(1, 1) \).
- For \( j = 5 \), equation (17) implies the relation that \( \mathcal{P}_5 = \mathcal{P}_1 \mathcal{P}_{10}^2 \mathcal{P}_{12}^2 \mathcal{P}_{00} + \mathcal{P}_1 \mathcal{P}_{12}^2 \mathcal{P}_{2} + \mathcal{P}_2 \Delta_{52} \). Thus, the estimation error when the state is in the node 5 is given by

\[
\mathcal{P}_5 \mathbb{E}[e_{1 \text{dec}}^2(k)(e_{1 \text{dec}}^2(k))]^T | X(k) = 5 \\
= \mathcal{P}_2 \Delta_{52} \Xi_1 + \mathcal{P}_1 \mathcal{P}_{10}^2 \mathcal{P}_{12} \mathcal{P}_{00} \Sigma_{M,2}^{(1)}(1, 1) + \mathcal{P}_1 \mathcal{P}_{12}^2 \Xi_2
\]

where \( \Xi_1 \) and \( \Xi_2 \) are defined in (22).
- Finally, for \( j = 6 \), the error covariance is given by

\[
\mathcal{P}_6 \mathbb{E}[e_{1 \text{dec}}^2(k)(e_{1 \text{dec}}^2(k))]^T | X(k) = 6 \\
= \mathcal{P}_2 \Delta_{62} \Xi_1 + \mathcal{P}_1 \Delta_{61} \Sigma_{M,2}^{(1)}(1, 1)
\]

Combining (23)-(26), we obtain that \( \Pi_1 = \| \mathcal{E}_1 \|_1 \).

In a similar manner, we can calculate the estimation error covariance for plant \( S_2 \) for which the following equations hold.

\[
\mathbb{E}[e_{2 \text{dec}}^2(k)(e_{2 \text{dec}}^2(k))]^T | X(k) = 3 = 0 \\
\mathbb{E}[e_{2 \text{dec}}^2(k)(e_{2 \text{dec}}^2(k))]^T | X(k) = 5 = 0 \\
\mathbb{E}[e_{2 \text{dec}}^2(k)(e_{2 \text{dec}}^2(k))]^T | X(k) = 1 = R_{w2} \\
\mathcal{P}_2 \mathbb{E}[e_{2 \text{dec}}^2(k)(e_{2 \text{dec}}^2(k))]^T | X(k) = 2 = \mathcal{P}_1 \mathcal{P}_{00} \Xi_3 + \mathcal{P}_0 \Xi_4 \\
\mathbb{E}[e_{2 \text{dec}}^2(k)(e_{2 \text{dec}}^2(k))]^T | X(k) = 4 = \Xi_3 \\
\mathbb{E}[e_{2 \text{dec}}^2(k)(e_{2 \text{dec}}^2(k))]^T | X(k) = 6 = \Sigma_{M,2}^{(2)}(1, 1)
\]

Together with the probabilities from Theorem 3, this yields the desired expressions.

**Remark 5:** For the single plant case, we have \( e_{\text{dec}}^2(k) = 0 \) if \( X(k) = 0 \). Moreover, \( X(k) = j > 0 \) indicates that the estimation error in all the previous steps was less than \( \varepsilon \). As a result, the error covariance under the mode \( X(k) = j > 0 \) is simply

\[
\Pi(j) \equiv \operatorname{Pr}(X(k) = j) \mathbb{E}[e_{\text{dec}}^2(k)(e_{\text{dec}}^2(k))]^T | X(k) = j \\
= \operatorname{Pr}(X(k) = j) \Sigma_{M,j}(j, j)
\]

and the average estimation error covariance can be calculated as \( \sum_{j=1}^{T} \Pi(j) \). For the multiple plant case, however, any state of the model can be reached either due to the triggering of local event for each plant or due to the contention for the use of the network between plants. The error covariance is different depending on the state reached.

**Example 5:** Consider a network with two plants using the static scheduling policy. We assume that the plant \( S_1 \) has higher priority. The dynamics for the plant \( S_1 \) are given by

\[
x_1(k + 1) = 0.8x_1(k) + w_1(k) \\
y(k) = x_1(k)
\]

and the dynamics for the plant \( S_2 \) are given by

\[
x_2(k + 1) = 0.5x_2(k) + w_2(k) \\
y(k) = x_2(k)
\]

where the noises \( w_1 \) and \( w_2 \), the initial conditions \( x_1(0) \) and \( x_2(0) \) are independent white, zero mean, Gaussian random variables with covariance unity. We assume \( T = 2 \) and the triggering levels \( \varepsilon_1 = \varepsilon_2 = \varepsilon \). For various values of \( \varepsilon \) from 0 to 4, we evaluate the communication rate and error covariance as predicted by Theorems 3 and 4 respectively and compare the analytic results to Monte Carlo simulations of the system (10,000 simulations) in Fig. 9. It can be seen that the analytic results match the Monte Carlo simulations very closely.

**Random Scheduler:** In the random scheduling policy, any plant is allowed to access the network at every time \( k \) with a given probability. We assume that \( \mu_1 < \mu_2 \) with probability \( P_\alpha \) for any time \( k \) where \( 0 < P_\alpha < 1 \). In other words, \( P_\alpha \) is the probability for plant \( S_1 \) to win the contention when the both plants contend to use the network simultaneously.
Consider the model shown in Fig. 8. Similar to the case of static scheduler, one has to track the past states to calculate the probabilities \( p_{ij} \)'s. As an approximation, we calculate the probability \( p_{ij} \) by considering only one-step transition. The matrix \( \Delta \) in (13) for a random scheduler can then be approximated as (27). By solving equation (13) with \( \Delta \) given in (27), we can get the probability of the state of model being in each node. We can therefore characterize the average communication rate and the average error covariance along the same lines as for the case of static scheduler. We present the following result without proof.

**Theorem 5:** Consider the problem defined above with a random scheduler for resolving contentions between two plants. The average communication rate for the network is given by \( P_1 + P_2 + P_3 + P_5 \), where \( P_i \)'s are determined by solving the equation \( \Delta \mathcal{P} = \mathbf{b} \), with the matrix \( \Delta \) given by (27) and \( p_{ij}^{(t)} \)'s given by (18). Further, the steady state average error covariance \( \Pi \) can be approximated by \( \| \mathcal{E}_1 \|_1 + \| \mathcal{E}_2 \|_1 \), where \( \mathcal{E}_i \) \((i=1,2)\) is a \( 4 \times 1 \) vector given by

\[
\mathcal{E}_1 = \begin{bmatrix}
\mathbf{P}_{\text{MD}} & \mathbf{P}_{\text{MU}} & \mathbf{E}_7 & \mathbf{E}_6
\end{bmatrix}
\]

where \( \mathbf{E}_6 = \mathbf{P}_3 \mathbf{E}_2 + \mathbf{P}_{\text{MD}}(\mathbf{P}_{\text{M1}}(1,1) \mathbf{E}_6 + \mathbf{P}_{\text{M1}}^{(2)}(1,1)) \mathbf{P}_{\text{MU}} \), \( \mathbf{E}_7 = \mathbf{P}_7 \mathbf{E}_4 + \mathbf{P}_3 \mathbf{E}_4(1-\mathbf{P}_{\text{MU}})) \mathbf{P}_{\text{MD}}(1-\mathbf{P}_{\text{MU}})) \mathbf{E}_3 + \mathbf{P}_3 \mathbf{E}_3(1-\mathbf{P}_{\text{MU}}) \mathbf{P}_{\text{MD}}(1-\mathbf{P}_{\text{MU}})) \mathbf{E}_2 + \mathbf{P}_3 \mathbf{E}_2(1-\mathbf{P}_{\text{MU}}) \mathbf{P}_{\text{MD}}(1-\mathbf{P}_{\text{MU}})) \mathbf{E}_1 + \mathbf{P}_3 \mathbf{E}_1(1-\mathbf{P}_{\text{MU}}) \mathbf{P}_{\text{MD}}(1-\mathbf{P}_{\text{MU}})) \mathbf{E}_0 + \mathbf{P}_3 \mathbf{E}_0(1-\mathbf{P}_{\text{MU}}) \mathbf{P}_{\text{MD}}(1-\mathbf{P}_{\text{MU}})) \mathbf{E}_0").

Theorem 5: Consider the problem defined above with a random scheduler for resolving contentions between two plants. The average communication rate for the network is given by \( P_1 + P_2 + P_3 + P_5 \), where \( P_i \) are determined by solving the equation \( \Delta \mathcal{P} = \mathbf{b} \), with the matrix \( \Delta \) given by (27) and \( p_{ij}^{(t)} \)'s given by (18). Further, the steady state average error covariance \( \Pi \) can be approximated by \( \| \mathcal{E}_1 \|_1 + \| \mathcal{E}_2 \|_1 \), where \( \mathcal{E}_i \) \((i=1,2)\) is a \( 4 \times 1 \) vector given by

\[
\mathcal{E}_1 = \begin{bmatrix}
\mathbf{P}_{\text{MD}} & \mathbf{P}_{\text{MU}} & \mathbf{E}_7 & \mathbf{E}_6
\end{bmatrix}
\]

and

\[
\mathcal{E}_2 = \begin{bmatrix}
\mathbf{P}_1 \mathbf{R}_{\text{e_2}} & \mathbf{P}_3 \mathbf{R}_{\text{e_2}} & \mathbf{E}_7 & \mathbf{E}_6
\end{bmatrix}
\]

where \( \mathbf{E}_6 = \mathbf{P}_3 \mathbf{E}_2 + \mathbf{P}_{\text{MD}}(\mathbf{P}_{\text{M1}}(1,1) \mathbf{E}_6 + \mathbf{P}_{\text{M1}}^{(2)}(1,1)) \mathbf{P}_{\text{MU}} \), \( \mathbf{E}_7 = \mathbf{P}_7 \mathbf{E}_4 + \mathbf{P}_3 \mathbf{E}_4(1-\mathbf{P}_{\text{MU}})) \mathbf{P}_{\text{MD}}(1-\mathbf{P}_{\text{MU}})) \mathbf{E}_3 + \mathbf{P}_3 \mathbf{E}_3(1-\mathbf{P}_{\text{MU}}) \mathbf{P}_{\text{MD}}(1-\mathbf{P}_{\text{MU}})) \mathbf{E}_2 + \mathbf{P}_3 \mathbf{E}_2(1-\mathbf{P}_{\text{MU}}) \mathbf{P}_{\text{MD}}(1-\mathbf{P}_{\text{MU}})) \mathbf{E}_1 + \mathbf{P}_3 \mathbf{E}_1(1-\mathbf{P}_{\text{MU}}) \mathbf{P}_{\text{MD}}(1-\mathbf{P}_{\text{MU}})) \mathbf{E}_0 + \mathbf{P}_3 \mathbf{E}_0(1-\mathbf{P}_{\text{MU}}) \mathbf{P}_{\text{MD}}(1-\mathbf{P}_{\text{MU}})) \mathbf{E}_0").

Example 6: Consider the setup in Example 5. We consider the case when the network uses the random scheduling policy. We set \( P_0 = 0.7 \) in our simulation. The simulation results for the average communication rate and the error covariance are compared with those obtained by using the approximation discussed above are shown in Fig. 10. It can be seen that the approximate results match closely to the Monte Carlo simulations.

**Dynamic Scheduler:** With a dynamic scheduling policy, the priority order depends on the error \( e_{i_{\text{comp}}} \) as defined in (3). If both plants contend to use the network, then the network grants the one with the larger error \( |e_{i_{\text{comp}}}| \) the use of the network at time \( k \). In other words, \( \mu_1(k) < \mu_2(k) \) if \( |e_{1_{\text{comp}}}(k)| > |e_{2_{\text{comp}}}(k)| \).

Define the conditional probability \( P_d \) as follows,

\[
P_d \triangleq \Pr(|e_{1_{\text{comp}}}| > |e_{2_{\text{comp}}}|, |e_{1_{\text{comp}}}| > \varepsilon, |e_{2_{\text{comp}}}| > \varepsilon)
\]

where the dependence of the errors \( e_{i_{\text{comp}}}(k) \) on time \( k \) is omitted for notational convenience. \( P_d \) can be exactly evaluated through Gaussian integrals because the errors \( e_{i_{\text{comp}}} \) are Gaussian random variables as defined in (5). However, for simplicity, we can approximate \( P_d \) by

\[
P_d \approx \lambda \triangleq \Pr(|e_{1_{\text{comp}}}| > |e_{2_{\text{comp}}}|).
\]

By symmetry, \( \lambda \) is approximately given by 0.5. Note that by definition \( P_d \) as defined in (28) is the probability that the plant \( S_1 \) has higher priority when contending with the plant \( S_2 \) to use the network. Comparing with the case when a random scheduler is used, \( P_d \) is the probability of the plant \( S_1 \) to win the contention. Thus, we have \( P_d \approx P_\alpha \). Then the average communication rate by using the dynamic scheduler can be approximated by using the random scheduler with \( P_\alpha = 0.5 \).

**Theorem 6:** Consider the problem defined above with a dynamic scheduler for resolving contentions between two plants. The average communication rate for the network is given by \( P_1 + P_2 + P_3 + P_5 \), where \( P_1 \) are determined by...
solving the equation $\Delta P = b$, where the matrix $\Delta$ is the same as in Theorem 5 with $P_0 = \frac{1}{T}$.

Example 7: Consider again the setup in Example 5. We consider the case when the network uses dynamic scheduling policy. The simulation results by using the approximation discussed above are shown in Fig. 11. It can be seen from Fig. 11 that the approximate result for the average communication rate for plant $S_1$ (in the top plot) and for plant $S_2$ (in the bottom plot) match closely to the Monte Carlo simulations. □

Remark 6: While the average communication rate for the network is well approximated by the expressions for the random scheduler by setting $P_0 = \frac{1}{T}$, the error covariance cannot be obtained in a similar manner. This is because, there exists an additional condition on comparing the magnitudes of $e^{\text{comp}}_1$ and $e^{\text{comp}}_2$. Thus, the calculation for the error covariance becomes more complicated than the results in Theorem 5.

V. RESULTS FOR FULLY UTILIZED NETWORK

In this section, we study the estimation error covariance when the network is fully utilized, i.e. the network transmits information at each time step $k$, either for the plant $S_1$ or for the plant $S_2$. In this special case that is of high interest to practical applications, we can show analytically that event-triggered methods with static and random scheduling policies may lead to larger error covariance than time-triggered method with periodic scheduling policy.

Time-triggered algorithm: We consider two possible packet sequences $S_1 = \{1, 2, 1, 2 \cdots \}$ and $S_2 = \{2, 1, 2, 1 \cdots \}$. If $A_1 = A_2$ and $R_{w_1} = R_{w_2}$, then the two round robin schedules $S_1$ and $S_2$ are both optimal. Without loss of generality, we consider the schedule $S_1$.

Event-triggered algorithm: If the triggering level $\varepsilon_1$ in (3) approaches zero, then event (3) is triggered at every time step $k$. Thus, the network is fully utilized. We consider static and random scheduling policies for the event-triggered method as discussed in Section IV.

1) Static scheduler when $\varepsilon_1$ approaches zero. Without loss of generality, we assume that $\mu_1 < \mu_2$. The discrete-time discrete-state model for this case has $T+1$ $(T \geq 1)$ states as shown in Fig. 12.

2) Random scheduler when $\varepsilon_1$ approaches zero. We assume that $\mu_1 < \mu_2$ with probability $P_0$ for any time $k$. The discrete-time discrete-state model for this case is presented in Fig. 13 with $2T$ states.

Theorem 7: Consider the scheduling policies discussed above when the network is fully utilized. Let $J_T$, $J_S$ and $J_R$ be the steady state average error covariance $\Pi = \lim_{k \to \infty} \Pi(k)$ under the round-robin schedule $S_1$, static schedule and random schedule, respectively. Then,

$$J_T = \frac{1}{2}(R_{w_2} + R_{w_1}), \quad (30)$$

$$J_S = \frac{1}{T+1} R_{w_1} + \frac{1}{T+1} \sum_{i=1}^{T} A_i^2 R_{w_2} (A_i^T)^2, \quad (31)$$

$$J_R = \frac{\rho}{1 - P_0} \sum_{i=1}^{T} (1 - P_0)^{i} \sum_{j=0}^{i-1} A_i^2 R_{w_2} (A_i^T)^2 + \frac{\rho}{P_0} \sum_{i=1}^{T} P_0^i \sum_{j=0}^{i-1} A_i^2 R_{w_2} (A_i^T)^2 \quad (32)$$

where $\rho = \frac{P_0 (1-P_0)}{1-P_0^2 + (1-P_0)^T+1}$. Further, if $R_{w_1} = R_{w_2}$, then $J_T < J_S$ and $J_T < J_R$.

Proof: First, we consider the time-triggered algorithm with round robin schedule $S_1$. At the $i$-th $(i=1, 2)$ estimator, the estimation error evolves as

$$e^{\text{dec}}_1(k) = \begin{cases} 0, & \text{if } k = 2j+i, \quad \forall 0 \leq j \in \mathbb{Z} \\ w_i(k-1), & \text{otherwise} \end{cases}$$

Thus, the steady state average error covariance given by (4) can be calculated as $J_T = \frac{1}{T+1}(R_{w_2} + R_{w_1})$. Next, we consider the event-triggered algorithm with the static scheduler. Based on the structure of the model as shown Fig. 12 and the fact that the probabilities $p_{ij}$ are all equal to one, we can obtain that the probability of the state being in each node is equal to $\frac{1}{T+1}$. The average error covariance for $S_1$ is simply $\frac{1}{T+1} R_{w_1}$. The estimation error for $S_2$ under the node $[0,j]^T$ for $1 \leq j \leq T$ is given by $Z_j$. Thus, the error covariance for $S_2$ can be calculated as

$$\Pi_S = \sum_{j=1}^{T} Pr(X(k) = [0;j]) E\{Z_{j-1} Z_{j-1}^T\}$$

$$= \sum_{j=1}^{T} \frac{1}{T+1} \sum_{i=0}^{j-1} A_i^2 R_{w_2} (A_i^T)^2.$$
Therefore, \( J_S \) is given by \( J_S = T \Pi + \frac{1}{T+1} R_{w_1} \), which yields the desired expression. To calculate the error covariance for event-triggered algorithm with the random scheduler, we consider the model as shown in Fig. 13. Based on the structure of the model, we can obtain that the probability for the state of the model being in the node \([0,1]^T\) equals to \( \rho \). Then similar to the case of static scheduler, the error covariance for \( S_2 \) can be calculated as

\[
\Pi_R = \sum_{j=1}^{T} Pr(X(k) = [0; j])E\{Z_{j-1}Z_{j-1}^T\} = \sum_{j=1}^{T} \rho P{j-1} \sum_{i=0}^{j-1} A_2^i R_{w_2} (A_2^i)^T.
\]

Note that the error covariance for \( S_1 \) can be calculated in a similar manner. This yields the desired expression for \( J_R \). Now we consider the case when \( R_{w_1} = R_{w_2} = R \). We have \( J_T = R \). Then, based on (30), (31) and (32), we obtain

\[
J_S > 1 - \frac{1}{T+1} R_{w_1} + \frac{T}{T+1} R_{w_2} = R,
\]

\[
J_R > \rho \sum_{i=0}^{T} \left( (1 - P_\alpha)^{i-1} R_{w_1} + P_\alpha^{i-1} R_{w_2} \right) = \rho R \left( \frac{1 - (1 - P_\alpha)^T}{P_\alpha} + \frac{1 - P_\alpha^T}{1 - P_\alpha} \right) = R.
\]

Therefore, we have \( J_S > J_T \) and \( J_R > J_T \).

**Remark 7:** We have shown that \( J_S > J_T \) and \( J_R > J_T \), i.e. the time-triggered method leads to smaller error covariance than the event-triggered method with static and random scheduling policies when the level \( \epsilon_i = 0 \). This is consistent with our simulation results as shown in the bottom plot of Fig. 2.

VI. CONCLUSION AND FUTURE DIRECTION

This paper studies state estimation for multiple plants across a shared communication network. Each plant transmits information through the common network according to either a time-triggered or an event-triggered rule. For a time-triggered algorithm combined with TDMA, each plant uses the network according to a schedule that is decided \textit{a priori}. For an event-triggered algorithm with CSMA, each plant is assumed to access the network based on one of the following scheduling strategies, i.e. static, random or dynamic schedulers. Performance in terms of the communication rate and estimation error covariance is analytically characterized. Using these characterizations, we show that for the same average communication rate, event-triggered schemes may perform worse than time-triggered schemes when the effect of communication strategies is explicitly considered. For future work, we are interested in characterizing system performance when ‘hybrid’ scheduling policies are considered, when the use of either a time-triggered algorithm or an event-triggered algorithm is adaptive to the system performance and network constraints.

REFERENCES


