

Estimation over Communication Networks: Performance Bounds and Achievability Results

A. F. Dana, V. Gupta, J. P. Hespanha, B. Hassibi, R. M. Murray

Abstract—This paper considers the problem of estimation over communication networks. Suppose a sensor is taking measurements of a dynamic process. However the process needs to be estimated at a remote location connected to the sensor through a network of communication links that drop packets stochastically. We provide a framework for computing the optimal performance in the sense of expected error covariance. Using this framework we characterize the dependency of the performance on the topology of the network and the packet dropping process. For independent and memoryless packet dropping processes we find the steady-state error for some classes of networks and obtain lower and upper bounds for the performance of a general network. We also illustrate how this framework can be used in the synthesis of networks for the purpose of estimation. Finally we find a necessary and sufficient condition for the stability of the estimate error covariance for general networks with spatially correlated and Markov type dropping process. This interesting condition has a max-cut interpretation.

I. INTRODUCTION AND MOTIVATION

In recent years, systems comprising of multiple sensors cooperating with each other have received wide-spread interest (see, e.g., [1], [2]). Although such systems admittedly have a higher complexity than the strategy of using only one sensor, the increased accuracy often make these systems worthwhile. From an estimation and control perspective, such systems present many new challenges, such as dealing with data delay or data loss imposed by the communication links, fusion of data emerging from multiple nodes and so on. Most of these issues arise because of the tight coupling between the estimation and control tasks that depend on the sensed data and the communication channel effects that affect the transmission and reception of data. Communication links introduce many potentially detrimental phenomena, such as quantization error, random delays, data loss and data corruption to name a few. It is imperative to understand and counteract the effects of the communication channels.

Motivated by this, there has been a lot of work done on estimation and control over networks of communication links (see, e.g., [3], [4] and the references therein). Beginning with the seminal paper of Delchamps [5], quantization effects have been variously studied both in estimation and control context by Tatikonda [6], Nair and Evans [7], Hespanha et al [8] and many others. The effect of delayed packet delivery using various models for network delay has also been considered by many researchers.

In this work, we focus on estimation across a network of communication links that drop packets. We consider a dynamical process evolving in time that is being observed

by a sensor. The sensor needs to transmit the data over a network to a destination node. However the links in the network stochastically drop packets. Prior work in this area has focused on studying the effect of packet drops by a single link in an estimation or control problem. Assuming certain statistical models for the packet drop process, stability of such systems was analyzed, e.g., in [10], [11] and the control performance by Seiler in [10] and by Ling and Lemmon in [12]. Approaches to compensate for the data loss were proposed by Nilsson [9], Hadjicostis and Touri [13], Ling and Lemmon [12], [14], Azimi-Sadjadi [15], Sinopoli et al. [16] and Imer et al. [17]. Sinopoli et al. [18] also considered the problem of optimal estimation across a packet-dropping link that drops packet in an i.i.d. fashion and obtained bounds on the expected error covariance. Most of the above designs aimed at designing a packet-loss compensator. The compensator accepts those packets that the link successfully transmits and produces an estimate for the time steps when data is lost. If the estimator is used inside a control loop, the estimate is then used by the controller. A more general approach is to design an encoder and a decoder for the communication link. This was considered for the case of a single communication link in [19]. It was demonstrated that using encoders and decoders can improve both the stability margin and the performance of the system.

For general networks, the problem is much more complicated than the case of a single communication link since potentially there are multiple paths from the source to the destination. Recent work [20] identified optimal information processing schemes that should be followed by the nodes of the network to allow the sink to calculate the optimal estimate at every time step. That work also identified the condition on the network for the estimate error covariance to be stable under this algorithm. In this paper, we calculate the performance of such a strategy. This performance also provides a lower bound on the performance that can be achieved by any other scheme (e.g., transmitting measurements without any processing). We also generalize the condition for stability of the estimate error covariance from the independent and memoryless packet drop processes to ones that are described by Markov chains or are spatially correlated across the network. We provide a mathematical framework for evaluating the performance for a general network and provide expressions for networks containing links in series and parallel. We also provide lower and upper bounds for the performance over general networks. As an example of how such results can be used for *synthesis* of

networks to improve estimation performance, we provide a simple example in which the optimal number of relay nodes to be placed is identified. Simulation results are provided to illustrate the results. The better estimation performance can also translate to better control performance if the estimate is used for control purposes [19], [16], [17].

The paper is organized as follows. In the next section, we set up the problem and state the various assumptions. Then, we provide the mathematical framework needed to calculate the steady-state performance and introduce the concept of latency. We show how to evaluate latency for series and parallel networks. We then provide bounds for the performance for a general network. We conclude with some remarks and avenues for future work.

II. PROBLEM SETUP

Consider a process evolving in discrete-time as

$$x_{k+1} = Ax_k + w_k, \quad (1)$$

where $x_k \in \mathcal{R}^n$ is the process state and w_k is the process noise modeled white and Gaussian with mean zero and covariance matrix Q . The process is observed using a sensor that generates measurements of the form

$$y_k = Cx_k + v_k, \quad (2)$$

where $v_k \in \mathcal{R}^m$ is the measurement noise also assumed to be white, Gaussian with mean zero and covariance R . Furthermore, the noises v_k and w_k are assumed to be independent of each other. We consider the scenario in which the process needs to be estimated in the minimum mean square error (MMSE) sense at a remote point denoted by a destination node d . We assume that the sensor (denoted by s) and the destination node d are connected via a communication network. The communication graph is represented by a directed graph \mathcal{G} with node set \mathcal{V} (that contains, in particular, s and d) and link set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. The link $e = (u, v)$ models a communication channel between node u and node v . For any node $i \in \mathcal{V}$, the set of outgoing edges corresponds to the links along which the node can transmit messages while the set of incoming edges corresponds to the links along which the node receives messages. We denote the set of in-neighbors of node v by $\mathcal{N}(v)$.

The communication links are modeled using a packet erasure model. The links take in as input a finite vector of real numbers. For every link, at each time-step, a packet is either dropped or received completely at the output node. In this paper, we assume independent and memoryless packet drop processes, i.e., the probability of dropping a packet on link $e \in \mathcal{E}$ is given by p_e independent of other links and time. We ignore quantization issues, data corruption or random delays. We also assume a global clock so that each node is synchronized. We further assume that each node can listen to all the messages over the different incoming links without interference from each other.¹

¹This property can be achieved by using a division multiple access scheme like FDMA, TDMA, CDMA etc.

The operation of different nodes in the network at every time-step k can be described as follows:

- 1) Each node computes a function of all the information it has access to at that time.
- 2) It transmits the function on all the outgoing edges. We allow some additional information in the message that tells us the time step j such that the function that the node transmits corresponds to the state x_j . The destination node calculates the estimate of the current state x_k based on the information it possesses.
- 3) Every node observes the signals from all the incoming links and updates its information set for the next time step. For the source node, the message it receives at time step k corresponds to the observation y_k .

The timing sequence we have specified leads to strictly causal estimates. At time step k , the function that the source node transmits depends on measurements y_0, y_1, \dots, y_{k-1} . Further even if there were no packet drops, if the destination node is l hops away from the source node, its estimate for the state x_k at time k can only depend on measurements $y_0, y_1, \dots, y_{k-l-1}$ till time $k-l-1$.

In [20], the optimal information processing strategy at each node in the network that results in MMSE estimate at the destination node was identified. We restate the algorithm in this paper for the sake of completeness. We derive the necessary and sufficient condition for stability of the expected estimate error covariance for the algorithm using an alternate method. The chief contribution of this paper is to find the MMSE steady-state error at the destination node and to identify its dependency on the topology of the network and the packet drop probabilities. The framework developed here also allows us to generalize the condition for stability of the estimate error covariance to more complicated packet dropping models.

III. OPTIMAL ENCODING AND DECODING

We now describe an algorithm \mathcal{A} , originally developed in [20], that achieves the optimal performance at the expense of constant memory and transmission (modulo the transmission of the time stamp). At each time step k , every node v takes the following actions:

- 1) Calculate its estimate \hat{x}_k^v of the state x_k based on any data received at the previous time step $k-1$ and its previous estimate. The estimate can be computed using a switched linear filter as follows. The source node implements a Kalman filter and updates its estimate at every time step with the new measurement received. Every other node checks the time-stamps on the data coming on the incoming edges. The time-stamps correspond to the latest measurement used in the calculation of the estimate being transmitted. Let the time-stamp for node $u \in \mathcal{V}$ at time k be $t_u(k)$. Also let $D_{uv}(k)$ be the binary random variable describing the packet drop event on link $(u, v) \in \mathcal{E}$ at time k . $D_{uv}(k)$ is '0' if the packet is dropped on link

(u, v) at time k and ‘1’ otherwise. For a network with independent and memoryless packet drops, $D_{uv}(k)$ is distributed according to Bernoulli with parameter p_{uv} . We define $D_{uu}(k) = 1$. Node v updates its time-stamp using the relation

$$t_v(k) = \max_{u \in \mathcal{N}(v) \cup \{v\}} D_{uv}(k) t_u(k-1). \quad (3)$$

Note that for the source node s , $t_s(k) = (k-1)$ for all $k \geq 1$. Suppose that the maximum of (3) is given by node $n \in \mathcal{N}(v) \cup \{v\}$. The node v updates its estimate as $\hat{x}_k^v = A \hat{x}_{k-1}^n$.

- 2) Affix a time stamp corresponding to the last measurement used in the calculation of its estimate and transmit the estimate on the outgoing edges.
- 3) Receive any data on the incoming edges and store it for the next time step.

Proposition 1: (Optimality of Algorithm \mathcal{A}): The algorithm \mathcal{A} is optimal in the sense that it leads to the minimum possible error covariance at any node at any time step. The proof of the above theorem is provided in [20]. We should remark that the above result holds for *any* packet drop sequence. Thus the algorithm \mathcal{A} is optimal for any packet drop pattern, i.e., irrespective of whether the packet drops are occurring in an i.i.d. fashion or are correlated across time or space or if packet drops are time-varying or even adversarial in nature. The algorithm also does not assume any knowledge of the statistics of the packet drops at any of the nodes.

IV. STEADY-STATE ERROR COVARIANCE CALCULATION

In this section we calculate the steady-state estimate error covariance at any node using the algorithm \mathcal{A} . For node v and time k , $t_v(k)$ denotes the time-stamp of the most recent observation used in estimating x_k . This time-stamp evolves according to (3). The expected estimation error covariance at time k at node v can be written as

$$\begin{aligned} & \mathbb{E} |x_k - \hat{x}_{k|t_v(k)}|^2 \\ &= \sum_{l=0}^k \Pr(t_v(k) = k-l-1) \mathbb{E} |x_k - \hat{x}_{k|k-l-1}|^2 \\ &= \sum_{l=0}^k \Pr(l_v(k) = l) \left[A^l P_{k-l-1} A^l + \sum_{j=0}^{l-1} A^j Q A^j \right] \end{aligned} \quad (4)$$

where P_k is the estimation error covariance of x_k based on $\{y_0, y_1, \dots, y_{k-1}\}$ and $l_v(k) = k-1-t_v(k)$ is the *latency* for node v at time k . P_k evolves according to a Riccati recursion. The above equation gives the expected estimation error covariance for a general network with any packet dropping process. The effect of the packet dropping process appears in the distribution of the latency $l_v(k)$. We consider the steady-state error covariance in the limit as k goes to infinity, i.e.,

$$P^\infty = \lim_{k \rightarrow \infty} \mathbb{E} |x_k - \hat{x}_{k|t_v(k)}|^2.$$

If P^∞ is bounded, we will say that the estimate error is *stable*; otherwise it is *unstable*. As we can see from (4), the stability of the system depends on how fast the probability distribution of the latency decreases.

For now we focus on an i.i.d. packet drop model. At any time and for any link $e = (u, v)$, the packet is dropped with probability p_e independent of time k and of other links of the network. From (3), $D_{uv}(k)$ indicates the event of packet drop for link $e = (u, v)$. For each link e and time k let $Z_e(k)$ be the difference between k and the most recent successful transmission on link e preceding time k , i.e.,

$$Z_e(k) = \min\{j \geq 1 | D_{uv}(k+1-j) = 1\}$$

Using the definition of $Z_e(k)$, the last time that any message is received at node v from link (u, v) is $k - Z_{uv}(k) + 1$ and that message has time-stamp $t_u(k - Z_{uv}(k))$. Then (3) can be written in terms of $Z_e(k)$ as

$$t_v(k) = \max_{u \in \mathcal{N}(v)} t_u(k - Z_{uv}(k))$$

Now $Z_e(k)$ is distributed as a truncated geometric random variable with $\Pr(Z_e(k) = i) = (1-p_e)p_e^{i-1} \quad \forall k > i \geq 1$ and $\Pr(Z_e(k) = k) = 1 - \sum_{i=1}^{k-1} \Pr(Z_e(k) = i)$. We can get rid of the truncation by extending the definition of $t_u(k)$ for negative k 's as well. For $k < 0$ we define $t_u(k) = 0$. Then, for instance, for the source node s we have $t_s(k) = (k-1)^+$, where $x^+ = \max\{0, x\}$. In general, we have

$$t_v(k) = \max_{u \in \mathcal{N}(v)} t_u(k - Z_{uv}),$$

where Z_e 's are independent random variables distributed according to a geometric distribution, i.e., $\Pr(Z_e = i) = (1-p_e)p_e^{i-1} \quad \forall i \geq 1$. Further note that Z_e 's do not depend anymore on k . Solving the above recursive formula, we can write $t_v(k)$ in terms of the time-stamp at the source node (i.e., $(k-1)^+$) as

$$t_v(k) = \max_{P: \text{an } s\text{-}v \text{ path}} (k-1 - \sum_{e \in P} Z_e)^+, \quad (5)$$

where the maximum is taken over all paths P in the graph \mathcal{G} from source s to the node v . Therefore the latency at node v can be written as

$$l_v(k) = k-1-t_v(k) = \min\{k-1, \min_{P: \text{an } s\text{-}v \text{ path}} (\sum_{e \in P} Z_e)\}.$$

From the above equation it can be seen that as $k \rightarrow \infty$ the distribution of $l_d(k)$ approaches the distribution of l_d defined as

$$l_d = \min_{P: \text{an } s\text{-}d \text{ path}} (\sum_{e \in P} Z_e). \quad (6)$$

We refer to l_d as the *steady-state latency* of the network. Therefore, the steady-state error covariance can be written as

$$P^\infty = \sum_{l=0}^{\infty} \Pr(l_d = l) \left[A^l P^* A^l + \sum_{j=0}^{l-1} A^j Q A^j \right], \quad (7)$$

where P^* is the steady-state estimation error covariance of x_k based on $\{y_0, y_1, \dots, y_{k-1}\}$ and is the solution to the Discrete Algebraic Riccati Equation (DARE)

$$P^* = AP^*A^T + Q - AP^*C^T(CP^*C^T + R)^{-1}CP^*A^T.$$

We assume that the system $\{A, Q^{\frac{1}{2}}\}$ is stabilizable. Hence the rate of convergence of P_k to P^* is exponential [21] and the substitution of P^* for P_{k-l} in (4) does not change the steady-state error covariance.

Let us define the generating function of the complementary density function $G(X)$ and the moment generating function $F(X)$ of the steady state latency l_d

$$\begin{aligned} G(X) &= \sum_{l=0}^{\infty} \Pr(l_d \geq l+1)X^l \\ F(X) &= \sum_{l=0}^{\infty} \Pr(l_d = l)X^l, \end{aligned} \quad (8)$$

where X is a matrix. On vectorizing (7) we obtain

$$\text{vec}(P^\infty) = F(A \otimes A)\text{vec}(P^*) + G(A \otimes A)\text{vec}(Q),$$

where $A \otimes B$ is the Kronecker product of matrices A and B . Using the fact that $F(X) = (X - I)G(X) + I$ yields

$$\text{vec}(P^\infty) = ((A \otimes A - I)G(A \otimes A) + I)\text{vec}(P^*) + G(A \otimes A)\text{vec}(Q). \quad (9)$$

We can see from (9) that the performance of the system depends on the value of $G(X)$ evaluated at $X = A \otimes A$. In particular, the system is stable if and only if $G(X)$ is bounded at $A \otimes A$. Since $G(X)$ is a power series, this is equivalent to the boundedness of $G(x)$ (evaluated for a scalar x) at the square of the norm of the eigenvalue of A with the largest norm. We summarize the result of the above arguments in the following theorem.

Theorem 1: Consider the system model described in Section II. Let the packet drops are independent from one time step to the next and across links. Then the minimum expected steady-state estimation error covariance is given by (9). Furthermore, the error covariance is stable, iff $|\lambda_{\max}(A)|^2$ lies in the region of convergence of $G(x)$ where $\lambda_{\max}(A)$ is the maximum-norm eigenvalue of A .

The above theorem allows us to calculate the steady state expected error covariance for any network as long as we can evaluate the function $G(X)$ for that network. We now consider some special networks and evaluate the performance explicitly. We start with a network consisting of links in series, or a line network.

1) *Line Networks:* In this case, the network consists of only one path from the source to the destination. Since the drops across different links are uncorrelated, the variables Z_e 's are independent. Thus

$$F(X) = \mathbb{E}[X^{l_d}] = \mathbb{E}[X^{\sum_e Z_e}] = \prod_e \mathbb{E}[X^{Z_e}].$$

Since Z_e is a geometric random variable, $\mathbb{E}[X^{Z_e}] = (1 - p_e)X(I - p_eX)^{-1}$ provided that $\lambda_{\max}(X)p_e < 1$. Therefore,

$$F(X) = \mathbb{E}[X^{l_d}] = \prod_e \left[(1 - p_e)X(I - p_eX)^{-1} \right].$$

Using partial fractions, we can easily show that

$$G(X) = \sum_{i=0}^{n-1} X^i + X^n \sum_e c_e \frac{p_e}{1 - p_e} (I - p_eX)^{-1},$$

where $c_e = (\prod_{e' \neq e} (1 - \frac{p_{e'}}{p_e}))^{-1}$. Therefore the cost can be written as

$$\text{vec}(P^\infty) = \prod_e \left[(A \otimes A) \left(\frac{I - p_e(A \otimes A)}{1 - p_e} \right)^{-1} \right] \text{vec}(P^*) + G(A \otimes A)\text{vec}(Q). \quad (10)$$

Remark 1: We can see from the above argument that the system is stable if for every link e we have $p_e |\lambda_{\max}(A)|^2 < 1$ or equivalently $\max_e p_e |\lambda_{\max}(A)|^2 < 1$. This matches with the condition in [20]

Remark 2: For the case that some of p_e 's are equal, a different partial fraction expansion applies. In particular for the case when there are n links all with the erasure probability p , we obtain

$$\begin{aligned} \text{vec}(P^\infty) &= (A \otimes A)^n \left(\frac{I - p(A \otimes A)}{1 - p} \right)^{-n} \text{vec}(P^*) \\ &+ \sum_{i=0}^{n-1} \left[\frac{p}{1 - p} (A \otimes A)^n \left(\frac{I - p(A \otimes A)}{1 - p} \right)^{-i-1} \right] \text{vec}(Q) \\ &+ \sum_{i=0}^{n-1} (A \otimes A)^i \text{vec}(Q). \end{aligned}$$

When there is only one link between the source and the destination the steady state error covariance will be the solution to the Lyapunov equation

$$P^\infty = \sqrt{p}AP^\infty\sqrt{p}A + (Q + (1 - p)AP^*A).$$

This matches with the expression derived in [19] using Markov jump linear system theory.

2) *Network of Parallel Links:* Now consider a network with one sensor connected to a destination node through n links with probabilities of packet drop p_1, \dots, p_n . In this case the steady state latency is given by $l_d = \min_{1 \leq i \leq n} (Z_i)$. Since the minimum of independent geometrically distributed random variables with parameters $\{p_i\}$ is itself geometrically distributed with parameter $p_{eq} = \prod_i p_i$, $G(X)$ can be written as

$$G(X) = (I - \prod_i p_i X)^{-1}$$

Thus the steady-state error can be evaluated using (9). Note that the region of convergence of $G(X)$ enforces $\prod_i p_i |\lambda_{\max}(A)|^2 < 1$ for stability which again matches with the condition in [20].

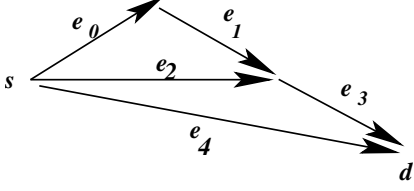


Fig. 1. Example of a network of combination of parallel and serial links

3) *Arbitrary Network of Parallel and Serial Links*: Using similar arguments as in previous sections, we can find the steady-state error covariance of any network of parallel and serial links. These networks are derived from the parallel and serial concatenations of sub-networks. The following two simple rules can give the steady state error of any network of parallel and series links. Let $l_d(\mathcal{G})$ denote the steady-state latency function of network \mathcal{G} . Also given two subnetworks \mathcal{G}_1 and \mathcal{G}_2 , denote their series combination by $\mathcal{G}_1 \oplus \mathcal{G}_2$ and their parallel combination by $\mathcal{G}_1 \parallel \mathcal{G}_2$.

- 1) For series connection, we have $l_d(\mathcal{G}_1 \oplus \mathcal{G}_2) = l_d(\mathcal{G}_1) + l_d(\mathcal{G}_2)$. Using the independence of latency functions of the two sub-networks, the generating function of the network is given as

$$G(X) = (X - I)G_1(X)G_2(X) + G_1(X) + G_2(X).$$

- 2) For parallel connection, we have $l_d(\mathcal{G}_1 \parallel \mathcal{G}_2) = \min\{l_d(\mathcal{G}_1), l_d(\mathcal{G}_2)\}$. Using the independence of $l_d(\mathcal{G}_1)$ and $l_d(\mathcal{G}_2)$, the complementary distribution function of $l_d(\mathcal{G})$ can be written as the product of the functions for \mathcal{G}_1 and \mathcal{G}_2 .

As an example consider the network depicted in Fig. 1. In this case the network \mathcal{G} can be written as $((\mathcal{G}_0 \oplus \mathcal{G}_1) \parallel \mathcal{G}_2) \oplus \mathcal{G}_3 \parallel \mathcal{G}_4$ where each of the sub-networks \mathcal{G}_i is just a link with probability of packet drop p . Using the above rules and denoting the moment generating function of the parallel combination of any network (with $G(X)$) and a link with probability of packet drop p by $\mathcal{L}_p(G)(X)$, the generating function of the network can be written as

$$G(X) = \mathcal{L}_p(\mathcal{L}_p(G_0 * G_1) * G_3)(X)$$

where $G_i(X) = (I - pX)^{-1}$, $i = 0, 1, 3$ is the generating function for the i -th link and for each function $F(\cdot)$, \mathcal{L}_p is an operator that such that $\mathcal{L}_p(F)(X) = F(pX)$. The steady state error covariance can thus be evaluated.

4) *Networks with Arbitrary Topology*: Finding the distribution of the steady-state latency l_d of a general network is not an easy task. However, we can provide upper and lower bounds on the performance. We first mention the following intuitive lemma without proof.

Lemma 1: Let $P^\infty(\mathcal{G}, \{p_e, e \in \mathcal{E}\})$ denote the expected steady-state error of a system with communication network represented by graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and probabilities of packet drop p_e , $e \in \mathcal{E}$. Then the expected steady-state error is non-increasing in p_e 's, i.e., if $p_e \leq q_e \quad \forall e \in \mathcal{E}$

$$P^\infty(\mathcal{G}, \{p_e, e \in \mathcal{E}\}) \succeq P^\infty(\mathcal{G}, \{q_e, e \in \mathcal{E}\}),$$

where $A \succeq B$ means that $A - B$ is positive semi-definite.

Using the above lemma we can lower bound the steady-state error by making a subset of links erasure free. In particular, consider any source-destination cut in the network (which is simply a partition of the nodes in two sets one containing the source node (the *source set*) and the other containing the destination node (the *destination set*)). Setting the probability of erasure equal to zero for every link except those crossing the cut gives a lower bound on the error. Therefore,

$$P^\infty(\mathcal{G}, \{p_e, e \in \mathcal{E}\}) \succeq P^\infty(\mathcal{G}, \{q_e, e \in \mathcal{E}\})$$

where $q_e = p_e$ iff e is in the cut and zero otherwise. Now the left side of the above equation can be calculated easily using the results from Section IV-.3. In particular, it can be shown that for stability we require that

$$\max_{C:s-d \text{ cut}} \left(\prod_{e \in C} p_e \right) |\lambda_{\max}(A)|^2 < 1$$

We refer to $p_{\text{mc}}(\mathcal{G}) = \max_{C:s-d \text{ cut}} \left(\prod_{e \in C} p_e \right)$ as the max-cut value of the network.

One way to upper bound the steady-state error is by setting the probability of packet drop of some of the edges equal to one. In [20], it is shown that the performance of the network \mathcal{G} is lower bounded by the performance of another network \mathcal{G}' with the following properties:

- \mathcal{G}' has the same node set.
- \mathcal{G}' is the combination of edge-disjoint paths from the source to destination.
- Along each path the links have the same probability of dropping packets equal to the probability of packet drop of one of links in the max-cut of the original network \mathcal{G} .
- Based on the previous property the value of the max-cut in \mathcal{G}' is the same as the original network \mathcal{G} .

Now \mathcal{G}' is a network with series and parallel links only. Thus its performance can be computed and provides an upper bound on the steady-state error covariance of \mathcal{G} . In particular, since all the paths from s to d are disjoint,

$$\Pr(l_d(\mathcal{G}') \geq l + 1) = \prod_i \Pr(l_d(\mathcal{P}_i) \geq l + 1)$$

where $l_d(\mathcal{P}_i)$ is the steady-state latency of path \mathcal{P}_i . But for any path with n links,

$$\Pr(l_d \geq l + 1) = \sum_{i=0}^{n-1} \binom{i+l-n}{l-n} (1-p)^i p^{l-n+1}$$

Using the Stirling formula for large l , we obtain

$$c_1 \leq \frac{\Pr(l_d \geq l + 1)}{\left(\frac{1}{p} - 1\right)^{n-1} (l-1)^{n-1} p^l} \leq c_2 \quad (11)$$

where c_1, c_2 are two positive constants independent of l . Therefore, for large l , $\Pr(l_d(\mathcal{G}') \geq l + 1)$ behaves like

$$f(l) \left(\prod_i p_i \right)^l = f(l) (p_{\text{mc}}(\mathcal{G}'))^l = f(l) (p_{\text{mc}}(\mathcal{G}))^l$$

where $f(l)$ grows polynomially in l . Thus it is easy to verify that for network \mathcal{G}' the system is stable if $p_{mc}(\mathcal{G})$ satisfies

$$p_{mc}(\mathcal{G})|\lambda_{\max}(A)|^2 < 1.$$

Therefore the above condition is both necessary and sufficient for stability.

5) *Synthesis of a Network*: One can use the results on the performance of networks to design networks that result in minimal error covariance. To consider a simple example, consider a scalar system observed by sensor s . Assume that the destination is located at distance d_0 from the sensor. The probability of dropping a packet on a link depends on its physical length. A reasonable model for probability of dropping packets is given by² $p(d) = 1 - \exp(-\beta d^\alpha)$, where β, α are positive constants. α denotes the exponent of power decay in the wireless environment. We are interested in the optimal number n of relay nodes between sensor and the destination so as to minimize the expected steady-state error covariance. Assuming that the sensor are uniformly placed, P^∞ satisfies

$$P^\infty = \left(\frac{a^2(1-p)}{1-pa^2} \right)^{n+1} \left(P^* + \frac{Q}{a^2-1} \right) - \frac{Q}{a^2-1}$$

Thus the optimal n (assuming that $a^2 > 1$) is the solution to the problem

$$\min_n \left(\frac{a^2(1-p(\frac{d_0}{n+1}))}{1-p(\frac{d_0}{n+1})a^2} \right)^{n+1}$$

If $a^2 < 1$ then minimization is replaced with maximization.

V. EXAMPLES

In this section, we illustrate the above results using a simple example. Consider a scalar process evolving as

$$x_{k+1} = 0.8x_k + w_k,$$

that is being observed through a sensor of the form

$$y_k = x_k + v_k.$$

The noises w_k and v_k are assumed zero-mean, white and Gaussian with covariances $Q = 1$ and $R = 1$ respectively. Further, the two noises are assumed independent of each other. To begin with, suppose that the source and the destination node are connected using two links in series, each with a probability of packet erasure p . Figure 2 shows the performance of our strategy as the probability p is varied. The simulation results refer to data generated by a random run averaged over 100000 time steps while the theoretical values refer to the value predicted by using (9). We can see that the two sets of values match quite closely.

We also carried out a similar exercise for the source and destination nodes connected by two links in parallel, with packet erasure probability p each. The results are plotted in

²This expression can be derived by considering the probability of outage in a Rayleigh fading environment.

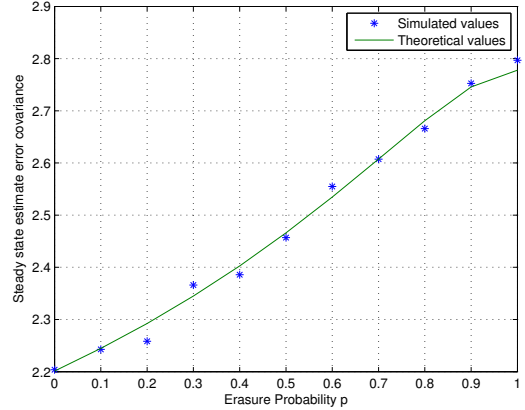


Fig. 2. Simulated and theoretical results for a line network.

Figure 3. We can once again see that the simulated values match quite closely with the theoretical values.

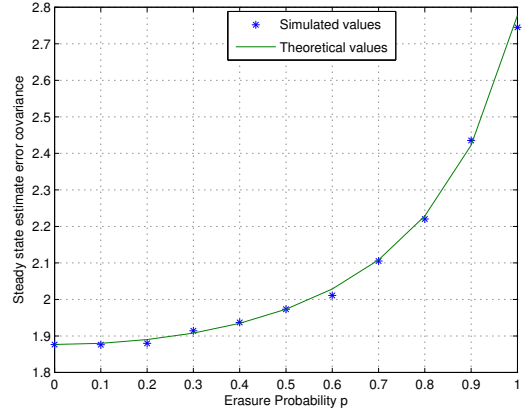


Fig. 3. Simulated and theoretical results for a parallel network.

As a final example, we consider the source and destination nodes connected by a bridge network shown in Figure 4. We assume all the links in the network to have probability of erasure p . This network cannot be reduced to a series of series and parallel sub-networks. We can however, calculate the performance analytically in this particular case and compare it to the upper and lower bounds presented earlier. The networks used for calculating the bounds are also shown in figure 4. Figure 5 shows a comparison of the analytical and simulated values with the lower and upper bounds. The simulated values do not fall below the upper bound every-time because of numerical issues; otherwise the bounds are tight. We also calculated the optimal number of nodes to be placed between the source and the destination node using our synthesis results. The values we used are $d_0 = 5$, $\alpha = 2$, $\beta = 1$. In this case, the optimal number of relays turns out to be $n = 4$.

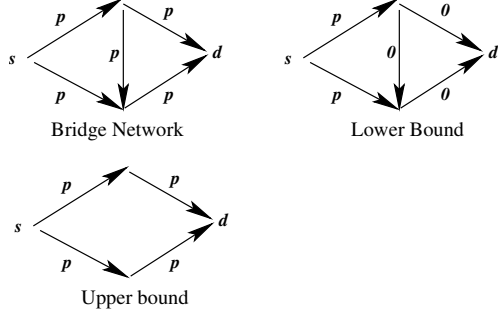


Fig. 4. Bridge network and the networks used for calculating lower and upper bounds.

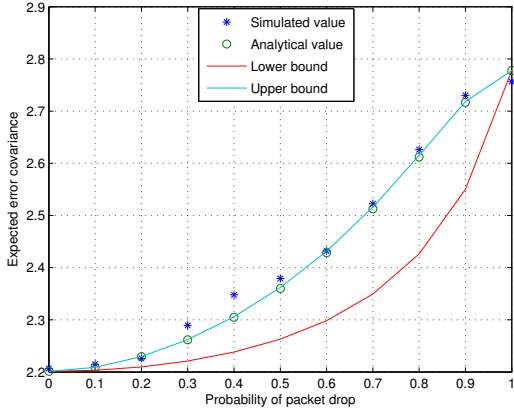


Fig. 5. Simulated values and theoretical bounds for the bridge network.

VI. GENERALIZATIONS

Correlated erasure events: The analysis so far assumed that the erasure events are memoryless and independent across different links in the network. We could thus formulate the performance in terms of a generating function of the steady-state latency distribution as defined in (6). We now look at the effect of dropping these assumptions.

Markov events: If we assume that the drop events on each link are governed by a Markov chain (but are still independent of other links), we can obtain the performance as follows. Let us assume that the packet drop event on link (u, v) , denoted by $D_{uv}(k)$ evolves according to a Markov chain with transition matrix M_{uv} . We further assume that M_{uv} is irreducible and reversible. Let us first consider the case where the initial distribution of packet drop on each link is the stationary distribution of the Markov chain on that link. Then we can rewrite (3) as (5) as before, where Z_l is a geometric random variable with distribution

$$\Pr(Z_{uv} = l) = \begin{cases} \alpha_{uv} M_{uv}(1, 2) M_{uv}(1, 1)^{l-2} & \forall l \geq 2 \\ 1 - \alpha_{uv} & l = 1 \end{cases},$$

with α_{uv} as the probability of packet drop based on the stationary distribution of link $e = (u, v)$ and $M_{uv}(i, j)$ as the (i, j) -th element of M_{uv} . Therefore, all the previous

analysis goes through. In particular, the stability condition is

$$\left(\max_{c:s-d \text{ cut}} \prod_{e \in c} M_e(1, 1) \right) |\lambda_{\max}(A)|^2 < 1.$$

Now, if the initial distribution is not the stationary distribution, the variables $Z_{uv}(k)$ will not be time-independent and the analysis does not go through. However, since for large k the chains approach their stationary distribution, the stability condition remains unchanged.

Spatially correlated events: Suppose that the packet drop events are correlated across the network but memoryless over time. In other words, at each time step k , the packet drop events occur according to distribution $\Pr_0(D_{uv}, (u, v) \in \mathcal{E})$. Now $Z_e(k)$'s are not independent across the network and hence finding the steady-state error covariance does not seem to be tractable. However, we can find the condition for stability. For this, we define a generalized notion of equivalent probability of packet drop for correlated events. Consider a $s - d$ cut c , and let $\mathcal{B}(c)$ denote the set of edges crossing this cut. Then the equivalent probability of packet drop for this cut is defined as

$$p_{eq}(c) = \Pr(D_{uv} = 0, \forall (u, v) \in \mathcal{B}(c)).$$

The value of the max-cut for the network is the maximum of $p_{eq}(c)$ over all the cuts, $p_{mc}(\mathcal{G}) = \max_{c:s-d \text{ cut}} p_{eq}(c)$. We can show that the condition for stability of the system is

$$p_{mc}(\mathcal{G}) |\lambda_{\max}(A)|^2 < 1.$$

To see this, consider the scenario when only one packet is to be routed from the source to destination starting at time t_0 . For each time-step $t \geq t_0$ let $\mathcal{V}_r(t)$ denote the set of nodes that have received the packet at time t . Clearly $\mathcal{V}_r(t_0) = \{s\}$. We want to bound the probability that at time $t_0 + T$, destination node has not yet received the packet. Note that for every time-step between t_0 and $t_0 + T$, $\mathcal{V}_r(t)$ clearly forms a cut-set since it contains s and not d . Now the size of $\mathcal{V}_r(t+1)$ does not increase with respect to time-step t iff all the links that cross the cut generated by $\mathcal{V}_r(t)$ drop packets. However by the definition of $p_{mc}(\mathcal{G})$ the probability of this event is at most $p_{mc}(\mathcal{G})$. Therefore, we have

$$|\mathcal{V}_r(t+1)| \begin{cases} \geq |\mathcal{V}_r(t)| + 1 & \text{with prob. at most } p_{mc}(\mathcal{G}) \\ = |\mathcal{V}_r(t)| & \text{with prob. at least } 1 - p_{mc}(\mathcal{G}) \end{cases}$$

Thus for large T , the probability that at time $t_0 + T$ the destination node has not received the packet is upper bounded by $n(1 - p_{mc}(\mathcal{G}))^n T^n p_{mc}(\mathcal{G})^{T-n}$, where n is the number of nodes in the network. In the original scenario, a new packet is generated at the source at each time step. However, since the importance of the packets is increasing with time, we can upper bound the error by considering that the network is only routing packet generated at time $k - l$. The probability that the latency is larger than l grows like $f(l) p_{mc}(\mathcal{G})^l$, where $f(l)$ is polynomial in l with bounded degree and thus the sufficiency of the stability condition

follows. The necessity part involves similar ideas and is omitted.

Unicast Networks: So far, we assumed that the topology of the network was given and any node could transmit a message on all the out-going links. If the network is unicast, each node chooses one link out of a set to transmit its message. The problem is to choose the optimal path for the data to flow from the source to the destination node. Clearly once the path is chosen the optimal operation at each node on the path is given by algorithm \mathcal{A} described in Section III.

In order to choose the optimal path, we need to define a metric for the cost of a path. If the metric is the condition for stability of the estimate error covariance, then the problem can be recast as choosing the shortest path in a graph with the length of a path being given by its equivalent probability of packet drop, i.e, for each path P , $p_{eq} = \max_{e \in P} p_e$. Thus the shortest path problem is to find the path that has the minimum equivalent probability of packet drop among all the paths, i.e, $\min_{P:s-d \text{ path}} \max_{e \in P} p_e$. The above problem is well studied in the computer science society and can be solved as a short-path problem over min-max semi-ring in a distributed fashion [24]. If the metric is the steady-state error then the problem is more complicated in general. For the special case of a scalar system and no process noise, from (10), we have for path q ,

$$\log P_q^\infty = \sum_{e \in q} \log \left(\frac{(1 - p_e)a^2}{1 - p_e a^2} \right)$$

Now the problem is equivalent to

$$\min_{q:s-d \text{ path}} \sum_{e \in q} \log \left(\frac{(1 - p_e)a^2}{1 - p_e a^2} \right)$$

This problem can also be solved in a distributed way [25].

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we considered the problem of optimal estimation across a network. We modeled the links as packet erasure links. We provided a framework for computing the optimal estimate error covariance and gave upper and lower bounds on the performance of general networks. We showed how to utilize this framework for the synthesis of networks for the purpose of estimation. We also carried out the stability analysis for arbitrary networks and for packet erasure processes that are possibly correlated across time or the network.

In this paper, we have ignored issues of quantization. One interesting and challenging problem is to include constraints of a limited bit rate into the framework. The work of Sahai [22] and Ishwar et al [23] may be relevant to this problem. In the future, we would like to explore these connections.

REFERENCES

- [1] "An Introduction to Multisensor Data Fusion," D. L. Hall and J. Llinas, *Proceedings of the IEEE*, 85(1), 1997, pp. 6-23.
- [2] "Distributed Detection with Multiple Sensors: Part I - Fundamentals", R. Viswanathan and P. K. Varshney, *Proceedings of the IEEE*, 85(1), 1997, pp. 54-63.
- [3] "Special Issue on Networks and Control," L. Bushnell (Guest Editor), *IEEE Control Systems Magazine*, 21(1), Feb 2001.
- [4] "Special Issue on Networked Control Systems," P. Antsaklis and J. Baillieul (Guest Editors), *IEEE Transactions on Automatic Control*, 49(9), Sept 2004.
- [5] "Stabilizing a Linear System with Quantized State Feedback," D. F. Delchamps, *IEEE Transactions on Automatic Control*, 35, 1990, pp. 916-924.
- [6] "Control under Communication Constraints," S. Tatikonda, PhD Thesis, MIT, Cambridge, MA 2000.
- [7] "Stabilizability of Stochastic Linear Systems with Finite Feedback Data Rates", G. N. Nair and R. J. Evans, *SIAM Journal on Control and Optimization*, 43(2), July 2004, pp. 413-436.
- [8] "Towards the Control of Linear Systems with Minimum Bit-rate", J. Hespanha, A. Ortega and L. Vasudevan, *Proceedings of the 15th International Symposium on the Mathematical Theory of Networks*, 2002.
- [9] "Real-time Control Systems with Delays," J. Nilsson, PhD Thesis, Department of Automatic Control, Lund Institute of Technology, 1998.
- [10] "Coordinated Control of Unmanned Aerial Vehicle", P. Seiler, PhD Thesis, University of California, Berkeley, 2001.
- [11] "Stability of Networked Control Systems", W. Zhang, M. S. Branicky and S. M. Philips, *IEEE Control System Magazine*, 21(1), Feb 2001, pp. 84-89.
- [12] "Robust Performance of Soft Real-time Networked Control Systems with Data Dropouts", Q. Ling and M. D. Lemmon, *Proceedings of the IEEE Conference on Decision and Control*, 2002.
- [13] "Feedback Control Utilizing Packet Dropping Network Links", C. N. Hadjicostis and R. Touri, *Proceedings of the IEEE Conference on Decision and Control*, 2002.
- [14] "Optimal Dropout Compensation in Networked Control Systems", Q. Ling and M. D. Lemmon, *Proceedings of the IEEE Conference on Decision and Control*.
- [15] "Stability of Networked Control Systems in the Presence of Packet Losses", *Proceedings of IEEE Conference on Decision and Control*, 2003.
- [16] "Time Varying Optimal Control with Packet Losses", B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla and S. S. Sastry, *Proceedings of the IEEE Conference on Decision and Control*, 2004.
- [17] "Optimal Control of Dynamical Systems over Unreliable Communication Links", O. C. Imer, S. Yuskel and T. Basar, *NOLCOS*, 2004.
- [18] "Kalman Filtering with Intermittent Observations", B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan and S. S. Sastry, *IEEE Transactions on Automatic Control*, 49(9), Sept. 2004, pp. 1453-1464.
- [19] "On LQG Control Across Packet-Dropping Links," V. Gupta, D. Spanos, B. Hassibi and R. M. Murray, *System and Control Letters*, submitted July 2005.
- [20] "Data Transmission over Networks for Estimation", V. Gupta, A. F. Dana, J. P. Hespanha and R. M. Murray, *17th International Symposium on Mathematical Theory of Networks and Systems*, MTNS 2006, Submitted.
- [21] "Linear Estimation", T. Kailath, A. H. Sayed and B. Hassibi, Prentice Hall, New Jersey, 2000.
- [22] "Anytime Information Theory", A. Sahai, PhD Thesis, MIT, Cambridge, MA 2001.
- [23] "On Rate-constrained Distributed Estimation in Unreliable Sensor Networks", P. Ishwar, R. Puri, K. Ramchandran and S. S. Pradhan *IEEE Journal on Selected Areas in Communications: Special Issue on Self-organizing Distributed Collaborative Sensor Networks*, 23(4), Apr. 2005, pp. 765-775.
- [24] "Semiring frameworks and algorithms for shortest-distance problems", M. Mohri *Journal of Automata, Languages and Combinatorics*, 2002, pp. 321-350
- [25] "Introduction to Algorithms", T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, MIT Press and McGraw-Hill, 1990.