Data Transmission over Networks for Estimation and Control

Vijay Gupta *Member, IEEE*, Amir F. Dana *Member, IEEE*, Joao P. Hespanha *Fellow, IEEE*, Richard M. Murray *Fellow, IEEE* and Babak Hassibi *Member, IEEE*

Abstract— **We consider the problem of controlling a linear time invariant process when the controller is located at a location remote from where the sensor measurements are being generated. The communication from the sensor to the controller is supported by a communication network with arbitrary topology composed of analog erasure channels. Using a separation principle, we prove that the optimal LQG controller consists of an LQ optimal regulator along with an estimator that estimates the state of the process across the communication network. We then determine the optimal information processing strategy that should be followed by each node in the network so that the estimator is able to compute the best possible estimate in the minimum mean squared error sense. The algorithm is optimal for any packet-dropping process and at every time step, even though it is recursive and hence requires a constant amount of memory, processing and transmission at every node in the network per time step. For the case when the packet drop processes are memoryless and independent across links, we analyze the stability properties and the performance of the closed loop system. The algorithm is an attempt to escape the viewpoint of treating a network of communication links as a single end-to-end link with the probability of successful transmission determined by some measure of the reliability of the network.**

I. INTRODUCTION

Recently a lot of attention has been directed towards networked control systems in which components communicate over wireless links or communication networks that may also be used for transmitting other unrelated data (see, e.g., [1], [2] and the references therein). The estimation and control performance in such systems is severely affected by the properties of the communication channels. Communication links introduce many potentially detrimental phenomena, such as random delays, data loss and data corruption to name a few, that lead to performance degradation or even loss of stability.

In this work, we are specifically interested in the problem of estimation and control across a *network* of communication links that drop data 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 remote node, which can either be an estimator or a controller. However, the

V. Gupta is with the Department of Electrical Engineering, University of Notre Dame. *vgupta2@nd.edu*

A. F. Dana is with Qualcomm Inc., San Diego. *dana@systems.caltech.edu*

J. P. Hespanha is with the Department of Electrical and Computer Engineering Department, University of California, Santa Barbara.*hespanha@ece.ucsb.edu*, work partially supported under the ARO grant DAAD19-03-D-0004 and the NSF Grant CCR-0311084

R. M. Murray is with the Department of Control and Dynamical Systems, California Institute of Technology.*murray@caltech.edu*

B. Hassibi is with the Department of Electrical Engineering, California Institute of Technology.*hassibi@systems.caltech.edu*

links in the network stochastically drop packets. Preliminary work in this area has concentrated on networks consisting of a single link between the sensor and the remote estimator / controller. Within the one-link framework, both the stability [19], [23] and the performance [15], [19] have been analyzed. Approaches to compensate for the data loss to counteract the degradation in performance have also been proposed [16], [9], [15], [21]. Also relevant are the works of Azimi-Sadjadi [3], Schenato et al. [18] and Imer et al. [12] who looked at controller structures to minimize quadratic costs for systems in which both sensor-controller and controller-actuator channels are present. The related problem of estimation across a packet-dropping link was considered by Sinopoli et al. in [20] for the case of one sensor and packet drops occurring in an i.i.d. fashion, while Gupta et al. [7] considered multiple sensors and more general packet drop models.

It has often been recognized that typical network / communication data packets have much more space for carrying information than required inside a traditional control loop. For instance, the minimum size of an ethernet data packet is 72 bytes, while a typical data point may only consume 2 bytes. Many other examples are given in Lian et al. [14]. Moreover, many of the devices used in networked control systems possess processing and memory capabilities on account of being equipped to communicate across wireless channels or networks. Thus, the question arises if we can exploit these capabilities to pre-process information prior to transmission and transmit extra data in every packet to combat the performance degradation due to communication channels. In Gupta et al. [8] it was shown that pre-processing (or encoding) information before transmission over the communication link can indeed yield significant improvements in terms of stability and performance. In this paper, we consider the design of encoders and decoders when the sensor data has to be transmitted over a network of arbitrary topology.

Transmission of data over networks for the purpose of estimation and control is largely an open problem. Tatikonda [22] studied some issues related to the quantization rates required for stability when data is being transmitted over a network of digital noiseless channels. Also relevant is the work of Robinson and Kumar [17] who considered the problem of optimal placement of the controller when the sensor and the actuator are connected via a series of communication links. They ignore the issue of delays over paths of different lengths (consisting of different number of links) and under a *Long Packet Assumption* come up with the optimal controller structure. There are two main reasons why the problem of

encoding data for transmission is much more complicated in the case of transmission over a network:

1) Intermediate nodes have memory and processing ability. This memory should be used and one should not view the network as a "passive" memoryless erasure channel.

2) Typically there are many paths from the sensor to the remote estimator/controller. These paths typically exhibit different delays and levels of reliability. This diversity should be exploited by the system designer.

The main contributions of the paper are as follows:

1) We present a separation principle that decomposes the optimal control problem into an LQ optimal regulator design and estimation of a process across a network. Moreover, we show that for the estimation problem, the intermediate nodes of the network do not require access to the control inputs.

2) We propose a recursive algorithm for information processing at the nodes of the network so that the estimator can calculate the optimal estimate at every time step. The estimate is optimal given the maximal possible information set that the estimator can have access to given any causal information processing algorithm. Our algorithm is optimal for *any realization* of the packet-dropping process yet requires a constant amount of memory, processing and transmission at any node per time step.

3) We analyze the stability of the expected error covariance for this strategy when the packet drops are independent across time steps and across channels. For any other scheme (e.g., transmitting measurements without any processing), these conditions are necessary for stability. The analysis identifies a property of the network called the max-cut probability that completely characterizes the network for its ability to be used for stabilizing a control loop. For channels with correlated drops, we show how to extend this analysis.

4) We calculate the performance for our algorithm for channels that drop packets independently. 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. For any other strategy, these provide lower bounds for achievable performance.

As shown in [5], the results can also be used for synthesis of networks from a control oriented viewpoint.

II. PROBLEM SETUP

Consider the problem setup shown in Figure 1. Let a discrete-time linear process evolve as

$$z(k+1) = Az(k) + Bu(k) + w(k), \qquad (1)$$

where $z(k) \in \mathbf{R}^n$ is the process state, $u(k) \in \mathbf{R}^m$ is the control input and $w(k)$ is the process noise assumed to be white, Gaussian, and zero mean with covariance matrix $R_w > 0$. The initial condition $z(0)$ is assumed to be independent of $w(k)$ and to have mean zero and covariance matrix $R(0)$.
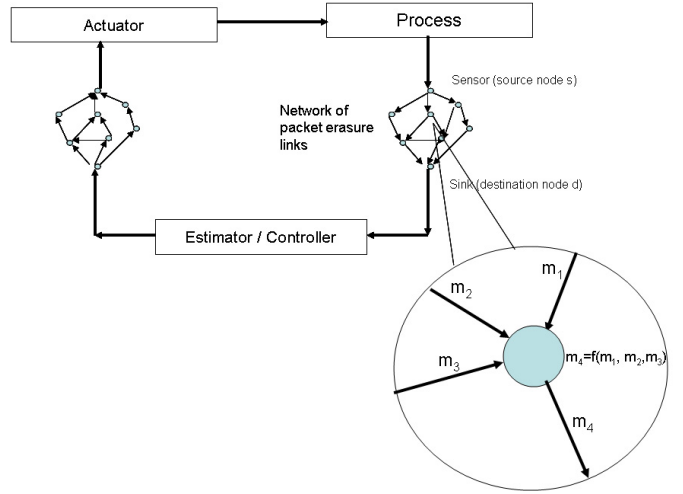

Fig. 1. The set-up of the control across communication networks problem. Every node computes a function of the incoming messages and transmits it. For most of the discussion in the paper, we ignore the network between the controller and the actuator. See, however, Remark 4.

The process state is measured by a sensor that generates measurements according to the equation

$$l(k) = Cz(k) + v(k). \qquad (2)$$

The measurement noise $v(k)$ is white, zero-mean, Gaussian (with covariance matrix $R_v > 0$) and independent of the plant noise $w(k)$. We assume that the pair (A, B) is stabilizable and the pair (A, C) is observable. Denote the above setup as system $\mathbb{S}_1$. Even though we consider the time-invariant case to simplify the presentation, most of the results in the paper continue to hold for time-varying systems. A time-varying model can be useful, e.g., if the discrete-time process (1) results from non-uniform sampling of a continuous-time process.

The sensor communicates with a controller across a network of communication links that stochastically drop packets. We make no assumptions about the topology of the network. The sensor constitutes the source node and is denoted by s. The controller is designated as the destination node d. The communication network defines a directed graph $\mathcal{G}$ with node set $\mathcal{V}$ (in particular, $\mathcal{V}$ contains s and d) and edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. The edges of the graph represent the communication links and are, in general, directed. Specifically, the link $e = (u, v)$ models a communication channel from node u to node v. We assume there are M edges or links present in the network. For any node $i \in \mathcal{V}$, the set of outgoing edges corresponds to the channels along which the node can transmit messages while the set of incoming edges corresponds to the channels 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 an analog erasure model. Each link takes in as input a vector of real numbers with a finite dimension. At each time-step, this message is either dropped or received without any error at the output node. We assume sufficient bits per data packet so that the quantization error is negligible. This assumption makes sense if the communication packet provides enough bits for transmitting data (as in most modern communication network

protocols) so that the effect of quantization error is dominated by the effect of the process and the measurement noises. We nominally consider the delays introduced by the channel to be less than one time step according to which the discrete-time dynamic process evolves. Most of the results in the paper can however directly be extended to the case when larger delays are present. In particular, the algorithm in the case of arbitrary delays and packet rearrangements is provided in Remark 3. In this paper, we also assume a global clock so that each node is synchronized. Finally, we assume that each node can receive all the messages coming along the incoming links without interference from each other[1].

If the packet dropping process is independent from one time step to the next (or, in other words, memoryless), the probability of dropping a packet on link $e = (u, v) \in \mathcal{E}$ is given by p_e (or, equivalently, p_{uv}) independent of time. If the process is uncorrelated in space, each such packet drop event is independent of packet drops in other links. While our stability and performance analysis is largely limited to packet drop processes that are independent in time and uncorrelated in space, the algorithm that we propose is optimal for more sophisticated models such as drops occurring according to a Markov chain. We refer to individual realizations of the random processes that describes the drops for the networks as a *packet drop sequence*. The operation of the different nodes in the network at every time-step k can be described as follows:

1) Every node computes a function of the information it has access to at time k.
2) It transmits the function on the out-going edges. Potentially the node can transmit different functions along different edges. The destination node calculates the control input $u(k)$ based on the information it possesses.
3) Every node observes the messages from the incoming links and updates its information set for the next time step. For the source node, this message is $l(k)$.

This time line ensures a strictly causal operation. At time step k, the function that the source node transmits depends on measurements $l(0), \cdots, l(k-1)$. Similarly, even if there were no packet drops, if the destination node is d hops away from the source node (i.e., the shortest path from the source node to the destination node involves d edges), its control input $u(k)$ can only depend on measurements $l(0), \cdots, l(k-d-1)$. Thus, every communication edge consumes one hop, or one time step, as data is transmitted across it. We can easily adapt the discussion presented below to the causal case.

At every time step, the controller calculates a control input and transmits it to the actuator. For the present, we ignore any communication channel between the controller and the actuator. The controller aims at minimizing the cost function

$$J_T = E\Big[\sum_{k=0}^{T} \big(z^T(k)Qz(k) + u^T(k)Ru(k)\big) + z^T(T+1)P^c_{T+1}z(T+1)\Big], \quad (3)$$

[1]This property can be easily achieved by using a division multiple access scheme like FDMA, TDMA, CDMA etc. Technologies like Software Radio (SWR) also have this property.

where the expectation is taken over the uncorrelated variables $x(0)$, $\{w(k)\}$ and $\{v(k)\}$ and the horizon T is finite. Note that the cost functional J_T also depends on the random packet-drop sequences in each link. However, we do *not* average across packet-drop processes; *the solution we present is optimal for arbitrary realizations of the packet dropping processes*. We assume that the controller has access to all the previous control inputs $u(0), \cdots, u(k-1)$ while calculating $u(k)$. Without the communication network, the control problem is the same as the classical LQG control synthesis problem. However, in the presence of the network, it is unclear *a priori*, what the structure of the optimal control algorithm should be, and in what way the links in the network should be used to transmit information. Clearly, transmitting measurements might not be optimal, since in such a scheme, dropping a packet would mean information loss that cannot be compensated for in the future. We are particularly interested in strategies that do not entail an increasing amount of transmission, memory and processing at the nodes. We are not interested in interleaving bits to transfer infinite amount of data since it is unclear what the effect of having finite (even though large) number of bits would be for such a strategy. We will also identify the conditions on the network for stability. We are interested in stability in the bounded second moment or the mean squared sense. Thus, the system is stable if $E[J_\infty]$ is bounded, where $J_\infty = \lim_{T\to\infty} \frac{J_T}{T}$ and the expectation is taken over the packet dropping processes in the network. We denote this problem set-up as problem $\mathcal{P}_1$.

III. A SEPARATION PRINCIPLE

For the node i, denote by $\mathcal{I}^i(k)$ the information set that it can use to generate the message that it transmits at time step k. This set contains the aggregate of the information the node has received on the incoming edges at time steps 0, 1, $\cdots$, $k-1$. As an example, for the source node s, $\mathcal{I}^s(k) = \{l(0), l(1), \cdots, l(k-1)\}$. Without loss of generality, we can restrict our attention to information-set feedback controllers, i.e., controllers of the form $u(k) = u(\mathcal{I}^d(k), k)$. For a given information set at the destination $\mathcal{I}^d(.)$, denote the minimal value of J_T that can possibly be achieved by $J_T^\star(\mathcal{I}^d)$. Let $\lambda_{pq}(k)$ be the binary random variable describing the packet drop event on link $(p, q) \in \mathcal{E}$ at time k. $\lambda_{pq}(k)$ assumes the value 0 if the packet is erased on link (p, q) at time k and 1 otherwise. For a network with independent and memoryless packet drops, $\lambda_{pq}(k)$ is distributed according to Bernoulli distribution with parameter p_{pq}. We define $\lambda_{pp}(k) = 1$. Given the packet drop sequences in each link, at time step k we can define a time stamp $t^i(k)$ for node i such that the packet drops did not allow any information transmitted by the source after $t^i(k)$ to reach the i-th node in time for it to be a part of $\mathcal{I}^i(k)$.

Now consider an algorithm $\mathcal{A}_1$ in which every node takes the following actions at every time step:

1) Transmit its entire information set on the outgoing edges.
2) Receive data successfully transmitted along the incoming edges.

3) Update its information set and affix a time stamp corresponding to the time of the latest measurement in it.

For any drop sequence, the information set at node i with this algorithm is of the form $\mathcal{I}^i(k) = \{l(0), l(1), \cdots, l\left(t^i(k)\right)\}$, where $t^i(k) < k$ is the time stamp as defined above. This is the maximal information set $\mathcal{I}^{i,\max}(k)$ that the node i can possibly have access to with any algorithm. For any other algorithm, the information set is smaller than $\mathcal{I}^{i,\max}(k)$ in the sense that the smallest sigma algebra generated by the information set is a subset of the smallest sigma algebra generated by $\mathcal{I}^{i,\max}(k)$.

Consider two arbitrary information sets $\mathcal{I}^d(k, 1)$ and $\mathcal{I}^d(k, 2)$. Let the smallest sigma algebras generated by the two information sets be denoted by $\mathbb{I}^d(k, 1)$ and $\mathbb{I}^d(k, 2)$ respectively. If the two information sets are such that $\mathbb{I}^d(k, 1) \subseteq \mathbb{I}^d(k, 2)$, we have $J_T^\star(\mathcal{I}^d(k, 1)) \leq J_T^\star(\mathcal{I}^d(k, 2))$. Thus, one way to achieve the optimal value of J_T is through the combination of an information processing algorithm that makes the information set $\mathcal{I}^{d,\max}(k)$ available to the controller and a controller that optimally utilizes the information set. Further, one such information processing algorithm is the algorithm $\mathcal{A}_1$ described above. However, this algorithm requires increasing amount of data transmission as time evolves. Surprisingly, we can achieve the same performance using a constant amount of transmission and memory. To see this, we first state the following separation principle. For any random variable $\alpha(k)$, denote by $\hat{\alpha}\left(k|\beta(k)\right)$ the minimum mean squared error (MMSE) estimate of $\alpha(k)$ given the information $\beta(k)$.

Proposition 1 (Separation Principle). *Consider the packet-based optimal control problem $\mathcal{P}_1$ defined in section II. Suppose that each node transmits all the measurements it has access to at every time step, so that the decoder has access to the maximal information set $\mathcal{I}^{d,\max}(k)$ at every time step k. Then, the optimal control input at time k is calculated to be*

$$u(k) = \hat{u}_{LQ}\left(k|\mathcal{I}^{d,\max}(k), \{u(t)\}_{t=0}^{k-1}\right)$$

where $u_{LQ}(k)$ is the optimal LQ control law and $\hat{u}_{LQ}\left(k|\mathcal{I}^{d,\max}(k), \{u(t)\}_{t=0}^{k-1}\right)$ denotes its minimum mean squared error (MMSE) estimate given the information set $\mathcal{I}^{d,\max}(k)$ and the previous control inputs $u(0), \cdots, u(k-1)$.

Proof: Proof is similar to the standard separation principle [10, Chapter 9] and is omitted for space constraints. □

There are two reasons this principle is useful to us:

1) We recognize that the optimal controller does not need the information set $\mathcal{I}^{d,\max}(k)$. The encoders and the decoder only need to ensure that the controller receives the quantity $\hat{u}_{LQ}\left(k|\mathcal{I}^{d,\max}(k), \{u(t)\}_{t=0}^{k-1}\right)$, or equivalently, $\hat{z}\left(k|\mathcal{I}^{d,\max}(k), \{u(t)\}_{t=0}^{k-1}\right)$.
2) If the controller has access to this quantity, the optimal controller is the solution to the LQ control problem.

We can make another simplification by separating the dependence of the estimate on measurements from the effect of the control inputs. In the context of our problem, this implies that the nodes in the network do not need access to the control inputs, and can concentrate solely on the effect of measurements. The effect of the control inputs can be included by the controller that has access to previous control inputs.

Proposition 2 (Separation of the effect of control inputs). *Consider the problem $\mathcal{P}_1$ defined in section II. The estimate $\hat{z}\left(k|\mathcal{I}^{d,\max}(k), \{u(t)\}_{t=0}^{k-1}\right)$, where $\mathcal{I}^{d,\max}(k)$ is of the form $\{l(0), \cdots, l(t^d(k))\}$, can be calculated as*

$$\hat{z}\left(k|\mathcal{I}^{d,\max}(k), \{u(t)\}_{t=0}^{k-1}\right) = \bar{z}\left(k|\mathcal{I}^{d,\max}(k)\right) + \psi(k),$$

where $\bar{z}\left(k|\mathcal{I}^{d,\max}(k)\right)$ depends only on $\mathcal{I}^{d,\max}(k)$ and $\psi(k)$ depends only on the control inputs. Further, both $\bar{z}\left(k|\mathcal{I}^{d,\max}(k)\right)$ and $\psi(k)$ can be calculated recursively.

Proof: For simplicity of notation, let $t^d(k) = j$. Define the quantity $\breve{z}(j + 1|j)$ calculated at any time using the measurements from time $m = 0$ to $k - 1$ according to the following *modified Kalman filter* equations.

Measurement update for the modified Kalman filter:

$$M^{-1}(k|k) = M^{-1}(k|k-1) + C^T R_v^{-1} C$$
$$M^{-1}(k|k)\breve{z}(k|k) = M^{-1}(k|k-1)\breve{z}(k|k-1) + C^T R_v^{-1} l(k),$$

Time update for the modified Kalman filter:

$$M(k|k-1) = AM(k-1|k-1)A^T + R_w$$
$$\breve{z}(k|k-1) = A\breve{z}(k-1|k-1).$$

The initial conditions are given by $\breve{z}(0|-1) = 0$ and $M(0|-1) = P(0)$. The effect of the control inputs can be taken care of through the term $\breve{\psi}(j + 1)$ that evolves as

$$\breve{\psi}(m) = Bu(m-1) + \Gamma(m-1)\breve{\psi}(m-1)$$
$$\Gamma(m) = A\left(M(m-1|m-1)\right)^{-1} M(m-1|m-2),$$

with $\breve{\psi}(0) = 0$. It can readily be verified that

$$\hat{z}\left(k|\{l(t)\}_{t=0}^j, \{u(t)\}_{t=0}^{k-1}\right) = A^{k-j-1}\breve{z}(j+1|j)$$
$$+ A^{k-j-1}\breve{\psi}(j+1) + \sum_{i=0}^{k-j-2} A^i Bu(k-i-1).$$

To complete the proof, we simply identify

$$\bar{z}\left(k|\mathcal{I}^{d,\max}(k)\right) = A^{k-j-1}\breve{z}(j+1|j) \tag{4}$$
$$\psi(k) = A^{k-j-1}\breve{\psi}(j+1) + \sum_{i=0}^{k-j-2} A^i Bu(k-i-1).$$

□

IV. A RECURSIVE OPTIMAL ENCODING ALGORITHM

We now describe an algorithm $\mathcal{A}_2$ that achieves the same performance as algorithm $\mathcal{A}_1$ with constant memory, processing and transmission (modulo transmission of a time stamp). At each time step k, every node i takes the following actions:

1) Calculate the quantity $\bar{z}^i(k|\{l(t)\}_{t=0}^{t^i(k)})$ using data received from other nodes at the previous time step $k-1$ and the quantity $\bar{z}^i(k-1|\{l(t)\}_{t=0}^{t^i(k-1)})$ that it calculated at time $k - 1$. This can be computed using a switched linear filter, as follows. The source node implements a modified Kalman filter and calculates the quantity $\bar{z}^s(k|\{l(t)\}_{t=0}^{k-1})$. It sets the time-stamp to $k - 1$. Every other node i checks the time-stamps on the data coming on the incoming edges. The time-stamp on the data

Encoder at Sensor

New measurement arrives → Update Modified Kalman filter with new measurement → Transmit $\bar{z}(k|\{l(t)\}_{t=0}^{k-1})$

Intermediate Nodes

Collect quantities along incoming edges and the one in memory → Find the one with latest measurement → Time update, store and transmit

Decoder at Controller

Collect quantities along incoming edges and the one in memory → Find the one with latest measurement → Calculate $\hat{z}(k|\{l(t)\}_{t=0}^{j}, \{u(t)\}_{t=0}^{k-1})$

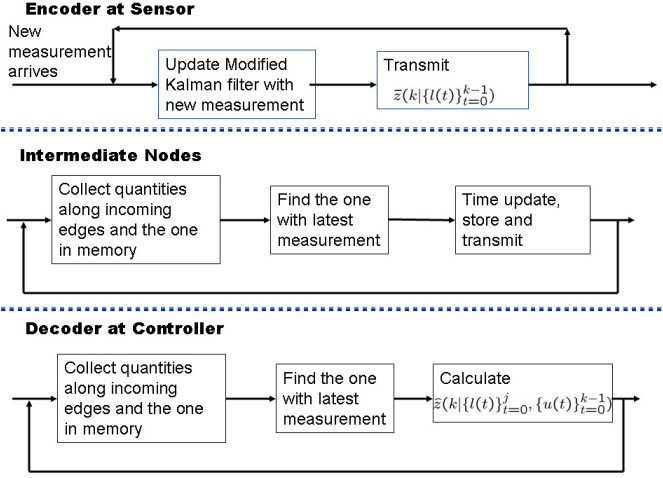Fig. 2.   A summary of the algorithm $\mathcal{A}_2$.

transmitted by a node j at time $k-1$ is simply the time-stamp $t^j(k-1)$ of the node j and corresponds to the latest measurement used in the calculation of the quantity being transmitted by node j. Then node i updates its time-stamp using the relation

$$t^i(k) = \max_{j \in \mathcal{N}_i \cup \{i\}} \lambda_{ji}(k-1) t^j(k-1). \qquad (5)$$

Suppose the maximum in (5) is achieved by node n. The node i calculates $\bar{z}^i(k|\{l(t)\}_{t=0}^{t^i(k)}) = A\bar{z}^n(k-1|\{l(t)\}_{t=0}^{t^n(k-1)})$ and time-stamp $t^i(k) = t^n(k-1)$.

2) Transmit the quantities $\bar{z}^i(k|\{l(t)\}_{t=0}^{t^i(k)})$ and the time stamp $t^i(k)$ on the outgoing edges.

3) Receive data on the incoming edges, if any, and store it for the next time step.

A 'flow-chart' version of the algorithm is provided in Figure 2. To prove that algorithm $\mathcal{A}_2$ is indeed optimal, we need the following intermediate result.

Lemma 1. *Consider any node i and any packet drop pattern. At time step k, the node i transmits the measurement set $\mathcal{I}^{i,\max}(k) = \{l(0), l(1), \cdots, l(t^i(k))\}$ on all outgoing edges if algorithm $\mathcal{A}_1$ is executed. If, instead, algorithm $\mathcal{A}_2$ is executed, the node i transmits the quantity $\bar{z}\left(k|\mathcal{I}^{i,\max}(k)\right) = \bar{z}(k|\{l(0), l(1), \cdots, l(t^i(k))\})$ along all outgoing edges.*

Proof: The statement about the algorithm $\mathcal{A}_1$ follows from the definition of the algorithm. The proof of the second statement can be proven by induction on the time step k. For time $k = 1$, by definition, the source node s transmits the quantity $\bar{z}(1|y(0))$ along all outgoing edges while executing algorithm $\mathcal{A}_2$. Since $t^s(1) = \{y(0)\}$, the statement is true. No other node transmits any information at this time. Thus the statement is true for $k = 1$. Now assume that the statement is true for $k = n$ for all nodes. Consider the node i at time $k = n + 1$. If the node i is the source node, the statement is true once again by definition of $t^s(k)$ and the first step of the algorithm. Let us assume that node i is not the source node. Consider all nodes that transmitted data at time $k = n$ to node i. For $p \in \mathcal{N}_i$ such that $\lambda_{pi}(n) = 1$, by the assumption of the

statement being true at time $k = n$, the node i receives from node p the quantity $\bar{z}(n|\{l(0), \cdots, l(t^p(n))\})$. Also, since at time $k = n$ the node transmitted $\bar{z}(n|\{l(0), \cdots, l(t^i(n))\})$ on all outgoing edges, it has access to this quantity. Let v be the node for which

$$t^v(n) = \max\{t^i(n) \cup \{t^p(n)|p \in \mathcal{N}_i \text{ and } \lambda_{pi}(n) = 1\}\}.$$

Clearly, the set $\{l(0), \cdots, l(t^v(n))\}$ is the superset of all sets $\{l(0), \cdots, l(t^p(n))\}$ for different nodes $p \in \mathcal{N}_i$ and the set $\{l(0), \cdots, l(t^i(n))\}$. Thus, from (4), we have

$$\bar{z}(n+1|\{l(0), \cdots, l(t^i(n+1))\}) = A\bar{z}(n|\{l(0), \cdots, l(t^v(n))\}),$$

with the time-stamp $t^i(n + 1) = t^v(n)$. But, by definition from the step 1 of algorithm $\mathcal{A}_2$, at time $k = n + 1$, the node i transmits along all outgoing edges the quantity $A\bar{z}(n|\{l(0), l(1), \cdots, l(t^v(n))\})$ and the time-stamp $t^v(n)$. Thus the statement is true at time step $k = n + 1$ for the node i. Since the node i was arbitrary, the statement is true for the entire graph. Thus we have proven that if the statement is true at time $k = n$, it is true at time $k = n + 1$. But it is true at time $k = 1$. Thus, by the principle of mathematical induction, it is true at all time steps. $\square$

Proposition 3. *The algorithm $\mathcal{A}_2$ is optimal in the sense that it allows the controller to calculate the control input $u(k)$ that minimizes the quadratic cost in equation (3).*

Proof: Consider the controller node d. At time k, let $j \in \{d\} \cup \mathcal{N}_d$ be a node for which $\lambda_{jd}(k-1) = 1$. Denote the measurement set transmitted from node j to node d at time step k under algorithm $\mathcal{A}_1$ by S^{jd}. Let v be the node for which S^{vd} is the superset of all the sets S^{jd}. Thus, the controller at time k under algorithm $\mathcal{A}_1$ calculates the quantity $\bar{z}(k|S^{vd})$. From Lemma 1, when algorithm $\mathcal{A}_2$ is executed, at time step k, the controller has access to the quantities $\bar{z}(k-1|S^{jd})$. Since S^{vd} is the superset of all the sets S^{jd}, the controller under algorithm $\mathcal{A}_2$ also calculates at time k $A\bar{z}(k-1|S^{vd}) = \bar{z}(k|S^{vd})$. Now, because of the Propositions 1 and 2, the control input calculated when the controller has access to $\bar{z}(k|S^{vd})$ leads to the minimum possible quadratic cost. Thus, the algorithm $\mathcal{A}_2$ is optimal. $\square$

Remark 1 (*Boundedness of the Transmitted Quantities*). It is important to emphasize that the quantity $\bar{z}\left(k|\mathcal{I}^{d,\max}(k)\right)$ is *not* the estimate of $z(k)$ based only on the measurements $l(0)$ through $l(t^d(k))$ that form the set $\mathcal{I}^{d,\max}(k)$. In particular, under the constraints of stabilizability, detectability and the conditions on erasure probabilities for stability that are derived in the next section, the system $z(k)$ is stable and hence the measurements $l(k)$ are bounded. Thus the quantity $\bar{z}^i(k|\mathcal{I}^{i,\max}(k))$ is stable. This can also be seen by rewriting the equations that govern its evaluation. From the equations of the Modified Kalman Filter, we have

$$\begin{aligned} \check{z}(k+1|k) &= A\check{z}(k|k-1) + K(k)(l(k) - C\check{z}(k|k-1)) \\ &= (A - K(k)C)\check{z}(k|k-1) + K(k)l(k), \end{aligned}$$

where $K(k)$ is the (conventional) Kalman filter gain. Due to the detectability and stabilizability assumptions, the matrix $(A - K(k)C)$ is stable. Thus, if $l(k)$ is a bounded term, the

quantity $\check{z}(k|k-1)$ and, in turn, $\bar{z}^i(k\mathcal{I}^{i,\max}(k))$ is stable. Thus, the quantities that are transmitted by the nodes in the optimal algorithm are stable. If the closed loop system $z(k)$ is not stable due to high erasure probabilities, the measurements $l(k)$ and hence the quantity $\bar{z}^i(k\mathcal{I}^{i,\max}(k))$ would, of course, not be stable. However, the optimality result means that the system cannot be stabilized by sending any other quantity (whether measurements $l(k)$ or innovations etc).

Remark 2 (*Optimality for any Drop Sequence and the 'Washing Away' Effect*)**.** We have made no assumptions on the packet drop pattern or knowledge of the statistics of the packet drops at any of the nodes. The algorithm is optimal for an arbitrary packet drop sequence, irrespective of whether the packet drop can be modeled as an i.i.d. process or a more sophisticated model like a Markov chain. The algorithm is optimal at every time step for any instantiation of the packet drop sequence, not merely in the optimal average performance. Also note that any received data vector $\bar{z}^d(k|\{l(j)\}_{j=0}^{t^d(k)})$ 'washes away' the effect of all previous packet drops. It ensures that the control input is identical to the case when all measurements $l(0)$, $\cdots$, $l(t^d(k))$ were available, irrespective of which previous data packets had been dropped.

Remark 3 (*Presence of Delays*)**.** If the links introduce delays, the algorithm remains optimal irrespective of the possibility of packet rearrangements. Each node i, at every time step k, still calculates the quantity $\bar{z}^i(k|\{l(j)\}_{j=0}^{t^i(k)})$ based on any information received at that time step and the previous estimate from its memory, affixes the correct time stamp and transmits it along outgoing edges. Further if the graph is finite, the stability conditions in the next section also do not change.

Remark 4 (*Channel between the controller and the actuator*)**.** The crucial assumption in the separation principle is that the controller knows what control input is applied. Thus, if we have a channel between the controller and the plant, the separation principle would still hold, provided there is a provision for acknowledgment from the receiver to the transmitter for packets successfully received over that channel[2]. The optimal information processing algorithm presented above also carries over to this case. We can also ask the question of the optimal encoder-decoder design for the controller-actuator channel. The optimal decoding at the actuator end depends on the information assumed to be known to the actuator (e.g. the cost matrices Q and R). Design of the decoder for various information sets is an interesting open problem.

It should be noted that a priori we had not made any assumption about a node transmitting the same message along all the out-going edges. It turned out that in this optimal algorithm, the messages are the same along all the edges. This property is useful in the context of wireless communication which is inherently broadcast in nature. Finally, the extent of time synchronization that is strictly necessary for the above algorithm is that the controller and the process be synchronized in the sense that a new control input is generated at every time step. Apart from that, the information that the nodes need to

---

know is the time stamp of the last measurement used to calculate the estimate that they have received. They do not require to be synchronized to the process. However, for simplicity of the analysis, we will assume time synchronization.

We now analyze the stability and performance of the above algorithm by assuming that packets are dropped independently from one time step to next and uncorrelated in space. We return to more general packet dropping processes in Section VII.

V. Stability Analysis

We begin by posing an alternate problem. Consider an open loop system $\mathbb{S}_2$ that evolves as

$$
\begin{align}
x(k+1) &= Ax(k) + w(k) \tag{6}\\
y(k) &= Cx(k) + v(k), \tag{7}
\end{align}
$$

where the noises have the same value at every time step as those appearing in the description of the system $\mathbb{S}_1$ and $x(0) = z(0)$. An encoder has access to the measurements $y(k)$. The node d now denotes an estimator across a network of analog erasure links that needs to calculate the MMSE estimate of $x(k)$ based on information it receives. The network has the same topology as the network for the problem $\mathcal{P}_1$. The cost function is to minimize the mean squared error based cost function

$$
D_k = E\left[\left(x(k) - \hat{x}(k)\right)\left(x(k) - \hat{x}(k)\right)^T\right], \tag{8}
$$

where the expectation is taken over the uncorrelated variables $x(0)$, $\{w(k)\}$ and $\{v(k)\}$. Every communication link in the network suffers an erasure at time k if and only if the corresponding communication link in the network for the system $\mathbb{S}_1$ suffers an erasure. Assume that the algorithm $\mathcal{A}_2$ is implemented for both the problems. Note that for problem $\mathcal{P}_2$, the quantity $\bar{x}(k|\{y(t)\}_{t=0}^{j})$ is precisely the MMSE estimate of $x(k)$ given the measurements $y(0)$, $y(1)$, $\cdots$, $y(j)$. At time k, the decoder for the system $\mathbb{S}_1$ has access to the quantity $\bar{z}^d(k|\{l(j)\}_{j=0}^{t^d(k)})$ and the decoder for the systems $\mathbb{S}_2$ has access to the quantity $\bar{x}^d(k|\{y(j)\}_{j=0}^{t^d(k)})$. For the system $\mathbb{S}_2$, denote by $e_2^d(k)$ the error between the state $x(k)$ and the estimate at the decoder $\hat{x}^d(k|\{y(j)\}_{j=0}^{t^d(k)})$. Also, denote for the system $\mathbb{S}_1$, the error between the state $z(k)$ and the estimate at the controller $\hat{z}(k|\{y(j)\}_{j=0}^{t^d(k)}, \{u(j)\}_{j=0}^{k-1})$ by $e_1^d(k)$. Due to Proposition 2, $e_1^d(k) = e_2^d(k)$ at every time step k. Also, because of Proposition 1, $e_1^d(k)$ determines the stability and performance of the system $\mathbb{S}_1$. Thus, to analyze the stability and performance for the closed-loop system, we can concentrate on the open-loop system $\mathbb{S}_2$. We will denote this estimation problem as $\mathcal{P}_2$.

Remark 5. Note that this problem has been posed merely for analytical convenience. We need not worry about the quantity being transmitted being stable for the problem $\mathcal{P}_2$ because as we have seen, the quantity for the closed-loop system $\mathbb{S}_1$ is stable. Of course, problem $\mathcal{P}_2$ may be of individual interest. Estimating an open-loop unstable process has been studied by many researchers recently for analog erasure channels under a variety of settings. It may be noted that the quantity transmitted in all these works is unstable.

---

[2]We do not require acknowledgements for the sensor-controller channel.

For the problem $\mathcal{P}_2$, denote the covariance of the error $e_2^d(k)$ at time k by $R^d(k) = E\left[e_2^d(k)(e_2^d(k))^T\right]$, where the expectation is taken over the initial condition $x(0)$, the process noise and the measurement noise. We can further compute the expectation with respect to the packet dropping process and denote $P^d(k) = E\left[R^d(k)\right]$. Finally, compute the steady-state error covariance $P^d(\infty) = \lim_{k\to\infty} P^d(k)$. If the limit exists and is bounded, we say that the estimate error is stable. Because of Proposition 1, the stability conditions for problems $\mathcal{P}_1$ and $\mathcal{P}_2$ are identical.

As in problem $\mathcal{P}_1$, for node d and time k, let $t^d(k)$ denote the time-stamp of the most recent observation used in estimating $x(k)$ at the destination node d. This time-stamp evolves according to (5). We have

$$P^d(k) = \sum_{l=0}^{k-1} E\left[e_2^d(k)(e_2^d(k))^T | t^d(k) = l\right] \Pr\left(t^d(k) = l\right),$$

where the expectation with respect to the packet dropping process has been evaluated. Thus, the effect of the packet dropping process enters through the distribution of the time-stamp of the most recent observation used in estimating $x(k)$. For future use, we denote the *latency* for the the node d at time k as $l^d(k) = k - 1 - t^d(k)$. Also, denote the MMSE estimate of $x(k)$ given all the measurements $\{y(0), y(1), \cdots, y(k-1)\}$ by $P(k)$. We can now rewrite the error covariance $P^d(k)$ as

$$P^d(k) = \sum_{l=0}^{k-1} \left[A^l P(k-l)\left(A^l\right)^T + \sum_{j=0}^{l-1} A^j R_w \left(A^j\right)^T\right] \\ \times \Pr\left(l^d(k) = l\right). \quad (9)$$

Thus, the stability of the system depends on how fast the probability distribution of the latency decreases. To analyze the stability, we use the following result from [8].

Proposition 4. *[From [8]] Consider a process of the form (6) being estimated using measurements from a sensor of the form (7) over a packet-dropping link that drops packets in an i.i.d. fashion with probability q. Suppose that the sensor transmits the MMSE estimate of the state at every time step. Then the estimate error at the receiver is stable in the bounded second moment sense if and only if $q|\rho(A)|^2 < 1$, where $\rho(A)$ is the spectral radius of the matrix A appearing in (6).*

We will also use the following definition. Consider every possible division of the nodes of the network into two sets with the source and the destination node being in different sets (also called a cut-set). For any such cut-set, let $p_1, p_2, \cdots, p_n$ denote the packet erasure probabilities of the edges that connect the two sets. Define the cut-set erasure probability as $p_{\text{cut set}} = p_1 p_2 \cdots p_n$. Then, the max-cut probability is given by

$$p_{\text{max-cut}} = \max_{\text{all possible cut-sets}} p_{\text{cut set}}. \quad (10)$$

A. Network with Links in Parallel

Let the source and the destination node be connected by a network with m links in parallel with the probability of packet drop in the i-th link being p_i. Since the same data is being transmitted over all the links, the distribution of the latency remains the same if the network is replaced by a single link that drops packets when all the links in the original network drop packets and transmits the information if even one link in the original network allows transmission. Thus the packet drop probability of this equivalent link is $p_1 p_2 \cdots p_m$. The necessary and sufficient condition for the error covariance to converge thus becomes $p|\rho(A)|^2 < 1$, where $p = p_1 p_2 \cdots p_m$.

B. Necessary Condition for Stability in Arbitrary Networks

Proposition 5. *Consider the problem $\mathcal{P}_2$ where the links drop packets i.i.d. in time and independently across links. A necessary condition for stability is $p_{\text{max-cut}}|\rho(A)|^2 < 1$.*

Proof: Consider a cut set of the given network $\mathcal{N}_1$, with the source s being in set A and the destination node d in set B and the links $1, 2, \cdots, n$ joining the sets A and B. Form another network $\mathcal{N}_2$ by replacing all links within the sets A and B by links that do not drop packets and do not consume one time step to transmit data. For any packet drop pattern, denote the information sets that the destination node has access to at time step k over networks $\mathcal{N}_1$ and $\mathcal{N}_2$ by $\mathcal{I}^{d,\mathcal{N}_1}(k)$ and $\mathcal{I}^{d,\mathcal{N}_2}(k)$ respectively. Clearly, the estimate error covariances at the destination node for the two networks are related by $P^d(k|\mathcal{I}^{d,\mathcal{N}_1}(k)) \geq P^d(k|\mathcal{I}^{d,\mathcal{N}_2}(k))$. Hence, by considering the stability of error covariance over network $\mathcal{N}_2$, we can obtain a necessary condition for the stability of error covariance over network $\mathcal{N}_1$. Since the edges within the source and the destination sets do not introduce any delay or error, $\mathcal{N}_2$ consists of the source and the destination joined by edges $1, 2, \cdots, n$ in parallel. The condition for the error covariance across $\mathcal{N}_2$ to converge is thus $p_{\text{cut set}}|\rho(A)|^2 < 1$, where $p_{\text{cut set}} = p_1 p_2 \cdots p_n$. This is, thus, a necessary condition for error covariance across $\mathcal{N}_1$ to be stable. By considering all cut-sets, we obtain that a necessary condition for the error covariance to converge is $p_{\text{max-cut}}|\rho(A)|^2 < 1$. $\square$

C. Network with Links in Series

Consider a network consisting of two links in series with probabilities of packet drop p_1 and p_2 respectively. Denote the nodes as v_1, v_2 and v_3 with v_1 being the source node and v_3 the destination. Also, denote the estimate at node v_i at time k by $\hat{x}^i(k)$. Let $e^1(k)$ be the error between $x(k)$ and $\hat{x}^2(k)$. Similarly let $e^2(k)$ be the error between $\hat{x}^2(k)$ and $\hat{x}^3(k)$. We are interested in the second moment stability of $e^1(k) + e^2(k)$. Clearly a sufficient condition is that both $e^1(k)$ and $e^2(k)$ individually be second moment stable. Applying Proposition 4, if $p_1 \mid \rho(A) \mid^2 < 1$, $e^1(k)$ would be stable. Now for the second link, we can consider the sensor

$$\hat{x}^2(k) = x(k) + e^1(k),$$

generating the measurements. The quantity transmitted by node 2 at any time step in the algorithm $\mathcal{A}_2$ can be seen to be the MMSE estimate of $x(k)$ given all the measurements $\{\hat{x}^2(j)\}_{j=0}^{k-1}$. Consequently, if $p_2 \mid \rho(A) \mid^2 < 1$, then the error $e^2(k)$ is stable. If p be the greater of the probabilities p_1 and

p_2, the sufficient condition thus is $p|\rho(A)|^2 < 1$. But this is identical to the necessary condition stated in Proposition 5. Thus the condition above is both necessary and sufficient.[3] This argument can be extended to any finite number of links in series. If there are m links in series with the probability of drop of the i-th link being p_i, then a necessary and sufficient condition for the estimate error to converge at the destination node is $p|\rho(A)|^2 < 1$, where $p = \max(p_1, p_2, \cdots, p_m)$.

D. Sufficient Condition for Arbitrary Networks

Proposition 6. *For problem $\mathcal{P}_2$ with the assumptions of Proposition 5 on the packet drops, the estimation error covariance with algorithm $\mathcal{A}_2$ is stable if $p_{\max-\text{cut}}|\rho(A)|^2 < 1$.*

Proof: First note that if a packet dropping link between two nodes v and u with probability of drop p_e is replaced by two parallel links with drop probabilities $p_i^{(1)}$ and $p_i^{(2)}$ such that $p_i = p_i^{(1)} p_i^{(2)}$, then $P^d(k)$ under algorithm $\mathcal{A}_2$ does not change. This is true because the probability distribution of the latency does not change with this replacement.

We will prove the proposition by exploiting a technique used in fluid networks. Consider the set $\Gamma = \{\gamma_1, \gamma_2, \cdots \gamma_m\}$ of all simple directed paths from the source to the destination in the network graph. Note that these paths may overlap. If the edge i is in path γ_j, we denote that as $i \in \gamma_j$. Now consider the following optimization problem

$$F_1^\star = \min_{\beta_j} \prod_{j=1}^m \beta_j \qquad (11)$$

$$\text{subject to} \prod_{j:i\in\gamma_j} \beta_j \geq p_i \quad \forall \text{ edges } i \qquad (12)$$

$$0 \leq \beta_j \leq 1 \quad \forall j = 1, 2, \cdots, m. \qquad (13)$$

A simple change of variables $\psi_j = -\log \beta_j$, transforms the above optimization problem into the following linear program in the variables ψ_j's.

$$F_2^\star = \max_{\psi_j} \sum_{j=1}^m \psi_j \qquad (14)$$

$$\text{subject to} \sum_{j:i\in\gamma_j} \psi_j \leq -\log p_i \quad \forall \text{edges } i$$

$$\psi_j \geq 0 \quad \forall j = 1, 2, \cdots, m.$$

The solutions $\{\beta_j^\star\}$ and $\{\psi_j^\star\}$ of the optimization problems (11) and (14) are related through $\psi_j^\star = -\log \beta_j^\star$. The structure of the linear program (14) is the same as the one used for finding the maximum flow possible in a fluid network [4, Page 59], which has the same topology as our packet dropping network with the capacity of the link i equal to $-\log p_i$. The solution to the problem of finding the maximum flow through a fluid network is given by the max-flow min-cut theorem as

$$F_2^\star = \min_{\text{all possible cut-sets}} \sum_{i\in cut} -\log p_i.$$

---

[3]Another approach to find the necessary and sufficient condition for stability is to evaluate P_∞^d for a series of links. This approach is taken in Section VI-A.

Thus for the optimization problem (11), the solution is

$$
\begin{aligned}
F_1^\star &= \max_{\text{all possible cut-sets}} \prod_{i\in cut} p_i \qquad (15) \\
&= \max_{\text{all possible cut-sets}} p_{\text{cut set}} \\
&= p_{\text{max-cut}}.
\end{aligned}
$$

Consider the paths in the set Γ. Form a new set $\mathcal{B}$ of all those paths γ_j's for which the associated optimal variable $\beta_j^\star$ is strictly less than one. Now form a new network $\mathcal{N}'$ as follows. The node set of $\mathcal{N}'$ is the union of those nodes of the original network $\mathcal{N}$ that are present on any path in $\mathcal{B}$. Each pair of nodes (u, v) in the node set of $\mathcal{N}'$ is connected by (possibly) multiple links. If an edge i between nodes u and v is present in a path $\gamma_j \in \mathcal{B}$, add an edge between nodes u and v in $\mathcal{N}'$ and assign an erasure probability $\beta_j^\star$. By considering all the edges in $\mathcal{N}$ and following this procedure, we construct the edge set of $\mathcal{N}'$. The following properties of $\mathcal{N}'$ are easily verified.

- By construction, $\mathcal{N}'$ is a union of edge-disjoint paths. Furthermore, for each path, the probabilities of packet drop on all the links are equal.
- By virtue of (15) and the procedure followed to construct $\mathcal{N}'$, the product of the probabilities of packet drop of the different paths is equal to the $p_{\text{max-cut}}$.
- Because of (12), for any pair of nodes that were connected by a link with erasure probability p in $\mathcal{N}$, the product of the probabilities of packet dropping of the links in $\mathcal{N}'$ connecting these two nodes is no less than p.

Therefore the estimate error covariance at the destination by following algorithm $\mathcal{A}_2$ in the original network $\mathcal{N}$ is less than or equal to the error covariance by following $\mathcal{A}_2$ in the new network $\mathcal{N}'$. Thus, to obtain a sufficient condition on stability, we can analyze the performance of $\mathcal{A}_2$ in the network $\mathcal{N}'$. For this we consider another algorithm $\mathcal{A}_3$ in which we assume that estimates on the disjoint paths in $\mathcal{N}'$ are routed separately. Thus, if a node lies on many paths, on each path it forwards only the packets it received on that path. The performance of $\mathcal{A}_3$ cannot be better than $\mathcal{A}_2$ since in $\mathcal{A}_2$ we send the most recent estimate received from different paths at any node.

Therefore, to prove the theorem, we only need to show the stability of estimation using protocol $\mathcal{A}_3$ assuming that the condition of Proposition 5 holds. Since we do not mix the estimates obtained from different paths in $\mathcal{A}_3$, the network can be considered as a collection of parallel paths, with each path consisting of links with equal drop probability. We further upper-bound the error covariance through two means:

1) Convert the network $\mathcal{N}'$ into a network $\mathcal{N}''$ by introducing links in the different paths from the source to the destination in $\mathcal{N}'$ such that each path consists of equal number of links, while retaining the property that every link in a path has the same probability of drop. Thus, we can consider nodes present in layers, with layer 0 corresponding to the source, layer 1 consisting of all nodes one hop away along the different paths from the source and so on. Let the destination be at layer m.
2) Let the nodes in level t have estimate error covariances $P_i^t(k)$'s at time k. When selecting messages to be transmitted to the nodes in level $t + 1$, we use the algorithm

$\mathcal{A}_4$, in which the nodes transmit the estimate that has the maximum $P_i^t(k)$[4].

Let us now prove that the condition in Proposition 5 is sufficient for the stability of the error covariance when algorithm $\mathcal{A}_4$ is executed over network $\mathcal{N}''$. Clearly $p_{\text{max-cut}} \mid \rho(A) \mid^2 < 1$ implies $p_i \mid \rho(A) \mid^2 < 1$ for the probability p_i corresponding to any path i. Thus the error covariance at the nodes in the layer 1 is stable. Repeating an argument similar to the one in Section V-C, we can obtain that all the nodes in layer 2, 3, $\cdots$, $m-1$ have stable estimate error covariance as well. Finally for the estimate at the destination node, all the estimates from the nodes in layer $m-1$ can be considered to be in parallel with each other. Since for this network $p_{\text{max-cut}} = \prod_{\text{all paths } j} p_j$, if $p_{max-cut} \mid \rho(A) \mid^2 < 1$, the estimate at the destination have bounded error covariance as well. □

Remark 6. We have provided a necessary and sufficient condition for the expected error covariance to remain bounded for a network of arbitrary topology. For any other causal data processing algorithm, it provides a necessary condition for stability. Let us, in particular, compare the stability conditions for the algorithm $\mathcal{A}_2$ to those for a simpler algorithm $\bar{\mathcal{A}}$ in which the intermediate nodes do not have *any* memory. At each time step k, the source node forwards the measurement $y(k-1)$. The intermediate nodes compare the time stamps of the measurements they received at the previous time step along different incoming edges and forward the most recent one. If they did not receive any measurement on the previous time step, they do not transmit anything. The probability that the destination node receives any particular measurement $y(k)$ from the source over the network is upper-bounded by the *reliability* [6] of the network. For a line network in which n edges each with drop probability p are combined in series, the probability that any measurement is received by the destination node is $q = 1 - (1-p)^n$. By a method similar to the one used in [20], it can be proven that a *necessary* condition for stability is $q|\rho(A)|^2 < 1$. With our optimal algorithm, the necessary and sufficient condition for expected estimate error covariance to be stable is $p|\rho(A)|^2 < 1$. As an example, for $n = 5$ links and drop probability $p = 0.2$, $q = 0.67$. Thus our algorithm improves the stability margin from $\rho(A) \leq 1.22$ for $\bar{\mathcal{A}}$ to $\rho(A) \leq \sqrt{5} \approx 2.23$.

VI. PERFORMANCE ANALYSIS

In this section we calculate the probability mass function of the latency at *any* node for various networks, assuming the erasures to be independent in time and uncorrelated across various links. This will allow us to study the performance of the algorithm $\mathcal{A}_2$. The latency at any node depends on the erasure patterns along different links in all the paths from the source to the node. Since the paths may be overlapping, the evaluation of the latency may, in general, be very complicated.

Let $Z_{uv}(k)$ denote the time elapsed since any information was received by the node v over the link (u,v), i.e.,

$$Z_{uv}(k) = \min\{j \geq 1 | \lambda_{uv}(k+1-j) = 1\}.$$

We define $Z_{uu}(k) = 1$. 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 $t^u(k - Z_{uv}(k))$. Since the erasures are independent across links and i.i.d. in time, $Z_{uv}(k)$ is a truncated geometric random variable with the mass function

$$\Pr\left(Z_{uv}(k) = i\right) = \begin{cases} (1-p_{uv})p_{uv}^{i-1}, & 1 \leq i < k \\ 1 - \sum_{i=1}^{k-1} \Pr\left(Z_{uv}(k) = i\right), & i = k. \end{cases}$$

We can express the latest information used by node v in terms of the variables $Z_{uv}(k)$. From (5) the time stamp is given by

$$t^v(k) = \max_{u \in \mathcal{N}_v \cup v} t^u(k - Z_{uv}(k)). \tag{16}$$

For analytical ease, we get rid of the truncation in the definition of the variable $Z_{uv}(k)$ by extending the definition of the time stamp $t^u(k)$. For all $k < 0$, define $t^u(k) = 0$. As an example, for the source node s, without extending the definition we have $t^s(k) = k - 1$ for $k \geq 1$. Using the extended definition, $t^s(k) = (k-1)^+$ for all k, where $x^+ = \max\{0, x\}$. For any node u, using the extended definition of $t^u(k)$ for all k, (16) continues to hold; however, $Z_{uv}(k)$'s are now independent random variables distributed according to a geometric distribution with parameters p_{uv}'s. Thus

$$\Pr\left(Z_{uv}(k) = i\right) = (1-p_{uv})p_{uv}^{i-1} \quad \forall i \geq 1, \forall k. \tag{17}$$

Since $Z_{uv}(k)$'s do not depend anymore on k, we omit this argument.

We can now begin to combine the variables Z_{uv}'s to calculate the latest information received by any node from the source. Along any path, this involves adding the times since last successful transmission along different paths. Since there may be multiple paths from the source, we need to compute the maximum such sum. From (16), we can write $t^v(k)$ in terms of the time-stamp at the source node $(k-1)^+$ as

$$t^v(k) = \max_{P:\text{an s-v path}} (k - 1 - \sum_{(u,v) \in P} Z_{uv})^+, \tag{18}$$

where the maximum is taken over all paths from source s to the node v. The latency at node v can now be written as

$$l^v(k) = k - 1 - t^v(k) = \min\{k-1, \min_{P:\text{an s-v path}} (\sum_{(u,v) \in P} Z_{uv})\}.$$

For the steady-state error covariance, we consider the steady-state behavior of the latency $l^v(k)$. As $k \to \infty$, the distribution of $l^v(k)$ approaches that of the variable l^v defined as

$$l^v = \min_{P:\text{an s-v path}} (\sum_{(u,v) \in P} Z_{uv}). \tag{19}$$

We now concentrate on the destination node[5]. For the destination node d, denote by l^d the *steady-state latency* of the network. From (9), the steady-state error covariance is

$$P^d(\infty) = \sum_{l=0}^{\infty} \Pr\left(l^d = l\right) \left[A^l P^\star A^l + \sum_{j=0}^{l-1} A^j R_w A^j \right], \tag{20}$$

---

[4] Note that all $P_i^t(k)$'s form an ordered set.

[5] For covariance at another node v, simply denote v as the destination node.

where $P^\star$ is the steady-state estimation error covariance of $x(k)$ based on $\{y(0), y(1), \cdots, y(k-1)\}$ and is the solution to the Discrete Algebraic Riccati Equation (DARE)

$$P^\star = AP^\star A^T + R_w - AP^\star C^T(CP^\star C^T + R_v)^{-1}CP^\star A^T.$$

Because of the observability and stabilizability assumptions, the rate of convergence of $P(0)$ to $P^\star$ is exponential and the substitution of $P^\star$ for $P(k)$ in (9) does not change the steady-state error covariance. On vectorizing (20) we obtain

$$\text{vec}(P^d(\infty)) = F(A\otimes A)\text{vec}(P^\star) + G(A\otimes A)\text{vec}(R_w), \quad (21)$$

where $A \otimes B$ is the Kronecker product of matrices A and B, $G(X)$ is the generating function of the complementary density function and $F(X)$ is the moment generating function $F(X)$ of the steady state latency l_d defined as

$$G(X) = \sum_{l=0}^{\infty} \Pr\left(l^d \geq l+1\right)X^l, \quad F(X) = \sum_{l=0}^{\infty} \Pr\left(l^d = l\right)X^l,$$

for an arbitrary matrix X. Note that

$$F(X) = (X - I)G(X) + I. \quad (22)$$

Proposition 7. *Consider the problem $\mathcal{P}_2$. Let the packet drops be independent from one time step to the next and across links. The minimum expected steady-state estimation error covariance at the receiver is given by (21).*

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, boundedness of $G(x)$ at $A \otimes A$ is equivalent to the boundedness of $G(x)$ (evaluated for a scalar x) at the square of the spectral radius of A.

A. Networks with Links in Series

In this case, the network consists of only one path from the source to the destination. Thus we have

$$F(X) = E\left[X^{l^d}\right] = E\left[X^{\sum_{(u,v)} Z_{uv}}\right],$$

where the summation is taken over all the edges in the path. Since the drops across different links are uncorrelated, the variables Z_{uv}'s are independent. Thus we have

$$F(X) = E\left[X^{\sum_{(u,v)} Z_{uv}}\right] = \prod_{(u,v)} E\left[X^{Z_{uv}}\right],$$

where we have used the independence of Z_{uv}'s. Since from (17) Z_{uv} is a geometric random variable,

$$E\left[X^{Z_{uv}}\right] = (1 - p_{uv})X\left(I - p_{uv}X\right)^{-1},$$

provided that $\rho(X)p_{uv} < 1$, where $\rho(X)$ is the spectral radius of matrix X. Therefore,

$$F(X) = E\left[X^{l_d}\right] = \prod_{(u,v)} \left[(1 - p_{uv})X\left(I - p_{uv}X\right)^{-1}\right].$$

Using partial fractions and the relation in (22), we then obtain

$$G(X) = \sum_{i=0}^{n-1} X^i + X^n \sum_{(u,v)} c_{uv}\frac{p_{uv}}{1 - p_{uv}}(I - p_{uv}X)^{-1},$$

where $\quad c_{uv} = (\prod_{(r,s)\neq(u,v)} (1 - \frac{p_{uv}}{p_{rs}}))^{-1}.$

Therefore the cost can be written as

$$\text{vec}(P(\infty)) = G(A \otimes A)\text{vec}(R_w)$$
$$+ \prod_{(u,v)} \left[(A \otimes A)\left(\frac{I - p_{uv}A \otimes A}{1 - p_{uv}}\right)^{-1}\right]\text{vec}(P^\star).$$

Moreover, the system is stable if for every link (u,v) we have $p_{uv}|\rho(A)|^2 < 1$ or equivalently $\max_{(u,v)} p_{uv}|\rho(A)|^2 < 1$. This matches with the condition in section V. Also note that for the case that some of p_{uv}'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

$$\text{vec}(P(\infty)) = (A \otimes A)^n \left(\frac{I - pA \otimes A}{1 - p}\right)^{-n}\text{vec}(P^\star)$$
$$+ \sum_{i=0}^{n-1}\left[\frac{p}{1 - p}(A \otimes A)^n\left(\frac{I - pA \otimes A}{1 - p}\right)^{-i-1}\right]\text{vec}(R_w)$$
$$+ \sum_{i=0}^{n-1}(A \otimes A)^i\text{vec}(R_w).$$

Finally, when there is only one link between the source and the destination, the steady state error covariance is the solution to the Lyapunov equation

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

B. Network of Parallel Links

Consider a network with the sensor connected to a destination node through n links with probabilities of packet drop $p_1, \ldots, p_n$. Since the same data is transmitted over all the links, using (19) the steady state latency is $l_d = \min_{1 \leq i \leq n}(Z_i)$. Since Z_i's are all independent geometrically distributed variables with parameters p_i's respectively, their minimum is itself geometrically distributed with parameter $p_{eq} = \prod_i p_i$. Thus $F(X)$ can be evaluated as

$$F(X) = (1 - p_{eq})X(I - p_{eq}X)^{-1},$$

and $G(X)$ can, in turn, be written as

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

Thus the steady-state error can be evaluated using (21). Note that convergence of $G(X)$ enforces for stability $\prod_i p_i|\rho(A)|^2 < 1$, which is the condition derived in Section V.

C. Arbitrary Network of Parallel and Serial Links

We can obtain the following two rules for the steady-state error covariance of any network derived from the parallel and serial concatenations of sub-networks. 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 $\mathcal{G}_1 \| \mathcal{G}_2$.

1) Suppose the network $\mathcal{G}$ can be decomposed as a series of two subnetworks $\mathcal{G}_1$ and $\mathcal{G}_2$. Since packet erasures in

the two subnetworks are independent of each other,

$$l^d(\mathcal{G}_1 \oplus \mathcal{G}_2) = l^d(\mathcal{G}_1) + l^d(\mathcal{G}_2).$$

Thus,
$$\begin{aligned}
F_{\mathcal{G}_1 \oplus \mathcal{G}_2}(X) &= E\left[X^{l^d(\mathcal{G}_1 \oplus \mathcal{G}_2)}\right] \\
&= E\left[X^{l^d(\mathcal{G}_1)+l^d(\mathcal{G}_2)}\right] \\
&= E\left[X^{l^d(\mathcal{G}_1)}\right] E\left[X^{l^d(\mathcal{G}_2)}\right] \\
&= F_{\mathcal{G}_1}(X) F_{\mathcal{G}_2}(X).
\end{aligned}$$

Finally using (22), the complementary density function of the network $\mathcal{G}$ is given by

$$\begin{aligned}
G_{\mathcal{G}_1 \oplus \mathcal{G}_2}(X) &= (X-I)^{-1}\left(F_{\mathcal{G}_1 \oplus \mathcal{G}_2}(X) - I\right) \\
&= (X-I)G_{\mathcal{G}_1}(X) G_{\mathcal{G}_2}(X) \\
&\quad + G_{\mathcal{G}_1}(X) + G_{\mathcal{G}_2}(X),
\end{aligned}$$

where in the second equation we have used the fact that $G(X)(X-I) = (X-I)G(X)$.

2) If the network $\mathcal{G}$ can be decomposed as parallel combination of two sub-networks $\mathcal{G}_1$ and $\mathcal{G}_2$, we have

$$l^d(\mathcal{G}_1 \| \mathcal{G}_2) = \min\{l^d(\mathcal{G}_1), l^d(\mathcal{G}_2)\}.$$

Once again, the erasures in the two subnetworks are independent of each other. Thus

$$\Pr\left(l^d(\mathcal{G}_1 \| \mathcal{G}_2) \geq l\right) = \Pr\left(l^d(\mathcal{G}_1) \geq l\right)\Pr\left(l^d(\mathcal{G}_2) \geq l\right).$$

Thus we see that if

$$G_{\mathcal{G}_1}(X) = \sum_{i=0}^{\infty} a_i X^i \qquad G_{\mathcal{G}_2}(X) = \sum_{i=0}^{\infty} b_i X^i,$$

then
$$G_{\mathcal{G}_1 \| \mathcal{G}_2}(X) = \sum_{i=0}^{\infty} a_i b_i X^i.$$

Thus, we can use (21) with the above two rules to derive the steady state error of any network consisting of links in series and parallel with each other.

D. Networks with Arbitrary Topology

Finding the distribution of the steady-state latency l_d of a general network is not an easy task because different paths may overlap. This can introduce dependency in the delays incurred along different paths. Thus, the calculation of the minimum delay and, hence the steady-state latency, becomes involved. However, we can provide upper and lower bounds on the performance. We first mention the following intuitive lemma without proof.

Lemma 2. *Let $P^{\infty}(\mathcal{G}, \{p_{uv}, (u,v) \in \mathcal{E}\})$ 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 $p_{uv}, (u,v) \in \mathcal{E}$. Then the expected steady-state error is non-increasing in p_{uv}'s, i.e., if $p_{uv} \leq q_{uv} \quad \forall(u,v) \in \mathcal{E}$*

$$P^{\infty}(\mathcal{G}, \{p_{uv}, (u,v) \in \mathcal{E}\}) \leq P^{\infty}(\mathcal{G}, \{q_{uv}, (u,v) \in \mathcal{E}\}),$$

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

1) Lower Bound: We can lower bound the steady-state error by making a subset of links erasure free. This is similar to the method we used to obtain a necessary condition for stability in section V. Thus once again consider any cut-set of the network. Setting the probability of erasure equal to zero for every link except those crossing the cut (i.e., of the form (u,v) where u is in the source set and v in the destination set) gives a lower bound on the error. Therefore,

$$P^{\infty}(\mathcal{G}, \{p_{uv}, (u,v) \in \mathcal{E}\}) \geq P^{\infty}(\mathcal{G}, \{q_{uv}, (u,v) \in \mathcal{E}\}),$$

where
$$q_{uv} = \begin{cases} p_{uv} & (u,v) \text{ crosses the cut} \\ 0 & \text{otherwise.} \end{cases}$$

Now $P^{\infty}(\mathcal{G}, \{q_{uv}, (u,v) \in \mathcal{E}\})$ can be evaluated using the results given above for a network of parallel links. By considering the maximum along all possible cut-sets, we obtain the closest lower bound.

2) Upper Bound: We use a method similar to the one used to obtain the sufficient condition for stability in Section V. In the proof of Proposition 6, it is shown that the performance of the network $\mathcal{N}$ is lower bounded by the performance of another network $\mathcal{N}'$ that has series and parallel links only and has the following properties:

- $\mathcal{N}$ and $\mathcal{N}'$ have the same node set.
- $\mathcal{N}'$ is the combination of edge-disjoint paths from the source to destination.
- The value of the max-cut in $\mathcal{N}'$ is the same as in the original network $\mathcal{N}$.

The performance of $\mathcal{N}'$ can be computed based on the results given above for arbitrary networks composed of subnetworks in series and parallel. This provides an upper bound on the performance of the original network.

VII. CORRELATED ERASURE EVENTS

Even though the algorithm $\mathcal{A}_2$ is optimal for any packet dropping pattern, the stability and performance analysis so far assumed that the erasure events are memoryless and independent across different links in the network. We now look at the effect of dropping these assumptions.

A. Markov erasure 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. Suppose that the packet drop event on link (u,v), denoted by $\lambda_{uv}(k) = 0$ evolves according to a Markov chain with transition matrix M_{uv} assumed to be 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 (5) in a similar fashion as (18) with Z_l being a geometric random variable with distribution

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

where α_{uv} is 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}. Thus, all the previous analysis goes through. In particular, the stability condition is

$$\max_{c:\text{s-d cut}} \prod_{(u,v)\in c} M_{uv}(1,1)|\rho(A)|^2 < 1.$$

If the initial distribution is not the stationary distribution, the variables $Z_{uv}(k)$ have time-varying distributions and the analysis fails. However, since for large k the Markov chains approach their stationary distribution, the stability condition remains unchanged.

B. 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 $\Pr(\lambda_{uv}, (u,v) \in \mathcal{E})$. Now $Z_{uv}(k)$'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 cut-set 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

$$p_{eq}(c) = \text{Prob}\left(\lambda_{uv} = 0, \quad \forall (u,v) \in \mathcal{B}(c)\right).$$

The value of the max-cut for the network is the maximum of $p_{eq}(c)$ over all the cuts, $p_{\mathrm{mc}}(\mathcal{G}) = \max_{\text{all cut-sets } c} p_{eq}(c)$. We can then show that the condition for stability of the system is $p_{\mathrm{mc}}(\mathcal{G})|\rho(A)|^2 < 1$. 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 $\mathcal{V}_r(t_0) = \{s\}$. Note that for every time-step between t_0 and $t_0 + T$, $\mathcal{V}_r(t)$ defines 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 $p_{\mathrm{mc}}(\mathcal{G})$ the probability of this event is at most $p_{\mathrm{mc}}(\mathcal{G})$. Therefore, we have

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

Thus for large T, the probability that at time $t_0 + T$ the destination node has not received the packet is upper bounded by $n(1 - p_{\mathrm{mc}}(\mathcal{G}))^n T^n p_{\mathrm{mc}}(\mathcal{G})^{T-n}$, where n is the number of nodes in the network. Now, the error covariance can be upper-bounded by considering that the network is only routing the latest packet generated at time $k - l$. The probability that the latency is larger than l thus grows like $f(l)p_{\mathrm{mc}}(\mathcal{G})^l$, where $f(l)$ is polynomial in l with bounded degree and the sufficiency of the stability condition follows. The necessity part involves similar ideas and is omitted.

VIII. CONCLUSIONS

In this paper, we considered the problem of estimation and control of a process across an arbitrary network of communication links. We identified an optimal information processing algorithm to be followed by each node in the network that allows the estimator to calculate the best possible estimate in the minimum mean square sense and the controller to minimize a quadratic cost. The recursive algorithm requires a constant amount of memory, processing and transmission at every node in the network per time step yet is optimal for any packet-dropping process and at every time step. It has numerous other desirable properties as well such as being able to take care of delays and packet reordering. We also carried out the stability and performance analysis for this algorithm.

REFERENCES

[1] P. Antsaklis and J. Baillieul, "Special issue on networked control systems," *IEEE Transactions on Automatic Control*, 49(9), Sep. 2004.
[2] P. Antsaklis and J. Baillieul, "Special issue on networked control systems technology," *Proceedings of the IEEE*, 95(1), Jan. 2005.
[3] B. Azimi-Sadjadi, "Stability of networked control systems in the presence of packet losses," in *Proceedings of 2003 IEEE Conference on Decision and Control*, Dec 2003, pp. 676-681.
[4] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver, *Combinatorial Optimization*. John Wiley and Sons, New York, 1998.
[5] A. F. Dana, V. Gupta, J. P. Hespanha, B. Hassibi and R. M. Murray, "Estimation over Communication Networks: Performance Bounds and Achievability Results," in *American Control Conference, 2007*, July 2007, pp. 3450-3455.
[6] K. Dohmen, "Inclusion-exclusion and network reliability," *The Electronic Journal of Combinatorics*, vol. 5, no. 1, Research paper 36, 1998.
[7] V. Gupta, T. H. Chung, B. Hassibi, and R. M. Murray, "On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage," *Automatica*, 42(2):251–260, February 2006.
[8] V. Gupta, D. Spanos, B. Hassibi, and R. M. Murray, "Optimal LQG control across a packet-dropping link," *Systems and Controls Letters*, vol. 56, no. 6, pp. 439-446, June 2007.
[9] C. N. Hadjicostis and R. Touri, "Feedback control utilizing packet dropping network links," in *IEEE Conf. on Decision and Control*, 2002, pp. 1205-1210.
[10] B. Hassibi, A. H. Sayed, and T. Kailath, *Indefinite-Quadratic Estimation and Control*. Studies in Applied and Numerical Mathematics, 1999.
[11] J. P. Hespanha, P. Naghshtabrizi and Y. Xu, "*Networked Control Systems: Analysis and Design,*" Proc. of IEEE, Special Issue on Networked Control Systems, vol. 95, no. 1, pp. 138-162, January 2007.
[12] O.C. Imer, S. Yuksel and T. Basar, "*Optimal Control of LTI Systems over Communication Networks*", Automatica, 42(9):1429-1440, Sep. 2006.
[13] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*. Prentice Hall, 2000.
[14] F.L. Lian, J.R. Moyne and D.M. Tilbury, "*Performance evaluation of control networks*", IEEE Control Systems Magazine 21(1):6683, 2001.
[15] Q. Ling and M. D. Lemmon, "Power spectral analysis of netwroked control systems with data droputs," *IEEE Transactions on Automatic control*, vol. 49, no. 6, pp. 955–960, June 2004.
[16] J. Nilsson, "Real-time control systems with delays," Ph.D. dissertation, Department of Automatic Control, Lund Institute of Technology, 1998.
[17] C. Robinson and P. R. Kumar, "Control over networks of unreliable links: Controller location and performance bounds," in *ConCom: Control over Communication Channels*, Cyprus, 2007.
[18] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla and S. S. Sastry, "*Foundations of Control and Estimation over Lossy Networks*", Proc. of IEEE, 95(1):163-187, January 2007.
[19] P. J. Seiler, "Coordinated control of unmanned aerial vehicles," Ph.D. dissertation, University of California, Berkeley, 2002.
[20] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, and S. Sastry, "Kalman filtering with intermittent observations," *IEEE Transactions on Automatic Control*, 49(9):1453–1464, Sep 2004.
[21] S.C. Smith and P. Seiler, *Estimation with lossy measurements: jump estimators for jump systems*. IEEE Transactions on Automatic Control, 48(12):1453-1464, 2003.
[22] S. Tatikonda, "Some scaling properties of large scale distributed control systems," in *IEEE Conf. on Decision and Control*, 3:3142-3147, 2003.
[23] W. Zhang, M. S. Branicky, and S. M. Philips, "Stability of networked control systems," *IEEE Contr. System Magazine*, 21(1):84-89, Feb 2001.