Submitted, 2006 International Symposium on Mathematical Theory of Networks and Systems (MTNS)
http://www.cds.caltech.edu/~murray/papers/2005r_gdhm06-mtns.html

Data Transmission over Networks for Estimation

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

***Abstract*— In this paper, we consider the following problem. Suppose a sensor is taking measurements of a dynamic process. It needs to communicate the information over a network of communication links that can drop packets stochastically. What is the optimal processing at each node in the network? We provide a strategy that yields the optimal performance at the cost of constant memory and processing at each node. We also provide conditions on the network for the estimate error covariance to be stable under this algorithm.** ***Keywords*— Estimation across networks**

I. Introduction and Motivation

Recently there has been a surge of interest in systems containing multiple sensors (see, e.g., [1], [2], [3] among others) that communicate with each other. Such systems can be used for a wide variety of applications ranging from environmental monitoring to surveillance and tracking. 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, association of measurements with targets in case multiple targets are present 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. In extreme cases, these effects can even cause an estimate error covariance or a control loop to go unstable. 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., [4], [5] and the references therein). Beginning with the seminal paper of Delchamps [6], quantization effects have been variously studied both in estimation and control context by Tatikonda [7], Elia and Mitter [8], Nair and Evans [9], Brockett and Liberzon [10], Hespanha et al [11] and many others. The effect of delayed packet delivery using various models for network delay has also been considered by many researchers, some representative examples being the works of Nilsson [12], Blair and Sworder [13], Luck and Ray [14] etc.

In this work, we will 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 sink node. However the links in the network stochastically drop packets. We focus on optimal information processing schemes that should be followed by the nodes of the network that will allow the sink to calculate the optimal estimate at every time step. 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 by [15], [16], [17] and the control performance by Seiler in [16] and by Ling and Lemmon in [18]. Approaches to compensate for the data loss were proposed by Nilsson [12], Hadjicostis and Touri [19], Ling and Lemmon [18], [20], Azimi-Sadjadi [21], Sinopoli et al. [22] and Imer et al. [23]. Sinopoli et al. [24] 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. The work was extended to multiple sensors by Liu and Goldsmith in [25] and Gupta et al. in [26].

Most of the above designs aimed at designing a packet-loss compensator. The compensator accepts those packets that the link succesfully transmits and comes up with 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 [27] and [28]. It was demonstrated that using encoders and decoders can improve both the stability margin and the performance of the system. Moreover for a given communication level, it can lead to reduced amount of communication. In this paper, we consider the design of encoders and decoders when the data has to be transmitted over a network of arbitrary topology. Clearly the problem is much more complicated than the case of a single communication link since there are potentially multiple paths from the source to the destination. We aim to find the optimal information processing scheme that each node can follow. The main result of this paper is a strategy that is optimal yet requires a constant amount of memory, processing and transmission by any node. We also analyze the stability of the error covariance for this strategy.

Division of Engineering and Applied Science, Caltech `vijay@cds.caltech.edu`
Division of Engineering and Applied Science, Caltech `amirf@caltech.edu`
Department of Electrical and Computer Engineering, University of California, Santa Barbara `hespanha@ece.ucsb.edu`
Division of Engineering and Applied Science, Caltech `murray@cds.caltech.edu`

The paper is organized as follows. In the next section, we set up the problem and state the various assumptions. Then, we identify the optimal processing and transmission algorithm. We then do a stability analysis of the algorithm to obtain conditions on the packet drop probabilities under which the estimate error at the sink retains a bounded covariance. We conclude with some avenues for future work.

II. Problem Setup

Consider a process evolving in discrete-time as

$$x_{k+1} = Ax_k + w_k, \tag{1}$$

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 R_w. The process is observed using a sensor that generates measurements of the form

$$y_k = Cx_k + v_k, \tag{2}$$

where $v_k \in \mathcal{R}^m$ is the measurement noise also assumed to be white, Gaussian with mean zero and covariance R_v. Furthermore, the noises v_k and w_k are assumed to be independent of each other. The sensor acts as the source node. The process needs to be estimated in the minimum mean square error (MMSE) sense at another node that is designated the sink node. The source and the sink are connected via a network of communication links. We can model the network as a graph in a natural way. The edges of the graph represent the communication links and are in general directed. We assume there are M edges or links present in the network. We number the nodes as 0, 1, 2, $\cdots$, $N+1$, with 0 denoting the source node and $N+1$ the sink node. For any node i, 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 make no a priori assumptions on the topology of the graph depicting the network.

The communication links are modeled using a packet erasure model. The links take in as input a vector of real numbers. With probability p_i, the i-th link drops the packet and yields an empty vector at the output. The rest of the time, it yields the same vector at the output as the input. We assume that the packets are dropped in an i.i.d. fashion. Moreover the packet drop processes in the different links are independent of each other. 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 the network is wired in the sense it can listen to all the messages coming from the incoming links without interference from each other.

Thus at every time-step k,

1) Every node computes a function of all the information it has access to at that time.
2) It transmits the function on all the out-going 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 sink node calculates the estimate of the current state x_k based on the information it posseses.
3) Every node then observes the messages 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, $\cdots$, y_{k-1}. Further even if there were no packet drops, if the sink node is d 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, $\cdots$, y_{k-d-1} till time $k-d-1$. We aim to solve the following problems:

1) Identify the optimal processing and transmission algorithm at the nodes that allow the sink to calculate the MMSE estimate. Clearly sending measurements alone might not be the optimal thing to do since in such a scheme, dropping a packet would mean loss of information that cannot be compensated for in the future. We are particularly interested in strategies that do not entail an increasing amount of memory and transmission at the nodes.
2) Identify the conditions that would lead to a stable estimate error at the sink node. Clearly at a high enough value of packet loss, the error would grow unbounded since not enough information is being received.

III. Optimal Encoding and Decoding

In this section we propose an algorithm to be followed by each node and prove that it is optimal in the sense defined in the above section. First we define the notation used in the description of the algorithm. For the node i, denote by I_k^i the information set to which it has access at time step k and that it can use to generate the message 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 $t = 0, 1, \cdots, k-1$. As an example, for the source node, denoted by $i = 0$,

$$I_k^0 = \{y_0, y_1, \cdots, y_{k-1}\}.$$

Based on the information set I_k^i, the i-th node can calculate its MMSE estimate of the state x_k. We denote the estimate by $\hat{x}^i_{k|I_k^i}$ or more shortly as $\hat{x}_k^i$, where it is understood that the estimate is based on the information set I_k^i. We denote the error covariance associated with this estimate by $P^i_{k|I_k^i}$ or more compactly as P_k^i. We aim to minimize the error P_k^{N+1}. Clearly for two information sets $I_{k,1}^i$ and $I_{k,2}^i$ related by $I_{k,1}^i \subseteq I_{k,2}^i$, we have $P^i_{k|I^i_{k,1}} \leq P^i_{k|I^i_{k,2}}$.

The packet drops occur according to a random process. Consider a $M \times 1$ vector α that tracks the packet drops in the network. The i-th component of the vector is a binary random variable that takes the value *received* when a packet is successfully transmitted on the i-th edge and *failed* otherwise. We refer to instantiations of the process as packet drop sequences. At time step k, for any packet drop sequence and for any node i, we can define a time t_k^i such that the packet drops did not allow any information transmitted by the source after t_k^i to reach the i-th node in time for it to be a part of I_k^i.

Now consider an algorithm in which at time step k, every node takes the following actions:

1) Calculate the estimate of state x_k based on the information set at the node.
2) Transmit its entire information set on the outgoing edges.
3) Receive any data on the incoming edges.
4) Update its information set and affix a time stamp corresponding to the time of the latest measurement in it.

By running this algorithm, the information set at each node will be of the form

$$I_k^i = \{y_0, y_1, \cdots, y_{t_k^i}\},$$

where $t_k^i < k$ has been defined above. This is the maximal information set that the node i can possibly have access to with any algorithm. For any other algorithm, the information set will be smaller than this since earlier packets, and hence, measurements might have been dropped. We denote the information set at the i-th node using this algorithm by $I_{k,\max}^i$. The error covariance at any node when it calculates its estimate of the state x_k is the least when the information set it has access to is $I_{k,\max}^i$. We will denote this algorithm by $\mathcal{A}_1$. The algorithm $\mathcal{A}_1$ requires an increasing amount of memory and transmission as time goes on. We will now describe an algorithm $\mathcal{A}_2$ that achieves the same performance at the expense of constant memory and transmission (modulo the transmission of the time stamp). The algorithm proceeds as follows. At each time step k, every node takes the following actions:

1) Calculate its estimate $\hat{x}_k^i$ 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 shown later.
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.

We claim that the algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ lead to identical state estimates at every node for the same packet drop sequence. To see this, consider the data coming along the n incoming edges to node i. In $\mathcal{A}_1$, it will be sets S_1, S_2, $\cdots$, S_n where set S_j is of the form

$$S_j = \{y_0, y_1, \cdots, y_{t_j}\}.$$

In addition the node will have access to its information set from the previous time step, which we denote by S_{n+1}. Let

$$t_m = \max\left(t_1, t_2, \cdots, t_{n+1}\right).$$

Then the estimate of the i-th node is simply

$$\hat{x}_k^i = \hat{x}^i_{k|S_m},$$

i.e., the estimate of x_k based on the measurement set $\{y_0,\ y_1,\ \cdots,\ y_{t_m}\}$. Now for the same packet drop sequence, in $\mathcal{A}_2$, the data coming along the incoming edge j to the node i will be the vector $\hat{x}^i_{k|S_j}$. In addition the node has access to its previous estimate, which we denote by $\hat{x}^i_{k|S_{n+1}}$. Since there is at least one set S_m that is the superset of all other sets, the estimate of x_k based on this data will simply be $\hat{x}^i_{k|S_m}$. Thus the estimate of the i-th node under either of the algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ is the same. But algorithm $\mathcal{A}_1$ leads to the minimum possible error covariance at each node. This allows us to state the following result.

Proposition 1: (Optimality of Algorithm $\mathcal{A}_2$): The algorithm $\mathcal{A}_2$ is optimal in the sense that it leads to the minimum possible error covariance at any node at any time step.

a) Remarks::

1) The step of calculating the estimate at each node in the algorithm $\mathcal{A}_2$ can be implemented 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 on edge j be t_j with

$$t_m = \max_j t_j.$$

Also let the time-stamp corresponding to the previous estimate at node i be t_{n+1}. Then

- If $m > n+1$, the node calculates $\hat{x}_k^i = \hat{x}_{k|S_m}^i$ as its estimate and sets the time stamp to m.
- If $m \leq n+1$, the node discards the incoming data. It time-updates its own estimate as $\hat{x}_k^i = A\hat{x}_{k-1}^i$ and sets the time stamp to $n+1$.

Thus the processing can be done as a switched linear filter.

2) Note that we have proved the result for *any* packet drop sequence. Thus the algorithm $\mathcal{A}_2$ is optimal for any packet drop pattern, i.e., irrespective of whether the packet drops are occuring 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. We also do not assume any knowledge of the statistics of the packet drops at any of the nodes. In the next section, we will adopt a particular model of the packet drops and analyze the stability of the estimate error covariance under our strategy.
3) We have proved that the algorithm is optimal for any node. Thus we do not need to assume only one sink. The algorithm is also optimal for multiple sources if all sources have access to measurements from the same sensor. For multiple sources with each source obtaining measurements from its own sensor, the problem remains open.
4) Note 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.
5) The communication requirements can be reduced somewhat by adopting an event-based protocol in which a node transmits only if it updated its estimate based on data arriving on an incoming edge. This will not degrade the performance but reduce the number of transmissions, especially if packet drop probabilities are high.
6) If there are finite delays in the links, the algorithm remains optimal irrespective of the possibility of packet rearrangements. Further if the graph is finite, the stability conditions of the algorithm do not change.

IV. Stability Analysis for Arbitrary Graphs

In this section, we compute the conditions for the estimate error at the sink node to be stable under algorithm $\mathcal{A}_2$ (or equivalently $\mathcal{A}_1$) when the source and the sink are connected with a network of arbitrary topology. We assume that packets are dropped in each link in an i.i.d. fashion with the loss probability of the i-th link being p_i. We also assume the packet drop processes in two different links to be independent of each other.

We will consider the stability in the bounded second moment sense. Thus for a sink node trying to estimate a process of the form (1), denote the error at time step k as

$$e_k = x_k - \hat{x}_k,$$

where $\hat{x}_k$ is the estimate of the node. We can compute the covariance of the error e_k at time k as

$$P_k = E\left[x_k x_k^T\right],$$

where the expectation is taken over the initial condition x_0, the process noise w_j, the measurement noise v_j and the packet dropping sequence in the network. We consider the estimate to be stable if the covariance P_k remains bounded as $k \to \infty$. We use the following result from [27].

Proposition 2: (Proposition 4 from [27]) Consider a process of the form (1) being estimated using measurements from a sensor of the form (2) over a packet-dropping link that drops packets in an i.i.d. fashion with probability q. Suppose that the sensor calculates the MMSE estimate of the measurements at every time step and transmits it over the channel. Then the estimate error is stable in the bounded second moment sense if and only if

$$q|\lambda|^2 \leq 1,$$

where λ is the eigenvalue with the maximum magnitude of the matrix A appearing in (1).

From now on, we will denote the eigenvalue with the maximum magnitude of the matrix A by λ. We begin by considering a network consisting only of links in parallel. Consider the source and the sink node being connected by a network with m links in parallel with the probability of packet drop in link i being p_i. Since the same data is being transmitted over all the links, the network can be 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 diverge thus becomes

$$p|\lambda|^2 \leq 1,$$

where

$$p = p_1 p_2 \cdots p_m.$$

Using this result, we can obtain a necessary condition for stability for general networks as follows.

Proposition 3: Consider a process of the form (1) being observed using a sensor of the form (2) through an arbitrary network of packet dropping links with drop probabilities p_i's. Consider every possible division of the nodes of the network into two sets with the source and the sink node being in different sets (also called a cut-set). For any such division, let $\gamma_1, \gamma_2, \cdots, \gamma_p$ denote the packet erasure probabilities of the edges that connect the two sets. Define the cut-set erasure probability as

$$p_{\text{cut set}} = \gamma_1 \gamma_2 \cdots \gamma_p.$$

Then a neccesary condition for the error covariance to converge is

$$p_{\text{network}} |\lambda|^2 \leq 1,$$

where p_{network} is the network erasure probability defined as

$$p_{\text{network}} = \max_{\text{all possible cut-sets}} p_{\text{cut set}}.$$

Proof: If the given network $\mathcal{N}_1$ is stable, it will remain stable if we substitute some of the edges by perfect edges that do not drop packets. Consider a cut set C, with the source being in set A and the sink in set B and the links γ_1, γ_2, $\cdots$, γ_p joining the sets A and B. Form another network $\mathcal{N}_2$ by replacing all links within the sets A and B by perfect links, i.e., links that do not drop packets. Then $\mathcal{N}_2$ consists of the source and the sink joined by edges γ_1, γ_2, $\cdots$, γ_p in parallel. The condition for the error covariance across $\mathcal{N}_2$ to converge is thus

$$p_{\text{cut set}} |\lambda|^2 \leq 1,$$

where

$$p_{\text{cut set}} = \gamma_1 \gamma_2 \cdots \gamma_p.$$

This is thus a necessary condition for error covariance across $\mathcal{N}_1$ to be stable. One such condition is obtained by considering each cut-set. Thus a neccesary condition for the error covariance to converge is

$$p_{\text{network}} |\lambda|^2 \leq 1,$$

where

$$p_{\text{network}} = \max_{\text{all possible cut-sets}} p_{\text{cut set}}.$$

■

We now proceed to prove that the condition stated above is sufficient as well for stability. We already know that the condition is necessary and sufficient for a network with all links in parallel. Consider a case where the network consists of two links in series, with probability of packet drops p_1 and p_2. Denote the nodes as N_1, N_2 and N_3 with N_1 being the source node and N_3 the sink. Denote the estimate at node N_i at time k by $\hat{x}_k^i$. Also let e_k^1 be the error between x_k and $\hat{x}_k^2$. Similarly let e_k^2 be the error between $\hat{x}_k^2$ and $\hat{x}_k^3$. We are interested in the second moment stability of $e_k^1 + e_k^2$. Clearly a sufficient condition is that both e_k^1 and e_k^2 individually be second moment stable. Applying Proposition 2, this translates to the condition

$$\begin{aligned} p_1 |\lambda|^2 &\leq 1 \\ p_2 |\lambda|^2 &\leq 1. \end{aligned}$$

If p be the greater of the probabilities p_1 and p_2, the sufficient condition thus is

$$p|\lambda|^2 \leq 1.$$

But this is identical to the necessary condition stated in Proposition 3. Thus in this case as well, the condition is both necessary and sufficient. Clearly this argument can be extended to any 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 diverge at the sink node is

$$p|\lambda|^2 \leq 1,$$

where

$$p = \max(p_1, p_2, \cdots, p_m).$$

We use the conditions for parallel and series networks to prove the following result for arbitrary networks.

Proposition 4: Consider the assumptions of Proposition 3 on the process and the network. Then algorithm $\mathcal{A}_2$ will stabilize the process in (1) given that

$$p_{\text{network}}|\lambda|^2 \leq 1$$

Proof: As shown in Section III, the estimate of the i-th node of x_k, assuming that algorithm $\mathcal{A}_2$ is used, is given by

$$\hat{x}_k^i = \hat{x}_{k|t_k^i}$$

where $t_k^i < k$ as defined earlier is the last time index at which information is received at node i. The expected error covariance matrix at node i and time k can be written as

$$P_k^i = \sum_{t=0}^{k} \text{Cov}\,(x_k - \hat{x}_{k|t})\text{Pr}\,(t_k^i = t)$$

Thus the dependency of the expected error covariance on the network parameters shows up only in the probability distribution of t_k^i.

Now note that if a packet dropping link between two nodes v and u with probability of drop p_i is replaced by two parallel links with drop probabilities $p_i^{(1)}$ and $p_i^{(2)}$ such that $p_i = p_i^{(1)}p_i^{(2)}$, the average error covariance of the estimation under algorithm $\mathcal{A}_2$ will not change at any node. This is true simply because the probability distribution of t_k^u will not change with this replacement.

Next consider the set $\Gamma = \{\gamma_1,\, \gamma_2,\, \cdots\, \gamma_m\}$ of all simple directed paths from the source to the sink in the network graph. An edge i may be in more than one of these paths. If the edge i is in path γ_j, we will denote that as $i \in \gamma_j$. Consider the following optimization problem

$$\min_{\beta_j} \prod_{j=1}^{m} \beta_j, \tag{3}$$

subject to the following constraints

$$\begin{aligned} &\prod_{i\in\gamma_j} \beta_j \geq p_i \quad \forall \text{edges } i \\ &1 \geq \beta_j \geq 0 \quad \forall j = 1, 2, \cdots, m. \end{aligned} \tag{4}$$

A simple change of variables

$$\psi_j = -\log \beta_j, \tag{5}$$

transforms the above optimization problem into the following linear program in the variables ψ_j's.

$$\max_{\psi_j} \sum_{j=1}^{m} \psi_j \tag{6}$$

subject to

$$\begin{aligned} &\sum_{i\in\gamma_j} \psi_j \leq -\log p_i \quad \forall \text{edges } i \\ &\psi_j \geq 0 \quad \forall j = 1, 2, \cdots, m. \end{aligned}$$

The solutions of the optimization problems (3) and (6), denoted by $\{\beta_j^\star\}$ and $\{\psi_j^\star\}$, are related through the relation

$$\psi_j^\star = -\log \beta_j^\star.$$

The structure of the linear program (6) is the same as the one used for finding the maximum flow possible in a fluid network [31, 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 well-known to begiven by the max-flow min-cut theorem. Using this fact, we see that the solution to the optimization problem (6) is given by

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

Thus for the optimization problem (3), the solution is given by

$$\begin{aligned} \beta_j^\star &= \max_{\text{all possible cut-sets}} \prod_{i\in cut} p_i \\ &= \max_{\text{all possible cut-sets}} p_{\text{cut set}} \\ &= p_{\text{network}}, \end{aligned} \tag{7}$$

where $p_{\text{cut set}}$ and p_{network} have been defined before.

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. The remaining paths in Γ have equivalent erasure probability as unity and can thus be ignored. Now form a new network $\mathcal{N}'$ as follows. For ease of exposition, we will refer to the original network as $\mathcal{N}$. The node set of $\mathcal{N}'$ is the union of the nodes of $\mathcal{N}$ that are present on any path in $\mathcal{B}$. To form the edge set, we proceed as follows. Each pair of nodes (u, v) in the node set of $\mathcal{N}'$ is connected by (possibly) multiple links. Consider the edges in $\mathcal{N}$. If an edge i between two nodes u and v is present in a path $\gamma_j \in \mathcal{B}$, we add an edge between nodes u and v in $\mathcal{N}'$ and associate with it 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, $\mathcal{N}'$ can be presented as union of edge-disjoint paths. Each path in $\mathcal{N}'$ corresponds to one path in $\mathcal{B}$. Furthermore, for each path, the probabilities of packet drop on all the links of that path are equal.
- By virtue of (7) and the procedure followed to construct $\mathcal{N}'$, the product of the probabilities of packet drop of the different paths is equal to the equivalent probability of the network, p_{network}, for the network $\mathcal{N}$.
- For any pair of nodes that were connected by a link in $\mathcal{N}$, the product of the probabilities of packet dropping of the links in $\mathcal{N}'$ connecting these two nodes is greater than or equal to the drop probability of the link between the same pair of nodes in $\mathcal{N}$. This can be seen from the first inequality constraint of (4).

Therefore the estimate error covariance at the sink 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, which we denote as $\mathcal{A}_3$. In this algorithm we consider the disjoint paths given in $\mathcal{N}'$ and assume that estimates on different paths are routed separately. Thus if a node lies on many paths, on each path it forwards the packets it received on that path only. Clearly the performance $\mathcal{A}_3$ 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 compared to forwarding the estimates on different paths separately from each other.

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 3 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. Therefore using the stability analysis of serial networks presented earlier, each path (from a stability point of view) can be viewed as an erasure channel with drop probability equal to the drop probability of one link in that path. Using the stability analysis of parallel networks, we see that the stability of the new network under protocol $\mathcal{A}_3$ operation is equivalent to the stability of a packet erasure link with probability of erasure equal to the product of the drop probabilities of different path, which as mentioned earlier is equal to the network erasure probability defined in Proposition 3. Therefore, assuming that the network erasure probability satisfies

$$p_{\text{network}}|\lambda|^2 \leq 1, \tag{8}$$

the network $\mathcal{N}'$ is stablizable under protocol $\mathcal{A}_3$. But the performance of $\mathcal{A}_3$ cannot be better than of $\mathcal{A}_2$. Thus $\mathcal{N}'$ is stable under $\mathcal{A}_2$. Therefore the original network $\mathcal{N}$ is stable under protocol $\mathcal{A}_2$ assuming (8) is satisfied. This completes the proof. ■

Unicast Networks: So far, we have assumed that the topology of the network, as given by the graph, was fixed. Any node could transmit a message on all the out-going edges. We can consider networks that are unicast in the sense that each node should choose one out of a set of possible edges to transmit the message on. Thus the problem is two-fold.

1) Choose the optimal path for data to flow from the source node to the sink node.
2) Find the optimal information processing scheme to be followed by each node.

By a procedure similar to above, we can prove that there is a separation between the two parts of the problem such that given any path, the optimal processing strategy is the algorithm $\mathcal{A}_2$ described above. 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, 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

$$p_{\text{path}} = \max_{j\text{: edge } j \text{ is in the path}} p_j.$$

Thus the shortest path problem is to find the path that has the minimum p_{path} among all the paths. Choosing the shortest path in a graph is a standard problem. The condition for stability of the estimate is also easily obtained.

V. 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 and identified an optimal strategy that requires a constant amount of memory, processing and transmission at every node in the network. We carried out the stability analysis for this algorithm for arbitrary networks.

This work can be extended in many ways. We are currently working on the performance analysis of the algorithm. Also, for unicast networks, we solved the routing problem when stability is the metric. Another possible extension is to calculate the optimal routing through the network if the metric is performance. Finally we have ignored issues of quantization so far. If we include constraints of a limited bit rate into the framework, the problem is much harder. The work of Sahai [29] and Ishwar et al [30] may be relevant to this problem. In the future, we would like to explore these connections.

REFERENCES

[1] "Distributed Multi-robot Localization," S. I. Roumeliotis and G. A. Bekey, *IEEE Transactions on Robotics and Automation*, 18(5), 2002, pp. 781-795.

[2] "An Introduction to Multisensor Data Fusion," D. L. Hall and J. Llinas, *Proceedings of the IEEE*, 85(1), 1997, pp. 6-23.

[3] "Distributed Detection with Multiple Sensors: Part I - Fundamentals", R. Viswanathan and P. K. Varshney, *Proceedings of the IEEE*, 85(1), 1997, pp. 54-63.

[4] "Special Issue on Networks and Control," L. Bushnell (Guest Editor), *IEEE Control Systems Magazine*, 21(1), Feb 2001.

[5] "Special Issue on Networked Control Systems," P. Antsaklis and J. Baillieul (Guest Editors), *IEEE Transactions on Automatic control*, 49(9), Sept 2004.

[6] "Stabilizing a Linear System with Quantized State Feedback," D. F. Delchamps, *IEEE Transactions on Automatic Control*, 35, 1990, pp. 916-924.

[7] "Control under Communication Constraints," S. Tatikonda, PhD Thesis, MIT, Cambridge, MA 2000.

[8] "Stabilization of Linear Systems with Limited Information," N. Elia and S. K. Mitter, *IEEE Transactions on Automatic Control*, 46(9), 2001, pp. 1384-1400.

[9] "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.

[10] "Quantized feedback Stabilization of Linear Systems", R. W. Brockett and D. Liberzon, *IEEE Transactions on Automatic Control*, 45(7), 2000, pp. 1279-89.

[11] "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.

[12] "Real-time Control Systems with Delays," J. Nilsson, PhD Thesis, Department of Automatic Control, Lund Institute of Technology, 1998.

[13] "Feedback Control of a Class of Linear Discrete Systems with Jump Parameters and Quadratic Cost Criteria," W. P. Blair and D. D. Sworder, *International Journal of Control*, 21(5), 1975, pp. 833-841.

[14] "An Observer-based Compensator for Distributed Delays", R. Luck and A. Ray, *Automatica*, 26(5), 1990, pp. 903-908.

[15] "Control of Asynchronous Dynamical Systems with Rate Constraints on Events", A. Hassibi, S. P. Boyd and J. P. How, *Proceedings of IEEE Conference on Decision and Control*, 1999, pp. 1345-1351.

[16] , "Coordinated Control of Unmanned Aerial Vehiclea", P. Seiler, PhD Thesis, University of California, Berkeley, 2001.

[17] "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.

[18] "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.

[19] "Feedback Control Utilizing Packet Dropping Network Links", C. N. Hadjicostis and R. Touri, *Proceedings of the IEEE Conference on Decision and Control*, 2002.

[20] "Optimal Dropout Compensation in Networked Control Systems", Q. Ling and M. D. Lemmon, *Proceedings of the IEEE Conference on Decision and Control*.

[21] "Stability of Networked Control Systems in the Presence of Packet Losses", *Proceedings of IEEE Conference on Decision and Control*, 2003.

[22] "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.

[23] "Optimal Control of Dynamical Systems over Unreliable Communication Links", O. C. Imer, S. Yuskel and T. Basar, *NOLCOS*, 2004.

[24] "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.

[25] "Kalman Filtering with Partial Observation Losses", X. Liu and A. Goldsmith *Proceedings of the IEEE Conference on Decision and Control*, 2004.

[26] "On a Stochastic Sensor Selection Algorithm with Applications in Sensor Scheduling and Sensor Coverage", V. Gupta, T. H. Chung, B. Hassibi and R. M. Murray, *Automatica*, Accepted Sept. 2005.

[27] "On LQG Control Across Packet-Dropping Links," V. Gupta, D. Spanos, B. Hassibi and R. M. Murray, *System and Control Letters*, submitted July 2005.

[28] "Networked Control Systems: Analysis and Design", J. Hespanha, P. Naghshtabrizi and Y. Xu, *Submitted*, 2005.

[29] "Anytime Information Theory", A. Sahai, PhD Thesis, MIT, Cambridge, MA 2001.

[30] "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.

[31] "Combinatorial Optimization", W. J. Cook, W. H. Cunningham, W. R. Pulleyblank and A. Schrijver, John Wiley and Sons, New York, 1998.