Optimizing Controller Location in Networked Control Systems with Packet Drops*

C. L. Robinson[1] and P. R. Kumar[2]

[1] Department of Industrial and Enterprise Systems Engineering, and the Coordinated Science Lab, University of Illinois at Urbana-Champaign, clrobnsn@uiuc.edu

[2] Department of Electrical and Computer Engineering, and the Coordinated Science Lab, University of Illinois at Urbana-Champaign, prkumar@uiuc.edu

Abstract

In networked control, there is locational freedom in choosing the node at which to locate the controller, so as to mitigate the effects of packet losses in the network. What is the optimal location for the placement of the control logic? Second, what is the optimal control law in that position? The difficulty in answering these two questions is that analysis of optimality in networked control systems subject to random packet drops suffers from Witsenhausen's 'non-classical information pattern'. Thus, the general problem is considered intractable.

We make headway on this problem by using a "Long Packet Assumption", *LPA*, which allows packets to be arbitrarily long. This is not intended for implementation, but only to develop a lower bound on the cost. In particular, under this assumption the optimal controller location can be shown to be collocated with the actuator. For this position, under the *LPA*, we can also calculate the optimal cost, which is then a lower bound on the optimal cost for the original problem for all locations. Despite the apparent strength of the *LPA*, we have found that this lower bound is often close to currently realizable upper bounds. This establishes the near optimality of currently implementable controllers in such instances.

Using the lower bound on cost we obtain a necessary condition for stabilizability over *all* controller locations. This condition matches known sufficient conditions for some special cases, thus establishing a necessary and sufficient condition for location optimized stabilizability of networked control systems with packet loss.

* This material is based upon work partially supported by NSF under Contract Nos. ECCS-0701604, CCR-0325716 and CNS 05-19535, USARO under Contract No. DAAD19-01010-465, DARPA/AFOSR under Contract No. F49620-02-1-0325, DARPA under Contact No. N00014-0-1-1-0576, and AFOSR under Contract No. F49620-02-1-0217.

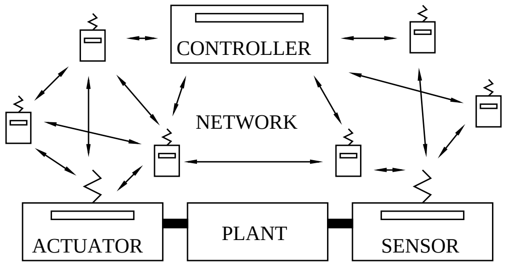

Fig. 1. A wireless networked control system. The controller logic and plant are connected by a network which may delay or lose packets sent from the sensor to the controller ('up' link) or from the controller to the actuator ('down' link).

1 Introduction

We may be on the cusp of seeing networked control systems deployed over wireless networks. In these systems, components communicate with each other using the network, as shown Figure 1; see [1, 2, 11, 12]. Efforts are already underway to develop *deployment mechanisms* for wireless networked control systems [8, 6, 5]. In this paper we consider the issue of optimal *policies* when control is exercised over wireless networks. A key limitation of the wireless medium is channel fading and interference which result in unpredictable packet loss. This loss of information will have a detrimental effect on performance. It is therefore important to mitigate the effect of packet drops. One way to do this is by fully exploring the capabilities of networked control, which allows choice of the *location* where the control law is computed.

In this paper we address the following questions. (1) What is a good controller placement? Closer to the sensor, actuator or somewhere in between? (2) What is the optimal control law for that optimal placement? We shall henceforth refer to this as the *location optimized controller*. (3) What are the conditions for stabilizability of the plant when control is exercised over the wireless network? (4) What is the best performance achievable?

Analysis of these problems is difficult because, as a consequence of the random packet losses, the information pattern that results is *non-classical*. Such problems were studied in [33, 15, 16] decades ago, and it is generally recognized that computing optimal control laws is intractable [23]. Thus, developing an optimal controller for a given location of the controller, never mind optimizing over controller position, is already a difficult problem. Some of the issues are briefly illustrated by the example in Figure 2.

We present a brief literature review in Section 2 and then formally introduce the problem in Section 3. We then consider the problem of optimal controller *placement* in Section 4 . In order to make the problem of obtaining a lower bound tractable, we consider a theoretically convenient assumption called the 'Long Packets Assumption' (*LPA*). The *LPA* has been considered in [26, 25], and is discussed in Section 4.1. The assumption allows packets to be infinitely long. This approach is *not* intended to be implemented, but rather to allow us to obtain a lower bound on cost for the location optimized controller. In [14, 13], a "maximal

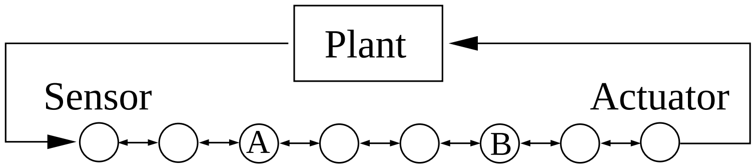

Fig. 2. A string of nodes separated by i.i.d. erasure channels. Two potential controller positions are identified by *A* and *B*. As a result of packet drops, controller *A* will receive more observations from the sensor than the controller at *B*. However, control commands from *B* will arrive at the actuator more often than those from *A*. Consequently, in choosing the controller's location we must choose between having *greater knowledge of the system state* (as with position *A*) or having a *greater impact on the system dynamics* (as with position *B*). This requires us to solve for the optimal costs for the two positions and then compare them. However, each of these problems is intractable.

information set," similar to the information contained in a long packet is considered. It is shown how by using an encoder consisting of a modified Kalman filter at the sensor, a sufficient statistic that captures its information content can be transmitted without requiring packets of growing length, and decoded at the controller. This scheme requires processing logic for the modified Kalman filter at the sensor. Our result for the optimal controller for the average cost per time in Theorem 6 provides a complete proof of existence and optimality. In particular, we take into account the technical issues associated with the average cost per time problem, for which the average cost dynamic programming equation does not always admit a solution [20].

Under the *LPA* we are able to show that optimal placement of the controller is to collocate it with the actuator. Our arguments are based solely on the problems information structure, and not the actual control law or cost, which is what makes this determination possible. In Section 6 we provide a complete proof, including existence, of the optimal control policy under the LPA for the collocated controller-actuator system for the infinite horizon average cost per time problem. The optimal control law exhibits a slightly sharpened separation structure. This result allows us establish a necessary condition for stabilizability of the networked control system over all controller locations.

We can also upper bound the optimal cost. For this, all we need to do is consider any implementable scheme without the *LPA*. For this purpose we can exploit the fact that when the controller is located at the actuator, solving for the optimal controller is actually tractable even when the *LPA* does not hold. Indeed, such an arrangement has already been studied, and bounds on critical packet drop probability for estimator and controller stability for it have been established [29, 30, 17, 13, 28]. The lower bound on the packet drop rate for stability in these works is sufficient for stability for the location optimized controller. Our results (see Section 8) yield that this sufficient condition is in fact also necessary in some special cases [17, 29, 35] even when location of controller can be optimized.

In Section 9 we present some simulation results. Despite the apparent strength of the *LPA*, the upper and lower bounds are in fact fairly close in several examples that we have tested.

2 Related Work

The results presented in the literature depend heavily upon the assumptions made regarding controller location, the presence of delays, and the existence of a packet delivery notification mechanism. The first topic, controller location, has three possible choices: (i) Collocation of the controller and actuator, (ii) collocation of the controller and sensor, and (iii) placement such that the network delivers both sensor information to the controller, and control commands to the actuator.

Controllers which are collocated with the actuator are the simplest to analyze as the controller has access to all past implemented control actions. It also has knowledge of when observation packets were dropped, since an observation arrival is itself a delivery notification. This is in fact a 'classical information structure' [33] where the controller has full access to all available observations and past control actions. Under this structure, the optimal controller can be decomposed into forming the optimal state estimate, and using a certainty equivalent optimal controller.

In this paper, as in [29, 35, 17], we will not consider random packet delay. References that include delay are [3, 31, 24, 22]. The general no-delay problem is one of forming a stable Kalman filter or LQG controller in the presence of packet loss. A sufficient condition for the existence of a stable Kalman filter estimator is determined as a function of the packet loss rate and the system dynamics in [29]:

Theorem 1. *([29]). Suppose C is invertible in (3). A sufficient condition for a stable Kalman filter estimator is:*

$$\lambda < \frac{1}{\overline{\mid eig(A) \mid}^2}, \tag{1}$$

where $\overline{\mid eig(A) \mid}$ *represents the largest eigenvalue magnitude of A and* λ *is the packet drop probability.*

The LQG control result has C replaced by B. The work in [35] relaxes the invertibility condition on C but requires processing of the raw sensor measurements before transmission. This result has been extended to the case of partial observations in [21]. Our results in Corollary 1 show the condition in Theorem 1 is necessary in general, even without requiring invertibility of C.

In [17] the packet drop probabilities for 'up-link' (from sensor to controller), λ_α, and 'down-link' (from controller to actuator), λ_β, are considered. Under a TCP-type transport protocol (discussed later) a sufficient condition for control loop stability is obtained as:

$$\max\{\lambda_\alpha, \lambda_\beta\} < \frac{1}{\overline{\mid eig(A) \mid}^2}, \tag{2}$$

This is a consequence of the duality between the optimal state estimation and optimal control problems. In fact, results such as in [29, 17] depend on the stability of a 'Modified' Ricatti equation. In the state estimation problem, this takes the form: $P_{k+1} = A'P_kA + Q - \lambda A'P_kC'(CP_kC' + R)^{-1}CP_kA$, where λ represents the packet drop probability. The equation for a stable controller is defined analogously. The stability of the Ricatti Equation with random coefficients has been studied in [18].

The approaches to the second and third cases introduced above, where the controller is separated from the actuator by a network, depend critically on whether there is a message delivery notification system in place. This divides the problem into two broad cases, the so called TCP and UDP problems [17]. These refer to the transport layer protocol used for message delivery. TCP provides guaranteed message delivery by using delivery acknowledgments and retransmission of dropped packets. On the other hand, UDP is a best effort protocol that provides no acknowledgment of packet drops, but potentially faster delivery. If a TCP-type protocol is used, and a packet containing a control action to the actuator is dropped en-route, then the controller is made aware of the failure. Provided it has knowledge of what control the actuator will implement in the case of the packet drop, the controller can plan future control actions. Thus, even if the control action actually implemented by the actuator is not what was sent in the control packet, it is still *known*. This type of information structure, where the controller has knowledge of the entire history of control actions and observation information is a 'classical' information structure [33]. These problems can be solved using the separation theorem. However, if a UDP protocol is used, then the controller has no idea what control action was *actually* implemented, as it is unaware of whether the packet was delivered or not. This type of problem is an example of a 'non-classical' information structure and such problems have been shown to be intractable [33, 15, 16, 23]. It is for this reason that under the UDP structure, optimal control laws have been determined only for the particular placement of the controller at the actuator; while in general only suboptimal control laws are known [4]. The nature of our results and the assumptions we make do not require the existence of a packet delivery notification system.

3 Problem Formulation

The objective is to control a system described by:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + Gw_k, \\ y_k &= Cx_k + Hv_k, \end{aligned} \tag{3}$$

where x_k represents the system state, u_k the control action, and y_k the measurement at time k. The state and observation noises w_k and v_k are mutually independent with distributions $N(0, \Sigma^w)$ and $N(0, \Sigma^v)$, respectively.

Also, x_0 is $N(\hat{x}_0, \Sigma^{x_0})$ and is independent of $\{w_k, v_k\}$. For simplicity we will assume that (A, B) controllable, (A, C) observable and $\Sigma^w > 0$ and $\Sigma^v > 0$. These assumptions can be relaxed.

The sensor, controller and actuator are separated by a communication network, as shown in Figure 1. An abstraction of such a system is shown in Figure 3. The network randomly drops packets independently and identically with probability that depends on the source, the destination, and the path followed by the packet. We assume that a packet will either arrive at the destination perfectly, or not arrive at all. We do not assume the existence of a message delivery notification system. We assume that transmission time through the network is exactly one time step, and do not allow for random packet delay.

A quadratic performance measure is used in which $F \geq 0$, $Q \geq 0$ and $R > 0$:

$$E\left[x_N' F x_N + \sum_{k=0}^{N-1} \left(x_k' Q x_k + u_k' R u_k\right)\right]. \tag{4}$$

We also consider the infinite horizon average cost per time version of this cost:

$$\text{Min} \limsup_{N\to\infty} \frac{1}{N} E\left[x_N' F x_N + \sum_{k=0}^{N-1} \left(x_k' Q x_k + u_k' R u_k\right)\right]. \tag{5}$$

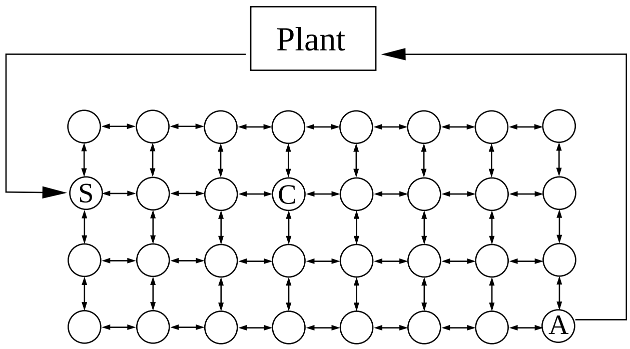

Fig. 3. A networked control system is represented by an array of nodes separated by communication links which randomly drop packets. Node S is the sensor and node A is the actuator. The controller location is represented by node C. If the controller is collocated with the actuator then it will have knowledge of all past actuation commands, but will lack the observations which were dropped in the network.

4 Optimality of Controller Placement on Shortest Path

Since the performance of a controller depends on its location in the network, several questions naturally arise. Firstly, what is the optimal location within a network? Secondly, what is the optimal control given such a position? Thirdly, what is the optimal cost under such a placement and control law? We consider these questions in the following sections.

Consider the network of nodes shown in Figure 3, which represents the network in a networked control system. The objective is to choose where in the array of nodes to locate the control logic, shown as node C. Each link represents an i.i.d. Bernoulli erasure channel with drop probability p. Once a location has been chosen, packets from the sensor node (represented by node S) will be routed along a chosen path to the controller. A control action will be computed by the controller, and forwarded in a packet to the actuator at node, A, along a particular path. Intermediate nodes only forward packets; they do not store them for broadcast later, nor do they do any computation. Packets only move along a predetermined path and are not broadcast and forwarded by all surrounding nodes. We assume that the sensor will only transmit observations over the network, and not, as in [13, 34, 27], some function of the measurements that they compute.

As a packet moves over the network from a node f to a node g it traces a path π_{fg}. The drop probability along such a path is $p^{\pi}_{fg} := p^{|\pi|}$ where $|\pi|$ = length of π. A path between node f and node g with the lowest drop probability is a *shortest path*, and has a drop probability denoted by p^{*}_{fg}. Theorem 2 below establishes that the controller should be placed on a shortest path (which is not necessarily unique) between sensor and actuator, denoted by π^{*}_{SA} and illustrated in Figure 4.

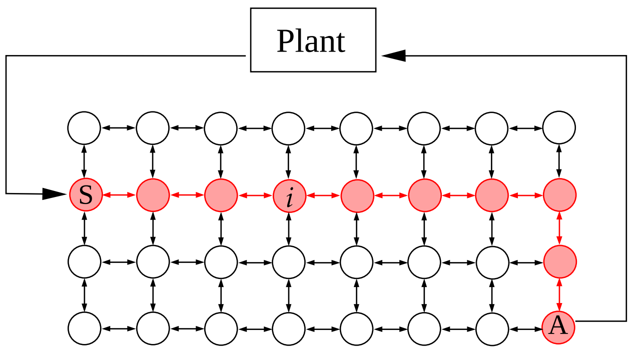

Fig. 4. A minimal path π^{*}_{SA} from S to A is highlighted. It is a path with $p^{|\pi^{*}_{SA}|}$ equal to the smallest total drop probability p^{*}_{SA}. It is not necessarily unique.

We first present a Lemma which proves the information dominance of a shorter path:

Lemma 1. *Given two chains of computational nodes separated by i.i.d. erasure links each with drop probability as shown in Figure 5, the information at a node in the longer chain is stochastically dominated by the information at a corresponding node in the shorter chain.*

Proof. Consider two general sequences of nodes denoted by C_1 and C_2. Assume the length of C_1 is greater than the length of C_2, as shown in Figure 5. Consider nodes A and B such that the packet drop probability en-route to node A in C_1 is identical to that en-route to node B in C_2. The information arrival process at these

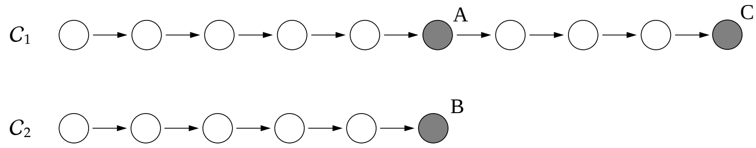

Fig. 5. Two chains of computational nodes separated by i.i.d. erasure channels.

two nodes can be stochastically coupled. Since further packets are lost between nodes A and C in C_1, the information at node B stochastically dominates that at C. Similar dominance arguments can be made that any choice of node C in C_1 is stochastically dominated by a node in an equivalent position in C_2. *Q.E.D.*

Theorem 2. *Given an array of nodes, an optimal controller placement is on* π^*_{SA}.

Proof. This follows directly from Lemma 1. *Q.E.D.*

Note that Theorem 2 holds regardless of the control policy.

4.1 Long Packet Assumption (*LPA*) and Optimal Location under *LPA*

We now consider the placement of the controller on the minimum path. As we discussed in Section 2, in the *absence* of a packet disposition notification system or a specific placement of the controller (e.g., collocated with the actuator), the problem of determining the optimal controller at a node, let alone optimizing the controller over all controller locations, is intractable.

In order to make progress on this problem, we consider a convenient assumption, the 'Long Packets Assumption' (*LPA*) [26, 25, 13, 14]. Under this assumption, packets are allowed to be infinitely long, and each packet can in fact contain the entire history of information from a node. For example, the sensor could send the complete history of *all* observations until that transmission, on every transmission. However, despite the size of the packet we assume it will have the same drop probability, transmission time, and bandwidth usage, as sending a *single* observation. *We stress that this assumption is not intended to be implemented*. Our end goal is analytical: to obtain a lower bound on optimal cost. Since this is an optimistic assumption, the optimal cost obtainable under this information structure will be a lower bound on the optimal cost realizable for the original problem without the *LPA*. It has also been shown that similar performance is achievable with finite sized packets by using an encoder/decoder pair on either side of the channel [13]. In [13] the *LPA* is used as a reference optimal information passing mechanism with the controller collocated with the actuator. In our work here we rigorously establish the optimal controller position, and then investigate the optimal controller and its cost in that location.

The advantage of the LPA is that, under the *LPA* we can precisely determine the optimal location:

Theorem 3. *Under the* LPA, *the optimal placement of an optimized controller is to collocate it with the actuator.*

Proof. We can restrict the network to consisting of just a shortest path π^*_{SA}, and consider the placement of a controller at node $i \in \pi^*_{SA}$, $i \neq A$. Call this Case I, and consider the case where the controller is located at node A as Case II. We will stochastically couple Cases I and II by supposing that events of long packets from i reaching A are identical in both cases. Then, since every long packet contains all past history, whenever node A receives a packet it will contain all past sensor information. Hence, at that time both nodes i and A would have the same set of observation information. However, node A also has the entire history of all implemented control actions, and hence the information at node A dominates that at node i, thus establishing the optimality of Case II under the *LPA*. *Q.E.D*

Note that the proofs thus far have been based solely on information domination arguments and no assumption has been made on the actual control law that is used. In the following section we shall consider what is the optimal control law under *LPA*, and what the optimal cost associated with that law is. Note also, as discussed in the sequel, that when the controller and actuator are collocated, the optimized controller is actually tractable even *without* the *LPA*. However, such a location may itself be not optimal.

5 Optimal Controller for the Finite Horizon Quadratic Cost under the *LPA*

Following the result in Theorem 3 we shall place the controller at the actuator. The packet drop probability is therefore $p_{SA}^{\pi^*_{SA}}$ which we will henceforth denote by λ. We use the variable D_k to indicate the number of *successive* dropped packets at time k, since the last packet reception. We also denote by τ_k^D the event that at time k, the D consecutive previous packets have been dropped.

Denote by y^{k-D_k-1} the set of all observations $\{y_j : j \leq k - D_k - 1\}$, by $u_{k-D_k}^{k-1}$ the set of control actions $\{u_j : k - D_k \leq j \leq k-1\}$, and by u^{k-1} the set of control actions $\{u_j : j \leq k-1\}$. The term $\hat{x}_{k|k-D_k-1}$ will be used to represent the conditional mean estimate of state x at time k based on the noisy observations y^{k-D_k-1} and control actions u^{k-1}. The term $\tilde{x}_{k|k-D_k-1}$denotes the error of that estimate, and $\Sigma^{\tilde{x}}_{k|k-D_k-1}$ its conditional covariance given y^{k-D_k-1} and u^{k-1}, which is actually the conditional covariance given just D_k, due to the Gaussianity and linearity of the system.

Theorem 4. *For the quadratic cost function in (4), under the* LPA, *the optimal control law is given by*

$$u_k^* = -\left[R + B'W_{k+1}B\right]^{-1} B'W_{k+1}A\hat{x}_{k|k-D_k-1}, \tag{6}$$

where W_{k+1} is as given in (20), and $\hat{x}_{k|k-D_k-1}$ is the conditional mean estimate of the state estimate at time k, given noisy observations of states up to time $k - D_k - 1$, contained in the packets received at time $k - D_k$, and that no further observations arrive after that time.

Remark: The result (6) features a slight sharpening of the separation theorem in that only the most recent state estimate is used, and not the full hyperstate [20], which consists of the estimate made at the time of the last packet reception and the list of control actions taken since that last received packet. In a similar vein, we will see that the optimal cost-to-go function also depends only on the most recent state estimate.

Proof. Rewrite the cost (4) as:

$$\begin{aligned} E\left[x_N' F x_N + \sum_{k=0}^{N-1}\left(x_k' Q x_k + u_k' R u_k\right)\right] &= E\Bigg[\left(\hat{x}_{N|N-D_N-1} + \tilde{x}_{N|N-D_N-1}\right)' F \left(\hat{x}_{N|N-D_N-1} + \tilde{x}_{N|N-D_N-1}\right) \\ &\quad + \sum_{k=0}^{N-1}\left(\left(\hat{x}_{k|k-D_k-1} + \tilde{x}_{k|k-D_k-1}\right)' Q \left(\hat{x}_{k|k-D_k-1} + \tilde{x}_{k|k-D_k-1}\right) + u_k' R u_k\right)\Bigg] \\ &= E\left[\hat{x}'_{N|N-D_N-1} F \hat{x}_{N|N-D_N-1} + \sum_{k=0}^{N}\left(\hat{x}'_{k|k-D_k-1} Q \hat{x}_{k|k-D_k-1} + u_k' R u_k\right)\right] \\ &\quad + E\left[Tr\left(F \Sigma^{\tilde{x}}_{N|N-D_N-1}\right) + \sum_{k=0}^{N} Tr\left(Q \Sigma^{\tilde{x}}_{k|k-D_k-1}\right)\right]. \end{aligned} \tag{7}$$

We focus on minimizing the first term above: $E\left[\hat{x}'_{N|N-D_N-1} F \hat{x}_{N|N-D_N-1} + \sum_{k=0}^{N}\left(\hat{x}'_{k|k-D_k-1} Q \hat{x}_{k|k-D_k-1} + u_k' R u_k\right)\right]$, since the remaining term does not depend on control actions. We will add the remaining term at the end though to get the optimal cost. We use dynamic programming to solve for the optimal controller for this first term. For simplicity we will write $\hat{x}_k$ rather than $\hat{x}_{k|k-D_k-1}$ to denote the estimate of the state at time k based on the information available until time k. Similarly we will write $\tilde{x}_k$ for $\tilde{x}_{k|k-D_k-1}$. We will also denote by $\bar{x}_j$ the conditional mean estimate of x_j given y^j and u^{j-1}. We will now prove by backward induction that the cost-to-go function V consists of a quadratic term in the *current* state estimate, and an additional constant f_{k,D_k}:

$$V_k\left(\hat{x}_{k-D_k-1}, u^{k-1}_{k-D_k}, \tau^{D_k}_k\right) = \hat{x}_k' W_k \hat{x}_k + f_{k,D_k} =: \bar{V}_k\left(\hat{x}_k, \tau^{D_k}_k\right). \tag{8}$$

We note that due to the conditional Gaussianity of the linear system, the information state $\left(y^{k-D_k-1}, u^{k-1}, \tau^{D_k}_k\right)$ can be replaced by $\left(\bar{x}_{k-D_k-1}, u^{k-1}_{k-D_k}, \tau^{D_k}_k\right)$, as above. The claim above strengthens this dependence only on $\left(\hat{x}_k, \tau^{D_k}_k\right)$ and also asserts the quadratic structure of the cost-to-go. This is clearly true at the final time step $k = N$, since the expected cost using (4) consists of only the quadratic F term.

Assuming (8) is true for $k+1$, performing a one-step expansion of the dynamic programming equation for k yields,

$$V_k\left(\bar{x}_{k-D-1}, u_{k-D_k}^{k-1}, \tau_k^{D_k}\right) = \min_{u_k}\Big\{ \underbrace{\hat{x}_k' Q \hat{x}_k + u_k' R u_k}_{\text{Cost in current state}} + \underbrace{\lambda \bar{V}_{k+1}(A\hat{x}_k + Bu_k, \tau_{k+1}^{D_k+1})}_{\text{Packet dropped}} + \underbrace{(1-\lambda)E\left[\bar{V}_{k+1}(\hat{x}_{k+1}, \tau_{k+1}^0) \mid \bar{x}_{k-D_k-1}, u_{k-D_k}^{k}, \tau_k^{D_k}, \tau_{k+1}^0\right]}_{\text{Long packet arrives}}\Big\}$$

$$= \min_{u_k}\Big\{\hat{x}_{k|k-D_k-1}' Q \hat{x}_{k|k-D_k-1} + u_k' R u_k + \lambda\left(A\hat{x}_{k|k-D_k-1} + Bu_k\right)' W_{k+1}\left(A\hat{x}_{k|k-D_k-1} + Bu_k\right) + \lambda f_{k+1,D_k+1}$$
$$+(1-\lambda)E\left[\hat{x}_{k+1}' W_{k+1}\hat{x}_{k+1} \mid \bar{x}_{k-D_k-1}, u_{k-D_k}^{k-1}, \tau_k^{D_k}, \tau_{k+1}^0\right] + (1-\lambda) f_{k+1,0}\Big\}. \quad (9)$$

When a packet is dropped, the state estimate is a simple open-loop prediction using the previous state estimate. This is deterministic and hence the expectation has been removed in the above. The proof is now focused on developing the expectation in the second last term of (9). For notational convenience, define the information set $\mathbb{I}_{k-D_k}$ as:

$$\mathbb{I}_{k-D_k} := \left\{\hat{x}_{k-D_k-1|k-D_k-1}, u_{k-D_k}^{k-1}, \tau_k^D, \tau_{k+1}^0\right\} \quad (10)$$

It should be noted that in $\mathbb{I}_{k-D}$ is contained the information that a packet arrives at time $k+1$, but *not* the value of that observation. Nevertheless when such a long packet arrives at $k+1$, the estimation error covariance is set to $\Sigma_{k+1|k}$, since the conditional covariance is the unconditional covariance for the linear Gaussian system. Using also the conditional independence of $\hat{x}_{k+1}$ and $\tilde{x}_{k+1}$, we obtain

$$\begin{aligned} E\left[\hat{x}_{k+1}' W_{k+1}\hat{x}_{k+1} \mid \mathbb{I}_{k-D_k}\right] &= E\left[(x_{k+1} - \tilde{x}_{k+1})' W_{k+1}(x_{k+1} - \tilde{x}_{k+1}) \mid \mathbb{I}_{k-D_k}\right] \\ &= E\left[x_{k+1}' W_{k+1} x_{k+1} \mid \mathbb{I}_{k-D_k}\right] + Tr\left(W_{k+1}\Sigma_{k+1|k}^{\tilde{x}}\right) \\ &\quad - 2E\left[\hat{x}_{k+1}' W_{k+1}\tilde{x}_{k+1} \mid \mathbb{I}_{k-D_k}\right] - 2E\left[\tilde{x}_{k+1}' W_{k+1}\tilde{x}_{k+1} \mid \mathbb{I}_{k-D_k}\right] \\ &= E\left[x_{k+1}' W_{k+1} x_{k+1} \mid \mathbb{I}_{k-D_k}\right] - Tr\left(W_{k+1}\Sigma_{k+1|k}^{\tilde{x}}\right). \end{aligned} \quad (11)$$

Recursively using (3) we obtain

$$x_{k+1} = A^{D_k+2} x_{k-D_k-1} + \sum_{i=0}^{D_k+1}\left(A^i B u_{k-i} + A^i w_{k-i}\right), \quad (12)$$

which yields

$$E\left[x'_{k+1}W_{k+1}x_{k+1} \mid \mathbb{I}_{k-D_k}\right] = E\Bigg[\overbrace{\left(A^{D_k+2}x_{k-D_k-1}\right)' W_{k+1}\left(A^{D_k+2}x_{k-D_k-1}\right)}^{\Gamma_1} + \overbrace{2\left(\sum_{i=0}^{D_k+1} A^i Bu_{k-i} + A^i w_{k-i}\right)' W_{k+1}A^{D_k+2}x_{k-D_k-1}}^{\Gamma_2}$$
$$+ \underbrace{\left(\sum_{i=0}^{D_k+1} A^i Bu_{k-i} + A^i w_{k-i}\right)' W_{k+1}\left(\sum_{i=0}^{D_k+1} A^i Bu_{k-i} + A^i w_{k-i}\right)}_{\Gamma_3} \mid \mathbb{I}_{k-D_k}\Bigg] - Tr\left(W_{k+1}\Sigma^{\tilde{x}}_{k+1|k}\right). \quad (13)$$

The mechanical part of the proof is now to develop each of the terms in the expectation in (13). Taking the first term, and making the substitution $x_{k-D_k} = \tilde{x}_{k-D_k} + \hat{x}_{k-D_k}$ yields:

$$E\left[\Gamma_1 \mid \mathbb{I}_{k-D_k}\right] = \hat{x}'_{k-D_k-1|k-D_k-1}A^{D_k+2\prime}W_{k+1}A^{D_k+2}\hat{x}_{k-D_k-1|k-D_k-1} + Tr\left(A^{D_k+2\prime}W_{k+1}A^{D_k+2}\Sigma^{\tilde{x}}_{k-D_k-1|k-D_k-1}\right). \quad (14)$$

For the second term in (13):

$$E\left[\Gamma_2 \mid \mathbb{I}_{k-D_k}\right] = 2\left(\sum_{i=0}^{D_k+1} A^i Bu_{k-i}\right)' W_{k+1}A^{D_k+2}\hat{x}_{k-D_k-1|k-D_k-1}. \quad (15)$$

Last, expand the third term in the expectation in (13) as

$$E\left[\Gamma_3 \mid \mathbb{I}_{k-D_k}\right] = \left(\sum_{i=0}^{D_k+1} A^i Bu_{k-i}\right)' W_{k+1}\left(\sum_{i=0}^{D_k+1} A^i Bu_{k-i}\right) + \sum_{i=0}^{D_k+1} Tr\left(A^{i\prime}W_{k+1}A^i\Sigma^w\right). \quad (16)$$

Substituting (14), (15) and (16) back into (13) yields a new expression for the right hand side of (9):

$$\min_{u_k}\Big\{\hat{x}'_{k|k-D_k-1}Q\hat{x}_{k|k-D_k-1} + u'_kRu_k + \left(A\hat{x}_{k|k-D_k-1} + Bu_k\right)' W_{k+1}\left(A\hat{x}_{k|k-D_k-1} + Bu_k\right) +$$
$$+(1-\lambda)\Bigg(Tr\left(A^{D_k+2\prime}W_{k+1}A^{D_k+2}\Sigma^{\tilde{x}}_{k-D_k-1|k-D_k-1}\right) + \sum_{i=0}^{D_k+1} Tr\left(A^{i\prime}W_{k+1}A^i\Sigma^w\right)$$
$$-Tr\left(W_{k+1}\Sigma^{\tilde{x}}_{k+1|k}\right)\Bigg) + \lambda f_{k+1,D_k+1} + (1-\lambda)f_{k+1,0}\Big\}. \quad (17)$$

The minimizing control action u^*_k is

$$u^*_k = -\left[R + B'W_{k+1}B\right]^{-1}B'W_{k+1}A\hat{x}_{k|k-D_k-1}. \quad (18)$$

Since the minimum can be written purely as a quadratic function of $\hat{x}_{k|k-D_k-1}$ plus f_{k,D_k} only, without explicitly needing $\hat{x}_{k-D_k-1|k-D_k-1}$, we see that V_k is as shown in (8). *Q.E.D.*

The results for the optimal controller derived in Theorem 4 yield the following expressions for the finite horizon cost.

Theorem 5. *The finite horizon cost is given by:*

$$J_{min}(N) = \hat{x}_0' W_0 \hat{x}_0 + f_{0,0} + E\left[\sum_{k=0}^{N-1} Tr\left(\Sigma^{\tilde{x}}_{k|k-D_k-1} Q\right)\right], \tag{19}$$

where

$$W_k = Q + A'W_{k+1}A - A'W_{k+1}B\left[R + B'W_{k+1}B\right]^{-1} B'W_{k+1}A, \tag{20}$$

and

$$f_{k,D} = \lambda f_{k+1,D+1} + (1-\lambda) f_{k+1,0} + (1-\lambda)\phi_{k,D}, \tag{21}$$

with

$$\phi_{k,D} := Tr\left[A^{D+2\prime} W_{k+1} A^{D+2} \Sigma^{\tilde{x}}_{k-D-1|k-D-1} - A'W_{k+1}A\Sigma^{\tilde{x}}_{k|k} + \sum_{i=1}^{D+1} A^{i\prime} W_{k+1} A^{i} \Sigma^{w}\right]. \tag{22}$$

The associated boundary conditions are $W_N = F$, *and* $f_{N,D} = Tr\left(F\Sigma^{\tilde{x}}_{N|N-D-1}\right)$.

Proof. Substitute (18) into (17) and match terms.

6 The Infinite Horizon Average Cost per Time Problem

We now turn to the infinite horizon average cost per time problem, also considered in [13, 14], and proceed to give a complete proof of the infinite horizon average cost per time problem under *LPA*, attending to the technical issues arising from the fact that the average cost dynamic programming equation does not always admit a solution or solve the problem, as well as an explicit representation of the optimal average cost per time in terms of a power series that allows us to readily deduce the necessary condition for stabilizability. We also provide the behavior at the critical drop probability that is at the boundary of the stabilizable and unstabilizable regions. The infinite horizon average cost per time problem is:

$$\text{Min} \limsup_{N\to\infty} E\left[\frac{1}{N}\sum_{k=0}^{N-1} (x_k' Q x_k + u_k' R u_k)\right].$$

We note that due to assumptions on the system, $\lim_{k\to\infty} \Sigma_{k-D|k-D-n} = \Sigma^{\tilde{x}}_n > 0$ exists for all n, with $\Sigma^{\tilde{x}}_0$ denoting the steady state error covariance of the usual Kalman filter when the observation is always received immediately. Also, W defined by $W = Q + A'WA - A'WB\left[R + B'WB\right]^{-1} B'WA$, is positive definite. This results in the following solution of the infinite horizon average cost per time problem.

Theorem 6. *Under the* LPA*, there exists a policy with finite value of infinite horizon average cost per time and* $\lim_{k\to\infty} \frac{E[x_k' x_k]}{k} = 0$*, if and only if* $\lambda < \frac{1}{\max(eig(A))^2}$*. The optimal average cost per unit time over the class of all policies satisfying* $\lim_k \frac{E[x_k' x_k]}{k} = 0$ *is*

$$J_{min}(\lambda) := \sum_{D=0}^{\infty} \lambda^D (1-\lambda) \left\{ (1-\lambda) Tr\left(A' \left(A^{D+1'} W A^{D+1} - W \right) A \Sigma_0^{\tilde{x}} + \sum_{i=1}^{D+1} A^{i'} W A^i \Sigma^w \right) + Tr\left(\Sigma_{D+1}^{\tilde{x}} Q \right) \right\}. \tag{23}$$

Proof. Note first that since the sumands in (23) involve D only as products of λ^D and quadratics in A^D, it follows that $J_{\min}(\lambda) < \infty$ if and only if $\lambda < \frac{1}{eig(A)^2}$. Moreover, as $\lambda \nearrow \frac{1}{eig(A)^2}$, $J_{min}(\lambda) \to \infty$.

The average cost per time unit can written as:

$$\lim_{N\to\infty} \frac{1}{N} E\left[\sum_{k=0}^{N-1} \left(x_k' Q x_k + u_k' R u_k \right) \right] = \lim_{N\to\infty} \frac{1}{N} E\left[\sum_{k=0}^{N-1} Tr\left(\Sigma_{k|k-D_k-1}^{\tilde{x}} Q \right) + \sum_{k=0}^{N-1} \left(\hat{x}_k' Q \hat{x}_k + u_k' R u_k \right) \right] \tag{24}$$

We focus on the last summation term in (24), which reduces to solving the average cost dynamic programming equation [20]:

$$J^*(\lambda) + V(\hat{x}_k, D_k) = \min_{u_k} E\left[\hat{x}_k' Q \hat{x}_k + u_k' R u_k + V(\hat{x}_{k+1}, D_{k+1}) \right]. \tag{25}$$

Let us consider $\lambda < \frac{1}{eig(A)^2}$. Assume that the form for V is $V(\hat{x}_k, D) = \hat{x}_k' W \hat{x}_k + f(D)$. Note that since this is the infinite horizon case, the constant term $f(D)$ and V depend only on the number of dropped packets and not the time index. Proceeding as in the finite horizon case, (25) can be reduced to:

$$J^*(\lambda) + f(D) = \lambda f(D+1) + (1-\lambda) f(0) + (1-\lambda)\phi_D, \tag{26}$$

where

$$\phi_D := Tr\left[A' \left(A^{D+1'} W A^{D+1} - W \right) A \Sigma_0^{\tilde{x}} + \sum_{i=1}^{D+1} Tr\left(A^{i'} W A^i \Sigma^w \right) \right]. \tag{27}$$

From (26), multiplying by λ^D and summing gives

$$\sum_{D=0}^{M} \lambda^D J^*(\lambda) = \lambda^{M+1} \left[f(M+1) - f(0) \right] + \sum_{D=0}^{M} (1-\lambda) \lambda^D \phi_D. \tag{28}$$

So if $\lim_{M\to\infty} \lambda^{M+1} \left[f(M+1) - f(0) \right] = 0$, then

$$J^*(\lambda) = \sum_{D=0}^{\infty} (1-\lambda)^2 \lambda^D \phi_D \tag{29}$$

Now note that if $f(D)$ is a solution of (26), then so is $f(D)+\alpha$ for all α. So we fix $f(0)=\frac{J^*(\lambda)}{1-\lambda}$ for simplicity. Rewriting (28) with this gives

$$\begin{aligned}\lambda^{M+1} f(M+1) &= \frac{J^*(\lambda)}{1-\lambda} - \sum_{D=0}^{M}(1-\lambda)\lambda^D \phi_D \\ &= (1-\lambda)\sum_{D=M+1}^{\infty}\lambda^D\phi_D. \end{aligned} \tag{30}$$

Hence,

$$\begin{aligned}\sum_{M=0}^{N}\lambda^{M+1} f(M+1) &= (1-\lambda)\sum_{M=0}^{N}\sum_{D=M+1}^{\infty}\lambda^D\phi_D \\ &= (1-\lambda)\sum_{D=1}^{N+1}\sum_{M=0}^{D-1}\lambda^D\phi_D + (1-\lambda)\sum_{D=N+2}^{\infty}\sum_{M=0}^{N}\lambda^D\phi_D \\ &= (1-\lambda)\sum_{D=1}^{N+1}D\lambda^D\phi_D + \sum_{D=N+2}^{\infty}(N+1)\lambda^D\phi_D \end{aligned} \tag{31}$$

Now choose $\lambda < \bar{\lambda} < \frac{1}{eig(A)^2}$. Hence, $J^*(\bar{\lambda}) < +\infty$, and so

$$\sum_{D=1}^{N+1} D\lambda^D\phi_D = \sum_{D=1}^{N+1} D\left(\frac{\lambda}{\bar{\lambda}}\right)^D\left(\bar{\lambda}^D\phi_D\right),$$

which converges by the ratio test. Also

$$(N+1)\sum_{D=N+2}^{\infty}\lambda^D\phi_D \le (N+1)\left(\frac{\lambda}{\bar{\lambda}}\right)^{N+1}\sum_{D=N+1}^{\infty}\bar{\lambda}^D\phi_D \tag{32}$$

converges to 0. Now since the probability that $D_k = D$ is $(1-\lambda)\lambda^D$, we have $E[f(D_k)] \le (1-\lambda)\sum_{M=0}^{\infty}\lambda^M f(M) < +\infty$.

Hence, in the class of policies for which $\lim_{k\to\infty}\frac{E[\hat{x}_k'\hat{x}_k]}{k}=0$, we can deduce as in ([20], page 158) that $J^*(\lambda)$ is the optimal average cost per unit time among all such policies. Hence for $\tilde{\lambda} \ge \frac{1}{eig(A)^2}$ the optimal average cost is greater or equal to $\lim_{\lambda\nearrow\frac{1}{eig(A)^2}} J_{min}(\lambda)=\infty$.

Now we note that the average cost per unit time is monotone non-decreasing in λ, since the control policy can simply discard more packets to emulate a larger drop probability.

For $J^*(\lambda)$, (26) is the cost associated with the last summation term of (24). The contribution to the cost from the first term is given by calculating the probability that the time interval between two packets is D, which is $\lambda^D(1-\lambda)$. Hence,

$$\lim_{N\to\infty} E\left[\frac{1}{N}\sum_{k=0}^{N-1} Tr\left(\Sigma^{\tilde{x}}_{k|k-D_k-1}Q\right)\right] = \sum_{D=0}^{\infty}\lambda^D(1-\lambda)Tr\left(\Sigma^{\tilde{x}}_{D+1}Q\right).$$

Combining them gives (23). This completes the proof. *Q.E.D*

7 An Upper Bound on Cost without *LPA*

The optimistic *LPA* assumption leads to (23) being a *lower bound* on the achievable optimal cost for the location optimal controller without the *LPA*, when it is finite. The question then is how tight is this lower bound?

We can upper bound the achievable optimal cost by the cost of any implementable scheme. In particular, we consider the scheme using a Kalman filter for state estimation, and collocate the controller with the actuator [13]. In this placement, the separation theorem with classical information structure holds, and the Kalman filter will simply perform open-loop state estimation in the case of packet drops. The LQG certainty-equivalent controller is optimal for that collocated position.

It turns out that these upper and lower bounds are in fact fairly close in many cases, as will be seen in Section 9.

8 Necessary Conditions for Stability

We now consider the case where the LPA does not hold. We can use the results under the *LPA* to derive a necessary condition for the stabilizability of the system without the *LPA*, allowing optimization over all controller locations and admissible control laws. Here, by stability we mean finiteness of the infinite horizon average quadratic cost per time, as well as $\lim_{k\to\infty}\frac{E[x_k' x_k]}{k} = 0$. In fact it will be seen to yield a necessary and sufficient condition in Corollary 2 in certain special systems.

Corollary 1. *If (A, B) is controllable, (A, C) is observable, $\Sigma^w > 0$ and $\Sigma^v > 0$, then the condition*

$$\lambda < \frac{1}{\overline{\mid eig(A)\mid}^2}, \tag{33}$$

is necessary and sufficient for the existence of a stabilizing controller at any location, in the presence of packet drops.

Proof. This follows immediately fr Theorem 6. *Q.E.D*

For certain special systems the following corollary shows that we can deduce the sharp packet drop probability threshold and region for stabilizability of networked control systems with packet drops:

Corollary 2. *If C^{-1} exists then the condition:*

$$\lambda < \frac{1}{|\overline{eig(A)}|^2}$$

is necessary and sufficient for the existence of a stabilizing controller in the presence of packet drops.

Proof. Follows directly from Theorem 1 and Corollary 1. *Q.E.D*

9 Simulation Results

To investigate the utility of the results obtained in the previous sections we performed several simulations to study the tightness of the bounds.

Consider the candidate system with

$$A = \begin{bmatrix} 1.7000 & 0.0000 & 0.0000 & 0.0000 \\ 0.6211 & 0.6029 & 0.3028 & 0.6965 \\ 0.3200 & 0.7210 & 0.3685 & 0.1146 \\ 0.2272 & 0.7875 & 0.0138 & 0.2518 \end{bmatrix}, \quad (34)$$

which has eigenvalues 1.7000, 1.3544, 0.2925 and −0.4237. Identity matrices were used for all other system matrices as well as noise covariances. The resulting cost for various drop probabilities is shown in Figure 6. The lower and upper bounds are seen to be fairly tight.

Figure 7 shows several simulation results and compares the percentage deviation of the transport layer protocol described in [27] from the lower bound derived in Theorem 6. The upper bound is also shown. The closeness of the two bounds again demonstrates the utility of the two bounds.

Table 1 shows a summary of results obtained from simulations of various examples of systems taken from the literature. It should be noted however, that for cases where the necessary condition is not in fact sufficient, the gap may be unbounded near the critical drop probability. The systems were chosen from some of the existing literature on networked control systems to facilitate comparison to the results in those works. However, not all models were fully specified. Some did not include model noise covariance or observation matrices. In these cases we used identity matrices. In most cases this represents a large measurement and model covariance matrix. Hence, when more realistic values are used (such as those given in system [32]) the bounds are closer.

Finally, Figure 8 gives some indication as to why the upper and lower bounds are tight. When a packet drop occurs, both the lower and upper bound error covariances diverge from the steady state value. However, when a packet arrives, the system with the *LPA* immediately jumps back to the original error covariance. What is of importance is that within only a couple of time steps the error covariance for the upper bound system has also converged.

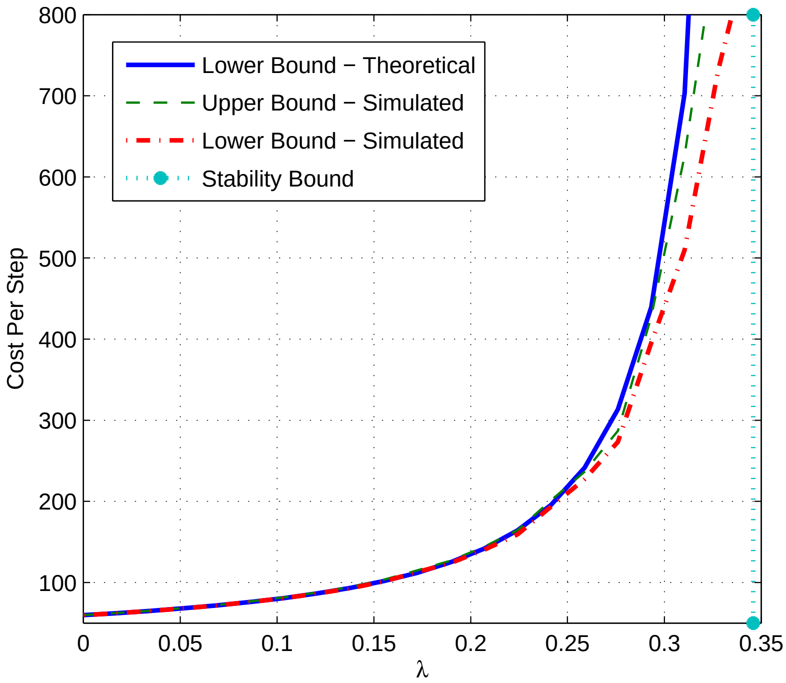

Fig. 6. Plot of performance bounds for various λ. The deviation at higher values of λ is a consequence of insufficient simulation steps. The system used is described in (34) with identity noise covariances. Hence, the stability bound is limited by the largest eigenvalue of A and is computed as 0.346.

10 Conclusion

In networked control systems, location of the controller is also a "control variable" and it too can be optimized to mitigate the effect of packet drops in wireless networks. Moreover, it can also be migrated in

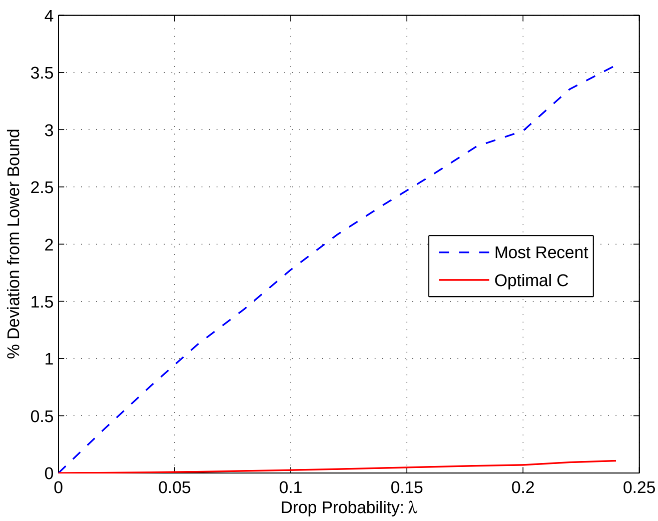

Fig. 7. Simulation results showing the percentage deviation in cost from the lower bound achievable using the *LPA*. λ is the packet loss probability. 'Optimal C' refers to the cost obtained by using the optimal ratio in [27].

Table 1. Summary of performance simulation results for various systems.

System	λ_{max}	Limit	0 Drops		25% λ_{max}		50% λ_{max}		75% λ_{max}	
			Cost	%	Cost	%	Cost	%	Cost	%
System in (34)	0.275	UB - Simulated	53.27		65.57		88.69		145.25	
		LB - Simulated	53.27	0	65.01	0.8	87.24	1.6	140.65	3.1
		LB - Theoretical	53.29		65.05		88.13		154.20	
Batch Reactor [32]	0.671	UB - Simulated	58.96		65.88		79.20		117.37	
		LB - Simulated	58.96	0	65.35	0.8	77.87	1.6	114.20	2.7
		LB - Theoretical	58.90		65.30		77.89		114.20	
General [4]	0.602	UB - Simulated	274		379		654		1805	
		LB - Simulated	274	0	375	1	640	2	1756	2.7
		LB - Theoretical	274		375		642		1790	
General 3x3 [29]	0.64	UB - Simulated	412.6	0	573.1		1018		3304	
		LB - Simulated	412.6		564.9	1.46	975.4	4.37	3022	9.34
		LB - Theoretical	412.4		564.4		974		2931	
General Scalar [29]	0.64	UB - Simulated	8.91		10.50		13.57		22.54	
		LB - Simulated	8.91	0	10.17	3.1	12.69	6.4	20.26	10.12
		LB - Theoretical	8.91		10.19		12.71		20.12	
Vehicle Speed [10] (Stable)	1	UB - Simulated	59305		59352		59443		59722	
		LB - Simulated	59305	0	59340	0.02	59414	0.05	59658	0.11
		LB - Theoretical	60156		60192		60267		60514	
DC Motor [19] (Stable)	1	UB - Simulated	256.21		256.55		257.22		259.22	
		LB - Simulated	256.21	0	256.55	0.2	257.22	0.3	259.22	0.6
		LB - Theoretical	253.51		253.84		254.50		256.45	

real time as network conditions change or nodes fail [5, 7, 9]. Hence it is important to address the probelm of location of controller logic. We have studied the problem of where to locate a controller in a wireless network with packet losses.

Determining the optimal location is however a difficult problem as a result of the 'non-classical' information structures that result. This makes the general problem intractable.

We are able to make progress by considering the artifice of a 'Long Packet Assumption,' (*LPA*), under which we can solve the optimal location problem as well as the optimal controller and cost for that location. We show that, under the *LPA*, the optimal controller *placement* in a networked control system is collocated with the actuator and that the optimal controller for that location is obtained from a slightly strengthened separation theorem. The former result is based purely on information dominance arguments. The resulting cost, when finite, is a lower bound on optimal cost for systems in which the *LPA* does not generally hold.

The utility of this lower bound on optimal cost over all locations and all controllers is that it permits a comparison with known realizable upper bounds or other controller placements in the network. In many example problems we have seen that existing schemes are nearly optimal, though the optimization problem is intractable. The cost expression also yields a necessary condition for stabilizability over all controller locations which complements existing sufficiency results. In some special systems the necessary condition coincides with the known sufficient condition. When they do not coincide, we should note that the gap

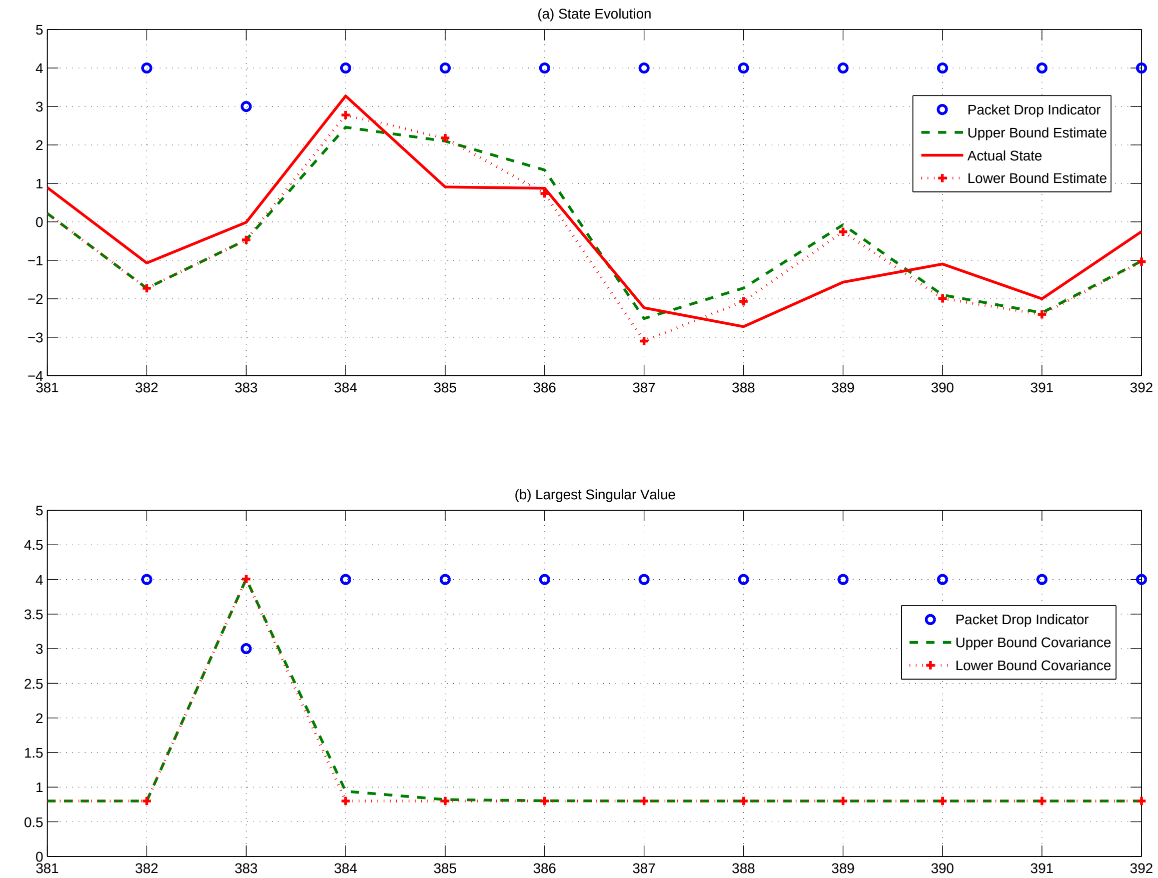

Fig. 8. Figure (a) is a trace of the actual state and its estimate using systems with the upper and lower bounds. When a packet is dropped, the indicator value is 3. Figure (b) plots the largest singular value of the state error covariance matrices for the two bounds.

between lower and upper bounds will diverge for drop probabilities beyond the critical drop probability for which the system is stabilizable if that is less that that given by the necessary condition.

References

1. *Special edition on Networks and Control*, volume 21. IEEE Control Systems Magazine, Feb 2001.
2. *Special Issue: Technology of Networked Control Systems*. Proceedings of the IEEE, Jan 2007.
3. M. Adès, P. E. Caines, and R. P. Malhamé. Stochastic optimal control under poisson-distributed observations. *IEEE Transactions on Automatic Control*, 45(1), January 2000.
4. B. Azimi-Sadjadi. Stability of networked control systems in the presence of packet losses. In *Proc. of the IEEE Conf. on Decision and Control.*, 2003.
5. G. Baliga. *A Middleware Framework for Networked Control Systems*. PhD thesis, University of Illinois at Urbana-Champaign, 2005.
6. G. Baliga, S. Graham, and P. R. Kumar. Middleware and abstractions in the convergence of control with communication and computation. In *Proc. of the IEEE Conf. on Decision and Control.*, pages 4245–4250, 2005.
7. G. Baliga, S. Graham, L. Sha, and P. R. Kumar. Service continuity in networked control using etherware. *IEEE Distributed Systems Online*, 5(9), 2004.
8. G. Baliga, S. Graham, Lui Sha, and P. R. Kumar. Etherware: Domainware for wireless control networks. In *Proceedings of The 7th IEEE International Symposium on Object-oriented Real-time Distributed Computing*, May 2004.
9. G. Baliga and P. R. Kumar. A middleware for control over networks. *Proc. of the IEEE Conf. on Decision and Control.*, 2005.
10. H. Chan and U. Ozguner. Closed-loop control of systems over a communication network with updates. In *Proc. of the American Controls Conf.*, 1994.
11. M. Chow and Y. Tipsuwan. Network-based control systems: A tutorial. In *The* 27^{th} *Annual Conference of the IEEE Industrial Electronics Society.*, pages 1593–1602, 2001.

12. S. Graham, G. Baliga, and P. R. Kumar. Issues in the convergence of control with communication and computing: proliferation, architecture, design, services, and middleware. In *Proc. of the IEEE Conf. on Decision and Control.*, pages 1466–1471, 2004.
13. V. Gupta, B. Hassibi, and R. M. Murray. Optimal LQG control across packet-dropping links. *Systems & Control Letters*, 56(6):439–446, June 2007.
14. V. Gupta, D. Spanos, B. Hassibi, and R. R. Murray. On lqg control across a stochastic packet-dropping link. In *Proc. of the American Controls Conf.*, June 2005.
15. Y. C. Ho and K. C. Chu. Team decision theory and information structures in optimal control problems - Part 1. *IEEE Transactions on Automatic Control*, 1972.
16. Y. C. Ho and K. C. Chu. Team decision theory and information structures in optimal control problems - Part 2. *IEEE Transactions on Automatic Control*, 1972.
17. O. C. Imer, S. Yuksel, and T. Basar. Optimal control of dynamical systems over unreliable communication links. *Automatica*, June 2006.
18. T. Katayama. On the matrix riccati equation for linear systems with a random gain. *IEEE Trans. on Automatic Control*, 21(2):770–771, October 1976.
19. A. Khan, N. Agarwal, D. Tilbury, and J. Moyne. The impact of random device processing delays on networked control system performance. In 42^{nd} *Annual Allerton Conference on Communication, Control, and Computing.*, 2004.
20. P. R. Kumar and P. Varaiya. *Stochastic Systems: Estimation, Identification and Adaptive Control*, volume 1. Englewood Cliffs, 1986.
21. X. Liu and A. Goldsmith. Kalman filtering with partial observation losses. In *Proc. of the IEEE Conf. on Decision and Control.*, pages 4180–4186, Atlantis, Bahamas, December 2004.
22. J. Nilsson, B. Bernhardsson, and B. Wittenmark. Stochastic analysis and control of real-time systems with random time delays. *Automatica*, 34(1):57–64, 1998.
23. C. H. Papadimitriou and J. Tsitsiklis. Intractable problems in control theory. *SIAM Journal on Control and Optimization*, 24(4):639–654, July 1986.
24. I. B. Rhodes and D. L. Snyder. Estimation and control performance for space-time point-process observations. In *IEEE Transactions on Automatic Control*, volume AC-22, pages 338–346, June 1977.
25. C. L. Robinson and P. R. Kumar. Control over networks of unreliable links: Location of controllers, control laws and bounds on performance. Technical report, University of Illinois at Urbana-Champaign, 2006.
26. C. L. Robinson and P. R. Kumar. Control over networks of unreliable links: Location of controllers, control laws and bounds on performance. In *Proceedings of Control over Communication Channels (ConCom)*, Cyprus, April 2007.
27. C. L. Robinson and P. R. Kumar. Sending the most recent observation is not optimal in networked control: Linear temporal coding and towards the design of a control specific transport layer. In *Proc. of the IEEE Conf. on Decision and Control.*, 2007.
28. P. Seiler and R. Sengupta. Analysis of communication losses in vehicle control problems. In *Proc. of the American Controls Conf.*, June 2001.
29. B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry. Kalman filtering with intermittent observations. In *IEEE Transactions on Automatic Control*, volume 49, 2004.
30. B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, and S. S. Sastry. Optimal control with unreliable communication: the TCP case. In *Proc. of the American Controls Conf.*, 2005.
31. D. L. Snyder and P. M. Fishman. How to track a swarm of fireflies by observing their flashes. In *IEEE Transactions on Information Theory*, 1975.
32. G. C. Walsh, H. Ye, and L. G. Bushnell. Stability analysis of networked control systems. *IEEE Transactions on control systems technology*, 10(3):438–446, May 2002.
33. H. S. Witsenhausen. A counter example in stochastic optimal control. *Siam J. Control*, 6:131–147, 1968.
34. Y. Xu and J. P. Hespanha. Optimal communication logics in networked control systems. In *Proc. of the IEEE Conf. on Decision and Control.*, December 2004.
35. Y. Xu and J. P. Hespanha. Estimation under uncontrolled and controlled communications in networked control systems. In *Proc. of the IEEE Conf. on Decision and Control.*, pages 842–847, 2005.