2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Reducing Packet Loss Bursts in a Wireless Mesh Network for Stochastic Bounds on Estimation Error

Phoebus Chen, Chithrupa Ramesh, and Karl H. Johansson

***Abstract*— A big challenge for wireless networked control systems is how to design the underlying networking algorithms and protocols to provide high reliability, defined as the end-to-end probability of packet delivery, despite the high packet loss rates of individual wireless links. This paper formulates the problem of jointly designing a set of packet forwarding policies on a multipath mesh network to meet control application requirements.**

We derive several results to help understand the problem space. First, we demonstrate that some common approaches, like applying a single forwarding policy to all packets or always routing packets on disjoint paths, are not optimal for the application when the links are bursty. Second, we introduce the notion of *dominance* to give a partial ordering to sets of forwarding policies, used to prove that an optimal policy schedules all outgoing links at each node and that an upper bound on the performance attained by unicast forwarding policies on the network graph can be computed assuming a flooding policy. Third, we demonstrate how to convert application performance metrics to packet forwarding policy objectives, using the probability that the error covariance of a Kalman filter stays within a bound as our application metric. Fourth, we provide an algorithm to compute the joint probability mass function that a sequence of packets are delivered, given a set of policies and a network graph. Finally, we describe how to obtain optimal policies via an exhaustive search, motivating future research for more computationally efficient solutions.

I. Introduction

Increasingly, control systems are operated over large-scale, networked infrastructures. In fact, several companies today are introducing devices that communicate over low-power wireless mesh networks for industrial automation and process control to save wiring costs [1], [2]. Unfortunately, wireless communication is inherently unreliable, introducing packet losses and delays, which are detrimental to control system performance and stability.

This work focuses on how to co-design the network with the control system to reduce the probability of getting bad packet loss patterns for the control application, e.g., a particular number of consecutive packet losses or a particular periodic pattern of packet losses. Our main contribution is the motivation and problem formulation for how to design a set of unicast hop-by-hop multipath packet forwarding policies on a mesh network to get high *reliability*, defined as the probability of end-to-end packet delivery, while avoiding bursts of correlated packet losses.

Our problem setup is motivated by the difficulty of getting high reliability when the packet deliveries have latency constraints and links experience periods of outage (bursty links). For context, flooding on a mesh network can fully utilize all paths in the network for reliability, but is unacceptable for many applications because it results in lower aggregate network throughput, longer delays, and more energy consumption. Without latency constraints, unicast hop-by-hop routing with link acknowledgments and retransmissions has also been demonstrated to use path diversity in real deployments to achieve high reliability [3].

A. Related Work

The optimal packet forwarding policy for reliably delivering a single packet before a deadline was studied in [4]. Many works on multipath routing place a strong emphasis on finding edge-disjoint and node-disjoint paths to serve as alternate end-to-end routes during link outages [5], [6]. However, sending packets on a collection of single paths is less reliable than hop-by-hop multipath routing with link acknowledgments and retransmissions on alternate links, where packets can route around link outages.

Several theoretical results on networked control systems characterize network conditions that guarantee some measure of application performance [7], [8], [9]. Most similar to our work is [9], where Epstein et al. relate the probability that the Kalman filter error covariance exceeds a bound to the network reliability when the deliveries are i.i.d. . We also adopt their strategy of sending multiple past consecutive measurements in one packet to mitigate the effects of packet loss in Section IV.

II. Problem Formulation

A. Plant and Network Models

We consider the problem of observing the state x_k of the plant $\mathcal{P}$ from across the mesh network, as shown in Figure 1. Let the system sampling times be indexed by k. The plant has linear dynamics given by

$$x_{k+1} = Ax_k + w_k \quad , \quad y_k = Cx_k + v_k \ ,$$

where $A \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{m\times n}$ are the system matrices. The process noise w_k and the measurement noise v_k are i.i.d. zero mean Gaussian with covariance matrices $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$, respectively. The initial state $x_0 \in \mathbb{R}^n$ is a known constant.

The work was supported by the EU project FeedNetBack, the Swedish Research Council, the Swedish Strategic Research Foundation, the Swedish Governmental Agency for Innovation Systems, and the Knut and Alice Wallenberg Foundation.

The authors are with the ACCESS Linnaeus Center, KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden. {phoebus,cramesh,kallej} @ ee.kth.se

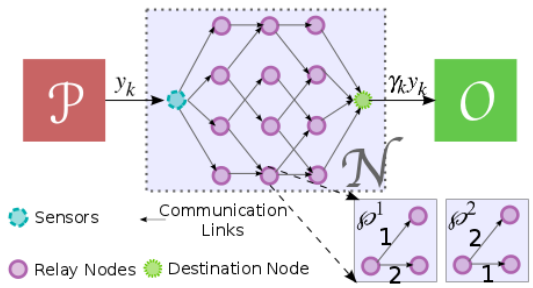

Fig. 1. Open loop estimation of plant $\mathcal{P}$ with the mesh network $\mathcal{N}$ on the sensing channel. The Kalman filter $\mathcal{O}$ performs the estimation. Two packet forwarding policies $\wp^1$ and $\wp^2$ for one relay node are depicted, where the links are labeled with the order of transmission attempts.

The measurements y_k reach the Kalman filter $\mathcal{O}$ over a lossy network. The Kalman filter uses the recursive equations

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q \, , \tag{1}$$
$$P_{k|k} = P_{k|k-1} - \gamma_k P_{k|k-1}C^T(CP_{k|k-1}C^T + R)^{-1}CP_{k|k-1} \, ,$$

to update the error covariance of the MMSE estimate [8], where the binary random variable $\gamma_k \in \{0, 1\}$ takes the value 1 when a measurement packet is delivered, and 0 otherwise. We assume that the system (A, C) is observable, the initial state x_0 is known to the estimator, and $P_0 = 0$. The estimation error covariance $P_{k|k}$, which we write as P_k in shorthand, will be treated as a cost function. The study of a 2-channel networked control system (lossy measurement and actuation channel) will be left for future work.

The packet delivery random variable γ_k is obtained from the model of the wireless mesh network ($\mathcal{N}$ in Figure 1), which consists of a static link model, a routing topology G, and a set of packet forwarding policies $\{\wp^\kappa\}_{\kappa=1}^K$, all of which will be described below. We assume the packet delivery latency constraint is not too stringent, such that all packet forwarding policies can deliver the packets before the next system sample time unless the packets arrive at nodes where all outgoing links fail.

The static link model assumes that each link l in the network is either up (packets can be transmitted over that link) or down (outage), and does not switch between these two states. Although in reality links do switch states, we use this simple model of a bursty link to focus on the problem of correlated packet losses from link outages. When designing the packet forwarding policies, we only know the a priori probability p_l that link l is up, not the state of the link. The links are mutually independent, and each link has an acknowledgement indicating whether a transmission was successful.

The routing topology is described by $G = (\mathcal{V}, \mathcal{E}, p)$, a weighted, connected, Destination-Oriented Directed Acyclic Graph (DODAG), which is a DAG where only the destination node has no outgoing edges. The set of vertices (nodes) is $\mathcal{V} = \{1, \ldots, V\}$. The set of directed edges (links) is $\mathcal{E} \subseteq \{(i, j) \ : \ i, j \in \mathcal{V}, \ i \neq j\}$, where the number of edges is denoted E. There are 2^E possible topology realizations $\tilde{G} = (\mathcal{V}, \tilde{\mathcal{E}})$, where $\tilde{\mathcal{E}} \subseteq \mathcal{E}$ represents the links that are up. The edge weights are given by the function $p : \mathcal{E} \mapsto [0, 1]$, where $p(l)$ (equivalently, p_l or p_{ij} when $l = (i, j)$) is the a priori probability link l is up. The source node is denoted a and the sink (destination) node is denoted b.

Each packet forwarding policy $\wp^\kappa$ in the set of policies $\{\wp^\kappa\}_{\kappa=1}^K$ defines how a unicast packet is routed hop-by-hop through the routing topology. The policy $\wp^\kappa = \{\wp_i^\kappa\}_{i=1}^V$ contains the forwarding policy at each node i. Node i's forwarding policy $\wp_i^\kappa = (l_1, \ldots, l_{\delta^+(i)})$ is an ordered set of node i's outgoing links, where $\delta^+(i)$ is the outdegree of node i. When a packet arrives at node i, node i will try transmission on l_1, then l_2 if transmission on l_1 failed, and so forth, dropping the packet when all links have been tried and failed. Thus, the forwarding policy $\wp^\kappa$ is deterministic and time invariant. In comparison, a cyclical TDMA schedule on a mesh network, like that considered in [3], behaves like a packet forwarding policy where the order of link transmissions at a node depends on the time the packet arrives at the node. Unless stated otherwise in the examples later, we assume that a packet sent at system sample time k will select the packet forwarding policy $\wp^{(k \mod K)+1}$ from the set of K packet forwarding policies $\{\wp^\kappa\}_{\kappa=1}^K$.

B. Forwarding Policy Optimization Problem

Our goal is to design the set of packet forwarding policies $\{\wp^\kappa\}_{\kappa=1}^K$, given the plant $\mathcal{P}$ and the routing topology G, for the problem

$$\max_{\{\wp^\kappa\}_{\kappa=1}^K} \Pr(\forall k \in \mathbb{N}, P_k < M) \quad , \tag{2}$$

where the error covariance bound M is a positive semidefinite matrix that is a given design parameter. The number of policies K is not a given design parameter, but should come from the solution to the problem.

The policies $\{\wp^\kappa\}_{\kappa=1}^K$ are related to the error covariances P_k through packet delivery sequences. A packet delivery sequence of length H, denoted $\boldsymbol{g}_1^H \in \{0, 1\}^H$, is a realization of the random vector $\boldsymbol{\gamma}_1^H$ that is used in the equations (1) to compute a sequence of error covariances $P_1, \ldots, P_H$. The computation of the joint probability mass function (pmf) of $\boldsymbol{\gamma}_1^H$, $p_{\boldsymbol{\gamma}^H} : \{0, 1\}^H \mapsto [0, 1]$, from $\{\wp^\kappa\}_{\kappa=1}^K$ and G will be discussed in Section V. Thus, $\{\wp^\kappa\}_{\kappa=1}^K$ and G induce a joint pmf on sequences of error covariances, so (2) expresses the probability of getting a sequence of error covariances that violates the bound.

Note that our objective (2) is stated in terms of an infinite sequence of error covariances, although with static links and the "resetting" estimator setup considered in Section IV, we only need to check whether a length K sequence of error covariances violate the bound. This is because for a given set of policies $\{\wp^\kappa\}_{\kappa=1}^K$, the packet delivery sequence will repeat (recall that we select policy $\wp^{(k \mod K)+1}$ at time k).

III. Properties of Unicast Forwarding Policies

This section provides several examples to motivate why one cannot just consider maximizing reliability or minimizing packet correlation, but must consider the packet delivery joint pmf when designing the packet forwarding policies.

It also establishes several lemmas characterizing the joint packet forwarding policy problem.

A. Examples of Reliability and (Un)correlation Tradeoffs

For our examples below, let the application requirement translate to maximizing the $1/K$ *joint reliability* of a set of K packets, which is defined as the probability that at least one of the packets is delivered.

Remark Designing each packet forwarding policy unilaterally to improve single packet reliability may worsen $1/K$ joint reliability.

Example 3.1 (Max individual reliability is not optimal): Consider the routing topology G depicted in Figure 2a, where $p_{1b} > p_{2b}$. If we maximize the $1/2$ joint reliability of two packets, they will use the packet forwarding policies $\wp^1$ and $\wp^2$ that differ only in

$$\wp_a^1 = \big((a,1),(a,2)\big) \quad , \quad \wp_a^2 = \big((a,2),(a,1)\big)$$

(the policies at the other nodes are trivial, since those nodes have only one link). If we tried to maximize the individual reliability of packets 1 and 2 separately, packet 2 would also use policy $\wp^1$ and be completely correlated with packet 1. Using $\wp^1$ for both packets is worse in the event that link $(1,b)$ is down and all the other links are up, because $\wp^1$ will never send a packet down the path $(a,2,b)$ but $\wp^2$ will. ∎

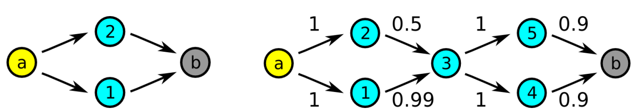

Fig. 2. (a) G for Example 3.1. (b) G for Example 3.2, where links are labeled with their success probabilities.

Remark Packet forwarding policies that minimize the probability that packets try the same paths in order to reduce packet delivery correlation may not be optimal for $1/K$ joint reliability.

Example 3.2 (Minimizing correlation is not optimal): Consider the routing topology G depicted in Figure 2b, where the links are labeled with their success probabilities. We want to maximize the $1/2$ joint reliability of two packets. Assume that the policy for the first packet has been chosen to be $\wp^1$, where

$$\begin{Bmatrix} \wp_a^1 = \big((a,1),(a,2)\big) \\ \wp_3^1 = \big((3,4),(3,5)\big) \end{Bmatrix} \quad .$$

Let use choose between two policies $\wp^2$ and $\wp^3$ for the second packet, where

$$\begin{Bmatrix} \wp_a^2 = \big((a,2),(a,1)\big) \\ \wp_3^2 = \big((3,5),(3,4)\big) \end{Bmatrix} \quad , \quad \begin{Bmatrix} \wp_a^3 = \big((a,1),(a,2)\big) \\ \wp_3^3 = \big((3,5),(3,4)\big) \end{Bmatrix} \quad .$$

If we wish for the packets to take edge-disjoint paths to minimize correlation, we would choose $\wp^2$ for the second packet. However, choosing $\wp^3$ for the second packet maximizes the $1/2$ joint reliability because

$$\Pr(\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2 = 0) = (\bar{p}_{13} + p_{13}\bar{p}_{4b})(\bar{p}_{23} + p_{23}\bar{p}_{5b}) = 0.06$$
$$\Pr(\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_3 = 0) = \bar{p}_{13} + p_{13}\bar{p}_{4b}\bar{p}_{5b} = 0.0199 \quad ,$$

where the packet delivery random variable $\boldsymbol{\gamma}$ is indexed by the policy, and $\bar{p}_l = (1 - p_l)$. We see that when p_{23} is very low, the second packet should take the more reliable path $(a,1,3)$ and then "utilize the path diversity" in path $(3,5,b)$. ∎

B. Partial Ordering of Forwarding Policies

For all applications where it is better to receive a packet than not receive the packet, there is a partial ordering of the packet delivery joint pmf's that allows us to compare certain packet forwarding policies to determine which is better for application performance. To simplify the following exposition, we only consider packet sequences of length K.

Definition 3.1 (Dominant Packet Delivery Sequence): Let $\boldsymbol{g}_1'^K, \boldsymbol{g}_1^K \in \{0,1\}^K$ be two packet delivery sequences. The sequence $\boldsymbol{g}_1'^K$ *dominates* the sequence $\boldsymbol{g}_1^K$, written as $\boldsymbol{g}_1'^K \succeq \boldsymbol{g}_1^K$, if $\boldsymbol{g}_\kappa' \geq \boldsymbol{g}_\kappa, \forall \kappa = 1, \ldots, K$.

The following lemma states that given two packet delivery sequences, the dominant sequence will result in equal or better performance for our Kalman filtering application performance metric (2).

Lemma 3.1: Let $P_k(\boldsymbol{g}_1'^K)$ and $P_k(\boldsymbol{g}_1^K)$ be the estimation error covariance at time $k \in \{1, \ldots, K\}$ from (1) induced by packet delivery sequences $\boldsymbol{g}_1'^K$ and $\boldsymbol{g}_1^K$, respectively (given the same initial condition and noise realizations). If $\boldsymbol{g}_1'^K \succeq \boldsymbol{g}_1^K$, then $P_k(\boldsymbol{g}_1'^K) - P_k(\boldsymbol{g}_1^K)$ is positive semidefinite for all k, written as $P_k(\boldsymbol{g}_1'^K) \succeq P_k(\boldsymbol{g}_1^K), \forall k$.

We will not prove this lemma, but it can be seen from the time-varying Kalman filter equations (1), which subtracts a positive semidefinite matrix from the error covariance matrix whenever the estimator receives a measurement packet.

Definition 3.2 (Dominant Packet Delivery Joint PMF): Let $p'_{\boldsymbol{\gamma}^K}, p_{\boldsymbol{\gamma}^K}$ be two joint pmf's of K packet deliveries. The pmf $p'_{\boldsymbol{\gamma}^K}$ *dominates* the pmf $p_{\boldsymbol{\gamma}^K}$, written as $p'_{\boldsymbol{\gamma}^K} \geq p_{\boldsymbol{\gamma}^K}$, if for any packet delivery sequence realization $\boldsymbol{g}_1^K \in \{0,1\}^K$,

$$\sum_{\boldsymbol{g}_1'^K \succeq \boldsymbol{g}_1^K} p'_{\boldsymbol{\gamma}^K}(\boldsymbol{g}_1'^K) \geq \sum_{\boldsymbol{g}_1'^K \succeq \boldsymbol{g}_1^K} p_{\boldsymbol{\gamma}^K}(\boldsymbol{g}_1'^K) \quad .$$

Definition 3.3 (Dominant Packet Forwarding Policies): Let the sets of packet forwarding policies $\{\wp'^\kappa\}_{\kappa=1}^K$ and $\{\wp^\kappa\}_{\kappa=1}^K$ induce the packet delivery pmf's $p'_{\boldsymbol{\gamma}^K}$ and $p_{\boldsymbol{\gamma}^K}$, respectively, given a weighted DODAG $G = (\mathcal{V}, \mathcal{E}, p)$ with static links. The set of policies $\{\wp'^\kappa\}_{\kappa=1}^K$ *dominates* the set of policies $\{\wp^\kappa\}_{\kappa=1}^K$, written $\{\wp'^\kappa\}_{\kappa=1}^K \geq \{\wp^\kappa\}_{\kappa=1}^K$, if $p'_{\boldsymbol{\gamma}^K} \geq p_{\boldsymbol{\gamma}^K}$.

We state the following without proof.

Property 3.1 (Dominance is Transitive): Dominance of packet delivery sequences, joint pmf's, and packet forwarding policies as defined in Definitions 3.1, 3.2, and 3.3, are transitive properties (e.g., $\{\wp'^\kappa\}_{\kappa=1}^K \geq \{\wp^\kappa\}_{\kappa=1}^K$ and $\{\wp''^\kappa\}_{\kappa=1}^K \geq \{\wp'^\kappa\}_{\kappa=1}^K$ implies $\{\wp''^\kappa\}_{\kappa=1}^K \geq \{\wp^\kappa\}_{\kappa=1}^K$).

C. Characterizing the Optimal Policy Search Space

Lemma 3.2: Given a set of packet forwarding policies $\{\wp^\kappa\}_{\kappa=1}^K$ where there exists a node i and packet k such that $\wp_i^k$ does not contain all the outgoing links of i, there exists a set of policies $\{\bar{\wp}^\kappa\}_{\kappa=1}^K$ where for all nodes j and packets

κ, $\bar{\wp}^{\kappa}_{j}$ contains all the outgoing links of j, and $\{\bar{\wp}^{\kappa}\}_{\kappa=1}^{K}$ dominates $\{\wp^{\kappa}\}_{\kappa=1}^{K}$.

Proof: Proof by construction. Select a node i and packet κ such that $\wp^{\kappa}_{i}$ does not contain all the outgoing links of i. Construct a new set of policies $\{\wp'^{\kappa}\}_{\kappa=1}^{K}$ from $\{\wp^{\kappa}\}_{\kappa=1}^{K}$ by replacing $\wp^{\kappa}_{i}$ with $\wp'^{\kappa}_{i}$, where $\wp'^{\kappa}_{i}$ consists of appending the remaining outgoing links of i (in any order) to the ordered list of links $\wp^{\kappa}_{i}$. Under the assumption that failures only occur when a packet cannot leave a node, using $\wp'^{\kappa}_{i}$ instead of $\wp^{\kappa}_{i}$ can only move probability mass from a packet delivery sequence $\boldsymbol{g}_1^K$ where $\boldsymbol{g}_\kappa = 0$ to a dominant sequence $\boldsymbol{g}'^K_1 \geq \boldsymbol{g}_1^K$ where $\boldsymbol{g}'_\kappa \in \{0,1\}$.Therefore, the pmf $p'_{\boldsymbol{\gamma}^K}$ induced by $\{\wp'^{\kappa}\}_{\kappa=1}^{K}$ dominates the pmf $p_{\boldsymbol{\gamma}^K}$ induced by $\{\wp^{\kappa}\}_{\kappa=1}^{K}$, so $\{\wp'^{\kappa}\}_{\kappa=1}^{K}$ dominates $\{\wp^{\kappa}\}_{\kappa=1}^{K}$. Following the same procedure, we select another node i and packet κ such that $\wp'^{\kappa}_{i}$ does not contain all the outgoing links of i, and construct a new set of policies $\{\wp''^{\kappa}\}_{\kappa=1}^{K}$ from $\{\wp'^{\kappa}\}_{\kappa=1}^{K}$. We can show that $\{\wp''^{\kappa}\}_{\kappa=1}^{K} \geq \{\wp'^{\kappa}\}_{\kappa=1}^{K}$ so by Property 3.1 $\{\wp''^{\kappa}\}_{\kappa=1}^{K} \geq \{\wp^{\kappa}\}_{\kappa=1}^{K}$. Repeat this process until we have constructed a set of policies $\{\bar{\wp}^{\kappa}\}_{\kappa=1}^{K}$ where for all i and κ, $\bar{\wp}^{\kappa}_{i}$ contains all the outgoing links of i. Through the application of Property 3.1 at each step, we have $\{\bar{\wp}^{\kappa}\}_{\kappa=1}^{K} \geq \{\wp^{\kappa}\}_{\kappa=1}^{K}$. ■

Lemmas 3.2 and 3.1 motivate why we do not need to consider packet forwarding policies that do not schedule all outgoing links on a node when finding the optimal policy. This reinforces the point made in Example 3.2 that routing on disjoint paths is not optimal.

D. Unicast Hop-by-hop Routing vs. Flooding

Let $\wp^{\star\kappa}$ represent the packet forwarding policy for flooding along the DAG, where $\forall i$, $\wp^{\star\kappa}_{i} = \{(i,j) \in \mathcal{E}\}$ is an unordered set, since node i will transmit on all outgoing links with a broadcast. Despite this difference, a set of policies $\{\wp^{\star\kappa}\}_{\kappa=1}^{K}$ which floods each packet (same policy for all packets) also induces a packet delivery pmf, so the notion of dominance in Definition 3.3 can still be applied.

Lemma 3.3: Given a weighted DODAG $G = (\mathcal{V}, \mathcal{E}, p)$ with static links, the set of K flooding policies $\{\wp^{\star\kappa}\}_{\kappa=1}^{K}$ on G will dominate any set of K unicast packet forwarding policies $\{\wp^{\kappa}\}_{\kappa=1}^{K}$ on G.

Proof: All of the probability mass in the joint pmf $p^{\star}_{\boldsymbol{\gamma}^K}$ induced by $\{\wp^{\star\kappa}\}_{\kappa=1}^{K}$ are assigned to the events associated with the delivery sequences $\mathbf{1}_1^K$ and $\mathbf{0}_1^K$. Thus, for $\{\wp^{\star\kappa}\}_{\kappa=1}^{K} \geq \{\wp^{\kappa}\}_{\kappa=1}^{K}$ we need to show

$$p^{\star}_{\boldsymbol{\gamma}^K}(\mathbf{1}_1^K) \geq \sum_{\boldsymbol{g}_1^K \neq \mathbf{0}_1^K} p_{\boldsymbol{\gamma}^K}(\boldsymbol{g}_1^K) \quad , \tag{3}$$

where $p_{\boldsymbol{\gamma}^K}$ is the joint pmf induced by $\{\wp^{\kappa}\}_{\kappa=1}^{K}$. Since flooding tries all paths in the DODAG, $p^{\star}_{\boldsymbol{\gamma}^K}(\mathbf{1}_1^K)$ is the probability of getting any graph realization where a path exists between a and b. Also, $\sum_{\boldsymbol{g}_1^K \neq \mathbf{0}_1^K} p_{\boldsymbol{\gamma}^K}(\boldsymbol{g}_1^K)$ is the probability of getting graph realizations where the policies $\{\wp^{\kappa}\}_{\kappa=1}^{K}$ would yield a packet delivery sequence $\boldsymbol{g}_1^K \neq \mathbf{0}_1^K$. Inequality (3) holds because the set of graph realizations where a path exists between a and b is a superset of the set of graph realizations yielding $\boldsymbol{g}_1^K \neq \mathbf{0}_1^K$. ■

Lemma 3.3 motivates us to use the joint pmf from flooding as a reference to measure the suboptimality of a set of unicast packet forwarding policies.

The joint pmf from flooding is simply

$$p^{\star}_{\boldsymbol{\gamma}^K}(\boldsymbol{g}_1^K) = \begin{cases} p_{a\to b} & : \quad \boldsymbol{g}_1^K = \mathbf{1}_1^K \\ 1 - p_{a\to b} & : \quad \boldsymbol{g}_1^K = \mathbf{0}_1^K \\ 0 & : \quad \text{otherwise} \end{cases} ,$$

where $p_{a\to b}$, the probability of a packet reaching the sink with flooding, can be computed from the FAST_FPP Algorithm in [10].

IV. Translating Application Performance to Network Metrics

This section derives network performance objectives from the application metric in (2). We demonstrate that for two broad classes of systems, our objective of minimizing the probability that the Kalman filter's estimation error covariance exceeds a bound can be conservatively translated to a r/K joint reliability network metric, which is the probability that at least r out of K packets are delivered. Section VI will demonstrate how the r/K joint reliability metric is used to find the optimal set of packet forwarding policies.

Class 1 (Full State Information Systems): For the class of systems where we get full state information ($R = 0$, C is invertible) from each measurement packet, the application metric (2) translates to a $1/K$ joint reliability metric. In this case, the estimation error covariance of the Kalman filter is reset to zero upon every packet delivery. Consider one such reset instance, at time k, such that $P_k = 0$. From (1), we can express the probability of exceeding the bound at a future time $k+h$ as $\Pr(P_{k+h} > M) = \mathbb{I}(\sum_{\ell=1}^{h} A^{\ell-1}Q(A^T)^{\ell-1} > M) \cdot \Pr(\beta_{k+h} = h)$, where $\mathbb{I}(x)$ is an indicator function which takes the value 1 when the condition x is true and 0 otherwise, and β_{k+h} is the packet loss burst length at time $k+h$. Since Q is positive definite, the expression $\sum_{\ell=1}^{h} A^{\ell-1}Q(A^T)^{\ell-1}$ increases monotonically with the burst length h, and exceeds the bound for some $h = K$. Thus, maximizing the metric in (2) translates to maximizing the $1/K$ joint reliability network metric. ■

The $1/K$ joint reliability metric does not extend easily to systems with partial state information (C not invertible, $R \neq 0$), because a number of consecutive measurement vectors are required to reach a "reset" instance. Also, this "reset" does not drive the estimation error to zero, but to $\bar{P} = \lim_{k\to\infty} P_k$, the value of the estimation error covariance when *all* the packets are delivered. We can try to obtain a "reset" in a lossy network using r/K joint reliability if we pack multiple past measurement vectors in a packet, like in [9]. This is explained below.

Class 2 (Partial State Information Systems): To obtain consecutive measurement vectors at the Kalman filter, we can pack the previous $K - r + 1$ measurement vectors $\{y_k\}$ in each packet, where $r > 0$. With each packet delivery, the Riccati updates corresponding to previous lost packets can be reiterated with the relevant measurements, and the estimation error covariance can be "reset" to $\bar{P}$. With this scheme, $K-r+1$ packets out of K packets must be lost to lose any of the previous K measurement vectors.

Now, the longest burst corresponds to $K-r$ non-arrivals in any window of K packets, which increases the estimation error covariance from $\bar{P}$ to $P_{\max} = A^{K-r}\bar{P}(A^T)^{K-r} + \sum_{\ell=1}^{K-r} A^{\ell-1}Q(A^T)^{\ell-1}$. Receiving r out of K packets will meet the application requirement if $P_{\max} < M$ and $K-r+1$ measurement vectors fit in the packet. We say such a pair (r, K) is feasible, and the metric (2) translates to maximizing the r/K joint reliability across all feasible pairs (r, K). ■

The classification of systems given above is very coarse because it does not distinguish between systems whose (A, C) matrix pair have different structures (e.g., repeated eigenvalues, eigenspaces observable with one measurement vector). Finding a more refined system classification for translating application to network metrics is a promising direction for future work. Such a system classification may need to consider other bad packet loss patterns besides long packet loss bursts if we cannot pack multiple past measurement vectors in one packet, as shown in the following example.

Example 4.1 (Rotating System): Consider the rotating system $A = \left[\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right]$, $C = \left[\begin{smallmatrix} 1 & 0 \end{smallmatrix}\right]$, $Q = \left[\begin{smallmatrix} 0.1 & 0 \\ 0 & 0.1 \end{smallmatrix}\right]$ and $R = 0$. If we receive every other measurement (even or odd), the system is not observable. This can be shown by computing the observability matrices for the matrix pair (A^2, C) and the matrix pair (A^2, CA), both of which are not full rank. Assume we only have 2 packet forwarding policies $\{\wp^1, \wp^2\}$, like for G in Example 3.1. In a network with static links where only policy $\wp^1$ or policy $\wp^2$ succeeds, the system will not be observable if we select the policy $\wp^{(k \mod K)+1}$ at time k, but will be observable if we select the policy $\wp^{(\lfloor k/2 \rfloor \mod K)+1}$ (i.e., apply the $1/2$ joint reliability metric on *blocks* of packets). Selecting policy $\wp^{(k \mod K)+1}$ at time k minimizes the probability of packet loss bursts but is not better for estimation. ■

V. Joint Packet Delivery PMF

To calculate the joint pmf $p_{\boldsymbol{\gamma}^K}$ from the policies $\{\wp^\kappa\}_{\kappa=1}^K$ and routing topology G, we can simply simulate the packet delivery of the K packets for each graph realization $\tilde{G}$ (by iterating through $2^{\mathcal{E}}$, the set of all subsets of $\mathcal{E}$) and tabulate the probability of each packet delivery sequence. This is computationally expensive, taking $O((D\Delta^+K + E)2^E)$, where D is the length of the longest path in the network and Δ^+ is the maximum outdegree of the nodes in G. This order of growth expression comes from the $O(D\Delta^+)$ operations to simulate one packet (in the worst case trying Δ^+ links on each hop), the K packets to simulate per graph realization, and the E operations to compute the probability of each graph realization.

If there are a small number of packet policies K and a large number of links E, then Algorithm 1 is faster. The algorithm traverses through cut sets of the DODAG and calculates the joint pmf's of packet arrival at each cut set, in the spirit of the FAST_FPP Algorithm in [10]. Lines 14-21 construct the transition probability matrix between cut sets and performs matrix multiplication.

Algorithm 1 TI_UPD_JPMF2

Input: $G = (\mathcal{V}, \mathcal{E}, p), a, b, \{\wp^\kappa\}_{\kappa=1}^K$
Output: $p_{\boldsymbol{\gamma}^K} : \{0,1\}^K \mapsto [0,1]$
$\mathcal{C} := \{a, 0\}$ ▷ $\mathcal{C}$ is the vertex cut and 0 (failure).
$\mathcal{V}' := \mathcal{V} \backslash a$ ▷ $\mathcal{V}'$ is the set of remaining vertices.
5: $\mathcal{E}' := \mathcal{E}$ ▷ $\mathcal{E}'$ is the set of remaining edges.
$p_{\mathcal{C}^K}(\boldsymbol{c}_1^K) := 0,\ \forall \boldsymbol{c}_i^K \in \{0, a\}^K$
$p_{\mathcal{C}^K}(a \cdot \mathbf{1}_1^K) := 1$ ▷ pmf for vertex cut $\mathcal{C}$.
while $\mathcal{V}' \neq \emptyset$ **do**
  [Find nodes $\mathcal{U}$ to remove from vertex cut $\mathcal{C}$]
10: $\mathcal{U} := \{j \in \mathcal{C} \backslash 0 \,:\, \forall i, (i,j) \notin \mathcal{E}'\}$
  $\mathcal{D} := \mathcal{C} \backslash \mathcal{U} \cup \{j \in \mathcal{V}' \,:\, \exists i \in \mathcal{U}, (i,j) \in \mathcal{E}'\}$
  $p'_{\mathcal{D}^K}(\boldsymbol{d}_1^K) := 0, \forall \boldsymbol{d}_1^K \in \mathcal{D}^K$ ▷ next vertex cut's pmf.
  [Transition to $\mathcal{D}^K$, downstream vertex cut]
  for all $\boldsymbol{c}_1^K \in \mathcal{C}^K$ **do**
15: **for all** $\boldsymbol{d}_1^K \in \mathcal{D}^K$ **do**
      $\mathcal{L}_\kappa := L(\boldsymbol{c}_\kappa, \boldsymbol{d}_\kappa, \wp^\kappa_{\boldsymbol{c}_\kappa}, \mathcal{U}), \forall \kappa = 1, \ldots, K$
      $\mathcal{L} := \bigcup_\kappa \mathcal{L}_\kappa$
      $T_{\boldsymbol{c}_1^K \boldsymbol{d}_1^K} := \text{VALID}(\mathcal{L}) \cdot \text{TOPROB}(\mathcal{L})$
      $p'_{\mathcal{D}^K}(\boldsymbol{d}_1^K) := p'_{\mathcal{D}^K}(\boldsymbol{d}_1^K) + p_{\mathcal{C}^K}(\boldsymbol{c}_1^K) \cdot T_{\boldsymbol{c}_1^K \boldsymbol{d}_1^K}$
20: **end for**
  end for
  $\mathcal{E}' := \mathcal{E}' \backslash \{(i,j) \in \mathcal{E}' \,:\, i \in \mathcal{U}\}$
  $\mathcal{V}' := \mathcal{V}' \backslash \mathcal{U}$
  $\mathcal{C} := \mathcal{D}$; $p_{\mathcal{C}^K} := p'_{\mathcal{D}^K}$
25: **end while**
$p_{\boldsymbol{\gamma}^K}(\boldsymbol{g}_1^K) := p_{\mathcal{C}^K}(b \cdot \boldsymbol{g}_1^K),\ \forall \boldsymbol{g}_1^K \in \{0,1\}^K$
Return: $p_{\boldsymbol{\gamma}^K}$

The algorithm calls the functions L, VALID, and TOPROB to construct the transition probabilities $T_{\boldsymbol{c}_1^K \boldsymbol{d}_1^K}$ between the states of the cut sets. Function L outputs the set of events for a transition from $\boldsymbol{c}_1^K$ to $\boldsymbol{d}_1^K$ in lines 16-17, and line 18 computes the probability of this transition, if it is valid. The functions are defined as:

$$L(i, j, \wp_i^\kappa, \mathcal{U}) = \begin{cases} \mathcal{L}^* & : \ i \in \mathcal{U} \wedge (i,j) \in \wp_i^\kappa \\ \mathcal{L}^{**} & : \ i \in \mathcal{U} \wedge j = 0 \\ \{\bot\} & : \ \begin{array}{l} (i \in \mathcal{U} \wedge \neg((i,j) \in \wp_i^\kappa \vee j = 0)) \ \vee \\ (i \notin \mathcal{U} \wedge i \neq j) \end{array} \\ \{\top\} & : \ (i \notin \mathcal{U} \wedge i = j) \end{cases}$$

where

$$\mathcal{L}^* = \{\neg(i,v) \,:\, (i,v) \text{ precedes } (i,j) \text{ in } \wp_i^\kappa\} \cup \{(i,j)\}$$
$$\mathcal{L}^{**} = \{\neg(i,v) \,:\, (i,v) \in \wp_i^\kappa\} \quad .$$

$\mathcal{L}^*$ are the events (link up or down, with $\neg$ meaning down) for the packet to be transmitted from i to j, and $\mathcal{L}^{**}$ are the events for the packet to be dropped. The impossible event $\{\bot\}$ is output when a packet cannot be sent from i to j and the possible event $\{\top\}$ is output when node i holds onto the packet because it is in both cut sets.

$$\text{VALID}(\mathcal{L}) = \begin{cases} 0 & : \ \exists (i,j) \text{ s.t. both } (i,j), \neg(i,j) \in \mathcal{L} \\ 1 & : \ \text{otherwise.} \end{cases}$$

TOPROB($\mathcal{L}$) returns the product of the probabilities associated with each element in $\mathcal{L}$, where each element is converted to a probability following the rules:

$$(i,j) \longrightarrow p_{ij} \quad , \quad \top \longrightarrow 1 \text{ (possible event)}$$
$$\neg(i,j) \longrightarrow 1 - p_{ij} \quad , \quad \bot \longrightarrow 0 \text{ (impossible event)} \quad .$$

Property 5.1: The running time of Algorithm 1 is $O(S\Delta^{+}K(\hat{\mathcal{C}}+1)^{2K})$, where S (greater than D) is the number of cut sets $\mathcal{C}$ traversed in the algorithm and $\hat{\mathcal{C}}$ is the number of nodes in the largest cut set encountered in the algorithm.

Proof: [Derivation of Property 5.1] Lines 16-18 take $O(K\Delta^{+})$ to compute $T_{\boldsymbol{c}_1^K \boldsymbol{d}_1^K}$. Therefore, lines 14-21 take $O(K\Delta^{+}|\mathcal{C}|^K|\mathcal{D}|^K)$. These lines dominate the computation time of the while loop, which has S iterations. The longest running time of an iteration of the while loop is $O(K\Delta^{+}(\hat{\mathcal{C}}+1)^{2K})$ because $|\mathcal{C}|, |\mathcal{D}| \leq \hat{\mathcal{C}}+1$ (we do not include the failure "node" 0 in the count of $\hat{\mathcal{C}}$). ∎

In general, to compute the application performance measure (2) (for a finite horizon H) from the joint pmf $p_{\boldsymbol{\gamma}^K}$, we iterate through all $\boldsymbol{g}_1^K \in \{0,1\}^K$. For each $\boldsymbol{g}_1^K$ we stack the vectors $\boldsymbol{g}_1^K$ until we get a packet delivery sequence of some desired length H (truncate the extra elements), compute the sequence of error covariances $P_1, \ldots, P_H$, check whether any P_k violate the bound, and accumulate the probability $p_{\boldsymbol{\gamma}^K}(\boldsymbol{g}_1^K)$ if not. This would take $O(q^3 H 2^K)$ operations, if solving (1) at each step k takes $O(q^3)$ operations, where $q = \max(m,n)$.

VI. Optimal Packet Forwarding Policy Problem

To describe the problem of finding the optimal set of packet forwarding policies, we express the steps of Algorithm 1 using matrix notation. Let the cut sets include the 0 element for failures, and number them from $0, \ldots, S$. Write the joint pmf $p_{\mathcal{C}^K}$ on the cut set $\mathcal{C}_{(s)}$ as a row vector $\boldsymbol{\pi}^{(s)} \in [0,1]^{|\mathcal{C}_{(s)}|^K}$, where for convenience we use the vectors $\boldsymbol{c}_1^K \in \mathcal{C}_{(s)}^K$ instead of natural numbers as indices. The initial pmf $\boldsymbol{\pi}^{(0)} \in [0,1]^{2^K}$ at node a is equal to 1 at index $\boldsymbol{c}_1^K = [a \ \ldots \ a]$ and 0 at all other indices. The packet delivery joint pmf $p_{\boldsymbol{\gamma}^K}$ is $\boldsymbol{\pi}^{(S)} \in [0,1]^{2^K}$.

Let $T_{\mathcal{C}^K\mathcal{D}^K} \in [0,1]^{|\mathcal{C}|^K \times |\mathcal{D}|^K}$ be the transition probability matrix whose entries are computed in lines 16-18 of Algorithm 1. Each $T_{\mathcal{C}^K\mathcal{D}^K}$ is a function of $\{\wp_i^\kappa\}_{\kappa=1, i\in\mathcal{C}}^K$, as described in lines 16-18 in Algorithm 1. If we write $T^{(s)}$ as a shorthand for $T_{\mathcal{C}_{(s-1)}^K \mathcal{C}_{(s)}^K}$, the problem of finding the optimal set of policies can be written as

$$\arg\max_{\{\wp^\kappa\}_{\kappa=1}^K} \boldsymbol{\pi}^{(0)} T^{(1)} \cdots T^{(S)} \boldsymbol{w} \tag{4}$$

where $\boldsymbol{w} \in [0,1]^{2^K}$ is a weighting vector for assigning importance to joint packet delivery events, indexed by the vectors $\boldsymbol{g}_1^K \in \{0,1\}^K$. For instance, to describe the r/K packets network metric,

$$\boldsymbol{w}_{\boldsymbol{g}_1^K} = \begin{cases} 0 & : \quad \sum_{i=1}^K \boldsymbol{g}_i < r \\ 1 & : \quad \sum_{i=1}^K \boldsymbol{g}_i \geq r \end{cases} \quad .$$

The optimal $T^{\star(s)}$ is a function of both the optimal upstream (closer to source) joint pmf $\boldsymbol{\pi}^{\star(s)} = \boldsymbol{\pi}^{(0)} T^{\star(1)} \cdots T^{\star(s)}$ and the optimal downstream (closer to destination) weighted reward vector $\boldsymbol{w}^{\star(s+1)} = T^{\star(s+2)} \cdots T^{\star(S)} \boldsymbol{w}$. Thus, we cannot apply dynamic programming backwards starting from S. To get the optimal solution, one could perform an exhaustive search on all combinations of packet forwarding policies to maximize the objective function in (4).

We defer proposing approximate/heuristic algorithms for future work, when we have better criteria for evaluations and comparisons.

VII. Conclusions and Future Work

This paper demonstrated the need to consider application performance metrics when designing unicast hop-by-hop packet forwarding policies for a mesh network, and points out many promising directions for future research. One is to characterize how our Kalman filtering performance metric translates to different network performance objectives for a classification of plants based on the (A, C) matrix pair that is more discriminating than the classification given in Section IV. Another is to develop and compare lower computational complexity algorithms for finding a good set of forwarding policies, preferably with bounds on the suboptimality of the solution. Using suboptimal forwarding policies in practice may be acceptable because real estimates of link probabilities may be noisy, making the optimal policy very fragile.

References

[1] Wireless Industrial Networking Alliance, "WINA website," http://www.wina.org, 2010.

[2] International Society of Automation, "ISA-SP100 wireless systems for automation website," http://www.isa.org/isa100, 2010.

[3] K. S. J. Pister and L. Doherty, "TSMP: Time synchronized mesh protocol," in *Proc. of the IASTED International Symposium on Distributed Sensor Networks (DSN)*, Nov. 2008.

[4] Z. Zou, P. Soldati, H. Zhang, and M. Johansson, "Delay-constrained maximum reliability routing over lossy links," in *Proc. of the 49th IEEE Conference on Decision and Control*, Dec. 2010.

[5] P. Papadimitratos, Z. J. Haas, and E. G. Sirer, "Path set selection in mobile ad hoc networks," in *Proc. of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*. New York, NY, USA: ACM, 2002, pp. 1–11.

[6] M. K. Marina and S. R. Das, "Ad hoc on-demand multipath distance vector routing," *Wireless Communications and Mobile Computing*, vol. 6, no. 7, pp. 969–988, 2006.

[7] G. Walsh, H. Ye, and L. Bushnell, "Stability analysis of networked control systems," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 3, pp. 438–446, May 2002.

[8] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, and S. Sastry, "Kalman filtering with intermittent observations," *IEEE Transactions on Automatic Control*, September 2004.

[9] M. Epstein, L. Shi, A. Tiwari, and R. M. Murray, "Probabilistic performance of state estimation across a lossy network," *Automatica*, vol. 44, no. 12, pp. 3046–3053, 2008.

[10] P. Chen, K. H. Johansson, P. Balister, B. Bollobás, and S. Sastry, "Multi-path routing metrics for reliable wireless mesh routing topologies," KTH Royal Institute of Technology, Tech. Rep. TRITA-EE 2011:033, March 2011. [Online]. Available: http://arxiv.org/abs/1103.5957