

Network Estimation and Packet Delivery Prediction for Control over Wireless Mesh Networks^{*}

Phoebus Chen Chithrupa Ramesh Karl H. Johansson

*ACCESS Linnaeus Center
KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden
(email: {phoebus, cramesh, kalleg} @ ee.kth.se)*

Abstract:

Much of the current theory of networked control systems uses simple point-to-point communication models as an abstraction of the underlying network. As a result, the controller has very limited information on the network conditions and performs suboptimally. This work models the underlying wireless multihop mesh network as a graph of links with transmission success probabilities, and uses a recursive Bayesian estimator to provide packet delivery predictions to the controller. The predictions are a joint probability distribution on future packet delivery sequences, and thus capture correlations between successive packet deliveries. We look at finite horizon LQG control over a lossy actuation channel and a perfect sensing channel, both without delay, to study how the controller can compensate for predicted network outages.

Keywords: Control and estimation with data loss; Networked embedded control systems; Stochastic control

1. 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 (Wireless Industrial Networking Alliance, 2010; International Society of Automation, 2010). While wireless mesh networks can connect control processes that are physically spread out over a large space to save wiring costs, these networks are difficult to design, provision, and manage (Bruno et al., 2005). Furthermore, wireless communication is inherently unreliable, introducing packet losses and delays, which are detrimental to control system performance and stability.

Several works in Networked Control Systems (NCSs) (Hespanha et al., 2007) study stability and controller synthesis for control systems using different network models to provide statistics about packet losses and delays. Schenato et al. (2007) and Ishii (2008) model the network as an i.i.d. Bernoulli process, Seiler and Sengupta (2005) describe the network using a Gilbert-Elliott model, and Elia (2005) uses a stochastic LTI system to model fading. Olfati-Saber et al. (2007) and other works on consensus of multi-agent systems simply model the multihop network as a connectivity graph. The multihop network model in Gupta et al. (2009) is most similar to the network model used in our work.

^{*} 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.

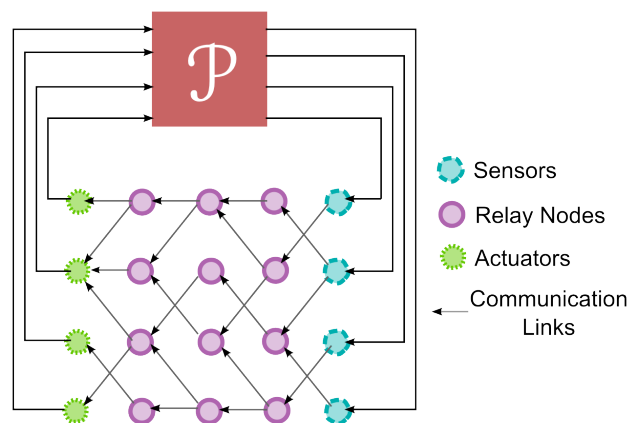


Fig. 1. A networked control system over a mesh network, where the controllers can be located on any node.

Most of these research studies suffer from two key limitations. The first limitation is the use of simple models of packet delivery over a point-to-point link or a star network topology to represent the network, which are often multihop and more complex. For instance, a real communication protocol that may be used in a NCS is TSMP (Pister and Doherty, 2008), which uses a TDMA schedule to route packets over a directed acyclic graph routing topology. The second limitation is the treatment of the network as something designed and fixed a priori before the design of the control system. Very little information is passed through the interface between the network and the control system, limiting the interaction between the two “layers” to tune the controller to the network conditions, and vice versa.

To the best of our knowledge, our work is the first to use a reasonably sophisticated multihop network model to monitor the network conditions and dynamically tune the controller to compensate for predicted packet losses. In this work, we consider a special case of the general system architecture of a NCS over a mesh network proposed by Robinson and Kumar (2007), depicted in Fig. 1. We place the controller at the sensor and assume lossless delivery of acknowledgments on the actuation channel, so the optimal LQG controller and estimator can be designed separately (Schenato et al., 2007).

Similar to Gupta et al. (2009), we model the network routing topology as a graph of independent links, where transmission success on each link is described by a two-state Markov chain. A key difference is that in addition to the routing topology, our network model also includes a global TDMA transmission schedule. Such a minimalist network model captures the essence of how a network with bursty links can have correlated packet deliveries (Willig et al., 2002), which are particularly bad for control when they result in bursts of packet losses.

Using this model, we propose a network estimator to estimate, without loss of information, the state of the network given the past packet deliveries.¹ The network state estimate is translated to a joint probability distribution predicting the success of future packet deliveries, which is passed through the network-controller interface so the controller can compensate for unavoidable network outages. The network estimator can also be used to notify a network manager when the network is broken and needs to be reconfigured or reprovisioned, a direction for future research.

Section 2 describes our plant and network models. We propose the Gilbert-Elliott Independent links, Hop-by-hop routing, Scheduled (GEIHS) network estimator in Section 3. Next, we design a finite-horizon, Future-Packet-Delivery-optimized (FPD) LQG controller to utilize the packet delivery predictions provided by the network estimators, presented in Section 4. Section 5 provides an example and simulations demonstrating how the GEIHS network estimator combined with the FPD controller can provide better performance than a classical LQG controller or a controller assuming i.i.d. packet deliveries. Finally, Section 6 describes the limitations of our approach and future work.

2. PROBLEM FORMULATION

This paper studies an instance of the general system architecture depicted in Fig. 1, with a single control loop containing one sensor and one actuator. One network estimator and one controller are placed at the sensor, and we assume that an end-to-end acknowledgement (ACK) that the controller-to-actuator packet is delivered is always received at the network estimator, as shown in Fig. 2. For simplicity, we assume that the plant dynamics are significantly slower than the end-to-end packet delivery deadline, so that we can ignore the delay introduced by the network. The general problem is to jointly design a

¹ Strictly speaking, we obtain the probability distribution on the states of the network, not a single point estimate.

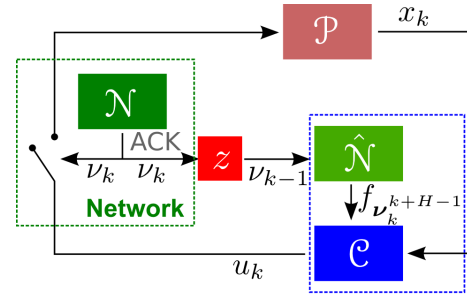


Fig. 2. A control loop for plant \mathcal{P} with the network on the actuation channel. The network estimator $\hat{\mathcal{N}}$ passes packet delivery predictions $f_{\nu_k^{k+H-1}}$ to the FPD controller \mathcal{C} , with past packet delivery information obtained from the network \mathcal{N} over an acknowledgement (ACK) channel.

network estimator and controller that can optimally control the plant using our proposed GEIHS network model. In our problem setup, the controller is only concerned with the past, present, and future packet delivery sequence and not with the detailed behavior of the network, nor can it affect the behavior of the network. Therefore, the network estimation problem decouples from the control problem. The information passed through the network-controller interface is the packet delivery sequence, specifically the joint probability distribution describing the future packet delivery predictions.

2.1 Plant and Network Models

The state dynamics of the plant \mathcal{P} in Fig. 2 is given by

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

where $A \in \mathbb{R}^{\ell \times \ell}$, $B \in \mathbb{R}^{\ell \times m}$, and w_k are i.i.d. zero-mean Gaussian random variables with covariance matrix $R_w \in \mathbb{S}_+^\ell$, where \mathbb{S}_+^ℓ is the set of $\ell \times \ell$ positive semidefinite matrices. The initial state x_0 is a zero-mean Gaussian random variable with covariance matrix $R_0 \in \mathbb{S}_+^\ell$ and is mutually independent of w_k . The binary random variable ν_k indicates whether a packet from the controller reaches the actuator ($\nu_k = 1$) or not ($\nu_k = 0$), and each ν_k is independent of x_0 and w_k (but the ν_k 's are not independent of each other).

Let the discrete sampling times for the control system be indexed by k , but let the discrete time for schedule time slots (described below) be indexed by t . The time slot intervals are smaller than the sampling intervals. The time slot when the control packet at sample time k is generated is denoted t_k , and the deadline for receiving the control packet at the receiver is t'_k . We assume that $t'_k \leq t_{k+1}$ for all k .

The model of the TDMA wireless mesh network (\mathcal{N} in Fig. 2) consists of a routing topology G , a link model describing how the transmission success of a link evolves over time, and a fixed repeating schedule $\mathbf{F}^{(T)}$. Each of these components will be described in detail below.

The routing topology is described by $G = (\mathcal{V}, \mathcal{E})$, a connected directed acyclic graph with the set of vertices (nodes) $\mathcal{V} = \{1, \dots, M\}$ and the set of directed edges (links) $\mathcal{E} \subseteq \{(i, j) : i, j \in \mathcal{V}, i \neq j\}$, where the number of

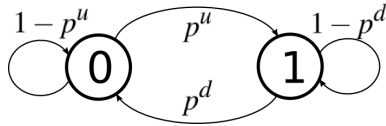


Fig. 3. Gilbert-Elliott link model

edges is denoted E . The source node is denoted a and the sink (destination) node is denoted b . Only the destination node has no outgoing edges.

At any moment in time, the links in G can be either be up (succeeds if attempt to transmit packet) or down (fails if attempt to transmit packet). Thus, there are 2^E possible topology realizations $\tilde{G} = (\mathcal{V}, \tilde{\mathcal{E}})$, where $\tilde{\mathcal{E}} \subseteq \mathcal{E}$ represents the edges that are up.²

At time t_k , the actual state of the topology is one of the topology realizations but it is not known to the network estimator. With some abuse of terminology, we define $G^{(k)}$ to be the random variable representing the state of the topology at time t_k .³

This paper considers the network under the Gilbert-Elliott (G-E) link model and assumes all the links in the network are independent. The G-E link model represents each link l by the two-state Markov chain shown in Fig. 3. At each sample time k , a link in state 0 (down) transitions to state 1 (up) with probability p_l^u , and a link from state 1 transitions to state 0 with probability p_l^d .⁴ The steady-state probability of being in state 1, which we use as the a priori probability of the link being up, is

$$p_l = p_l^u / (p_l^u + p_l^d) .$$

The fixed, repeating schedule of length T is represented by a sequence of matrices $\mathbf{F}^{(T)} = (F^{(1)}, F^{(2)}, \dots, F^{(T)})$, where the matrix $F^{(t-1 \pmod{T)+1)}$ represents the links scheduled at time t . The matrix $F^{(t)} \in \{0, 1\}^{M \times M}$ is defined from the set $\mathcal{F}^{(t)} \subseteq \mathcal{E}$ containing the links scheduled for transmission at time t . We assume that nodes can only unicast packets, meaning that for all nodes i , if $(i, j) \in \mathcal{F}^{(t)}$ then for all $v \neq j$, $(i, v) \notin \mathcal{F}^{(t)}$. Furthermore, a node holds onto a packet if the transmission fails and can retransmit the packet the next time an outgoing link is scheduled (hop-by-hop routing). Thus, the matrix $F^{(t)}$ has entries

$$F_{ij}^{(t)} = \begin{cases} 1 & \text{if } (i, j) \in \mathcal{F}^{(t)}, \text{ or} \\ & \text{if } i = j \text{ and } \forall v \in \mathcal{V}, (i, v) \notin \mathcal{F}^{(t)} \\ 0 & \text{otherwise.} \end{cases}$$

An exact description of the network consists of the sequence of topology realizations over time and the schedule $\mathbf{F}^{(T)}$. Assuming a topology realization \tilde{G} , the links that are scheduled and up at any given time t are represented by the matrix $\tilde{F}^{(t; \tilde{G})} \in \{0, 1\}^{M \times M}$, with entries

$$\tilde{F}_{ij}^{(t; \tilde{G})} = \begin{cases} 1 & \text{if } (i, j) \in \mathcal{F}^{(t)} \cap \tilde{\mathcal{E}}, \text{ or} \\ & \text{if } i = j \text{ and } \forall v \in \mathcal{V}, (i, v) \notin \mathcal{F}^{(t)} \cap \tilde{\mathcal{E}} \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Define the matrix $\tilde{F}^{(t, t'; \tilde{G})} = \tilde{F}^{(t; \tilde{G})} \tilde{F}^{(t+1; \tilde{G})} \dots \tilde{F}^{(t'; \tilde{G})}$, such that entry $\tilde{F}_{ij}^{(t, t'; \tilde{G})}$ is 1 if a packet at node i at time t will be at node j at time t' , and is 0 otherwise. Since the destination b has no outgoing links, a packet sent from the source a at time t reaches the destination b at or before time t' if and only if $\tilde{F}_{ab}^{(t, t'; \tilde{G})} = 1$. To simplify the notation, let the function δ_κ indicate whether the packet delivery $\tilde{v} \in \{0, 1\}$ is consistent with the topology realization \tilde{G} , assuming the packet was generated at t_κ , i.e.,

$$\delta_\kappa(\tilde{v}; \tilde{G}) = \begin{cases} 1 & \text{if } \tilde{v} = \tilde{F}_{ab}^{(t_\kappa, t'_\kappa; \tilde{G})} \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

The function assumes the fixed repeating schedule $\mathbf{F}^{(T)}$, the packet generation time t_κ , the deadline t'_κ , the source a , and the destination b are implicitly known.

2.2 Network Estimators

As shown in Fig. 2, at each sample time k the network estimator \hat{N} takes as input the previous packet delivery ν_{k-1} , estimates the topology realization using the network model and all past packet deliveries, and outputs the joint probability distribution of future packet deliveries $f_{\nu_k^{k+H-1}}$. For clarity in the following exposition, let $\mathcal{V}_\kappa \in \{0, 1\}$ be the value taken on by the packet delivery random variable ν_κ at some past sample time κ . Let the vector $\mathbf{V}_0^{k-1} = [\mathcal{V}_0, \dots, \mathcal{V}_{k-1}]$ denote the history of packet deliveries at sample time k , the values taken on by the vector of random variables $\nu_0^{k-1} = [\nu_0, \dots, \nu_{k-1}]$. Then,

$$f_{\nu_k^{k+H-1}}(\tilde{\nu}_0^{H-1}) = \mathbb{P}(\nu_k^{k+H-1} = \tilde{\nu}_0^{H-1} | \nu_0^{k-1} = \mathbf{V}_0^{k-1}) \quad (4)$$

is the prediction of the next H packet deliveries, where $\nu_k^{k+H-1} = [\nu_k, \dots, \nu_{k+H-1}]$ is a vector of random variables representing future packet deliveries and $\tilde{\nu}_0^{H-1} \in \{0, 1\}^H$.

The parameters of the network models — topology G , schedule $\mathbf{F}^{(T)}$, link probabilities $\{p_l\}_{l \in \mathcal{E}}$ or $\{p_l^u, p_l^d\}_{l \in \mathcal{E}}$, source a , sink b , packet generation times t_k , and deadlines t'_k — are known a priori to the GEIHS network estimator and are left out of the conditional probability expressions.

In Section 3, we will use the probability distribution on the topology realizations (our network state estimate),

$$\mathbb{P}(G^{(k)} = \tilde{G} | \nu_0^{k-1} = \mathbf{V}_0^{k-1}) ,$$

to obtain $f_{\nu_k^{k+H-1}}$ from \mathbf{V}_0^{k-1} and the network model.⁵

2.3 FPD Controller

The FPD controller (\mathcal{C} in Fig. 2) optimizes the control signals to the statistics of the future packet delivery sequence, derived from the past packet delivery sequence. We choose the optimal control framework because the cost function allows us to easily compare the FPD controller

⁵ For compact notation, we will often use \mathbf{V}_0^{k-1} in place of $\nu_0^{k-1} = \mathbf{V}_0^{k-1}$ and only write the random variable and not its value in the probability expressions.

² Symbols with a tilde ($\tilde{\cdot}$) denote values that can be taken on by random variables, and can be the arguments to probability distribution functions (pdfs).

³ Strictly speaking, $G^{(k)}$ is a function mapping events to the set of all topology realizations, not to the set of real numbers.

⁴ We can easily instead use a G-E link model that advances at each time step t , but it would make the following exposition and notation more complicated.

with other controllers. The control policy operates on the information set

$$\mathcal{I}_k^c = \{x_0^k, u_0^{k-1}, \nu_0^{k-1}\} \quad (5)$$

The control policy minimizes the linear quadratic cost function

$$J = \min_{u_0, \dots, u_{N-1}} \mathbb{E} \left[x_N^T Q_0 x_N + \sum_{n=0}^{N-1} x_n^T Q_1 x_n + \nu_n^T u_n^T Q_2 u_n \right] \quad (6)$$

where Q_0 , Q_1 , and Q_2 are positive definite weighting matrices and N is the finite horizon. Section 4 will show that the resulting architecture separates into a network estimator and a controller which uses the pdf $f_{\nu_k^{k+H-1}}$ supplied by the network estimator (\hat{N} in Fig. 2) to find the control signals u_k .

3. NETWORK ESTIMATION AND PACKET DELIVERY PREDICTION

We will use recursive Bayesian estimation to estimate the state of the network, and use the network state estimate to predict future packet deliveries.

3.1 GEIHS Network Estimator

The steps in the GEIHS network estimator are derived from (4). We introduce new notation for conditional pdfs (i.e., α_k, β_k, Z_k), which will be used later to state the steps in the estimator compactly.⁶ Since all the links in the network are independent, the probability that a given topology \tilde{G}' at sample time $k-1$ transitions to a topology \tilde{G} after one sample time is given by

$$\Gamma(\tilde{G}; \tilde{G}') = \mathbb{P}(G^{(k)} | G^{(k-1)}) = \left(\prod_{l_1 \in \tilde{\mathcal{E}}' \cap \tilde{\mathcal{E}}} 1 - p_{l_1}^d \right) \left(\prod_{l_2 \in \tilde{\mathcal{E}}' \setminus \tilde{\mathcal{E}}} p_{l_2}^d \right) \times \left(\prod_{l_3 \in \tilde{\mathcal{E}} \setminus \tilde{\mathcal{E}}'} p_{l_3}^u \right) \left(\prod_{l_4 \in \tilde{\mathcal{E}} \setminus (\tilde{\mathcal{E}}' \cup \tilde{\mathcal{E}})} 1 - p_{l_4}^u \right) \quad (7)$$

First, express $f_{\nu_k^{k+H-1}}(\tilde{\nu}_0^{H-1})$ as

$$f_{\nu_k^{k+H-1}}(\tilde{\nu}_0^{H-1}) = \overbrace{\mathbb{P}(\nu_k^{k+H-1} | \nu_0^{k-1})} = \sum_{G^{(k+H-1)}} \underbrace{\mathbb{P}(\nu_{k+H-1} | G^{(k+H-1)})}_{\delta_{k+H-1}(\tilde{\nu}_{H-1}; \tilde{G}_{H-1})} \cdot \underbrace{\mathbb{P}(\nu_k^{k+H-2}, G^{(k+H-1)} | \nu_0^{k-1})}_{\alpha_{H-1|k-1}(\tilde{\nu}_0^{H-2}, \tilde{G}_{H-1})}$$

where for $h = 2, \dots, H-1$

$$\alpha_{h|k-1}(\tilde{\nu}_0^{h-1}, \tilde{G}_h) = \overbrace{\mathbb{P}(\nu_k^{k+h-1}, G^{(k+h)} | \nu_0^{k-1})} = \sum_{G^{(k+h-1)}} \left(\underbrace{\mathbb{P}(G^{(k+h)} | G^{(k+h-1)})}_{\Gamma(\tilde{G}_h; \tilde{G}_{h-1})} \cdot \underbrace{\mathbb{P}(\nu_{k+h-1} | G^{(k+h-1)})}_{\delta_{k+h-1}(\tilde{\nu}_{h-1}; \tilde{G}_{h-1})} \right) \times \underbrace{\mathbb{P}(\nu_k^{k+h-2}, G^{(k+h-1)} | \nu_0^{k-1})}_{\alpha_{h-1|k-1}(\tilde{\nu}_0^{h-2}, \tilde{G}_{h-1})} \quad (8)$$

⁶ A semicolon is used in the conditional pdfs to separate the values being conditioned on from the remaining arguments.

We used the relation

$$\mathbb{P}(\nu_{k+h-1} | G^{(k+h-1)}, \nu_k^{k+h-2}, \nu_0^{k-1}) = \mathbb{P}(\nu_{k+h-1} | G^{(k+h-1)})$$

When $h = 1$, replace $\mathbb{P}(\nu_k^{k+h-2}, G^{(k+h-1)} | \nu_0^{k-1})$ in (8) with $\mathbb{P}(G^{(k)} | \nu_0^{k-1}) = \beta_{k|k-1}(\tilde{G})$. The value $\beta_{k|k-1}(\tilde{G})$ comes from

$$\underbrace{\mathbb{P}(G^{(k)} | \nu_0^{k-1})}_{\beta_{k|k-1}(\tilde{G})} = \sum_{G^{(k-1)}} \underbrace{\mathbb{P}(G^{(k)} | G^{(k-1)})}_{\Gamma(\tilde{G}; \tilde{G}')} \cdot \underbrace{\mathbb{P}(G^{(k-1)} | \nu_0^{k-1})}_{\beta_{k-1|k-1}(\tilde{G}')} \\ \underbrace{\mathbb{P}(G^{(k-1)} | \nu_0^{k-1})}_{\beta_{k-1|k-1}(\tilde{G})} = \frac{\overbrace{\mathbb{P}(\nu_{k-1} | G^{(k-1)})}_{\delta_{k-1}(\nu_{k-1}; \tilde{G})} \cdot \overbrace{\mathbb{P}(G^{(k-1)} | \nu_0^{k-2})}_{\beta_{k-1|k-2}(\tilde{G})}}{\underbrace{\mathbb{P}(\nu_{k-1} | \nu_0^{k-2})}_{Z_{k-1}}}$$

Again, we used the independence of future packet deliveries from past packet deliveries given the network state,

$$\mathbb{P}(\nu_{k-1} | G^{(k-1)}, \nu_0^{k-2}) = \mathbb{P}(\nu_{k-1} | G^{(k-1)})$$

Note that $\mathbb{P}(\nu_{k-1} | \nu_0^{k-2})$ is the same for all \tilde{G} , so it is treated as a normalization constant. Finally, $\beta_{0|-1}(\tilde{G}) = \mathbb{P}(G^{(0)})$, where all links are independent and have link probabilities equal to their steady-state probability of being in state 1, and is expressed in (14) below.

To summarize, the *GEIHS Network Estimator and Packet Delivery Predictor* is a recursive Bayesian estimator. The measurement output step consists of

$$f_{\nu_k^{k+H-1}}(\tilde{\nu}_0^{H-1}) = \sum_{\tilde{G}_{H-1}} \delta_{k+H-1}(\tilde{\nu}_{H-1}; \tilde{G}_{H-1}) \cdot \alpha_{H-1|k-1}(\tilde{\nu}_0^{H-2}, \tilde{G}_{H-1}) \quad (9)$$

where the function $\alpha_{H-1|k-1}$ is obtained from the following recursive equation for $h = 2, \dots, H-1$:

$$\alpha_{h|k-1}(\tilde{\nu}_0^{h-1}, \tilde{G}_h) = \sum_{\tilde{G}_{h-1}} \left(\Gamma(\tilde{G}_h; \tilde{G}_{h-1}) \times \delta_{k+h-1}(\tilde{\nu}_{h-1}; \tilde{G}_{h-1}) \cdot \alpha_{h-1|k-1}(\tilde{\nu}_0^{h-2}, \tilde{G}_{h-1}) \right) \quad (10)$$

with initial condition

$$\alpha_{1|k-1}(\tilde{\nu}_0^0, \tilde{G}_1) = \sum_{\tilde{G}} \Gamma(\tilde{G}_1; \tilde{G}) \cdot \delta_k(\tilde{\nu}_0; \tilde{G}) \cdot \beta_{k|k-1}(\tilde{G}) \quad (11)$$

The prediction and innovation steps consist of

$$\beta_{k|k-1}(\tilde{G}) = \sum_{\tilde{G}'} \Gamma(\tilde{G}; \tilde{G}') \cdot \beta_{k-1|k-1}(\tilde{G}') \quad (12)$$

$$\beta_{k-1|k-1}(\tilde{G}) = \frac{\delta_{k-1}(\nu_{k-1}; \tilde{G}) \cdot \beta_{k-1|k-2}(\tilde{G})}{Z_{k-1}} \quad (13)$$

$$\beta_{0|-1}(\tilde{G}) = \left(\prod_{l \in \tilde{\mathcal{E}}} p_l \right) \left(\prod_{l \in \mathcal{E} \setminus \tilde{\mathcal{E}}} 1 - p_l \right), \quad (14)$$

where $\alpha_{h|k-1}$, $\beta_{k-1|k-1}$, and $\beta_{k|k-1}$ are functions, Z_{k-1} is a normalization constant such that $\sum_{\tilde{G}} \beta_{k-1|k-1}(\tilde{G}) = 1$, and the functions δ_κ (for the different values of κ above) and Γ are defined by (3) and (7), respectively.

3.2 Discussion

The network estimators are trying to estimate network parameters using measurements collected at the border

of the network, a general problem studied in the field of network tomography (Castro et al., 2004) under various problem setups. One of the greatest challenges in network tomography is getting good estimates with low computational complexity estimators.

Our proposed network estimator is “optimal” with respect to our models in the sense that there is no loss of information, but it is computationally expensive. The GEIHS network estimator takes $O(E2^{2E})$ to initialize and $O(2^H 2^{2E})$ for each update step. The proofs are given in Chen et al. (2010). A good direction for future research is to find lower complexity, suboptimal network estimators for our problem setup, and compare them to our optimal network estimators.

Our network estimators can easily be extended to incorporate additional observations besides past packet deliveries, such as the packet delay and packet path traces. The latter can be obtained by recording the state of the links that the packet has tried to traverse in the packet payload. The function δ_{k-1} in (13) just needs to be replaced with another function that returns 1 if the received observation is consistent with a network topology \tilde{G} , and 0 otherwise. The advantage of using more observations than the one bit of information provided by a packet delivery is that it will help the GEIHS network estimator more quickly detect changes in the network state.

Note that the network state probability distribution, $\beta_{k-1|k-1}(\tilde{G})$ in (13), does not need to converge to a probability distribution describing one topology realization to yield good packet predictions $f_{\nu_k^{k+H-1}}(\tilde{\nu}_0^{H-1})$. Several topology realizations \tilde{G} may result in the same packet delivery sequence.

Also, note that the GEIHS network estimator performs better when the links in the network are more bursty. Long bursts of packet losses from bursty links result in poor control system performance, which is when the network estimator would help the most.

4. FPD CONTROLLER

The FPD controller is derived using dynamic programming in Chen et al. (2010). From the derivation, we find that the optimal cost-to-go is of the form $V_k = x_k^T S_k(\nu_0^{k-1})x_k + s_k(\nu_0^{k-1})$, where $S_k : \{0, 1\}^k \mapsto \mathbb{S}_+^L$ and $s_k : \{0, 1\}^k \mapsto \mathbb{R}_+$ are given by

$$\begin{aligned} S_k(\nu_0^{k-1}) &= Q_1 + A^T \mathbb{E} [S_{k+1}(\nu_0^k) | \nu_0^{k-1}] A - \mathbb{P}(\nu_k=1 | \nu_0^{k-1}) \\ &\quad \left[A^T S_{k+1}(\nu_k=1, \nu_0^{k-1}) B (Q_2 + B^T S_{k+1}(\nu_k=1, \nu_0^{k-1}) B)^{-1} \right. \\ &\quad \left. \times B^T S_{k+1}(\nu_k=1, \nu_0^{k-1}) A \right] \\ s_k(\nu_0^{k-1}) &= \text{trace} \left\{ \mathbb{E} [S_{k+1}(\nu_0^k) | \nu_0^{k-1}] R_w \right\} + \mathbb{E} [s_{k+1}(\nu_0^k) | \nu_0^{k-1}] \end{aligned} \quad (15)$$

Theorem 1. For a plant with state dynamics given by (1), the optimal control policy operating on the information set (5) which minimizes the cost function (6) results in an optimal control signal, $u_k = -L_k x_k$, where L_k is

$$L_k = (Q_2 + B^T S_{k+1}(\nu_k=1, \nu_0^{k-1}) B)^{-1} B^T S_{k+1}(\nu_k=1, \nu_0^{k-1}) A \quad (16)$$

and $S_k(\nu_0^{k-1})$ is given by (15).

The proof of this theorem is presented in Chen et al. (2010). Notice that S_k and s_k are functions of the variables ν_0^{k-1} . Since the value of S_{k+1} is required at sample time k , we compute its conditional expectation as

$$\begin{aligned} \mathbb{E} [S_{k+1}(\nu_0^k) | \nu_0^{k-1}] &= \mathbb{P}(\nu_k = 1 | \nu_0^{k-1}) S_{k+1}(\nu_k=1, \nu_0^{k-1}) \\ &\quad + \mathbb{P}(\nu_k = 0 | \nu_0^{k-1}) S_{k+1}(\nu_k=0, \nu_0^{k-1}) \quad . \quad (17) \end{aligned}$$

Using the above expressions, we obtain the net cost to be

$$J = \text{trace} S_0 R_0 + \sum_{k=0}^{N-1} \text{trace} \left\{ \mathbb{E} [S_{k+1}(\nu_0^k)] R_w \right\} \quad . \quad (18)$$

Notice that the control inputs u_k are only carried over the network and applied to the plant. They do not influence ν_k^N or the network itself. Thus, the architecture separates into a network estimator and controller, as shown in Fig. 2.

4.1 Algorithm to Compute Optimal Control Gain

At sample time k , we have ν_0^{k-1} . To compute L_k , we need $S_{k+1}(\nu_k=1, \nu_0^{k-1})$, which can only be obtained through a backward recursion from S_N . This requires knowledge of ν_k^{N-1} , which are unavailable at sample time k . Thus, we must evaluate $\{S_{k+1}, \dots, S_N\}$ for every possible sequence of arrivals ν_k^{N-1} . This algorithm is described below.

- (1) Initialization: $S_N(\nu_k^{N-1} = \tilde{\nu}_0^{N-k-1}, \nu_0^{k-1}) = Q_0$,
 $\forall \tilde{\nu}_0^{N-k-1} \in \{0, 1\}^{N-k}$.
- (2) for $n = N-1 : -1 : k+1$
 - a) Using (17), compute $\mathbb{E} [S_{n+1}(\nu_0^n) | \nu_0^{n-1}, \tilde{\nu}_0^{n-k-1}]$,
 $\forall \tilde{\nu}_0^{n-k-1} \in \{0, 1\}^{n-k}$.
 - b) Using (15), compute $S_n(\nu_k^{n-1} = \tilde{\nu}_0^{n-k-1}, \nu_0^{k-1})$,
 $\forall \tilde{\nu}_0^{n-k-1} \in \{0, 1\}^{n-k}$.
- (3) Compute L_k using $S_{k+1}(\nu_k=1, \nu_0^{k-1})$.

For $k=0$, the values S_0 , $\mathbb{E}[S_1(\nu_0)]$, and the other values obtained above can be used to evaluate the cost function according to (18).

4.2 Discussion

The FPD controller is optimal but computationally expensive, as it requires an enumeration of all possible packet delivery sequences from the current sample time until the end of the control horizon to calculate the optimal control gain (16) at every instant k . Running the algorithm at each sample time k takes $O(q^3(N-k)2^{N-k})$ operations, where $q = \max(\ell, m)$ and ℓ and m are the dimensions of the state and control vectors. The proof is given in Chen et al. (2010).

5. EXAMPLES AND SIMULATIONS

Using the system architecture depicted in Fig. 2, we will demonstrate the GEIHS network estimator on a small mesh network and use the packet delivery predictions in our FPD controller. Fig. 4 depicts the routing topology and short repeating schedule of the network. Packets are generated at the source every 409 time slots,⁷ and the

⁷ Effectively, the packets are generated every $9 + 4K$ time slots, where K is a very large integer, so we can assume slow system dynamics with respect to time slots and ignore the delay introduced by the network.

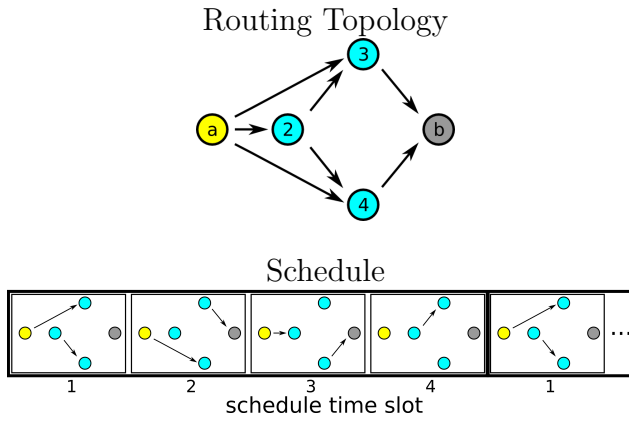


Fig. 4. Example of a simple mesh network for network estimation.

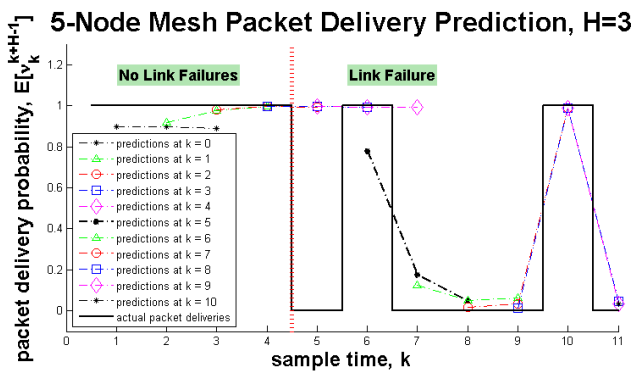


Fig. 5. Packet delivery predictions when network in Fig. 4 has all links up and then link (3, b) fails.

packet delivery deadline is $t'_k - t_k = 9, \forall k$. The network estimator assumes all links have $p^u = 0.0135$ and $p^d = 0.0015$.

The packet delivery predictions from the network estimator are shown in Fig. 5. Although the network estimator provides $f_{\nu_k^{k+H-1}}(\hat{\nu}_0^{H-1})$, at each sample time k we plot the average prediction $E[\nu_k^{k+H-1}]$. In this example, all the links are up for $k \in \{1, \dots, 4\}$ and then link (3, b) fails from $k = 5$ onwards. After seeing a packet loss at $k = 5$, the network estimator revises its packet delivery predictions and now thinks there will be a packet loss at $k = 7$. The average prediction for the packet delivery at a particular sample time tends toward 1 or 0 as the network estimator receives more information (in the form of packet deliveries) about the new state of the network.

The prediction for $k = 7$ (packet generated at schedule time slot 3) at $k = 5$ is influenced by the packet delivery at $k = 5$ (packet generated at schedule time slot 1) because hop-by-hop routing allows the packets to traverse the same links under some realizations of the underlying routing topology G . Mesh networks with many interleaved paths allow packets generated at different schedule time slots to provide information about each others' deliveries, provided the links in the network have some memory. As discussed in Section 3.2, since a packet delivery provides only one bit of information about the network state, it may take several packet deliveries to get good predictions after the network changes.

FPD, IID, and ON LQG Control Comparison

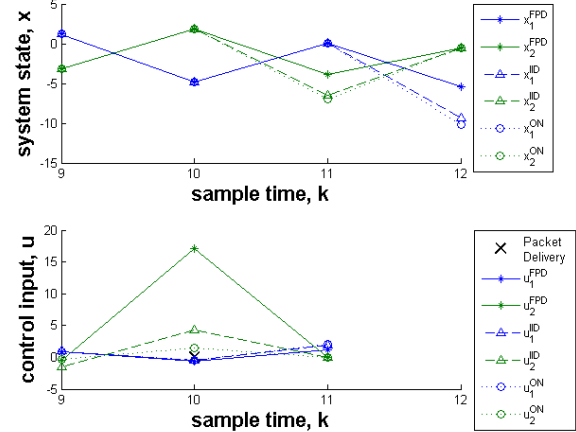


Fig. 6. Plot of the different control signals and state vectors when using the FPD controller, an IID controller, and an ON controller (see text for details).

Now, consider a linear plant with the following parameters

$$A = \begin{bmatrix} 0 & 1.5 \\ 1.5 & 0 \end{bmatrix}, B = \begin{bmatrix} 5 & 0 \\ 0 & 0.2 \end{bmatrix}, R_w = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}, R_0 = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$$

$$Q_1 = Q_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, Q_n = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$$

The transfer matrix A flips and expands the components of the state at every sampling instant. The input matrix B requires the second component of the control input to be larger in magnitude than the first component to have the same effect on the respective component of the state. Also, the final state is weighted more than the other states in the cost criterion. We compare three finite horizon LQG controllers: the FPD controller, an IID controller, and an ON (always-online) controller. The IID controller was described in (Schenato et al., 2007) and assumes that the packet deliveries are i.i.d. with the a priori probability of delivering a packet through the network,⁸ and the ON controller is the classical LQG controller which assumes no packet losses on the control link. See Chen et al. (2010) for a more detailed description of the controllers.

The controllers are connected to the plant at sample times $k = 9, 10, 11$ through the network example given in Fig. 5. Fig. 6 shows the control signals computed by the different controllers and the plant states when the control signals are applied following the actual packet delivery sequence. From the predictions at $k = 8, 9, 10$ in Fig. 5, we see that the FPD controller has better knowledge of the packet delivery sequence than the other two controllers. The FPD controller uses this knowledge to compute an optimal control signal that outputs a large magnitude for the second component of u_{10} , despite the high cost of this signal. The IID and ON controllers believe the control packet is likely to be delivered at $k = 11$ and choose, instead, to output a smaller increase in the first component of u_{11} , since this will have the same effect on the final state if the control packet at $k = 11$ is successfully delivered.

The FPD controller is better than the other controllers at driving the first component of the state close to zero at

⁸ Using the stationary probability of each link under the G-E link model to calculate the end-to-end probability of delivering a packet through the network.

Table 1. Simulated LQG Costs (10,000 runs)
Example Described in Section 5

FPD Controller	IID Controller	ON Controller
681.68	1,008.2	1,158.9

the end of the control horizon, $k = 12$. Thus, the packet delivery predictions from the network estimator help the FPD controller significantly lower its LQG cost, as shown in Table 1. The reported costs are averages over 10,000 Monte-Carlo simulation runs of the system, where the network state is set to the one described above.

6. CONCLUSIONS

This paper proposes two network estimators based on simple network models to characterize wireless mesh networks for NCSs. The goal is to obtain a better abstraction of the network, and interface to the network, to present to the controller and (future work) network manager. To get better performance in a NCS, the network manager needs to *control and reconfigure* the network to reduce outages and the controller needs to *react to or compensate* for the network when there are unavoidable outages. We studied a specific NCS architecture where the actuation channel was over a lossy wireless mesh network and a network estimator provided packet delivery predictions for a finite horizon, Future-Packet-Delivery-optimized LQG controller.

There are several directions for extending the basic problem setup in this paper, including those mentioned in Section 3.2. For instance, placing the network estimator(s) on the actuators in the general system architecture depicted in Fig. 1 is a more realistic setup but will introduce a lossy channel between the network estimator(s) and the controller(s). Also, one can study the use of packet delivery predictions in a receding horizon controller rather than a finite horizon controller.

Aside from studying variations of the basic problem setup, there are still several fundamental questions that deserve further investigation. We know that the Gilbert-Elliott link model is a simplification of a lossy wireless link, but can we use more sophisticated network models without overmodeling? Are there classes of routing topologies where packet delivery statistics are less sensitive to the parameters in the Gilbert-Elliott link models, p_l^u and p_l^d , which may be difficult to estimate on all links in the network? How do we build networks (e.g., select routing topologies and schedules) that are “robust” to link modeling error and provide good packet delivery statistics (e.g., low packet loss, low delay) for NCSs? The latter half of the question is partially addressed by works like Soldati et al. (2010), which studies how to schedule a network to maximize reliability under packet deadline constraints.

REFERENCES

Bruno, R., Conti, M., and Gregori, E. (2005). Mesh networks: commodity multihop ad hoc networks. *IEEE Communications Magazine*, 43(3), 123–131.

Castro, R., Coates, M., Liang, G., Nowak, R., and Yu, B. (2004). Network tomography: Recent developments. *Statistical Science*, 19(3), 499–517.

Chen, P., Ramesh, C., and Johansson, K.H. (2010). Network estimation and packet delivery prediction for control over wireless mesh networks. Technical Report TRITA-EE:043, KTH Royal Institute of Technology. URL <http://arxiv.org/abs/1103.5405>.

Elia, N. (2005). Remote stabilization over fading channels. *Systems & Control Letters*, 54(3), 237–249.

Gupta, V., Dana, A., Hespanha, J., Murray, R., and Hassibi, B. (2009). Data transmission over networks for estimation and control. *IEEE Trans. on Automatic Control*, 54(8), 1807–1819.

Hespanha, J.P., Nagnshtabrizi, P., and Xu, Y. (2007). A survey of recent results in networked control systems. *Proc. of the IEEE*, 95, 138–162.

International Society of Automation (2010). ISA-SP100 wireless systems for automation website. <http://www.isa.org/isa100>.

Ishii, H. (2008). H_∞ control with limited communication and message losses. *Systems & Control Letters*, 57(4), 322–331.

Olfati-Saber, R., Fax, J., and Murray, R. (2007). Consensus and cooperation in networked multi-agent systems. *Proc. of the IEEE*, 95(1), 215–233.

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

Robinson, C. and Kumar, P. (2007). Control over networks of unreliable links: Controller location and performance bounds. In *Proc. of the 5th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, 1–8.

Schenato, L., Sinopoli, B., Franceschetti, M., Poolla, K., and Sastry, S.S. (2007). Foundations of control and estimation over lossy networks. *Proc. of the IEEE*, 95, 163–187.

Seiler, P. and Sengupta, R. (2005). An H_∞ approach to networked control. *IEEE Transactions on Automatic Control*, 50, 356–364.

Soldati, P., Zhang, H., Zou, Z., and Johansson, M. (2010). Optimal routing and scheduling of deadline-constrained traffic over lossy networks. In *Proc. of the IEEE Global Telecommunications Conference*. Miami FL, USA.

Willig, A., Kubisch, M., Hoene, C., and Wolisz, A. (2002). Measurements of a wireless link in an industrial environment using an IEEE 802.11-compliant physical layer. *IEEE Trans. on Industrial Electronics*, 49(6), 1265–1282.

Wireless Industrial Networking Alliance (2010). WINA website. <http://www.wina.org>.