Latency and Connectivity Analysis Tools for Wireless Mesh Networks

Phoebus Chen and Shankar Sastry
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
Berkeley, California 94720
{phoebusc,sastry}@eecs.berkeley.edu

***Abstract*—There has been a recent rise in interest in building networked control systems over a wireless network, whether they be for robot navigation, multi-robot systems, or traditional industrial automation. The wireless networks in these systems must deliver packets between the controller and the actuators/sensors reliably and with low latency. Furthermore, they should be amenable to modeling and characterization so they can be designed as part of a complete control system. Mesh networks are particularly suited for control applications because they provide greater reliability through path diversity.**

This paper introduces tools for characterizing the end-to-end connectivity of two points in a wireless mesh network as a function of latency. In particular, we use tools derived from Markov chain models to compare end-to-end connectivity in two routing protocols running on the Data Link/MAC layer provided by Dust Network's Time Synchronized Mesh Protocol (TSMP): Directed Staged Flooding (DSF) and Dust Network's Unicast Path Diversity (UPD). These models also allow us to calculate the traffic load, the sensitivity of end-to-end connectivity to link estimation error, and the robustness of the network to node failure. The paper gives an example of how these tools can be used to evaluate the feasibility of running control applications over sensor networks.

I. Introduction

Wireless mesh networking has enabled a new generation of pervasive devices with the potential to provide reliable communication in environments with limited fixed infrastructure. Wireless sensor and actuator networks, sometimes simply referred to as sensor networks, are one such class of devices which can use mesh networking to connect sensors and actuators that monitor our environment and control other instruments [1]. Sensor networks enable a large variety of applications including outdoor environmental monitoring for scientific research, diagnosing civil structures for damage under earthquakes, monitoring the sick and elderly for assisted living at home, and sensing and control of industrial automation equipment, among others. Feedback control systems are among the hardest types of applications for sensor networks because they place stringent requirements on reliability and latency. These applications motivate the need for tools to characterize wireless mesh networking on sensor networks for control systems.

A. Control over Lossy Networks

The recent increased interest in using wireless networks for industrial automation culminated in the formation of the ISA-SP100 committee to set up an industrial wireless standard [2]. The current version of the ISA-SP100 standard plans to build on the PHY layer provided by the IEEE 802.15.4 standard [3] for low-power, ad-hoc, wireless, personal area networks.

The key issues in using wireless communications for control systems is reliability and latency. The designer of the control system needs to know the probability of end-to-end delivery of the packet, p_{net}, as a function of delay, t_d, to provide performance guarantees on the controller. Many papers in the area of *Networked Control Systems* study the impact of packet loss on the stability of discrete-time estimators [4], [5], [6], assuming that packets arriving after a deadline (the sampling period) are lost. But despite using the moniker "network", many of these papers derive results using a simple, point-to-point communication channel. For instance, in [5] Sinopoli et al. assume the packet loss in the channel can be modeled by an i.i.d. Bernoulli random variable.

The goal of this paper is to model examples from two classes of TDMA mesh networks for control systems, multi-path routing with retransmissions and constrained flooding. We wish to derive the function $p_{net}^{(t_d)}$ relating the probability of end-to-end delivery to delay for a packet in a wireless mesh network providing communication for a control system so we can use the existing theory in Networked Control Systems to characterize the system's stability and performance. The paper focuses on TDMA networks because of the difficulty modeling and providing probabilistic guarantees on latency for networks using CSMA/CA contention protocols. It focuses on mesh networks because multiple paths between a source and destination are necessary for good end-to-end reliability.

B. Related Work on Multi-path Routing

To increase reliable end-to-end delivery of packets, many routing schemes propose sending multiple copies of a packet on multiple paths. These range from controlled, probabilistic flooding schemes like ARRIVE [7] to schemes that code the data over a set of packets and send them along disjoint or braided (partially disjoint) paths such that only a subset of the packets need to be received for reconstruction [8]. Multi-path routing schemes are also distinguished by whether an end-to-end path is selected at the source for a packet, such as the braided and disjoint paths of [9], or whether the packet can switch paths during transit, as in "true mesh" routing protocols

ROBOCOMM 2007, 14th – 16th Oct 2007, Athens, Greece.

ISBN 978-963-9799-08-0
DOI 10.4108/ICST.ROBOCOMM2007. 2121

like Unicast Path Diversity (UPD) over Time Synchronized Mesh Protocol (TSMP) [10][1] and ARRIVE. The protocols studied in this paper will be of the latter type.

Many of the multi-path routing papers use simulations to demonstrate qualitative features of their routing schemes [9], [7]. A small set of papers try to mathematically model and analyze the benefits of multi-path routing, but they either model at the level of paths or assume networks with a very large number of nodes. In [8], Dulman et al. perform some simple analysis to get the tradeoff between traffic and reliability, but the analysis does not consider latency. Furthermore, the calculations use the end-to-end connection probability of disjoint *paths*, not individual link probabilities, and hence do not account for varying path lengths or link probabilities. In [11], Nasipuri et al. propose a multi-path extension to DSR and the analysis focuses on finding the statistics of the time between successive route discoveries. Again, the paper builds on a path model with path lifetimes drawn from a distribution instead of a link model with individual link probabilities. In [12], the authors use a geometry-based argument on networks with a very large number of nodes to argue that k-shortest path routing algorithms only distribute the load evenly through a network when it uses a very large number of paths.

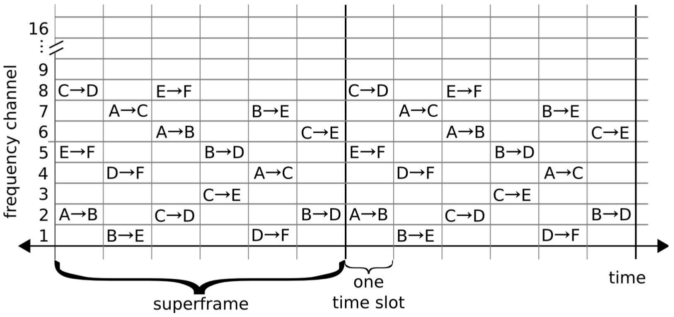
This paper takes a different approach from the papers mentioned above, deriving link-probability-based analysis tools applicable to networks of any size for two examples of mesh routing protocols. We propose models for Unicast Path Diversity and Directed Staged Flooding, with the goals of computing the end-to-end connectivity of the network $p_{net}^{(t_d)}$, identifying *hot spots* in the network where traffic is concentrated, and finding the robustness of the calculations to link probability modeling uncertainty. Most of the arguments in this paper are presented in more detail in [13].

II. Unicast Path Diversity

Dust Networks, Inc. proposed Unicast Path Diversity (UPD) over Time Synchronized Mesh Protocol (TSMP) [10] for reliable networking in sensor networks. The algorithm exploits frequency, time, and space diversity to achieve what they claim is over 99.9% typical network reliability [14]. We use a general Mesh TDMA Markov Chain (MTMC) model to analyze the performance of UPD over TSMP (hereafter referred to simply as UPD) for incorporation into a control system.

A. Modeling Characteristics

We model UPD as a frequency-hopping TDMA scheme with multi-path routing. UPD is a many-to-one routing protocol, i.e. there is one sink in the network. Each node in the network has multiple parents and the routing graph has no cycles. The links selected for routing are bidirectional, and hence every link transmission can be acknowledged. If a packet transmission is not acknowledged, it is queued in the node for retransmission. As for scheduling, time is divided into time slots, and grouped into superframes (See Figure 1). At each time slot, pairs of nodes are scheduled for transmitting a packet on different frequencies. The superframe containing the schedule of transmissions is repeated over time. Our model uses frequency hopping to justify the assumption that links are independent over retransmissions.



16												
⋮												
9												
8	C→D		E→F				C→D		E→F			
7		A→C			B→E			A→C			B→E	
6			A→B			C→E			A→B			C→E
5	E→F			B→D			E→F			B→D		
4		D→F			A→C			D→F			A→C	
3				C→E						C→E		
2	A→B		C→D			B→D	A→B		C→D			B→D
1		B→E			D→F			B→E			D→F	

Fig. 1. Example of a UPD schedule with superframes and time slots. Here, only 8 of the 16 frequency channels are used.

To construct a model of mesh TDMA routing, we assume knowledge of the routing topology, schedule, and all the link probabilities. Furthermore, we study single packet transmission in the network and do not analyze the effects of queuing.

B. Mesh TDMA Markov Chain Model

Let us represent the routing topology as a graph $G = (\mathcal{V}, \mathcal{E})$, and denote a node in the network as $i \in \mathcal{V} = 1, \ldots, N$, and a link in the network as $l \in \mathcal{E} \subset \{(i, j) \mid i, j \in \mathcal{V}\}$, where $l = (i, j)$ is a link for transmitting packets from node i to node j. Time t will be measured in units of time slots, and let T denote the number of time slots in a superframe. The link success probability for link $l = (i, j)$ at time slot t is denoted $p_l^{(t)}$, or $p_{ij}^{(t)}$. We set $p_l^{(t)} = 0$ when link l is not scheduled to transmit at time t.

For a packet originating from a source node a routed to a sink node b, we wish to compute $p_{net}^{(t_d)}$, the probability the packet reaches b at or before time t_d has elapsed. This is done by a time-varying, discrete-time Markov chain.

Mesh TDMA Markov Chain Model Let the set of states in the Markov chain be the nodes in the network, $\mathcal{V}$. The transition probability from state i to state j at time t is simply $p_{ij}^{(t)}$, with $p_{ii}^{(t)} = 1 - \sum_{j \neq i} p_{ij}^{(t)}$. Let $P^{(t)} = [p_{ij}^{(t)}]^T \in [0, 1]^{N \times N}$ be the column stochastic transition probability matrix for a time slot and $P^{(\underline{T})} = P^{(T)} P^{(T-1)} \ldots P^{(1)}$ be the transition probability matrix for a repeating superframe.[2] Assume

$$P^{(T+h)} = P^{(cT+h)}, \qquad \forall c, h \in \mathbb{Z}_+ \tag{1}$$

meaning that the link probabilities in a time slot do not vary over superframes.

A packet originating at node a is represented by $\mathbf{p}^{(0)} = \mathbf{e}^{[a]}$, where $\mathbf{e}^{[a]}$ is an elementary vector with the a-th element equal

[1] The name *Unicast Path Diversity* is not explicitly mentioned in the reference, but this is the name of the routing protocol that is described.

[2] $[0, 1]$ denotes the closed interval between 0 and 1. $[\cdot]^T$ denotes the transpose of a matrix while $P^{(T)}$ denotes the transition matrix at time T.

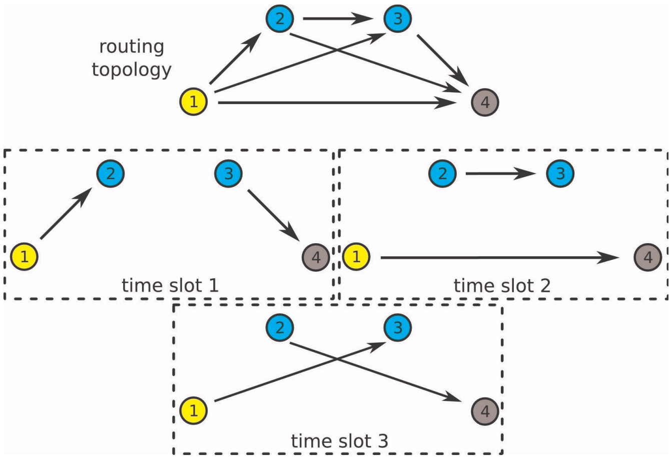


Fig. 2. Multi-path routing example corresponding to (3).

to 1 and all other elements equal to 0. Then,

$$\mathbf{p}^{(t_d)} = P^{(t_d)} \dots P^{(2T+1)} \underbrace{P^{(2T)} P^{(2T-1)} \dots P^{(T+1)}}_{P^{(\underline{T})}} \cdot \underbrace{P^{(T)} P^{(T-1)} \dots P^{(1)}}_{P^{(\underline{T})}} \mathbf{p}^{(0)} \tag{2}$$

represents the probability distribution of the packet over the nodes at time t_d. ■

The sink node b is an absorbing state in the Markov chain, meaning there are no transitions out of that state. This means $p_{net}^{(t_d)} = \mathbf{p}_b^{(t_d)}$, the b-th element of the vector $\mathbf{p}^{(t_d)}$. A good routing schedule would have $p_{net}^{(t_d)} \xrightarrow{t_d \to \infty} 1$, meaning the packet will eventually reach the sink. This condition is satisfied when the MTMC model has only one recurrent class consisting of the sink (See [15]).

C. MTMC Examples and Discussion

An example of a small UPD routing schedule is given in Figure 2, where p_{ij} is the link probability for link (i, j) and $\bar{p}_{ij} = 1 - p_{ij}$. We get the transition probability matrices,

$$P^{(1)} = \begin{bmatrix} \bar{p}_{12} & 0 & 0 & 0 \\ p_{12} & 1 & 0 & 0 \\ 0 & 0 & \bar{p}_{34} & 0 \\ 0 & 0 & p_{34} & 1 \end{bmatrix} \quad P^{(2)} = \begin{bmatrix} \bar{p}_{14} & 0 & 0 & 0 \\ 0 & \bar{p}_{23} & 0 & 0 \\ 0 & p_{23} & 1 & 0 \\ p_{14} & 0 & 0 & 1 \end{bmatrix}$$
$$P^{(3)} = \begin{bmatrix} \bar{p}_{13} & 0 & 0 & 0 \\ 0 & \bar{p}_{24} & 0 & 0 \\ p_{13} & 0 & 1 & 0 \\ 0 & p_{24} & 0 & 1 \end{bmatrix} \tag{3}$$
$$\mathbf{p}^{(0)} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T \qquad P^{(\underline{3})} = P^{(3)} P^{(2)} P^{(1)}$$

The MTMC model can be modified to represent routing topologies and schedules not used by UPD. For instance, UPD avoids creating cycles in the routing graph, as one would want from a good routing algorithm. The MTMC model, however, can model routing cycles that may arise when the network malfunctions. We can still calculate $p_{net}^{(t_d)}$, and we still have $p_{net}^{(t_d)} \xrightarrow{t_d \to \infty} 1$ if no recurrent classes besides the sink are added to the Markov chain. The MTMC model can also be extended to represent mesh networks with multiple collection points (ex. two Internet gateways to a sensor network). In this case, if we let $\mathcal{B}$ be the set of sinks, $p_{net}^{(t_d)} = \sum_{i \in \mathcal{B}} \mathbf{p}_i^{(t_d)}$. Finally, if we wish to model schedules that never retransmit packets, we simply remove the requirement in our MTMC model that $p_{ii}^{(t)} = 1 - \sum_{j \neq i} p_{ij}^{(t)}$, instead replacing it with $p_{ii}^{(t)} = 0$. To ensure that the transition probability matrix $P^{(t)}$ is a column stochastic matrix, we add a dummy state $N+1$ to represent a packet being lost after transmission. Now, $P^{(t)} = [p_{ij}^{(t)}]^T \in [0,1]^{N+1 \times N+1}$, where $p_{i(N+1)}^{(t)} = 1 - \sum_{j \neq i} p_{ij}^{(t)}$, $p_{(N+1)i}^{(t)} = 0$ for all $i \neq N+1$, and $p_{(N+1)(N+1)}^{(t)} = 1$.

D. MTMC Analysis

1) Network-wide Rate of Convergence for $p_{net}^{(t_d)}$: Besides calculating $p_{net}^{(t_d)}$ for one node transmitting to the sink, we would like to calculate the rate of convergence of $p_{net}^{(t_d)} \to 1$ for the *entire* network from $P^{(\underline{T})}$. This may be a useful metric for designing routing schedules to optimize the performance of the network.

Theorem 2.1 (MTMC $p_{net}^{(t_d)}$ converges exponentially to 1): Let $P^{(\underline{T})} \in [0,1]^{N \times N}$ be a diagonalizable, column stochastic matrix with $\lim_{k \to \infty} (P^{(\underline{T})})^k \mathbf{p} = \mathbf{e}^{[b]}$ for all probability vectors $\mathbf{p}$. Here, $\mathbf{e}^{[b]}$ is an elementary vector with the b-th element equal to 1 and all other elements equal to 0, meaning that the routing topology has a unique sink node b which is the unique recurrent state in the Markov chain. Then,

$$p_{net}^{(t_d)} \geq 1 - C(\rho_*)^k, \quad k = \left\lfloor \frac{t_d}{T} \right\rfloor \tag{4}$$

for some constant C dependent on the initial distribution $\mathbf{p}^{(0)}$ and $\rho_* = \max\{|\lambda| : \lambda \text{ is an eigenvalue of } P^{(\underline{T})} \text{ and } |\lambda| < 1\}$. ■

Therefore, $p_{net}^{(t_d)}$ converges to 1 exponentially with a rate ρ_*. The proof of this can be found in [13]. The rate ρ_* gives a sense of how the *worse case* end-to-end connection probability in the network varies as a function of delay.

2) Traffic Distribution: To identify hot spots in the network, we compute the probability that the packet visits a node i *at or before* time t. This can be done by making i an absorbing state in the MTMC model and finding $\mathbf{p}_i^{(t)}$ on the new model.

In other words, $\forall t \in \mathbb{N}, \forall j \in \mathcal{V}$, let

$$\tilde{P}_{ji}^{(t)} = 0$$
$$\tilde{P}_{ii}^{(t)} = 1$$
$$\tilde{P}_{mn}^{(t)} = P_{mn}^{(t)} \quad \forall m, n \in \mathcal{V}, n \neq i$$

(See Figure 3). The resulting model has two absorbing states, b and i. $\alpha_i^{(t)} = \tilde{\mathbf{p}}_i^{(t)} = \tilde{P}^{(t)} \dots \tilde{P}^{(1)} \mathbf{p}^{(0)}$ is the probability that the packet visits node i in the original model at or before time t, while $\alpha_b^{(t)} = \tilde{\mathbf{p}}_b^{(t)}$ is the probability that the packet arrives at the sink at or before time t through an alternate path disjoint with node i.

To find $\alpha_i = \lim_{t \to \infty} \tilde{\mathbf{p}}_i^{(t)}$, the probability the packet ever visits node i, we solve a system of equations for the probability that any state $j \neq i$ is absorbed into state i.

Theorem 2.2 (Absorption Probability Equations [15]): For a given Markov chain, choose an absorbing state i. Then, the

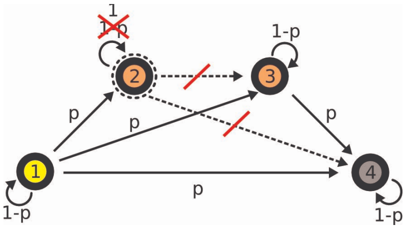


Fig. 3. Illustration of how to create absorbing states in the Markov chain to calculate the probability that a packet sent from node 1 to node 4 passes through node 2 by time t, using the routing topology of Figure 2.

probabilities α_j of reaching state i starting from j are the unique solution to the equations

$$\begin{aligned} \alpha_i &= 1 \\ \alpha_j &= 0 \quad \text{for all absorbing } j \neq i \\ \alpha_j &= \sum_{k=1}^{N} p_{jk}\alpha_k \quad \text{for all transient } j \end{aligned} \tag{5}$$

■

3) Link Perturbation: Sometimes, we only know that the probability of a link lies within an interval $(p_{ij}^{(t)} + \epsilon, p_{ij}^{(t)} - \epsilon)$, and estimate it as $p_{ij}^{(t)}$. Unfortunately, we *cannot* bound the range of $p_{net}^{(t_d)}$ by simply recomputing $p_{net}^{(t_d)}$ using the endpoints $p_{ij}^{(t)} + \epsilon$ and $p_{ij}^{(t)} - \epsilon$. This is argued carefully in [13], and is essentially because our routing scheme keeps one copy of the packet in the network and retransmits a packet when a link transmission fails, causing $p_{net}^{(t_d)}$ to be a polynomial function of the link probability.

The alternative is to try bounding the distance of the eigenvalues $\hat{\lambda}$ of the *actual* transition probability matrix $\hat{P}^{(\underline{T})}$ from the eigenvalues λ of our estimated transition probability matrix $P^{(\underline{T})}$, a standard problem in matrix perturbation analysis. In other words, if $\hat{\lambda}_x$ is an eigenvalue of $\hat{P}^{(\underline{T})} = P^{(\underline{T})} + \delta F, \delta \in (-\epsilon, +\epsilon)$ and F a matrix corresponding to the perturbed link ($F_{ii} = -\delta, F_{ji} = \delta$ and all other entries 0), then there is some eigenvalue λ_y of $P^{(\underline{T})}$ such that $|\hat{\lambda}_x - \lambda_y| < \mathcal{C}(F, \epsilon)$, where $\mathcal{C}(\cdot, \cdot)$ is some function of F and ϵ. There are several standard techniques to do this, some that require $P^{(\underline{T})}$ to be diagonalizable or $P^{(\underline{T})}$ to be normal ($A^*A = AA^*$), which may not always hold. These techniques are applicable on a case by case basis. For more details, see [16].

III. Directed Staged Flooding

We propose a simple constrained flooding scheme called *Directed Staged Flooding* (DSF) for one-to-many and one-to-one routing, focusing on the latter. Unlike UPD, DSF provides increased end-to-end connectivity with less latency by multicasting packets instead of using acknowledgments and retransmissions. We use a Directed Staged Flooding Markov Chain (DSFMC) model to find $p_{net}^{(t_d)}$. As with UPD, we build the model assuming we are provided with a routing schedule, the way nodes are grouped into stages (discussed below), and all the link probabilities. We leave the development of an algorithm to construct such a routing schedule for future work.

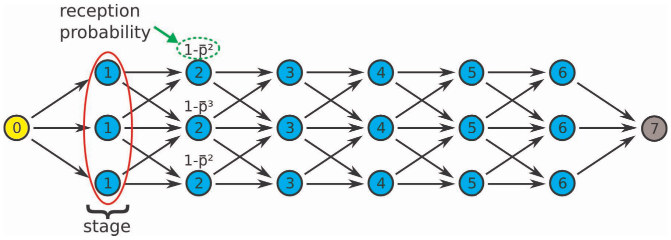


Fig. 4. Directed Staged Flooding example on a wide path topology containing stages with a path width of 3. Discussed in more detail in Section III-C.

A. Modeling Characteristics

Like UPD, DSF also assumes that the nodes follow a TDMA routing schedule. During a transmission each node transmits to a subset of its neighboring nodes. Furthermore, we group the nodes along the end-to-end transmission path such that a packet is modeled as being passed between groups of nodes, and we call each group of nodes a *stage*. Figure 4 illustrates this on a *wide path* topology between a source and destination where the nodes lie on a regular grid and each stage, except the first and last, consists of a column of 3 nodes.

DSF does not use acknowledgments to signal a node to retransmit a packet on a failed link. Thus, with careful scheduling consecutive packets will not queue in the network if there is only a single source transmitting to a single sink.

Our DSFMC model of DSF requires the *sets* of link transmissions between *distinct* pairs of stages to be independent. Like UPD, DSF uses frequency hopping over time to help justify this assumption. However, the model allows the link transmissions between the *same* pair of stages to be correlated. This mirrors reality because on any single multicast transmission, all the receiving nodes are listening on the same channel.

Our DSFMC model also assumes that all nodes in one stage transmit their copy of the packet before the nodes in the next stage transmit their copy of the packet. Furthermore, the transmissions of nodes within a stage will interfere with each other, so they must be scheduled in separate time slots. We make this assumption because most sensor network nodes have only one radio and can only listen to one channel at a time.

In DSF routing schedules, a node can be shared between multiple stages (See Figure 5). Like UPD, we assume that the links in the routing topology for DSF do not form a cycle. Complications arise when sharing nodes between stages because unlike flooding, staged flooding puts the constraints that a packet can only be transmitted from a node if it received the packet *prior* to the time another node in its stage first transmits. This is necessary for the DSFMC model developed below to hold. To enforce this condition, packets may carry with them a field indicating during which stage they were last transmitted. Last of all, we assume that if a node i is shared between stages k and $k+1$, then node i will retain the packet after transmission in stage k so it "receives the packet" with probability 1 in stage $k+1$.

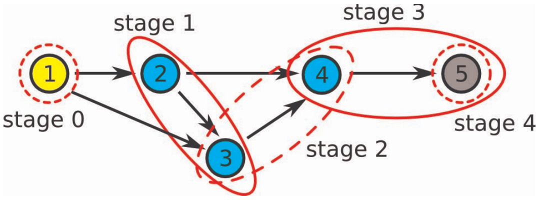


Fig. 5. Directed staged flooding example corresponding to (9).

B. Directed Staged Flooding Markov Chain Model

As before, we represent the routing topology as a graph $G = (\mathcal{V}, \mathcal{E})$ and denote a node in the network as $i \in \mathcal{V} = 1, \ldots, N$ and a link in the network as $l \in \mathcal{E} \subset \{(i,j) \,|\, i,j \in \mathcal{V}\}$, where $l = (i,j)$ is a link for transmitting packets from node i to node j. Because each link is used only once when transmitting a single packet, the link success probability for link $l = (i,j)$ is treated as being time-invariant and is denoted p_l, or p_{ij}.

Unlike the MTMC model, in the DSFMC model a state in the Markov chain at a stage represents the *set* of nodes in the stage that successfully received a copy of the packet. The transition probabilities between the states depend on the joint probability of successful link transmissions between stages. We state the DSFMC model for the special case where the links are all independent. For the general model, see [13].

Directed Staged Flooding Markov Chain Model Let's assume we have a routing topology with $K+1$ stages $0, \ldots, K$. Each stage k has N_k nodes, and the set of 2^{N_k} possible states in stage k is represented by the set of numbers $\mathcal{S}^{(k)} = \{0, \ldots, 2^{N_k} - 1\}$. Let $\mathcal{K}^{(k)}$ be the set of nodes in stage k and for each state $\sigma^{(k)} \in \mathcal{S}^{(k)}$, let $\mathcal{R}_\sigma^{(k)} \subset \mathcal{K}^{(k)}$ be the set of nodes that have received a copy of the packet and $\mathcal{U}_\sigma^{(k)} = \mathcal{K}^{(k)} \backslash \mathcal{R}_\sigma^{(k)}$ be the set of nodes that have not received a copy of the packet (See Figure 6). Let $\omega^{(k)}$ denote the state where no nodes received a copy of the packet in stage k.

The conditional probability of the next state $\mathbf{X}^{(k+1)}$ being state $\sigma^{(k+1)}$ given that the current state $\mathbf{X}^{(k)}$ is $\sigma^{(k)}$ can be expressed as

$$\mathbb{P}(\mathbf{X}^{(k+1)} = \sigma^{(k+1)} | \mathbf{X}^{(k)} = \omega^{(k)}) = \begin{cases} 1 & : \quad \sigma^{(k+1)} = \omega^{(k+1)} \\ 0 & : \quad \text{otherwise} \end{cases}$$

if $\sigma^{(k)} \neq \omega^{(k)}$

$$\mathbb{P}(\mathbf{X}^{(k+1)} = \sigma^{(k+1)} | \mathbf{X}^{(k)} = \sigma^{(k)}) = \left(\prod_{\substack{u \in \mathcal{U}_\sigma^{(k+1)} \\ i \in \mathcal{R}_\sigma^{(k)}}} (1 - p_{iu}) \right) \prod_{r \in \mathcal{R}_\sigma^{(k+1)}} \left(1 - \prod_{i \in \mathcal{R}_\sigma^{(k)}} (1 - p_{ir}) \right) \tag{6}$$

The transition probability matrices between stage k and $k+1$ are $P^{(k+1)} \in [0,1]^{N_{k+1} \times N_k}$, where the entry in position $(\sigma^{(k+1)}, \sigma^{(k)})$ of the matrix is $\mathbb{P}(\mathbf{X}^{(k+1)} = \sigma^{(k+1)} | \mathbf{X}^{(k)} = \sigma^{(k)})$.

The initial state $\mathbf{X}^{(0)}$ is the state $\sigma^{(0)}$ corresponding to $\mathcal{R}_\sigma^{(0)} = \{a\}$, where a is the node sending the initial packet. Then, the probability distribution $\mathbf{p}^{(k)} \in [0,1]^{N_k}$ of the state at stage k is

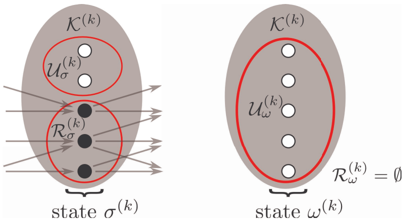

$\mathcal{K}^{(k)}$
$\mathcal{U}_\sigma^{(k)}$
$\mathcal{R}_\sigma^{(k)}$
state $\sigma^{(k)}$
$\mathcal{K}^{(k)}$
$\mathcal{U}_\omega^{(k)}$
$\mathcal{R}_\omega^{(k)} = \emptyset$
state $\omega^{(k)}$

Fig. 6. Mapping of states to nodes that received a packet in the DSFMC model. On the left is an example of a state $\sigma^{(k)}$ and on the right is the state $\omega^{(k)}$ where no packets have been received.

$$\mathbf{p}^{(k)} = \underbrace{P^{(k)} \cdots P^{(2)} P^{(1)}}_{P^{(\underline{k})}} \mathbf{p}^{(0)} \tag{7}$$

■

We can obtain the probability that a copy of the packet is at a node i at time t directly from our model by translating t to k from the relation $t = \sum_{i=0}^{k-1} N_i$ and looking at $\sum_{\{\sigma^{(k)} | i \in \mathcal{R}_\sigma^{(k)}\}} \mathbb{P}(\mathbf{p}^{(k)} = \sigma^{(k)})$. In the case where the last stage contains only the sink node and only the nodes in stage $K-1$ transmit to the sink, if b is the state in stage K where the sink receives a copy of the packet, we have

$$p_{net}^{(t_d)} = \begin{cases} 0 & : \quad t_d \leq \sum_{i=0}^{K-2} N_i \\ \mathbf{p}_b^{(K)} & : \quad t_d \geq \sum_{i=0}^{K-1} N_i \end{cases} \tag{8}$$

and $0 \leq p_{net}^{(t_d)} \leq \mathbf{p}_b^{(K)}$ when $\sum_{i=0}^{K-2} N_i < t_d < \sum_{i=0}^{K-1} N_i$.

Finally, note that except in the special case where there exists a path from the source to the destination with all link probabilities equal to 1, $p_{net}^{(t_d)} < 1$ for all t_d. All copies of a packet can be lost in the network because we do not use acknowledgments and retransmissions to guarantee a copy of the packet has been delivered.

C. DSFMC Examples and Discussion

As an example, let's consider the stages with path width 3 in Figure 4. Assume the links are independent, that each link has the same transmission success probability p, and let $\bar{p} = 1 - p$. Then, the probability that a node in stage $k+1$ receives a copy of the packet, given the state of stage k, is 1 minus the product of incoming link failure probabilities, as shown in Figure 7. The transition probability between states can be obtained by applying (6). Figure 8 illustrates the transitions out of state 7. The full 8×8 transition matrix is omitted here because of space limitations.

In the example in Figure 5, the dimensions of the state probability distribution vector vary with time, and also some of the nodes are shared between stages. To represent the state at each stage k, we first order the nodes in each stage from smallest to largest node id and re-index them from $0, \ldots, N_k - 1$. Then, for each node with a new index n we set $i_n = 1$ if the node has a copy of the packet and $i_n = 0$ otherwise. The state is then just $\sigma^{(k)} = \sum_{n=0}^{N_k - 1} i_n 2^n$. Assuming the links are

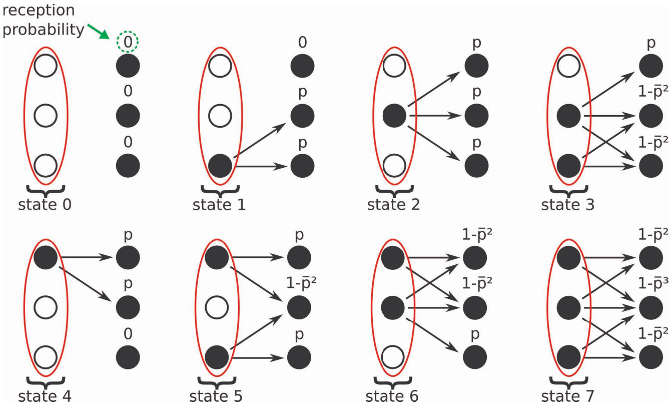


Fig. 7. Markov chain states for the routing topology in Figure 4, excluding the states for the source and the destination.

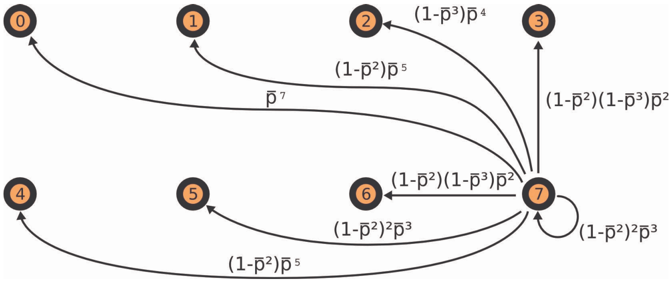

0
1
2
3
4
5
6
7
$(1-\bar{p}^3)\bar{p}^4$
$(1-\bar{p}^2)\bar{p}^5$
$\bar{p}^7$
$(1-\bar{p}^2)(1-\bar{p}^3)\bar{p}^2$
$(1-\bar{p}^2)(1-\bar{p}^3)\bar{p}^2$
$(1-\bar{p}^2)^2\bar{p}^3$
$(1-\bar{p}^2)^2\bar{p}^3$
$(1-\bar{p}^2)\bar{p}^5$

Fig. 8. Markov chain transition diagram for a stage of path width 3 in the routing topology in Figure 4. Here, only the outgoing transitions and associated transition probabilities from state 7 are shown.

independent, the equations that describe the DSFMC model are

$$
\begin{gathered}
P^{(1)}=\begin{bmatrix}1 & \bar{p}_{12}\bar{p}_{13}\\ 0 & p_{12}\bar{p}_{13}\\ 0 & \bar{p}_{12}p_{13}\\ 0 & p_{12}p_{13}\end{bmatrix}\quad P^{(2)}=\begin{bmatrix}1 & \bar{p}_{23}\bar{p}_{24} & 0 & 0\\ 0 & p_{23}\bar{p}_{24} & \bar{p}_{34} & \bar{p}_{24}\bar{p}_{34}\\ 0 & \bar{p}_{23}p_{24} & 0 & 0\\ 0 & p_{23}p_{24} & p_{34} & (1-\bar{p}_{24}\bar{p}_{34})\end{bmatrix}\\
P^{(3)}=\begin{bmatrix}1 & \bar{p}_{34} & 0 & 0\\ 0 & p_{34} & \bar{p}_{45} & \bar{p}_{45}\\ 0 & 0 & 0 & 0\\ 0 & 0 & p_{45} & p_{45}\end{bmatrix}\quad P^{(4)}=\begin{bmatrix}1 & \bar{p}_{45} & 0 & 0\\ 0 & p_{45} & 1 & 1\end{bmatrix}\\
\mathbf{p}^{(0)}=\begin{bmatrix}1 & 0\end{bmatrix}^T\quad P^{(\underline{4})}=P^{(4)}P^{(3)}P^{(2)}P^{(1)}
\end{gathered}\tag{9}
$$

where p_{ij} is indexed by the original node ids and again $\bar{p}_{ij}=1-p_{ij}$. As mentioned in Section III-A, we assume that if a node i in stage k has a copy of the packet and node i is also in stage $k+1$, then node i will have a copy of the packet in stage $k+1$ with probability 1.

D. DSFMC Analysis

1) $p_{net}^{(t_d)}$ for Wide Paths with Repeated Stages: For the purposes of choosing a network topology before deployment, it is useful to get a grasp of how $p_{net}^{(t_d)}$ scales as we extend the length K of a wide path topology without having to calculate $p_{net}^{(t_d)}$ for each new network explicitly. We consider the case of a wide path with repeated stages containing a constant number of nodes N_{stage} per stage and the same transition probability matrix $P^{(k)}=P$ between all stages, like the middle stages in the example in Figure 4. For simplicity, the discussion below will ignore the first stage containing the source and the last stage containing the destination.

A good characterization of how end-to-end connectivity scales with the number of stages K comes from the eigenvalues of P.

Theorem 3.1 (DSFMC $p_{net}^{(t_d)}$ converges exponentially to 0): Let P be diagonalizable and $\lim_{K\to\infty}P^K\mathbf{p}^{(0)}=\mathbf{e}^{[\omega]}$, where ω is the state where *no* nodes received a copy of the packet. Then

$$
p_{net}^{(t_d)}\le C(\rho_*)^K,\quad t_d=KN_{stage} \tag{10}
$$

for some constant C dependent on the initial distribution $\mathbf{p}^{(0)}$ and $\rho_*=\max\{|\lambda|:\lambda\text{ is an eigenvalue of }P\text{ and }|\lambda|<1\}$. ■ While this relation is an upper bound, ρ_* is the dominant decay rate for large K because all the eigenvectors of P with eigenvalue magnitudes less than 1 decay exponentially with K. In practice, a good routing topology has ρ_* very close to 1. When choosing a routing topology for wide paths, one can use ρ_* for wide paths with repeated stages of different widths to quickly compare the gain in reliability at the cost of extra latency.

2) Traffic Distribution: To calculate the probability that a copy of the packet visits a node i at or before time t, $\alpha_i^{(t)}$, we first remove all the outgoing edges of i, and add a "self transmission" link of probability 1 from node i to itself over all time slots. Then, we compute $\alpha_i^{(t)}=\sum_{\{\sigma^{(k)}|i\in\mathcal{R}_\sigma^{(k)}\}}\mathbb{P}(\tilde{\mathbf{p}}^{(k)}=\sigma^{(k)})$, where $\tilde{\mathbf{p}}^{(\mathbf{k})}$ is the state probability distribution on the modified routing schedule and topology.

3) Link Perturbation in Topology with Independent Links: In the DSFMC model where no nodes are shared between stages, it turns out that to compute the sensitivity of $p_{net}^{(t_d)}$ to errors in estimating a link probability p_l, we can compute bounds on the actual end-to-end connectivity $\hat{p}_{net}^{(t_d)}$ using the maximum and minimum possible values of the real link probability $\hat{p}_l$, $p_l+\epsilon$ and $p_l-\epsilon$ respectively. This is because unlike UPD, there are no link retransmissions in DSF, which implies that $p_{net}^{(t_d)}$ is a linear function of the single-link estimation error δ.

To see this, note that in (6), the transition probability between states in adjacent stages are a linear function of the individual link probabilities (the probability associated with a link appears in the expression once). This means that the transition matrices $\hat{P}^{(k)}$ are a linear function of each link probability p_l. Also, each link probability p_l appears in only one matrix $\hat{P}^{(k)}$ because each link is used only once to transmit a packet. This is because there are no retransmissions in the network and no nodes are shared between stages, so no node will transmit more than once when routing a single packet through the network. As a result, $\hat{P}^{(\underline{K})}$ is also a linear function of p_l. Finally, $p_{net}^{(t_d)}$ is a linear function of $\hat{P}^{(\underline{K})}$ and hence also a linear function of p_l, meaning it is a linear function of δ.

IV. UPD and DSF Comparisons

We wish to compare UPD and DSF using end-to-end connectivity as a function of latency, $p_{net}^{(t_d)}$, as the primary metric. This is effectively a comparison to see when retransmissions in UPD is better than "preemptive retransmissions" by multicast

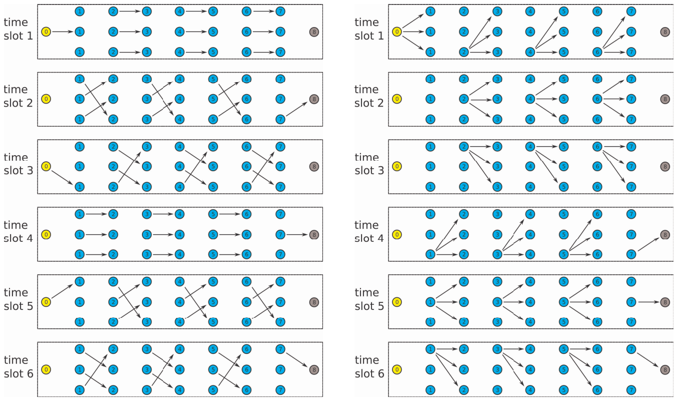


Fig. 9. (left) UPD and (right) DSF schedules for routing on a grid of width 3, used in the calculations for the graphs in Section IV.

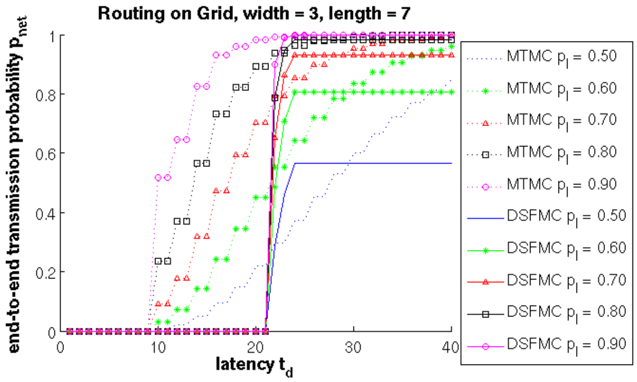


Fig. 10. End-to-end connectivity as a function of latency for varying link probabilities using the routing schedules described in Figure 9.

in DSF. We chose the example of routing on a wide path grid, where the width of the path is the number of rows and the length of the path is the number of columns. Here, every node in one column of a grid (a stage in DSF) can route to every other node in the next column with equal, independent link probabilities p_l. The routing schedule for Directed Staged Flooding and Unicast Path Diversity routing is described in Figure 9 for paths of width 3. Also, for all our comparisons, we assume that the time to send an acknowledgment for UPD is negligible and can be sent back in the same time slot as the original transmission.

A. End-to-end Connectivity Comparisons

Figure 10 compares $p_{net}^{(t_d)}$ of the two routing schemes under a range of different link probabilities.[3] UPD has the potential to deliver packets from the source to the sink in a shorter period of time, but the packet delivery time has a larger variance. Also, because $\lim_{t\to\infty} p_{net}^{(t_d)} = 1$ for UPD and p_{net} for DSF is a fixed value less than 1 after the last stage transmits (assuming $p_l \neq 1$), UPD can always provide better end-to-end connectivity at high latencies t_d.

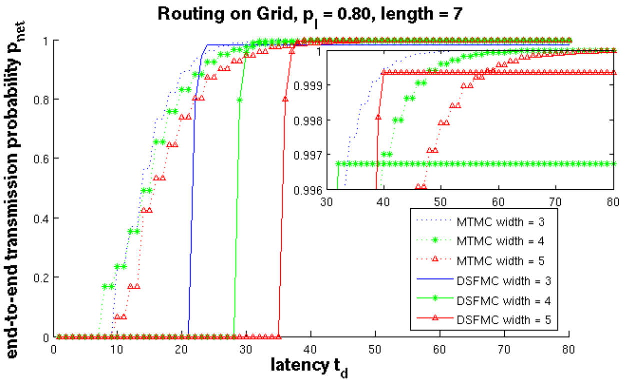


Fig. 11. End-to-end connectivity as a function of latency for varying path widths using the routing schedules described in Figure 9, with magnification of the plot for p_{net} near 1.

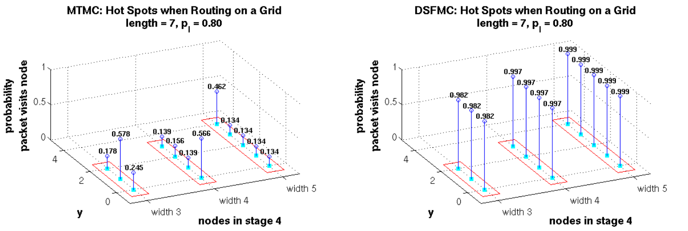


Fig. 12. Traffic distribution of nodes in the middle stage $\mathcal{K}$ of routing topologies of varying widths. Note that due to errors in rounding, the probabilities for the middle stages in MTMC may not add exactly to 1. These graphs use the routing schedules described in Figure 9.

Figure 11 shows that the final end-to-end connectivity p_{net} for DSF is higher for wider paths at the cost of larger latency. Also, the figure illustrates the limitations of our MTMC model—the model is unable to capture the benefit of diversity from using multiple paths instead of a single path because it assumes that all links are independent. Hence, retransmission on the same link is just as good as transmitting on a different link. What *is* modeled is that wider paths require more time slots to schedule transmissions from the nodes in the last column to the sink, and hence Figure 11 shows that UPD on wide paths with a smaller width tend to perform better.

B. Robustness Comparisons

We computed the traffic distribution α on the group of nodes $\mathcal{K}$ in the middle column of our routing grid, assuming the packet is sent during the first time slot. The traffic distribution for UPD is highly dependent on the schedule and link probabilities, where lower link probabilities generally tend to spread the traffic throughout the network. Figure 12 shows that even with the simple, regular schedule shown in Figure 9 and a relatively low link probability $p_l = 0.8$, UPD does not distribute the packets completely evenly over the nodes in $\mathcal{K}$. Finding an optimal schedule for spreading traffic in UPD is an interesting area for future research. On the other hand, DSF tends to spread copies of the packet over the nodes in $\mathcal{K}$ better than UPD for all path widths because it multicasts the packets.

In the same sense, DSF is less sensitive than UPD to link estimation error under short latencies because it multicasts

[3] Note that in this and subsequent plots, we perform the DSFMC calculations at the time granularity of time slots, not stages, unlike the description of (8) in Section III-B.

packets and thus tends to spread packets over more paths. Given the routing schedule in Figure 9, with $p_l = 0.8$ and a link estimation error $\delta = -0.1$ on a link in the middle of the network, the *maximum* error in end-to-end link probability estimates, Δp_{net}, over all latencies is on the order of 0.01 for DSF and on the order of 10^{-5} for UPD. Of course, because $\lim_{t \to \infty} p_{net}^{(t_d)} = 1$ in UPD for all routing schedules with one sink, $\Delta p_{net}^{(t_d)}$ for UPD is less than that of DSF for large t_d.

C. Communication Tradeoffs for Control Systems

In Section I-A we mentioned that when designing a control application, it may be reasonable to impose a delivery deadline and drop the packet if it takes too long to arrive. In UPD routing, if a packet arrives at a node that has a queued old packet, we can either combine the data in the two packets into one packet or we can drop the older packet and send only the newer packet. The implications of these two schemes is studied in [4].

Using the graphs in Figure 10 and some simple calculations, we can check the feasibility of running a control application on an IEEE 802.15.4 wireless network running UPD or DSF using the routing schedules in Figure 9. Assume we have a width 3 path from the controller to the actuators, and a width 3 path from the sensors back to the controller, and all the links have a transmission success probability $p_l = 0.8$. Then after 24 time slots we can get end-to-end transmission probability $p_{net} > 0.95$ between the controller and actuators and between the sensors and controller, both of which are separated by 8 hops. In Dust Network's UPD over TSMP 1.0, there are 32 slots a second, which corresponds to ≈ 1.5 seconds round trip time. This round trip time can be decreased in future versions of UPD because the theoretical limit of an 802.15.4 radio is $\frac{250\,\text{kbps}}{(6+25+10\,\text{Bytes/pkt})(8\,\text{bits/Byte})} \approx 762\,\text{pkts/sec}$ (10 Byte payload, 25 Byte MAC header and CRC, 6 Byte PHY header), resulting in a round trip time of $\approx 63\,\text{ms}$. Therefore, the types of control applications that we can hope to run on wireless sensor networks spanning 8 hops would have to tolerate round trip latencies on the order of magnitude of tenths of a second under optimal conditions, and seconds if we use current routing algorithms.

V. Conclusions

In this paper, we developed Markov chain models for UPD and DSF routing algorithms that can be used for planning a new network deployment. In order to construct these models for existing networks, the user must have full knowledge of the estimated link probabilities and routing schedule in the network. One possibility is to have a network periodically route back the routing schedule and link probability estimates of all the links in the network. The assumption is that the time scale over which the link probabilities change is much larger than the time scale for sending a packet through the network with high probability.

In wireless networked control systems, if we can calculate $p_{net}^{(t_d)}$ of the network in real-time, we can tune the controller/switch controllers based on the conditions of the network. For instance, in manufacturing we can use an aggressive controller for higher yield when the wireless network is good and a less aggressive controller that does not compromise safety and the quality of the products when the network is bad. We will study in detail the issues of running controllers over mesh wireless networks in an upcoming paper.

Acknowledgment

The authors would like to thank Songhwai Oh, Ian Tan, and David Tse for offering feedback on the ideas in this paper, and Kris Pister for providing more details about UPD over TSMP and 802.15.4 radios.

References

[1] D. Culler, D. Estrin, and M. Srivastava, "Overview of sensor networks," in *IEEE Computer, Special Issue in Sensor Networks*, August 2004.

[2] Industrial Standards and Automation Committee, "ISA-SP100 wireless systems for automation," http://www.isa.org, 2007.

[3] *Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*, LAN/MAN Standards Committee of the IEEE Computer Society, 3 Park Avenue, New York, NY 10016-5997, USA, October 2003, 802.15.4 Standard.

[4] L. Schenato, "Optimal estimation in networked control systems subject to random delay and packet loss," in *Proc. of the 45th IEEE Conference on Decision and Control*, December 2006.

[5] 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.

[6] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proceedings of the IEEE*, vol. 95, pp. 138–162, 2007.

[7] C. Karlof, Y. Li, and J. Polastre, "ARRIVE: Algorithm for robust routing in volatile environments," University of California at Berkeley, Tech. Rep. UCB/CSD-03-1233, May 2002.

[8] S. Dulman, T. Nieberg, J. Wu, and P. Havinga, "Trade-off between traffic overhead and reliability in multipath routing for wireless sensor networks," in *Proceedings of the Wireless Communications and Networking Conference*, 2003.

[9] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-resilient, energy-efficient multipath routing in wireless sensor networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 5, no. 4, pp. 11–25, 2001.

[10] Dust Networks, Inc., "Technical overview of time synchronized mesh protocol (TSMP)," http://www.dustnetworks.com/docs/TSMP_Whitepaper.pdf, 2006.

[11] A. Nasipuri, R. Castañeda, and S. R. Das, "Performance of multipath routing for on-demand protocols in mobile ad hoc networks," *Mob. Netw. Appl.*, vol. 6, no. 4, pp. 339–349, 2001.

[12] Y. Ganjali and A. Keshavarzian, "Load balancing in ad hoc networks: Single-path routing vs. multi-path routing," in *INFOCOM, 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2, 2004, pp. 1120–1125.

[13] P. Chen and S. S. Sastry, "Latency and connectivity analysis tools for wireless mesh networks," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2007-87, Jun 2007. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2007/EECS-2007-87.html

[14] Dust Networks, Inc., *SmartMesh-XT M2030 Product Specification*, http://www.dustnetworks.com/docs/M2030.pdf, 2006, datasheet.

[15] D. P. Bertsekas and J. N. Tsitsiklis, *Introduction to Probability*. Belmont, Massachusetts: Athena Scientific, 2002.

[16] R. A. Horn and C. R. Johnson, *Matrix Analysis*. New York: Cambridge University Press, 1999.