Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008



# Performance analysis of different routing protocols in wireless sensor networks for real-time estimation

Damiano Varagnolo, Phoebus Chen, Luca Schenato, Shankar S. Sastry

***Abstract*— In this paper we analyze the performance of two different routing protocols specifically designed for Wireless Sensor Networks (WSNs) for real-time estimation, control, and monitoring. These protocols are designed to compensate for the lossy nature of the wireless links and the delay from sending messages over multiple hops from the sensors to the controller. The routing protocols are designed to reduce packet delay and packet loss using either retransmissions or multicasting. For some routing topologies one protocol may be better than the other at reducing the worst case packet delay but may have a worse packet loss rate. Here, we apply mathematical tools to analytically compute the average real-time performance based on end-to-end packet delay statistics for two recently proposed routing strategies. We show that the performance is strongly related to the dynamics of the systems being estimated, and we construct a computationally efficient estimation strategy based on the delay statistics. This suggests that routing protocols are to be designed based on the specific real-time estimation and control application under consideration.**

***Index Terms*— packet drop, random delay, remote estimation, wireless sensor networks, routing, multipath**

## I. INTRODUCTION

Wireless Sensor Networks (WSNs) can be employed for real-time estimation, control and monitoring applications. However, they suffer from the usual problems in wireless communications, such as time-varying channels and large packet loss probabilities. Moreover, these problems are exacerbated by the need to multi-hop messages through intermediate nodes to communicate with far away nodes. As a consequence, multi-hopping potentially increases the end-to-end packet loss rate and induces varying delay due to retransmission, multiple path routing, and out-of-order packet arrival. These problems pose two main challenges: how to design routing protocols which give rise to low end-to-end packet loss and small delay (latency), and how to design real-time estimation algorithms which can cope with random delay and packet loss. In the following, we briefly review the most relevant literature in these two areas and our contribution.

This work was supported in part by the Italian CaRiPaRo Foundation, by EU PF7 project “FeedNetBack”, and by TRUST (Team for Research in Ubiquitous Secure Technology), which receives support from the National Science Foundation (NSF award number CCF-0424422) and the following organizations: AFOSR (#FA9550-06-1-0244), BT, Cisco, ESCHER, HP, IBM, iCAST, Intel, Microsoft, ORNL, Pirelli, Qualcomm, Sun, Symantec, Telecom Italia, and United Technologies.

L. Schenato and D. Varagnolo are with the Department of Information Engineering, University of Padova, Via Gradenigo, 6/b, Padova, Italy {schenato|damiano.varagnolo}@dei.unipd.it

P. Chen and S. Sastry are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, USA. {phoebusc|sastry}@eecs.berkeley.edu

### A. WSN Routing for Estimation and Control

WSN routing protocols developed for estimation and control need to provide good reliability with low latency. As a result, many protocols like to use a TDMA scheduling scheme, as opposed to CSMA schemes with randomized contention schemes to access the network. For instance, both Time Synchronized Mesh Protocol (TSMP) [1] and RT-Link [2] schedule link transmissions across the network to bound the end-to-end latency of a packet. However, RT-Link is less robust to link failure than TSMP because it is a single-path routing protocol while TSMP is a mesh routing protocol.

Other WSN protocols designed for industrial control (ex. SERAN [3], Breath [4], DSF [5]) use cluster-based routing to get higher reliability. This routing is a form of constrained flooding, where copies of a packet are passed between groups of nodes. SERAN and Breath assume the independence of links, node wake up times, and random attempts to access the channel so that the Central Limit Theorem can be employed to get probabilistic guarantees on end-to-end delivery and reliability. DSF assumes the links are mutually independent and uses individual link probabilities to compute end-to-end connectivity as a function of latency.

### B. Estimation and control subject to random delay and packet loss

Recently, there has been a considerable effort in analyzing and designing estimation and control schemes in networked control systems (NCS) subject to packet loss and packet delay, as surveyed in [6] and [7]. Most of the results are concerned with finding stability conditions for filtering and control, and in general very few results provide a quantitative measure of performance based on packet delay and loss statistics. Among these, in [8] upper and lower bounds are provided for the optimal mean square estimator in systems subject to packet loss but not to packet delay. In [9] Nilsson et al. considered designing an optimal LQG regulator when packets are subject to random packet delay with known statistics, but not to packet loss. Recently, in [10] we proposed different estimation algorithms with quantifiable performance under known and i.i.d. packet delay statistics.

Another important related area of research addresses the problem of finding numerically efficient algorithms to compute the optimal mean square estimator subject to delayed measurements, as in [11] and [12]. These are general algorithms which require little memory and are also valid for time-varying dynamics and out-of-order packet arrival. However, they do not provide performance evaluation tools based on packet delay statistics, which is of primary concern in our work.

### C. Contribution

In this paper, we study the performance (in the sense of estimation error covariance) of real-time filtering running over two of the most promising routing protocols: Directed Staged Flooding (DSF) and Unicast Path Diversity (UPD), which is a specific implementation of a protocol based on TSMP [5]. In particular, we show how to derive the end-to-end packet loss latency and connectivity statistics in terms of $\lambda_h^{ab}$, which is the probability that a packet sent from sensor $a$ is delivered to sensor $b$ with a delay $\tau$ not greater than $h$ (i.e. $\lambda_h^{ab} = \mathbb{P}[\tau \leq h]$). These statistics are used off-line to compute the performance of a Kalman-like estimator with a buffer of dimension $N$ storing the various measurements. These types of filters have been proposed in [10], and here we extend them to consider a shifted buffer, i.e. only measurements with a delay between $M$ and $M+N$, where $M$ is the buffer shift. Through numerical simulations, we show that there is a trade off between performance, computational complexity and system dynamics, which might lead to regimes where one routing protocol is better than the other and regimes where the opposite occurs. This implies that protocols must be chosen with the specific application in mind.

## II. Networking protocols for WSN: UPD and DSF

This section provides brief descriptions and Markov Chain models of two mesh routing protocols designed to provide high reliability for industrial control applications. For more details and examples, see [5].

### A. Unicast Path Diversity

Dust Networks, Inc. proposed Unicast Path Diversity (UPD) over Time Synchronized Mesh Protocol (TSMP) [1], which exploits frequency, time, and space diversity for reliable networking in sensor networks. UPD is a many-to-one, multi-path routing protocol where each node in the network has multiple parents and the routing graph has no cycles. The links selected for routing are bidirectional, hence every link transmission can be acknowledged. If a packet transmission is not acknowledged, it is queued in the node for retransmission. To schedule the network, time is divided into time slots, and grouped into superframes. 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.

In order to calculate $\lambda_h^{ab}$, we construct a general Mesh TDMA Markov Chain (MTMC) model for UPD that assumes knowledge of the routing topology, schedule, and all the link probabilities. MTMC models single packet transmission in the network without the effects of queuing.

*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 $i \in \mathcal{V} = 1, \dots, N$, 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 $h$ will be measured in units of time slots, and let $H$ denote the number of time slots in a superframe. The link success probability for link $l = (i,j)$ at time slot $h$ is denoted $p_l^{(h)}$, or $p_{ij}^{(h)}$. We set $p_l^{(h)} = 0$ when link $l$ is not scheduled to transmit at time $h$. It is possible to construct the following time-varying, discrete-time Markov Chain:

*Definition 1 (MTMC 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 $h$ is simply $p_{ij}^{(h)}$, with $p_{ii}^{(h)} = 1 - \sum_{j \neq i} p_{ij}^{(h)}$. Let $\mathrm{P}^{(h)} = [p_{ij}^{(h)}]^T \in [0,1]^{N \times N}$ be the column stochastic transition probability matrix for a time slot and $\mathrm{P}^{(\underline{H})} := \mathrm{P}^{(H)}\mathrm{P}^{(H-1)} \dots \mathrm{P}^{(1)}$ be the transition probability matrix for a repeating superframe. Let a packet originated at node $a$ be represented by $\mathbf{p}^{(0)} := \mathbf{e}^{[a]}$, where $\mathbf{e}^{[a]}$ is an elementary vector with the $a$-th element equal to 1 and all other elements equal to 0. The probability distribution of the state at time $h$ given the initial condition $\mathbf{e}^{[a]}$ is given by $\mathbf{p}^{(h)} = \left(\prod_{t=1}^{h} \mathrm{P}^{(t)}\right) \mathbf{p}^{(0)}$. With this definition the probability that a packet is delivered from $a$ to $b$ with a delay not greater than $h$ is $\lambda_h^{ab} = \mathbf{p}_b^{(h)}$ (the $b$-th element of $\mathbf{p}^{(h)}$). ■

### B. Directed Staged Flooding

Directed Staged Flooding (DSF) uses simple constrained flooding for one-to-many or one-to-one routing. *Unlike* UPD, DSF provides increased end-to-end connectivity with less latency by multicasting packets instead of using acknowledgments and retransmissions. *Like* UPD, DSF 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*. For instance, looking at the node topology for one time slot on the right of Fig. 1, each column of nodes is a stage. For simplicity, here we assume the nodes are not shared between stages, although this is not required (see [5]).

We use a Directed Staged Flooding Markov Chain (DSFMC) model to calculate $\lambda_h^{ab}$, assuming the routing schedule, the stage grouping, and all the link probabilities are known. The model 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.

*Directed Staged Flooding Markov Chain Model:* The notation for the routing topology is the same as in the MTMC model. Note that here the link success probability for link $l = (i,j)$ is treated as being time-invariant and is denoted $p_l$ (or $p_{ij}$), since each link is used only once when transmitting a single packet.

It is possible to construct a time-invariant, discrete-time Markov Chain where the state at a stage represents the *set* of nodes that successfully received a copy of the packet. The

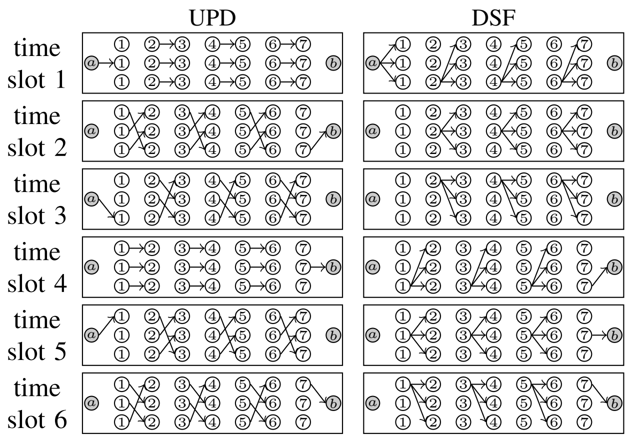


Fig. 1. (left) UPD and (right) DSF schedules for routing on a grid of width 3, used in the calculations for the graph in Fig. 2.

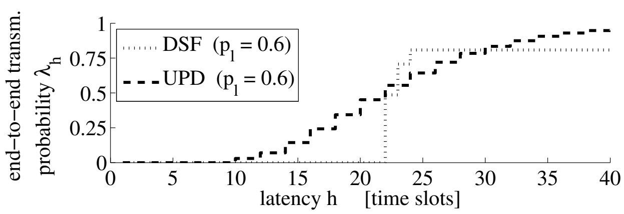


Fig. 2. *Calculated* end-to-end connectivity $\lambda_h^{ab}$ as a function of latency $h$ using the routing schedules described in Fig. 1, where all the links have probability $p_l = 0.6$, $a = 0$ and $b = 8$.

transition probabilities between the states depend on the joint probability of successful link transmissions between stages. When the links are all independent, the model is:

*Definition 2 (DSFMC Model):* Let's assume we have a routing topology with $K+1$ stages $0, \dots, K$. 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 $\mathcal{S}^{(k)} = \{0, \dots, 2^{N_k} - 1\}$. 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 Fig. 3). Let $\omega^{(k)}$ denote the state where no nodes received a copy of the packet in stage $k$. The conditional probability that the next state $\mathbf{X}^{(k+1)}$ equals $\sigma^{(k+1)}$ given that the current state $\mathbf{X}^{(k)}$ equals $\omega^{(k)}$ is 0 if $\sigma^{(k+1)} \neq \omega^{(k)}$, and 1 if $\sigma^{(k+1)} = \omega^{(k)}$. If $\sigma^{(k)} \neq \omega^{(k)}$:

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

The transition probability matrices between stage $k$ and $k+1$ are $\mathrm{P}^{(k+1)} \in [0,1]^{N_{k+1} \times N_k}$, 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 $\mathbf{p}^{(k)} = \left( \prod_{t=1}^{k} \mathrm{P}^{(t)} \right) \mathbf{p}^{(0)}$. Assume the transmissions of nodes within a stage must be scheduled in separate time slots so they do not interfere with each other, so time $h$ is related to stage $k$ by $h = \sum_{j=0}^{k-1} N_j$. Then, $\lambda_h^{ab} = \sum_{\{\sigma^{(k)} | b \in \mathcal{R}_\sigma^{(k)}\}} \mathbb{P}[\mathbf{X}^{(k)} = \sigma^{(k)}]$, a summation over the corresponding elements in the vector $\mathbf{p}^{(k)}$. ■

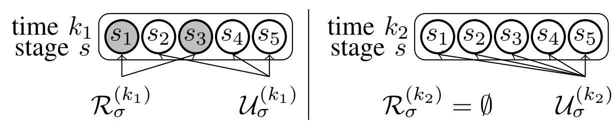


Fig. 3. Mapping of states to nodes that received a packet in the DSFMC model. A state $\sigma^{(k)}$ of a stage is in correspondence with the set of nodes $\mathcal{R}_\sigma^{(k)}$ that correctly received the packet (greyed circles).

### C. UPD and DSF Comparison

UPD can deliver packets from $a$ to $b$ in a shorter time than DSF, but with a larger variance. Also $\lim_{h \to \infty} \lambda_h^{ab} = 1$ for UPD, while $\lambda_h^{ab} \leq 1$ for DSF after the last stage transmits (assuming $p_l \neq 1$). This imply that UPD can always provide better end-to-end connectivity at high latencies $h$. DSF tends to perform better when there are a few very poor links scattered throughout the network. Fig. 2 compares $\lambda_h^{ab}$ for various $h$ for UPD and DSF under the schedules in Fig. 1, assuming $p_l = 0.6 \ \ \forall l \in \mathcal{V}$.

### D. Usage of UPD and DSF for estimation

Both protocols can be used for estimation purposes. Assume that nodes $a = 0$ and $b = 8$ in Fig.2 are respectively a sensor and an estimator; $a$ collects data and then sends packets towards $b$ through the UPD or DSF network. How should $b$ handle packet loss and delay to have some kind of optimal estimate? Which protocol behaves better and under what conditions? In the next sections we will formally state the problem assuming that $a$ measures the following discrete time linear stochastic plant:

$$x_{t+1} = Ax_t + w_t \qquad\qquad y_t = Cx_t + v_t \tag{1}$$

where $t \in \mathbb{N}$, $x, w \in \mathbb{R}^n$, $A \in \mathbb{R}^{n \times n}$, $y, v \in \mathbb{R}^m$, $C \in \mathbb{R}^{m \times n}$, $(x_0, w_t, v_t)$ are Gaussian, uncorrelated, white, with mean $(\bar{x}_0, 0, 0)$ and covariance $(P_0, Q, R)$ respectively (note $P \neq \mathrm{P}$ from Defs. 1 and 2). We also assume that the pair $(A, C)$ is observable, $(A, Q^{1/2})$ is reachable, and $R > 0$.

Note that measurements are time-stamped, encapsulated into packets, and then transmitted through the digital communication network. Time-stamping of measurements is necessary to reorder packets at the receiver side since they can arrive out of order.

## III. Minimum variance estimators subject to packet loss and delay

The packet arrival process can be modeled via the random variable $\gamma_k^t$, defined as follows ($k$ is transmit time):

$$\gamma_k^t = \begin{cases} 1 & \text{if } y_k \text{ is received at or before time } t, \ t \geq k \\ 0 & \text{otherwise} \end{cases}$$

We also define the packet delay $\tau_k \in \{\mathbb{N}, \infty\}$ for observation $y_k$ as follows:

$$\tau_k = \begin{cases} \infty & \text{if } \gamma_k^t = 0, \forall t \geq k \\ t_k - k & \text{otherwise } (t_k := \min\{t | \gamma_k^t = 1\}) \end{cases} \tag{2}$$

where $t_k$ is the arrival time of observation $y_k$ at the estimator. If the delay of the arriving packets is bounded, i.e. if there exists $\bar{N}$ such that $\gamma_k^t = \gamma_k^{t+1}$ for $t-k+1 \geq \bar{N}$, then it has been shown in [10] that the minimum variance estimator $\hat{x}_{t|t}^t = \mathbb{E}[x_t \,|\, \text{arrived measurements}] = \mathbb{E}[x_t \,|\, \gamma_1^t, .., \gamma_t^t, \tilde{y}_1^t, .., \tilde{y}_t^t]$ (where $\tilde{y}_k^t = \gamma_k^t y_k$) and its corresponding prediction error covariance $P_{t+1|t}^t = \mathbb{E}[(x_{t+1} - A\hat{x}_{t|t}^t)(x_{t+1} - A\hat{x}_{t|t}^t)^T \,|\, \gamma_1^t, \dots, \gamma_t^t]$ is given by a time-varying Kalman Filter with a buffer of size $\bar{N}$ whose equations are:

$$
\begin{aligned}
\hat{x}_{t-\bar{N}|t-\bar{N}}^t &= \hat{x}_{t-\bar{N}|t-\bar{N}}^{t-1}, \quad P_{t-\bar{N}+1|t-\bar{N}}^t = P_{t-\bar{N}+1|t-\bar{N}}^{t-1} \\
\hat{x}_{k|k}^t &= A\hat{x}_{k-1|k-1}^t + \gamma_k^t K_k^t (\tilde{y}_k^t - CA\hat{x}_{k-1|k-1}^t) \\
K_k^t &= P_{k|k-1}^t C^T (C P_{k|k-1}^t C^T + R)^{-1} \\
P_{k+1|k}^t &= A P_{k|k-1}^t A^T + Q - \gamma_k^t A K_k^t C P_{k|k-1}^t A^T
\end{aligned}
$$

where $k = t - \bar{N} + 1, \dots, t$, and $\hat{x}_{h|h}^t = \bar{x}_0, P_{h|h-1}^t = P_0$ for $h \leq 0$. Because the error covariance $P_{t+1|t}^t$ depends on the packet arrival sequence $\gamma_k^t$, it is time-varying and does not converge to a steady state, unlike the standard Kalman Filter with no packet loss. Moreover, it requires the inversion of up to $\bar{N}$ matrices at every time step $t$ and might be too expensive for on-line implementation. Also, the buffer size $\bar{N}$ needed for the optimal estimator might be too large. Although in theory even very old measurements help reduce the estimation error, in practice their contribution is marginal. In Sec. IV we propose a new strategy requiring no matrix inversion and whose buffer size can be reduced to trade off performance with computational complexity.

## IV. Estimation with shifted buffer and constant gains

In this section we propose a suboptimal estimator design strategy which does not require any matrix inversion and has a buffer with length smaller than the WSN maximum packet delay $\bar{N}$. Since we want to quantify the performance of the estimator, we need to specify the statistics of the packet arrival process from sensor $a$ to estimator $b$. We assume it to be stationary and i.i.d. with probability function (in the following we will omit the superscripts): $\lambda_h := \lambda_h^{ab} = \mathbb{P}[\tau_t \leq h]$, where $t \geq 0$, $0 \leq \lambda_h \leq 1$ is non-decreasing in $h = 0, 1, \dots, \bar{N}$, and $\tau_t$ was defined in Eqn. (2). Although arrivals might not be i.i.d. because of correlation in packet delays, the i.i.d. assumption allows us to explicitly compute the performance of the proposed estimators and to find the optimal gains within this class.

Starting from the buffer of the optimal filter described in Sec. III, we consider the subset of the measurements with time delays in $M, \dots, M+N$ (the subset will be called a *shifted buffer*), where $M = 0, \dots, \bar{N}$ is the starting point of the shifted buffer, and $N = 1, \dots, \bar{N} - M$ is its length (an example is shown in Fig. 4). The estimation scheme has the following structure:

$$
\begin{aligned}
\tilde{x}_{t-M-N|t-M-N}^t &= \hat{x}_{t-M-N|t-M-N}^{t-1} \\
\hat{x}_{k|k}^t &= A\hat{x}_{k-1|k-1}^t + \gamma_k^t \tilde{K}_{t-k} (\tilde{y}_k^t - CA\tilde{x}_{k-1|k-1}^t), \\
\tilde{x}_{t|t}^t &= A^M \tilde{x}_{t-M|t-M}^t
\end{aligned} \quad (3)
$$

where $k = t-M-N+1, \dots, t-M$, which mimics the time-varying estimator with the buffer in the previous section, but with gains $\{\tilde{K}_h\}_{h=M}^{M+N-1}$ not depending on the packet arrival sequence $\gamma_k^t$, unlike the gains $\{K_k^t\}$ of the optimal filter of Sec. III.

The performance of this new estimator is measured in terms of its prediction error covariance $\tilde{P}_{t+1|t} = \mathbb{E}[(x_{t+1} - A\tilde{x}_{t|t}^t)(x_{t+1} - A\tilde{x}_{t|t}^t)^T \,|\, \gamma_1^t, ..., \gamma_t^t]$. Obviously it must be $P_{t+1|t}^t \leq \tilde{P}_{t+1|t}^t$ for every sequence $\gamma_k^t$ since the filter in the previous section is the minimum variance linear filter. Just like $P_{t+1|t}^t$, the prediction error covariance $\tilde{P}_{t+1|t}^t$ is a random variable since it depends on the specific realization of the arrival process $\gamma_k^t$. Therefore, we are interested in computing the expected prediction error covariance with respect to all possible realizations of $\gamma_k^t$, i.e. $\overline{P}_{t+1|t}^t = \mathbb{E}_\gamma[\tilde{P}_{t+1|t}^t] = \overline{P}_{t+1|t}^t(\tilde{K}, N, M)$, where we made explicit the dependence on the gains $\tilde{K} = (\tilde{K}_M, \dots, \tilde{K}_{M+N-1})$, the length of the buffer $N$, and its initial position $M$. The following theorem provides stability conditions for the proposed filter.

*Theorem 1:* Consider the following modified A.R.E.:

$$
P = \Phi_\lambda(P) = APA^T + Q - \lambda APC^T(CPC^T + R)^{-1}CPA^T \quad (4)
$$

and the gain $K_P = g(P) = PC^T(CPC^T + R)^{-1}$. If $A$ is unstable, then there exists a unique positive semidefinite solution if and only if $\lambda > \lambda_c$ where:

- $\lambda_c$ depends only on the pair $(A, C)$;
- $\lambda_c$ satisfies the following inequalities (where the $\sigma_i^u(A)$'s are the unstable eigenvalues of $A$):

$$
p_{min} = \frac{1}{\prod_i |\sigma_i^u(A)|^2} \leq 1 - \lambda_c \leq \frac{1}{\max_i |\sigma_i^u(A)|^2} = p_{max}
$$

- $p_{min} = 1 - \lambda_c$ if $C$ is rank one;
- $p_{max} = 1 - \lambda_c$ if $C$ is invertible.

If $A$ is strictly stable, then there always exists a unique positive semidefinite solution. Consider also the class of filters defined by Eqn. (3), and suppose the packet arrival process is i.i.d.. If $\lambda_{M+N-1} < \lambda_c$ then $\lim_{t\to\infty} \sup_t \overline{P}_{t+1|t}^t(\tilde{K}, N, M) = \infty$ for any choice of the gains $\tilde{K}$. If $\lambda_{M+N-1} > \lambda_c$, then consider the following positive semidefinite matrices:

$$
\begin{aligned}
&V_{M+N-1} = \Phi_{\lambda_{M+N-1}}(V_{M+N-1}) \\
&V_k = \Phi_{\lambda_{k+1}}(V_{k+1}), \quad k = M+N-2, \dots, M \\
&V_k = AV_{k+1}A^T + Q = \Phi_0(V_{k+1}), \quad k = M-1, \dots, 0
\end{aligned} \quad (5)
$$

and the gains $\tilde{K}_k^* = g(V_k), k = M+N-1, \dots, M$. Then:

$$
\begin{aligned}
&\lim_{t\to\infty} \overline{P}_{t+1|t}^t(\tilde{K}^*, N, M) = V_0(N, M) \\
&\lim_{t\to\infty} \overline{P}_{t+1|t}^t(\tilde{K}, N, M) \geq V_0(N, M), \quad \forall \tilde{K} \quad .
\end{aligned}
$$

Finally $V_0(N, M) \geq V_0(N+1, M)$.

*Proof:* The proof is a straightforward application of the results presented in [10] and is therefore omitted. ∎

Thm. 1 states that if the packet arrival probability for the last slot in the buffer $\lambda_{M+N-1}$ is sufficiently high, then there exists a stable estimator within the class of filters here proposed. Thm. 1 also shows how to find the best estimator in terms of minimum variance within this class. The best expected prediction error covariance $V_0(N, M)$ is a function of the buffer length $N$ and initial position $M$. The memory and computational complexity for such estimators do not depend on $M$. Therefore, we would like to find the best $M$ which minimizes $V_0(N, M)$. Unfortunately it is not possible to guarantee that there exits $M^*$ such that $V_0(N, M^*) \leq V_0(N, M)$, and indeed this is actually false in general. To overcome this limitation, we will consider a cost function which is linear and positive in $V \geq 0$, i.e. a function $f : \mathbb{R}^{n\times n} \to \mathbb{R}^+$. Some examples are $f(V) = \mathrm{trace}(V)$ and $f(V) = z^T V z$, where $z \in \mathbb{R}^n$. Using this cost function we will compute the optimal shifted buffer $M$ for any fixed $N$ as:

$$M^*(N) = \arg\min_M f\left(V_0(N, M)\right)$$

and the corresponding minimum cost $v^*(N) = \min_M f(V_0(N, M))$. Since $M$ is an integer, it is not possible to find the minimum in closed form. Therefore, we need to explicitly compute $f(V_0(N, M))$ for all $M$. However, this can be done off-line and then used for on-line estimation.

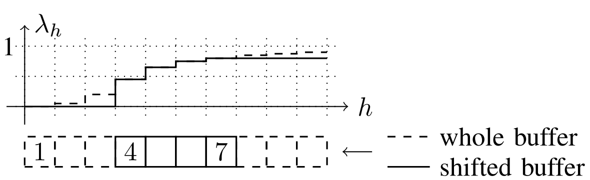


Fig. 4. Example of shifted buffer with $M=3$ and $N=4$; the elements of the buffer and the $\lambda_h$'s used in Eqn. (4) are plotted with a continuous line. The dashed $\lambda_h$ refers to the trivial buffer with $M=0$ and $N=\bar{N}$.

## V. Estimation performance under UPD and DSF routing protocols

In this section we apply the results of Sec. III to the situation proposed in Sec. II-D to evaluate performances for the 2D target tracking application. A popular model for the dynamics of a moving target is given by a double integrator subject to white noise, i.e. $\ddot{\xi}_x(t) = w_x(t)$ where $\xi_x$ is the position of the moving target along the $x$-axis and $w_x(t)$ is continuous time white noise with zero-mean and variance $q$. We also assume that the position measure is noisy, i.e. $y_x(t) = \xi_x(t) + v(t)$, where $v(t)$ is zero mean white noise with variance $I$. The dynamics along the $y$-axis are modeled similarly and the noises are assumed to be uncorrelated along the two axes. The state space dynamics, discretized with period T, are written as:

$$x_{t+1} = \begin{bmatrix} G & 0 \\ 0 & G \end{bmatrix} x_t + w_t, \quad y_t = \begin{bmatrix} H & 0 \\ 0 & H \end{bmatrix} x_t + v_t$$

$$G = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \; H = \begin{bmatrix} 1 & 0 \end{bmatrix}, \; S = \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T \end{bmatrix}$$

where $x^T = [\xi_x \, \dot{\xi}_x \, \xi_y \, \dot{\xi}_y]$, $w_t$ and $v_t$ are white Gaussian noise with covariance $Q = q \begin{bmatrix} S & 0 \\ 0 & S \end{bmatrix}$ and $R = rI$ respectively,

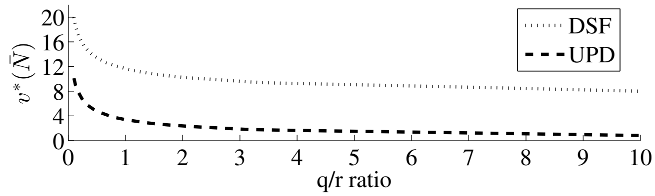


Fig. 5. Estimation error cost $v^*(\bar{N})$ for the full length buffer as a function of the ratio $q/r$ with $q = 1$ for the UPD and DSF protocols.

and $I$ is the identity matrix. In this case $(A, Q^{1/2})$ is reachable, $(A, C)$ is observable and the critical packet arrival probability introduced in Thm. 1 is $\lambda_c = 0$ (all the eigenvalues are one). Since the behavior of the filter is regulated by the ratio $q/r$, we fix w.l.o.g. $q = 1$ and evaluate the performance as a function of $r$, in terms of the mean square prediction error $v^*(N) = f(V_0(N, M^*(N)) = z^T V_0(N, M^*(N))z$ on the position of the moving target, where $z^T = [1\,0\,1\,0]$ and $V_0(N, M)$ is the expected prediction error covariance of the estimator with a shifted buffer with size $N$ and initial position $M$ defined in Sec. IV.

First we compute the best achievable performance of filters with constant gains as a function of $r$ for UPD and DSF protocols with end-to-end packet delay statistics shown in Fig. 2. Noting that $\lambda_h^{\mathrm{UPD}} = 0$ for $h < 10$ and $\lambda_h^{\mathrm{UPD}} \approx 1$ for $h > 40$ we can set $\bar{N}^{\mathrm{UPD}} = 40$, i.e. almost all packets arrive with a delay between 10 and 40 time steps. On the other hand $\lambda_h^{\mathrm{DSF}} = 0$ for $h < 22$ and $\lambda_h^{\mathrm{DSF}} = \lambda_{h+1}^{\mathrm{DSF}} = 0.81$ for $h \geq 24$, which implies that $\bar{N}^{\mathrm{DSF}} = 24$ and that packet loss probability is $p_{loss}^{\mathrm{DSF}} = 1 - 0.81 = 0.19$.

Fig. 5 compares the best achievable performance in terms of $v^*(\bar{N})$ and shows that UPD always performs better than DSF for the topology and link probabilities of Fig. 1. This is to be expected for the two extreme regimes, i.e. for large $q/r$ and for small $q/r$. In fact, for large $q/r$, old measurements cannot reduce the estimation error since $x_t$ changes too rapidly. Since UPD delivers some packets with much smaller delay than DSF, it should perform better. For small $q/r$, $x_t$ changes very slowly, so old measurements reduce the estimation error. Therefore, a relevant parameter is the packet loss probability, which is bigger for DSF. Note that UPD performs better for all the $q/r$ ratios for *this* particular topology and link probabilities. It can be shown that in other cases the situation can be inverted.

When using a buffer of size $N < \bar{N}$ there are tradeoffs between estimation and computational complexity. Fig. 6 shows the performance of the filters as a function of $N$ for three different noise ratios $q/r$ (buffer shift $M$ being chosen optimally $\forall N$). As expected, the performance for DSF becomes constant for $N > 3$ since all packets arrives with delay $h \in \{22, 23, 24\}$. Instead the performance of UPD improves until $N = 30$ (range of delay of the packets), and after becomes constant; anyway the improvements after $N > 20$ are so small that it is not useful to use longer buffers. Note that, if both $q/r$ and the buffer are very small, DSF performs better than UPD. Therefore, it is not possible to claim that UPD is always superior to DSF.

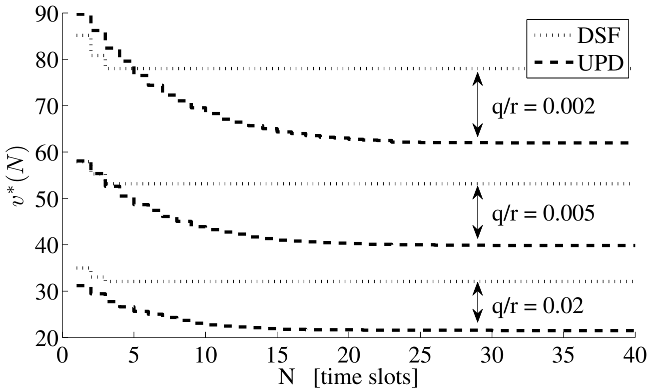


Fig. 6. Estimation error cost $v^*(N)$ relative to the optimal buffer shift $M^*(N)$ as a function of the buffer length $N$. The functions are plotted for three different ratios of $q/r$ and for both UPD and DSF.

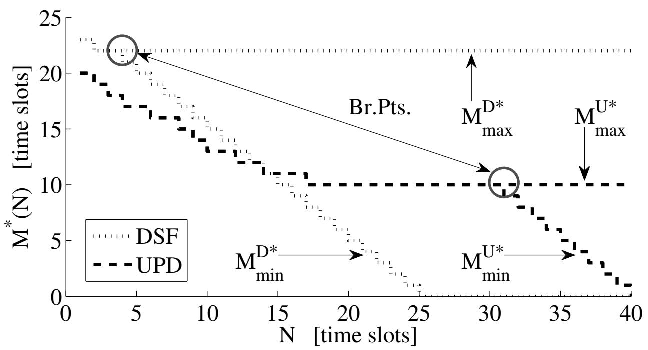


Fig. 7. Optimal buffer shift $M^*(N)$ as a function of the buffer length $N$ for $q = 1$ and $r = 0.005$. After the branching point (Br.Pts. in figure), the value $M^*(N)$ can be any point between the two branches, i.e. $M^*_{min}(N) \leq M^*(N) \leq M^*_{max}(N)$.

Finally, Fig. 7 shows the optimal buffer shift $M^*(N)$ as a function of the buffer length. As expected, $M^*_{\text{DSF}}(N)$ is around the minimum delay of the measurements (i.e. 22). For $N > 3$, the performance becomes constant, and the optimal $M^*_{\text{DSF}}(N)$ is not unique, since any buffer including delays $h \in \{22, 23, 24\}$ performs optimally. Therefore $M^*_{\text{DSF}}(N)$ s.t. $N > 3$ can be chosen such that $M^{D*}_{min}(N) \leq M^*_{\text{DSF}}(N) \leq M^{D*}_{max}(N)$. The UPD case depends more on the $q/r$ ratio. When $N$ is small we will use a few recent measurements, while when $N$ is large we will include more older measurements. In the case of Fig. 7, $M^*_{\text{UPD}}$ initially decreases as $N$ increases, indicating that the buffer adds packets with smaller delays. Then for $h < 10$ the buffer adds the packets with bigger delays (a further decrease in $M$ does not help). When $N > 30$, the addition of packets with smaller delay also provides negligible improvements in performance so each $M^*_{\text{UPD}}(N)$ can be chosen s.t. $M^{U*}_{min}(N) \leq M^*_{\text{UPD}}(N) \leq M^{U*}_{max}(N)$.

## VI. CONCLUSIONS

In this paper we evaluate the performance of filters when measurements are subject to packet loss and random delay [10] while using two different WSN communication protocols designed for real-time monitoring and tracking [5]. We also propose a new set of estimators with constant gains and a shifted buffer, which allows the design to trade off computational complexity and performance. In particular, we show that unless all the packet delay probabilities $\lambda_h^\alpha$ of a communication protocol $\alpha$ are greater than the relative $\lambda_h^\beta$'s of another protocol $\beta$, it is not possible to claim that one protocol is better in absolute terms. The performance of a communication protocol depends on the ratio between process and measurement noises $q/r$, the dynamics of the system $A$, and the buffer length $N$. Nonetheless, the tools developed in this paper can be readily used by a control engineer to compare protocols for a specific application.

There are still several research avenues which deserve to be explored. First, the performance was evaluated in terms of estimation error covariances averaged over all possible packet delay realizations, but it would also be important to know the spread of these covariances along a typical realization. This spread is directly related to the jitter experienced by the estimation error, which is known to give rise to poor control performance. Second, this work can be extended to handle non-i.i.d. packet arrival processes, i.e. extended to communication protocols with correlated delays between consecutive packets.

## REFERENCES


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

[2] A. Rowe, R. Mangharam, and R. Rajkumar, "RT-Link: A time-synchronized link protocol for energy-constrained multi-hop wireless networks," in *3rd Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, vol. 2, 2006, pp. 402–411.

[3] A. Bonivento, C. Fischione, L. Necchi, F. Pianegiani, and A. Sangiovanni-Vincentelli, "System level design for clustered wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 3, no. 3, pp. 202–214, Aug. 2007.

[4] P. G. Park, C. Fischione, A. Bonivento, K. H. Johansson, and A. Sangiovanni-Vincentelli, "Breath: A self-adapting protocol for wireless sensor networks in control and automation," in *Proc. 5th Annual IEEE Comm. Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks SECON '08*, 2008, pp. 323–331.

[5] P. Chen and S. Sastry, "Latency and connectivity analysis tools for wireless mesh networks," in *Proceedings of the First International Conference on Robot Communication and Coordination (ROBOCOMM)*, October 2007.

[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] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. Sastry, "Foundations of control and estimation over lossy networks," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 163–187, January 2007.

[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] J. Nilsson, B. Bernhardsson, and B. Wittenmark, "Stochastic analysis and control of real-time systems with random time delays," *Automatica*, vol. 34, no. 1, pp. 57–64, January 1998.

[10] 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.

[11] Y. Bar-Shalom, H. Chen, and M. Mallick, "One-step solution for the general out-of-sequence measurement problem in tracking," *IEEE Transactions on Aerospace and Electronics Systems*, vol. 40, no. 1, pp. 27–37, 2004.

[12] K.-S. Zhang, X. R. Li, and Y.-M. Zhu, "Optimal update with out-of-sequence observations for distributed filtering," *IEEE Transactions on Signal Processing*, vol. 53, no. 6, pp. 1992–2004, 2005.