INTERNATIONAL JOURNAL OF ROBUST AND NONLINEAR CONTROL

Int. J. Robust Nonlinear Control 2008; **1**:1–19 *Prepared using* rncauth.cls *[Version: 2002/11/11 v1.00]*

Wireless Digital Control of Continuous Passive Plants Over Token Ring Networks

Nicholas Kottenstette[1], and Panos J. Antsaklis[1]

[1] *Department of Electrical Engineering*

University of Notre Dame

Notre Dame, IN 46556

Email: nkottens@nd.edu antsaklis.1@nd.edu

SUMMARY

In a recent paper we have shown how *wave variables* can be used to interconnect *passive* plants with *passive* controllers such that the system remains l^2-stable in spite of time varying delays and data dropouts. The present paper further enhances these results by providing a detailed model which captures time varying delays, data dropouts and network capacity for wireless ring token networks. It also provides a new theorem showing how an asynchronous controller can be implemented which maintains an l^2-stable system. Furthermore, this asynchronous controller reduces the overall distortion when compared to synchronous controllers which rely on lossy data reduction techniques. Copyright © 2008 John Wiley & Sons, Ltd.

KEY WORDS: *passivity*, non-linear control, Markov chains, *Passive Asynchronous Transfer Unit*, *PATRU*, scattering theory, switched systems, queuing theory, networked control, CC2420, 802.15.4

*Correspondence to: Department of Electrical Engineering

University of Notre Dame

Notre Dame, IN 46556

Received 01/15/2008

Copyright © 2008 John Wiley & Sons, Ltd.

1. Introduction

In [7, Theorem 4] it has been shown how to digitally control a *passive* [7, Definition 3-I] plant over a network in which data can be subject to arbitrary time delays and data dropouts while remaining l^2-stable. Initial simulations in [7] showed the system to perform quite well under fixed time delays however it remained to be shown how well the system performed under time varying delays. The purpose of this paper is twofold: first, to present a model which accurately captures both the time varying delays associated with routing data over a token network and the data dropouts that occur due to finite queue length; and second, is to compare how well a *passive* asynchronous controller compares to a rate adaptive *passive* compression scheme [6, Section 4.3.1]. In order to evaluate performance we compare how our control scheme is able to effectively minimize the distortion (Definition 3) between a desired set point and the plants position. Since most digital control systems are concerned with sampling at a constant bit rate (*CBR*) we will focus our analysis on such *CBR* generating sources.

The networking delays and data dropout rate is intimately linked to the data rate of a *CBR* source, the length of the data queue and the capacity of the network. The capacity of the network is typically limited by the distance between stations, the radio, medium access control (*MAC*) and routing policy chosen. Therefore, much work has been done in order to determine optimum ways to route messages from various sources to desired sinks. In [14] classical optimal control techniques [10] were used to determine an optimal static routing policy which minimizes energy consumption. [11] provides an algorithm which specifies the nodes' order of transmission and power level in order to maximize the network lifetime known as *maximum lifetime accumulative broadcast* (*MLAB*). [5] illustrates how to use feedback techniques in order to find the optimal number of redundant packets to send in order to

Copyright © 2008 John Wiley & Sons, Ltd.

Prepared using rncauth.cls

maximize an objective function $J(u_b, n, p)$. [4] shows how to use feedback to control uncooperative users of networking resources. Unfortunately, most of these analysis are not concerned with attempting to characterize or understand the effects of the delay of the information transmitted in the network.

Papers which look specifically at *MAC* protocols typically have a stronger relationship between network capacity and routing delays. Two popular wireless *MAC* protocols which have received much attention are m-phase time division multiple access (*TDMA*), and *ALOHA*. *TDMA* typically outperforms a random access protocol such as *ALOHA*, however, for correlated flows such as those under a constant bit rate source (*CBR*), *ALOHA* can obtain greater capacity than *TDMA* [16]. Also, in [17] the time varying delay statistics can be computed for *TDMA* and *ALOHA* *MAC*s subject to various sources such as a *CBR* source. These papers, do not characterize the delay characteristics for round trip communication patterns which will be seen in the l^2-stable *passive* networks introduced in [7]. Therefore, we shall study a simple token passing *MAC* protocol which will allow us to compute the network capacity and delays for our l^2-stable *passive* networks.

In Section 2 we determine the delay characteristics of *MAC*s which use token passing by developing the appropriate Markov chains to describe the system. As such, the capacity of our network is calculated from the average number of round trip steps $\tau_m = \{M_m[v]\}$ taken for the delivery of control data which originated from the starting node m. τ_m is calculated using a convenient set of formulas provided by [13] and summarized in [6, Appendix C.1]. Note that a wireless token ring protocol was chosen to control the spacing of vehicles in the Automated Highway System program and the Berkeley Aerobot Project [3]. This protocol is robust and can handle problems such as duplicate tokens, lost stations (nodes), and dynamically adding new stations to the ring. Furthermore, if m is large, smaller rings of stations could be established with an alternating carrier frequency in order to increase network capacity. In Section 2 we provide the basic analysis of a single ring of m stations. Looking at performance of

Copyright © 2008 John Wiley & Sons, Ltd.

Prepared using rncauth.cls

a single ring is further justified when looking at potential wireless telesurgery applications in which $m = 4$, consisting of a station at the operator, robot and an aerial unmanned autonomous vehicle (the aerial vehicle can be treated as two stations in the ring) [12].

In [8] methods to reduce the data communicated over a network using sample based lossy data reduction (*LDR*) for telemanipulation systems has been investigated. It is concerned with developing *LDR* algorithms which provide good compression while attempting to maintain a high level of *transparency*. They proposed a measure which could be reflective of *transparency* as

$$I = \int_0^t [(f_m - f_{sd})^2 + (e_c - e_m)^2] d\tau \quad (1)$$

[6, see Figure 2.1] in which the flows (f_m, f_{sd}) were velocities (v_{hsi}, v_{to}) of the human system interface (*HSI*) and teleoperator (TO) and the efforts (e_c, e_m) were the corresponding forces (F_{to}, F_{hsi}). We are interested in evaluating the performance of our plant G_p controller G_c network as governed by the discrete random delays which occur over a wireless ring network. In Section 3 we will study the resulting distortion between a desired position set point $\theta_{set}(i)$ and the resulting position output from the plant $\theta_{act}(i)$. In particular we will show how a *novel passive* asynchronous control technique, in which the controller is only run when new data is received from the plant, outperforms a *passive* discrete time varying *LDR* algorithm [6, Section 4.3.1]. When the asynchronous controller is not run, it will drop the current reference input and its output will reset to 0 after being held at its last computed level for a period equal to that of the sample and hold period of the plant. Typically, when data for the controller is dropped over a *passive* control network, the controller either assumes a 0-input and updates its output accordingly or it attempts to make a prediction of what the correct input should be. The variable compression scheme compared attempts to make a smooth prediction on the input to tolerate delay and data dropouts. Although less steady state error and distortion occurs when using a variable compression scheme over a 0-input prediction scheme, it still does not perform as well as

Copyright © 2008 John Wiley & Sons, Ltd.

Prepared using rncauth.cls

using an asynchronous controller. A detailed simulation and discussion is provided in Section 3.1.1 with a corresponding Theorem 1 (Section 3.1.2) which shows that the asynchronous controller is indeed *strictly-output passive*, as indicated by our simulations. Conclusions follow and are presented in Section 4.

2. Capacity and Delay of Single Ring Token Networks

A ring network consists of m stations in which each station has a successor (the station it will pass its token to) and a predecessor (the station it will receive a token from). For simplicity we will consider rings in which station i will have the following predecessor, successor pairs $(pr_i, sc_i), \forall i \in \{1, \ldots, m\}$:

$$(pr_i, sc_i) = \begin{cases} (i+1, m), & \text{if i=1;} \\ (1, m-1), & \text{if i=m;} \\ (i+1, i-1), & \text{otherwise.} \end{cases} \quad (2)$$

Figure 1 illustrates a corresponding ring network in which each station has a probability P_i of successfully sending a packet of n_p bits (which typically includes n_d data bits, n_h header bits and n_{fcs} frame check sequence bits) and receiving an acknowledgment of n_{ack} from its successor. For simplicity we assume that the successful transmission of a packet and receiving an acknowledgment is equivalent to passing a token to its successor. The packet dropout rate is denoted as P_{per} which is derived from the bit error rate P_{ber} such that $P_{per} = 1 - (1 - P_{ber})^{n_p}$ ([6, Appendix C.2]). Therefore, we conservatively estimate $P_i = (1 - P_{ber})^{n_p + n_{ack}}$. This assumption is conservative, since most people neglect the acknowledgment entirely.

Definition 1. *Let the index* $i = \{0, 1, 2, \ldots\}$ *denote a packet time slot. Let station* m *generate a*

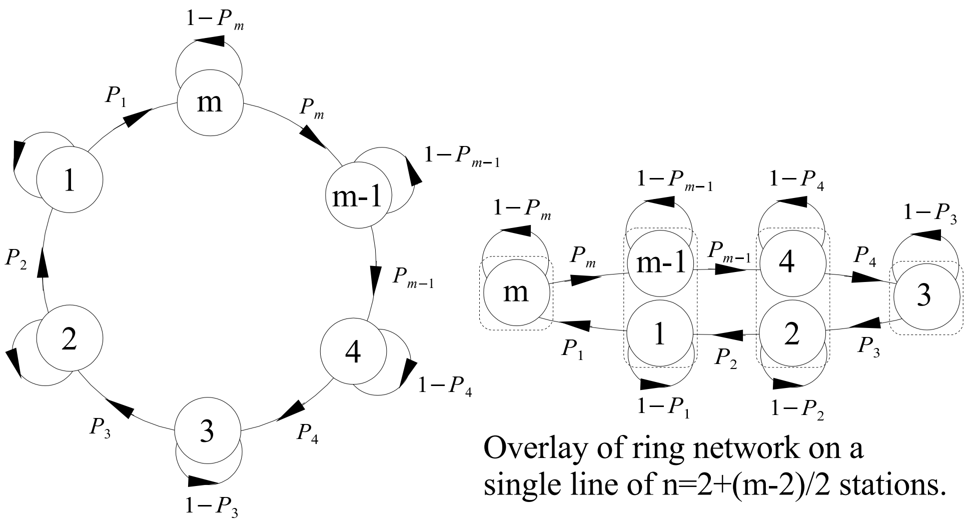

Figure 1. Ring networks consisting of m stations and $n = \frac{m-2}{2} + 2$ stations.

packet of data to transmit around a ring network as depicted in Figure 1 at a given rate r *such that the next transmission index* $i_{next} = i_{current} + r$ *in which* $i_{current} = i$ *when a packet is currently being transmitted from station* m *(the remaining stations will only relay* m*'s message around the ring network). The minimum ideal rate in which station* m *can generate a packet is* $r_{pc} = m$*. Therefore, we define the ideal packet capacity as*

$$\lambda_{pc} = \frac{1}{r_{pc}} = \frac{1}{m} \tag{3}$$

The average packet capacity λ_p *depends on the average round trip time* τ_m *and has the following form*

$$\lambda_p = \frac{1}{\tau_m} \tag{4}$$

In order to calculate τ_m we note that the following Markov chain describes a single packet journey in our ring network from station m and back to m. We capture the final round trip packet delivery from

Copyright © 2008 John Wiley & Sons, Ltd.

Prepared using rncauth.cls

station 1 to station m with the corresponding absorbing state 0. The transition matrix P is as follows:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ P_1 & 1-P_1 & 0 & 0 & \dots & 0 \\ 0 & P_2 & 1-P_2 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots\dots\dots & & 0 & P_m & P_m - 1 \end{bmatrix} = \begin{bmatrix} I & \mathbf{0} \\ R & Q \end{bmatrix} \tag{5}$$

This is in canonical form [6, Appendix C.1]. We can now compute $\mathbf{N}$ in which each $N_{i,j}$ element $i,j \in \{1,\dots,m\}$ is the average number of times each station j is visited if the packet originated in station i before the token reaches the absorbing state 0.

$$\mathbf{N} = (I - Q)^{-1} \tag{6}$$

$$\mathbf{N} = \begin{bmatrix} \frac{1}{P_1} & 0 & \dots\dots\dots & & & 0 \\ \frac{1}{P_1} & \frac{1}{P_2} & 0 & \dots\dots & & 0 \\ \frac{1}{P_1} & \frac{1}{P_2} & \frac{1}{P_3} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \frac{1}{P_1} & \frac{1}{P_2} & \frac{1}{P_3} & \dots\dots & & \frac{1}{P_m} \end{bmatrix} \tag{7}$$

Using the formulas in [6, Table C.1] the mean arrival time to station m when the initial packet started at station i is obtained by solving τ:

$$\tau = \begin{bmatrix} \frac{1}{P_1} \\ \frac{1}{P_1} + \frac{1}{P_2} \\ \cdot \\ \cdot \\ \sum_{i=1}^{m} \frac{1}{P_i} \end{bmatrix} \tag{8}$$

Copyright © 2008 John Wiley & Sons, Ltd.

Prepared using rncauth.cls

Or equivalently

$$\tau_i = \begin{cases} \frac{1}{P_i}, i = 1; \\ \tau_{i-1} + \frac{1}{P_i}, 1 < i \leq m. \end{cases} \tag{9}$$

Using [6, (C.2) in Appendix C.1] to solve for σ_v^2 results in the following

$$\sigma_{v_i}^2 = 2\sum_{j=1}^{i} \frac{\tau_j}{P_j} - \tau_i(1+\tau_i). \tag{10}$$

Solving for $\sigma_{v_\delta}^2 = (\sigma_{v_i}^2 - \sigma_{v_{i-1}}^2)$ results in the following:

$$\sigma_{v_\delta}^2 = \frac{2\tau_i}{P_i} - \tau_i(1+\tau_i) + \tau_{i-1}(1+\tau_{i-1}) \tag{11}$$

$$= \frac{2\tau_i}{P_i} - \tau_i - \tau_i^2 + (\tau_i - \frac{1}{P_i})(1 + \tau_i - \frac{1}{P_i}) \tag{12}$$

$$= \frac{1-P_i}{P_i^2}. \tag{13}$$

Therefore, $\sigma_{v_i}^2$ can be equivalently written in the following recursive form

$$\sigma_{v_i}^2 = \begin{cases} \frac{1-P_1}{P_1^2}, \text{if } i = 1; \\ \sigma_{v_{i-1}}^2 + \frac{1-P_i}{P_i^2}, 1 < i \leq m. \end{cases} \tag{14}$$

We now state the following lemma.

Lemma 1. *Given a ring network as described by (2) and depicted in Figure 1. In which each station $i \in \{1, \ldots, m\}$ has a probability P_i of successfully sending a message to its successor station and $1 - P_i$ of unsuccessfully sending the message per attempt, the following holds:*

i. The average steps it takes to relay a message from station m around the ring network is $\tau_m = \sum_{i=1}^{m} \frac{1}{P_i}$, for the special case $P_i = p$ then $\tau_m = \frac{m}{p}$.

ii. The corresponding variance in the round trip time is $\sigma_{v_m}^2 = \sum_{i=1}^{m} \frac{1-P_i}{P_i^2}$, for the special case $P_i = p$ then $\sigma_{v_m}^2 = \frac{m(1-p)}{p^2}$.

Copyright © 2008 John Wiley & Sons, Ltd.

Remark 1. *The corresponding packet capacity of the ring network is* $\lambda_p = \frac{1}{\tau_m}$*, for the special case* $P_i = p$ *then* $\lambda_p = \frac{p}{m}$*. The solution of* λ_p *is a direct substitution of* $\tau_m = \frac{m}{p}$ *(Lemma 1-i) into (4) from Definition 1.*

Proof 1. *Most of the proof has been provided in the subsequent discussion. In summary:*

i. The solution for τ_m *is provided by (8).*

ii. The solution for $\sigma^2_{v_m}$ *is easily obtained by our recursive solution given by (14).*

Remark 2. *For the* l^2*-stable digital control network depicted in [7, Fig. 2] with* $n-2$ *equally spaced relay motes, we can create a ring network as depicted in Figure 1 in which station* m *is the plant* G_p *and station* $\frac{m}{2} = n-1$ *is the controller* G_c*. The controller only returns control data if it has received data from the plant. If we neglect the queuing delay associated with a CBR source and assume that the controller can compute a control command and immediately issue a response to the plant and we assume each node is equally likely to successfully transmitting a packet to its corresponding successor, then the average packet transmission delay from* G_p *to* G_c *and vice versa is* $\frac{m}{2p} = \frac{n-1}{p}$ *with a corresponding variance of* $\frac{(n-1)(1-p)}{p^2}$*.*

Although, the packet capacity is a convenient abstraction for characterizing the network, we still need to quantify the actual rate of data being transmitted. Therefore, we provide the following definition:

Definition 2. *The data capacity of a ring network is the number of actual data bits per second which can be transmitted round trip from a given source node, in which the data is relayed from every station in the network. Therefore, the data capacity in the ring network is*

$$\lambda_d = \frac{f_{bit}\lambda_p n_d}{(n_p + n_{ack})} \tag{15}$$

in which λ_p *is the packet capacity,* n_d *is the number of actual data bits which get transmitted for a*

Copyright © 2008 John Wiley & Sons, Ltd.

Prepared using rncauth.cls

successful packet delivery, n_p *is the number of packet bits,* n_{ack} *is the number of bits required for the acknowledgment, and* f_{bit} *is the transmit data rate (bits per second).*

For the special case when each node is equally spaced in a line network with n nodes ($m = 2(n-1)$ stations) then the corresponding data capacity is

$$\lambda_d = \frac{f_{bit} n_d p}{m(n_p + n_{ack})} = \frac{f_{bit} n_d p}{m(n_d + n_h + n_{fcs} + n_{ack})} \tag{16}$$

in which p is dependent on $n_d,\ n_p,$ and n_{ack} and the node spacing d. Building from the analysis given in [6, Appendix C.2] for determining the packet error rate for the CC2420 and using the following parameters given in Table I: we can calculate λ_d as a function of d and n_d which is displayed in

Table I. CC2420 PARAMETERS SUMMARY.

Term	Symbol	Value
bits per second	f_{bit}	250×10^3 bps
header bits	n_h	$13 * 8 = 104$ bits
frame check sequence bits	n_{fcs}	$2 * 8 = 16$ bits
ack bits	n_{ack}	$11 * 8 = 88$ bits
data bits	n_d	$0 \leq n_d \leq 960$ bits
typical transmit power	P_T	$-24 \leq P_T \leq 0$ dBm
worst case transmit power	P_T	$-27 \leq P_T \leq -3$ dBm
typical noise figure	NF	15.44 dB
worst case noise figure	NF	20.44 dB

Figure 2.

These are fairly easy estimates to remember and use when attempting to determine the average delays in a ring network. However, there is a way to get an even closer estimate of the corresponding

Copyright © 2008 John Wiley & Sons, Ltd.

Prepared using rncauth.cls

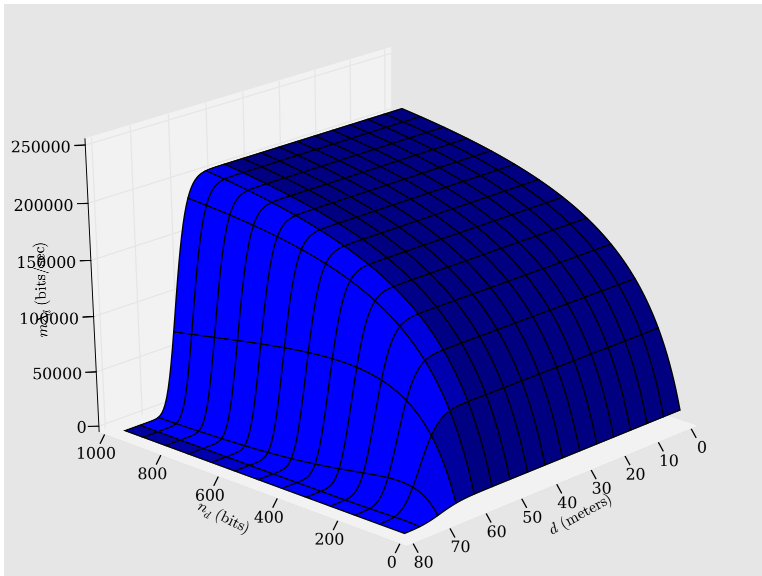

Figure 2. $m\lambda_d$ for the CC2420 as a function of d and n_d for a ring network not in free space ($n = 3.3,\ d_o =$ 8 meters, $P_T = -3$ dBm, $NF = 20.44$dB).

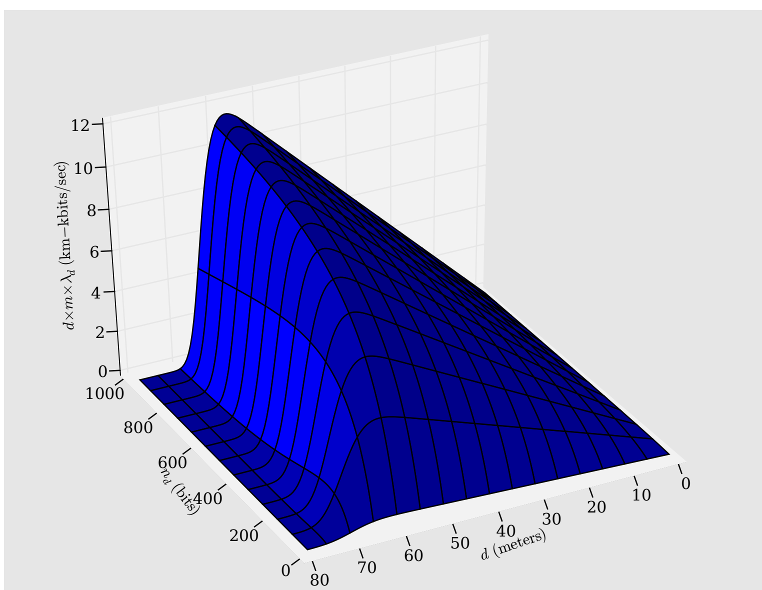

Figure 3. $d \times m \times \lambda_d$ for the CC2420 as a function of d and n_d for a ring network not in free space ($n = 3.3,\ d_o = 8$ meters, $P_T = -3$ dBm, $NF = 20.44$dB).

Copyright © 2008 John Wiley & Sons, Ltd.

delay in the network. Using techniques similar to those used by [9, 15], we describe a two dimensional Markov chain to track the head of queue *HoQ* delay at the m^{th} station of an outgoing packet which is generated from a *CBR* source of a packet every r^{th} slot and transmitted over a token ring network. The chain will be described by the (d, s) tuple in which $d \in \{-r+1, -r+2, \ldots, D\}$ denotes the delay of the *HoQ* in which D is the maximum delay before the packet will be dropped or compressed and $s \in \{1, 2, \ldots, m\}$ is the station where the token is in the network.

Lemma 2. *Given a ring network as described by (2) and depicted in Figure 1 and assuming that a packet of n_p bits will be rotated through the network (even if no new data is present). In which each station $i \in \{1, \ldots, m\}$ has a probability P_i of successfully sending a message to its successor station and $1 - P_i$ of unsuccessfully sending the message per attempt, the following transition matrix $P \in \mathbb{R}^{((D+r)m)\times((D+r)m)}$ describes the HoQ delay:*

$$P = \begin{array}{c} (\mathbf{-r+1,s}) \\ (\mathbf{-r+2,s}) \\ \cdots \\ (\mathbf{-1,s}) \\ (\mathbf{0,s}) \\ (\mathbf{1,s}) \\ \cdots \\ (\mathbf{D-1,s}) \\ (\mathbf{D,s}) \end{array} \left[\begin{array}{cccccccccc} \mathbf{0} & \mathbf{P_r} & \mathbf{0} & \multicolumn{6}{c}{\cdots\cdots\cdots\cdots\cdots\cdots} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{P_r} & \mathbf{0} & \multicolumn{5}{c}{\cdots\cdots\cdots\cdots\cdots} & \mathbf{0} \\ \multicolumn{10}{c}{\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{P_r} & \mathbf{0} & \multicolumn{3}{c}{\cdots\cdots\cdots} & \mathbf{0} \\ \mathbf{P_m} & \mathbf{0} & \multicolumn{2}{c}{\cdots\cdots} & \mathbf{0} & \mathbf{P_r - P_m} & \mathbf{0} & \multicolumn{2}{c}{\cdots} & \mathbf{0} \\ \mathbf{0} & \mathbf{P_m} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{P_r - P_m} & \mathbf{0} & . & \mathbf{0} \\ \multicolumn{10}{c}{\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots} \\ \mathbf{0} & \multicolumn{2}{c}{\cdots\cdots} & \mathbf{0} & \mathbf{P_m} & \mathbf{0} & \multicolumn{2}{c}{\cdots\cdots} & \mathbf{0} & \mathbf{P_r - P_m} \\ \mathbf{0} & \multicolumn{3}{c}{\cdots\cdots\cdots} & \mathbf{0} & \mathbf{P_r} & \mathbf{0} & \multicolumn{2}{c}{\cdots} & \mathbf{0} \end{array} \right] \quad (17)$$

Copyright © 2008 John Wiley & Sons, Ltd.

Prepared using rncauth.cls

in which $\mathbf{P_r} \in \mathbb{R}^{m\times m}$ *is the transition matrix which describes the evolution of the token,*

$$\mathbf{P_r} = \begin{bmatrix} 1-P_1 & 0 & \dots\dots\dots\dots & 0 & P_1 \\ P_2 & 1-P_2 & 0 & \dots\dots\dots\dots & 0 \\ 0 & P_3 & 1-P_3 & 0 \quad \dots\dots\dots & 0 \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ 0 & \dots\dots\dots & 0 \quad P_{m-1} & 1-P_{m-1} & 0 \\ 0 & \dots\dots\dots\dots & 0 & P_m & 1-P_m \end{bmatrix} \tag{18}$$

and $\mathbf{P_m} \in \mathbb{R}^{m\times m}$ *is the part of the transition matrix of actually shrinking the HoQ delay from* d *to* $d-r+1$ *when the token is at station* m

$$\mathbf{P_m} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \dots\dots & & 0 \\ 0 & \dots\dots & & 0 \\ \dots\dots\dots\dots \\ 0 & \dots\dots & & 0 \\ 0 & 0 & P_m & 0 \end{bmatrix}. \tag{19}$$

The $(\mathbf{d},\mathbf{s})$ *is a short hand to show how the states of the chain correspond to the rows of the transition matrix* P *in which* d *is a fixed column of integers and* s *would have each row correspond to the next state in the chain describing the evolution of the token i.e.*

$$(\mathbf{d},\mathbf{s}) = \begin{matrix} (d,1) \\ (d,2) \\ (d,3) \\ (\cdot,\cdot) \\ (d,m-1) \\ (d,m) \end{matrix}. \tag{20}$$

Copyright © 2008 John Wiley & Sons, Ltd.

Proof 2. *Creating the transition matrix is fairly straight forward:*

1. $(-r+1) \leq d < 0, s \in \{1, \ldots, m\}$*: we track the evolving state of the token* s*, this is done using* $\mathbf{P_r}$*, since the delay will increase by* 1 *after a transition from any state* s*, we simply offset the matrix* $\mathbf{P_r}$ *by* $m(d+1)$ *columns and place zeros elsewhere.*

2. $0 \leq d < D, s \in \{1, \ldots, m\}$*: once the delay* d *from the HoQ is* ≥ 0 *a packet is available for transmission at station* m*, hence if station* m *has the token and successfully transmits to station* 1 *with probability* P_m *then the delay* d *will be reduced such that* $d = d - (r-1)$*. This outcome is captured by matrix* $\mathbf{P_m}$ *and is offset by* $d(m+1)$ *columns. All other transitions will occur and the delay* d *will increase by one, hence* $\mathbf{P_r} - \mathbf{P_m}$ *is correspondingly offset by* $m(d+1)$ *columns with zeros elsewhere.*

3. $d = D, s \in \{1, \ldots, m\}$*: once the delay* $d = D$ *the packet will either be successfully delivered or dropped, therefore, the delay will shrink such that* $d = d - (r-1)$ *after the transition. Thus, we only need to track the state of the token* s *with* $\mathbf{P_r}$ *which will correspondingly be offset by* $d(m+1)$ *columns with zeros elsewhere.*

Remark 3. *To calculate the average delay of the HoQ we simply solve for the steady state distribution of our Markov chain:*

$$\pi P = \pi \tag{21}$$

in which $\pi = (\pi_{(-r+1,1)}, \pi_{(-r+1,2)}, \ldots, \pi_{(-r+1,m)}, \pi_{(-r+2,1)}, \ldots, \pi_{(D,m)})$ *is a row vector. The delay distribution,* $d_i(0, \leq d \leq D)$ *is given by*

$$d_i = \frac{P_m \pi_{(i,m)}}{\sum_{k=0}^{D} P_m \pi_{(k,m)}} = \frac{\pi_{(i,m)}}{\sum_{k=0}^{D} \pi_{(k,m)}} \tag{22}$$

where $\sum_{k=0}^{D} P_m \pi_{(k,m)}$ *is the normalizing constant, in which we only consider successful transmissions with probability* P_m *from station* m *to station* 1 *in deriving the delay distribution.*

Copyright © 2008 John Wiley & Sons, Ltd.

Remark 4. *Calculating the packet loss probability p_o is simply*

$$p_o = \frac{(1-P_m)\pi_{(D,m)} + \sum_{k=1}^{m-1} \pi_{(D,k)}}{\lambda}$$
$$= r[(1-P_m)\pi_{(D,m)} + \sum_{k=1}^{m-1} \pi_{(D,k)}] \quad (23)$$

in which $\lambda = \frac{1}{r}$, if we are in any other state besides m the packet will be dropped, when in state m there is only a probability of $(1-P_m)$ of dropping the packet and that is accounted for.

Remark 5. *For the l^2-stable digital control network depicted in [7, Fig. 2] with $n-2$ equally spaced relay motes, we can create a ring network as depicted in Figure 1 in which station m is the plant G_p and station $\frac{m}{2} = n-1$ is the controller G_c. The controller only returns control data if it has received data from the plant. If we consider the queuing delay associated with a CBR source at m and consider that the controller will not immediately compute a control command but pass along computations from previous transmissions. Then we can more closely approximate the delay of the delivery of data from the controller to the plant as if it was provided a CBR source r as well. This average delay from G_p to G_c and vice versa can be computed as follows:*

$$\tau_{pc} = \sum_{i=1}^{D} i d_{p_i} + \sum_{i=m-1}^{\frac{m}{2}-1} \frac{1}{P_i} \text{ for the delay from } G_p \text{ to } G_c, \quad (24)$$

$$\tau_{cp} = \sum_{i=1}^{D} i d_{c_i} + \sum_{i=\frac{m}{2}}^{m-1} \frac{1}{P_i} \text{ for the delay from } G_c \text{ to } G_p \quad (25)$$

in which d_{x_i} is the corresponding delay distribution for either the plant or controller if they were supplied a CBR source r. If $P_i = p$ then $\tau_{pc} = \tau_{cp}$ such that

$$\tau_{pc} = \tau_{cp} = \sum_{i=1}^{D} i d_i + \frac{(m-2)}{2p} \quad (26)$$

and the corresponding variance is

$$\sigma_v^2 = \sum_{i=1}^{D} i^2 d_i - (\sum_{i=1}^{D} i d_i)^2 + \frac{(n-2)(1-p)}{p^2}. \quad (27)$$

Copyright © 2008 John Wiley & Sons, Ltd.

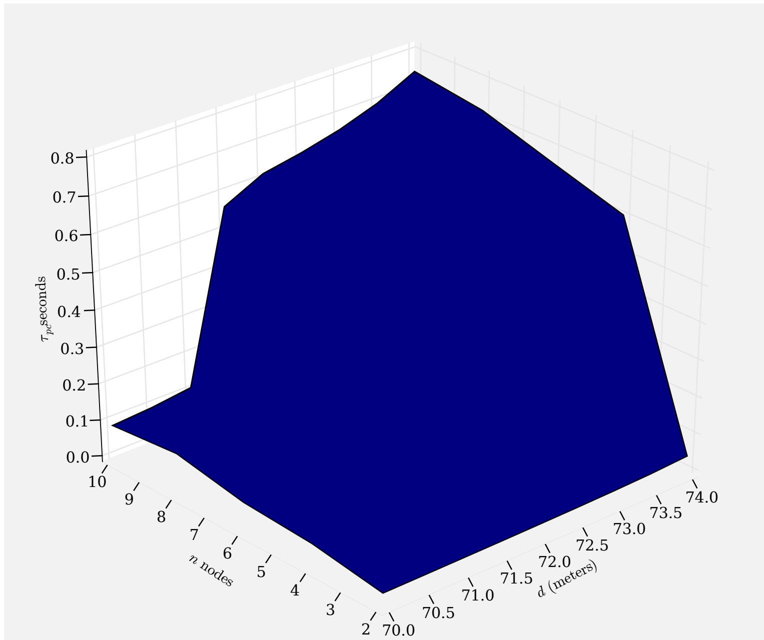

Figure 4. Theoretical mean delay of a data packet which contains 2 samples of u_{oc} in which each sample has 40 bits ($n_p = 288$ bits, $S = -90$dBm, $P_T = -3$dBm, $n = 3.3$, $d_o = 8.0$ meters, $r = 86$, $D = 516$) calculated using (26).

3. Distortion in Single Ring Token Networks

We have characterized the delay in Section 2, however, in order to account for the time varying queuing delay we assume that when the delay exceeds D slots that the packet should be dropped. However, dropping packets leads to drift in the actual position of the plant $\theta_{act}(i)$ when it is controlled using velocity feedback. Instead of dropping the data when it reaches the maximum delay D we chose to implement a *LDR* algorithm described in [6, Section 4.3.1] which is similar to the *compressor-expander* scheme used in [1, 2]. We use a distortion measure (Definition 3) in order to evaluate the performance of these compression schemes we.

Copyright © 2008 John Wiley & Sons, Ltd.

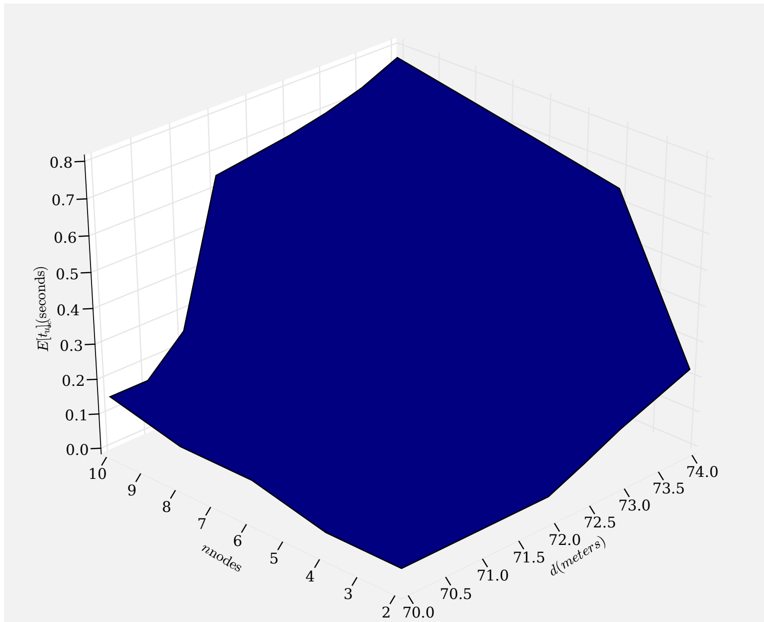

Figure 5. Simulated mean delay of u_{oc} in which up to 2 40 bit samples of u_{oc} can be transmitted (maximum $n_p = 288$ bits, $S = -90$dBm, $P_T = -3$dBm, $n = 3.3$, $d_o = 8.0$ meters, maximum $r = 86$, $D = 516$).

3.1. Evaluating the LDR Algorithm by Measuring Distortion.

Definition 3. *For the l^2-stable digital control network depicted in [7, Fig. 2] in which the flow output is denoted as $f_p(t)$. Denote the sampled integrated output of the plant as $\theta_{act}(i)$, assume that the user will provide a desired set point $\theta_{set}(i)$ to the input of a discrete second order trajectory generator which is a zero-order hold equivalent of $H_t(s)$ as described by [7, (41)]. The mean squared distortion is*

$$I_\theta = \frac{1}{T} E\{\sum_{i=0}^{T} (\theta_{set}(i) - \theta_{act}(i))^{\mathsf{T}} (\theta_{set}(i) - \theta_{act}(i))\} \tag{28}$$

in which $E[\cdot]$ denotes the expectation of the summation of the squared error which is dependent on the set point, the controller, the plant dynamics, and the time varying delays incurred due to the communication network.

Copyright © 2008 John Wiley & Sons, Ltd.

Prepared using rncauth.cls

For the example given in [7, Section IV], we estimated the corresponding distortion by averaging the summation given in (28) over 50 trials for a given number of n nodes and spacing of d meters. We chose $\theta_{set}(i)$ to use a square wave profile which is 0.0 radians for $0 \leq t < 8.0$ seconds, 1.0 radians for $8 \leq t < 16.0$ seconds, -1.0 radians for $16.0 \leq t < 24.0$ seconds, 0 radians for $24.0 \leq t$ seconds (Figure 8). Since we chose a modest sampling rate of $.05$ seconds we generate data at $5*8/.05 = 800.0$ bits per second. Which is a small fraction of the maximum data rate that can be achieved between a small number of nodes. As such, the delay between receiving data from the plant and controller is roughly the sampling rate ($.05$ seconds) for distances less than 70 meters. Furthermore, since the arrival delay is driven by the low sampling rate, the variance of the delay is extremely small. However, as the distance exceeds 70 meters, the capacity drop off is extremely sharp if we allow up to 12 samples to be transmitted per packet, such that the average delays are around 14 seconds when $d = 72.25$ meters and $n = 10$ nodes (Figures 6). Figure 6 provides the mean of the delays as a function of inter node distance d and number of nodes n. On the other hand if we still store up to 12 samples but only transmit up to 2 samples per attempt, the increase in the mean delay and the corresponding variance is significantly reduced for the same range of nodes and transmission distance (Figures 7). In spite of the significant random delays being incurred with increase in nodes n and node spacing d, the overall distortion I_θ degrades fairly gradually as is seen in [6, Figure 4.18] (note that $I_\theta = 0.5$ corresponds to keeping $\theta_{act} = 0$).

3.1.1. Comparing LDR Algorithm With Dropping Data in a Full FIFO. As we have seen, the *LDR* algorithm behaves exceptionally well for extremely unreliable communication channels. When the nodes are spaced 72.0 meters apart, the probability of a successful packet transmission is $P = (1.0 - .00631781)^{8\times(11+2+13+2\times5)} = 16\%$. Which corresponds to a capacity of 5596 bits/second for two nodes, and 622 bits/second for ten nodes ($m = 18$). Thus, to maintain an effective 800 bit/second

Copyright © 2008 John Wiley & Sons, Ltd.

Prepared using rncauth.cls

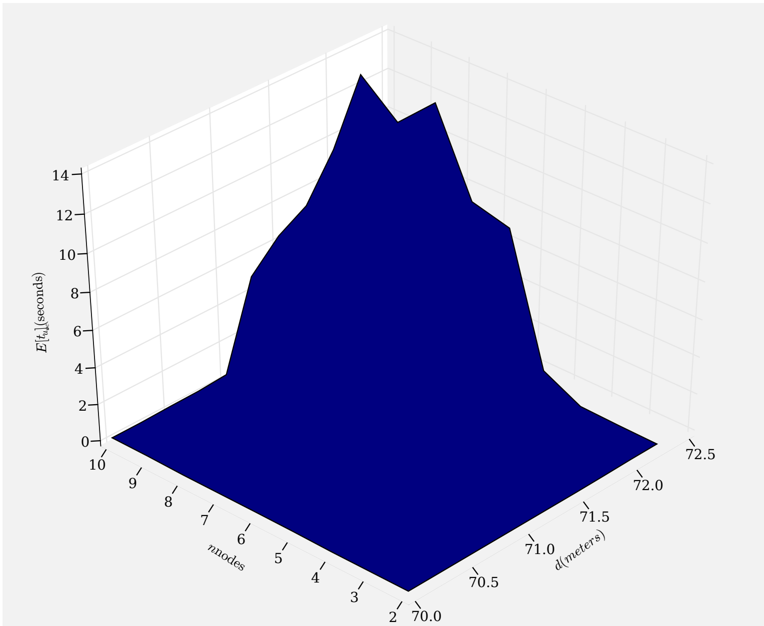

Figure 6. Mean delay of u_{oc} assuming $S = -90$dBm, $P_T = -3$dBm, $n = 3.3$, $d_o = 8.0$ meters, each sample consists of 40 bits, and up to 12 samples will be transmitted if available in a single packet.

data rate for ten nodes the data needs a compression ratio of $800/622$ and negligible distortion is seen. However, as the compression ratio increases so does the distortion in [6, Figure 4.18], which is seen in the delay and resulting steady state errors in [6, Figure 4.19]. Although the step responses are fairly smooth, when the nodes are 74.0 meters apart the probability of a successful packet transmission is $P = (1.0 - .010108)^{8\times(11+2+13+2\times5)} = 5.3$%. Which corresponds to a capacity of 1861 bits/second for two nodes, and 207 bits/second for ten nodes ($m = 18$) with a required compression ratio nearing 4.

However, when a FIFO is full and if we choose to use no compression and either drop the oldest or the current data sample, we can reduce the distortion at the lower data rates. This was an *unexpected* result, after reading about previous simulations in which dropped data resulted in steady state error [2, Figure 9] [1, Figure 5, Figure 6]. However, we have been implementing our controller in a slightly

Copyright © 2008 John Wiley & Sons, Ltd.

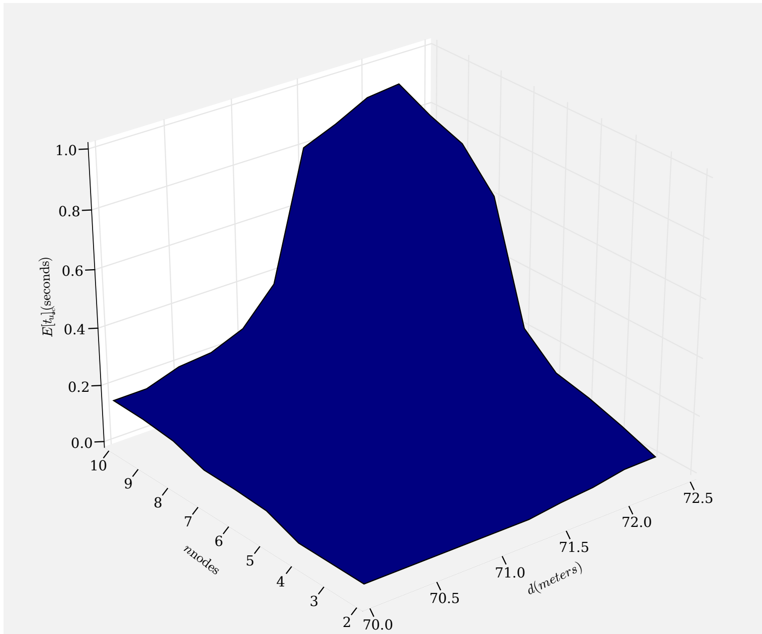

Figure 7. Mean delay of u_{oc} assuming $S = -90$dBm, $P_T = -3$dBm, $n = 3.3$, $d_o = 8.0$ meters, each sample consists of 40 bits, and up to 2 samples will be transmitted if available in a single packet.

different manner when not using compression. The controller only computes a new command when data from the plant is received, which implies we do not calculate a control command which will steer us away from our desired location by using a zero input. Furthermore, we achieved a significant improvement in distortion by simply dropping the sampled control input when data is not available from the plant. These two seemingly simple changes lead to a significant improvement in reducing distortion as can be seen in the step responses in Figure 8 and distortion plot in Figure 9.

3.1.2. Asynchronous Passivity Figure 10 indicates how to implement a *passive* digital controller in an asynchronous manner. The transfer of data between the controller and the plant is handled by the *Passive Asynchronous Transfer Unit* (*PATRU*).

Definition 4. *Define the set I as the set of received indexes $l = (i - p(i))$ from the plant which*

Copyright © 2008 John Wiley & Sons, Ltd.

Prepared using rncauth.cls

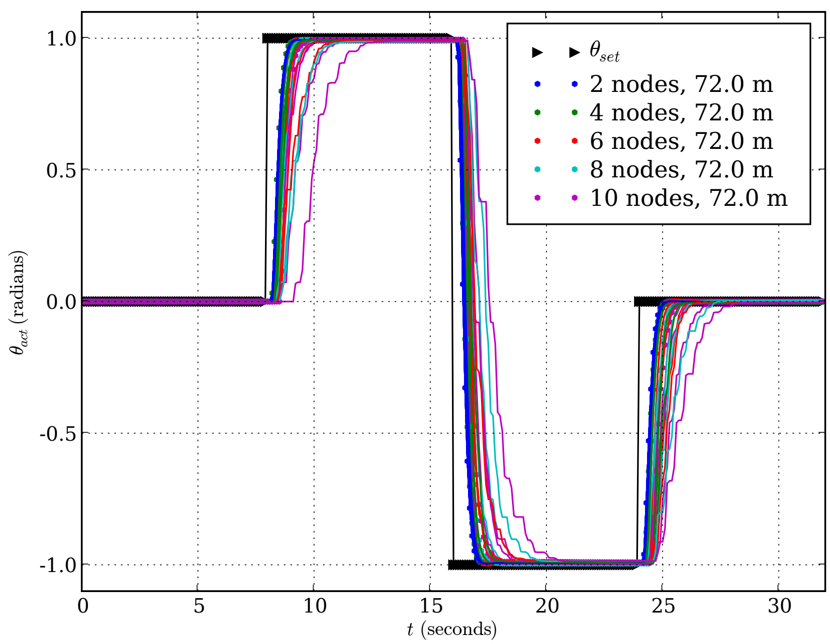

Figure 8. Typical step response by dropping either the latest sample or current sample when FIFO is full for nodes of $2, 4, 6, 8, 10$ and transmission distances of $70,\ 70.5,\ 71,\ 71.5,\ 72.0,\ 72.5,\ 73.0,\ 73.5,\ 74$ meters (each color in plot corresponds to a unique number of nodes).

correspond to the received tuple $(l, u_{op}(l))$ and the set J as the set of received indexes $k = (i - c(i))$ from the controller (via the PATRU) which correspond to the received tuple $(k, v_{oc}(k))$. When the plant and controller are initially enabled the sets I and J are empty. For simplicity of discussion we assume that the controller can instantly compute a new control command e_{oc} when new data arrives from the plant. The PATRU then handles the transfer of data as follows:

1. *If the periodically generated tuple $(l, u_{op}(l))$ from the plant has arrived to the PATRU on the controller side then:*

   if $l \in I^c$:

Copyright © 2008 John Wiley & Sons, Ltd.

Prepared using rncauth.cls

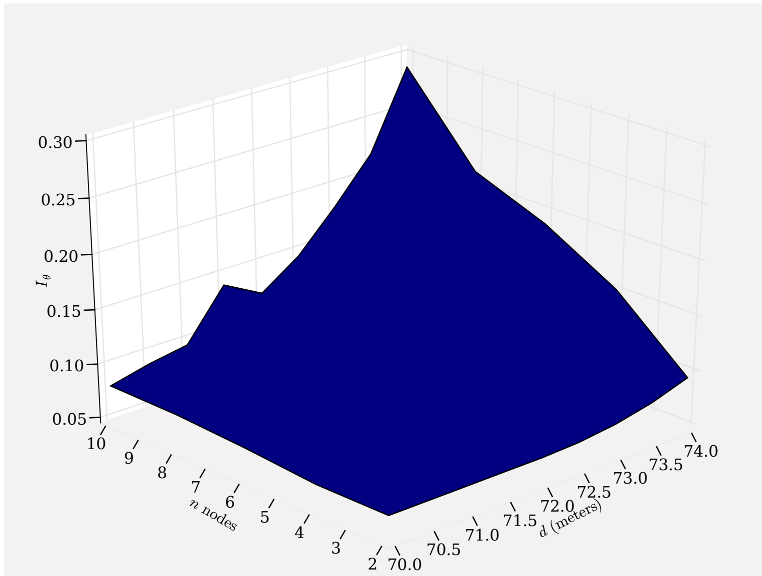

Figure 9. Distortion plot by dropping either the latest sample or current sample when FIFO is full.

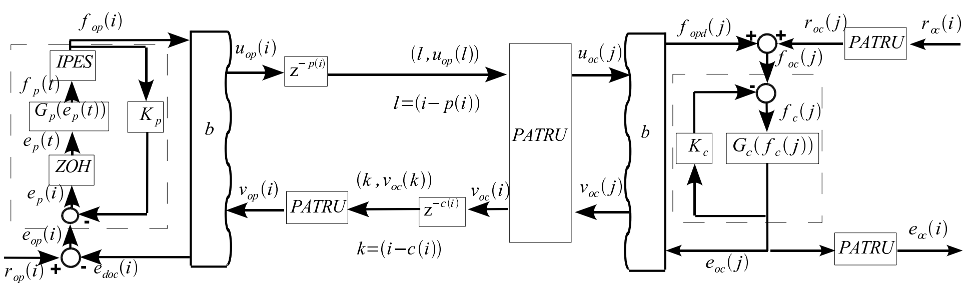

Figure 10. *Passive* digital control network with *Passive* Asynchronous Transfer Unit (*PATRU*).

$$u_{oc}(j) = u_{op}(l)$$

$$r_{oc}(j) = r_{oc}(i)$$

$$I = l \cup I$$

calculate new $v_{oc}(j),$ *and* $e_{oc}(j)$

$$v_{oc}(i) = v_{oc}(j)$$

$e_{oc}(i) = e_{oc}(j)$

$j = j + 1$

else:

$v_{oc}(i) = 0$

$e_{oc}(i) = 0$

transmit $(i, v_{oc}(i))$

2. *Otherwise if no periodically generated tuple* $(l, u_{op}(l))$ *from the plant has arrived to the PATRU on the controller side then:*

$v_{oc}(i) = 0$

$e_{oc}(i) = 0$

transmit $(i, v_{oc}(i))$

3. *If the periodically generated tuple* $(k, v_{oc}(k))$ *from the PATRU on the controller side has arrived to the PATRU on the plant side then:*

if $k \in J^c$*:*

$v_{op}(i) = v_{oc}(k)$

$J = k \cup J$

else:

$v_{op}(i) = 0$

transmit $(i, u_{op}(i))$

4. *Otherwise if no periodically generated tuple* $(k, v_{oc}(k))$ *from the PATRU on the controller side has arrived to the PATRU on the plant side then:*

$v_{op}(i) = 0$

transmit $(i, u_{op}(i))$

Copyright © 2008 John Wiley & Sons, Ltd.

Using definition 4 we give the following lemma:

Lemma 3. *Using the PATRU as defined by Definition 4,*

$$\langle f_{op}(i), e_{doc}(i)\rangle_{N_i} \geq \langle e_{oc}(j), f_{opd}(j)\rangle_{N_j} \tag{29}$$

holds for all N_i and N_j.

Proof 3. *To begin, we note that $N_i > N_j$ since the controller will only process received data from the plant. From the scattering transform we also know that (29) can be equivalently written as*

$$\|(u_{op}(i))_{N_i}\|_2^2 - \|(v_{op}(i))_{N_i}\|_2^2 \geq \|(u_{oc}(j))_{N_j}\|_2^2 - \|(v_{oc}(j))_{N_j}\|_2^2. \tag{30}$$

It is sufficient for (30) to hold if both

$$\|(u_{op}(i))_{N_i}\|_2^2 \geq \|(u_{oc}(j))_{N_j}\|_2^2 \tag{31}$$

and

$$\|(v_{oc}(j))_{N_j}\|_2^2 \geq \|(v_{op}(i))_{N_i}\|_2^2 \tag{32}$$

hold. By definition 4 we know that $u_{oc}(j)$ can only consist of unique samples of $u_{op}(i)$ therefore (31) is obviously satisfied. Likewise, $v_{op}(i)$ can only consist of unique samples of $v_{oc}(j)$ or the value 0 therefore (32) is satisfied.

With lemma 3 we state the additional lemma which shows that using the *PATRU* an expression can be obtained which is sufficient for a *strictly-output passive* system when $i = j$.

Lemma 4. *Using the PATRU as defined by Definition 4 in the control network depicted in Figure 10. The following inequality is satisfied:*

$$\langle f_{op}(i), r_{op}(i)\rangle_{N_i} + \langle e_{oc}(j), r_{oc}(j)\rangle_{N_j} \geq \epsilon(\|(f_{op}(i))_{N_i}\|_2^2 + \|(e_{oc}(j))_{N_j}\|_2^2) - \beta \tag{33}$$

in which $\epsilon = \min(\epsilon_{op}, \epsilon_{oc})$ and $\beta = \beta_{op} + \beta_{oc}$. When $i = j$ the network is strictly-output passive.

Copyright © 2008 John Wiley & Sons, Ltd.

The proof follows along the lines as the one provided for [7, Theorem 4].

Proof 4. *First, by [7, Theorem 3-I], G_p is transformed to a discrete passive plant. Next, by [7, Theorem 2] both the discrete plant and controller are transformed into strictly-output passive systems. The strictly-output passive plant satisfies*

$$\langle f_{op}(i), e_{op}(i)\rangle_{N_i} \geq \epsilon_{op}\|(f_{op}(i))_{N_i}\|_2^2 - \beta_{op} \tag{34}$$

while the strictly-output passive controller satisfies (35).

$$\langle e_{oc}(j), f_{oc}(j)\rangle_{N_j} \geq \epsilon_{oc}\|(e_{oc}(j))_{N_j}\|_2^2 - \beta_{oc} \tag{35}$$

Substituting, $e_{doc}(i) = r_{op}(i) - e_{op}(i)$ and $f_{opd}(j) = f_{oc}(j) - r_{oc}(j)$ into (29) (which holds by Lemma 3) yields

$$\langle f_{op}(i), r_{op}(i) - e_{op}(i)\rangle_{N_i} \geq \langle e_{oc}(j), f_{oc}(j) - r_{oc}(j)\rangle_{N_j}$$

which can be rewritten as

$$\langle f_{op}(i), r_{op}(i)\rangle_{N_i} + \langle e_{oc}(j), r_{oc}(j)\rangle_{N_j} \geq \langle f_{op}(i), e_{op}(i)\rangle_{N_i} + \langle e_{oc}(j), f_{oc}(j)\rangle_{N_j} \tag{36}$$

so that we can then substitute (34) and (35) to yield

$$\langle f_{op}(i), r_{op}(i)\rangle_{N_i} + \langle e_{oc}(j), r_{oc}(j)\rangle_{N_j} \geq \epsilon(\|(f_{op}(i))_{N_i}\|_2^2 + \|(e_{oc}(j))_{N_j}\|_2^2) - \beta \tag{37}$$

in which $\epsilon = \min(\epsilon_{op}, \epsilon_{oc})$ and $\beta = \beta_{op} + \beta_{oc}$. Thus (37) satisfies [7, Definition 3-II] when $i = j$ in which the input is the row vector of $[r_{op}, r_{oc}]$, and the output is the row vector $[f_{op}, e_{oc}]$.

Interestingly, we can describe the controllers behavior in terms of i since the *PATRU* only transfers data to the controller when available from the plant or sends a 0 to the plant when no control data is available. Equivalently we can simply transfer a 0 to the controller when no data is available from the plant and use a switched controller G_c in which $G_c = G_{co}$ when data is present from the plant and set $G_c = 0$ when the *PATRU* fills in the missing data for $v_{oc}(i),\ v_{op}(i),$ and $e_{oc}(i)$ with 0.

Copyright © 2008 John Wiley & Sons, Ltd.

Theorem 1. *Using the PATRU as defined by Definition 4 in the control network depicted in Figure 10. The digital control network in Figure 10 is strictly-output passive.*

Proof 5. *From lemma 4 we have shown that (33) holds. Given definition 4, both*

$$\|(e_{oc}(j))_{N_j}\|_2^2 = \|(e_{oc}(i))_{N_i}\|_2^2 \tag{38}$$

and

$$\langle e_{oc}(j), r_{oc}(j)\rangle_{N_j} = \langle e_{oc}(i), r_{oc}(i)\rangle_{N_i} \tag{39}$$

will hold, therefore,

$$\langle f_{op}(i), r_{op}(i)\rangle_{N_i} + \langle e_{oc}(i), r_{oc}(i)\rangle_{N_i} \geq \epsilon(\|(f_{op}(i))_{N_i}\|_2^2 + \|(e_{oc}(i))_{N_i}\|_2^2) - \beta \tag{40}$$

holds in which $\epsilon = \min(\epsilon_{op}, \epsilon_{oc})$ *and* $\beta = \beta_{op} + \beta_{oc}$.

4. Conclusions

We have presented several results related to digital control of continuous *passive* plants over wireless networks. Of particular importance is that we provided a much needed analysis which captured time varying delays (Lemma 1) and data dropouts (Lemma 2) for two way wireless digital communication token passing medium access control (*MAC*) networks. Also we showed how a *novel* asynchronous controller can be used to maintain an l^2-stable system (Theorem 1) while improving control performance over synchronous controllers which rely on lossy data reduction *LDR* algorithms. More specifically:

1. the *passive* plant (station $m = 2(n-1)$) and *passive* controller (station $m/2$) were treated as stations of a n node token ring network depicted in Figure 1,

Copyright © 2008 John Wiley & Sons, Ltd.

2. we provided a Markov chain (with transition matrix (5) and Lemma 1) in order to determine the network capacity, mean round trip travel time (τ_m) and variance of a packet of data in a ring network,

3. we account for the overhead of the data acknowledge, header, and frame control sequence with the corresponding definition for *data capacity* (Definition 2)

4. Definition 2 (and a useful analysis relating packet error rate to node spacing with wireless transceivers such as the CC2420 in [6, Appendix C.2]) allows one to generate figures such as:

   a) Figure 2 showing the maximum data capacity is attained by sending the longest possible packet until a distance spacing of the nodes is such that p is fairly low,

   b) and Figure 3 shows that a maximum spacing exists which provides the maximum data capacity $\times$ distance for relaying data over a network,

5. in order to account for queuing delays and data dropouts we provide a more precise networking delay model in Lemma 2,

6. Lemma 2 shows that the delay will undergo a *phase* shift in which the delay will suddenly increase at a critical number of nodes and node separation distance, as seen in Figure 4 and verified by simulation in Figure 5 such that

   a) the sudden increase in delay is equal to the maximum allowed buffer delay D and it occurs when the data rate $r < \frac{1}{\lambda_p} = \frac{2(n-1)}{p}$,

   b) this is intuitive since once data is generated at a rate that exceeds the capacity of the network, then the delay will continue to grow unbounded until packets are dropped which occurs when the First-In, First-Out (FIFO) buffer is full.

Copyright © 2008 John Wiley & Sons, Ltd.

In order to evaluate control performance over our *passive* wireless network we:

1. introduced a new definition for distortion Definition 3 which allowed us to evaluate and compare:
   a) an adaptive *LDR* algorithm as described in [6, Section 4.3.1] which has been shown to be *passive* [6, Lemma 8],
   b) a *novel strictly-output passive* asynchronous controller as depicted in Figure 10 which only computes a new possibly non-zero control command when valid data from the plant is received (Section 3.1.1)
2. provided a new Theorem 1 (with corresponding new Lemma 4, and Lemma 3) showing that the asynchronous controller, governed by the *PATRU* as defined by Definition 4, is indeed *strictly-output passive*,
3. Figure 9 shows the corresponding improvement in distortion for the asynchronous controller, as compared to the one which uses the adaptive compression scheme as shown in [6, Figure 4.18].

REFERENCES

1. P. Berestesky, N. Chopra, and M.W. Spong. Theory and experiments in bilateral teleoperation over the internet. In *Proceedings of the 2004 IEEE International Conference on Control Applications (IEEE Cat. No.04CH37596)*, volume Vol.1, pages 456 – 63, Taipei, Taiwan, 2004.
2. Paul Berestesky, Nikhil Chopra, and Mark W. Spong. Discrete time passivity in bilateral teleoperation over the internet. *Proceedings - IEEE International Conference on Robotics and Automation*, 2004(5):4557 – 4564, 2004.
3. M. Ergen, Duke Lee, Raja Sengupta, and P. Varaiya. Wtrp - wireless token ring protocol. *IEEE Transactions on Vehicular Technology*, 53(6):1863 – 81, 2004/11/.
4. X. Fan, K. Chandrayana, M. Arcak, S. Kalyanaraman, and J. T. Wen. A two-time scale design for detection and rectification of uncooperative network flows. *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005*, pages 1842–1847, December 12-15 2005.

Copyright © 2008 John Wiley & Sons, Ltd.

Prepared using rncauth.cls

5. Oscar Flärdh, Karl H. Johansson, and Mikael Johansson. A new feedback control mechanism for error correction in packet-switched networks. *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005*, pages 488–493, December 12-15 2005.

6. Nicholas Kottenstette. *Control of Passive Plants With Memoryless Nonlinearitites Over Wireless Networks*. PhD thesis, University of Notre Dame, August 2007.

7. Nicholas Kottenstette and Panos J. Antsaklis. Stable digital control networks for continuous passive plants subject to delays and data dropouts. *Proceedings of the 46th IEEE Conference on Decision and Control*, pages 4433 – 4440, 2007.

8. M. Kuschel, P. Kremer, S. Hirche, and M. Buss. Lossy data reduction methods for haptic telepresence systems. In *Proceedings. 2006 Conference on International Robotics and Automation (IEEE Cat. No. 06CH37729D)*, pages 2933 – 8, Orlando, FL, USA, 2006.

9. Kelvin K. Lee and Samuel T. Chanson. Packet loss probability for bursty wireless real-time traffic through delay model. *IEEE Transactions on Vehicular Technology*, 53(3):929 – 938, 2004.

10. F. L. Lewis. Optimal control. In W. S. Levine, editor, *The Control Handbook*, pages 759–778. CRC Press, Boca Raton, FL, 1996.

11. Ivana Maric and Roy D. Yates. Cooperative multicast for maximum network lifetime. *IEEE Journal on Selected Areas in Communications*, 23(1):127 – 135, 2005.

12. J. Rosen and B. Hannaford. A doc at a distance [remotely-controlled surgical robots]. *IEEE Spectrum*, 43(10):34 – 9, 2006/10/.

13. J. Laurie Snell. Finite markov chains and their applications. *The American Mathematical Monthly*, 66(2):99–104, February 1959.

14. Xiaoyi Wu and Christos G. Cassandras. A maximum time optimal control approach to routing in sensor networks. In *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005*, pages 1137–1142, 2005.

15. Min Xie and Martin Haenggi. Delay-reliability tradeoffs in wireless networked control systems. In *Workshop on Networked Embedded Sensing and Control*, Lecture Notes in Control and Information Sciences. Springer, October 2005.

16. Min Xie and Martin Haenggi. A study of the correlations between channel and traffic statistics in multihop networks. *To appear IEEE Transactions on Vehicular Technology*, 2007.

17. Min Xie and Martin Haenggi. Towards an end-to-end delay analysis of wireless multihop networks. In *submitted Ad Hoc Networks*. Elsevier, 2007.

Copyright © 2008 John Wiley & Sons, Ltd.

Prepared using rncauth.cls