Resource allocation and congestion control in task-oriented distributed sensor networks

Zhang, Jinsong, Kulasekere, Ernst, Premaratne, Kamal, Bauer, Peter

Jinsong Zhang, Ernst C. Kulasekere, Kamal Premaratne, Peter H. Bauer, "Resource allocation and congestion control in task-oriented distributed sensor networks," Proc. SPIE 4578, Fiber Optic Sensor Technology and Applications 2001, (14 February 2002); doi: 10.1117/12.456076

Event: Environmental and Industrial Sensing, 2001, Boston, MA, United States

Resource allocation and congestion control in task-oriented distributed sensor networks

J. Zhang[a], E.C.Kulasekere[b], K. Premaratne[a], and P.H. Bauer[c]

[a]University of Miami, Coral Gables, FL 33124 USA
[b]University of Moratuwa, Moratuwa SRI LANKA
[c]University of Notre Dame, Notre Dame, IN 46556 USA

ABSTRACT

Resource allocation and congestion control are two interrelated critical issues that arise in a task-oriented distributed sensor network. An effective resource management policy must account for these and their impact on the overall objectives of the network. In this paper, the viability of a *'virtual' per-flow* framework for addressing both resource allocation and congestion control in an integrated environment is demonstrated. In this framework, the resources being allocated to a physical buffer at a decision node are established by allocating and maintaining certain 'virtual' resources to each incoming data flow. The virtual per-flow framework allows the design of controllers for each link independently of the others thus enabling a decoupled analysis and allowing one to incorporate different delay models and nonlinearities for each input data link. The effectiveness of the per-flow strategy is demonstrated via the design of a robust H_∞-norm based feedback controller that ensures extremely good tracking of a dynamically changing set-point of a decision node buffer of a distributed sensor network. The design is robust against the time-varying and uncertain nature of network-induced delays.

Keywords: Delay system,robust control,uncertainty,distributed sensor network

1. INTRODUCTION

Distributed sensor networks (DSNs) utilize a variety of sensors that may be distributed logically, spatially, and geographically. Higher level nodes in a DSN can gather data from the lower level nodes and perform multisensor data fusion to obtain inferences that may not be possible from a single sensor.

In a highly dynamic environment where the observed 'scene' can change frequently and abruptly, the enormous amount of data generated by a DSN need to be processed in near real-time for effective detection, tracking, and/or identification of objects within the scene. Clearly, a DSN operating under such fast changing conditions requires a resource management scheme that allocates, establishes and maintains resources and avoids congestion at each node. In essence, a resource management scheme must incorporate and address the following concerns in an effective manner:

(a) Time-variant (TV) nature of network-induced delays and models that account for them: Recent work[1] has identified and proposed the models for essentially two types of TV delays that data may encounter in a DSN; further insight into these delays appear in the work of Sichitiu, et. al..[2]

(b) A scheme that allows each node of a DSN to dynamically allocate available bandwidth to its lower level nodes: Such dynamic allocation of bandwidth (according to perhaps how important the incoming data is) is essential for effective utilization of bandwidth and adaptability to fast scene changes and network faults.

ATM networks, for example, can support a wide variety of traffic and diverse classes of service through its ABR traffic class. Hence it is an ideal communication carrier for a DSN. A rate-based feedback control scheme is in fact in place for congestion control within the ABR traffic class of ATM.[3] However the rate calculation details were left open for the implementation stage. Effective network buffer set-point control strategies have since appeared in the literature.[4–6] However, most of these do not address the concerns (a) and (b) effectively. Application of these control strategies while ignoring the issues raised in (a) and (b) may provide results that are completely unsatisfactory in a DSN environment.

In this paper, the efficiency of a virtual per-flow framework which enables the resource management of a DSN to be easily implemented and simplifies the congestion controller design is demonstrated. With the

Fiber Optic Sensor Technology and Applications 2001, Michael A. Marcus, Brian Culshaw, Editors,
Proceedings of SPIE Vol. 4578 (2002) © 2002 SPIE · 0277-786X/02/$15.00

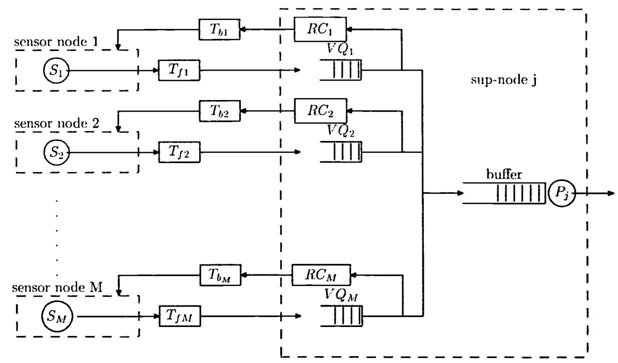

Figure 1: The virtual per-flow queuing framework.

derivation of an H_∞-design based F-norm (Frobenius-norm) uncertainty bounded controller, it is shown that the buffer level of a supervisory node (sup-node) can be tightly maintained at a set-point that is dynamically selected according to perhaps how important the data is perceived to be. This presentation is organized as follows: Section 2 introduces the virtual per-flow framework as a tool for DSN congestion control; Section 3 discusses the models of delays encountered in a DSN environment; Section 4 contains the derivation of the new controller that is robust against the types of multiple TV delays present in a DSN; Section 5 is reserved for a simulation example to show the effectiveness of the controller derived previously; Section 6 contains some concluding remarks and suggestions for future work.

2. THE VIRTUAL PER-FLOW FRAMEWORK

In a highly dynamic DSN environment, a mechanism that facilitates nodes at one hierarchical level to allocate system resources dynamically and treat each sensor node separately would be highly desirable and a convenient analytical tool. The *virtual per-flow framework* in Figure 1 has been proposed to fulfill these requirements.[7] In this framework, a 'virtual' buffer is assigned to each incoming flow. It is assumed that data cells from each flow gets queued up in its own virtual buffer before it goes to the bottleneck sup-node buffer.

In Figure 1, for $i = \overline{1, M}$, S_i are sensors, T_{f_i}, are forward delays, T_{b_i} are backward delays, RC_i are rate controllers located at the sup-node that are expected to dictate the instantaneous sensor rate, and VQ_i are virtual buffers. In general, both forward and backward delays are TV; these are studied in further detail in Section .

It is our claim that, with this per-flow queuing framework, set-point control of the buffer at sup-node can be realized by controlling the buffer level of each of the virtual buffers. The bandwidth and buffer set-point the sup-node allocates to each flow would depend on how important it perceives the information from the corresponding sensor is. One may for example utilize a suitably defined *importance measure*[7] for this purpose. Indeed, suppose the importance measure of sensor node i assigned by its sup-node at the higher level node is w_i where $\sum_{i=1}^{M} w_i = 1$. It is our contention that a reasonable allocation of the depletion rate b_i and set-point level Q_i of the virtual buffer corresponding to sensor node i must then conform to $b_i = w_i b$ and $Q_i = w_i Q$. The controller design carried out in this paper is based on this strategy. The advantages offered by the proposed per-flow framework are as follows:

(i) System resource allocation algorithm can be seamlessly implemented by distributing available system bandwidth and buffer level set-point to the flow of each sensor nodes.

(ii) The decoupling of the control loops make the analysis of the system easier.

(iii) TV delays of different characteristics associated with the flow of each loop can be easily modeled and accounted for.

(iv) It can be extended to end-to-end congestion control in a straightforward manner.

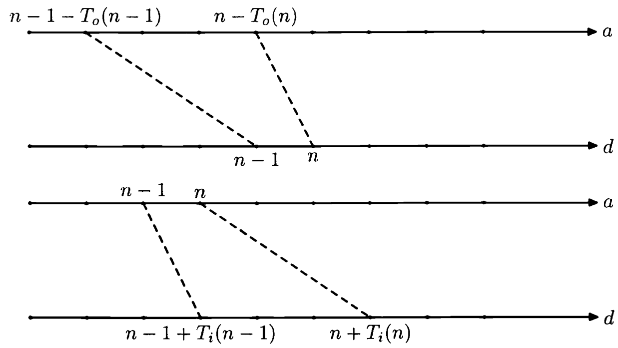

Figure 2: Output and input delay models.

3. TV DELAYS: TYPES AND MODELS

Two distinct types of TV delays that may occur in a discrete-time system have been previously identified.[1] One type of delay is experienced by the data flow such as that is present in the forward path from a sensor to a sup-node; it is called a *forward delay* and is denoted by $T_f(n)$. The second type of delay is caused by the delay between the instant the control rate information is generated at a sup-node and the instant when this rate takes effect at a sensor node. Since the control rate information is usually inserted into a control cell and sent along the backward path from a sup-node to a sensor node, it is called a *backward delay* and is denoted by $T_b(n)$. Both of them are in general highly TV and taken to be so in this paper.

3.1. Input and output delay models

Depending on when the value of delay is 'determined,' two types of models can be used to describe the behavior of a TV delay element. When the delay value is pre-determined at the instant data enters the delay element, it can be modeled via the *input delay model* $d(n + T_i(n)) = a(n)$. When the delay value is determined only at the instant data exits the delay element, it can be modeled via the *output delay model* $d(n) = a(n - T_o(n))$. Figure 2 further clarifies these situations.

3.2. Forward and backward delay types

3.2.1. Forward delay type

Forward delay affects data flow in the forward path. Due to its TV nature, the fact that data cells are being transmitted at each time instant from a source node is no guarantee that data cells are received at each time instant at a destination node. Indeed, it is quite possible that no data cell arrives at a destination node at certain time instants. An input delay model is an appropriate tool to describe such a situation, i.e., $D(k+T_f(k)) = A(k)$, where $A(k)$ and $D(k)$ are the accumulated number of cells that the forward delay element receives and transmits respectively within time interval $[0, k]$.

A corresponding discrete-time state-space model for this forward delay type may be obtained as follows: Suppose the change of variables $k + T_f(k) = n$ has l solutions k_i, $i = \overline{1, l}$, for a given n. If $l = 0$, no cells arrive at the destination node at time n and we set $\hat{T}_f(n) = \hat{T}_f(n-1) + 1$; if $l \neq 0$, let k_n be the solution that corresponds to a minimum value of $T_f(k_n)$ and denote this delay as $\hat{T}_f(n)$. Then an output delay model can be used to describe the forward delay type as $D(n) = A(n - \hat{T}_f(n))$. We may now write

$$\sum_{i=0}^{n} d(i) = \sum_{i=0}^{n-\hat{T}_f(n)} a(i); \quad \sum_{i=0}^{n-1} d(i) = \sum_{i=0}^{n-1-\hat{T}_f(n-1)} a(i), \tag{1}$$

where $a(i)$ and $d(i)$ are respectively the rates of data flow at the input and output of the forward delay element at time i. From (1) we get $d(n) = \sum_{n-1-\hat{T}_f(n-1)+1}^{n-\hat{T}_f(n)} a(i)$. We may express this as[1] $d(n) = \sum_{i=1}^{L_f} \beta_i(n-i)\, a(n-i)$,

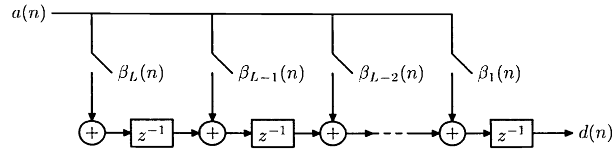

Figure 3: A realization of the forward delay type.

where L_f denotes the maximum value of forward delay and

$$\beta_i(n) = \begin{cases} 1, & \text{for } T_f(n) = i; \\ 0, & \text{elsewhere.} \end{cases} \tag{2}$$

A state-space realization of this appears in Figure 3; a corresponding state-space model is $\{A_f, B_f(n), C_f, D_f\}$ where

$$A_f = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}; \quad B_f(n) = \begin{bmatrix} \beta_1(n) \\ \beta_2(n) \\ \vdots \\ \beta_{L_f}(n) \end{bmatrix};$$

$$C_f = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \end{bmatrix}; \qquad D_f = \begin{bmatrix} 0 \end{bmatrix}.$$

We have assumed that the minimum possible value of delay is 1*.

3.2.2. Backward delay type

Backward delay is the delay experienced by control cells which, for example, carry control rate information. It affects system behavior in a very different manner than the forward delay due to the strategies employed in generating control rate information and control cells:

(a) New control rate information is generated at each time instant.

(b) Control cells are generated periodically with respect to the number of data cells and *not* periodically with time. Hence there may not be a control cell ready and waiting to carry the new control rate information that is being generated at each time instant. The resulting delay is referred to as the *rate mismatch delay.*

In the absence of a new control cell, the sensor node may continue using the rate that was specified by the control cell it last received†. It is quite clear that an input delay model is not suitable to describe this backward delay; what captures its characteristics is the output delay model: $d(n) = a(n - T_b(n))$. We may express this as[1] $d(n) = \sum_{i=1}^{L_b} \alpha_i(n) a(n-i)$, where L_b denotes the maximum value of backward delay and

$$\alpha_i(n) = \begin{cases} 1, & \text{if } T_b(n) = i; \\ 0, & \text{elsewhere.} \end{cases} \tag{3}$$

A state-space realization of this appears in Figure 4; a corresponding state-space model is $\{A_b, B_b, C_b(n), D_b\}$

*In a discrete-time setting, the delay in fact cannot be zero thus making this assumption justifiable.

†Another very reasonable strategy would be for the sensor node to assume network congestion when it does not receive a control cell within one time instant and reduce its rate accordingly. The delay model must then be modified accordingly.

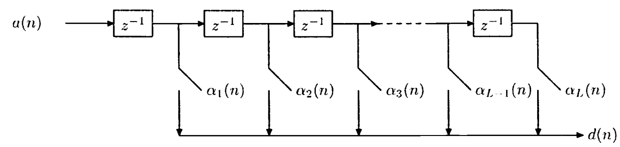

Figure 4: A realization of the backward delay type.

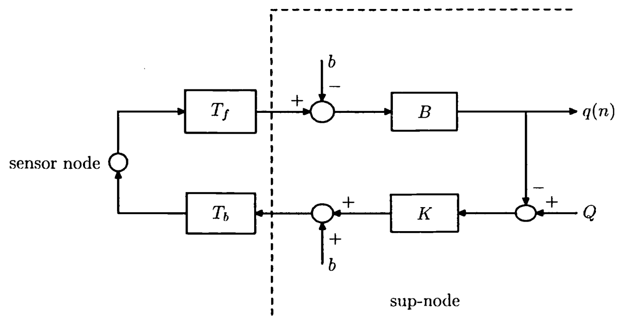

Figure 5: Per-flow Congestion control of a sensor/sup-node virtual loop.

where

$$A_b = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 1 & 0 \end{bmatrix}; \qquad B_b = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix};$$
$$C_b(n) = \begin{bmatrix} \alpha_1(n) & \alpha_2(n) & \dots & \alpha_{L_b}(n) \end{bmatrix}; \quad D_b = \begin{bmatrix} 0 \end{bmatrix}.$$

As before, we have assumed that the minimum possible value of delay is 1.

4. ROBUST CONTROLLER DESIGN

The per-flow framework allows one to design the controller of each loop independently of the the others. Hence, we focus on the controller design of a single loop.

4.1. Virtual per-flow congestion control model

With respect to the congestion control problem, each sensor/sup-node virtual loop may be modeled as in Figure 5. The buffer level we desire to control is located at the sup-node; $T_f(n)$ is the forward delay block with the state-space realization $\{A_f, B_f(n), C_f, D_f\}$ and $T_b(n)$ is the backward delay block with the state-space realization $\{A_b, B_b, C_b(n), D_b\}$. B is the buffer at sup-node modeled by a discrete-time integrator; K is the controller which is taken to be located at the sup-node; $Q(n)$ is the buffer set-point level; $q(n)$ is the actual buffer level at time instant n; $b(n)$ is the buffer depletion rate. We intend to utilize H_∞ techniques to design the congestion controller. For this purpose, we first need to express the system in Figure 5 as in Figure 6. Here, as is usual within the H_∞ framework, P denotes the open-loop system, K denotes the controller, z denotes the controlled output signal, $w(n) = \begin{bmatrix} b(n) & Q(n) \end{bmatrix}^T$ denotes the disturbance input signal, and u is the control input signal.

One may now show that the complete system in Figure 5 possesses the following state-space realization:

$$\begin{aligned} x(n+1) &= A_\Delta(n)\, x(n) + B_1 w(n) + B_2 u(n); \\ z(n) &= C_1 x(n) + D_{11} w(n) + D_{12} u(n); \\ y(n) &= C_2 x(n) + D_{21} w(n) + D_{22} u(n), \end{aligned} \tag{4}$$

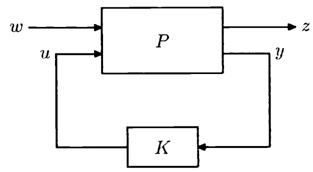

Figure 6: Per-flow congestion controller design within the H_∞ framework.

where

$$A_\Delta(n) = \begin{bmatrix} 1 & C_f & D_f C_b \\ \mathbf{0} & A_f & B_f C_b \\ \mathbf{0} & \mathbf{0} & A_b \end{bmatrix}; \; B_1 = \begin{bmatrix} -1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix};$$

$$B_2 = \begin{bmatrix} D_f D_b \\ B_f D_b \\ B_b \end{bmatrix} = \begin{bmatrix} 0 & 0 \ldots & 0 & 1 & 0 \ldots & 0 \end{bmatrix}^T,$$

and

$$C_1 = \begin{bmatrix} 1 & \mathbf{0} & \mathbf{0} \end{bmatrix}; \quad D_{11} = \begin{bmatrix} 0 & 1 \end{bmatrix}; \; D_{12} = 1;$$
$$C_2 = \begin{bmatrix} -1 & \mathbf{0} & \mathbf{0} \end{bmatrix}; \; D_{21} = \begin{bmatrix} 0 & 1 \end{bmatrix}; \; D_{22} = 0.$$

Also, $x(n) = \begin{bmatrix} q(n) & x_f(n) & x_b(n) \end{bmatrix}^T$ and $w(n) = \begin{bmatrix} b(n) & Q(n) \end{bmatrix}^T$, where $x_f(n)$ and $x_b(n)$ are the state vectors contributed by the forward and backward delay elements respectively.

Note that (4) describes a linear system with only the system matrix $A_\Delta(n)$ possessing TV parameters. The latter are due to the presence of $B_f C_b(n)$. Let $B_f C_b(n) = U(n) = \{u_{ij}(n)\}$ where $u_{ij}(n) = \beta_i(n)\,\alpha_{L_b-j+1}(n) \in \{0,1\}$; $\beta_i(n)$ and $\alpha_i(n)$ are as in (2) and (3) respectively. When the forward and backward delays are within the ranges $[1, L_f]$ and $[1, L_b]$ respectively, the matrix $U(n)$ will possess the property

$$\sum_i \sum_j u_{ij}(n) \le L_f. \tag{5}$$

4.2. H_∞ sub-optimal controller design

The system matrix $A_\Delta(n)$ being a TV uncertain matrix subject to the constraint (5), the congestion control problem under consideration can be recast into a robust control problem of a class of discrete-time linear systems with TV parameter uncertainties. The problem of robust stability and control of this class of systems has been extensively studied in the past few years where much attention has been focused on the system being subject to ∞-norm bounded parameter uncertainties. Both state and output feedback controllers have been suggested by Carlos, et. al.,[8] Xie,[9] and Yuan, et. al.[10] Recently, the case of F-norm bounded parameter uncertainties was discussed by Boukas and Shi[11] where a state feedback controller was derived. Stimulated by this latter work, here we model our system as a discrete-time linear system with F-norm bounded parameter uncertainties and derive a corresponding output feedback controller. Let

$$A_\Delta(n) = A + \Delta A(n),$$

where

$$A = \begin{bmatrix} 1 & C_f & D_f C_b \\ \mathbf{0} & A_f & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & A_b \end{bmatrix};$$

$$\Delta A(n) = \begin{bmatrix} 0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & U(n) \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \equiv M\,\Delta(n)\,N,$$

with

$$M = \begin{bmatrix} \mathbf{0} \\ \sqrt{L_f} I \\ \mathbf{0} \end{bmatrix}; \quad N = \begin{bmatrix} \mathbf{0} & I & \mathbf{0} \end{bmatrix};$$

$$\Delta(n) = \{\Delta_{ij}\},\ i = \overline{1, L_f},\ j = \overline{1, L_b}.$$

Here

$$\|\Delta(n)\|_F^2 = \sum_i \sum_j |\Delta_{ij}(n)|^2 \leq 1. \tag{6}$$

Now the system model in (4) may be expressed as

$$\begin{aligned} x(n+1) &= [A + \Delta A(n)]\, x(n) + B_1 w(n) + B_2 u(n); \\ z(n) &= C_1 x(n) + D_{11} w(n) + D_{12} u(n); \\ y(n) &= C_2 x(n) + D_{21} w(n). \end{aligned} \tag{7}$$

The matrix $\Delta A(n) = M\, \Delta(n)\, N$ contains uncertain parameters with $\Delta(n)$ satisfying (6) while all the other matrices are time-invariant (TI).

Associated with the system in (7), we introduce the following scaled linear discrete-time system:

$$\begin{aligned} x(n+1) &= A\, x(n) + B_{1a} w_a(n) + B_2 u(n); \\ z_a(n) &= C_{1a} x(n) + D_{11a} w_a(n) + D_{12a} u(n) \\ y(n) &= C_2 x(n) + D_{2a} w_a(n), \end{aligned} \tag{8}$$

where

$$B_{1a} = \begin{bmatrix} B_1 & \varepsilon M \end{bmatrix}; \quad C_{1a} = \begin{bmatrix} C_1 \\ \varepsilon^{-1} N \end{bmatrix};$$

$$D_{11a} = \begin{bmatrix} D_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}; \quad D_{12a} = \begin{bmatrix} D_{12} \\ \mathbf{0} \end{bmatrix};$$

$$D_{2a} = \begin{bmatrix} D_{21} & \mathbf{0} \end{bmatrix}.$$

Here $\varepsilon > 0$ is a scaling parameter. Then, following the work of Xie[9] and Boukas,[11] we have

THEOREM 4.1. *The system (7) is quadratically stabilizable with H_∞ disturbance attenuation γ by a linear dynamic output feedback controller K iff, for some $\varepsilon > 0$, the closed-loop system in (8) with the same controller K is stable with H_∞ disturbance attenuation γ.*

With this result in hand, the problem of H_∞ controller design for the linear discrete-time system with TV parameter uncertainties can be recast into a standard H_∞ synthesis for linear discrete-time systems without parameter uncertainty but with a scaling parameter to be chosen. Standard methods are available to solve this problem.

5. EXAMPLE

To illustrate the controller design strategy proposed in this paper, we simulate a simple scenario with three sensors. In each loop, both forward and backward delays can be set as TI or TV. The controller 'sampling' interval is $T_s = 1$ ms; the maximum values of $T_f(n)$ and $T_b(n)$ are chosen as $L_b = L_f = 5T_s$; the buffer set-point trajectory is chosen to be a two-valued (5000/6000) step function; the three sensors are assigned the weights 0.1 : 0.3 : 0.6; the depletion rate is $b = 500$ cells/s. The simulation results in Figures 8-9 show that the proposed controller tracks the set-point trajectory quite satisfactorily in spite of the TV delays (see Figure 7) thus demonstrating its effectiveness. The results also show an interesting phenomenon: when the forward delay is TV (the backward delay can either be TI or TV), a steady-state equilibrium is not being maintained. This problem was in fact pointed out by Sichitiu, et. al.[2] However, the H_∞ design proposed in this current work simply ensures boundedness of buffer level (which of course the simulation verifies).

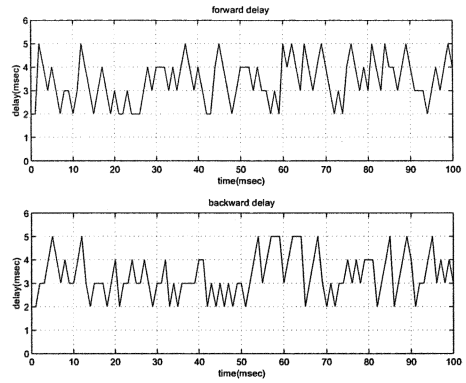

Figure 7: TV forward and backward delay traces utilized in the simulations.

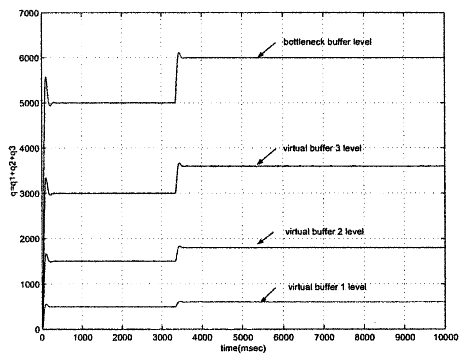

Figure 8: Multi-loop buffer set-point tracking.

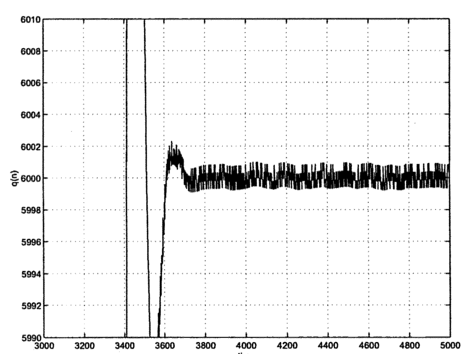

Figure 9: Buffer set-point tracking with TV forward delay.

6. CONCLUSION

The virtual per-flow framework enables one to track the bottleneck buffer set-point level via the maintenance of an appropriate set-point level at each of its corresponding virtual buffers. Maintenance of the buffer set-point at a constant (or almost constant) level assures avoidance of congestion and reduction of delay jitter generated at the buffer. Furthermore, the proposed controller design strategy can also track a dynamically changing set-point level thus enabling a dynamic DSN resource allocation to be implemented. In addition to its ability to track the set-point trajectory, the controller derived in this paper is robust against network-induced delays that are TV. In practice, a sensor may not be able to meet the rate requested by the sup-node due to its inherent limitations; for example, it may have a maximum rate beyond which no date can be generated. The arising nonlinearity has not been incorporated in this present work and it would be interesting to study its effect on the controller that has been proposed.

ACKNOWLEDGMENTS

The authors acknowledge the support received via the NSF Grant numbers 9726247 and 9726253.The work of E.C.Kulasekere was carried out while he was at the University of Miami.

REFERENCES

1. M.M. Ekanayake, *Robust Stability of Discrete-Time Nonlinear Systems.* PhD thesis, University of Miami, Coral Gables, FL, May 1999.
2. M.L. Sichitiu, P.H. Bauer, and K. Premaratne, "The effect of uncertain time variant delay in atm networks with explicit rate feedback: A control theoretic approach.," *IEEE/ACM Trans. Netw.* , in review.
3. ATM Forum, "The ATM Forum traffic management specifications, version 4.0," tech. rep., ATM Forum, Apr. 1996.
4. A.K. Parekh and R.G. Gallager, "A generalized processor sharing approach to flow control in integrated service network: the single-node case," *IEEE/ACM Trans. Netw.* **1**, pp. 344–357, 1993.
5. A. Kolarov and G. Ramamurthy, "A control-theortic approach to the design of an explicit rate controller for ABR service," *IEEE/ACM Trans. Netw.* **7**, pp. 741–753, 1999.
6. L. Benmohamed and S.M. Meekov, "Feedback control of congestion in packet switching networks: the case of single congested node," *IEEE/ACM Trans. Netw.* **1**, pp. 693–708, 1993.
7. J.S. Zhang, E.C. Kulasekere, K. Premaratne, and P.H. Bauer, "Resource management of task oriented distributed sensor networks," in *IEEE Int. Symp. Circ. Syst. (ISCAS'01)*, **III**, pp. 513–516, (Sydney, Australia), 2001.
8. C.E. de Souza and M.Y. Fu and L.H. Xie, "H_∞ analysis and synthesis of discrete-time systems with time-varying uncertainty," *IEEE Trans. Auto. Cont.* **38**, pp. 459–462, Mar. 1993.
9. L. Xie, "Output feedback H_∞ control of systems with parameter uncertainty," *Int. J. Cont.* **63**, pp. 741–750, 1996.
10. L.S. Yuan, E.K.A. Luke, and W.S. Jiang, "Robust H_∞ control for linear discrete-time systems with norm-bounded time varying uncertainty," *Syst. Cont. Lett.* **27**, pp. 199–208, 1996.
11. E.-K. Boukas and P. Shi, "H_∞ control for discrete-time linear systems with Frobenius norm-bounded uncertainty," *Automatica* **35**, pp. 557–561, 1999.