Turkish Journal of Electrical Engineering and Computer Sciences

Volume 12 | Number 3 Article 3

1-1-2004

Flow Controller Design and Performance Analysis for Self-Similar Network Traffic

PENG YAN

Turk J Elec Engin, VOL.12, NO.3 2004, © TÜBİTAK

Flow Controller Design and Performance Analysis for Self-Similar Network Traffic*

Peng YAN
Dept. of Electrical & Computer Engineering, The Ohio State University
2015 Neil Avenue, Columbus, OH 43210
e-mail: yanp@ece.osu.edu

Abstract

Recent studies of high-resolution traffic measurement discovered the self-similarity in both LAN and WAN traffic. In this paper, we introduce a two-degree of freedom rate based flow controller, which includes a robust H^∞ control block and an LMMSE based capacity predictor. The former part can guarantee the robust stability against time-varying time delay uncertainties and the latter improves the transient response by predicting the self-similar cross-traffic. Implementation issues are discussed and performance analysis is provided to validate our design. We also investigate the prediction and control in larger time scale which is more applicable for the real network environment.

1. Introduction

Congestion control for traffic management is one of the main concerns in the design of modern high speed communication networks. For best-effort services, flow control mechanisms are implemented to ensure a good Quality of Service (QoS) by means of regulating the data rate generated by the sources[5][9].

In ATM ABR (Available Bit Rate) services, the rate-based flow control has been chosen by the ATM forum[1], which allows for the intermediate nodes to adjust a data rate to ABR sources based on network feedback; this problem can be viewed as a feedback control problem where the queue length is used as explicit feedback[2][3][10]. In [3], for example, the problem was formulated as an LQG stochastic control problem with action delays, and the information of available rate as well as the queue length was used to compute the sending rate for each ABR source. In [2], the authors studied the problem in a game-theoretic context where each user minimized its own performance objective.

A challenging aspect of flow control is the presence of round trip time delays (RTT) which consist of forward delays and backward delays. The problem is further complicated due to the fact that these time delays are usually time-varying and uncertain because of data buffering at the bottleneck nodes. For such time delay systems, the $\mathcal{H}^\infty$ based robust control techniques were discussed in [12][13][4], where the robustness against delay uncertainties, weighted fairness, as well as tracking performance were considered in the framework of $\mathcal{H}^\infty$ optimization for infinite dimensional systems. Internally robust implementation of

*A shorter version appeared in the Proceedings of the 39th Annual Allerton Conference on Communication, Control, and Computing, Monticello, Illinois USA, October 2001. This work is supported in part by the National Science Foundation under grant numbers ANI-9806660, ANI-0073725, and by a SBC/Ameritech Faculty Research Grant.

this controller was discussed in [13], and the case of multiple time-delays at different channels was discussed in [4].

In a more recent result [15], the authors investigated a two-degree of freedom $\mathcal{H}^\infty$ flow controller design, which used capacity information as well as queue length as feedback. This controller, as demonstrated by the simulations of [15], had faster transient response with respect to time-varying available bandwidth. The available bandwidth was assumed to be generated by applying an unknown $\mathcal{L}_2$ signal to a known low-pass filter. Thus they designed an $\mathcal{H}^\infty$ based capacity predictor, because it was the most natural for this kind of disturbance.

However, the available bandwidth is actually a random variable determined by the network cross traffic. Recent statistical analysis of high-resolution traffic measurement have revealed the self-similar behaviors of network traffic [16][19]. Thus the available capacity $c(t) = c_0 - c_{cross}(t)$ also exhibits LRD (Long-Range-Dependence), which is not in the form considered in [15]. Here c_0 is the full link capacity and $c_{cross}(t)$ is the capacity taken out by the cross traffic. The Long-Range-Dependence can not be described by typical statistical models, such as Gaussian, ARMA, or the $\mathcal{L}_2$ signals passing a known low-pass filter [15]. This is the motivation for us to investigate the flow control problem for LRD network traffic. With the current state of the art in robust control of time varying time delay systems with self-similar disturbances, it is impossible to pose an optimal robust performance problem that is tractable. In this work, we assume the LRD cross traffic and design an LMMSE (Linear Minimum Mean Square Error) predictor for available bandwidth estimation. In fact, it has been demonstrated in [7] that the causal linear predictors are quite efficient for self-similar signals. The LMMSE is used with the $\mathcal{H}^\infty$ controller designed for robustness against the time-varying time delay uncertainty[4].

We also consider a more realistic case where control and prediction are in a larger time scale. In fact, for ATM networks, once every N cells the sources send a control cell which can be used by switches to convey feedback. And for TCP/IP networks, each source updates its sending window size once every RTT when an ACK is received[8]. For this scenario, implementation issues are discussed, and the performance is analyzed in the sense of the multifractal nature of cross traffic as well as the robustness of our controller. All the above results are demonstrated by MATLAB simulations, where the cross traffic comes from the data traces generated by packet-level simulations using *ns-2*.

2. Preliminary definitions

The measurements of LAN or WAN traffic have revealed the presence of self-similarity or fractal behavior. The most broadly studied fractal model is the *fractal Brownian motion* (fBm), whose increment process $G(k) := B((k+1)\Delta) - B(k\Delta)$ is called *fractional Gaussian noise* (fGn).

Heavy-Tailedness: A random variable X is called heavy-tailed if $Pr\{X > x\} \sim x^{-\alpha}$ as $x \to \infty$, where $0 < \alpha < 2$. A simple heavy-tailed distribution is the *Pareto* distribution whose probability density function is given by:

$$p(x) = \alpha w^\alpha x^{-\alpha-1} \tag{1}$$

where α is the shape parameter and $w > 0$ is the location parameter. Its distribution function obeys $F(x) = 1 - (w/x)^\alpha$. Note that smaller α means heavier tail of the distribution. In network models, we are interested in choosing $1 < \alpha < 2$, [14].

Self-Similarity: Consider a wide-sense stationary (W.S.S.) time series $X_t, t \in \mathbb{Z}$. The aggregated series $X_t^{(n)}$ is obtained by averaging X_t over non-overlapping blocks of length n and replacing each block by its mean:

$$X_t^{(n)} = \frac{X_{tn-n+1} + \cdots + X_{tn}}{n}. \tag{2}$$

Denote the auto-covariance of X_t and $X_t^{(n)}$ by $r_X(k)$ and $r_X^{(n)}(k)$ respectively. We say that X_t is self-similar if the following conditions hold:

$$r_X(k) \sim const \cdot k^{-\beta}, \quad \text{and} \quad r_X^{(n)} \sim r_X(k) \tag{3}$$

for $k, n \to \infty$ where $0 < \beta < 1$. That is, the correlation structure is preserved at different time scales and $r_X(k)$ decays according to a power-law. The Hurst parameter is denoted by $H = 1 - \beta/2$, where $1/2 < H < 1$. The Hurst parameter is an indicator of the degree of self-similarity in the sense that larger H means stronger LRD.

Long Range Dependence: X_t is said to exhibit LRD in that its auto-covariance decays hyperbolically with $\sum_{k=0}^{\infty} r_X(k) = \infty$, which is implied by (3). One example of an LRD process is fGn.

3. Mathematical Model

The congestion control problem to be considered here is for queue management and bandwidth allocation at the bottleneck node, where the output capacity is determined by self-similar cross traffic. For simplicity, consider the single bottleneck node network with one source, the model of which can be depicted in Figure 1.

The dynamics of the system can be modeled as:

$$\dot{q}(t) = r_b(t) - c(t) = r(t - \tau) - c(t) \tag{4}$$

where $q(t)$ is the queue length at time t, $r_b(t)$ is the data rate received at the bottleneck node, and $r(t) \geq 0$ is the flow rate assigned by the controller. The round-trip delay (RTT), $\tau(t)$ is defined as:

$$\tau(t) = \tau_b(t) + \tau_f(t) \tag{5}$$

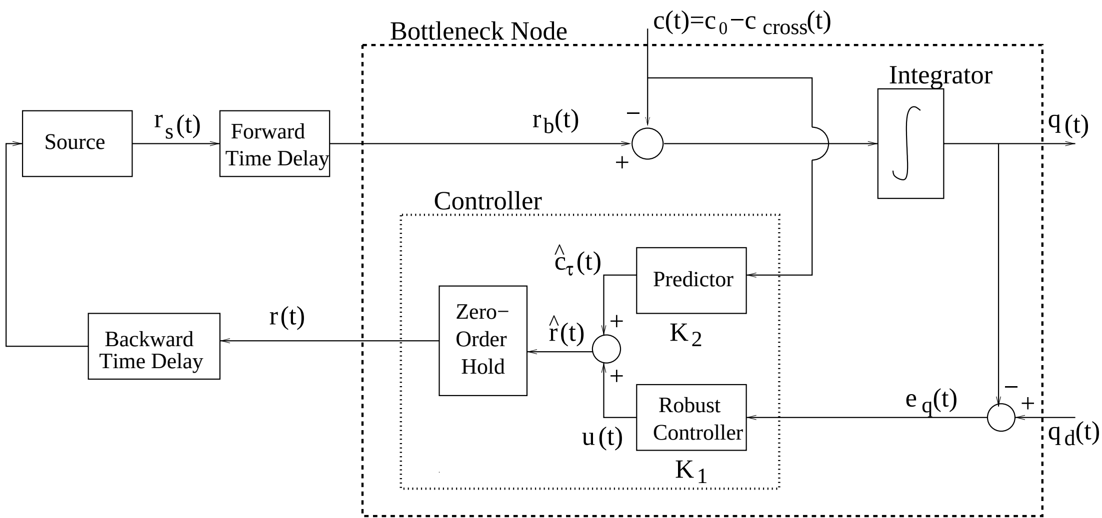

Figure 1. LMMSE based feedback control system.

where $\tau_f(t) := h_f + \delta_f(t)$ is the forward time delay from the source node to the bottleneck node where h_f is the nominal forward delay and $\delta_f(t)$ is the time-varying forward time delay uncertainty. Similarly, $\tau_b(t) := h_b + \delta_b(t)$ is the backward time delay from the bottleneck node (controller output) to the source node where h_b is the nominal backward delay and $\delta_b(t)$ is the time-varying backward time delay uncertainty.

Thus, the total nominal time-delay can be defined as $h = h_b + h_f$, and the total time delay uncertainty is $\delta(t) = \delta_b(t) + \delta_f(t)$.

Our two-degree of freedom flow controller consists of a stochastic predictor and a robust control block, together with a zero-order hold (ZOH). In digital implementation, the ZOH holds $\hat{r}(t)$ for n-step, which means a larger time scale prediction and control at the scale nT_s, where T_s is the sampling period. In the reasoning of this section, we set $n = 1$ for simplicity. Thus we have $r(t) = \hat{r}(t)$. This leads to:

$$\dot{q}(t) = \hat{r}(t-\tau) - c(t) \approx u(t-\tau) - (c(t) - \hat{c}_h(t-\tau)) \tag{6}$$

The cross traffic $c_{cross}(t)$ is assumed to be stochastically self-similar. Thus the output capacity $c(t) = c_0 - c_{cross}(t)$ also exhibits self-similarity. We introduce the LMMSE based predictor, the output of which is a causal estimate of $c(t+\tau)$, namely $\hat{c}_\tau(t)$, or $\hat{c}_h(t)$ for simplicity. Note that by omitting the time delay uncertainties, a good predictor can act as a compensator by which the impact of $c(t)$ can be canceled. In other words, $e_c(t) := c(t) - \hat{c}_h(t-\tau)$ should be small enough. Details of the predictor design and validation are discussed in the next section.

In this sense, we would like to have $\dot{q}(t) = u(t-\tau)$. Substitute τ with its nominal value h, and consider the frequency domain expression, we will have:

$$Q(s) = \frac{e^{-hs}}{s}U(s) = \frac{e^{-hs}}{s}K_1(s)E_q(s) \tag{7}$$

where

$$E_q(s) = Q_d(s) - Q(s) \tag{8}$$

where $q_d(t)$ is the desired queue length and $Q_d(s)$ is its frequency domain expression. The objectives in the design of robust control block K_1 are:

- Robust stability against the time-varying time delay uncertainty $\delta(t)$
- Queue management, e.g. minimizing $\|e_q\|_2$

Combining the robust stability and nominal performance conditions (e.g. tracking, weighted fairness, and transient response), we come up with an infinite dimensional $\mathcal{H}^\infty$ optimization problem as follows:

Minimize γ, such that K_1 is stabilizing P_0 and

$$\left\| \begin{bmatrix} W_s(1+P_0K_1)^{-1} \\ WK_1(1+P_0K_1)^{-1} \end{bmatrix} \right\|_\infty \leq \gamma \tag{9}$$

where $P_0(s) = \frac{e^{-hs}}{s}$, and W, W_s are chosen to meet desired robustness and performance respectively. See [12][13] for details.

The optimal controller K_1 is computed in [4]:

$$K_1(s) = \frac{\hat{\gamma}}{h\hat{\delta}}\frac{hs-k}{hs}\frac{1}{1+F(hs)} \tag{10}$$

where $\hat{\gamma}, \hat{\delta}, k$ are determined by the upper bound of delay uncertainty δ^+, the upper bound on the derivative of delay uncertainty β and the upper bound on the derivative of forward delay uncertainty β^f, as well as the nominal time delay h. We refer to [4] for details. An internally robust digital implementation of the controller (10) includes a PI term which is cascaded with a feedback block containing an FIR (Finite Impulse Response) filter $F(hs)$. The length of the FIR filter is h/T_s, where T_s is the sampling period, see [4] for the calculation of FIR coefficients.

In our setting, robust controller K_1 and predictor K_2 are combined together to form a flow controller which has access to queue information as well as capacity information.

4. Short-Term Prediction of Self-Similar Traffic

In the digital implementation of the flow controller of Figure 2, we should provide a $[h/T_s]$-step prediction of the available bandwidth, where T_s is the sampling period of the discretized system. The positive correlations of LRD traffic implies the possibility of nontrivial short-term traffic prediction. Since the derivation of an optimal nonlinear predictor will depend on higher order moment functions that may not be available, we will consider designing causal LMMSE predictors. In fact, It has been demonstrated in [7] that the causal linear predictors have good prediction results for self-similar signals.

First, let's consider one step prediction. Denote the available capacity observed at time k by $c(k)$. We have a random sequence $c(k), k = 0, 1, \ldots, N$ by time N. Suppose we will be using a sliding window of length l, $\{c(N), \ldots, c(N-l+1)\}$, to generate a one-step prediction, namely $\hat{c}_1(N)$, which will be the predicted value of $c(N+1)$. In order to have better estimation, we further assume $N \geq 2l$. Let $\bar{c}_l(N) = \frac{\sum_{k=0}^{l-1} c(N-k)}{l}$ be the mean value and denote $x(k) = c(k) - \bar{c}_l(N), k = N-l+1, \cdots, N$. Then we have

$$\hat{c}_1(N) = \bar{c}_l(N) + \hat{x}_1(N) \tag{11}$$

where $\hat{x}_1(N)$ is the one-step prediction of zero-mean W.S.S. random sequence $x(N+1)$ given $\{x(k),\ k = N-l+1, \cdots, N\}$, which can be written as:

$$\hat{x}_1(N) = \sum_{i=0}^{l-1} a_i^{(N)} x(N-l+1+i) \tag{12}$$

where $a_i^{(N)}, i = 0, \cdots, l-1$ are optimal parameters of the LMMSE predictor, which satisfy the orthogonality condition, [17],

$$\left[x(N+1) - \sum_{i=0}^{l-1} a_i^{(N)} x(N-l+1+i)\right] \perp x(k), \quad N-l+1 \leq k \leq N. \tag{13}$$

We define the column vector $\mathbf{a^{(N)}} := \begin{bmatrix} a_0^{(N)} & a_1^{(N)} & \cdots & a_{l-1}^{(N)} \end{bmatrix}^T$ and the row vector

$$\begin{aligned} \Gamma_1 &:= E\left[x(N+1)x(N-l+1)\ \ x(N+1)x(N-l+2)\ \ \cdots\ \ x(N+1)x(N)\right] \\ &= [K_{xx}[l]\ \ K_{xx}[l-1]\ \ \cdots\ \ K_{xx}[1]] \end{aligned} \tag{14}$$

and $l \times l$ Toeplitz covariance matrix $K_{\mathbf{xx}}$, where

$$(K_{\mathbf{xx}})_{ij} = E[x(N-l+1+i)x(N-l+1+j)] = K_{xx}[|i-j|]. \tag{15}$$

From (13), we have:

$$\mathbf{a}^{(\mathbf{N})^T} = \Gamma_1 K_{\mathbf{xx}}^{-1}. \tag{16}$$

This prediction is non-biased and the LMMSE is given by

$$\begin{aligned}\varepsilon_{min} &= E\left[(x(N+1) - \hat{x}_1(N))^2\right] \\ &= E\left[(x(N))^2\right] - \Gamma_1 K_{\mathbf{xx}}^{-1} \Gamma_1^T.\end{aligned} \tag{17}$$

From (11), (12) and (16), we can determine the algorithm for one-step prediction. Note that the covariance $K_{xx}[i], i = 0, \ldots, l$ can be estimated by the time average under the assumption that the sequence is ergodic in correlation, i.e.

$$\hat{K}_{xx}[i] = \frac{1}{m-i} \sum_{k=1}^{m-i} x[N-m+k]x[N-m+i+k] \tag{18}$$

where we choose $2l \leq m \leq N$ to guarantee accuracy.

Similarly, the multiple-step prediction can be easily formulated by substituting Γ_1 with $\Gamma_s := [K_{xx}[l+s-1] \ K_{xx}[l+s-2] \ \cdots \ K_{xx}[s]]$ in the above derivation, where s is the prediction steps.

5. Simulation Results of the LMMSE Predictor

In the following, the performance of the LMMSE predictor is evaluated for self-similar network traffic prediction by simulations using data traces generated by *ns-2*. We set the network topology to be a single bottleneck node with 50 UDP flows sharing the link. Each flow, at the application layer, obeys a Pareto distribution with shape parameter α. The link capacity is supposed to be 4 Mbps and we sample the aggregated traffic with $T_s = 0.05\ sec$. It has been shown that the heavy-tailed distributions of objects (e.g. files) at the application layer will result in the self-similarity of the aggregated traffic [14]. Thus this simulation can examine whether the LMMSE predictor designed here is qualified for self-similar traffic prediction.

For one step prediction, we set $l = 10, m = 100$ and define the relative error to be $E[\frac{x-\hat{x}}{x}]$. By Choosing $\alpha = 1.20, 1.45, 1.95$, we have the results shown in Figure 2. The error for each case is around 6%, which indicates good match between the traffic data and our predictions.

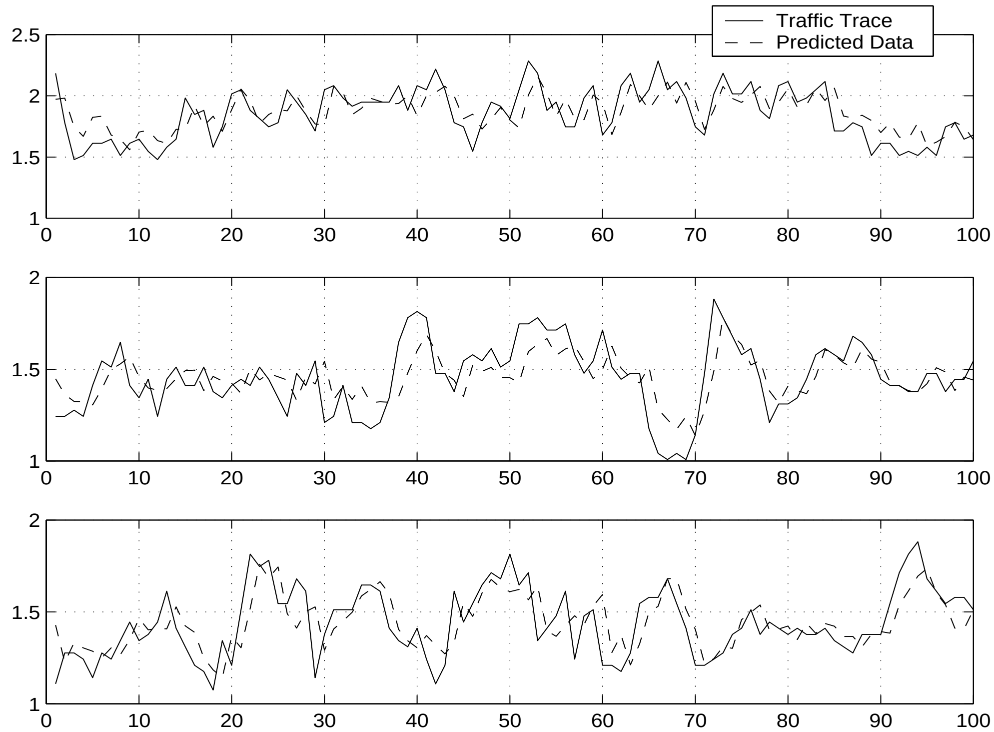

Figure 2. One-step prediction for different values of α

Remark 1 *It has been demonstrated in [11] that it makes little difference whether we know x on $(-\infty, t)$ or $(0, t)$ for short-term prediction of fBm traffic. What is more, our simulations show that larger l and m cannot improve the performance very much, although they increase complexity.*

Intuitively, for multiple-step prediction, the error might be larger with more prediction steps, which can be verified by Figure 3, where we plot the errors for up to 20 prediction steps for each case.

From Figure 3, we can also find that for smaller α, which means heavier distribution in the tail, the errors become smaller for both one-step and multiple-step predictions. In order to validate this claim, we introduce the relative error variance (RV) $Var(x-\hat{x})/Var(x)$ (see [11] for details), which is another index for evaluating the predictor. For $\alpha = 1.05,\ 1.20,\ 1.45,\ 1.80,\ 1.95$ respectively, we plot the errors and the relative error variances as depicted in Figure 4.

Remark 2 *Theoretically, the Hurst Parameter H can be calculated by $H = \frac{3-\alpha}{2}$, which comes from the ON/OFF model in the idealized case corresponding to a fGn process [6, 19]. In reality, H can still be linearly related to α, although its slope changes case by case[14, 18].*

The results in [11] show that the relative error variance decays to zero when H approaches 1, that means the LMMSE based predictor works better for the case of stronger LRD. In other words, our prediction is more reliable for smaller shape parameter α, which has been demonstrated in Figures 3 and 4. Note that estimation errors are not monotone with respect to the shape parameter α because our data traces came from the aggregation of heavy-tailed distributions, not idealized fGn.

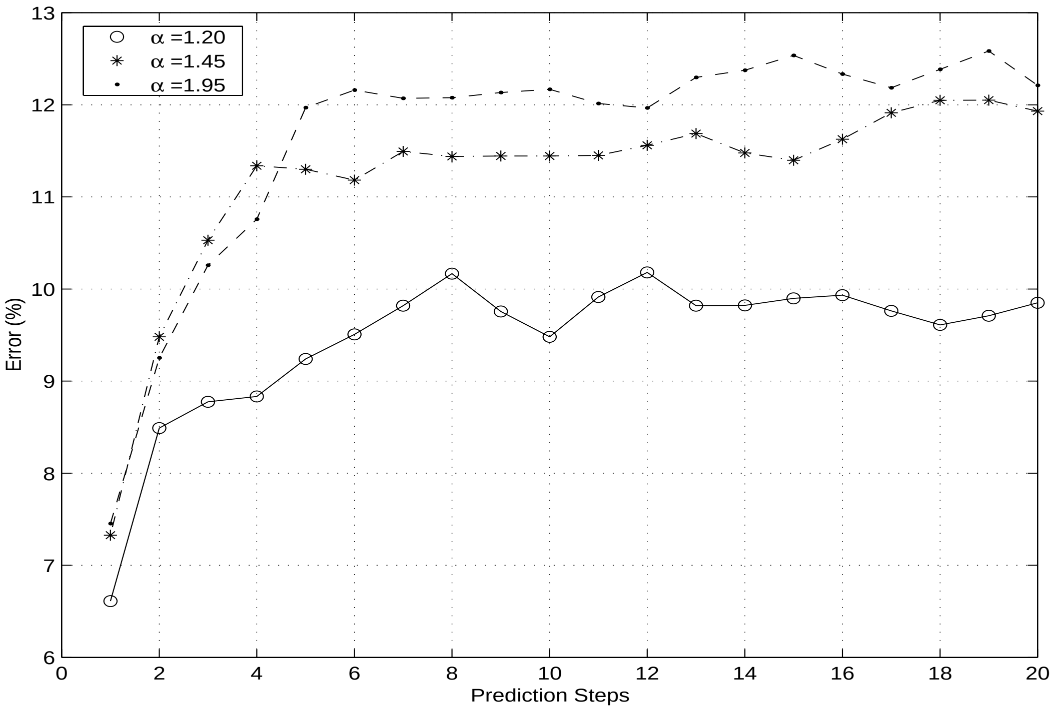

Figure 3. Error for different prediction steps

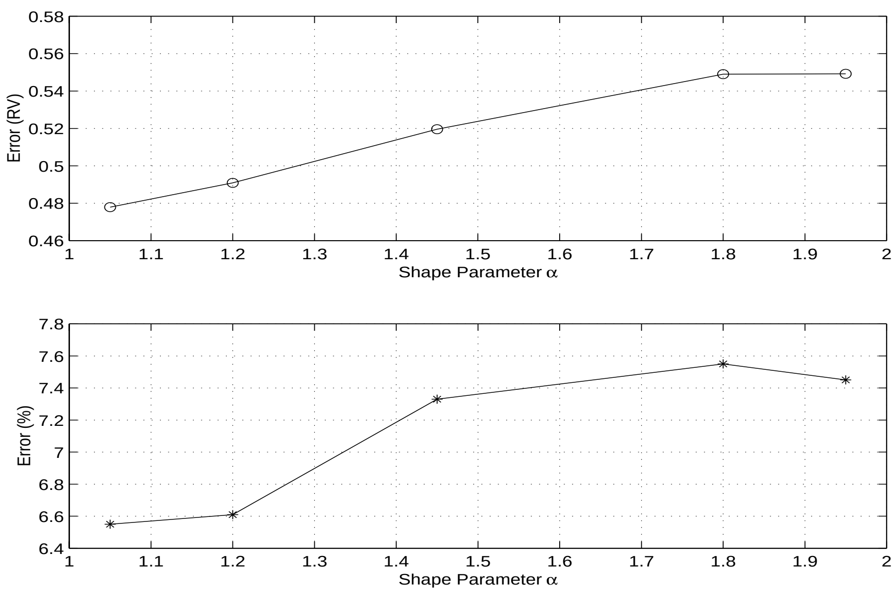

Figure 4. Prediction Error and relative error variance in one-step prediction

Meanwhile, the self-similarity implies nontrivial correlation structure at larger time scales, with which we can realize larger time scale predictions. For the above data traces, we change the time scale to nT_s, which means $c(k)$ is aggregated over n samples and replaced by its mean $c^{(n)}(k)$, as (2) shows. So all prediction at time scale nT_s will be based on this new sequence $c^{(n)}(k)$, $k \in \mathbb{Z}^+ \cup \{0\}$. Following the above simulations, we obtained similar results, which again verified the self-similar nature of network traffic as well as the feasibility of the LMMSE predictor for different time scale predictions. We only list the one step prediction error for the scenario of 1 *sec* time scale (which corresponds to $n = 20$) for different values of α in Table 1. Other results are omitted because they are quite similar to the above figures. Recall from the above discussion that for α between 1.2 and 1.95 the relative error was about 6% for 0.05 *sec* time scale, these results are comparable to the ones shown in Table 1. But for the same time interval of prediction, the larger time scale prediction needs fewer prediction steps compared with that at smaller time scale. We will make use of this fact in the system simulations in Section 6.

	$\alpha = 1.05$	$\alpha = 1.20$	$\alpha = 1.45$	$\alpha = 1.80$	$\alpha = 1.95$
Prediction error	5.08%	5.59%	5.16%	5.29%	6.41%

Table 1. One-step prediction at a larger time scale ($1sec$)

Remark 3 *Note that the LMMSE predictor is an adaptive estimator since the predictor updates its parameters at every sampling time as described by (12), (16) and (18). Moreover this scheme can be easily extended for a multi-sources network topology, since the predictor block K_2 is designed independently of the control block K_1 and that the design of the block K_1 for a multi-source network topology with uncertain time-varying time delays has been considered in [4].*

6. System Level Simulations

The control scheme combining the LMMSE predictor K_2 and the robust control block K_1 is applied to the discretized model depicted in Figure 5, where the sampling time is T_s and the time delay is equal to MT_s with $M = [h/T_s]$; K_2 predicts the traffic at time scale nT_s which results from the traffic aggregation obeying (2). Correspondingly the Zero Order Hold holds the signal for n steps which gives the output r_{LMMSE} of the overall controller sampled at nT_s. Note that $n = 1$ corresponds to the case where every signal is sampled at the value T_s.

Remark 4 *We should notice that the robust controller K_1 with a ZOH at $n = 1$ in Figure 5 is just a discretized implementation of K_1, which is a common practice of controller implementation (from continuous-time to discrete-time). At larger time scales (i.e. $n > 1$ in the ZOH), the robust stability could not be guaranteed theoretically. For this sake, we will investigate the system performance in the presence of the ZOH at different time scales in terms of settling time t_s and cost index J.*

The simulations of our control scheme is compared with those of the two-degree of freedom $\mathcal{H}^\infty$ controller of [15] which we call comparison controllers for short.

We assume $T_s = 0.05$ *sec*, $h = 1$ *sec*, $n = 1$, $q_d(k) = 80$ with $k \in \mathbb{Z}^+ \cup \{0\}$ in our first setting. The cross traffic $c_{\text{cross}}(k)$ is generated by using the similar network topology as mentioned in Section 5 by *ns-2*, where the shape parameter is $\alpha = 1.80$. The LMMSE system responses are depicted in Figure 6.

We can see that the LMMSE control scheme and the comparison controller have similar performance, with nearly the same overshoot, settling time, and oscillations. In fact, by adding time-varying time delay uncertainty, say $\delta(k) = 0.5\sin(\frac{2\pi}{50}kT_s)$, we can still get the same conclusion, which implies that the two kinds of controllers even have similar robustness.

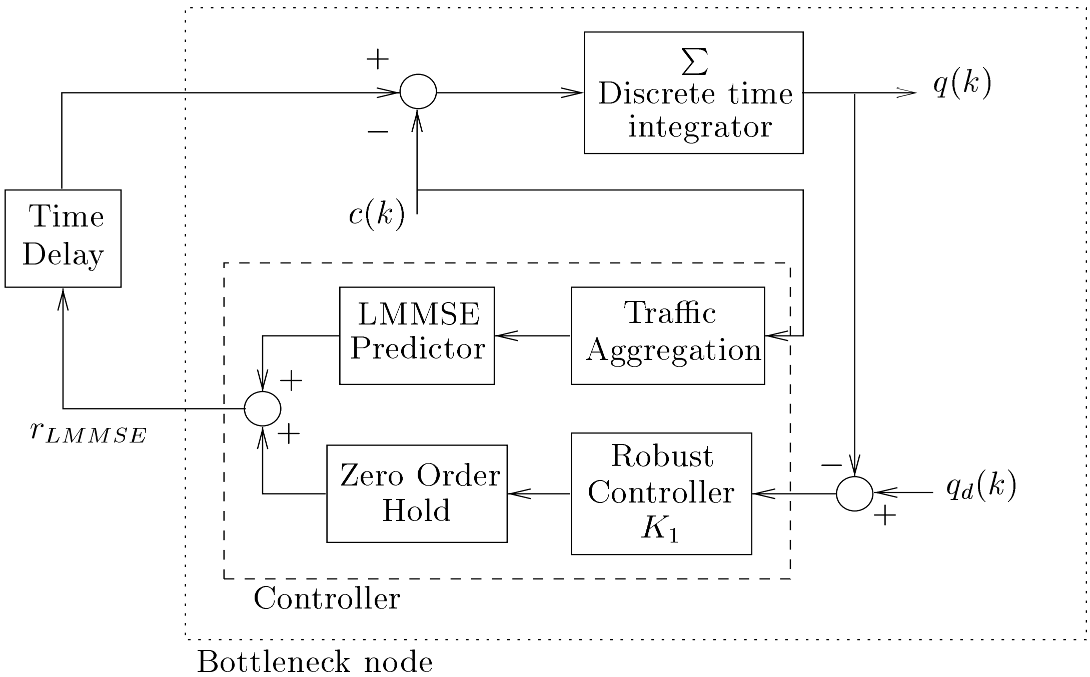

Figure 5. Discretized model

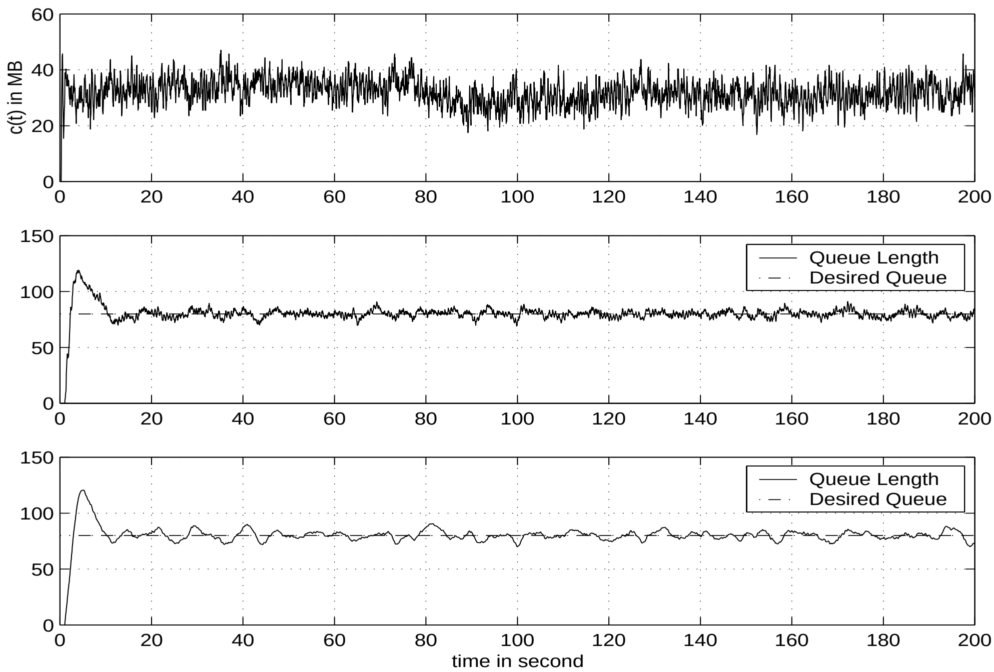

Figure 6. System responses for 0.05 *sec* time scale: the middle one denotes the comparison controller and the bottom one denotes the LMMSE based scheme

However, the LMMSE predictor is designed directly for self-similar network traffic, that works equally well in the larger time scale, with the extra advantage of reducing the prediction steps which has been discussed in Section 5. Thus we would like to implement it with the robust control block at larger time scales, that is more applicable for the real network environment, where control actions are updated once every RTT.

Before such implementation, we need to investigate the performance of the robust control block K_1 at different time scales (i.e. holding the control output for n steps in our simulations). In this case we quantify the performance of the control system in two ways:

- Settling time t_s, defined as $t_s = \min_t\{|q(s) - q_d(s)|/q_d(s) \leq 5\%, \forall s > t\}$,

- Cost index $J = \frac{1}{T}\sqrt{\sum_{i=0}^{[T/T_s]}(q(iT_s) - q_d(iT_s))^2 T_s}$.

For $h = 1$ sec, we choose holding steps n from 1 to 40 and set $c(k) = 60$ for $k \in \mathbb{Z}^+ \cup \{0\}$, by which we omit the impact of the cross traffic. The responses of K_1 alone and the comparison controller are examined with respect to different values of n: performance indexes, t_s and J are shown in Figure 7.

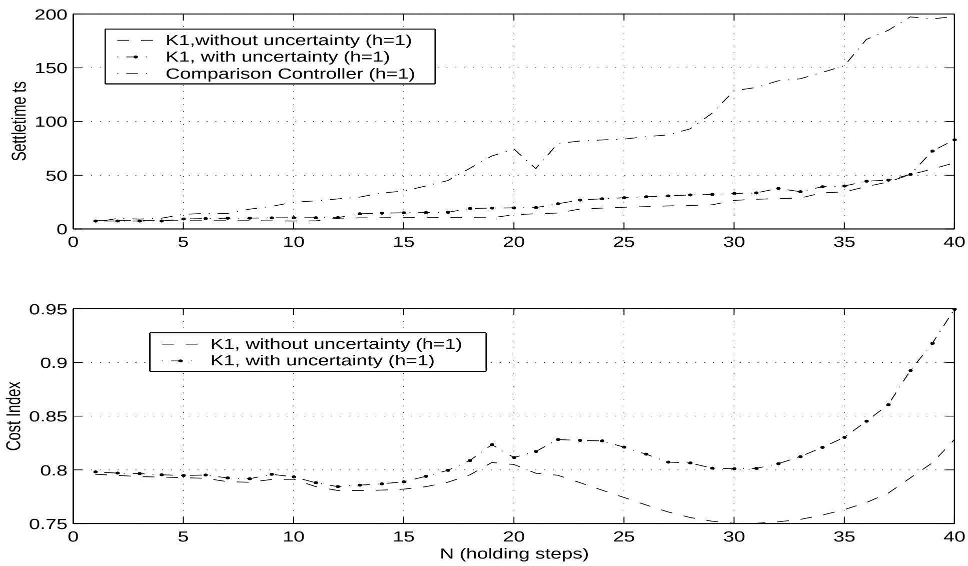

Figure 7. Performance indexes as a function of n

As shown in Figure 7, K_1 preserves good performance even for $n = 20$, i.e. for the case where the assigned rates (controller output) is updated every 1 sec while $T_s = 0.05$ sec. The comparison controller, however, does not possess this property. Larger holding time deteriorates the performance of the comparison controller significantly. Note that we did not plot the cost index curve for the comparison controller in Figure 7, because it diverges very rapidly.

Based on the above simulations at different time scales for K_1 and the multi-scale properties of the LMMSE predictor, we consider applying our control scheme at a larger time scale, where the control output is updated once every second. We adopt the same parameter setting as in the first scenario, except that in this case we have $n = 20$. The responses for the LMMSE controller and the comparison controller are shown in Figure 8, where the performance improvement of the LMMSE control scheme is apparent.

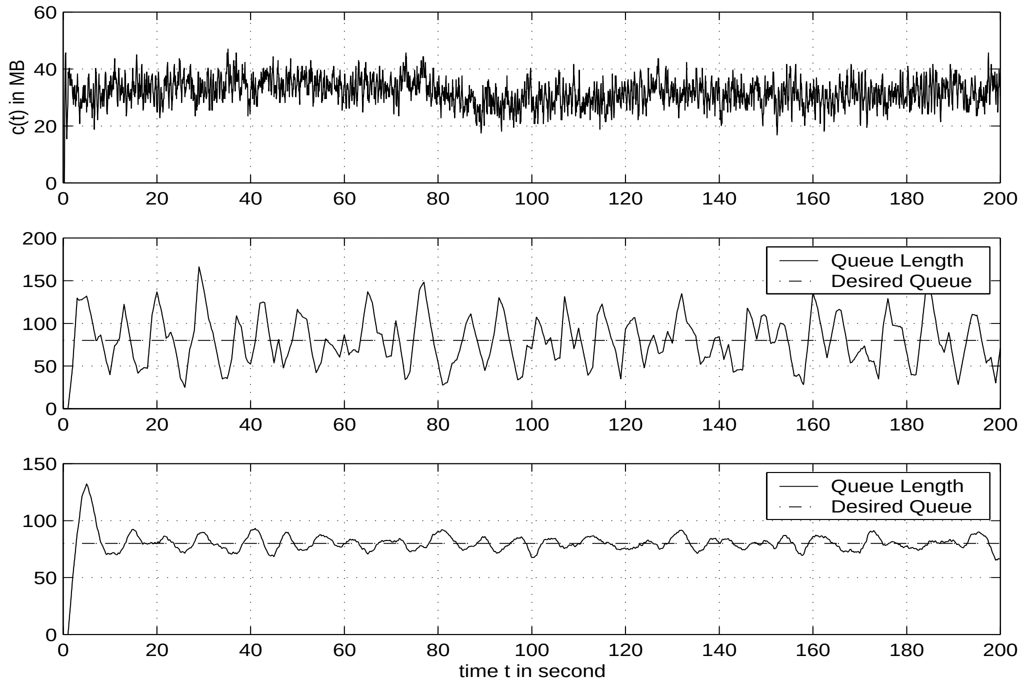

Figure 8. System response for 1 *sec* time scale: The middle one denotes the comparison controller and the bottom one denotes the LMMSE one

Furthermore, we list the performance index J for time scale 0.05 *sec* and 1 *sec*, for both control schemes in Table 2. In conclusion, the controller which uses the LMMSE predictor shows good performance with respect to self-similar traffic and time delay uncertainties, both in smaller time scales and in larger ones.

	time scale 0.05 sec		time scale 1 sec	
	$h=1$	$h=1+\delta(k)$	$h=1$	$h=1+\delta(k)$
LMMSE controller	0.6765	0.6869	0.7398	0.7928
Comparison controller	0.6305	0.6520	1.9751	1.9818

Table 2. Comparison of performance index J

7. Concluding Remarks

In this paper, an LMMSE based predictor is added to the robust $\mathcal{H}^\infty$ controller designed in earlier works [13][4]. The two-degree of freedom flow controller has better transient response for self-similar cross traffic and good robustness against time delay uncertainty. Notably, the flow controller developed here preserves good performance in the scenario of larger time scale prediction and control, which is more applicable for real networks. An interesting extension of this work is to consider prediction and control at different time scales or multiple time scales, by which we could make better use of the multifractal properties of network traffic and the robustness of the $\mathcal{H}^\infty$ controller.

Acknowledgement

The author thanks Professor Hitay Özbay for his guidance and support in this research work.

References

[1] ATM Forum Traffic Management AF-TM-0056.000, "The ATM forum traffic management specification version 4.0," April, 1996.

[2] E. Altman and T. Başar, "Multi-user rate-based flow control: distributed game-theoretic algorithms," *Proceedings of 36th IEEE Conference on Decision and Control*, 1997.

[3] E. Altman, T. Başar, R. Srikant, "Congestion control as a stochastic control problem with action delays," *Automatica*, vol. 35, pp.1937-1950, 1999.

[4] P-F. Quet, B. Ataşlar, A. İftar, H. Özbay, T. Kang, S. Kalyanaraman, "Rate-based flow controllers for communication networks in the presence of uncertain time-varying multiple time-delays," *Automatica*, vol. 38, pp. 917-928, 2002.

[5] F. Bonomi and K. Fendik, "The rate-based flow control framework for the available bit rate ATM service," *IEEE Network*, pp.25-39, March/April, 1995.

[6] D. Cox, "Long-Range-Dependence: A Review" in *Statistics: An Appraisal*, H. T. David and H. A. David (Ed), Iowa State University Press, 1984.

[7] G. Gripenberg and I. Norros, "On the prediction of fractional Brownian motion," *J. Applied Probability*, vol. 33, pp. 400-410, 1995.

[8] V. Jacobson, and M. Karels, "Congestion avoidance and control," *Proc. of ACM SIGCOMM*, 1988.

[9] R. Jain, "Congestion control and traffic management in ATM networks: recent advances and a survey," *Computer Networks and ISDN Systems*, vol. 28, pp.1723-1738, 1996.

[10] S. Mascolo, D. Cavendish, and M. Gerla, "ATM rate based congestion control using a Smith predictor: an EPRCA implementation," *Proc. IEEE INFOCOM*, 1996.

[11] I. Norros, "On the use of Fractional Brownian Motion in the theory of connectionless networks," *IEEE Journal on Selected Areas in Communication*, vol. 13, pp.953-962, August, 1995.

[12] H. Özbay, S. Kalyanaraman, and A. İftar, "On rate-based congestion control in high-speed networks: design of an H^∞ based flow controller for single bottleneck, *Proc. of American Control Conference*, June 1998.

[13] H. Özbay, T. Kang, S. Kalyanaraman, and A. İftar, "Performance and robustness analysis of an H^∞ based flow controller," *Proceedings of 38th IEEE Conference on Decision and Control*, December 1999.

[14] K. Park, G. Kim, and M. Crovella, "On the effect of traffic self-similarity on network performance," *Proc. of SPIE International Conference on Performance and Control of Network Systems*, Nov. 1997.

[15] P-F. Quet, S. Ramakrishnan, H. Özbay, S. Kalyanaraman, "On the H^∞ controller design for congestion control with a capacity predictor," *Proceedings of 40th IEEE Conference on Decision and Control*, December 2001.

[16] V. Ribeiro, R. Riedi, M. Crouse, and R. Baraniuk, "Multiscale queuing analysis of long-range-dependent network traffic," *Proc. of IEEE INFOCOM*, 2000.

[17] H. Stark and J. Woods, *Probability, Random Processes, and Estimation Theory for Engineers*, Prentice Hall, NJ, 1994.

[18] T. Tuan and K. Park, "Multiple time scale congestion control for self-similar network traffic", *Performance Evaluation*, vol. 36, pp. 359-386, 1999.

[19] W. Willinger, V. Paxson, and M. Taqqu, "Self-similarity and heavy tails: structural modeling of network traffic," in *A Practical Guide To Heavy Tails: Statistical Techniques and Applications*, R. Adler, R. Feldman, and M. Taqqu (Ed), Birkhauser, Boston, 1998.