

# A Predictive Flow Control Scheme for Efficient Network Utilization and QoS

Dongyu Qiu and Ness B. Shroff\*

School of Electrical and Computer Engineering

Purdue University

West Lafayette, IN 47907-1285, U.S.A.

E-mail:{dongyu, shroff}@ecn.purdue.edu

## Abstract

In this paper we develop a new predictive flow control scheme and analyze its performance. This scheme controls the non-real-time (controllable) traffic based on predicting the real-time (uncontrollable) traffic. The goal of the work is to operate the network in a low congestion, high throughput regime. We provide a rigorous analysis of the performance of our flow control method and show that the algorithm has attractive and useful properties. From our analysis we obtain an explicit condition that gives us design guidelines on how to choose a predictor. We learn that it is especially important to take the queueing effect into account in developing the predictor. We also provide numerical results comparing different predictors that use varying degrees of information from the network.

**Key Words:** congestion control; flow control; QoS; integrated traffic.

## 1 Introduction

In an efficiently utilized network, even if the dimensioning of network resources has been done correctly and the admission control mechanism is good, the network may go into periods of congestion

---

\*Please address correspondence to Ness B. Shroff (shroff@purdue.edu, Tel. +1-765-494-3471, Fax. +1-765-494-3358). This work has been partially supported by the National Science Foundation through the NSF CAREER award NCR-9624525, and the Indiana 21st Century Research and Technology Award 1220000634.

due to the transient oscillations in the network traffic. It is thus necessary to develop a mechanism to quickly reduce the congestion or pre-empt it, so as to cause the least possible degradation of quality of service (QoS) to the underlying applications. This is especially important for delay sensitive traffic such as video or voice traffic, which we will refer to as *real-time* (RT) traffic in this paper. For these types of traffic, if a packet arrives after a certain deadline it is assumed to be lost. There are also *elastic* types of traffic, such as data traffic (e-mail, file transfers, web browsing, etc.), that are more delay insensitive and we will call them *non-real-time* (NRT) traffic. In fact, in our framework, NRT traffic corresponds to any type of traffic that is controllable, while RT traffic corresponds to any type of traffic that is uncontrollable. Our goal will be to develop a predictive flow control algorithm that facilitates high network throughput and low probability of overflow.

A seemingly appealing (but naive) solution to prevent the NRT traffic from affecting the RT traffic is to simply give the RT traffic priority over the NRT traffic. While this ensures that the RT traffic is not delayed by the NRT traffic in the network, it also means that the NRT may have to be retransmitted many times (due to buffer overflows within the network), to be successfully transmitted, thus, reducing the overall throughput in the network. Hence, the problem of controlling the NRT traffic must be more carefully considered.

There have been different research efforts on how to control the NRT traffic arrival rate, from simple single-bit feedback control [1][2] to explicit-rate control [3][4][5][6][7]. Some early rate-based control schemes were fairly simple and did not consider the impact of high-priority RT traffic, nor did they consider the round trip propagation delay. In these schemes, low-frequency high-magnitude oscillations may exist in the queueing of packets that occurs within the network. By considering the high-priority traffic and round-trip delays, an explicit-rate control scheme based on the result of frequency domain analysis of multimedia traffic [8] was proposed in [7]. The basic idea of [7] is to apply a low-pass filter on the high-priority traffic, then predict the low frequency part of the high-priority traffic, and finally control the NRT traffic rate. Because of the low-pass filter, the high-frequency oscillations in the queueing process are eliminated, thus improving stability. In [7], the control objective was to minimize the unused link capacity by minimizing the mean squared prediction error of the high-priority traffic rate. But the authors in [7], use a simple step function, rather than a stochastic process, to model the smoothed RT traffic, In some cases, this approach could lead to poor performance, as will be illustrated in Section 6 of this paper. Further, modeling the RT traffic with a step function does not shed any light on the queueing behavior (congestion

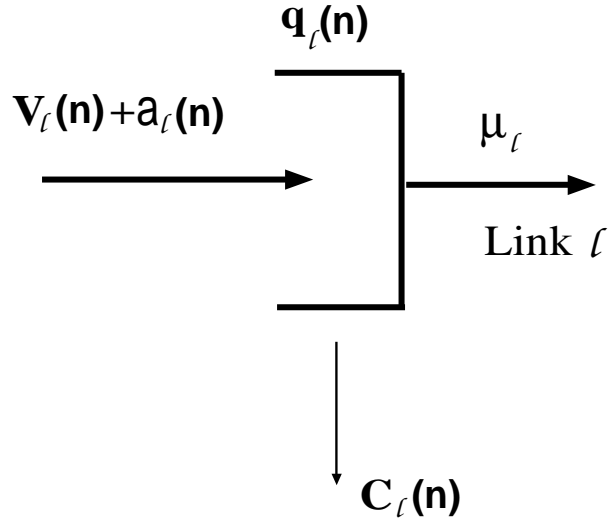


Figure 1: Individual link-level problem

characteristics) at the network node being controlled.

For an explicit-rate control scheme to be practical when high-priority RT traffic (or other uncontrollable traffic) is present, there are two main problems that need to be addressed.

1. *Individual Link-Level Problem:* This is the problem that we will focus on in this paper. Predicting the link capacity available for the NRT traffic at an individual link in the network. One approach to controlling the NRT traffic is to be able to predict the RT traffic (at the link of interest) at some time in the future. Then, based on this prediction, control the NRT traffic. Prediction is required due to the relatively large propagation delay component (large delay bandwidth product) in high-speed networks. Because of the stochastic variations in the aggregate high-priority RT traffic at each link, the available link capacity for the NRT traffic will be time-varying. Our objective is to find a way to predict the available link capacity for the NRT traffic such that the output link is efficiently utilized and the overflow probability is kept low. In Figure 1,  $V_l(n)$  and  $a_l(n)$  corresponds to the aggregate RT traffic and NRT traffic, respectively, arriving at a link of interest (link  $l$  having capacity  $\mu_l$ ), at time  $n$ . One can then estimate  $C_l(n)$ , the available link capacity for NRT traffic, at some time in the future. This information would then be used at the *network-level* to distribute the available link capacities to the NRT flows.

2. *Network-Level Problem:* Distributing the available link capacities for the NRT flows in the

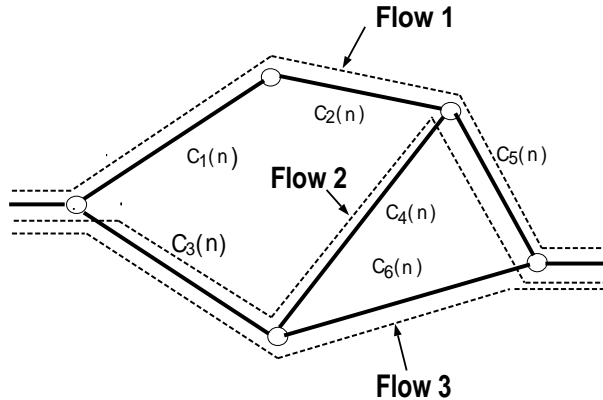


Figure 2: Network-level problem

network to maximize throughput (or more generally some utility function), based on appropriate fairness requirements. Here it is imperative that the resultant algorithms be easily implemented in a distributed fashion. An example network is shown in Figure 2, where flows traverse links with available capacities for the NRT flows calculated at the individual link level. While the network-level problem has been investigated in the case when the available link capacity for NRT flows at each node is a constant (e.g. in the case when no RT flows are present)[9][10][11][12], the problem remains open in the case when the available capacity is time-varying. This problem is not the focus of this paper (our preliminary work on this problem can be found in [13]).

One of the main difficulties of the link-level problem is the non-linearity in the system. The queueing system that determines the congestion on the link is itself a non-linear system. Additionally, the control system to be implemented is also non-linear, because the controlled NRT traffic rate can never be less than zero. Due to this non-linear property, it is difficult to analyze the performance of the system and analyze the stability of the explicit-rate controlled system. Hence, in some works [4][7][14], linear approximation techniques are used in either the queueing system or the control system or both (note that in [14], the queueing system is considered to be non-linear, but the available link capacity for controllable traffic is assumed to be driven by an ARMA model and the control system is assumed to be linear). In contrast, in our work, we will explicitly take into account the non-linearity in both the queueing and the control systems. What we present here is a general prediction and control framework. For example, depending on the information that is

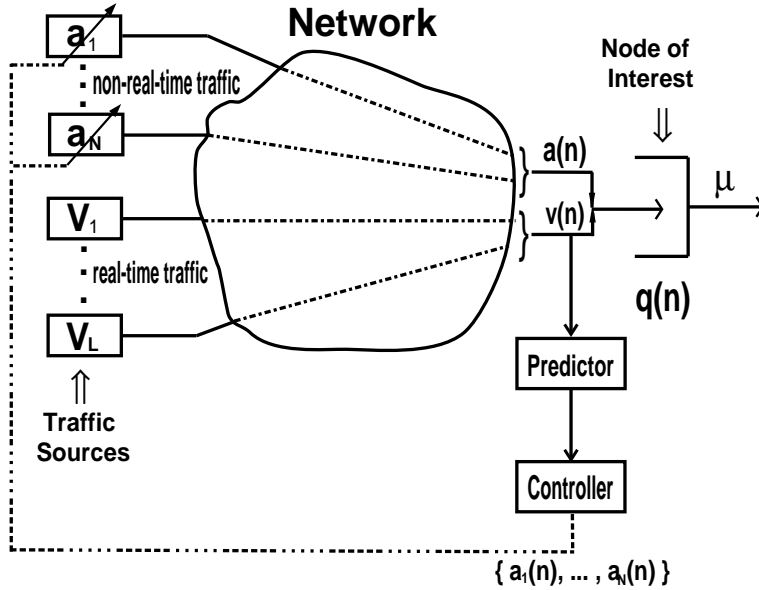


Figure 3: System diagram of predictive flow control

available from the network, our predictive flow control scheme can vary from one that uses a simple low pass filter with linear control algorithm to a more complex scheme that takes advantage of per-flow information, potentially improving the performance. In Section 6, we will describe three different approaches that take advantage of different levels of information available to us from the network and have different levels of complexity and performance.

## 2 Predictive Flow Control Model

As mentioned in the introduction, in this paper we focus on the individual link-level problem. Hence, we focus on a single multiplexing point in the network which consists of a link and an associated buffer that serve both RT and NRT traffic. The multiplexing point in the network could be an output port of a router/switch or a multiplexer. This multiplexing point is modeled by a work-conserving, discrete-time fluid queue with an infinite buffer. Note that we only require that the service discipline be work conserving and not necessarily FIFO. In fact, to prevent the NRT traffic from affecting the RT traffic, it may be better to use some kind of priority queue rather than a simple FIFO queue. For analytical tractability, we assume an infinite buffer system. This allows us to obtain valuable insights on how to design a predictive flow control system. We believe that

these insights will carry over to finite buffer systems, because of the close relationship of various congestion metrics in finite and infinite buffer systems [15, 16, 17]. For example, it is well known that when the queue level of an infinite buffer system is small, it also follows that the queue level of the corresponding finite buffer system is small. So, if our intention was to develop a control algorithm to maximize the goodput in a finite buffer system with buffer size  $x$ , one approach would be to develop a control algorithm to minimize the probability that the corresponding infinite buffer queue would exceed a level  $x$ , given a utilization constraint.

Our system diagram is shown in Fig 3. Let  $V(n)$  be the aggregate amount of RT traffic that arrives at the queue of interest at time  $n$ . We assume that  $V_{max} := \sup_{n \geq 0} \{V(n)\}$  is finite and that  $V(n)$  is stationary in the mean, i.e.,  $\bar{V} := \mathbb{E}\{V(n)\}$ . Note that we do not explicitly specify the units here because our theoretical analysis will hold once all units are consistent.

Our goal is to control the NRT traffic based on predicting the aggregate RT traffic arrival rate at the queue. We define  $a_i(n)$  to be the available link capacity for the  $i$ th NRT traffic computed at time  $n$  based on the predicted value of the RT traffic rate. This explicit rate information is sent back to the  $i$ th NRT traffic source. Since, in this paper, our focus is on the individual link-level problem, we assume that the calculated available link capacity for the NRT traffic can be fully utilized, i.e., the NRT traffic sources always have enough data to transmit. Let  $N$  be the number of NRT traffic sources and let  $n_i, i = 1, \dots, N$  be the round trip delay between the  $i$ th NRT source and the destination (note that we need the notation  $n_i$  for our theoretical analysis but in practice, the actual value of  $n_i$  may or may not be needed. For more discussion see Sections 5 and 6). Then  $a(n) = \sum_{i=1}^N a_i(n - n_i)$  is the aggregate NRT traffic arrival to the queue at time  $n$ . We assume that a control message is propagated from the queue of interest to the destination and back to the source. We further define  $\hat{V}_i(n)$  as the predicted value of  $V(n)$  based on the history of  $V$  before time  $n - n_i$ . We assume that the predictor is linear.<sup>1</sup>

Let  $V(z)$  be the Z-transform of  $V(n) - \bar{V}$  and  $\hat{V}_i(z)$  be the Z-transform of  $\hat{V}_i(n) - \bar{V}$ . Then

$$\hat{V}_i(z) = z^{-n_i} H_i(z) V(z), \quad (1)$$

where  $H_i(z)$ , for NRT traffic source  $i$ , is a causal, stable, linear, time-invariant system [18]. It should be noted that  $H_i(z)$  will be the same for all sources with the same round-trip time  $n_i$ . This

---

<sup>1</sup>Using a linear predictor has been found to give good results for video traffic [8, 7], and we find that it gives good results for other types of traffic as well. Unlike other works in the literature, we allow both the control algorithm and the queueing operation to be non-linear.

form of predictor is quite general and includes all time-series types of predictors (e.g. ARMA). For example, if there is only one NRT flow with round trip delay 5, a possible predictor (in time domain) could be  $\hat{V}_1(n+5) = \frac{1}{2}V(n) + \frac{1}{2}V(n-1)$ . In this case, we will have  $H_1(z) = \frac{1}{2}(1+z^{-1})$ . From Eq. (1), we can easily show that  $\mathbb{E}\{\hat{V}_i(n)\} = \bar{V}$ , i.e., the predictor is unbiased.

The workload or queue-length,  $q(n)$ , at time  $n$  at the queue of interest will be determined by  $a(n)$ ,  $V(n)$ , and the service rate (link rate) of the queue  $\mu$ . We assume that the queue process begins at time  $n=0$  and  $q(0)=0$ . For stability, we also require that  $\bar{V} < \mu$ .

The feedback control scheme is as follows. We predict the aggregate RT traffic rate and use the predicted value to compute  $a_i(n)$ ,  $1 \leq n \leq N$ . Here, we ignore the time needed for prediction and computing (however, we can just as easily add it to the delay  $n_i$ ). In this paper, we will present a predictive flow control algorithm and analyze the queueing behavior of the system. For simplicity of exposition, we will first consider only one NRT traffic source and develop a result for the queueing behavior at the queue of interest. We extend this result to the multiple NRT traffic sources case in Section 4.

### 3 Single NRT Traffic Model

#### 3.1 Predictive Flow Control Algorithm

In this section, we assume that there is only one NRT traffic source  $a_1(n)$  (or a group of NRT traffic loops with the same round trip delay  $n_1$ ). Note that by definition,  $a(n) = a_1(n-n_1)$ . Ideally, what we want to achieve is  $a(n) + V(n) = \mu$  at all time  $n$ . However, there are two difficulties in achieving this. First, since we do not know  $V(n)$  in advance, we need to estimate its value through prediction, resulting in a certain possibility of error. Second,  $V(n)$  could be greater than  $\mu$  but since  $a(n)$  cannot be negative, the sum  $a(n) + V(n)$  cannot be made equal to  $\mu$ .

Taking into account the possibility of prediction error and the possibility that  $V(n) > \mu$ , it seems plausible to control the NRT traffic  $a_1(n)$  such that

$$a_1(n) = [p\mu - \hat{V}_1(n+n_1)]^+, \quad (2)$$

where  $p$  is the percentage of output link capacity that we would like to utilize ( $p > \frac{\bar{V}}{\mu}$ ) and  $[x]^+ = x$ , if  $x \geq 0$ , and  $[x]^+ = 0$ , otherwise. Note that in [7], the non-linearity was not considered and a simplified version of Eq. (2), i.e.,  $a_1(n) = p\mu - \hat{V}_1(n+n_1)$ , was used. Although this method of

control is easy to implement, it has a major drawback. The drawback is that this type of control does not take into account the queueing behavior, but simply the difference between the service rate and the RT traffic. To elaborate, consider even the situation of perfect prediction, and let  $V(n)$  have exceeded  $\mu$  for some time. During the period that  $V(\cdot)$  has exceeded  $\mu$ , the above equation correctly sets  $a_1(\cdot)$  to be zero. But, even after  $V(\cdot)$  is no longer larger than  $\mu$ , there could still be a substantial backlog in the queue, during which time  $a_1(\cdot)$  should be set to zero. However, according to Eq. (2), the moment  $V(n)$  is less than  $\mu$ , the NRT source is allowed to transmit, thus potentially causing unnecessary congestion at the queue. This will also be shown in our simulation results. Hence, we need a better flow control algorithm. What we will attempt to do is to keep the queue length at the node of interest small, while maintaining a certain level of throughput given by

$$\lim_{n \rightarrow \infty} \frac{\sum_{j=1}^n (a(j) + V(j))}{n} = p\mu \quad (3)$$

Note again, that at time 0 the queue will be empty.

### Proposed Control Algorithm ( $N = 1$ case)

1. Define a virtual queueing process  $q_1(n)$  and set  $q_1(0) = 0$ .
2.  $q_1(n) = [q_1(n-1) + \hat{V}_1(n) - p\mu]^+$ . For  $n \leq 0$ , we let  $V(n) = 0$ .
3.  $a_1(n) = [p\mu - \hat{V}_1(n+n_1) - q_1(n+n_1-1)]^+$ . For  $n \leq 0$ , we let  $a_1(n) = 0$ .

In the above algorithm, we maintain a virtual queueing system  $q_1$  with associated workload (or queue-length)  $q_1(n)$  ( $n = 0, 1, 2, \dots$ ), the predicted RT traffic  $\hat{V}_1(n)$  as input, and  $p\mu$  as the service rate. Here we can see the difference between our algorithm and the one given by Eq. (2). In Eq. (2),  $a(n) = [p\mu - \hat{V}_1(n)]^+$ . The NRT traffic rate depends only on the predicted RT traffic rate at time  $n$ . But in our algorithm,  $a(n) = [p\mu - \hat{V}_1(n) - q_1(n-1)]^+$ . When computing the NRT traffic rate, we consider both the predicted RT traffic rate at time  $n$  and the predicted queue length at time  $n-1$ . This results in much lower network congestion, as will be illustrated in the paper. Note that, although we have used  $n_1$  in Step 3 of the algorithm, this does not mean that we need to explicitly know the actual value of  $n_1$ , or even an approximation of it. All we need to know is the value of  $\hat{V}_1(n+n_1)$  (note that  $q_1(n+n_1-1)$  can be calculated from  $\hat{V}_1$ ). To calculate  $\hat{V}_1(n+n_1)$ ,  $n_1$  may or may not be needed, depending on the type of predictor being used. For example, if we choose a simple predictor  $H_1(z) = 0.5 + 0.5z^{-1}$ .  $\hat{V}_1(n+n_1) = 0.5V(n) + 0.5V(n-1)$ . Here, we can see



that the value of  $n_1$  is not needed to calculate  $\hat{V}_1(n + n_1)$ . In Section 6, we will provide numerical results comparing different predictors (some that explicitly use values of  $n_1$  and some that do not).

Also note that when  $\hat{V}_1(n) \leq p\mu$  for all  $n$ ,  $q_1(n)$  will always be zero. In this case our control algorithm will reduce to the linear version of Eq. (2), i.e.,  $a_1(n) = p\mu - \hat{V}_1(n + n_1)$ . But since the high priority RT traffic does not always satisfy this condition, although the linear version is simple, it is not appropriate, especially for very bursty RT traffic (we will discuss this further in Section 6).

We now focus exclusively on our control algorithm. We consider two queueing systems with  $\hat{V}_1(n)$  as the RT traffic input. Note that  $q_1(n)$  and  $q_2(n)$  described below are not the queue lengths of an actual queue at a link, but merely useful descriptors for developing our theoretical analysis.

**Lemma 1** *Let  $q_1(n)$  be the workload at time  $n$  in a queueing system with input  $\hat{V}_1(n)$  and service rate  $p\mu$  and let  $q_2(n)$  be the workload at time  $n$  in a queueing system with input  $a(n) + \hat{V}_1(n)$  and service rate  $p\mu$ . Then  $q_2(n) = q_1(n)$  for any  $n \geq 0$ .*

*Proof:* See Appendix A ■

From Lemma 1, we observe that the queue length (or workload) of a system that uses our control mechanism having both  $a(n)$  and  $\hat{V}_1(n)$  as input is the same as the queue length of a system which only has  $\hat{V}_1(n)$  as input. This is a desirable property. It means that we can send more data and at the same time not increase the queue length. Of course, in a real network, the RT traffic rate will be  $V(n)$  rather than  $\hat{V}_1(n)$ , hence we will take this into account as well. Next, we will focus on the actual queue that we wish to control. This queue will have input from both the aggregate RT source  $V(n)$  and the controlled NRT source  $a(n)$ . As mentioned before the service rate of this queue is  $\mu$ . We will show that using our control algorithm, the output link utilization of the queue of interest is fixed at  $p$  (this is what we want from (3)).

**Proposition 1** *For our predictive flow control algorithm defined in steps 1–3 earlier, under the condition  $\bar{V} < p\mu$ , we have*

$$\lim_{n \rightarrow \infty} \frac{\sum_{j=1}^n a(j)}{n} = p\mu - \bar{V}.$$

*Proof:* See Appendix B ■

Since the input to the queue in consideration is  $a(n) + V(n)$ , and since  $\mathbb{E}\{V(n)\} = \bar{V}$ , what Proposition 1 tells us is that the average aggregate input rate to the queue is given by  $p\mu$ . As mentioned earlier, this is exactly the desired utilization we want. This also tells us that when we

fix  $p$  in our control algorithm, the output link utilization is also fixed and does not depend on other predictor parameters.

Next, we will fix  $p$ , and observe how the queue length of the queue of interest will be affected by different predictor parameters.

### 3.2 Properties of the Flow Controlled Queue

The queue in consideration has service rate  $\mu$  (output link rate) and the total input to it is  $a(n) + V(n)$  at time  $n > 0$ , where  $V(n)$  is the aggregate RT input and  $a(n)$  is the resultant NRT input, as determined through our predictive flow control algorithm. As mentioned before,  $q(n)$  denotes the workload or queue length of this queue.

Since the queue is empty at time  $n = 0$ ,  $q(n)$ , the queue length at time  $n$ , can be expressed as [19] [20]:

$$q(n) = \sup_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^n (a(j) + V(j) - \mu) \right\}. \quad (4)$$

In this section, we will analyze the relationship between  $q(n)$  and the queue length  $q_0(n)$  of a queueing system with only RT traffic input  $V(n)$ . The next theorem shows that our predictive flow control algorithm can achieve a queue length  $q(n)$  that is only an additive constant ( $2C_1$ ) larger than  $q_0(n)$ , for any  $n$ . Here,  $q_0(n)$  is defined as the queue length at time  $n$  of a queueing system  $q_0$  with arrival rate  $V(n)$  and service rate  $\mu$ . Note that  $q_0$  has the same RT arrival rate  $V(n)$  as in our controlled system, but has no NRT input!

**Theorem 1** *Under the definitions and predictive flow control algorithm defined above, if  $V_{max} < \infty$  and  $\frac{p\mu - \bar{V}}{\mu - \bar{V}} \leq H_1(1) \leq 1$ , we have  $q(n) - q_0(n) \leq 2C_1$ , where  $C_1$  is a constant that does not depend on  $n$ .*

*Proof:* See Appendix C.1 ■

What Proposition 1 tells us is that we can completely achieve the utilization constraint that we set to achieve in Eq. (3). Now what Theorem 1 tells us is that under the condition  $\frac{p\mu - \bar{V}}{\mu - \bar{V}} \leq H_1(1) \leq 1$ , the queue length at any time in our controlled queue will at most exceed the queue length in a queueing system  $q_0$  by a constant. This is quite remarkable since  $q_0$  is a queueing system that has no NRT traffic input at all and hence one would expect that its queue length would be a very loose lower bound on the queue length generated by a control scheme that attempts to achieve

a utilization of  $p\mu > \bar{V}$ . However, even when the  $q(n)$  of our scheme is compared to  $q_0(n)$  the queue length without NRT traffic, we can show that  $q(n)$  is only an additive constant (hence not dependent on the time  $n$ ) larger than  $q_0(n)$ .

We are now ready to extend this result to the case when multiple NRT traffic sources with different round trip delays are multiplexed at the node of interest.

## 4 Multiple NRT Traffic Model

We now consider a system with  $N$  NRT traffic sources. Remember that  $\hat{V}_i(n)$  is the predicted value of the aggregate RT traffic  $V(n)$  based on the history of  $V$  before time  $n - n_i$ . Again, note that  $q(0) = 0$ , our predictive flow control algorithm is given as follows:

### Proposed Control Algorithm

1. Set  $q_i(0) = 0, 1 \leq i \leq N$ .
2.  $q_i(n) = [q_i(n-1) + \hat{V}_i(n) - p\mu]^+$ . For  $n \leq 0$ , we let  $V(n) = 0$ .
3.  $a_i(n) = \frac{1}{N}[p\mu - \hat{V}_i(n+n_i) - q_i(n+n_i-1)]^+$ . For  $n \leq 0$ , we let  $a_i(n) = 0$ .

From the proof of Proposition 1 and the above algorithm, it follows that  $\lim_{n \rightarrow \infty} \frac{\sum_{j=1}^n a_i(j)}{n} = \frac{1}{N}(p\mu - \bar{V})$  for the multiple NRT traffic case. Note that how to assign the available link capacity is in fact a network-level problem. Since we only consider a single link here, for fairness, in step 3, we assign the available link capacity in such a way that each NRT traffic has the same average arrival rate. This is by no means a necessary condition, and can be relaxed by modifying step 3 above in favor of a more unfair system if the situation or the solution of the network-level problem requires it. Also note that in our algorithm, we treat each NRT source separately. We do this for two reasons. First, if we consider other NRT sources, we have to keep the information of other flows. Hence, per-flow information will be needed. On the other hand, if we treat each source separately, per-flow information may not be needed (see Section 6). Second, considering other NRT sources will not significantly improve the performance. We give a simple example to elaborate this point. Consider a bottleneck link with two NRT flows, flow 1 and flow 2, whose RTTs (round trip times) are 1 and 2, respectively. For simplicity, we assume  $\hat{V}_i(n) < p\mu$  for all  $n$ . Now, for flow 2 (with RTT 2), because it has the longer delay, when calculating  $a_2(n)$ , no information about the input

rate of flow 1 (with RTT 1) at the future time  $n + 2$  is available. So,

$$a_2(n) = \frac{1}{2}(p\mu - \hat{V}_2(n + 2)).$$

For flow 1, when calculating  $a_1(n)$ , there is information about the input rate of flow 2 at time  $n + 1$  available. If we want to use this information, we will have,

$$\begin{aligned} a_1(n) &= p\mu - \hat{V}_1(n + 1) - a_2(n - 1) \\ &= p\mu - \hat{V}_1(n + 1) - \frac{1}{2}(p\mu - \hat{V}_2(n + 1)) \\ &= \frac{1}{2}(p\mu - 2\hat{V}_1(n + 1) + \hat{V}_2(n + 1)). \end{aligned}$$

Now, if we let  $\tilde{V}_1(n) = 2\hat{V}_1(n) - \hat{V}_2(n)$  and use  $\tilde{V}_1$  as the predictor when we treat each source separately, we will get the same  $a_1(n)$ . Hence, intuitively it appears that considering other NRT sources will not significantly improve the performance.

We now have a version of Theorem 1 with multiple NRT sources having different round-trip delays. Once again let  $q_0(n)$  correspond to the queue length at time  $n$  of a queueing system with service rate  $\mu$  and input  $V(n)$ , while  $q(n)$  is the queue length at time  $n$  of the system with input  $V(n) + a(n)$ , where  $a(n) = \sum_{i=1}^N a_i(n - n_i)$ .

**Theorem 2** *With the definitions and predictive flow control algorithm defined above, If  $V_{max} < \infty$  and  $\frac{p\mu - \bar{V}}{\mu - \bar{V}} \leq H_i(1) \leq 1$  for all  $1 \leq i \leq N$ , there exists a constant  $C$  such that  $q(n) - q_0(n) \leq 2C$  for any  $n \geq 0$ .*

*Proof:* See Appendix C.2 ■

As before,  $q_0(n)$  is the queue length without the NRT traffic. Therefore, Theorem 2 tells us that with our predictive flow control algorithm, the queue length of the system will be less than the queue length with only RT traffic plus a constant finite value that does not depend on  $n$ .

## 5 Discussion

Consider the condition  $\frac{p\mu - \bar{V}}{\mu - \bar{V}} \leq H_i(1) \leq 1$  that is obtained from our main results. We can see that when  $H_i(1) = 1$ , this condition is satisfied for any  $p \leq 1$ . Hence, from now on the first requirement for our predictor will be that  $H_i(1) = 1$ . We now provide further justification why designing a predictor under this condition is desirable.

## 5.1 Tail of Queue Length Distribution (Tail Probability)

From Theorem 2, we know that the only requirement for the RT traffic is  $V_{max} < \infty$ . So, the result can be applied to both short range dependent and long range dependent RT traffic.

Now, consider the probability  $\mathbb{P}\{q(n) > x\}$ , when  $x$  is large. Since  $q(n) \leq q_0(n) + 2C$ , we have,

$$\mathbb{P}\{q(n) > x\} \leq \mathbb{P}\{q_0(n) > x - 2C\}. \quad (5)$$

Let  $\mathbb{P}\{Q_0 > x\} = \lim_{n \rightarrow \infty} \mathbb{P}\{q_0(n) > x\}$  be the steady state tail of the queue length distribution (also usually called the *tail probability*) of  $q_0$  and let  $\mathbb{P}\{Q > x\} = \lim_{n \rightarrow \infty} \mathbb{P}\{q(n) > x\}$  be the tail probability of  $q$ , our controlled system with queue length  $q(n)$ . When the aggregate RT traffic  $V(n)$  is short range dependent, for a large class of short range dependent traffic [19], we have  $\mathbb{P}\{Q_0 > x\} \sim Ae^{-Bx}$  when  $x$  is large, where  $A, B$  are constants. So, we have,

$$\lim_{x \rightarrow \infty} \frac{\mathbb{P}\{Q > x\}}{\mathbb{P}\{Q_0 > x\}} \leq \lim_{x \rightarrow \infty} \frac{Ae^{-B(x-2C)}}{Ae^{-Bx}} = e^{2BC}. \quad (6)$$

When the aggregate RT traffic  $V(n)$  is long range dependent, for a large class of traffic models [21], the tail probability  $\mathbb{P}\{Q_0 > x\} \approx Ae^{-Bx^d}$ , for  $x$  large and where  $A, B, d$  are constants. Further,  $0 < d < 1$ , hence,

$$\lim_{x \rightarrow \infty} \frac{\mathbb{P}\{Q > x\}}{\mathbb{P}\{Q_0 > x\}} \leq \lim_{x \rightarrow \infty} \frac{Ae^{-B(x-2C)^d}}{Ae^{-Bx^d}} = 1. \quad (7)$$

It should be noted here that the limit of Eq. (7) also holds if the RT traffic is heavy-tailed [22]. Since the right hand side of Eqs. (6) and (7) is a constant (1 for Eq. (7)), this tells us that the tail probability for the flow controlled queue does not diverge from the queue length distribution of the system with only RT traffic. Since the tail probability in infinite buffer systems is intimately connected to the overflow probability in finite buffer systems, this again makes a rigorous statement on the effectiveness of our flow control scheme.

We have show that when  $\frac{\nu\mu - \bar{V}}{\mu - \bar{V}} \leq H_i(1) \leq 1$  for all  $1 \leq i \leq N$ , the queue length of the predictive flow control system will not exceed the original queue length with only RT traffic plus a constant value. Now, the question is what happens when  $H_i(1)$  does not satisfy this condition? It turns out that one can easily construct many cases when, if this condition is not satisfied,  $\lim_{x \rightarrow \infty} \frac{\mathbb{P}\{Q > x\}}{\mathbb{P}\{Q_0 > x\}}$  will go to  $\infty$ . Due to space limitations, we will not explicitly provide such examples here (for such examples, see [23]), however, in our numerical section, this will be illustrated through the help of Figure 7.

## 5.2 Stability

Another important issue in explicit rate feedback flow control is the stability of the controlled queueing system. From our main result, it follows that if  $H_i(z)$  are stable and  $H_i(1) = 1$  for all  $i$ , then the queue length will not exceed that of a queue with only RT traffic plus a constant value. This means that if the queue with only RT traffic is stable, our queueing system will also be stable. Further, this stability does not depend on the predictor parameters, once  $H_i(1) = 1$  is imposed. In a real system, the round trip delays of NRT traffic are not always easily determined and in most feedback flow control schemes, this is an important factor that may cause the system to become unstable. However, in our system, if we keep  $H_i(1) = 1$ , stability will be maintained regardless of the errors in estimating the round-trip delay.

## 5.3 Per-Flow Information

As discussed in Section 3, in our predictive flow control scheme we may or may not choose to use the round-trip delay information in computing the parameters of the predictor. This is important because the round trip delay is the only per-flow information in our predictive flow control scheme. Our main result,  $H_i(1) = 1$ , tells us that a class of predictors satisfying this condition, will perform well. However, which predictor to choose from this class to further improve performance is still an open problem. It depends very much on how much information can be inferred from the network. For this reason, in the numerical examples (Section 6), we will numerically study predictors that require per-flow information and those that do not.

## 5.4 Minimizing which error?

An important idea that motivates the development of our predictive control method is that minimizing the error at each point in the prediction is less important than minimizing the accumulated error. For example, the *minimum mean squared error* (MMSE) predictor is a widely used predictor that can be used to predict the RT traffic. In this way, we have a minimum mean squared error at each prediction point. However, we know that a large queue length is mainly caused by the accumulated traffic in a time interval. Hence, a minimum error at each point in time may be less important than a minimum accumulated error over the time period in which the queue builds up. This is where the condition  $H_i(1) = 1$  is important. Because it tells us that if, in our predictor, we choose  $H_i(1) = 1$ , we are ensuring that the accumulated error is bounded (e.g. Theorem 2).

This is also demonstrated by our numerical results which show that the MMSE predictor results in poor performance compared to predictors that maintain  $H_i(1) = 1$ . In fact, what we will show is that although the MMSE predictor requires per-flow information, it does not perform as well as predictors that explicitly consider the queueing behavior (i.e., using  $H_i(1) = 1$ ) but do not require per-flow information.

### 5.5 More Explanation of $H_i(1) = 1$

In this subsection, we further illustrate the importance of  $H_i(1) = 1$ . For simplicity, we consider a system with only one NRT traffic source, i.e.,  $N = 1$  (hence we focus on  $H_1(1)$ ). First, we define  $\Delta(j) = V(j) - \hat{V}_1(j)$  as the prediction error. When we use the control algorithm proposed in Section 3.1, the queue length at the node of interest (see proof in Appendix D.1)

$$q(n) \leq q_0(n) + \sup_{0 \leq n_0 \leq n} \sum_{j=n_0+1}^n \Delta(j) - \inf_{0 \leq n_0 \leq n} \sum_{j=n_0+1}^n \Delta(j). \quad (8)$$

This inequality (Eq. (8)) gives an upper bound on  $q(n)$ . Note that the only requirement of Eq. (8) is the proposed control algorithm in Section 3.1. It has no requirement on the predictor  $H_1(z)$  and the high priority traffic rate  $V(n)$ . Hence, Eq. (8) is true even if  $V_{max}$  is infinite. Next, we assume  $V(n)$  is wide sense stationary, but we do not require a finite  $V_{max}$ .

Now, let  $l = n - n_0$  and define

$$X_{n,l} \triangleq \sum_{j=n-l+1}^n \Delta(j) = \sum_{j=n_0+1}^n \Delta(j).$$

We can see that  $\mathbb{E}\{X_{n,l}\} = 0$  (because the predictor is unbiased) and

$$\begin{aligned} \text{Var}\{X_{n,l}\} &= \text{Var}\left\{ \sum_{j=n-l+1}^n \Delta(j) \right\} = \sum_{j_1=n-l+1}^n \sum_{j_2=n-l+1}^n C_{\Delta}(j_1 - j_2) \\ &= \sum_{j_1=-l+1}^0 \sum_{j_2=-l+1}^0 C_{\Delta}(j_1 - j_2), \end{aligned} \quad (9)$$

where  $C_{\Delta}(j)$  is the auto-covariance function of  $\Delta(n)$ . Because of the stationarity of  $V(n)$ ,  $\text{Var}\{X_{n,l}\}$  does not depend on  $n$ . From Eqs. (8) and (9), we can see that  $\text{Var}\{X_{n,l}\}$  is closely related to the congestion in the system. In fact, from a practical point of view,  $\text{Var}\{X_{n,l}\}$  is more important than  $C_1$ . The constant  $C_1$  only bounds the workload caused by NRT traffic. This bound, while important in showing that under the condition  $H_1(1) = 1$  the workload does not keep growing in time, may not itself be tight. On the other hand,  $\text{Var}\{X_{n,l}\}$  determines the distribution of the

workload as we see in Section 6 (also see [13]). Further, as discussed in Section 5.4, the accumulated error is much more important than the error at each point. One way of making the overall error small is to minimize the asymptotic variance of the error, i.e., minimize  $\lim_{l \rightarrow \infty} \text{Var}\{X_{n,l}\}$ . In fact, as can be seen from the following proposition, when  $H_1(1) = 1$ , the asymptotic variance of the error can also be shown to converge.

**Proposition 2** *When  $H_1(1) = 1$ ,  $\lim_{l \rightarrow \infty} \text{Var}\{X_{n,l}\}$  exists and goes to a finite constant.*

*Proof:* See Appendix D.2 ■

In Section 6, we will discuss more about  $\lim_{l \rightarrow \infty} \text{Var}\{X_{n,l}\}$ , especially its relation to the utilization, the predictor, and the stochastic properties of RT traffic. Note that when  $H_1(1) \neq 1$ ,  $\lim_{l \rightarrow \infty} \text{Var}\{X_{n,l}\}$  may go to infinity. To see this, note that from Eq. (17) in Appendix D.2,  $z = 1$  will in most cases (not so when  $H_1(1) = 1$ ) be a second order pole point of  $S(z)$ . Hence, when  $l \rightarrow \infty$ ,  $S(l) = \text{Var}\{X_{n,l}\}$  will go to infinity. An illustration of the above proposition is also provided in Fig 8, in the numerical results section.

## 6 Numerical Results

In this section, we assume that  $V(n)$  is wide sense stationary. We also assume that  $H_i(z)$  is of the form  $\sum_{m=0}^{M_i} B_m^{(i)} z^{-m}$ , where  $M_i$  is the order of the predictor and  $B_m^{(i)}$  are the predictor parameters that we need to design. From the above discussion, we know that if we design predictors that do not satisfy the condition  $\frac{p\mu - \bar{V}}{\mu - \bar{V}} \leq H_i(1) \leq 1$ , the tail probability of the queue length may be much larger than that of predictors which satisfy the condition. Most current works on predictive flow control and even more general feedback flow control focus on minimizing the mean square of some control objective, for example, the mean square of unused link capacity. In [7], a low pass filter is also applied to the high priority traffic to improve the performance and stability. Under this type of method, the predictor may not satisfy the condition  $\frac{p\mu - \bar{V}}{\mu - \bar{V}} \leq H_i(1) \leq 1$  and in some cases could result in increased congestion. We will demonstrate this through our simulation results. As mentioned in Section 5, since the above condition is satisfied for all  $p$  when  $H_i(1) = 1$ , the first requirement for our predictor is that  $H_i(1) = 1$ . Under the condition that  $H_i(1) = 1$ , there are still a number of predictors that we can choose. How to choose an appropriate predictor depends on how much information we can infer from the network. We will discuss three approaches to designing the predictors under the condition of  $H_i(1) = 1$ .



*Predictor A1:* The first predictor we use is very simple. We simply choose a fixed low pass filter  $H_{LPF}(z)$  as the predictor. Since this predictor is fixed,  $H_i(z) = H_{LPF}(z)$ , for all  $i$ . This is in fact the predictor used in the  $H_2$  scheme of [7]. Note that although the  $H_2$  scheme uses a low pass filter as the predictor, it does not use our control algorithm. In [7], it is assumed that  $\hat{V}_1(n) \leq p\mu$  at all time  $n$  and a linear version of Eq. (2),  $a_1(n) = p\mu - \hat{V}_1(n + n_1)$ , is used to calculate the available link capacity. This could result in poor performance, when Eq. (2) is applied, as will be shown in Figure 9 of this Section. So, for predictor A1, we let  $H_i(z) = H_{LPF}(z)$ , which satisfies the condition  $H_i(1) = 1$ .

*Predictor A2:* In designing predictor A2, we assume that we can obtain information about both the NRT traffic and the aggregate RT traffic. Hence, here we assume that per-flow information will be available. Again, for simplicity, we first consider the case of one NRT traffic source. What we do now is to minimize  $\lim_{l \rightarrow \infty} \text{Var}\{X_{n,l}\}$ , where  $\text{Var}\{X_{n,l}\}$  is defined in Eq. (9). Define

$$\delta_k = -2 \sum_{m=-(k-1)}^{k-1} (k - |m|) C_v(m), \quad (10)$$

where  $C_v(m)$  is the auto-covariance function of  $V(n)$ . We get

$$\begin{aligned} \lim_{l \rightarrow \infty} \text{Var}\{X_{n,l}\} &= - \sum_{m=0}^{M_1} B_m^{(1)} \delta_{n_1+m} \\ &+ \sum_{m_1=0}^{M_1} \sum_{m_2=m_1+1}^{M_1} B_{m_1}^{(1)} B_{m_2}^{(1)} \delta_{m_2-m_1} \end{aligned} \quad (11)$$

From Eqs. (10) and (11), we can see that  $\lim_{l \rightarrow \infty} \text{Var}\{X_{n,l}\}$  is determined by the predictor parameters and the stochastic properties of the RT traffic, but has no relation to the target utilization. Hence, we can design the predictor independently of the target utilization. Define the Lagrangian function as

$$\begin{aligned} f(B_0^{(1)}, \dots, B_{M_1}^{(1)}, \lambda) &= - \sum_{m=0}^{M_1} B_m^{(1)} \delta_{n_1+m} \\ &+ \sum_{m_1=0}^{M_1} \sum_{m_2=m_1+1}^{M_1} B_{m_1}^{(1)} B_{m_2}^{(1)} \delta_{m_2-m_1} + \lambda \left( \sum_{m=0}^{M_1} B_m^{(1)} - 1 \right) \end{aligned}$$

From  $\frac{\partial f}{\partial B_m^{(1)}} = 0$ ,  $m = 0, \dots, M_1$  and  $\frac{\partial f}{\partial \lambda} = 0$ , we get,

$$\begin{bmatrix} 0 & \cdots & \delta_{M_1} & 1 \\ \delta_1 & \cdots & \delta_{M_1-1} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \delta_{M_1} & \cdots & 0 & 1 \\ 1 & \cdots & 1 & 0 \end{bmatrix} \begin{bmatrix} B_0^{(1)} \\ B_1^{(1)} \\ \vdots \\ B_{M_1}^{(1)} \\ \lambda \end{bmatrix} = \begin{bmatrix} \delta_{n_1} \\ \delta_{n_1+1} \\ \vdots \\ \delta_{n_1+M_1} \\ 1 \end{bmatrix} \quad (12)$$

From Eq. (12), we can calculate  $B_0^{(1)}, B_1^{(1)}, \dots, B_{M_1}^{(1)}$ , and we are done. In the case of multiple NRT sources, we can still use Eq. (12) to calculate  $H_i(z)$ . What we need to do now is to replace  $M_1$  with  $M_i$ ,  $n_1$  with  $n_i$ , and  $B_0^{(1)}, \dots, B_{M_1}^{(1)}$  with  $B_0^{(i)}, \dots, B_{M_i}^{(i)}$ .

From Eq. (12), it is immediately apparent that per-flow information (i.e., round trip delay  $n_i$ ) is required. Hence, for different NRT traffic loops with different round trip delays, we need to have different predictors. This approach is the most complex of the three approaches.

*Predictor A3:* In predictor A3, we still use Eq. (12) to calculate the predictor parameters. But rather than trying to estimate the round trip delay for each NRT flow, we roughly estimate a round trip delay  $n_0$  for all NRT flows. Once the round trip delay  $n_i$  is fixed to  $n_0$ , Eq. (12) will be the same for all NRT flows. Thus, as in A1, only one predictor is needed for all NRT traffic flows (this also implies that only one virtual queue needs to be maintained). Since A3 also does explicit prediction based on the stochastic properties of the RT traffic, rather than just use smoothing, we expect that A3 will perform better than A1. In fact, what we will demonstrate through our numerical results is that the performance of A3 (that does not require per-flow information) approximates the performance of A2 (requires per-flow information) quite well.

Under the condition  $H_i(1) = 1$ , there are still other approaches to choose the predictors. For example, in the above approaches, we have only considered the stochastic properties of the RT traffic up to two moments. Higher order moments and distributions are not considered, primarily because of the complexity this would entail in practice in terms of measurements. However, given more information on the RT traffic, for example, an accurate model for the RT traffic, one can always design a better predictor.

In the following simulations, we will use the control algorithm defined in section 3.1 (single NRT source case) and in section 4 (multiple NRT sources case). From Proposition 1, we know that once  $p$  is fixed, the output link utilization will also be fixed. Note that how to choose the target

link utilization  $p$  is not the main focus of this paper. A possible way of choosing the utilization is by attempting to maximize the goodput on the link. Given the utilization and buffer size, we can approximately estimate the loss probability [13] and hence the goodput. We can then choose the utilization that gives the maximum goodput. In this section, we will simply choose a fixed utilization and compare the performance of different predictors for the same output link utilization. For predictors A2 and A3, we also need to measure the covariances of the RT traffic. This can be done by using any standard method. Also note that we need to know the order of the predictor  $M_i$  before we use Eq. (12) to calculate the predictor parameters. The larger the value of  $M_i$ , the better the performance. But the predictor will also become more complex. Hence, the choice of  $M_i$  involves a trade-off. In our simulations, we find that when  $M_i > 5$ , the improvement in performance is not significant. So, in our simulation, we set  $M_i = 5$ . The predictor designed with minimizing mean square method is called MMSE. The MMSE predictor is designed as follows. First, a low pass filter is applied to the high-priority RT traffic. Here, we use a simple moving average filter  $H_{LPF}(z) = \frac{1}{4}(1 + z^{-1} + z^{-2} + z^{-3})$  as the low pass filter. Next, a standard minimizing mean square error linear predictor  $H_{MMSE}(z)$  with the form  $\sum_{m=0}^{M_i} B_m^{(i)} z^{-n_i-m}$  is calculated based on the low frequency part of the RT traffic. And the final MMSE predictor is  $z^{-n_i} H_i(z) = H_{LPF}(z) H_{MMSE}(z)$ . Note that  $H_{MMSE}(z)$  will require explicit knowledge of the round trip delays (i.e., per-flow information).

In A1, the fixed low pass filter is chosen as  $H_{LPF}(z) = \frac{1}{4}(1 + z^{-1} + z^{-2} + z^{-3})$ . In A2, A3, and MMSE predictors,  $M_i$  is set to 5. Our simulation results are shown in Figs. 4-6. In these simulations, five NRT sources are simulated and the round trip delays are varied from 5 to 9. For A3, the roughly estimated round trip delay  $n_0$  is set to 5 (the time unit will be specified later in the description of each simulation). Note that we use the same controller (proposed in Section 4) for all predictors. We also measured the actual link utilization with different predictors. They all equal to the target link utilization.

Fig. 4 shows the result of a simulation generated with 30 actual MPEG-video traces as the RT traffic input. The mean rate of the aggregate RT traffic is  $11.6Mbps$ . We set the output link rate  $\mu = 15Mbps$  and the utilization  $p = 98\%$ . The time unit used in this simulation is  $\frac{1}{24}sec$ . The unit of the queue length is  $1kbit$ . From this figure, we can see that A1, A2, and A3 predictors all have comparable performance, but are all much better than the traditional MMSE predictor.

In Fig. 5,  $V(n)$  is a generated Gaussian process which is multi-time scale correlated with  $C_v(k) = 479.599 \times 0.999^{|k|} + 161.787 \times 0.99^{|k|} + 498.033 \times 0.9^{|k|}$  and  $\bar{V} = 100kbps$ . This type of source has

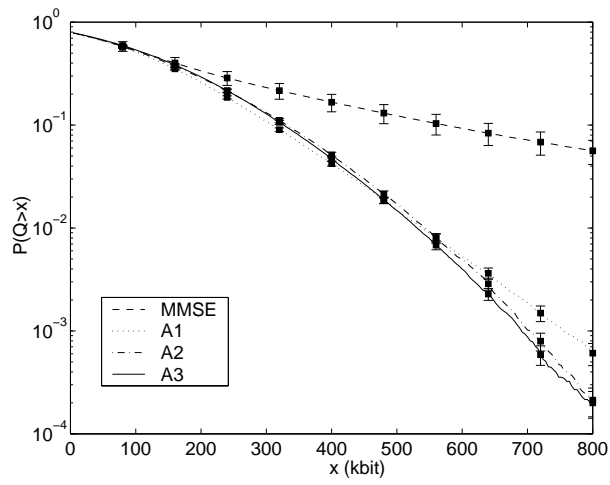


Figure 4: Tail probability with MPEG video traces as RT traffic

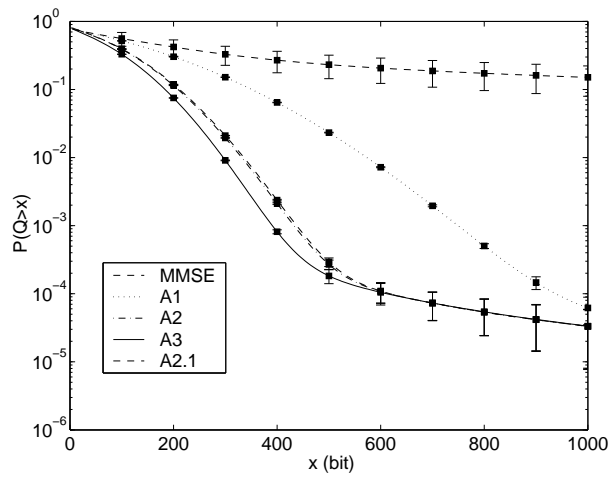


Figure 5: Tail probability with Gaussian process as RT traffic

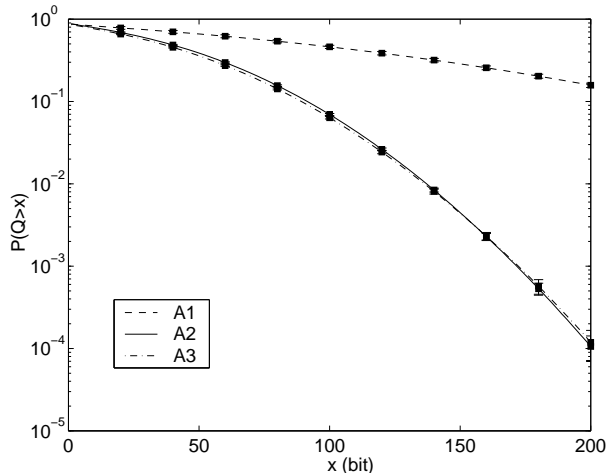


Figure 6: Tail probability with voice sources modeled by exponential *on-off* processes as RT traffic

often been used to represent the multiple time-scale correlation in network traffic [24, 15]. The link capacity is  $200\text{kbps}$ , and the utilization is set to 98%. The time unit is  $1\text{msec}$  and the unit of the queue length is  $1\text{bit}$ . In this case,  $V_{max}$  is no longer finite and our theorem will not hold. However, just as we discussed in Section 5.5, we can observe from Fig. 5, that our predictors still significantly outperform the MMSE predictor. In this case, predictors A2 and A3 have similar performance and both outperform A1, especially for small values of the buffer level  $x$ . Also note that A2.1 is designed in the same way as A2 except that we use the exact covariance of the RT traffic in designing A2.1, while in designing A2, we use the estimated covariance. Even though the error in the covariance estimation is about 10%, we can see that there is almost no difference in the performance of A2 and A2.1. We have made similar observations with all the cases that we have studied, indicating that our method of designing predictors is quite robust to the errors in covariance estimation.

In Fig. 6, 1000 multiplexed voice sources correspond to the aggregate RT traffic. Each voice source is modeled by a Markov modulated On-Off fluid process. The state transition matrix and rate vector are given as follows.

$$\text{State transition matrix : } \begin{bmatrix} 0.9833 & 0.01677 \\ 0.025 & 0.975 \end{bmatrix}$$

$$\text{Input rate vector : } \begin{bmatrix} 0 \\ 0.85 \end{bmatrix} \text{ kbps}$$

The mean rate of the aggregate RT traffic is  $341.2\text{kbps}$ . We set the link capacity to  $500\text{kbps}$  and

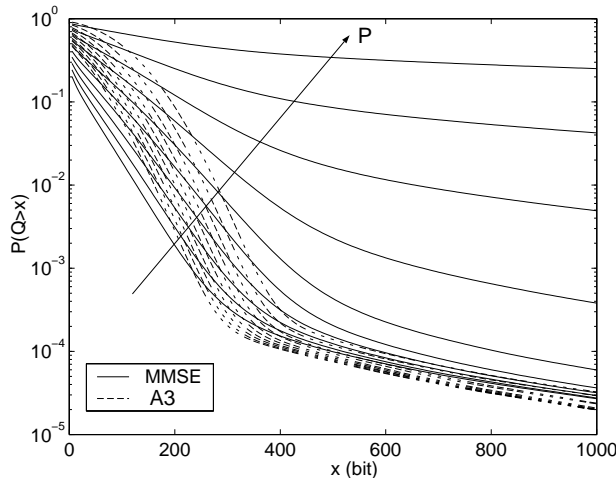


Figure 7: Tail probabilities with different utilization

the utilization to 99.8%. The time unit is  $1msec$  and the unit of queue length is  $1bit$ . In this figure, we compare the performance of three predictors A1, A2, and A3. Since all three predictors satisfy the condition  $H_i(1) = 1$ , in most cases, they perform quite well. For example, in the above two figures, A2 and A3 are better than A1, but not by a significant amount. However, in some cases, the difference between the A1 predictor and the other two predictors can be quite substantial, as is illustrated by Fig 6. This is the case, here, because the A2 and A3 predictors (unlike A1) try to predict the RT traffic at the queue at explicit times in the future, which can be done more accurately for voice types of traffic models (as is the case, here). We can also see from the above simulations that although predictor A3 does not require per-flow information, the performance of A3 is almost identical to A2 in most cases. Hence, compared to A1 and A2, the predictor A3 will be a better choice in practical networks.

To explain our main results more clearly (Theorem 1), we do the following simulation. In this simulation, we have only one NRT sources with round trip delay  $5msec$ . The RT source and the link capacities are the same as in Fig. 5. But we vary the utilization from 90% to 99% with a step of 1%. The result is shown in Fig. 7. We observe that, with an A3 predictor, the tail probability remains low even when the utilization is very high. But with the MMSE predictor, when the utilization increases, the tail probability also increases. For the MMSE predictor,  $H_1(1) = 0.878$ . When  $p = 94\%$ , we have  $\frac{p\mu - \bar{V}}{\mu - \bar{V}} = 0.88$ . So, when  $p \leq 94\%$ , the tail probabilities of the MMSE and A3 predictor are quite close. However, when  $p > 94\%$ , the difference is substantial. Hence, our

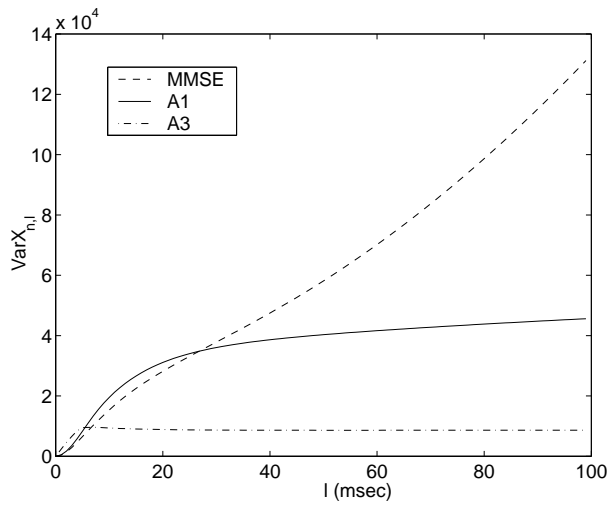


Figure 8:  $\text{Var}\{X_{n,l}\}$  with different predictors

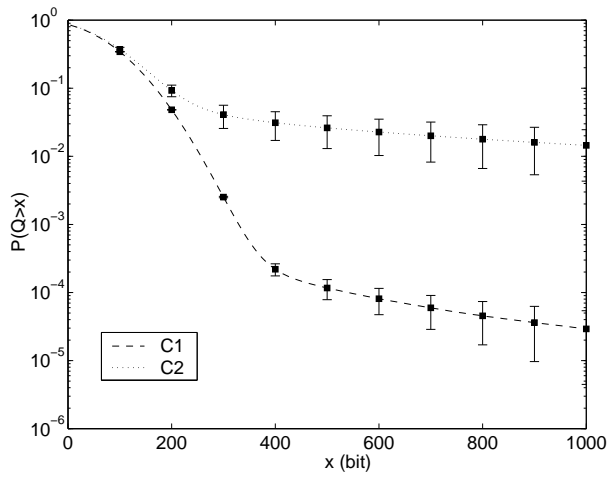


Figure 9: Tail probabilities with different control algorithms

analytical result in Theorem 1 provides a guideline as to when the MMSE predictor will work well and when it would not. We also show  $\text{Var}\{X_{n,l}\}$  in Fig. 8. We can see that since MMSE predictor does not satisfy  $H_1(1) = 1$ , when  $l$  increases,  $\text{Var}\{X_{n,l}\}$  also goes up. But for A1 and A3 predictors,  $\text{Var}\{X_{n,l}\}$  converges to some constant when  $l$  is large enough. The asymptotic variance of A3 is also smaller than that of A1. This also gives another explanation as to why A3 performs better than A1 and the MMSE predictor.

In Fig 9, we compare our control algorithm with the control algorithm that uses Eq. (2). As we mentioned in Section 3, when  $\hat{V}_1(n) \leq p\mu$  for all  $n$ , our control algorithm and Eq. (2) reduce to the same linear equation. But when this condition is not true, the two algorithms are quite different. We can see this difference in performance in Fig. 9. In this simulation, there is only one NRT source with round trip delay *5msec*. The RT source and the link capacity are still the same as in Fig. 5. To see the difference when the condition  $\hat{V}_1(n) \leq p\mu$  is violated, we use the same A3 predictor in both control algorithms. In the figure, our control algorithm is marked C1, and the one that uses Eq. (2) is marked C2. The utilization is set to 98%. But note that when using control algorithm C2, given  $p$ , the utilization is not equal to  $p$ . In this simulation, we set  $p$  such that the measured utilization for C2 is 98%, the same as in C1. From this figure, using the same predictor, we can see that our control algorithm outperforms the one that uses Eq. (2).

## 7 Conclusion

In this paper, we first present a predictive flow control model and motivate its development. We then analyze this predictive flow control system allowing both the control and the queueing process to be non-linear. Our analysis gives us insights on what parameters to choose for designing the predictor. Through simulations, we show that our predictive flow control algorithm can significantly outperform the MMSE based predictive flow control algorithm. In fact, even our predictive flow control algorithms that do not require per-flow information also do substantially better than the MMSE based algorithm that uses per-flow information. This tells us that simply minimizing the prediction error without taking into account the queueing in the system, can yield very poor performance.

Through our analysis, we are also able to determine when other predictors in the literature such as the MMSE predictor work well. For example, if a prediction scheme has  $H(1) = 1$ , it is generally expected to perform well. Our main contribution has been in solving the individual-link



level problem and to provide a scheme to predict the available link capacity when high priority traffic is present. It can be combined with other solutions for network-level problems to form a complete solution. Our scheme has also been shown to be flexible. For example, depending on the characteristic of the RT traffic, a control algorithm using a virtual queue or a simple linear control algorithm can be chosen. Similarly, depending on how much information we can obtain from the network, there are different approaches that can be used to design the predictor in our framework. In our numerical study, we compare the performance of three such approaches that use differing level of information from the network.

## References

- [1] J. Bolot and A. Shankar, “Dynamic Behavior of Rate-based Flow Control Mechanisms,” *ACM Comp. Comm. Review*, vol. 20, no. 2, pp. 35–49, 1992.
- [2] M. Hluchyj and N. Yin, “On Closed-loop Rate Control for ATM Networks,” in *Proc. INFOCOM’94*, 1994, pp. 99–108.
- [3] E. Altman, F. Baccelli, and J. Bolot, “Discrete-Time Analysis of Adaptive Rate Control Mechanisms,” *High Speed Networks and Their Performance (C-21)*, pp. 121–140, 1994.
- [4] E. Altman, T. Basar, and R. Srikant, “Congestion Control as a Stochastic Control Problem with Action Delays,” *Automatica (Special issue on Control Methods for Communication Networks)*, Dec 1999.
- [5] L. Benmohamed and S. Meerkov, “Feedback Control of Congestion in Store-and-Forward Networks: The case of Single Congestion Node,” *IEEE/ACM Transactions on Networking*, vol. 1, no. 6, pp. 693–798, Dec. 1993.
- [6] R. Jain, S. Kalyanaraman, and R. Viswandathan, “The OSU Scheme for Congestion Avoidance using Explicit Rate Indication,” Tech. Rep., OSU, Sept. 1994.
- [7] Y. D. Zhao, S. Q. Li, and S. Sigarto, “A linear dynamic model design of stable explicit-rate ABR control scheme,” in *Proceedings of IEEE INFOCOM*, 1997, pp. 283–292.

- [8] S. Q. Li, S. Chong, and C. Hwang, "Link Capacity Allocation and Network Control by Filtered Input Rate in High Speed Networks," *IEEE/ACM Transactions on Networking*, vol. 3, no. 1, pp. 10–15, Feb. 1995.
- [9] D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, NJ, 1992.
- [10] S. H. Low and D. E. Lapsley, "Optimization Flow Control, I: Basic Algorithm and Convergence," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, Dec. 1999.
- [11] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *Journal of Operations Research Society*, pp. 237–252, March 1998.
- [12] H. Yaiche, R. R. Mazumdar, and C. Rosenberg, "A Game Theoretic Framework for Bandwidth Allocation and Pricing in Broadband Networks," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 667–678, Oct. 2000.
- [13] D. Qiu and N. B. Shroff, "Queueing Properties of Feedback Flow Control Systems," in *the Fortieth Annual Allerton Conference on Communication, Control and Computing, Monticello, IL*, Oct. 2002.
- [14] O. Ait-Hellal, E. Altman, and T. Basar, "Rate based flow control with bandwidth information," in *In the proceedings of the 35th IEEE Conference on Decision and Control, Kobe, Japan*, Dec. 1996.
- [15] H. S. Kim and N. B. Shroff, "Loss Probability Calculations at a Finite Buffer Multiplexer," *IEEE/ACM Transactions on Networking*, vol. 9, no. 6, pp. 765–768, Dec. 2001.
- [16] H. S. Kim and N. B. Shroff, "On the Asymptotic Relationship between the Overflow Probability and the Loss Ratio," *Advances in Applied Probability*, vol. 33, no. 4, pp. 810–835, Dec. 2001.
- [17] A. P. Zwart, "A fluid queue with a finite buffer and subexponential input," *Advances in Applied Probability*, vol. 32, pp. 221–243, Mar. 2000.
- [18] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals & Systems*, Prentice Hall, NJ, 1997.

- [19] P. W. Glynn and W. Whitt, “Logarithmic asymptotics for steady-state tail probabilities in a single-server queue,” *Journal of Applied Probability*, pp. 131–155, 1994.
- [20] R. M. Loynes, “The Stability of a Queue with Non-independent Inter-arrival and Service Times,” *Proc. Cambridge Philos. Soc.*, vol. 58, pp. 497–520, 1962.
- [21] J. Choe and N. B. Shroff, “Use of the supremum distribution of Gaussian processes in queueing analysis with long-range dependence and self-similarity,” *Stochastic Models*, vol. 16, no. 2, Feb 2000.
- [22] N. B. Likhanov and R. Mazumdar, “Cell loss asymptotics in buffers fed by heterogeneous longtailed sources,” in *Proceedings of IEEE INFOCOM*, 2000, pp. 173–180.
- [23] D. Qiu and N. B. Shroff, “Study of Predictive Flow Control,” Tech. Rep., Purdue University, May 2001, <http://min.ecn.purdue.edu/~dongyu/paper/techrep.ps>.
- [24] J. Choe and N. B. Shroff, “A Central Limit Theorem Based Approach for Analyzing Queue Behavior in High-Speed Networks,” *IEEE/ACM Transactions on Networking*, vol. 6, no. 5, pp. 659–671, Oct. 1998.

## Appendices

### A Proof of Lemma 1

For  $q_1(n)$ , we have,  $q_1(0) = 0$  and

$$q_1(n) = [q_1(n-1) + \hat{V}_1(n) - p\mu]^+. \quad (13)$$

For  $q_2(n)$ , we have,  $q_2(0) = 0$  and

$$q_2(n) = [q_2(n-1) + \hat{V}_1(n) + a(n) - p\mu]^+,$$

where  $a(n) = a_1(n - n_1)$ . For  $n \leq n_1$ ,  $a(n) = a_1(n - n_1) = 0$ , Hence, for  $n \leq n_1$ ,  $q_2(n) = q_1(n)$ . For  $n \geq n_1$ , we will use induction to prove  $q_2(n) = q_1(n)$ . Assuming that  $q_2(n-1) = q_1(n-1)$ , for  $n \geq n_1$ , we have,

$$a(n) = a_1(n - n_1) = [p\mu - \hat{V}_1(n) - q_1(n-1)]^+.$$

Next, we will discuss two cases. When  $p\mu - \hat{V}_1(n) - q_1(n-1) \geq 0$ , we have  $q_1(n) = [q_1(n-1) + \hat{V}_1(n) - p\mu]^+ = 0$  and

$$\begin{aligned} q_2(n) &= [q_2(n-1) + \hat{V}_1(n) + a(n) - p\mu]^+ \\ &= [q_2(n-1) + \hat{V}_1(n) + p\mu - \hat{V}_1(n) \\ &\quad - q_1(n-1) - p\mu]^+ \\ &= [q_2(n-1) - q_1(n-1)]^+ = 0. \end{aligned}$$

When  $p\mu - \hat{V}_1(n) - q_1(n-1) < 0$ ,  $a(n) = 0$ ,

$$\begin{aligned} q_2(n) &= [q_2(n-1) + \hat{V}_1(n) - p\mu]^+ \\ &= [q_1(n-1) + \hat{V}_1(n) - p\mu]^+ \\ &= q_1(n) \end{aligned}$$

In both cases, we have

$$q_2(n) = q_1(n).$$

■

## B Proof of Proposition 1

Throughout this proof, we use the definitions of  $q_1(n)$  and  $q_2(n)$  defined in Lemma 1.

For any  $j > n_1$ , we have,

$$\begin{aligned} a(j) = a_1(j - n_1) &= [p\mu - \hat{V}_1(j) - q_1(j-1)]^+ \\ &= [p\mu - \hat{V}_1(j) - q_2(j-1)]^+ \\ &\geq p\mu - \hat{V}_1(j) - q_2(j-1). \end{aligned}$$

So,

$$\begin{aligned} a(j) + \hat{V}_1(j) + q_2(j-1) - p\mu &\geq 0 \\ q_2(j) &= [q_2(j-1) + a(j) + \hat{V}_1(j) - p\mu]^+ \\ &= q_2(j-1) + a(j) + \hat{V}_1(j) - p\mu. \\ a(j) &= q_2(j) - q_2(j-1) + p\mu - \hat{V}_1(j) \\ &= q_1(j) - q_1(j-1) + p\mu - \hat{V}_1(j). \end{aligned}$$

Because  $a(j) = 0$  for  $j \leq n_1$ , we have,

$$\begin{aligned} \sum_{j=1}^n a(j) &= \sum_{j=n_1+1}^n a(j) \\ &= q_1(n) - q_1(n_1) + (n - n_1)p\mu - \sum_{j=n_1+1}^n \hat{V}_1(j) \end{aligned} \quad (14)$$

Next, we focus on  $q_1(n) - q_1(n_1)$ . Let  $q_1$  and  $q_2$  be the queueing systems with queue lengths  $q_1(n)$  and  $q_2(n)$ , respectively. Further, let  $\mu(n)$  be the amount of traffic that leaves  $q_1$  at time  $n$ . We then have,

$$\begin{aligned} q_1(n_1) + \sum_{j=n_1+1}^n \hat{V}_1(j) &= q_1(n) + \sum_{j=n_1+1}^n \mu(j) \\ q_1(n) - q_1(n_1) &= \sum_{j=n_1+1}^n \hat{V}_1(j) - \sum_{j=n_1+1}^n \mu(j). \end{aligned}$$

Because  $\bar{V} < p\mu$ ,  $q_1$  will be a stable system. Hence,

$$\lim_{n \rightarrow \infty} \frac{\sum_{j=n_1+1}^n \mu(j)}{n} = \lim_{n \rightarrow \infty} \frac{\sum_{j=n_1+1}^n \hat{V}_1(j)}{n} = \bar{V},$$

i.e.,

$$\lim_{n \rightarrow \infty} \frac{q_1(n) - q_1(n_1)}{n} = 0. \quad (15)$$

So, from Eqs. (14) and (15),

$$\lim_{n \rightarrow \infty} \frac{\sum_{j=1}^n a(j)}{n} = p\mu - \bar{V}. \quad (16)$$

## C Proof of the Theorems

### C.1 Proof of Theorem 1

First, we will prove two lemmas that will be needed later to prove Theorem 1.

For the next lemma, we define two queueing systems with service rate  $\mu'$ . The first queueing system has input rate  $\hat{V}_1(n)$  and corresponding queue length  $q'_1(n)$ , while the second queueing system has input rate  $a(n) + \hat{V}_1(n)$  (where  $a(n)$  is determined by our flow control algorithm) and queue length  $q'_2(n)$ .

**Lemma 2** *Under the above definitions, if  $\mu' \geq p\mu$ , we have  $q'_2(n) = q'_1(n)$ , for any  $n \geq 0$ .*

*Proof:* From Lemma 1 and our algorithm, we know that  $a(n) = [p\mu - \hat{V}_1(n) - q_1(n-1)]^+$ ,  $n > n_1$  and  $a(n) = 0$ ,  $n \leq n_1$ . Now, we define  $a'(n)$  as,  $a'(n) = [\mu' - \hat{V}_1(n) - q'_1(n-1)]^+$ ,  $n > n_1$  and  $a'(n) = 0$ ,  $n \leq n_1$ .  $q_1$  and  $q'_1$  have the same input traffic rate  $\hat{V}_1(n)$ . But the output link rates are  $p\mu$  and  $\mu'$  respectively. Because  $\mu' \geq p\mu$ , we have  $q_1(n) \geq q'_1(n)$  for any  $n$ . So, we have  $a(n) \leq a'(n)$  for any  $n \geq 0$ .

Now, consider a queueing system with  $\mu'$  as the output link rate and  $a'(n) + \hat{V}_1(n)$  as the input traffic rate. We denote the queue length as  $q'_3(n)$ . Comparing  $q'_2$  and  $q'_3$ , we find they have the same output link rate. But since  $a(n) \leq a'(n)$  for any  $n$ , the input traffic rate of  $q'_2$  will always be less than or equal to the input traffic rate of  $q'_3$ . Hence, we have  $q'_2(n) \leq q'_3(n)$  for any  $n$ .

Similar to the proof of Lemma 1, we can also prove that  $q'_1(n) = q'_3(n)$ . Hence,  $q'_2(n) \leq q'_1(n)$ . Also, obviously,  $q'_2(n) \geq q'_1(n)$ . Thus,

$$q'_2(n) = q'_1(n).$$

■

It should be noted that the requirement that  $\mu' \geq p\mu$  is necessary for stability because our objective is to utilize the link at a level  $p\mu$ .

**Lemma 3** Define  $D(j) = H_1(1)(V(j) - \bar{V}) - (\hat{V}_1(j) - \bar{V})$ . If  $V_{max} < \infty$ , there exists a constant  $C_1$  such that  $\left| \sum_{j=n_0+1}^n D(j) \right| \leq C_1$  for any  $n \geq 0$  and  $0 \leq n_0 \leq n$ .

*Proof:* Let  $D(z)$  be the Z-transform of  $D(j)$ . We have,

$$D(z) = H_1(1)V(z) - \hat{V}_1(z) = (H_1(1) - z^{-n_1}H_1(z))V(z)$$

Because  $z = 1$  is a zero point of  $H_1(1) - z^{-n_1}H_1(z)$ , we can write the above equation as:

$$D(z) = (1 - z^{-1})H(z)V(z),$$

where  $H(z)$  is still causal and stable. Now, define  $Y(n) = \sum_{i=0}^n D(i)$ . We have,

$$Y(z) = \frac{1}{1 - z^{-1}}D(z) = H(z)V(z)$$

Because  $V(n)$  is bounded and  $H(z)$  is a stable linear system, we know that  $Y(n)$  will also be bounded. So,  $\sum_{j=n_0+1}^n D(j) = Y(n) - Y(n_0)$  is also bounded, i.e, there exists a constant  $C_1$  such that for any  $n \geq 0$  and  $0 \leq n_0 \leq n$ ,

$$\left| \sum_{j=n_0+1}^n D(j) \right| \leq C_1$$

■

Now, we can prove Theorem 1. First, we define  $\Delta(j) = V(j) - \hat{V}_1(j)$ .

$$\begin{aligned}\Delta(j) &= V(j) - \hat{V}_1(j) = (V(j) - \bar{V}) - (\hat{V}_1(j) - \bar{V}) \\ &= D(j) + (1 - H_1(1))(V(j) - \bar{V}).\end{aligned}$$

$$\begin{aligned}a(j) + V(j) - \mu &= a(j) + \hat{V}_1(j) + \Delta(j) - \mu \\ &= a(j) + \hat{V}_1(j) + D(j) + (1 - H_1(1))(V(j) - \bar{V}) - \mu \\ &= a(j) + \hat{V}_1(j) - \mu' + D(j) + (1 - H_1(1))(V(j) - \mu),\end{aligned}$$

where  $\mu' = H_1(1)\mu + (1 - H_1(1))\bar{V}$ .

$$\begin{aligned}q(n) &= \sup_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^n (a(j) + V(j) - \mu) \right\} \\ &\leq \sup_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^n (a(j) + \hat{V}_1(j) - \mu') \right\} \\ &\quad + \sup_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^n D(j) \right\} \\ &\quad + (1 - H_1(1)) \sup_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^n (V(j) - \mu) \right\} \\ &\leq q'_2(n) + \sup_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^n D(j) \right\} + (1 - H_1(1))q_0(n).\end{aligned}$$

If  $H_1(1) \geq \frac{p\mu - \bar{V}}{\mu - \bar{V}}$ , we will have  $\mu' \geq p\mu$ . From Lemma 2 we know,

$$q'_2(n) = q'_1(n) = \sup_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^n (\hat{V}_1(j) - \mu') \right\}.$$

$$\begin{aligned}\hat{V}_1(j) - \mu' &= (\hat{V}_1(j) - \bar{V}) - H_1(1)(V(j) - \bar{V}) \\ &\quad + H_1(1)(V(j) - \bar{V}) + \bar{V} - \mu' \\ &= -D(j) + H_1(1)(V(j) - \mu).\end{aligned}$$

So,

$$\begin{aligned}q'_2(n) &= q'_1(n) = \sup_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^n (\hat{V}_1(j) - \mu') \right\} \\ &\leq \sup_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^n -D(j) \right\}\end{aligned}$$

$$\begin{aligned}
& + H_1(1) \sup_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^n (V(j) - \mu) \right\} \\
\leq & - \inf_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^n D(j) \right\} + H_1(1) q_0(n).
\end{aligned}$$

$$\begin{aligned}
q(n) & \leq q_2'(n) + (1 - H_1(1)) q_0(n) \\
& + \sup_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^n D(j) \right\} \\
& \leq q_0(n) + \sup_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^n D(j) \right\} \\
& - \inf_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^n D(j) \right\} \\
& \leq q_0(n) + 2C_1.
\end{aligned}$$

■

## C.2 Proof of Theorem 2

First, we let  $Q_i(n)$  denote the queue length of a queue with  $Na_i(n - n_i) + V(n)$  as the input traffic rate and  $\mu$  as output link rate. From the definition of  $a_i(n)$  and Theorem 1, we know, there exists a constant  $C_i$  such that

$$Q_i(n) \leq q_0(n) + 2C_i,$$

where  $q_0(n)$  is the queue length of a queue without NRT traffic.

Because the input rate to  $q$  is  $a(n) + V(n)$ , we have,

$$\begin{aligned}
q(n) & = \sup_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^n (a(j) + V(j) - \mu) \right\} \\
& = \sup_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^n \left( \sum_{i=1}^N a_i(j - n_i) + V(j) - \mu \right) \right\} \\
& = \sup_{0 \leq n_0 \leq n} \left\{ \frac{1}{N} \sum_{i=1}^N \sum_{j=n_0+1}^n (Na_i(j - n_i) + V(j) - \mu) \right\} \\
& \leq \frac{1}{N} \sum_{i=1}^N \left( \sup_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^n (Na_i(j - n_i) + V(j) - \mu) \right\} \right) \\
& = \frac{1}{N} \sum_{i=1}^N Q_i(n) \leq \frac{1}{N} \sum_{i=1}^N (q_0(n) + 2C_i) \leq q_0(n) + 2C,
\end{aligned}$$

where  $C = \max_{1 \leq i \leq N} C_i$ .

■



## D Details of Section 5.5

### D.1 Proof of Eq. (8)

$$\begin{aligned}
q(n) &= \sup_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^n (a(j) + V(j) - \mu) \right\} \\
&= \sup_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^n (a(j) + \hat{V}_1(j) - \mu + \Delta(j)) \right\} \\
&\leq \sup_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^n (a(j) + \hat{V}_1(j) - \mu) \right\} + \sup_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^n \Delta(j) \right\}
\end{aligned}$$

Because we use the control algorithm proposed in Section 3.1, from the proof of Lemma 1, we have

$$\sup_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^n (a(j) + \hat{V}_1(j) - \mu) \right\} = \sup_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^n (\hat{V}_1(j) - \mu) \right\}.$$

Hence,

$$\begin{aligned}
q(n) &\leq \sup_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^n (\hat{V}_1(j) - \mu) \right\} + \sup_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^n \Delta(j) \right\} \\
&= \sup_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^n (V(j) - \mu - \Delta(j)) \right\} + \sup_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^n \Delta(j) \right\} \\
&\leq \sup_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^n (V(j) - \mu) \right\} + \sup_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^n (-\Delta(j)) \right\} + \sup_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^n \Delta(j) \right\} \\
&= q_0(n) + \sup_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^n \Delta(j) \right\} - \inf_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^n \Delta(j) \right\}
\end{aligned}$$

■

### D.2 Proof of Proposition 2

Since  $\text{Var}\{X_{n,l}\}$  does not depend on  $n$ , we define  $S(l) = \text{Var}\{X_{n,l}\}$  and let  $S(z) = \sum_{l=0}^{\infty} S(l)z^{-l}$  be the Z-transform of  $S(l)$ . From Eq. (9), we get,

$$S(l) = \sum_{j_1=-l+1}^0 \sum_{j_2=-l+1}^0 C_{\Delta}(j_1 - j_2) = \sum_{j=-l+1}^{l-1} (l - |j|)C_{\Delta}(j) = \sum_{j=-l}^l (l - |j|)C_{\Delta}(j).$$

So,

$$S(z) = \sum_{l=0}^{\infty} S(l)z^{-l} = \sum_{l=0}^{\infty} \sum_{j=-l}^l (l - |j|)C_{\Delta}(j)z^{-l}$$

$$\begin{aligned}
&= \sum_{j=-\infty}^{\infty} \sum_{l=|j|}^{\infty} (l-|j|)C_{\Delta}(j)z^{-l} = \sum_{j=-\infty}^{\infty} C_{\Delta}(j)z^{-|j|} \sum_{l=|j|}^{\infty} (l-|j|)z^{-(l-|j|)} \\
&= \sum_{j=-\infty}^{\infty} C_{\Delta}(j)z^{-|j|} \sum_{n=0}^{\infty} nz^{-n} \quad (n=l-|j|) \\
&= C_{\Delta}^{(+)}(z) \frac{z^{-1}}{(1-z^{-1})^2}, \tag{17}
\end{aligned}$$

where  $C_{\Delta}^{(+)}(z) = \sum_{n=-\infty}^{\infty} C_{\Delta}(n)z^{-|n|} = C_{\Delta}(0) + 2\sum_{n=1}^{\infty} C_{\Delta}(n)z^{-n}$ . Let  $C_{\Delta}(z) = \sum_{n=-\infty}^{\infty} C_{\Delta}(n)z^{-n}$  be the Z-transform of  $C_{\Delta}(n)$ . We can easily see that

$$C_{\Delta}(z) = \frac{1}{2} \left( C_{\Delta}^{(+)}(z) + C_{\Delta}^{(+)}(z^{-1}) \right). \tag{18}$$

So, if we can decompose  $C_{\Delta}(z)$  to be in the above form, we can get  $C_{\Delta}^{(+)}(z)$ . Now, let  $C_v(n)$  be the auto-covariance function of  $V(n)$  and let  $C_v(z) = \sum_{n=-\infty}^{\infty} C_v(n)z^{-n}$  be the Z-transform of  $C_v(n)$ . Since,

$$\Delta(n) = V(n) - \hat{V}_1(n) = (V(n) - \bar{V}) - (\hat{V}_1(n) - \bar{V}),$$

we have

$$\Delta(z) = (1 - z^{-n_1} H_1(z))V(z).$$

Now, if  $H_1(1) = 1$ ,  $z = 1$  will be a zero point of  $1 - z^{-n_1} H_1(z)$ . We can write the above equation as:

$$\Delta(z) = (1 - z^{-1})H(z)V(z) = (1 - z^{-1})Y(z),$$

where  $H(z)$  is still causal and stable, and  $Y(z) = H(z)V(z)$  is the Z-transform of  $Y(n)$ , which is the output of a linear system  $H(z)$  with  $V(n) - \bar{V}$  as the input. Let  $C_y(n)$  be the auto-covariance function of  $Y(n)$  and  $C_y(z)$  be the Z-transform of  $C_y(n)$ . We have,

$$\begin{aligned}
C_y(z) &= \sum_{n=-\infty}^{\infty} C_y(n)z^{-n} = C_y(0) + \sum_{n=1}^{\infty} C_y(n)z^{-n} + \sum_{n=-\infty}^{-1} C_y(n)z^{-n} \\
&= C_y(0) + C_y^{(+)}(z) + C_y^{(+)}(z^{-1}),
\end{aligned}$$

where  $C_y^{(+)}(z) = \sum_{n=1}^{\infty} C_y(n)z^{-n}$ . Hence,

$$\begin{aligned}
C_{\Delta}(z) &= (1 - z^{-1})(1 - z)C_y(z) = (1 - z^{-1})(1 - z) \left( C_y(0) + C_y^{(+)}(z) + C_y^{(+)}(z^{-1}) \right) \\
&= \left( (1 - z^{-1})C_y(0) + (1 - z^{-1})(1 - z)C_y^{(+)}(z) \right) \\
&\quad + \left( (1 - z)C_y(0) + (1 - z^{-1})(1 - z)C_y^{(+)}(z^{-1}) \right). \tag{19}
\end{aligned}$$

Comparing Eqs. (18) and (19), we can see that

$$\begin{aligned} C_{\Delta}^{(+)}(z) &= 2\left((1 - z^{-1})C_y(0) + (1 - z^{-1})(1 - z)C_y^{(+)}(z)\right) \\ &= 2(1 - z^{-1})\left(C_y(0) + (1 - z)C_y^{(+)}(z)\right). \end{aligned}$$

So, from Eq. (17), we have

$$\begin{aligned} S(z) &= C_{\Delta}^{(+)}(z)\frac{z^{-1}}{(1 - z^{-1})^2} = 2(1 - z^{-1})\left(C_y(0) + (1 - z)C_y^{(+)}(z)\right)\frac{z^{-1}}{(1 - z^{-1})^2} \\ &= \frac{2z^{-1}}{(1 - z^{-1})}C_y(0) - 2C_y^{(+)}(z). \end{aligned} \tag{20}$$

From Eq. (20), it follows that

$$S(l) = 2(C_y(0) - C_y(l)), \tag{21}$$

for any  $l \geq 0$ . Since  $C_y(0) = \text{Var}\{Y(n)\}$  is finite, Eq. (21) tell us that when  $H_1(1) = 1$ ,  $S(l)$  is bounded and  $\lim_{l \rightarrow \infty} S(l) = 2C_y(0)$ . ■

From the proof, we can also see that the reason that  $S(l)$  is bounded is that  $z = 1$  is a zero point of  $1 - z^{-n_1}H_1(z)$  and hence it is also a zero point of  $C_{\Delta}^{(+)}(z)$ . When  $H_1(1) \neq 1$ ,  $z = 1$  will no longer be a zero point of  $1 - z^{-n_1}H_1(z)$ . From Eq. (17), we can see that in most cases,  $z = 1$  will be a second order pole point of  $S(z)$ . Hence, when  $l \rightarrow \infty$ ,  $S(l)$  will go to infinity.