UNIVERSITY OF CALIFORNIA, SAN DIEGO

A Framework for Performance Guarantees in Adaptive Communication Networks

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
in Electrical and Computer Engineering
(Communication Theory and Systems)

by

Clayton M. Okino

Committee in charge:

Professor Rene L. Cruz, Chairperson
Professor Elias Masry
Professor George Polyzos
Professor Ramesh Rao
Professor Gill Williamson

1998

UMI Number: 9912847

The dissertation of Clayton M. Okino is approved, and it is acceptable in quality and form for publication on microfilm:

Elias Masry

Chair

University of California, San Diego

1998

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

To my thesis advisor, Rene Cruz, whom I express my deepest gratitude, for his guidance, encouragement, and support. Without his insight, patience, and belief in my abilities, this work would not have been possible.

I would like to thank my committee members - Ramesh Rao, Elias Masry, George Polyzos, and Gill Williamson for serving on my committee, reading my dissertation, and providing valuable comments. I would also like to thank Anthony Acampora for his valuable suggestions. A special thanks goes to Rajeev Agrawal and Rajendran Rajan for their collaboration efforts with Rene and I. To my colleagues and friends An Chen, Srikanth Krishnamurthy, Vassiriki Traore, Jeff Foerster, Prasad Shamain, Saleh Al-Harthi, Kevin Quirk, Lucy Chong, Sami Ayyorgun, and Haisong Cai, thank you all for the memorable and enjoyable experiences in the Communication Theory & Systems Lab.

Research supported in part by the National Science Foundation under grants NCR-93-05018 and NCR 94-15684, the Army Research Office under grant FRI DAAH04-95-1-0248, the Air Force Office of Scientific Research under grant F49620-95-1-0538, and by the Center for Wireless Communications at UCSD.

Finally, I would like to thank my mother and father for their immeasurable love and support. I dedicate this thesis to them with the utmost love and respect.

VITA

December 3. 1967	Born, Honolulu, Hawaii
1989	B.S. Oregon State University, Corvallis. Oregon
1989–1994	Member of the Technical Staff Applied Signal Technology. Sunnyvale. California
1993	M.S. Santa Clara University. Santa Clara. California
1994–1998	Teaching Assistant and Research Assistant Department of Electrical and Computer Engineering University of California. San Diego
1998	Doctor of Philosophy University of California. San Diego

FIELDS OF STUDY

Major Field: Electrical Engineering

Studies in Communication Networks.
Professors Rene Cruz. Ramesh Rao. Anthony Acampora

Studies in Queueing Theory.
Professor Elias Masry

Studies in Digital Communications.
Professor Laurence Milstein

ABSTRACT OF THE DISSERTATION

A Framework for Performance Guarantees in Adaptive Communication Networks

by

Clayton Okino
Doctor of Philosophy in Electrical Engineering
(Communication Theory and Systems)
University of California, San Diego, 1998
Professor Rene L. Cruz, Chair

We consider adaptive sessions using closed loop window based flow control in order to avail of excess bandwidth. To model service guarantees for adaptive sessions, we present a framework based on an *adaptive service guarantee.* We analyze the behavior of closed loop window based flow control and present some performance bounds. Vital to the flow control mechanism utilizing excess bandwidth are scheduling algorithms that allow the use of excess bandwidth without leading to unbounded maximum delay. We present an element called the *elastic regulator,* which in conjunction with a scheduling algorithm *synthesized* to guarantee a service curve, is capable of providing adaptive service guarantees.

Chapter 1

Introduction

This dissertation is a study of performance issues for adaptive sessions in integrated services networks. Specifically, we consider providing service guarantees to adaptive applications which utilize excess bandwidth when it is available. In order to avail of excess bandwidth, these sessions require a feedback mechanism. We consider closed loop window based flow control, a method known to provide throughput guarantees [16] while potentially allowing for full utilization of link bandwidth.

In high speed integrated services networks, closed loop window based flow control may be an efficient protocol for delay tolerant applications, while still providing a pre-negotiated level of service. Closed loop window based flow control [4] can directly eliminate the possibility of buffer overflow in a network by relying on feedback. It has the potential for providing statistical multiplexing of user sessions, and to avail of excess bandwidth in the network. In reality, networks with large bandwidth delay products may require large window sizes for a window based flow control protocol, and so it has been proposed that closed loop window based flow control be used for delay tolerant applications (e.g. Available Bit Rate (ABR) traffic, file transfers, email, web browsing), utilizing bandwidth that would otherwise go unused by the network.

We consider the unicast session as depicted in Figure 1.1. In this example,

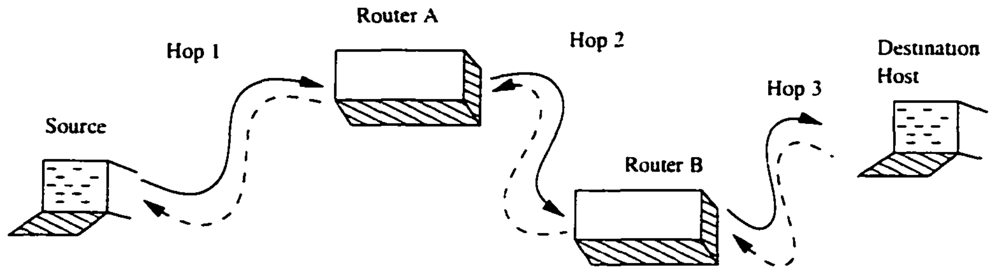


Figure 1.1: A unicast session with window flow control.

the session originates at a source and traverses two routers before terminating at the destination host. The router at the first hop returns acknowledgments for each packet received back to the source. Similarly, the router at the second hop returns acknowledgements back to the first router, and the destination returns acknowledgements back to the second router. All three hops use windows of size W to control the number of unacknowledged packets, i.e. hop-by-hop window based flow control. In order for the source to obtain a pre-negotiated level of service, each router must allocate buffers to prevent packet loss, and guarantee access to link bandwidth through the operation of a scheduling mechanism. Thus, a scheduling algorithm that is capable of providing service guarantees while allowing the use of excess bandwidth is of interest.

Service guarantees for window flow control have previously been obtained in [13, 2, 6], in terms of a service curve [10, 2, 21, 5, 12]. A service curve describes a non-probabilistic characterization of service provided by a network element, which implies a *throughput* guarantee. Upper bounds on *delay* for a window flow control protocol can then be obtained if the arrival traffic is constrained by an envelope, as in the mathematical framework proposed by Cruz [7, 8]. However, if the traffic generated by an adaptive application is constrained by an envelope, it may be unable to utilize excess bandwidth offered by the network. Thus, if an adaptive application wishes to utilize excess bandwidth, it may be impossible to guarantee any upper bound on maximum delay, using the concept of a service curve.

In this thesis, we develop a new mathematical framework for obtaining *both* delay and throughput guarantees, that can be applied to such adaptive applications. We propose a new "adaptive service guarantee," that implies throughput guarantees as well as delay guarantees, without requiring arrival traffic to be constrained by an envelope. We shall present composition results, which yield adaptive service guarantees for end-to-end systems whose components themselves are described in terms of adaptive service guarantees. In particular, we shall derive adaptive service guarantees provided by window flow control protocols. We shall also present a new scheduling policy that is capable of providing adaptive service guarantees to sessions that share a fixed capacity server.

In the remainder of this chapter, we review some of the network calculus developed in [7, 8, 21, 10, 2, 5, 6, 12], including the the concept of a service curve. We review some of our previous work for a closed loop window flow control model. In addition, we discuss a scheduling algorithm which synthesizes service curve guarantees [21, 22, 10, 11], which forms the basis for our new scheduling policy that provides adaptive service guarantees.

In Chapter 2, we present the new adaptive service definition, and the associated network calculus. Here, we will obtain adaptive service guarantees for window flow control protocols, which imply upper bounds on delay as well as throughput guarantees. Also, in this chapter, we present the new scheduling algorithm that provides adaptive service guarantees.

We conclude the thesis in Chapter 3 with a brief discussion of possible future directions for research.

1.1 Review of the Network Calculus

In this subsection, we state some well known terminology, definitions, notation and performance bounds.

1.1.1 Terminology

Consider a network element, where R_{in} is the arrival process to the network element and R_{out} is the departure process. We define a *process* as a function of time $R(t)$, where R is a mapping from real numbers into the extended non-negative real numbers, i.e. $R : \mathbb{R} \to \mathbb{R} + \cup\{+\infty\}$. All processes are assumed to be non-decreasing and right continuous. We will often refer to *causal* processes, which by definition are processes that are identically zero for all negative time.

The *backlog* of the network element is defined as the amount of traffic stored in the network element. The backlog at time t is $B(t)$ where

$$B(t) = R_{in}(t) - R_{out}(t) \ .$$

assuming that the network is empty at time 0. We say that a network element has *positive backlog* at time t if $B(t) > 0$ at time t.

Now suppose we have a network element that serves data in the same order that it arrives. For a packet arriving at time t and departing at time $t + d$, clearly we have $R_{out}(t + d) \geq R_{in}(t)$. The *virtual delay* of this packet at time t is $d(t)$, where

$$d(t) = \inf\{\Delta : \Delta \geq 0 \text{ and } R_{out}(t + \Delta) \geq R_{in}(t)\} \ .$$

1.1.2 Service Curves

In order to define a service guarantee, we first need to define the *convolution*. Given two processes, A and B, the convolution of A and B, defined to be the function $A * B : \mathbb{R} \to \mathbb{R}_+ \cup \{+\infty\}$, is such that for all t

$$A * B(t) = \inf_{\tau : \tau \in \mathbb{R}} \{A(\tau) + B(t - \tau)\} \ .$$

It is easy to verify that $A * B$ is a process, i.e. it is non-decreasing and right continuous. Moreover, if A and B are causal, then $A * B$ is causal. We now discuss a graphical interpretation of the convolution as illustrated in Figure 1.2.

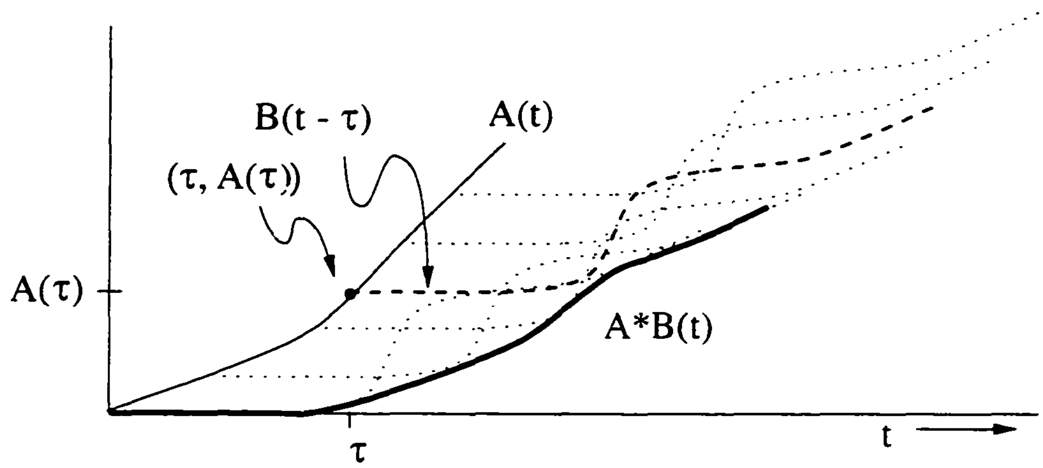


Figure 1.2: Graphical representation of convolution $A * B$.

For a fixed value of τ, the graph of $A(\tau) + B(t - \tau)$ versus t is obtained from the graph of $B(t)$ by horizontally shifting it by an amount τ and vertically shifting it by an amount $A(\tau)$. Essentially, we translate the graph of $B(t)$ by moving the origin of the graph onto the point $(\tau, A(\tau))$. By taking the pointwise minimum of all such translations of the graph of B onto the graph of A, we obtain the convolution. This is illustrated graphically in Figure 1.2.

It can be shown that the convolution is **commutative**(i.e. $A*B = B*A$) and **associative** (i.e. $(A * B) * C = A * (B * C)$). Furthermore, the convolution is **distributive** pointwise, i.e. $A * (B \wedge C) = (A * B) \wedge (A * C)$, where $B \wedge C$ is the pointwise minimum of B and C. Similarly, the pointwise maximum of B and C is $B \vee C$. Often, we will use the notation $[A]^+$, the pointwise maximum of A and 0, i.e. $[A]^+ = A \vee 0 = \max\{A, 0\}$. The identity element, δ, of the convolution satisfies $R * \delta = \delta * R = R$ and can be defined as

$$\delta(t) = \begin{cases} 0 & , \text{for all } t < 0 \\ +\infty & , \text{for all } t \geq 0 \end{cases} .$$

We then define a "delay shift" element as $\delta_d(t) = \delta(t - d)$. Note that $A * \delta_d(t) = A(t - d)$ for any process A.

Definition 1 (Minimum Service Curve [2] [5] [21]): Suppose S is a process. A network element is said to guarantee (deliver) a minimum service curve of S if for all arrival processes R_{in}, the departure process of the network element satisfies

$$R_{out} \geq R_{in} * S \ .$$

Definition 2 (Maximum Service Curve [12]): Suppose $\bar{S}$ is a process. A network element is said to guarantee (deliver) a maximum service curve of $\bar{S}$ if for all arrival processes R_{in}, the departure process of the network element satisfies

$$R_{out} \leq R_{in} * \bar{S} \ .$$

We define a *service curve element* [1] with minimum service curve S (maximum service curve $\bar{S}$) as a network element guaranteeing minimum service curve S (maximum service curve $\bar{S}$).

1.1.3 Envelopes

A process E is said to be an *envelope* for the process R if for all $\tau \leq t$ we have $R(t) - R(\tau) \leq E(t - \tau)$, or equivalently $R \leq R * E$. The concept of an envelope was proposed and developed in [7].

If E is an envelope for R then $E \wedge \delta$ is a causal envelope for R, since any process R is non-decreasing. If E is a causal envelope, note that $E \leq \delta$, and hence $R*E \leq R*\delta = R$. Thus, if E is a causal envelope for R, then $R = R*E$. A process E is said to be *sub-additive* if for all $t, \tau \in \mathbb{R}$ we have $E(\tau) + E(t - \tau) \geq E(t)$. Thus, if E is a sub-additive process, then $E * E \geq E$. We will often assume that envelopes are sub-additive.

In this paper we will often refer to a network element called a *regulator* and so for completeness it is defined here. The departure process R_{out} of a regulator with causal sub-additive envelope E and arrival process R_{in} is defined to be $R_{out} = R_{in} * E$, and it satisfies the following conditions:

R1. E is an envelope for R_{out}, i.e. $R_{out} \leq R_{out} * E$.

R2. $R_{out} \leq R_{in}$.

R3. R_{out} is the (pointwise) maximal function satisfying **R1** and **R2**.

We now define *deconvolution* which is used later in the paper. Given two processes A and B, where B is causal, it is useful to consider the smallest process F such that $A(t+\tau) - B(\tau) \leq F(t)$ for all $t, \tau \in \mathbb{R}$. The smallest such process is clearly given by $F = A \oslash B$, where

$$A \oslash B(t) = \sup_{\tau \in \mathbb{R}} \{A(t+\tau) - B(\tau)\}.$$

It can be shown that $A \oslash B$ is the smallest process F such that $F * B \geq A$. For this reason, we call $A \oslash B$ the *deconvolution* of A and B.

1.1.4 Performance Bounds

We consider a network element with arrival process R_{in} and departure process R_{out}. We will assume that E is an envelope for the arrival process R_{in}, and that the network element guarantees the minimum and maximum service curves S and $\bar{S}$, respectively. The results of this subsection appeared in [2], [5], [21], [12], and in an earlier form in [10].

Under these conditions, we state a bound on the delay through the network element. Let d_{max} be the maximum horizontal distance between E and S. In other words, d_{max} is how far the graph of E must be shifted to the right so that it lies below S:

$$d_{max} := \inf\{d : d \geq 0 \ , \ E * \delta_d \leq S\} \ . \tag{1.1}$$

It is easy to verify that $E * \delta_{d_{max}} \leq S$. Figure 1.3 is an example illustrating d_{max}.

Theorem 3 (Delay Bound): Given a service curve element with minimum service curve S, if the arrival traffic has envelope E, then the virtual delay is upper bounded by d_{max}.

The performance bounds on delay for end-to-end guaranteed service for a session with n service curves in tandem can easily be determine using the following composition rule.

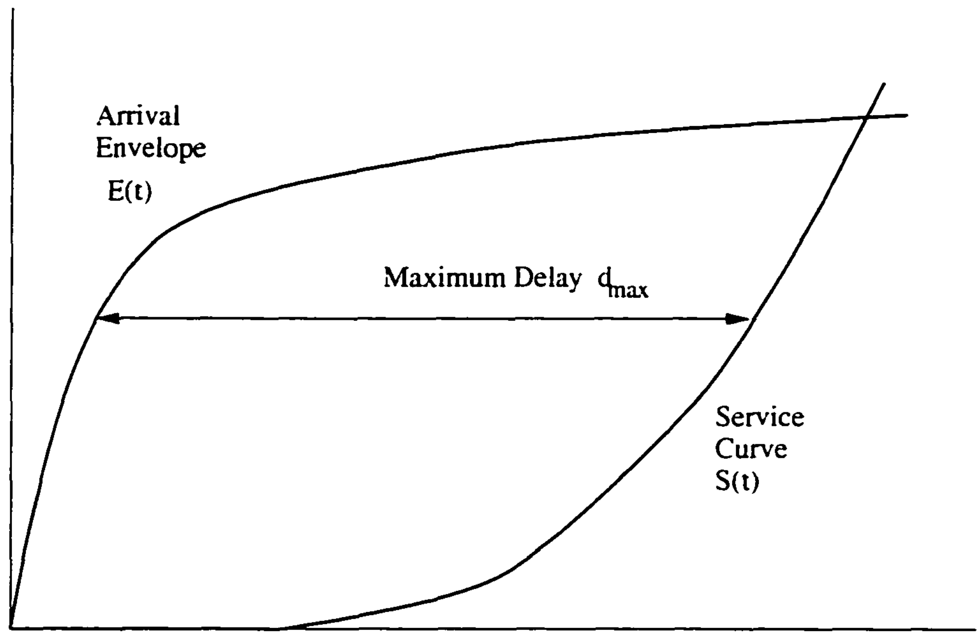


Figure 1.3: The worst case delay

Proposition 4 (Composition Rule [10]): Suppose a traffic stream passes through n service curve elements in series, where the i^{th} element has minimum service curve S_i and maximum service curve $\bar{S}_i$, $i = 1, 2, \ldots n$. Then the entire system is a service curve element with minimum and maximum service curves $S_1 * S_2 * \cdots * S_n$ and $\bar{S}_1 * \bar{S}_2 * \cdots * \bar{S}_n$, respectively.

1.2 Previous Results for Window Based Flow Control

In order to avail of excess bandwidth, a session needs feedback and should not be impeded by regulation within the network. We consider adaptive service where network resources are reserved to provide a minimum bandwidth guarantee while preventing buffer overflow. Window flow control is used as a mechanism for utilizing excess bandwidth in the network in addition to insuring no buffer overflow.

Recalling Figure 1.1, flowing through a few routers, we can first consider the single hop case. Thus, we have the two network element model guaranteeing

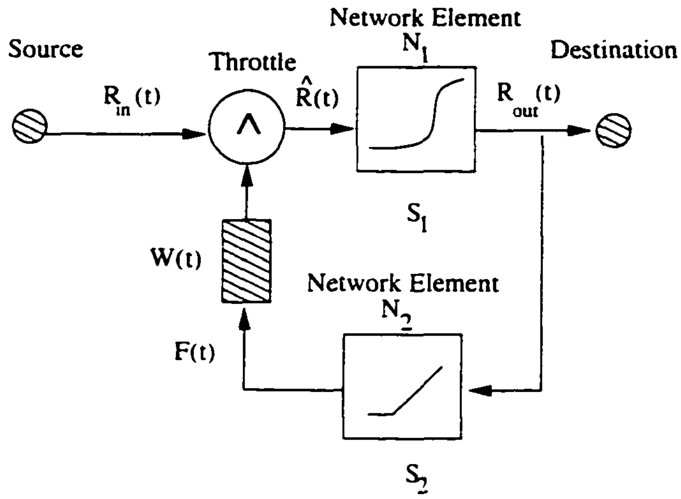


Figure 1.4: A single hop with window flow control

minimum service curves S_1 and S_2 respectively, as depicted in Figure 1.4. Window flow control is modeled through a network element called a *throttle*, which is controlled by a *throttle process*. In particular, a throttle receives traffic from an arrival process, say R_{in} and releases the traffic according to the departure process $\hat{R} = R_{in} \wedge (B+W)$, where $B+W$ is the throttle process which controls the throttle.

We shall assume that the window process is bounded below by positive constant w_{min}, i.e. we have

$$W(t) \geq w_{min}, \text{ for all } t.$$

In Figure 1.4, the output of the first network element is the destination of the session and is called the departure process R_{out}. The departure process R_{out} is also fed back to the throttle via network element N_2, corresponding to *acknowledgements*. The network element N_2 can be viewed as a model for delay, such as propagation delay in the return of acknowledgements to the throttle.

Theorem 5 (Closed Loop Minimum Service Curve [1]): Suppose we have arrival process R_{in} entering a throttle followed by a tandem of network elements N_1 and N_2. In addition, network element N_2 departure process F is fed back and added to the window process W resulting in throttle process $F + W$. If network ele-

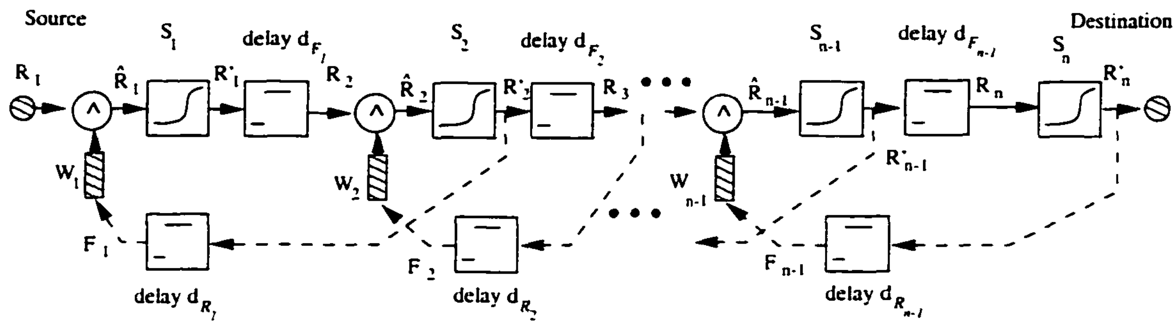


Figure 1.5: An n-hop session with hop-by-hop window flow control.

ments N_1 and N_2 guarantee minimum service curves S_1 and S_2, respectively, and $W(t) \geq w_{min}$ for all t, then

$$R_{out}(t) \geq R_{in} * S^*(t) \ .$$

where

$$S^* = S_1 * \wedge_{m=0}^{\infty}(G + w_{min})^{(m)} \ .$$

$G = S_1 * S_2$, and $(G + w_{min})^{(m)}$ is the m-fold convolution of $G + w_{min}$ with itself, e.g. $(G + w_{min})^{(1)} = G + w_{min}$, $(G + w_{min})^{(2)} = (G + w_{min}) * (G + w_{min})$, and so on, and we define $(G + w_{min})^{(0)} = \delta$.

Consider the system depicted in Figure 1.5, which models a network that employs hop-by-hop window flow control. In particular, at each hop a throttle is used, and the throttle is controlled by the departure process from the router at the next hop. The router at hop i is modeled by a service curve element with minimum service curve S_i. We assume that the departure process of the router at hop i encounters a fixed propagation delay d_{F_i} before reaching the throttle at hop $i + 1$. Similarly, acknowledgments generated by the departure process of the router at hop $i+1$ encounter a fixed propagation delay of d_{R_i} before reaching the throttle at hop i. The throttle at hop i uses the window process W_i, where we assume that $0 < w_i^{min} \leq W_i(t)$. Suppose $w_i^{min} \geq \sup_{t \in \mathbb{R}}\{G_i(t) - G_i * G_i(t - d_{R_i})\}$, where $G_i = S_i * S_{i+1} * \delta_{d_{F_i}}$ for all i. Under this condition, it can be shown that the end-to-end minimum service curve is $\delta_{d_F} * S_1 * S_2 * \cdots * S_n$, where $d_F = \sum_{i=1}^{n-1} d_{F_i}$

is the total forward propagation delay.

1.3 Previous work in Service Curve Scheduling

There has been significant progress in the analysis of scheduling algorithms for integrated service networks to support deterministic guaranteed service sessions. The use of a service curve[1] to study generalized processor sharing (GPS) also known as "fair queueing" [3], was first introduced by Parekh and Gallager [18, 19, 20]. The idea of using a service curve as a general characterization of a scheduling policy was proposed by Cruz [9], and refined in [10].

In general, given a minimum service curve S, it is possible to *synthesize* a scheduling algorithm so that a server guarantees the service curve S to a given traffic stream. This philosophy is taken in [21] [22] [23], and is closely related to the "Earliest Deadline First" (EDF) scheduling policies considered in [15] and [17].

The following Theorem from [11], states that the "so-called" SCED (Service Curve Earliest Deadline first) scheduling algorithm is capable of guaranteeing a minimum service curve for any session served by the server provided sufficient bandwidth is allocated.

Theorem 6 (Bandwidth allocation condition [11]): Consider a set of N sessions sharing a fixed rate server with a transmission capacity of c bits per second. If each session i has envelope E_i, and

$$\sum_{j=1}^{N} E_j * S_j(x) \leq cx, \text{ for all } x \geq 0,$$

then it is possible to schedule packets such that the server delivers minimum service curve $\hat{S}_i$ for arrival session i, where

$$\hat{S}_i(t) = [S_i(t - L/c) - L]^+ ,$$

and L is the maximum length in bits of a packet.

[1] The service curve presented by Parekh and Gallager is some what different in definition to this paper.

1.4 Discussion

As alluded to earlier in this chapter. Theorem 5 implies a minimum *bandwidth* guarantee for a session using window flow control to avail of excess bandwidth. However, in this mathematical framework. in order to obtain an upper bound on packet *delay* through the network. an envelope for the arrival process is required. The assumption of an envelope defeats the purpose of using window flow control to avail of excess bandwidth. since the envelope limits the maximum throughput of a session. Thus. with this framework, an adaptive session that avails of excess bandwidth must sacrifice a guarantee of bounded delay. It would therefore be desired to provide a bound on packet delay for a closed loop window flow control session without the assumption of an envelope for the arrival traffic. We present a framework for obtaining such a method in the following chapter.

1.5 Notes and Related Work

The results in Section 1.2. and collaboration with R. Agrawal and R. Rajan. has led to this dissertation work. The Theorem discussed in Section 1.2 is similar to the work of Hahne [16]. who was apparently the first to present rigorous mathematical bounds for window flow control. although Hahne required lower bounds on traffic. Similar results to Theorem 5 were obtained concurrently by Cruz & Okino [13], Chang [6]. and Agrawal & Rajan [2], although in all previous works. the window size was fixed constant.

In some related work [14]. we presented a slightly different model with joint scheduling and window flow control. where arriving packets at a server are first scheduled based on an *earliest deadline* algorithm and then eligibility of the packet to be served is based on the window flow control protocol. Although this method of coupling the flow control mechanism and the scheduling makes for an interesting academic problem. similar results are more easily obtained by decoupling the scheduling and the flow control mechanism [2], a technique apparently recognized by Zhang and Ferrari [24] for open loop rate based flow control.

Chapter 2

Adaptive Service Guarantees

In this chapter, we propose an "adaptive" service guarantee, present some performance bounds for a single network element, and then obtain performance bounds for closed loop window based flow control. We present a network element that is similar to the previously defined regulator, but this new network element allows excess traffic to be served. We conclude with a scheduling algorithm capable of delivering our proposed service guarantee.

2.1 Adaptive Sessions Network Calculus

In this section, we study the flow of traffic through network elements. We propose a service guarantee which is a function of both the arrival process and the departure process. We obtain bounds on delay and point out some relevance to the Generalized Processor Sharing(GPS) algorithm. We then present the composition rule as a method for obtaining end-to-end performance bounds.

2.1.1 Adaptive Service Guarantees

Definition 7 (Adaptive Service Guarantee): Given a network element with arrival process R_{in} and departure process R_{out}. Let $\tilde{S}$ be a causal process. We say that the network element adaptively guarantees $\tilde{S}$, if for all $s \le t$ there holds

$$R_{out}(t) \ge \{R_{out}(s) + \tilde{S}(t-s)\} \wedge \inf_{\tau : t \ge \tau \ge s} \{R_{in}(\tau) + \tilde{S}(t-\tau)\} \quad . \tag{2.1}$$

If (2.1) holds for all $s \le t$, then we write, as a shorthand notation,

$$R_{in} \rightarrow (\tilde{S}) \rightarrow R_{out} \quad .$$

Since R_{in} is a causal, non-decreasing function, for any $\epsilon > 0$, we have $R_{in}(-\epsilon) = 0$. Since $R_{out}(t) \le R_{in}(t)$ for any t, we have $R_{out}(-\epsilon) = 0$. Thus, for any $\epsilon > 0$, if we set $s = -\epsilon$ in (2.1) and since $\tilde{S}$ is causal, we have

$$\begin{aligned}
R_{out}(t) &\ge \{R_{out}(-\epsilon) + \tilde{S}(t+\epsilon)\} \wedge \inf_{\tau : t \ge \tau \ge -\epsilon} \{R_{in}(\tau) + \tilde{S}(t-\tau)\} \\
&= \{\tilde{S}(t+\epsilon)\} \wedge \inf_{\tau : t \ge \tau \ge -\epsilon} \{R_{in}(\tau) + \tilde{S}(t-\tau)\} \\
&= \inf_{\tau : \tau \le -\epsilon} \{R_{in}(\tau) + \tilde{S}(t-\tau)\} \wedge \inf_{\tau : t \ge \tau \ge -\epsilon} \{R_{in}(\tau) + \tilde{S}(t-\tau)\} \\
&= \inf_{\tau : \tau \le t} \{R_{in}(\tau) + \tilde{S}(t-\tau)\} \\
&= R_{in} * \tilde{S}(t) \ .
\end{aligned}$$

and so $R_{in} \rightarrow (\tilde{S}) \rightarrow R_{out}$ implies a minimum service curve $\tilde{S}$.

We now introduce a slightly more general adaptive service guarantee:

Definition 8 (Refined Adaptive Service Guarantee): Given a network element with arrival process R_{in} and departure process R_{out}. Let $\tilde{S}$ and S be causal processes such that $S \ge \tilde{S}$. We say that the network element adaptively guarantees $(S, \tilde{S})$, if for all $s \le t$ there holds

$$R_{out}(t) \ge \{R_{out}(s) + \tilde{S}(t-s)\} \wedge \inf_{\tau : t \ge \tau \ge s} \{R_{in}(\tau) + S(t-\tau)\} \quad . \tag{2.2}$$

In this context, S is called a *partial* service curve and $\tilde{S}$ is called an *absolute* service curve. If (2.2) holds for all $s \le t$, then we write, as a shorthand notation,

$$R_{in} \rightarrow (S, \tilde{S}) \rightarrow R_{out} \quad .$$

If we let $S = \tilde{S}$ in Definition 8 we get Definition 7. In general, since we assume that $S \ge \tilde{S}$, then $R_{in} \rightarrow (S, \tilde{S}) \rightarrow R_{out}$ implies $R_{in} \rightarrow (\tilde{S}) \rightarrow R_{out}$.

2.1.2 Performance Bounds

We begin this section by deriving a lower bound on the amount of traffic delivered over an interval and then show a closely related result of upper bound on virtual delay in terms of the backlog. The virtual delay bound is useful for adaptive applications since it does not require the arrival process to be constrained by an envelope.

Proposition 9: Suppose $R_{in} \rightarrow (\tilde{S}) \rightarrow R_{out}$, and the network element has a backlog at time $s \leq t$ such that $B(s) \geq \tilde{S}(t-s)$. Then $R_{out}(t) - R_{out}(s) \geq \tilde{S}(t-s)$.

Proof of Proposition 9: Fix $s \leq t$. It is sufficient to show that the second term on the right hand side of (2.1) is lower bounded by the first term on the right hand side when $B(s) \geq \tilde{S}(t-s)$. We have

$$\begin{aligned} \inf_{u:t \geq u \geq s} \{R_{in}(u) + \tilde{S}(t-u)\} &\geq R_{in}(s) \\ &\geq R_{out}(s) + \tilde{S}(t-s) \ . \end{aligned}$$

□

For all x, define the "pseudo inverse" of $\tilde{S}$, $\tilde{S}^{-1}$ according to

$$\tilde{S}^{-1}(x) = \inf\{y : y \geq 0, \tilde{S}(y) \geq x\} \ .$$

Using the right continuity of $\tilde{S}$, it follows that $\tilde{S}(\tilde{S}^{-1}(x)) \geq x$ for all x.

We now state a closely related proposition in terms of a virtual delay bound.

Proposition 10 (Delay bound from backlog): If $R_{in} \rightarrow (\tilde{S}) \rightarrow R_{out}$, then the virtual delay at time t, $d(t)$, is upper bounded according to $d(t) \leq \tilde{S}^{-1}(B(t))$.

Proof of Proposition 10: Fix t. It suffices to show that $R_{out}(t + \tilde{S}^{-1}(B(t))) \geq R_{in}(t)$. Since $t \leq t + \tilde{S}^{-1}(B(t)) =: \Delta(t)$, using (2.1) and the right continuity of $\tilde{S}$, we have

$$R_{out}(\Delta(t)) \geq \{R_{out}(t) + \tilde{S}(\Delta(t) - t)\} \wedge \inf_{\tau:\Delta(t) \geq \tau \geq t} \{R_{in}(\tau) + \tilde{S}(\Delta(t) - \tau)\}$$

$$
\begin{aligned}
&\geq \{R_{out}(t) + B(t)\} \wedge \inf_{\tau:\Delta(t)\geq\tau\geq t} \{R_{in}(\tau) + \tilde{S}(\Delta(t) - \tau)\} \\
&\geq \{R_{out}(t) + B(t)\} \wedge R_{in}(t) \\
&= R_{in}(t) \ .
\end{aligned}
$$

□

Note that the result above can be simplified to an upper bound on delay for all time given a fixed upper bound on the backlog. Thus, if $B(t) \leq B_{max}$ for all t, then $d(t) \leq d_{max}$ for all t, where $d_{max} = \tilde{S}^{-1}(B_{max})$.

2.1.3 Relevance to Generalized Processor Sharing Server

We now consider the traffic flow of a session served by a GPS server [18], [19], [20]. We assume there are N sessions, labeled $j = 1, 2, \cdots, N$, that are shared by the server, which has a capacity of c bits per second. The arrival and departure process for session j are denoted by $R_{in,j}$ and $R_{out,j}$, respectively. The weight assigned to session j is ϕ_j, and we assume that $\sum_{j=1}^{N} \phi_j = c$. By definition, under GPS, for any session i having positive backlog over any interval $(\tau, t]$, the amount of traffic served for session i over the interval is lower bounded by $\phi_i(t-\tau)$. Specifically, if $B_i(u) > 0$ for all $u \in (\tau, t]$, then

$$
R_{out,i}(t) - R_{out,i}(\tau) \geq \phi_i(t - \tau) \ . \tag{2.3}
$$

Let

$$
\mu_{\phi_i}(x) = \begin{cases} \phi_i x & , \text{for } x \geq 0 \\ 0 & \text{else} \end{cases} , \text{ for all } i = 1, 2, \cdots N.
$$

Proposition 11: Suppose that session i is served by a GPS server, i.e. for any interval $(\tau, t]$ where $B_i(u) > 0$ for $u \in (\tau, t]$, (2.3) holds. Then we have $R_{in,i} \rightarrow (\mu_{\phi_i}) \rightarrow R_{out,i}$.

Proof of Proposition 11: Fix $s \leq t$, choose session i, and let s^* be the last time that the buffer was empty for session i, i.e.

$$
s^* = \sup\{u : u \leq t, B_i(u) = 0\} \ .
$$

Suppose $s^* \leq s$. Then since $s \leq t$ and $B_i(u) > 0$ for $u \in (s, t]$, we have

$$\begin{aligned} R_{out,i}(t) - R_{out,i}(s) &\geq \phi_i(t-s) \\ &= \mu_{\phi_i}(t-s) \ . \end{aligned}$$

Now suppose that $s^* \in (s, t]$. Consider first, the case $B_i(s^*) = 0$. Then since $s^* \leq t$ and $s < s^*$, we have

$$\begin{aligned} R_{out,i}(t) &\geq R_{out,i}(s^*) + \phi_i(t-s^*) \\ &= R_{in,i}(s^*) + \phi_i(t-s^*) \\ &= R_{in,i}(s^*) + \mu_{\phi_i}(t-s^*) \\ &\geq \inf_{\tau : t \geq \tau \geq s} \{R_{in,i}(\tau) + \mu_{\phi_i}(t-\tau)\} \ . \end{aligned}$$

It remains to consider the case where the $B_i(s^*) \neq 0$. By the definition of s^*, if ϵ is an arbitrarily small positive number such that $s^* - \epsilon \geq s$, there exists a u^* such that $B_i(u^*) = 0$ and $s^* > u^* > s^* - \epsilon \geq s$. Since $R_{out,i}(s^*) \geq R_{out,i}(u^*) = R_{in,i}(u^*) \geq R_{in,i}(s^* - \epsilon)$, we have

$$\begin{aligned} R_{out,i}(t) &\geq R_{out,i}(s^*) + \phi_i(t-s^*) \\ &\geq R_{in,i}(s^*-\epsilon) + \phi_i(t-s^*) \\ &\geq \inf_{u : t \geq u \geq s} \{R_{in,i}(u) + \phi_i(t-u-\epsilon)\} \\ &= \inf_{u : t \geq u \geq s} \{R_{in,i}(u) + \mu_{\phi_i}(t-u)\} - \phi_i\epsilon \ . \end{aligned}$$

□

2.1.4 Fixed Propagation Delay Element

Proposition 12: Consider a fixed delay element with arrival process R_{in} and departure process R_{out} with delay $d > 0$, i.e. $R_{out}(t) = R_{in}(t-d)$ for all t. We have $R_{in} \rightarrow (\delta_d) \rightarrow R_{out}$.

Proof of Proposition 12: Fix $s \leq t$. If $t - s < d$, then

$$R_{out}(t) - R_{out}(s) \geq 0 = \delta_d(t-s) \ .$$

If $t - s \geq d > 0$, then since $R_{out}(t) = R_{in}(t-d) = R_{in} * \delta_d(t)$, we have

$$\begin{aligned}
R_{out}(t) &= R_{in} * \delta_d(t) \\
&= \inf_{\tau:\tau\in\mathbb{R}} \{R_{in}(\tau) + \delta_d(t-\tau)\} \\
&= \inf_{\tau:\tau\leq t} \{R_{in}(\tau) + \delta_d(t-\tau)\} \\
&= \inf_{\tau:\tau<s} \{R_{in}(\tau) + \delta_d(t-\tau)\} \wedge \inf_{\tau:t\geq\tau\geq s} \{R_{in}(\tau) + \delta_d(t-\tau)\} \\
&= \inf_{\tau:t\geq\tau\geq s} \{R_{in}(\tau) + \delta_d(t-\tau)\} .
\end{aligned}$$

□

2.1.5 Composition Rule

We consider a tandem series of n network elements.

Theorem 13 (Composition of Adaptive Service Guarantees): Suppose for $i = 1, \cdots, n$, we have

$$R_{i-1} \rightarrow (S_i, \tilde{S}_i) \rightarrow R_i .$$

Then

$$R_0 \rightarrow (S_1 * S_2 * \cdots * S_n, \tilde{G}) \rightarrow R_n \quad .$$

where $\tilde{G} = (\tilde{S}_1 * S_2 * \cdots * S_n) \wedge (\tilde{S}_2 * S_3 * \cdots * S_n) \wedge \cdots \wedge (\tilde{S}_{n-1} * S_n) \wedge \tilde{S}_n$. e.g. for $n = 2$ we have $\tilde{G} = (\tilde{S}_1 * S_2) \wedge \tilde{S}_2$. Moreover, if $S_i \geq \tilde{S}_i$ for all i, then this implies that $R_0 \rightarrow (S_1 * S_2 * \cdots * S_n, \tilde{S}_1 * \tilde{S}_2 * \cdots * \tilde{S}_n) \rightarrow R_n$.

Proof of Theorem 13: We prove the Theorem for the case $n = 2$. The general case can then be proved by induction on n. Fix $s \leq t$. Since $R_0 \rightarrow (S_1, \tilde{S}_1) \rightarrow R_1$ and $R_1 \rightarrow (S_2, \tilde{S}_2) \rightarrow R_2$, then

$$\begin{aligned}
R_2(t) &\geq \{R_2(s) + \tilde{S}_2(t-s)\} \wedge \inf_{\tau:t\geq\tau\geq s} \{R_1(\tau) + S_2(t-\tau)\} \\
&\geq \{R_2(s) + \tilde{S}_2(t-s)\} \wedge \inf_{\tau:t\geq\tau\geq s} \{\{R_1(s) + \tilde{S}_1(\tau-s)\} \wedge \inf_{u:\tau\geq u\geq s} \{R_0(u) \\
&\qquad + S_1(\tau-u)\} + S_2(t-\tau)\}
\end{aligned}$$

$$
\begin{aligned}
&= \{R_2(s) + \tilde{S}_2(t-s)\} \wedge \inf_{\tau: t \geq \tau \geq s} \{R_1(s) + \tilde{S}_1(\tau - s) + S_2(t-\tau)\} \\
&\quad \wedge \inf_{\tau: t \geq \tau \geq s} \inf_{u: \tau \geq u \geq s} \{R_0(u) + S_1(\tau - u) + S_2(t - \tau)\} \\
&\geq \{R_2(s) + \tilde{S}_2(t-s)\} \wedge \inf_{\tau: t \geq \tau \geq s} \{R_2(s) + \tilde{S}_1(\tau - s) + S_2(t-\tau)\} \\
&\quad \wedge \inf_{\tau: t \geq \tau \geq s} \inf_{u: \tau \geq u \geq s} \{R_0(u) + S_1(\tau - u) + S_2(t - \tau)\} \\
&\geq \{R_2(s) + ((\tilde{S}_1 * S_2) \wedge \tilde{S}_2)(t-s)\} \wedge \inf_{\tau: t \geq \tau \geq s} \inf_{u: \tau \geq u \geq s} \{R_0(u) \\
&\quad + S_1(\tau - u) + S_2(t - \tau)\} \\
&\geq \{R_2(s) + ((\tilde{S}_1 * S_2) \wedge \tilde{S}_2)(t-s)\} \wedge \inf_{\tau: t \geq \tau \geq s} \inf_{u: \tau \geq u \geq s} \{R_0(u) \\
&\quad + S_1 * S_2(t - u)\} \\
&= \{R_2(s) + ((\tilde{S}_1 * S_2) \wedge \tilde{S}_2)(t-s)\} \wedge \inf_{u: t \geq u \geq s} \{R_0(u) \\
&\quad + S_1 * S_2(t - u)\} \ . \qquad (2.4)
\end{aligned}
$$

Thus. $R_0 \rightarrow (S_1 * S_2, ((\tilde{S}_1 * S_2) \wedge \tilde{S}_2)) \rightarrow R_2$. It remains to show that this implies $R_0 \rightarrow (S_1 * S_2, (\tilde{S}_1 * \tilde{S}_2)) \rightarrow R_2$. Using (2.4), it is sufficient to show that $(\tilde{S}_1 * S_2) \wedge \tilde{S}_2 \geq \tilde{S}_1 * \tilde{S}_2$. Since $S_i \geq \tilde{S}_i$ for $i = 1, 2$. we have

$$
\begin{aligned}
(\tilde{S}_1 * S_2) \wedge \tilde{S}_2 &\geq (\tilde{S}_1 * \tilde{S}_2) \wedge \tilde{S}_2 \\
&= \tilde{S}_1 * \tilde{S}_2 \ .
\end{aligned}
$$

□

2.2 Adaptive Service Guarantee for networks with Feedback

In previous papers [13],[6], [2], the only method (within the mathematical framework) for obtaining bounds on delay for closed loop window based flow control protocols required the use of an access regulator as defined in Chapter 1. Unfortunately, using an access regulator in combination with window based flow control could result in a session under-utilizing its window based flow control link, thus defeating the purpose of using window flow control for adaptive sessions. In the previous section, Proposition 10 implies that we can obtain an upper bound

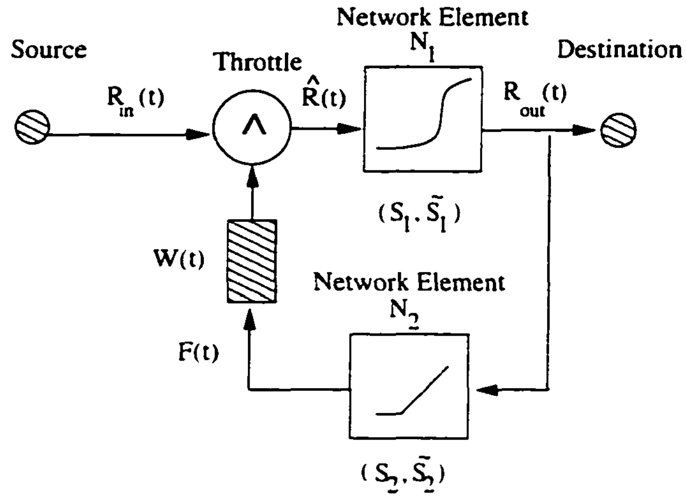


Figure 2.1: A single hop with window flow control

on virtual delay based on backlog but independent of an envelope for the arrival process. With the introduction of the adaptive service guarantee. we are motivated to revisit closed loop window based flow control in order to obtain virtual delay bounds without an explicit envelope constraint.

Consider the closed loop window based flow control model of network element N_1 and network element N_2 depicted in Figure 2.1. Window flow control is performed by a network element called a *throttle* where the throttle departure process is the pointwise minimum of the arrival process and the throttle control process. Specifically. let R_{in} be the arrival process to the throttle. $\hat{R}$ be the departure process from the throttle. and $F + W$ be the *throttle control process.* The throttle has an input queue for the arrival process and a server which is subject to window based flow control. enforcing the throttle to release traffic according to the minimum of the arrival process and the throttle control process $F + W$. i.e.

$$\hat{R}(t) \quad = \quad R_{in}(t) \wedge \{F(t) + W(t)\} \, ,$$

for all t. Clearly. the departure process $\hat{R}$ never exceeds the throttle control process and thus will buffer packets from the arrival process only as necessary to meet this constraint.

The throttle departure process $\hat{R}$ enters network element N_1 and departs having departure process R_{out}. In our model, the departure process R_{out} is our destination, but is also fed back to the throttle corresponding to acknowledgements. The departure process R_{out} enters network element N_2 and departs having departure process F which is added to the "process" W resulting in the throttle process $F + W$. We assume that $F + W$ is a process, i.e. it is non-decreasing. However, W is not a process, i.e. it is possible for W to be decreasing. We have not described the behavior of W and it is beyond the scope of this paper. However, it is reasonable to visualize an algorithm where W would decrease allowing other sessions to utilize needed buffer capacity while still insuring that $F + W$ is non-decreasing, and when the network is not congested, W could increase.

We now present a new adaptive service definition by conditioning the service guarantee over an interval.

Definition 14 (Adaptive Service Guarantee over an interval): Given a network element with arrival process R_{in} and departure process R_{out}. Let $\tilde{S}$ and S be causal processes such that $S \geq \tilde{S}$. We say a network element adaptively guarantees $(S, \tilde{S})$ over $[s^*, t^*]$, if for all $s, t \in [s^*, t^*]$ and $s \leq t$ we have

$$R_{out}(t) \geq \{R_{out}(s) + \tilde{S}(t - s)\} \wedge \inf_{u:t \geq u \geq s} \{R_{in}(u) + S(t - u)\} . \qquad (2.5)$$

If (2.5) holds for all $s, t \in [s^*, t^*]$ and $s \leq t$, then we write, as a shorthand notation,

$$R_{in} \rightarrow (S, \tilde{S})_{[s^*, t^*]} \rightarrow R_{out} \quad .$$

It is easy to show the following composition rule over the interval $[s^*, t^*]$ with a proof similar to Theorem 13.

Theorem 15 (Composition of Adaptive Service Guarantees over an interval): Suppose for $i = 1, \cdots, n$, we have

$$R_{i-1} \rightarrow (S_i, \tilde{S}_i)_{[s^*, t^*]} \rightarrow R_i \, .$$

Then

$$R_0 \rightarrow (S_1 * S_2 * \cdots * S_n, \tilde{G})_{[s^*, t^*]} \rightarrow R_n$$

where $\tilde{G} = (\tilde{S}_1 * S_2 * \cdots * S_n) \wedge (\tilde{S}_2 * S_3 * \cdots * S_n) \wedge \cdots \wedge (\tilde{S}_{n-1} * S_n) \wedge \tilde{S}_n$. Moreover, if $S_i \geq \tilde{S}_i$ for all i, then this implies that $R_0 \rightarrow (S_1 * S_2 * \cdots * S_n, \tilde{S}_1 * \tilde{S}_2 * \cdots * \tilde{S}_n)_{[s^*,t^*]} \rightarrow R_n$.

We now recall the two network element model, Figure 2.1. Before stating the Closed Loop Adaptive Service Guarantee Theorem, we have the following assumption and notation.

Assume that the window process $W(u)$ for all $u \in [s^*, t^*]$ is upper and lower bounded such that

$$w_{max} \geq W(u) \geq w_{min} \geq 0 . \tag{2.6}$$

Let

$$\tilde{S}_T = \left[\wedge_{n=0}^{\infty}[\tilde{S} * S^{(n)} + nw_{min}] + w_{min} - w_{max}\right]^+ .$$

$$S_T = \wedge_{n=0}^{\infty}[S^{(n)} + nw_{min}] .$$

where $[x]^+ = \min\{x, 0\}$, $S = S_1 * S_2$, $\tilde{S} = \tilde{S}_1 \wedge (S_1 * \tilde{S}_2)$, $S^{(0)} = \delta$, and for any function g, $g^{(n)} = \underbrace{g * g * \cdots g}_{n}$ is the n^{th} fold convolution.

Theorem 16 (Closed Loop Adaptive Service Guarantee): Suppose $\hat{R} = R_{in} \wedge (F + W)$, (2.6) holds over interval $[s^*, t^*]$, $\hat{R} \rightarrow (S_1, \tilde{S}_1)_{[s^*,t^*]} \rightarrow R_{out}$ and $R_{out} \rightarrow (S_2, \tilde{S}_2)_{[s^*,t^*]} \rightarrow F$. Then

$$R_{in} \rightarrow (S_1 * S_T, \tilde{S}_T)_{[s^*,t^*]} \rightarrow R_{out} .$$

To prove this theorem, we use the following useful lemma.

Lemma 17: Suppose $\hat{R} = R_{in} \wedge (F + W)$, (2.6) holds, $\hat{R} \rightarrow (S_1, \tilde{S}_1)_{[s^*,t^*]} \rightarrow R_{out}$ and $R_{out} \rightarrow (S_2, \tilde{S}_2)_{[s^*,t^*]} \rightarrow F$. Given $s, t \in [s^*, t^*]$ such that $s \leq t$, there holds

$$\begin{aligned} R_{out}(t) \geq & \{R_{out}(s) + \tilde{S}_T(t-s)\} \wedge \inf_{\tau: t \geq \tau \geq s}\{R_{in}(\tau) + S_1 * S_T(t-\tau)\} \\ & \wedge \inf_{\hat{t}: t \geq \hat{t} \geq s}\{R_{out}(\hat{t}) + S^{(m)}(t-\hat{t}) + mw_{min}\} . \end{aligned} \tag{2.7}$$

Proof of Lemma 17: We use induction to prove this lemma. First, we show (2.7) for $n = 1$. Choose $s, t \in [s^*, t^*]$ arbitrarily such that $s \le t$. Using $\hat{R} \rightarrow (S_1, \tilde{S}_1) \rightarrow R_{out}$, $R_{out} \rightarrow (S_2, \tilde{S}_2) \rightarrow F$, $W \ge w_{min}$ from (2.6), and finally that $\tilde{S} = \tilde{S}_1 \wedge (S_1 * \tilde{S}_2)$, we have

$$
\begin{aligned}
R_{out}(t) &\ge \{R_{out}(s) + \tilde{S}_1(t-s)\} \wedge \inf_{\tau: t \ge \tau \ge s} \{\hat{R}(\tau) + S_1(t-\tau)\} \\
&= \{R_{out}(s) + \tilde{S}_1(t-s)\} \wedge \inf_{\tau: t \ge \tau \ge s} \{R_{in}(\tau) + S_1(t-\tau)\} \\
&\quad \wedge \inf_{\tau: t \ge \tau \ge s} \{F(\tau) + W(\tau) + S_1(t-\tau)\} \\
&\ge \{R_{out}(s) + \tilde{S}_1(t-s)\} \wedge \inf_{\tau: t \ge \tau \ge s} \{R_{in}(\tau) + S_1(t-\tau)\} \\
&\quad \wedge \{F(s) + S_1 * \tilde{S}_2(t-s) + w_{min}\} \wedge \inf_{\tau: t \ge \tau \ge s} \{R_{out}(\tau) \\
&\quad + S(t-\tau) + w_{min}\} \\
&= \{R_{out}(s) + \tilde{S}_1(t-s)\} \wedge \inf_{\tau: t \ge \tau \ge s} \{R_{in}(\tau) + S_1(t-\tau)\} \\
&\quad \wedge \{F(s) + W(s) - W(s) + S_1 * \tilde{S}_2(t-s) + w_{min}\} \\
&\quad \wedge \inf_{u: t \ge u \ge s} \{R_{out}(u) + S(t-u) + w_{min}\} \\
&\ge \{R_{out}(s) + \tilde{S}_1(t-s)\} \wedge \inf_{\tau: t \ge \tau \ge s} \{R_{in}(\tau) + S_1(t-\tau)\} \\
&\quad \wedge \{R_{out}(s) + [S_1 * \tilde{S}_2(t-s) + w_{min} - w_{max}]\} \\
&\quad \wedge \inf_{u: t \ge u \ge s} \{R_{out}(u) + S(t-u) + w_{min}\} \\
&\ge \{R_{out}(s) + [\tilde{S}(t-s) + w_{min} - w_{max}]\} \wedge \inf_{\tau: t \ge \tau \ge s} \{R_{in}(\tau) \\
&\quad + S_1(t-\tau)\} \wedge \inf_{u: t \ge u \ge s} \{R_{out}(u) + S(t-u) + w_{min}\} . \qquad (2.8)
\end{aligned}
$$

Since $R_{out}(t) \ge R_{out}(s)$, recognize that

$$
\begin{aligned}
R_{out}(t) &\ge \{R_{out}(s)\} \wedge \inf_{\tau: t \ge \tau \ge s} \{R_{in}(\tau) + S_1(t-\tau)\} \wedge \inf_{u: t \ge u \ge s} \{R_{out}(u) \\
&\quad + S(t-u) + w_{min}\} . \qquad (2.9)
\end{aligned}
$$

Using (2.8) , (2.9), and the property $(a \wedge c) \vee (b \wedge c) = (a \vee b) \wedge c$, we have

$$
\begin{aligned}
R_{out}(t) &\ge \Big\{R_{out}(s) \vee \{R_{out}(s) + [\tilde{S}(t-s) + w_{min} - w_{max}]\}\Big\} \wedge \inf_{\tau: t \ge \tau \ge s} \{R_{in}(\tau) \\
&\quad + S_1(t-\tau)\} \wedge \inf_{u: t \ge u \ge s} \{R_{out}(u) + S(t-u) + w_{min}\} \\
&= \{R_{out}(s) + [\tilde{S}(t-s) + w_{min} - w_{max}]^+\} \wedge \inf_{\tau: t \ge \tau \ge s} \{R_{in}(\tau)
\end{aligned}
$$

$$\begin{aligned}
&+ S_1(t-\tau)\} \wedge \inf_{u:t\geq u\geq s}\{R_{out}(u) + S(t-u) + w_{min}\} \\
\geq\ & \{R_{out}(s) + \tilde{S}_{\mathcal{T}}(t-s)\} \wedge \inf_{\tau:t\geq\tau\geq s}\{R_{in}(\tau) + S_1(t-\tau)\} \\
& \wedge \inf_{u:t\geq u\geq s}\{R_{out}(u) + S(t-u) + w_{min}\} \qquad (2.10) \\
\geq\ & \{R_{out}(s) + \tilde{S}_{\mathcal{T}}(t-s)\} \wedge \inf_{\tau:t\geq\tau\geq s}\{R_{in}(\tau) + S_1 * S_{\mathcal{T}}(t-\tau)\} \\
& \wedge \inf_{u:t\geq u\geq s}\{R_{out}(u) + S(t-u) + w_{min}\}\ . \qquad (2.11)
\end{aligned}$$

which is (2.7) for $m = 1$.

Before proceeding with the induction, we prove that for any $n \geq 1$, we have

$$S^{(n)} * \tilde{S}_{\mathcal{T}} + nw_{min} \ \geq\ \tilde{S}_{\mathcal{T}}\ . \qquad (2.12)$$

Using the definition of $\tilde{S}_{\mathcal{T}}$, the property that $a * (b \vee c) \geq (a * b) \vee (a * c)$, and $S \geq \tilde{S}$, we have

$$\begin{aligned}
S^{(n)} * \tilde{S}_{\mathcal{T}} + nw_{min} \ &\geq\ \max\{\wedge_{j=0}^{\infty}\tilde{S} * S^{(j+n)} + jw_{min}] + w_{min} - w_{max}, 0\} \\
&\quad + nw_{min} \\
&=\ \{\wedge_{j=0}^{\infty}\tilde{S} * S^{(j+n)} + (j+n)w_{min}] + w_{min} - w_{max}\} \\
&\quad \vee \{nw_{min}\} \\
&\geq\ \tilde{S}_{\mathcal{T}}\ .
\end{aligned}$$

We now assume that (2.7) holds for $m = n$ and show that this implies that it holds for $m = n+1$. Suppose

$$R_{out}(t) < \{R_{out}(s) + \tilde{S}_{\mathcal{T}}(t-s)\} \wedge \inf_{\tau:t\geq\tau\geq s}\{R_{in}(\tau) + S_1 * S_{\mathcal{T}}(t-\tau)\}\ . \qquad (2.13)$$

Then, using (2.13), the induction hypothesis, (2.10) with $t = \hat{t}$, and finally (2.12), we have

$$\begin{aligned}
R_{out}(t) \ &\geq\ \inf_{\hat{t}:t\geq\hat{t}\geq s}\{R_{out}(\hat{t}) + S^{(n)}(t-\hat{t}) + nw_{min}\} \\
&\geq\ \inf_{\hat{t}:t\geq\hat{t}\geq s}\{\{R_{out}(s) + \tilde{S}_{\mathcal{T}}(\hat{t}-s)\} \wedge \inf_{\tau:\hat{t}\geq\tau\geq s}\{R_{in}(\tau) + S_1(\hat{t}-\tau)\} \\
&\quad \wedge \inf_{u:\hat{t}\geq u\geq s}\{R_{out}(u) + S(\hat{t}-u) + w_{min}\} + S^{(n)}(t-\hat{t}) + nw_{min}\}
\end{aligned}$$

$$\begin{aligned}
&\geq \{R_{out}(s) + S^{(n)} * \tilde{S}_T(t-s) + nw_{min}\} \wedge \inf_{\tau:t\geq\tau\geq s}\{R_{in}(\tau) \\
&\quad + S_1 * S^{(n)} * (t-\tau) + nw_{min}\} \wedge \inf_{u:t\geq u\geq s}\{R_{out}(u) \\
&\quad + S^{(n+1)}(t-u) + (n+1)w_{min}\} \\
&\geq \{R_{out}(s) + \tilde{S}_T(t-s)\} \wedge \inf_{\tau:t\geq\tau\geq s}\{R_{in}(\tau) + S_1 * S_T(t-\tau)\} \\
&\quad \wedge \inf_{u:t\geq u\geq s}\{R_{out}(u) + S^{(n+1)}(t-u) + (n+1)w_{min}\} . \qquad (2.14)
\end{aligned}$$

and so we have proven (2.7) for $m = n+1$ under the condition (2.13).

Recognize that (2.7) holds trivially if (2.13) does not hold. and we are done.

□

Proof of Theorem 16: Fix $s, t \in [s^*, t^*]$ such that $s \leq t$. We know that $R_{out}(t)$ is bounded. i.e. there exists an integer $\hat{m}$ such that

$$R_{out}(t) \quad < \quad \hat{m}w_{min} . \qquad (2.15)$$

Choose $m \geq \hat{m}$ in Lemma 17. Then (2.15) implies that the last term on the right hand side of (2.7) is greater than R_{out} and hence can be omitted.

□

Let

$$k_{min} = \sup_{t\in\mathbb{R}}\{\tilde{S}(t) - \tilde{S} * S(t)\} \vee \sup_{t\in\mathbb{R}}\{S_1(t) - S_1 * S(t)\} . \qquad (2.16)$$

Corollary 18: Suppose $\hat{R}\rightarrow(S_1, \tilde{S}_1)_{[s^*,t^*]}\rightarrow R_{out}$ and $R_{out}\rightarrow(S_2, \tilde{S}_2)_{[s^*,t^*]}\rightarrow F$. hold. If W is *constant* over the interval $[s^*, t^*]$, and is greater than or equal to k_{min}. then

$$R_{in}\rightarrow(S_1, \tilde{S})_{[s^*,t^*]}\rightarrow R_{out} \quad .$$

Proof of Corollary 18: Fix s, t such that $s, t \in [s^*, t^*]$ and $s \leq t$. Using Theorem 16 where $w_{max} = w_{min} = k_{min}$ over the interval $[s^*, t^*]$, it remains to show that $\tilde{S}_T = \tilde{S}$ and $S_1 * S_T = S_1$.

Recognize that the definition of k_{min} implies that $\tilde{S} * (S + k_{min}) \geq \tilde{S}$ and $S_1 * (S + k_{min}) \geq S_1$. Now, using the definition of $\tilde{S}_T$. we have

$$\tilde{S}_T \quad = \quad (\wedge_{n=0}^{\infty}[\tilde{S} * S^{(n)} + nw_{min}] + w_{min} - w_{max})^{+}$$

$$
\begin{aligned}
&= \wedge_{n=0}^{\infty}[\tilde{S} * S^{(n)} + nw_{min}] \\
&= \tilde{S} \wedge [\tilde{S} * (S + k_{min})] \wedge [\tilde{S} * (S + k_{min})^{(2)}] \wedge \cdots \\
&= \tilde{S} \, .
\end{aligned}
$$

Similarly, we have

$$
\begin{aligned}
S_1 * S_T &= S_1 * (\wedge_{n=0}^{\infty}[S^{(n)} + nw_{min}]) \\
&= S_1 * (\wedge_{n=0}^{\infty}[S^{(n)} + nk_{min}]) \\
&= \wedge_{n=0}^{\infty}[S_1 * S^{(n)} + nk_{min}] \\
&= S_1 \wedge [S_1 * (S + k_{min})] \wedge [S_1 * (S + k_{min})^{(2)}] \wedge \cdots \\
&= S_1 \, .
\end{aligned}
$$

□

Remark 19: Theorem 16 in conjunction with Proposition 10 imply that it is possible to avail of excess bandwidth (using closed loop flow control) and obtain adaptive service guarantees **provided** network elements with adaptive guarantees exist.

Proposition 11 demonstrated existence of network elements as described in Remark 19. We later show in Corollary 28 in Section 2.4 that a network element scheduling packets in an earliest deadline fashion also holds for the above remark.

2.2.1 Hop-by-Hop Window Flow Control

Consider the system depicted in Figure 2.2, which models a network that employs hop-by-hop window flow control. In particular, at each hop a throttle is used, and the throttle is controlled by the departure process from the router at the next hop. The router at hop i is modeled by

$$\hat{R}_i \rightarrow (G_i, \tilde{G}_i) \rightarrow R_i' \; .$$

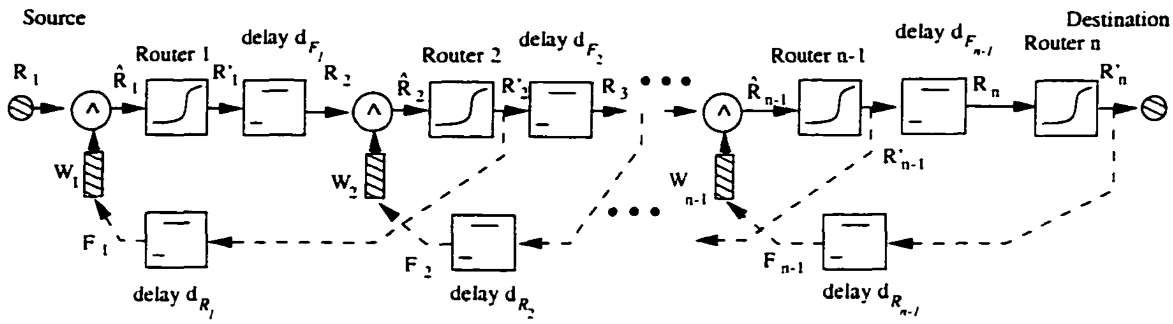


Figure 2.2: An n-hop session with hop-by-hop window flow control.

where $\hat{R}_i$ is the departure process of throttle i and R'_i is the departure process of the network element modeling the router at hop i. For simplicity of exposition[1], we assume that the departure process of the router at hop i encounters a fixed propagation delay d_{F_i} before reaching the throttle at hop $i+1$. i.e.

$$R'_i \rightarrow (\delta_{d_{F_i}}, \delta_{d_{F_i}}) \rightarrow R_{i+1} \quad .$$

Similarly, acknowledgements generated by the departure process of the router at hop $i+1$ encounter a fixed propagation delay of d_{R_i} before reaching the throttle at hop i. i.e.

$$R'_{i+1} \rightarrow (\delta_{d_{R_i}}, \delta_{d_{R_i}}) \rightarrow F_i \quad .$$

The throttle at hop i uses the window process W_i, where we assume that $W_i(t) \geq k_{min,i}$ which we define below. We analyze the system by exploiting the robustness of service curve definitions, and again by lumping network elements using Theorem 13.

Note that the throttle at hop i is contained in the cycle of elements which determine the throttle process at hop $i-1$. Thus, in order to determine an adaptive service guarantee for the throttle at hop $i-1$, we must first determine an adaptive service guarantee for the throttle at hop i. Since there is no throttle at the last hop, we may analyze the throttle at hop $n-1$ using the same method we used to analyze the simple cycle in Figure 2.1. Once an adaptive service guarantee for the

[1] More elaborate models can easily be handled.

throttle at hop $n-1$ is determined. we can incorporate that into the analysis of the throttle at hop $n-2$. By continuing in this manner. we may determine the service curves for each of the throttles. and then use Theorem 13 to determine the end-to-end adaptive service guarantee.

Suppose we let $S_i = G_i * G_{i+1} * \delta_{d_{F_i}}$ and $\tilde{S}_i = (\tilde{G}_i * \delta_{d_{F_i}}) \wedge (\tilde{S}_{i+1} * G_i * \delta_{d_{F_i}} * \delta_{d_{R_i}})$. for all $i \leq n-1$ and $\tilde{S}_n = \tilde{G}_n$. Similar to equation (2.16). define

$$k_{min,i} = \sup_{t \in \mathbb{R}} \{\tilde{S}_i(t) - \tilde{S}_i * S_i(t - d_{R_i})\} \vee \sup_{t \in \mathbb{R}} \{S_i(t) - S_i * S_i(t - d_{R_i})\} .$$

By following the procedure outlined above. if for all $i < n$ and over the interval $[s^*. t^*]$. all window sizes are constant. and $k_{min,i} \leq W_i(t)$. it follows that the end-to-end partial service curve is identical to that obtained with no throttles present. The absolute service curve can be obtained in the same manner but with precise application of the composition rule. Following the previous close loop example. we have

$$R_{n-1} \rightarrow (S_{n-1}. \tilde{S}_{n-1}) \rightarrow R'_n .$$

For all $i < n$. let

$$\tilde{H}_i = \{\tilde{G}_i \wedge (\tilde{G}_{i+1} * G_i * \delta_{d_{F_i}+d_{R_i}}) \wedge \cdots \wedge (G_n * G_{n-1} * \cdots * G_i * \delta_{\sum_{j=i}^{k}(d_{F_j}+d_{R_j})})\} .$$

Along the lines of the above description computation based on the previous hop. we get for $i < n-1$. we have

$$R_i \rightarrow (G_i, \tilde{H}_i)_{[s^*. t^*]} \rightarrow R'_i .$$

For the previous hop. we have

$$R_{i-1} \rightarrow (G_{i-1} * \delta_{d_{F_{i-1}}}. \tilde{H}_{i-1})_{[s^*. t^*]} \rightarrow R_i .$$

Note that the absolute service curve at the $i-1$-th hop does not incur an additional delay $d_{F_{i-1}}$.

By applying the composition rule for $n-1$ network elements in tandem. and using $G_i \geq \tilde{G}_i$ for all i, we have

$$R_1 \rightarrow (G_i * G_2 * \cdots G_{n-1} * G_n * \delta_d, \tilde{G}_1 * \tilde{G}_2 * \tilde{G}_3 * \cdots * \tilde{G}_{n-1} * \tilde{G}_n * \delta_{\dot{d}})_{[s^*,t^*]} \rightarrow R'_n \ ,$$

where $d = \sum_{j=1}^{n-1} d_{F_j}$ and $\dot{d} = \sum_{j=1}^{n-1}(d_{F_j} + d_{R_j})$.

Example 20 (Minimum Service Guarantee versus Adaptive Service Guarantee): Consider Figure 2.3, a four hop adaptive session using hop-by-hop window flow control where the arrival process is R_1. Each router is a latency rate server such that, for each hop i, the router has latency d_i and serves packets at rate μ. The output of each router $i < 4$ incurs a propagation delay d_F before reaching the $i+1$ router(corresponding to the input of throttle $i+1$). Similarly, acknowledgements are fed back from each router $1 < i < 4$ to the $i-1$ router throttle but first incur a propagation delay d_R. The fourth router is not subject to window flow control and has departure process R'_4. Using the minimum service curve definition from Chapter 1, for each router i, we have minimum service curve $S_{min,i}(x) = \mu(x - d_i)^+$ for all x. Using the adaptive service guarantee, for each router i, the absolute service curve $\tilde{S}_i$ is equal to the partial service curve S_i. Moreover, for each router, the partial service curve is equal to the minimum service curve, i.e. for each i, we have $\tilde{S}_i(x) = S_i(x) = S_{min,i}(x) = \mu(x - d_i)^+$ for all x. If the window sizes are constant over the interval $[s^*, t^*]$, and for all i, $W_i(t) \geq k_{min,i}$, where

$$k_{min,i} = \mu(d_i + d_{i+1} + d_F + d_R) \ ,$$

then we have an end-to-end minimum service curve of S_{min}, where

$$S_{min}(x) = \mu(x - d_1 - d_2 - d_3 - d_4 - 3d_F)^+ \ .$$

and

$$R_1 \rightarrow (S, \tilde{S})_{[s^*,t^*]} \rightarrow R'_4 \ ,$$

where the end-to-end partial service curve is

$$S(x) = S_{min}(x) = \mu(x - d_1 - d_2 - d_3 - d_4 - 3d_F)^+ \ ,$$

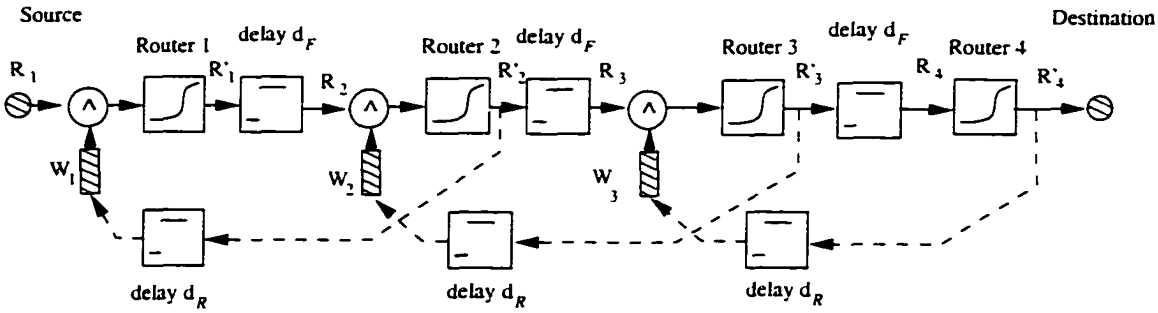


Figure 2.3: A four hop adaptive session with hop-by-hop window flow control

and the end-to-end absolute service curve is

$$\tilde{S}(x) = \mu(x - d_1 - d_2 - d_3 - d_4 - 3d_F - 3d_R)^+ \ .$$

Finally, suppose that the source adapts to the feedback, such that the backlog in the first throttle does not exceed b_T^{max}. Since the composition of all follow-on network elements in the tandem is equal to the end-to-end closed loop bandwidth delay product, we have maximum system backlog equal to b_{max}, where $b_{max} = \mu(d_1 + d_2 + d_3 + d_4 + 3d_F + 3d_R) + b_T^{max}$. Using the "pseudo inverse" of $\tilde{S}$, we have

$$d_{max} = \tilde{S}^{-1}(b_{max}) = 2(d_1 + d_2 + d_3 + d_4 + 3d_F + 3d_R) + b_T^{max}/\mu \ .$$

2.3 The Elastic Regulator

In this section, we present a new network element called the *elastic regulator*. Roughly speaking, the elastic regulator is a network element that allows traffic above an envelope constraint to be served. We desire a network element that, in conjunction with a service curve element can provide for adaptive service guarantees. Before describing the elastic regulator, we present the following proposition and discuss the implication of the result.

Proposition 21 (Adaptive Service Guarantee from Service Curve Guarantees): Consider a network element that guarantees minimum service curve G and maximum service curve $\bar{G}$. The arrival process is known to have the traffic envelope

E ($E(0) = 0$). Then the network element guarantees the absolute service curve $\tilde{G}$ and the partial service curve G, where

$$\tilde{G}(x) = \inf_{y:y\geq 0}\{\left[G(x+y) - E * \tilde{G}(y)\right]^{+}\} \quad \text{for all } x.$$

Proof of Proposition 21: Fix $s \leq t$. If $\tilde{G}(t-s) = 0$, then (2.2) holds trivially since R_{out} is non-decreasing. Assume then that $\tilde{G}(t-s) > 0$. It suffices to show that if

$$R_{out}(t) \quad < \quad \inf_{\tau:\tau\geq s}\{R_{in}(\tau) + G(t-\tau)\} \; . \tag{2.17}$$

then

$$R_{out}(t) \geq R_{out}(s) + \tilde{G}(t-s) \; .$$

Note that for all x such that $\tilde{G}(x) > 0$ we have $\tilde{G}(x) = \inf_{y:y\geq 0}\{G(x+y) - E * \tilde{G}(y)\}$ and hence

$$\tilde{G}(x) + E * \tilde{G}(y) \quad \leq \quad G(x+y) \qquad \text{if } \tilde{G}(x) > 0 \text{ and } y \geq 0. \tag{2.18}$$

Using the minimum service curve guarantee, followed by (2.17) , and then (2.18), we have

$$\begin{aligned}
R_{out}(t) &\geq R_{in} * G(t) \\
&= \inf_{\tau:\tau\leq s}\{R_{in}(\tau) + G(t-\tau)\} \\
&\geq \inf_{\tau:\tau\leq s}\{R_{in}(\tau) + E * \tilde{G}(s-\tau) + \tilde{G}(t-s)\} \\
&\geq R_{in} * (E * \tilde{G})(s) + \tilde{G}(t-s) \\
&= (R_{in} * E) * \tilde{G}(s) + \tilde{G}(t-s) \\
&\geq R_{in} * \tilde{G}(s) + \tilde{G}(t-s) \\
&\geq R_{out}(s) + \tilde{G}(t-s) \; .
\end{aligned}$$

□

By the definition of the impulse, it is clear that any network element has a maximum service curve δ, i.e. $R_{out} \leq R_{in} = R_{in} * \delta$. Using the above proposition,

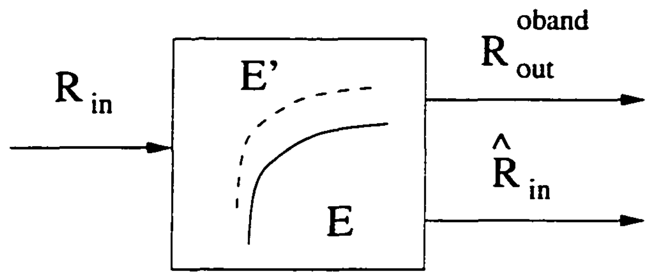


Figure 2.4: An Elastic Regulator

we lower bound absolute service curve $\tilde{G}$ by $\tilde{G}'$, where

$$\tilde{G}'(x) = \inf_{y:y\geq 0}\{[G(x+y) - E(y)]^+\}, \text{ for all } x .$$

The above result implies that the service curve element with minimum service curve G, and arrival process R_{in} with envelope E, is $R_{in} \rightarrow (G, \tilde{G}') \rightarrow R_{out}$.

The typical method of bounding the arrival process with an envelope E is by regulating the input with a regulator as defined in Chapter 1. The disadvantage of *always* bounding the arrival process R_{in} with an envelope E is that it prohibits additional traffic above the envelope from being delivered when the link is not congested. Ideally, we would like a network element which functions like a regulator when the link is congested, but will allow traffic above the envelope to be served when there is excess bandwidth.

2.3.1 The Elastic Regulator Element

Consider Figure 2.4 depicting a network element called the *elastic regulator*. The arrival process R_{in} feeds the elastic regulator with departure process $\hat{R}_{in}$ (corresponding to "conformant" traffic) and departure process R^{oband}_{out} (corresponding to "out-of-band" traffic). The elastic regulator backlog at time t is

$$B^r(t) = R_{in}(t) - R^{oband}_{out}(t) - \hat{R}_{in}(t) .$$

Packets may depart out-of-band at time t if $B^r(t) > 0$. We do not otherwise specify when packets may depart out-of-band, and in fact this is determined by a downstream network element. The elastic regulator is defined by specifying when conforming packets may depart.

Toward this end, we define the conformant departure process to be

$$\hat{R}_{in}(t) = \sum_{k=0}^{\infty} L^k u(t - \tau^k) ,$$

where k is the packet index, L^k is the number of bits of the k^{th} conforming packet that departs, and we define $\tau_0 = 0$ and $L^0 = 0$. We assume that all packets have at most L_{max} bits. We also assume $0 \leq \tau^k \leq \tau^{k+1}$ for all k, such that $\hat{R}_{in}$ is causal.

We define an envelope E' as follows:

$$E'(x) = \begin{cases} E(x) + L_{max} & , \text{for } x \geq 0 \\ 0 & \text{elsewhere} \end{cases} .$$

where E is a causal, sub-additive process. In fact, we shall also assume that $E(t)$ is *left*-continuous for $t > 0$, without loss of generality. We will refer to E as the "target" envelope since the conformant departure process $\hat{R}_{in}$ in general will not have envelope E but only have envelope E'.

We describe when conforming packets may depart in terms of a sequence $\{\tau'_k\}$, defined below. We call τ'_k the *eligibility time* of the k^{th} conforming departure from the elastic regulator. The eligibility time is based on *previous* conformant departures from the elastic regulator and is recalculated each time *any* conformant departure takes place for the session. The sequence $\{\tau'_k\}$ is defined as follows. Let $\tau'_1 := 0$, $\tau_0 := 0$, and then τ'_{k+1} is computed recursively from $(\tau_1, \tau_2, \cdots, \tau_k)$ at time τ_k. In particular

$$\tau'_{k+1} = \inf\{s : s \geq \tau_k \text{ and } \min_{j:1 \leq j \leq k} [\sum_{l=0}^{j-1} L^l + E(s - \tau_j)] \geq \sum_{l=0}^{k} L^l\} .$$

By definition, the k^{th} conformant departure can be no earlier than τ'_k. The k^{th} conformant departure is at time τ'_k, unless $\lim_{t \to \tau_k^-}(B_i^r(t)) = 0$ and there is no arrival in the elastic regulator at time τ'_k. In this case, the time of the k^{th} conformant departure is the time of the first arrival to the elastic regulator after time τ'_k. The elastic regulator is capable of delivering packets above the envelope constraints. The elastic regulator will allow a packet to depart as an out-of-band packet provided the elastic regulator has positive backlog at the time a *request* is made to send the packet. In general, the request to "violate" the envelope constraint will come from a follow-on server delivering out-of-band traffic. Note that these out-of-band packets are not considered part of the departure process $\hat{R}_{in}$.

Lemma 22: An elastic regulator with target envelope E has the following properties:

(i) $\hat{R}_{in} \leq \hat{R}_{in} * E'$

(ii) For $B^r(w) > 0$, there exists $x < w$ such that $\hat{R}_{in}(w) - \hat{R}_{in}(x) > E(w - x)$.

(iii) Given u' and any $\epsilon > 0$, there exists $w < u' + \epsilon$ such that

$$\hat{R}_{in} * E(u') \geq \hat{R}_{in}(w) + E(u' - w) - \epsilon \ . \tag{2.19}$$

and $B^r(w^*) = 0$ for some $w^* \in (w - \epsilon, w + \epsilon)$. Moreover, $\hat{R}_{in}(w) = \hat{R}_{in}(w^*)$.

Proof of Lemma 22: We first prove (i). We need to show that $\hat{R}_{in}(t) \leq \hat{R}_{in} * E'(t)$ for all t. It is sufficient to check at $t = \tau_k$ for all k. In general,

$$\hat{R}_{in} * E'(x) = \min\{\hat{R}_{in}(x), \min_{j:\tau_j < x}[\sum_{l=0}^{j-1} L^l + E'(x - \tau_j)]\} \ .$$

Consider first the case where $\hat{R}_{in} * E'(\tau_{k+1}) = \hat{R}_{in}(\tau_{k+1})$. Then trivially, we have

$$\hat{R}_{in}(\tau_{k+1}) \leq \hat{R}_{in} * E'(\tau_{k+1}) \ .$$

Now consider the case where $\hat{R}_{in} * E'(\tau_{k+1}) = \min_{j:1\leq j\leq k}[\sum_{l=0}^{j-1} L^l + E'(\tau_{k+1} - \tau_j)]$. Since $\tau_{k+1} \geq \tau'_{k+1}$, we have

$$\begin{aligned} \hat{R}_{in}(\tau_{k+1}) &= L^{k+1} + \sum_{l=0}^{k} L^l \\ &\leq \min_{j:1\leq j\leq k}[\sum_{l=0}^{j-1} L^l + E(\tau'_{k+1} - \tau_j)] + L^{k+1} \end{aligned}$$

$$\begin{aligned}
&\le \min_{j:1\le j\le k}[\sum_{l=0}^{j-1} L^l + E(\tau_{k+1}-\tau_j)] + L^{k+1} \\
&= \min_{j:1\le j\le k}[\sum_{l=0}^{j-1} L^l + E'(\tau_{k+1}-\tau_j)] \\
&= \hat{R}_{in} * E'(\tau_{k+1}) ,
\end{aligned}$$

and so we have (i).

We now prove (ii). If $B^r(w) > 0$. then define $k^* = \max\{k : \tau_k \le w\}$. We claim that $\tau'_{k^*+1} > w$. To see this. suppose to the contrary that $\tau'_{k^*+1} \le w$. Since $B^r(w) > 0$. it follows that for the definition of the elastic regulator. $\tau_{k^*+1} \le w$. which is a contradiction. and so $\tau'_{k^*+1} > w$. Therefore, by the definition of τ'_{k^*+1}. we have

$$\min_{j:1\le j\le k^*}[\sum_{l=0}^{j-1} L^l + E(w-\tau_j)] < \sum_{l=0}^{k^*} L^l . \tag{2.20}$$

where we recognize that the right hand side of (2.20) is equal to $\hat{R}_{in}(w)$. Also. recognize that there exists $j^* \le k^*$ such that

$$\min_{j:1\le j\le k^*}[\sum_{l=0}^{j-1} L^l + E(w-\tau_j)] = \sum_{l=0}^{j^*-1} L^l + E(w-\tau_{j^*}) .$$

and so. using (2.20), we have

$$\begin{aligned}
\hat{R}_{in}(w) &> \min_{j:1\le j\le k^*}[\sum_{l=0}^{j-1} L^l + E(w-\tau_j)] \\
&= \hat{R}_{in}(\tau_{j^*-1}) + E(w-\tau_{j^*}) .
\end{aligned}$$

Since E is right continuous. there exists $\epsilon > 0$ such that

$$\hat{R}_{in}(w) > \hat{R}_{in}(\tau_{j^*-1}) + E(w-(\tau_{j^*}-\epsilon)) .$$

Since $\hat{R}_{in}(\tau_{j^*}-\epsilon) \le \hat{R}_{in}(\tau_{j^*-1})$. we have

$$\hat{R}_{in}(w) > \hat{R}_{in}(\tau_{j^*}-\epsilon) + E(w-(\tau_{j^*}-\epsilon)) .$$

Since $\tau_{j^*}-\epsilon < \tau_{j^*} \le \tau_{k^*} \le w$, we have $x := \tau_{j^*}-\epsilon < w$, and we have (ii).

We now prove (iii). Note that

$$\hat{R}_{in} * E(u') = \min\{\hat{R}_{in}(u'), \min_{k:\tau_k<u'}[\sum_{l=0}^{k-1} L^l + E(u' - \tau_k)]\} .$$

Consider first the case where $\hat{R}_{in} * E(u') = \hat{R}_{in}(u')$. In this case, we may clearly choose $w \in (u', u' + \epsilon)$ such that (2.19) is true and $\hat{R}_{in}(u') = \hat{R}_{in}(w)$, since $\hat{R}_{in}$ is right continuous and E is causal. We set $w^* = u'$ in this case, and note that $w^* = u' \in (w - \epsilon, w + \epsilon)$. We also have $\hat{R}_{in}(w^*) = \hat{R}_{in}(u') = \hat{R}_{in}(w)$. It remains to show that $B^r(w^*) = 0$ in this case. Suppose to the contrary that $B^r(w^*) > 0$. Using Lemma 22 (ii) which we proved above, this implies that there exists an $x < w$ such that

$$\hat{R}_{in}(w) - \hat{R}_{in}(x) > E(w - x) .$$

Thus,

$$\begin{aligned}
\hat{R}_{in} * E(w^*) &= \hat{R}_{in} * E(u') \\
&= \hat{R}_{in}(u') \\
&= \hat{R}_{in}(w^*) \\
&> \hat{R}_{in}(x) + E(w^* - x) , \qquad (2.21)
\end{aligned}$$

which contradicts the definition of $\hat{R}_{in} * E(u')$.

Now consider the case where

$$\hat{R}_{in} * E(u') = \min_{k:\tau_k<u'}[\sum_{l=0}^{k-1} L^l + E(u' - \tau_k)] .$$

Recognize that there exists k^* such that

$$\hat{R}_{in} * E(u') = \sum_{l=0}^{k^*-1} L^l + E(u' - \tau_{k^*}) , \qquad (2.22)$$

and k^* is the smallest integer satisfying (2.22).

Note that $k^* \geq 1$, and $\tau_{k^*-1} < \tau_{k^*}$.

Using (2.22) and the right continuity of E, we may choose w such that $w \in (\tau_{k^*-1}, \tau_{k^*})$ and $w \in (\tau_{k^*} - \epsilon, \tau_{k^*})$ such that (2.19) is true.

We prove $B^r(w^*) = 0$ and $\hat{R}_{in}(w^*) = \hat{R}_{in}(w)$ for some $w^* \in (w^*-\epsilon, w^*+\epsilon)$.

Suppose first that $k^* = 1$. Recall that the first arrival is at time τ_1. Thus for any $w^* \in (w, \tau_1)$ we have $B^r(w^*) = 0$ and $\hat{R}_{in}(w^*) = \hat{R}_{in}(w)$. Also, by construction we have $(w, \tau_1) \subset (w - \epsilon, w + \epsilon)$ and so $w^* \in (w - \epsilon, w + \epsilon)$.

We now prove that for $k^* > 1$, there exists $w^* \in (w, \tau_{k^*})$ such that $B^r(w^*) = 0$ and $\hat{R}_{in}(w^*) = \hat{R}_{in}(w)$. Since $(w, \tau_{k^*}) \subset (w - \epsilon, w + \epsilon)$, this implies that $w^* \in (w - \epsilon, w + \epsilon)$.

Suppose to the contrary that $B^r(v) > 0$ for all $v \in (w, \tau_{k^*})$. In this case, we must have $\tau'_{k^*} = \tau_{k^*} > \tau'_{k^*-1}$. To see this, suppose to the contrary that $\tau'_{k^*} < \tau_{k^*}$. This implies that k^* could have departed from the elastic regulator earlier, and so $\tau'_{k^*} = \tau_{k^*}$. From the definition of eligibility time of the k^*-th departing packet, we have

$$\tau'_{k^*} = \inf\{s : s \geq \tau_{k^*-1} \text{ and } \min_{j:1\leq j\leq k^*-1} [\sum_{l=0}^{j-1} L^l + E(s - \tau_j)] \geq \sum_{l=0}^{k^*-1} L^l\} .$$

Then, for all $u \in (\tau_{k^*-1}, \tau_{k^*})$, we must have

$$\min_{j:1\leq j\leq k^*-1} [\sum_{l=0}^{j-1} L^l + E(u - \tau_j)] < \sum_{l=0}^{k^*-1} L^l .$$

Using the left continuity of E for $x > 0$, we have $\lim_{u\to\tau_{k^*}^-} E(u - \tau_j) = E(\tau_{k^*} - \tau_j)$, and so

$$\min_{j:1\leq j\leq k^*-1} [\sum_{l=0}^{j-1} L^l + E(\tau_{k^*} - \tau_j)] \leq \sum_{l=0}^{k^*-1} L^l .$$

Since there exists some $j^* \leq k^* - 1$, such that

$$\min_{j:1\leq j\leq k^*-1} [\sum_{l=0}^{j-1} L^l + E(\tau_{k^*} - \tau_j)] = \sum_{l=0}^{j^*-1} L^l + E(\tau_{k^*} - \tau_{j^*}) ,$$

we then have

$$\sum_{l=0}^{j^*-1} L^l + E(\tau_{k^*} - \tau_{j^*}) \quad \leq \quad \sum_{l=0}^{k^*-1} L^l . \tag{2.23}$$

Using (2.22), then (2.23) and the sub-additivity of E, we have

$$\hat{R}_{in} * E(u') \quad = \quad \sum_{l=0}^{k^*-1} L^l + E(u' - \tau_{k^*})$$

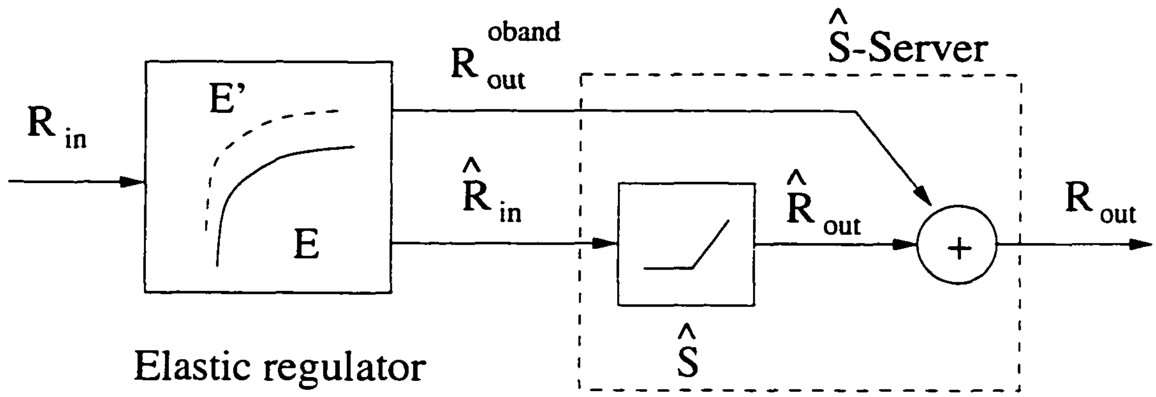


Figure 2.5: An Elastic Regulator with $\hat{S}$-Server

$$
\begin{aligned}
&\geq \sum_{l=0}^{j^*-1} L^l + E(\tau_{k^*} - \tau_{j^*}) + E(u' - \tau_{k^*}) \\
&\geq \sum_{l=0}^{j^*-1} L^l + E * E(u' - \tau_{j^*}) \\
&\geq \sum_{l=0}^{j^*-1} L^l + E(u' - \tau_{j^*}) \; .
\end{aligned}
$$

Since $j^* < k^*$, this either contradicts the definition of k^* or contradicts the definition of $\hat{R}_{in} * E(u')$. Therefore, there exists $w^* \in (w - \epsilon, w + \epsilon) \subset (\tau_{k^*-1}, \tau_{k^*})$ such that $B^r(w^*) = 0$. Since $(w - \epsilon, w + \epsilon) \subset (\tau_{k^*-1}, \tau_{k^*})$, no conformant departures occur in the interval $(w - \epsilon, w + \epsilon)$, and so $\hat{R}_{in}(w) = \hat{R}_{in}(w^*)$.

□

2.3.2 The Elastic Regulator with $\hat{S}$-Server

Consider the elastic regulator used in conjunction with a service curve element and a summing element as depicted in Figure 2.5. The conformant traffic $\hat{R}_{in}$ is the arrival process to a service curve element (with minimum service curve $\hat{S}$) and departure process $\hat{R}_{out}$. The out-of-band traffic R^{oband}_{out} bypasses the service curve element and is summed with departure process $\hat{R}_{out}$ resulting in total departure process $R_{out} = \hat{R}_{out} + R^{oband}_{out}$. The service curve element backlog at time t is

$$B^s(t) = \hat{R}_{in}(t) - \hat{R}_{out}(t) \; .$$

We define $\hat{S}$-server as a service curve element with minimum service curve $\hat{S}$ in tandem with the summing element. Note that the only buffering in the $\hat{S}$-server is in the service curve element. We assume that the $\hat{S}$-server has sufficient capacity to deliver minimum service curve $\hat{S}$. Furthermore, the $\hat{S}$-server may *request* that additional packets be sent from the elastic regulator violating the elastic regulator's envelope constraints, i.e. an out-of-band packet *can* be served at time t, if $B^r(t) > 0$.

Let

$$\tilde{S}(x) = \inf_{y:y\geq 0}\{\left[\hat{S}(x+y) - E'(y)\right]^+\} \quad \text{for all } x.$$

Theorem 23: Suppose a traffic session with arrival process R_{in} passes through an elastic regulator with target envelope E followed by a $\hat{S}$-server with departure process R_{out}. If $E \geq \hat{S} \oslash \hat{S}$, then the tandem system is a service curve element with minimum service curve $\hat{S}$, and $R_{in} \rightarrow (\hat{S}, \tilde{S}) \rightarrow R_{out}$.

Proof of Theorem 23: Fix s, t. For any ϵ, we have

$$\hat{R}_{out}(t+\epsilon) \geq \hat{R}_{in} * \hat{S}(t+\epsilon) \ . \tag{2.24}$$

Since $\hat{R}_{in}(t+\epsilon) \geq \hat{R}_{out}(t+\epsilon)$, this implies that $\hat{R}_{out}(t+\epsilon) \geq \inf_{\tau:\tau\leq t+\epsilon}\{\hat{R}_{in}(\tau) + \hat{S}(t+\epsilon-\tau)$. We will first prove that $R_{out} \geq R_{in} * \hat{S}$. Since $\hat{R}_{in} \geq \hat{R}_{in} * E$ and $E * \hat{S} \geq \hat{S}$, using (2.24), for any $\epsilon > 0$, there exists $u' \leq t+\epsilon$; also $w < u'+\epsilon \leq t+\epsilon$ using Lemma 22 (iii) such that

$$\begin{aligned}
\hat{R}_{out}(t+2\epsilon) &\geq \hat{R}_{out}(t+\epsilon) \\
&\geq \hat{R}_{in}(u') + \hat{S}(t+\epsilon-u') - \epsilon \\
&\geq \hat{R}_{in} * E(u') + \hat{S}(t+\epsilon-u') - \epsilon \\
&\geq \hat{R}_{in}(w) + E(u'-w) + \hat{S}(t+\epsilon-u') - 2\epsilon \\
&\geq \hat{R}_{in}(w) + E * \hat{S}(t+\epsilon-w) - 2\epsilon \\
&\geq \hat{R}_{in}(w) + \hat{S}(t+\epsilon-w) - 2\epsilon \ ,
\end{aligned} \tag{2.25}$$

and $B^r(w^*) = 0$ for some $w^* \in (w-\epsilon, w+\epsilon) \subset (w-\epsilon, t+2\epsilon)$. Recall that $\hat{R}_{in} * E(u') = \min\{\hat{R}_{in}(u'), \sum_{l=0}^{k^*-1} L^l + E(u'-\tau_{k^*})\}$ for some k^*. Then using Lemma

22 (iii). for $u' \le t+\epsilon$, there exists a $w^* \in [w-\epsilon, t+2\epsilon]$ such that $B^r(w^*) = 0$ and $\hat{R}_{in}(w) = \hat{R}_{in}(w^*)$.

Using (2.25), $w - \epsilon \le w^* \le t + 2\epsilon$ and $B^r(w^*) = 0$, we have

$$\begin{aligned}
R_{out}(t+2\epsilon) &= \hat{R}_{out}(t+2\epsilon) + R_{out}^{oband}(t+2\epsilon) \\
&\ge \hat{R}_{in}(w) + \hat{S}(t+\epsilon-w) - 2\epsilon + R_{out}^{oband}(t+2\epsilon) \\
&\ge \hat{R}_{in}(w) + \hat{S}(t+\epsilon-w) - 2\epsilon + R_{out}^{oband}(w^*) \\
&= \hat{R}_{in}(w) + \hat{S}(t+\epsilon-w) - 2\epsilon + R_{in}(w^*) - \hat{R}_{in}(w^*) \\
&= R_{in}(w^*) + \hat{S}(t+\epsilon-w) - 2\epsilon \\
&\ge R_{in}(w-\epsilon) + \hat{S}(t+\epsilon-w) - 2\epsilon \qquad (2.26) \\
&\ge R_{in} * \hat{S}(t) - 2\epsilon\,. \qquad (2.27)
\end{aligned}$$

Since ϵ is chosen arbitrarily, for a fixed ϵ', we have $R_{out}(t+\epsilon') \ge R_{in} * \hat{S}(t)$. By the right continuity of R_{out}, we have $R_{out} \ge R_{in} * \hat{S}$.

We now prove $R_{in} \to (\hat{S}, \tilde{S}) \to R_{out}$. Suppose first that $s \le w - \epsilon$. Using (2.26), we have

$$\begin{aligned}
R_{out}(t+2\epsilon) &\ge R_{in}(w-\epsilon) + \hat{S}(t+\epsilon-w) - 2\epsilon \\
&= \inf_{\tau: t \ge \tau \ge s} \{R_{in}(\tau) + \hat{S}(t-\tau)\} - 2\epsilon\,. \qquad (2.28)
\end{aligned}$$

Now suppose $s > w - \epsilon$ and $\tilde{S}(t-s) > 0$. Using (2.25), $s \le t$, then the definition of $\tilde{S}$ for $\tilde{S}(t-s) > 0$, followed by $\hat{R}_{out}(s) \le \hat{R}_{in}(s) \le \hat{R}_{in} * E'(s)$, we have

$$\begin{aligned}
R_{out}(t+2\epsilon) &= \hat{R}_{out}(t+2\epsilon) + R_{out}^{oband}(t+2\epsilon) \\
&\ge \hat{R}_{in}(w) + \hat{S}(t+\epsilon-w) - 2\epsilon + R_{out}^{oband}(t+2\epsilon) \\
&\ge \hat{R}_{in}(w) + \hat{S}(t-w) - 2\epsilon + R_{out}^{oband}(s) \\
&\ge \hat{R}_{in}(w) + E'(s-w) + \tilde{S}(t-s) - 2\epsilon + R_{out}^{oband}(s) \\
&\ge \hat{R}_{in} * E'(s) + \tilde{S}(t-s) - 2\epsilon + R_{out}^{oband}(s) \\
&\ge \hat{R}_{in}(s) + \tilde{S}(t-s) - 2\epsilon + R_{out}^{oband}(s) \\
&\ge \hat{R}_{out}(s) + \tilde{S}(t-s) - 2\epsilon + R_{out}^{oband}(s) \\
&= R_{out}(s) + \tilde{S}(t-s) - 2\epsilon\,. \qquad (2.29)
\end{aligned}$$

Recognize the trivial case, for $s \le t$ when $\tilde{S}(t-s) = 0$, we have

$$R_{out}(t + 2\epsilon) \ge R_{out}(s) + 0 = R_{out}(s) + \tilde{S}(t-s) \ . \tag{2.30}$$

Thus, from (2.28), (2.29), and (2.30), for a fixed $\epsilon' > 0$ we have

$$R_{out}(t + \epsilon') \ \ge \ \{R_{out}(s) + \tilde{S}(t-s)\} \wedge \inf_{\tau : t \ge \tau \ge s} \{R_{in}(\tau) + \hat{S}(t-\tau)\} - 2\epsilon \ . \tag{2.31}$$

Since ϵ' is chosen arbitrarily and R_{out} is right continuous, we have $R_{in} \rightarrow (\hat{S}, \tilde{S}) \rightarrow R_{out}$ and we are done.

□

2.4 Scheduling

In general, it is possible to *synthesize* a scheduling algorithm such that the server guarantees a service curve for a given traffic session. [12], [21], [22], [23]. In this section we consider a scheduling policy based on the so-called "SCED" scheduling policy proposed by Sariowan [21] [22], reported in [10], and later adapted to continuous time in [11]. We modify the SCED scheduling algorithm by allowing the server to go on vacations. This is only a slight modification to the model described in [11] and so the proof of the main theorem is similar in style to [11]. Since we allow the server to go on vacations, we appropriately call the scheduling algorithm, SCED - with vacations.

2.4.1 SCED - with vacations

Consider a network element with N traffic sessions where entering and exiting traffic are described by $\hat{R}_{in,i}$ and $\hat{R}_{out,i}$ for $i = 1, 2, \cdots, N$. For simplicity, we assume that the server is a fixed-rate server with capacity c bits/second. We assume a packet cannot begin service until it has completely arrived at the server. Thus, we assume packet arrivals occur instantaneously. Specifically, we will assume that the arrival processes are of the form,

$$\hat{R}_{in,i}(t) = \sum_{l=0}^{\infty} L_i^l u(t - \tau_i^k) \ ,$$

where

$$u(t-\tau_i^k) = \begin{cases} 0 & , \text{if } t < \tau_i^k \\ 1 & , \text{if } t \geq \tau_i^k \end{cases} .$$

the k^{th} packet of session i contains L_i^k bits and arrives at time τ_i^k, and $L_i^0 = 0$ and $\tau_i^0 = 0$ for all k and i.

We will assume that all packets for session i have at most $L_{max,i}$ bits, and furthermore, all packets in general have at most L_{max} bits, i.e. we have $L_i^k \leq L_{max,i} \leq L_{max}$ for all i and k. We shall assume that $0 \leq \tau_i^k \leq \tau_i^{k+1}$ for all k and i so that $\hat{R}_{in,i}$ is causal for all i.

We define the scheduler backlog for each session i at time t as

$$B_i^s(t) = \hat{R}_{in,i}(t) - \hat{R}_{out,i}(t) .$$

We first consider the effect of packetizing the server arrival session in order to obtain tighter worst case delay bounds.

Given the server arrival process $\hat{R}_{in,i}$, we define the process P_i to be

$$P_i(x) = \begin{cases} 0 & , \text{if } x < L_i^1 \\ \sum_{k=0}^{j} L_i^k & , \text{if } \sum_{k=0}^{j} L_i^k \leq x < \sum_{k=0}^{j+1} L_i^k \end{cases} .$$

Recognize that $P_i(\hat{R}_{in,i}(t)) = \hat{R}_{in,i}(t)$ for all t.

Suppose we wish to guarantee the minimum service curve S_j for session j, i.e. we want

$$\hat{R}_{out,j}(t) \geq \hat{R}_{in,j} * S_j(t), \text{ for all } t.$$

Thus, packets from session j arriving at time t need to meet deadline $d_j(\hat{R}_{in,j}(t))$, where we define $d_j(\gamma)$ for all $\gamma \geq 0$ as

$$d_j(\gamma) = \inf\{\Delta : \Delta \geq 0 \text{ and } \hat{R}_{in,j} * S_j(\Delta) \geq \gamma\} . \tag{2.32}$$

The scheduling algorithm in which deadlines are assigned to arriving packets based on a service curve is as follows: Packets are served in an earliest deadline fashion. A packet from session j which arrives at time $t = \tau_j^k$ is assigned deadline

$\Delta_j(\hat{R}_{in,j}(t))$. The server will always be busy serving a packet provided there is positive backlog. In addition, the server will serve packets non-preemptively in an *earliest deadline first* order. Since preemption is not allowed, this implies that packets departing the server do not necessarily have non-decreasing deadlines.

Finally, a server is allowed to go on vacation. A vacation can occur only when the server queue is empty, i.e. the server is allowed to go on vacation at time s if $\sum_{j=1}^{N} B_j^s(s) = 0$. The maximum period that the server is allowed to go on vacation is L_{max}/c seconds.

We first demonstrate the property that the deadline of a packet that arrives at time t can be calculated without knowledge of $\hat{R}_{in,i}(s)$ for $s > t$. We define

$$\hat{d}_i(t) = \inf\{s : s \geq t \text{ and } \inf_{\tau:\tau \leq t}\{\hat{R}_{in,i}(\tau) + S_i(s-\tau)\} \geq \hat{R}_{in,i}(t)\} . \tag{2.33}$$

Recognize that allowing the server to go on vacations does not affect the calculation of the deadlines, and thus, the following Lemma and proof are directly from Cruz [11].

Lemma 24 (Causality of deadlines [11]): If there is an arrival at time t from session i, i.e. $t = \tau_i^k$ for some packet k from session i, then

$$\hat{d}_i(t) = d_i(R_{in,i}(t)) .$$

Proof of Lemma 24 : See Appendix A.

□

Theorem 25: Suppose arrival process $\hat{R}_{in,i}$ has envelope E_i'.

$$\sum_{i=1}^{N} E_i' * S_i(x) \quad \leq \quad cx \text{ for all } x \geq 0, \tag{2.34}$$

and the server is allowed to go on vacations only when the backlog in the server is zero, and for a maximum of L_{max}/c seconds. Then no packet misses its deadline by more than L_{max}/c seconds. Furthermore, for each i, the system delivers a minimum service curve of $\hat{S}_i$ to the i^{th} arrival session, where

$$\hat{S}_i(t) = [S_i(t - L_{max}/c) - L_{max,i}]^+ .$$

We use the following useful lemmas to prove the theorem. Permitting the server to go on vacations is independent of the total number of bits with deadlines and so we use the following lemmas from Cruz [11].

Lemma 26 (Total traffic with deadlines $\leq t$ [11]): The total amount of traffic(in bits) from session i that has deadlines less than or equal to t is equal to $P_i(\hat{R}_{in,i} * S_i(t))$.

Proof of Lemma 26: See Appendix A.

□

Lemma 27 (Traffic in interval with deadlines $\leq t$ [11]): Suppose $T_i(s,t)$ is the total amount of traffic(in bits) from session i that arrived after time s and has deadlines less than or equal to t. We have $T_i(s,t) = [P_i(\hat{R}_{in,i} * S_i(t)) - \hat{R}_{in,i}(s)]^+$.

Proof of Lemma 27: See Appendix A.

□

Proof of Theorem 25: We will show that if any packet misses its deadline by more than L_{max}/c seconds, then (2.34) does not hold. Suppose at time t, a packet p^* departs that has deadline $t - L_{max}/c - \epsilon_d$, where $\epsilon_d > 0$. Suppose that this packet p^* begins service at time $t - \alpha$. Note that $\alpha \leq L_{max}/c$.

Define τ' as the last time no later than $t - \alpha$ that the server had a backlog of zero, i.e. $\tau' = \sup\{\tau : \tau \leq t - \alpha \text{ and } \sum_{j=1}^{N} B_j^s(\tau) = 0\}$. Suppose first that during the interval $[\tau', t - \alpha)$, no packet from any session with deadline later than $t - L_{max}/c - \epsilon_d$ began service, and furthermore that no vacation began in the interval. Note that $\tau' \leq t - L_{max}/c - \epsilon_d$, since p^* has arrived by time $t - L_{max}/c - \epsilon_d$, but does not leave until time t. Recognize that the server is busy throughout the interval $[\tau', t]$. If $\sum_{j=1}^{N} B_j^s(\tau') = 0$, then define $\tau^* = \tau'$. Otherwise, if $\sum_{j=1}^{N} B_j^s(\tau') > 0$, then choose $\hat{\tau}$ such that $\hat{\tau} \in \{s : \tau' - L_{max}/c \leq s < \tau' \text{ , } \sum_{j=1}^{N} B_j^s(v) = 0 \ \forall \ v \in [s, \tau')\}$ and define $\tau^* = \hat{\tau}$.

Let A be the set of sessions which received a non-zero amount of service in the interval $[\tau', t]$. It follows that all traffic served in the interval $[\tau', t]$ must have arrived after time τ^* and have deadlines at most $t - L_{max}/c - \epsilon_d$. Thus, from Lemma 27 we have

$$\begin{aligned}
c(t-\tau') &\leq \sum_{i\in A} T_i(\tau^*, t - L_{max}/c - \epsilon_d) \\
&= \sum_{i\in A} P_i(\hat{R}_i * S_i(t - L_{max}/c - \epsilon_d)) - \hat{R}_i(\tau^*) \\
&\leq \sum_{i\in A} \hat{R}_i * S_i(t - L_{max}/c - \epsilon_d) - \hat{R}_i(\tau^*) \\
&\leq \sum_{i\in A} \hat{R}_i * E'_i * S_i(t - L_{max}/c - \epsilon_d) - \hat{R}_i(\tau^*) \\
&\leq \sum_{i\in A} E'_i * S_i(t - L_{max}/c - \epsilon_d - \tau^*) \\
&\leq \sum_{i=1}^{N} E'_i * S_i(t - L_{max}/c - \epsilon_d - \tau^*) \ . \qquad (2.35)
\end{aligned}$$

Since $\tau' \leq \tau^* + L_{max}/c$ and $\epsilon_d > 0$, this shows that (2.34) does not hold for this case.

Now, consider the case where a vacation occurred or some packet with deadline later than $t - L_{max}/c - \epsilon$ begins service in the interval $[\tau', t-\alpha)$. Let s^* be the last occurrence of either the last time a vacation begins in the interval $[\tau', t-\alpha)$, or the last time a packet with deadline later than $t - L_{max}/c - \epsilon_d$ begins service in the interval $[\tau', t-\alpha)$.

Recognize that $s^* \leq t - L_{max}/c - \epsilon_d$. To see this, recall that p^* is queued in the system at time $t - L_{max}/c - \epsilon_d$, but does not begin service until $t - \alpha$. Thus if $s^* > t - L_{max}/c - \epsilon_d$, it would mean that packets are not served earliest deadline first or that a vacation occurs while a packet is queued in the scheduler.

Let u^* be the departure time of the packet that begins service at time s^*. Thus, the system is busy throughout the interval $(u^*, t]$ serving only packets with deadlines at most $t - L_{max}/c - \epsilon_d$. Let A^* be the set of sessions which received a non-zero amount of service in the interval $(u^*, t]$. We claim that for all $j \in A^*$, we have $B^s_j(s^*) = 0$. If a vacation begins at time s^*, then since the server will only begin a vacation when the backlog of the scheduler is zero, we have $B^s_j(s^*) = 0$. Otherwise, a packet with deadline greater than $t - L_{max}/c - \epsilon_d$ begins service at time s^*. If $B_j^s(s^*) > 0$ in this case, that would imply that all packets from session j queued at time s^* have deadlines greater than $t - L_{max}/c - \epsilon_d$. Else, another packet from session j would begin service at time s^* instead of the packet that actually began service. Hence, $B_j^s(s^*) > 0$ implies that session j does not receive service in the interval $(u^*, t]$, and so $j \notin A^*$. Therefore, it follows that all traffic served in the interval $(u^*, t]$ arrived after time s^*, and has deadlines no greater than $t - L_{max}/c - \epsilon_d$. Using Lemma 27, we have

$$\begin{aligned}
c(t-u^*) &\leq \sum_{i \in A^*} T_i(s^*, t - L_{max}/c - \epsilon_d) \\
&= \sum_{i \in A^*} P_i(\hat{R}_{in,i} * S_i(t - L_{max}/c - \epsilon_d)) - \hat{R}_{in,i}(s^*) \\
&\leq \sum_{i \in A^*} \hat{R}_{in,i} * S_i(t - L_{max}/c - \epsilon_d) - \hat{R}_{in,i}(s^*) \\
&\leq \sum_{i \in A^*} \hat{R}_{in,i} * E_i' * S_i(t - L_{max}/c - \epsilon_d) - \hat{R}_{in,i}(s^*) \\
&\leq \sum_{i \in A^*} E_i' * S_i(t - L_{max}/c - \epsilon_d - s^*) \\
&\leq \sum_{i=1}^{N} E_i' * S_i(t - L_{max}/c - \epsilon_d - s^*) \\
&\leq \sum_{i=1}^{N} E_i' * S_i(t - u^* - \epsilon_d) \ .
\end{aligned} \tag{2.36}$$

Since $\epsilon_d > 0$, (2.36) shows that (2.34) does not hold for this case.

It remains to show that for any session i and all t, we have

$$\hat{R}_{out,i}(t) \geq \hat{R}_{in,i} * \hat{S}_i(t) \ .$$

We first show that

$$\hat{R}_{out,i}(t) \geq P_i(\hat{R}_{in,i} * S_i(t - L_{max}/c)) \ . \tag{2.37}$$

To see this, suppose that $\hat{R}_{out,i}(t) < P_i(\hat{R}_{in,i} * S_i(t - L_{max}/c))$. By Lemma 26, this means that at least one conformant packet with deadline less than or equal to $t - L_{max}/c$ has not finished service by time t, which contradicts the first part of this Theorem.

Using (2.37), we have

$$\begin{aligned}
\hat{R}_{out,i}(t) &\geq P_i(\hat{R}_{in,i} * S_i(t - L_{max}/c)) \\
&= P_i(\inf_{\tau:\tau\in\mathbb{R}}\{\hat{R}_{in,i}(\tau) + S_i(t - L_{max}/c - \tau)\}) \\
&= \inf_{\tau:\tau\in\mathbb{R}} P_i(\{\hat{R}_{in,i}(\tau) + S_i(t - L_{max}/c - \tau)\}) \\
&\geq \inf_{\tau:\tau\in\mathbb{R}}\{\hat{R}_{in,i}(\tau) + (S_i(t - L_{max}/c - \tau) - L_{max,i})^+\} \\
&= \hat{R}_{in,i} * \hat{S}_i(t) \ , \qquad (2.38)
\end{aligned}$$

and we are done.

□

The result above roughly states that sufficient allocations of server bandwidth will guarantee a minimum service curve for each of the scheduler's sessions. We now consider a slightly more sophisticated scheduling algorithm, using elastic regulators at the input of all arriving sessions of the scheduler as depicted in Figure 2.6.

Corollary 28: Suppose each arriving session i first enters an elastic regulator with target envelope E_i prior to entering the server, and the server is allowed to server out-of-band packets when the scheduler queue is empty and there is positive backlog in at least one of the elastic regulators. If

$$\sum_{j=1}^{N} E'_j * S_i(x) \leq cx, \text{ for all } x \geq 0,$$

then for all i, we have $R_{in,i} \rightarrow (\hat{S}_i, \tilde{S}_i) \rightarrow R_{out,i}$, where

$$\hat{S}_i(t) = [S_i(t - L_{max}/c) - L_{max,i}]^+ \ ,$$

and

$$\tilde{S}_i(t) = \inf_{y:y\geq 0}\{[\hat{S}_i(t+y) - E'_i(y)]^+\} \ .$$

Proof of Corollary 28: Fix t. Recognize that the vacations (in the SCED-with vacations algorithm) having a maximum interval of time L_{max}/c, is equivalent to

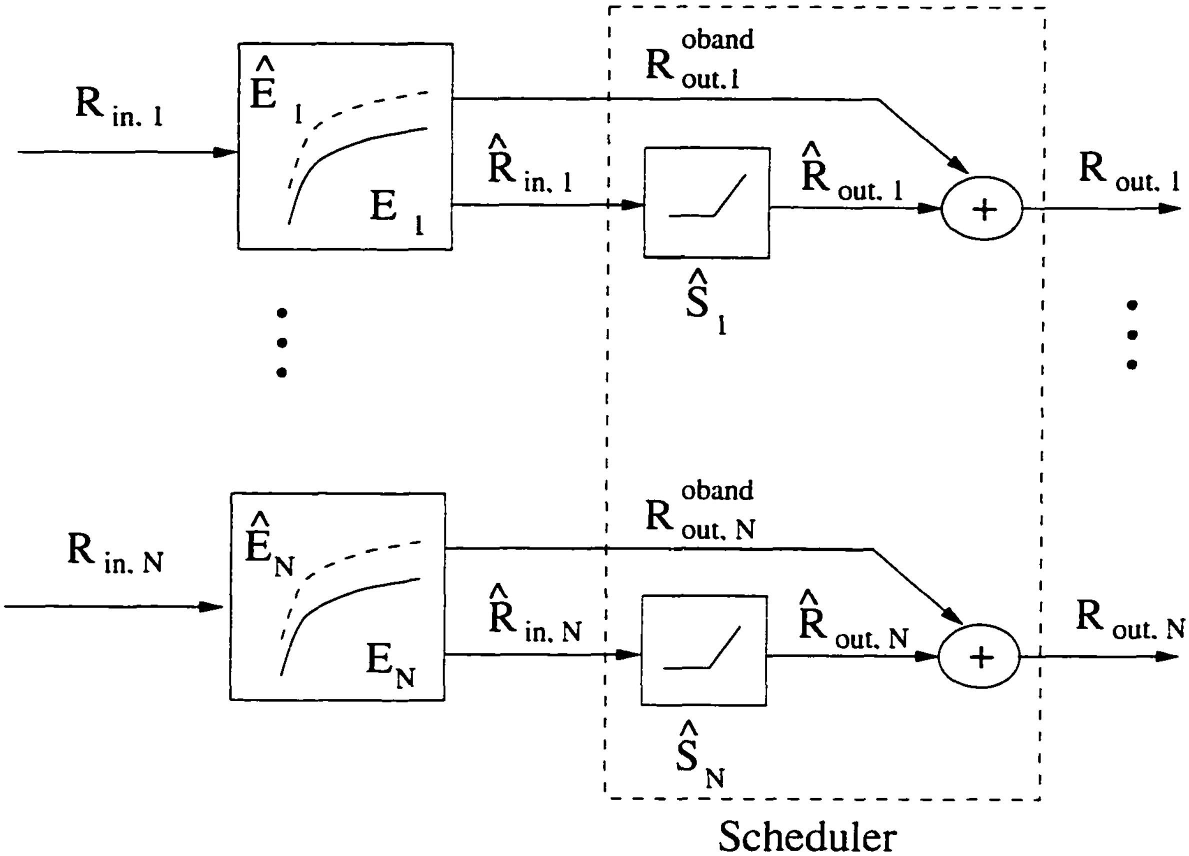


Figure 2.6: Scheduler with Elastic Regulators

a server with no vacations delivering out-of-band packet during these vacation intervals. Using Theorem 25, for all i, we have

$$\hat{R}_{out,i}(t) \geq \hat{R}_{in,i} * \hat{S}_i(t) \ .$$

Invoking Theorem 23, we have $R_{in,i} \rightarrow (\hat{S}_i, \tilde{S}_i) \rightarrow R_{out,i}$ and we are done.

□

2.5 Summary

In this chapter, we proposed a new adaptive service definition. With this new definition, we were able to find performance bounds, namely a virtual delay bound, without explicitly using an access regulator at the input to the network. We then obtained performance results for an adaptive session using a closed loop window flow control model. The closed loop window flow control results relied on the fact that there exists network elements capable of providing an adaptive service guarantee.

We recalled Figure 1.1, the unicast session traversing two routers employing hop-by-hop window flow control, where a link incurs some form of propagation delay and the router schedules packets via some scheduling algorithm. We first demonstrated in Section 2.1 that it was possible to obtain an adaptive service guarantee for a network element modeling propagation delay through the network. In Section 2.2 we showed in Example 20 that we could obtain end-to-end adaptive service guarantees for routers modeled as latency rate servers using hop-by-hop window flow control.

We proposed a new network element called the elastic regulator, where in combination with a scheduling algorithm with a minimum service curve guarantee and the ability to deliver additional traffic could in fact provide for adaptive service guarantees and moreover, allow for additional traffic to be served, i.e. to avail of excess bandwidth. We presented the SCED-with vacations scheduling algorithm, and then combined this algorithm with elastic regulators as in Corollary 28 to obtain adaptive service guarantees.

Chapter 3

Discussion

In this dissertation. we developed a mathematical framework for obtaining *both* delay and throughput guarantees. We considered closed loop window flow control as a protocol for adaptive applications and modeled network elements that may utilize excess bandwidth while providing adaptive service guarantees.

From our previous work [13. 1]. we determined the minimum window size (for hop-by-hop window flow control) at each hop to deliver minimum service curves unconstrained by the window flow control protocol. We will refer to this minimum window size at each hop as the minimum window requirement. In some sense. the minimum service curve represents worst case throughput. and so this previous result implied that it is possible to obtain worst case throughput unconstrained by window flow control. However. in the previous work. delay bounds were typically obtained with the use of an envelope. which in some cases implied under-utilization of network resources when the network was uncongested. In this work. we determined that for any hop(in a hop-by-hop window flow control model). we could deliver an adaptive service guarantee unconstrained by window flow control using the same minimum window requirement for each hop as in the previous results. Moreover. in this case. the hop-by-hop partial service curves were in fact equal to the minimum service curves of the original hop-by-hop model. We could also obtain an absolute service curve, at each hop, although the absolute service curve is typically strictly less than the minimum service curve since the absolute service curve contains a contribution due to the feedback network elements in the loop. Using the system backlog and the end-to-end absolute service curve, we obtained end-to-end worst case delay. Specifically, in Example 20, the worst case delay was on the order of *twice* the sum of the total round-trip end-to-end delay plus the time it takes to depart from the first throttle queue.

In our analysis, we have decoupled the adaptive sessions, to consider the flow control mechanism of a single session in isolation. It may be possible to eliminate the conditions which create the worst case effect of twice the round-trip delay by considering all traffic session simultaneously. One possibility for future work is to consider the aggregate effect of the competing traffic sessions and synchronize all sessions in order to reduce the worst case delay bound of twice the round-trip delay.

In our work, we assumed that an unspecified algorithm adjusted the window sizes, corresponding to sharing buffer resources in order to avail of excess bandwidth while providing adaptive service guarantees to each session. It would be interesting to consider the algorithms for adjusting the per hop window process $W(t)$. When considering possible algorithms for adjusting the window sizes at each hop, it is possible to consider algorithms that violate the buffer requirement at a particular time, although this could result in packet overflow, and essentially loss in the network.

It would be of interest to extend our framework to networks with loss. Although our current model is general, the results can not be directly applied to "lossy" networks, such as networks with buffer overflow or wireless networks, where multipath distortion and a deep fading environment can attribute to loss of packets along a wireless hop.

In this work, we have presented only an example of a scheduling algorithm capable of providing adaptive service guarantees. It is not clear what the necessary conditions are for providing these service guarantees, and so it remains an open issue to determine the "schedulability region" for adaptive service guarantees.

Appendix A

Lemma proofs for SCED - with vacations

The proofs in this appendix are from [11]. They are placed in this work for completeness and clarity.

Proof of Lemma 24 [11] : We wish to prove that $\hat{d}_i(t) = d_i(R_{in,i}(t))$.

Fix i and $t = \tau_i^k$. Suppose $\epsilon > 0$ is arbitrary. By definition of $\hat{d}_i(t)$, there exists τ^* such that $t \leq \tau^* \leq \hat{d}_i(t) + \epsilon$ and

$$\inf_{\tau:\tau\leq t}\{\hat{R}_{in,i}(\tau) + S_i(\tau^* - \tau)\} \geq \hat{R}_{in,i}(t) .$$

Since $\hat{R}_{in,i}(\tau) \geq \hat{R}_{in,i}(t)$ for all $\tau > t$, and S_i is non-negative, it follows that

$$\hat{R}_{in,i} * S_i(\tau^*) \geq \hat{R}_{in,i}(t) .$$

Thus, $d_i(\hat{R}_{in,i}(t)) \leq \tau^* \leq \hat{d}_i(t) + \epsilon$. By the definition of $d_i(\hat{R}_{in,i}(t))$, there exists a d' such that $d' \leq d_i(\hat{R}_{in,i}(t)) + \epsilon$ and $\hat{R}_{in,i} * S_i(d') \geq \hat{R}_{in,i}(t)$. We claim that $d' \geq t$. To see this, recognize that $\hat{R}_{in,i}(d) < \hat{R}_{in,i}(t)$ for all $d' < t$, since there is an arrival at time t. Thus, it follows that $\hat{R}_{in,i} * S_i(d) \leq \hat{R}_{in,i} * \delta(d) = \hat{R}_{in,i}(d) < \hat{R}_{in,i}(t)$ for all $d < t$. Hence, $d' \geq t$. We then have

$$\begin{aligned} \hat{R}_{in,i}(t) &\leq \hat{R}_{in,i} * S_i(d') \\ &\leq \inf_{\tau:\tau\leq t}\{\hat{R}_{in,i}(\tau) + S_i(d' - \tau)\} . \end{aligned} \tag{A.1}$$

From this, it follows that $\hat{d}_i(t) \leq d' \leq d_i(\hat{R}_{in,i}(t)) + \epsilon$. Since ϵ is arbitrary, we are done.

□

Proof of Lemma 26 [11]: We wish to prove that the total amount of traffic(in bits) from session i that has deadlines less than or equal to t is equal to $P_i(\hat{R}_{in,i} * S_i(t))$.

Let $T_i(t)$ be equal to the total amount of traffic(in bits) from session i that has deadlines less than or equal to t. Note that $P_i(T_i(t)) = T_i(t)$. First we show that $T_i(t) \leq P_i(\hat{R}_{in,i} * S_i(t))$. If $T_i(t) = 0$, this is trivial. Else, consider the last packet to arrive from session i that has deadline less than or equal to t, and suppose it arrived at time τ. Suppose the deadline of this packet is at time $t' \leq t$. Thus, $\hat{R}_{in,i} * S_i(t) \geq \hat{R}_{in,i} * S_i(t') \geq \hat{R}_{in,i}(\tau) = T_i(t)$. Since P_i is non-decreasing, we may apply P_i to both sides of the above inequality, resulting in $P_i(\hat{R}_{in,i} * S_i(t)) \geq P_i(T_i(t)) = T_i(t)$.

It remains to show that $T_i(t) \geq P_i(\hat{R}_{in,i} * S_i(t))$. Suppose that to the contrary, $T_i(t) < P_i(\hat{R}_{in,i} * S_i(t))$, and also suppose that j is such that $P_i(\hat{R}_{in,i} * S_i(t)) = \sum_{k=1}^{j} L_i^k$. Then the inequality $T_i(t) < P_i(\hat{R}_{in,i} * S_i(t))$ implies that packet j from session i is not counted in $T_i(t)$. However, the inequality $P_i(\hat{R}_{in,i} * S_i(t)) = \sum_{k=1}^{j} L_i^k$ implies that packet j from session i has a deadline at most t, and so packet j is in fact counted in $T_i(t)$. Thus, this contradiction implies that $T_i(t) \geq P_i(\hat{R}_{in,i} * S_i(t))$.

□

Proof of Lemma 27 [11]: We will prove that $T_i(s,t) = [P_i(\hat{R}_{in,i} * S_i(t)) - \hat{R}_{in,i}(s)]^+$.

If $\hat{R}_{in,i}(s) > P_i(\hat{R}_{in,i} * S_i(t))$, this implies that the last packet to arrive from session i up to time s has a deadline greater than t. In this case, all packets from session i that arrive after time s also have deadlines greater that t. Hence, $T_i(s,t) = 0 = [P_i(\hat{R}_{in,i} * S_i(t)) - \hat{R}_{in,i}(s)]^+$ in this case.

If $\hat{R}_{in,i}(s) \leq P_i(\hat{R}_{in,i} * S_i(t))$, then all packets that arrived up to time s

have deadlines at most t. Thus $T_i(s,t)$ is equal to the total amount of traffic from session i that has deadlines at most t, minus all arrivals up to $\hat{R}_{in,i}(s)$. By Lemma 26, this is equal to $P_i(\hat{R}_{in,i} * S_i(t)) - \hat{R}_{in,i}(s)$.

□

Bibliography

[1] R. Agrawal. R.L. Cruz. C. Okino. and R. Rajan. Performance bounds for flow control protocols. Technical Report Technical Report ECE-98-1. University of Wisconsin-Madison. ECE Department. may 1998.

[2] R. Agrawal and R. Rajan. Performance bounds for guaranteed and adaptive services. Technical Report Technical Report RC 20649. IBM Research Division. 1996.

[3] J. C. R. Bennett and H. Zhang. Hierarchical packet fair queueing algorithms. In *Proc. ACM SIGCOMM'96 Conference*. pages 143–156. Stanford University. CA. aug 1996.

[4] D. P. Bertsekas and R. G. Gallager. *Data Networks.*. Prentice Hall. 2nd edition. 1992.

[5] J.-Y. Le Boudec. *Network calculus made easy.* preprint edition. 1996.

[6] C.-S. Chang. On deterministic traffic regulation and service guarantee: a systematic approach by filtering. In *Proc. IEEE INFOCOM'97 Conference*. 1997.

[7] R. L. Cruz. A calculus for network delay. part I: Network elements in isolation. *IEEE Transactions on Information Theory*. 37:114–131. 1991.

[8] R. L. Cruz. A calculus for network delay. part II: Network analysis. *IEEE Transactions on Information Theory*. 37:132–141. 1991.

[9] R. L. Cruz. Service burstiness and dynamic burstiness measures: a framework. *Journal of High Speed Networks*. 1(2):105–127. 1992.

[10] R. L. Cruz. Quality of service guarantees in virtual circuit switched networks. *IEEE Journal on Selected Areas in Communications*. 13:1048–1056. aug 1995.

[11] R. L. Cruz. *Notes on Quality of Service Guarantees*. version 0.3α edition. Spring 1998.

[12] R. L. Cruz. Sced+: Efficient management of quality of service guarantees. In *Proc. IEEE INFOCOM'98 Conference*. IEEE Computer Society Press, mar 1998.

[13] R. L. Cruz and C. M. Okino. Service guarantees for window flow control. In *Proc. of the 34th Allerton Conference on Communication, Control, & Computing*, Monticello, IL, oct 1996.

[14] R. L. Cruz and C. M. Okino. Service guarantees for joint scheduling and flow control. In *Proc. of the 36th Conference on Decission & Control*, San Diego, CA, dec 1997.

[15] L. Georgiadis, R. Guérin, and A. Parekh. Optimal multiplexing on a single link: delay and buffer requirements. In *Proc. IEEE INFOCOM'94 Conference*, volume 2, pages 524–532, 1994. Also to appear in *IEEE Transactions on Information Theory*.

[16] E. L. Hahne. Round robin scheduling for fair flow control in data communications networks. *IEEE Journal on Selected Areas in Communications*, 9(7):1024–1039, sept 1991.

[17] J. Liebeherr, D.E. Wrege, and D. Ferrari. Exact admission control for networks with a bounded delay service. *IEEE/ACM Transactions on Networking*, 4(6):885–901, dec 1996.

[18] A. K. Parekh. *A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks*. Phd thesis, Massachusetts Institute of Technology, Laboratory for Information and Decision Systems, Cambridge, MA 02139, feb 1992. No. LIDS-TH-2089.

[19] A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The single node case. *IEEE/ACM Transactions on Networking*, 1(3):344–357, jun 1993.

[20] A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The multiple node case. *IEEE/ACM Transactions on Networking*, 2(2):137–150, apr 1994.

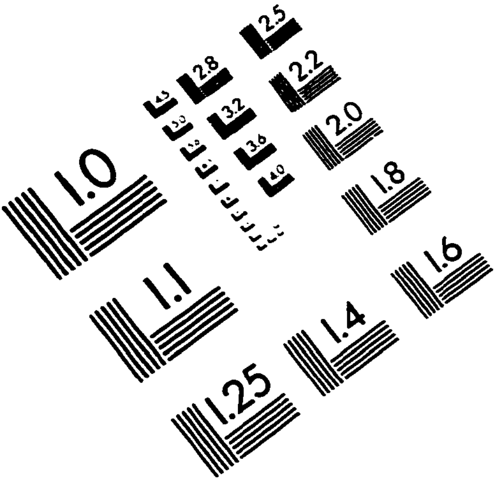
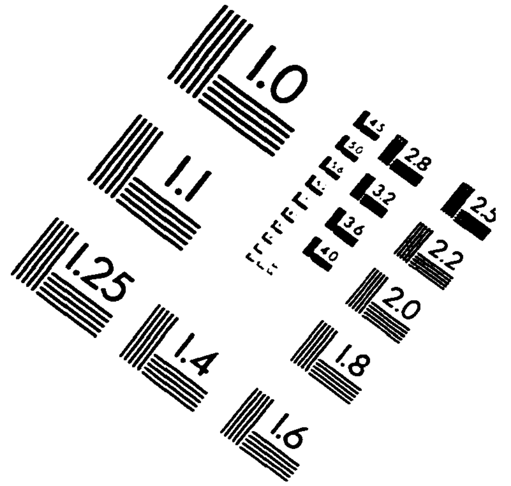
[21] H. Sariowan. *A service-curve approach to performance guarantees in integrated-service networks*. Ph.d. dissertation in electrical and computer engineering, University of California, San Diego, jun 1996.

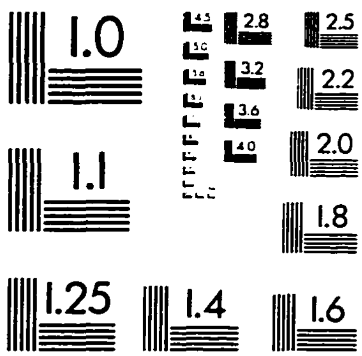
[22] H. Sariowan, R. L. Cruz, and G. C. Polyzos. Scheduling for quality of service guarantees via service cruves. In *Proc. International Conference on Computer Communications and Networks '95 (ICCCN'95)*, pages 512–20, sept 1995.

[23] I. Stoica, H. Zhang, and T.S. Eugene Ng. A hierarchical fair service curve algorithm for link-sharing, real-time, and priority services. In *Proc. ACM SIGCOMM'97 Conference*, 1997.

[24] H. Zhang and D. Ferrari. Rate-controlled static priority queueing. In *Proc. IEEE INFOCOM'93 Conference*, pages 227–236, San Francisco, apr 1993.

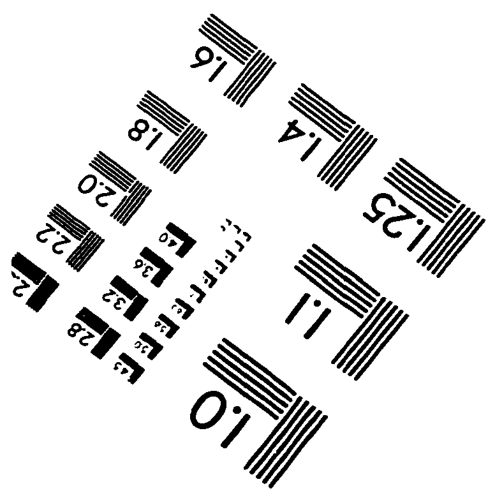
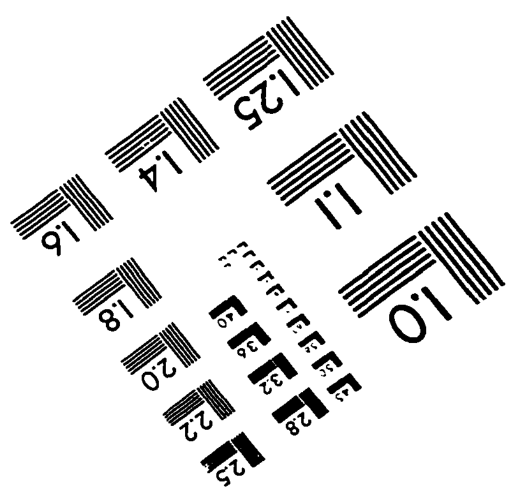
IMAGE EVALUATION TEST TARGET (QA-3)



150mm

6"

APPLIED IMAGE . Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989