

UC San Diego

UC San Diego Electronic Theses and Dissertations

Title

Quality of service guarantees for FIFO queues with constrained inputs

Permalink

<https://escholarship.org/uc/item/6wm911nz>

Author

Blanc, Alberto P.

Publication Date

2006

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA, SAN DIEGO

Quality of Service Guarantees for FIFO Queues with Constrained Inputs

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering (Communication Theory and Systems)

by

Alberto P. Blanc

Committee in charge:

Professor Rene Cruz, Chair
Professor Elias Masry
Professor Ramesh Rao
Professor Stefan Savage
Professor Alex Snoeren

2006

Copyright

Alberto P. Blanc, 2006

All rights reserved.

The dissertation of Alberto P. Blanc is approved,
and it is acceptable in quality and form for publi-
cation on microfilm:

Chair

University of California, San Diego

2006

TABLE OF CONTENTS

Signature Page	iii
Table of Contents	iv
List of Figures	vi
Acknowledgements	viii
Vita and Publications	ix
Abstract of the Dissertation	x
1 Introduction	1
1.1 Widespread Use of FIFO Queues	1
1.2 Existing and New Results	5
1.3 Network Calculus Overview	9
2 Service Curves for FIFO Queues	11
2.1 Introduction	11
2.2 QoS Bounds for a Single FIFO Queue with (σ, ρ) Envelopes	12
2.2.1 Maximum Delay	15
2.2.2 Output Envelope	17
2.2.3 Maximum Backlog	20
2.3 Delay Bounds for two FIFO Servers in Tandem	21
2.3.1 Considering Each Node in Isolation	23
2.3.2 Convolution of two service curves	23
2.3.3 Optimal Values of T_1 and T_2 for Worst Case Delay Computation	27
3 Service Mappings	33
3.1 Introduction	33
3.2 Service Models	34
3.2.1 Minimum Service Mappings	35
3.2.2 Shift Invariant Service Mappings	36

3.3	Quality of Service Guarantees for Shift Invariant Service Mappings	38
3.3.1	Linear Time Invariant Service Mappings	41
3.4	FIFO Multiplexers	42
3.4.1	A Single FIFO Multiplexer	42
3.4.2	FIFO Multiplexers in Tandem	43
4	Worst Case Average Delay for a Single FIFO Queue	46
4.1	Introduction	46
4.2	Problem Statement and Some Definitions	47
4.3	Preliminary Results	48
4.3.1	An optimization problem	49
4.3.2	Fixing the cross traffic and making some changes to the through traffic	61
4.4	Main Result	74
5	Conclusions	84
5.1	Open Problems	85
	Bibliography	88

LIST OF FIGURES

Figure 2.1:	The system	11
Figure 2.2:	$S_0^T(x)$ if $T < \frac{\sigma_1}{C}$	13
Figure 2.3:	$S_0^T(x)$ if $T = \frac{\sigma_1}{C}$	14
Figure 2.4:	$S_0^T(x)$ if $T > \frac{\sigma_1}{C}$	14
Figure 2.5:	Maximum horizontal distance between E_0 and S_0^T	16
Figure 2.6:	Service curve when $T = \frac{\sigma_0 + \sigma_1}{C}$	17
Figure 2.7:	$F(t + \tau)$ and $S_0^T(\tau)$ when $T \geq \frac{\sigma_1}{C}$ and $\frac{\sigma_0}{C - \rho_0} - t < T$	18
Figure 2.8:	$F(t + \tau)$ and $S_0^T(\tau)$ when $T \geq \frac{\sigma_1}{C}$ and $\frac{\sigma_0}{C - \rho_0} - t \geq T$	18
Figure 2.9:	$E_0(\tau)$ and $S_0^T(\tau)$	21
Figure 2.10:	A simple two node system	23
Figure 2.11:	Convolution: Case 1	24
Figure 2.12:	Convolution Case 2	24
Figure 2.13:	Argument of the infimum	26
Figure 2.14:	Two service curves for Case 1	28
Figure 2.15:	Sub-cases for Case 2	29
Figure 2.16:	The different cases in the a_1, a_2 plane	31
Figure 3.1:	\mathcal{S} -mapping: delay, backlog and output bounds	40
Figure 3.2:	Service curves: delay, backlog and output bounds	42
Figure 3.3:	Two nodes in tandem	44
Figure 4.1:	How to construct a better solution	52
Figure 4.2:	Water-filling condition	56
Figure 4.3:	How to construct $f'(x)$	56
Figure 4.4:	Constructing $\{T'_{nm}\}$ by switching T_{im_1} and T_{jm_2}	57
Figure 4.5:	Definition of α and γ	63
Figure 4.6:	$E_1(t_1) - Ct_1 \leq E_1(t_2) - Ct_2$ and $E_1(t_3) - Ct_3 \geq E_1(t_4) - Ct_4$	64
Figure 4.7:	Construction of $G(t)$	65
Figure 4.8:	Construction of $H(t)$	66
Figure 4.9:	A sample $R_{0,\text{in}}^{\text{III}}(t)$ and $R_{0,\text{in}}^*(t)$	71

Figure 4.10: The intervals for $\{T_{nm}^{\text{IV}}\}$ with two weights: $r_1 \geq r_2$ 76

ACKNOWLEDGEMENTS

I would like to acknowledge Professor Rene Cruz for his support as the chair of the committee. Through multiple drafts his guidance and teachings have proved to be invaluable.

I would also like acknowledge all the professors from the Electrical and Computer Engineering Department, the Computer Science and Engineering Department and the Mathematics Department that have taught all the classes that I have taken at UCSD. Their dedication and excellent technical preparation has greatly expanded my knowledge and understanding of many different topics.

Chapter 3, in full, is a reprint of the material as it appears in *Proceedings of the 2003 Allerton Conference on Communications and Control*. Alberto Blanc and Rene Cruz, 2003. The dissertation author was the primary investigator and author of this paper.

VITA

1998	"Laurea", <i>cum laude</i> , Politecnico di Torino, Italy
2001–2006	Research assistant, Department of Electrical and Computer Engineering, University of California San Diego
Summer 2003	Summer Internship Wireless Telephone Switching Systems, Carlsbad, California
Summer 2004	Summer Internship Sprint Advanced Technology Laboratory, Burlingame, California
2004	MSc in Electrical Engineering University of California at San Diego, La Jolla, California
2006	PhD in Electrical Engineering University of California at San Diego, La Jolla, California

PUBLICATIONS

Alberto Blanc, Yi-Kai Liu and Amin Vahdat, "Designing Incentives for Peer-to-Peer Routing," *Proceedings of IEEE INFOCOM 2005*, Miami FL, March 2005.

Alberto Blanc, Yi-Kai Liu and Amin Vahdat, "Designing Incentives for Peer-to-Peer Routing," *Proceedings of the 2nd Workshop on the Economics of Peer-to-Peer Systems*, Boston MA, June 2004.

Rene L. Cruz and Alberto P. Blanc, "A Service Abstraction with Applications to Network Calculus," *Proceedings of the 2003 Allerton Conference on Communications and Control*, Monticello IL, September 2003.

FIELDS OF STUDY

Major Field: Computer Engineering

Studies in Quality of Service guarantees for packet switched networks

Professor Rene L. Cruz

Studies in peer-to-peer networks

Professor Amin Vahdat

ABSTRACT OF THE DISSERTATION

Quality of Service Guarantees for FIFO Queues with Constrained Inputs

by

Alberto P. Blanc

Doctor of Philosophy in Electrical Engineering (Communication Theory and Systems)

University of California San Diego, 2006

Professor Rene Cruz, Chair

First-In First-Out (FIFO) queues are widely used in packet switched communication networks and they have been studied extensively, from a probabilistic point of view, in the Queuing Theory field. The Network Calculus framework was proposed as an alternative to the probabilistic approach and provides deterministic bounds provided the inputs do satisfy deterministic constraints (envelopes).

In the first part of this dissertation we use service curves to analyze FIFO queues with inputs constrained by sigma-rho envelopes. Using previously known results we show that for the case of a single FIFO queue it's possible to recover known and tight bounds for different Quality of Service (QoS) metrics. For the case of two FIFO queues in tandem we find the smallest possible end-to-end delay bound, with this approach, by choosing the appropriate service curve at each node but we found that, in general, this bound is still not achievable.

In order to find a better bound, we define a general service abstraction, which is defined in terms of a “service mapping,” which is a monotone operator that maps an arrival process to a lower bound on the corresponding departure process. For a network element with a shift invariant service mapping, we obtain bounds on the maximum delay, maximum backlog, and a traffic envelope for the departure process, assuming that the arrival process to the network element conforms to a traffic envelope. Using the service

mapping abstraction to analyze a network of two FIFO queues in tandem, whose arrival processes conform to traffic envelopes, we obtain achievable upper bounds on end-to-end delay which are smaller than those that can be obtained with previously proposed methods.

In the second part we address the problem of worst case *average delay* for a single FIFO queue with constrained inputs, where the average is over time. We show that if a flow has a piecewise linear and concave envelope and shares a single FIFO queue with another flow that has a concave envelope it's possible to obtain a tight bound on the worst case average delay.

1

Introduction

1.1 Widespread Use of FIFO Queues

First-In First-Out (FIFO) queues have always been widely used in packet switched communication networks. Most likely this popularity is due to the fact that this queuing discipline is very simple to implement and manage. There is no need for any sort of calculations, all that is needed it is to keep track of the order in which packets arrived, allowing simple implementations either in software or in hardware¹.

Another noteworthy consequence of this simplicity is that there are virtually no configuration parameters, a FIFO queue is completely described by the rate at which it serves packets and the length of the buffer used to hold packets waiting to be served. Often these two parameters are fixed at design time so that the operator does not need to configure anything at all, simplifying the management of devices using this queuing discipline.

Naturally FIFO queues do have shortcomings, not surprisingly caused by the very same simplicity that makes them attractive for ease of implementation and management. Whenever packets belonging to two or more flows share the same FIFO queue the service received by each flow depends on the characteristics of the other flows: the more traffic the competing flow sends the worst the service is. In the worst case the competing flows might be sending so much traffic that one or more flows could be completely starved, that is they would be receiving no service at all because all their packets would be discarded

¹Recently Chang has proposed an implementation that uses only optical components [10].

for lack of space in the buffer (in the case of an infinite buffer no packets are lost but the delay is not bounded).

Another aspect of this dependence on the behavior of all the traffic sharing a FIFO queue is the lack of fairness: the capacity of the server is not evenly divided among all the flows using the queue, instead the more packets a flow is sending the more service it will receive. Furthermore there are no mechanisms in place guaranteeing the stability of the system. That is a single source that is sending too much traffic can cause arbitrary large delays and packet losses for all the other flows sharing the same queue.

Because of these shortcomings FIFO queues with unregulated inputs cannot offer any Quality of Service (QoS) guarantees and are not suitable for carrying traffic that may need certain service guarantees like multimedia streams or other real-time traffic. To address this issue many scheduling algorithm have been proposed from simpler ones like priority queuing to more complicated ones like Packet-by-packet Generalized Processor Sharing (PGPS) [25, 26] (also known as Weighted Fair Queuing [16]), Virtual Clock [32], Service Curve Earliest Deadline (SCED), and SCED+ [15, 27], just to name a few. All these scheduling algorithms offer some form of QoS guarantees and do protect each flow from the behavior of the other flows sharing the same queue. This way, no matter what these flows do, some minimum service will always be offered to each flow (the details about these guarantees vary for each algorithm).

Using these more sophisticated scheduling disciplines it is possible to construct packet switched networks capable of delivering different QoS guarantees to different flows. One issue with this approach is that it requires each queue in the network to analyze each incoming packet to determine to which flow it belongs so that it can be treated accordingly. It is possible to simplify this problem by using a specific field in the header of each packet containing a value that uniquely identifies a single flow, for example the Flow Label field in the IPv6 header is specifically intended for this purpose. If such a field is not available, as is the case in IPv4, then a collection of fields must be analyzed to determine to which flow the packet belongs to. In the case of IPv4 five fields are usually used: source and destination IP addresses, source and destination ports and protocol id. In either case a certain amount of processing power and state information (with associated costs in terms of time and memory usage) is required to perform these lookup operations.

Another issue with some of these algorithms is that each node needs to maintain a

certain amount of state information for each active flow. In addition each scheduling decision can involve a non-negligible number of calculations. Often these shortcomings become less and less relevant with advances in processor, memory and other technologies allowing to limit their impact, but they are still present.

On the management and configuration side these algorithms have the disadvantage that they all need some sort of configuration. Given that they can treat each flow in a different way these algorithms have a series of parameters that need to be configured either by an operator or by some automated signaling mechanism or by a combination of both. Several signaling solutions have been proposed, one of them is the Resource Reservation Protocol (RSVP) [6, 7] developed by the Integrated Services (IntServ) working group of the Internet Engineering Task Force (IETF) [5, 7, 28–30]. One problem with this approach is that it does not scale well as the number of flows increases. As their number grows the requirements in terms of signaling, configuration and state information grow linearly. Again technological advances might mitigate this problem but at least the signaling part with the associated call admission control problem can be problematic: whenever two nodes want to start a flow they need to signal this intention to the network and the intermediate nodes (e.g. routers) need to decide whether they should admit the new flow into the network or not. Especially in the case where a single flow needs to cross multiple providers (as it is often the case in the Internet) this can be even more complicated because each provider might have different constraints and/or preferences in terms of what flows to admit and how to route them. Also, allowing automated signaling between different providers presents significant trust and security issues: each provider would have to trust completely its peers, even though some of their nodes could be misconfigured or compromised.

To address some of these issues the IETF created another working group called Differentiated Services with the aim of introducing a much simpler solution. The basic idea is that instead of dealing with each flow individually multiple flows would be grouped together in a single “aggregate flow”, sometimes referred to as a *class* or *macro flow* as opposed to *micro flows* (as the individual flows are often called) [3, 17, 20, 21, 24]. All the packets belonging to the same aggregate carry the same value in a specific field in the header [24], in IPv4 the first six bits of the Type of Service field were used to create a new field called DiffServ Codepoint (DSCP) while in IPv6 the first six bits of the Traffic Class octet were redefined in the same way. Within each aggregate packets are served in FIFO

manner. Similarly when a flow joins an aggregate at a certain node this node will insert these new packets into the aggregate using FIFO scheduling.

In a way, DiffServ is trying to combine the best of both worlds: the small number of aggregates makes it scalable and more easily manageable while the fact that flows are grouped in different macro flows allows service differentiation and QoS guarantees. As the number of flows increases the number of macro flows does not and the fact the FIFO is used within each aggregate means that there is no need for extra state or explicit configuration parameters every time a new flow is added. If, instead, IntServ is used every newly admitted flow forces all the nodes involved to increase the amount of state information kept, also each flow needs to specify certain number of configuration parameters (like required rate). At the same time DiffServ does offer service differentiation and can protect each macro flow from the behavior of the other macro flows sharing the same queue, provided appropriate scheduling algorithms are used.

The small number of macro flows simplifies the management and signaling aspect as well, but, unfortunately, it does not make this problem go away completely. One possible solution is for human operators to configure all the parameters. Given that there is only a small number of aggregates and that they are expected to be long lived makes this a viable, albeit probably not optimal, solution. For example one macro flow could be used for high-quality voice calls, another one for low-quality voice calls and a third one for video conferencing.

So far we have glossed over the problem of how flows are admitted to each aggregate. Clearly if there is no control at all and any node is allowed to join any macro flow sending an arbitrary amount of traffic it is impossible to offer QoS guarantees to the other flows using the same aggregate. Of course it is still possible to give certain guarantees to the other aggregates (the details depends on the specific algorithms used), but this is not enough. If we consider the example with the voice calls all the users using the high quality voice aggregate would expect some QoS guarantee.

As described by the DiffServ standards [3] it is possible to do this by limiting the amount of traffic that is allowed to join each aggregate. If all the flows sharing a FIFO queue are such that certain constraints are met it is possible to find the worst case arrival patterns, and corresponding worst case bounds, for different QoS metrics. Therefore, at least in theory, it should be possible to provide specific guarantees to all the flows

sharing the same aggregate, provided that all the flows involved meet certain constraints. In some cases these bounds might be known *a priori* (for example based on the specific implementation of a certain user application) but, in general, it is always possible to use *shapers* to make sure that all the flows joining a certain aggregate do meet all the requirements that they are supposed to. Simplifying things a bit we could say that a shaper is a network device that has as input an arbitrary flow and as output a flow that does satisfy certain constraints. Typically the shaper will hold (delay) some of the incoming packets until they can be transmitted without violating the constraints. In other cases the shaper might simply drop non-conforming packets or mark them accordingly.

The details of the DiffServ architecture are beyond the scope of this work and the interested reader is referred to [3, 24] and references therein for more details. What is relevant to this work is that DiffServ uses FIFO scheduling to serve packets within the same aggregate and that all the flows belonging to an aggregate can be forced to conform to certain requirements. In the next section we formalize these requirements and introduce some existing results.

1.2 Existing and New Results

Queuing theory is a vast and well-established field with a great number of well-known works and results. The great majority of these works address the problem from a probabilistic point of view and deal with random variables and associated distributions. In order to apply these results it is vital to know the statistical properties of the incoming work, whether it is customers to a bank, broken equipment to a repair shop or packets in a communication network. Unfortunately, if the wrong assumptions are used for the incoming traffic the results can be meaningless.

Another possible solution is to consider deterministic bounds: in [14] it is shown that, as long as the inputs do satisfy certain constraints, one can obtain QoS bounds for a single FIFO queue. Instead of dealing with random variable and probability distributions it is possible to use simple constraints on the input flows, which are very reasonable for data traffic.

Before we can precisely define these bounds we need to introduce some definitions: let $R_{\text{in}}(t)$ be a non-decreasing positive function representing the total amount of traffic

sent by a certain flow up to time t . We say that $R_{\text{in}}(t)$ has envelope $E_{\text{in}}(t)$ and write $R_{\text{in}}(t) \sim E_{\text{in}}(t)$ if for any $s \leq t$ we have:

$$R_{\text{in}}(t) - R_{\text{in}}(s) \leq E_{\text{in}}(t - s),$$

in other words we are upper bounding the amount of traffic that can be sent during a certain time interval by a function that depends only on the length of this interval. One envelope that is widely used because of its simplicity is the so called *sigma-rho* envelope, in this case $E(t) = \sigma + \rho t$. Often the notation (σ, ρ) is used to indicate this type of envelope. Sometimes a maximum rate constraint is added so that $E(t) = \max\{Ct, \sigma + \rho t\}$ where C is the maximum rate allowed. Note that this simple envelope has an interesting interpretation in the context of data traffic: the parameter σ implies an upper bound of the burst size, that is the number of back to back packets that the flow can send.

From [14] we know that if all the inputs to a single FIFO queue satisfy an envelope (each flow can have a different one) it is possible to derive some worst case bounds, among these a worst case delay bound and a worst case output envelope. Furthermore these bounds are *tight*, that is there exist arrival patterns that do achieve these bounds. At the same time, at least in the case of delay, only some bits (or packets) will experience a delay equal to the worst case bound, therefore it can be argued that these bounds give a somewhat pessimistic representation of reality given that only a (possibly small) portion of the data traffic will experience them, even in the worst case. Nonetheless these bounds can be useful especially for traffic that has specific delay constraints, like, for example, voice and video. In both cases if a packet is delivered too late it cannot be used therefore it is useful to have a delay bound that can be used as a worst case.

As suggested in [12], it is possible to use the results in [14] to analyze networks of FIFO queues. For example if we have two or more nodes in tandem and all the input flows have envelopes we can consider each node in isolation and derive a worst case delay bound at each hop, then simply add the delay bounds for all the nodes used by a certain flow. Note that when a flow first enters the network we know its envelope and, knowing the envelopes of all the other flows going through that node, we can find the output envelope as well. Next, we can use this output envelope as the input envelope for the following hop. This way we can analyze a vast class of networks (all those that do not have loops) but it is not immediately clear whether the bounds obtained using this process are tight or not. In particular if we consider a very simple network with two FIFO queues with one flow that

goes through both queues and two flows that go through only one hop (one at each node) it seems impossible to construct an arrival pattern where at least one packet achieves the sum of the worst case delays at each hop. Note that it is possible to construct arrival patterns where different packets will achieve the worst case delay at each hop but this is not enough to show that the *end-to-end* delay bound is tight. Of course the fact that no one has been able to provide an arrival pattern that does achieves this end-to-end delay does not mean that it does not exist but it justifies looking for a possibly smaller (and hopefully tight) bound.

One possibility is to use service curves to obtain delay bounds. Service curves have been introduced in [19] and are a formalism that allows, among other things, to find worst case delay bounds, provided the flows in question do have envelopes (the following section contains a brief overview of this topic). Even though service curves have been introduced in the context of a specific scheduling algorithm they can be used to characterize other queuing disciplines as well. In [15] Cruz shows that a single FIFO queue serving multiple flows, each with an envelope, can be characterized by an infinite number of service curves. In Chapter 2 we use this result to show that, in the specific case where all the envelopes are (σ, ρ) , it is possible to recover the results in [14] and that using some of the properties of service curves it is possible to find a smaller end-to-end delay for the two queues case. Yet, even for this smaller bound it seems that there is no arrival pattern achieving it, this leads us to believe that there might exist a smaller bound.

As we just mentioned, from [15] we know that there exists an infinite number of service curves that characterize a FIFO queue. Using this fact in Chapter 3 we introduce a new service abstraction, which includes service curves as a special case, that allows us to derive a smaller and *achievable* end-to-end delay bound for two FIFO queues in tandem.

So far we have been dealing with a bound of the worst case delay and, as noted above, this bound could be overly pessimistic in the sense that even if it achievable only a fraction of the traffic can achieve this delay even in the worst case. One way of addressing this concern is to consider the *average delay*, where we are averaging over time. Of course it is possible to use the worst case delay bound as a bound for the average delay: if every packet cannot have a delay bigger than this bound obviously the average cannot be any bigger. But it is natural to ask ourselves if it is possible to obtain a smaller bound. In [14] Cruz shows that if a single flow, with an envelope, is served by a FIFO queue it is possible

to find such a bound that it is, in general, smaller than the worst case delay. In Chapter 4 we will generalize this result to the case where two flows share the same FIFO queue.

In the next section we will see that it is useful to have a *point-wise* lower bound on the output of a network element. Such a bound can be used to obtain, among other things, bounds on delay and output envelope. If $R_{\text{in}}(t)$ represents the total amount of traffic arrived up to time t at a certain network element and $R_{\text{out}}(t)$ represents the total amount of traffic that has left, a lower bound is a function $L(t)$ such that $R_{\text{out}}(t) \geq L(t)$ for all the $R_{\text{in}}(t)$ that satisfy the envelope. Naturally we would like this bound to be as good as possible, ideally we would like that for any arbitrary t there exist an arrival pattern such that $R_{\text{out}}(t) = L(t)$. But even if this is true, in most cases, this bound is not as tight as it may seem: if we are interested in a time interval $[t_1, t_2]$ it is not possible, in general, to find an arrival pattern such that the output is exactly equal to $L(t)$ for all $t \in [t_1, t_2]$. Intuitively this is because the worst case can be achieved at different times but not infinitely often. For example, in the case of a sigma-rho envelope the worst case delay is achieved at the end of a burst of the maximum size but, because of the envelope, a burst of size σ can occur at most every $\frac{\sigma}{\rho}$ units of time, therefore if we want to achieve the output bound at time t we cannot achieve it anywhere else in the interval $[t - \frac{\sigma}{\rho}, t + \frac{\sigma}{\rho}]$. This shows that $L(t)$ might not be a good representative for the class of all possible outputs, it is simply a lower bound for this class.

At the same time the worst case average delay can be used to obtain a bound on the area between $R_{\text{in}}(t)$ and $R_{\text{out}}(t)$. Given that this is not a point-wise bound it can be achieved over arbitrary time intervals, and it can be used to better characterize the class of all possible output patterns. In particular given any non-decreasing function f such that f is consistent with the output envelope and $R_{\text{in}}(t) \geq f(t) \geq L(t)$ we can reject f as a possible output if the area between $R_{\text{in}}(t)$ and $f(t)$ is greater than the one prescribed by the worst case average delay. Note that if f does not violate this constraint we cannot guarantee that there exists an arrival pattern such that the output will be exactly equal to f but this extra condition allows us to better characterize the set of all feasible outputs.

1.3 Network Calculus Overview

We conclude this chapter with a brief introduction on what it is often called “Network Calculus;” for an exhaustive presentation the interested reader is referred to [9, 23] and references therein. As we have previously mentioned, let $R_{\text{in}}(t)$ and $R_{\text{out}}(t)$, respectively, represent the total amount of traffic that has arrived at (departed from) a certain network element up to time t . These positive and non-decreasing functions are often referred to as input and output processes.

We say that a network element offers a (minimum) service curve S if for any input process $R_{\text{in}}(t)$ the corresponding output process $R_{\text{out}}(t)$ is such that

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

where

$$(R_{\text{in}} * S)(t) = \inf_u \{R_{\text{in}}(u) + S(t - u)\}$$

is the convolution in the min-plus algebra.

If the process $R_{\text{in}}(t)$ has envelope $E_{\text{in}}(t)$ and it is fed to a network element that offers a minimum service curve $S(t)$ the delay through this element is upper bounded by the maximum *horizontal* distance between the input envelope and the service curve. If this distance is unbounded the corresponding system is not stable, that is the delay can be arbitrarily large. Formally we can define this quantity by using the “delta” function

$$\delta_{\Delta}(t) = \begin{cases} 0, & \text{if } t \leq \Delta \\ \infty, & \text{if } t > \Delta. \end{cases}$$

It is easy to see that when we convolve a function with $\delta_{\Delta}(t)$ we are simply shifting the function to the right by Δ , that is $f * \delta_{\Delta}(t) = f(t - \Delta)$. We can combine this with the fact that the maximum horizontal distance between the envelope and the service curve is such that if we shift the envelope to the right by such distance this shifted version of the envelope is less than or equal to the service curve. Formally we can express the bound on delay as:

$$\hat{d} = \inf \{ \Delta : \Delta \geq 0 \text{ and } S(x) \geq (E_{\text{in}} * \delta_{\Delta})(x) \forall x \geq 0 \}.$$

In the same scenario (network element offering service curve S to $R_{\text{in}}(t)$ that has envelope $E_{\text{in}}(t)$) the maximum *vertical* distance between the envelope and the service

curve is an upper bound for the amount of traffic stored at the network element. Formally if we define the backlog $b(t)$ as the difference between the input and output process at time t we have that (for any t):

$$b(t) = R_{\text{in}}(t) - R_{\text{out}}(t) \leq \sup_{u \geq 0} \{E_{\text{in}}(u) - S(u)\}.$$

Finally, again in the same scenario, we can find the output envelope $E_{\text{out}}(t)$ as

$$E_{\text{out}}(t) = (E_{\text{in}} \oslash S)(t)$$

where we have used the deconvolution operator in the min-plus algebra:

$$(F \oslash G)(t) = \sup_{u \geq 0} \{F(t+u) - G(u)\}.$$

We conclude this brief overview with a result that is very useful for analyzing networks: suppose the flow represented by the process $R_{\text{in}}(t)$ goes through N elements each offering a service curve S_i , $i = 1 \dots N$. In this case, the entire network is equivalent to a single node that offers the service curve $S_{\text{net}} = S_1 * S_2 * \dots * S_N$, that is by convolving the service curves of each element we can reduce an entire network to a single element that offers a “network” service curve S_{net} . Using this single service curve we can use all the results that we have just presented, simplifying significantly the analysis of the network case. Furthermore it is often the case that the bounds obtained using this network service curve are better than those obtained by considering a collection of elements in isolation, where the end-to-end bounds are obtained by adding the bound at each node.

2

Service Curves for FIFO Queues

2.1 Introduction

In the first part of this chapter we analyze a single FIFO queue (server) operating at rate C using the service curves given in Theorem 4 of [15]. For the sake of simplicity we consider only two streams entering the server as in figure 2.1. We are concerned with flow 0 described by the process R_0 which has envelope E_0 . Flow 1 represents all the other traffic entering the FIFO server (if more then two flows enter the system then all the “other” flows can be combined in one bigger flow). Again for the sake of simplicity we assume that both flows have a sigma-rho envelope.

For convenience this is Theorem 4 from [15] :

Theorem 2.1. *Suppose two traffic streams enter a network element, where $R_{i,in}$ and $R_{i,out}$ are the corresponding input and output streams, $i = 0, 1$. the aggregate input and output streams are given by $R_{in} = R_{0,in} + R_{1,in}$ and $R_{out} = R_{0,out} + R_{1,out}$. Suppose it is known that $R_{1,in}$ is E_1 -smooth, and that the aggregate stream is guaranteed the minimum*

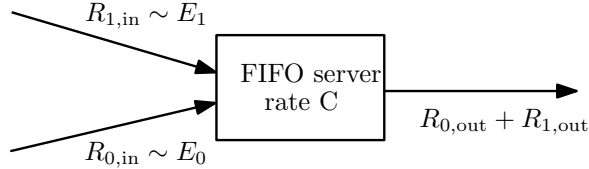


Figure 2.1: The system

service curve S , i.e. $R_{out} \geq R_{in} * S$. If the network element serves packets among the two streams in FCFS order, then for any fixed $T \geq 0$, the first stream $R_{0,in}$ is guaranteed the minimum service curve S_0^T , where

$$S_0^T(x) = \begin{cases} 0, & \text{if } x < T \\ [S(x) - E_1(x - T)]^+, & \text{if } x \geq T. \end{cases}$$

As we are considering a FIFO server operating at rate C the service curve guaranteed to the aggregate of flow 0 and 1 is $S(x) = Cx$.

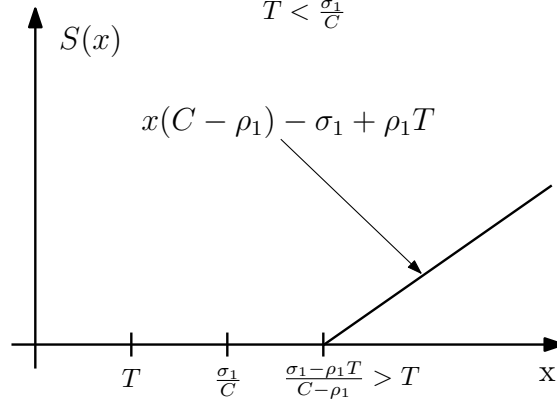
Using these service curves we can recover the tight bounds on delay and output envelope presented in [14], furthermore we can derive a previously unavailable upper bound on the backlog for flow 0. Note that without using the service curves it is possible to obtain an upper bound on the total backlog but not on the backlog of each flow.

In the second part of the chapter we will consider two FIFO queues in tandem, in this case flow 0 goes through both servers and it is therefore called the “through traffic” while at each server there is another interfering flow that uses only one server, not surprisingly these other flows are often referred to as “cross traffic.” Using the results derived in the first part of the chapter we find the best possible (i.e. smallest) end-to-end delay bound. Given that an infinite number of service curves characterize each node we have to find the “best” service curve for each node, that is the values for T_1 and T_2 that give the smallest delay bound. Exploiting the linearity of sigma-rho envelope we can find a closed formula expression for this optimization problem.

2.2 QoS Bounds for a Single FIFO Queue with (σ, ρ) Envelopes

In this section we will use the service curves given by Theorem 4 when the input flows are leaky bucket constrained, that is $E_i(x) = \rho_i x + \sigma_i$. Note that we allow an infinite incoming speed. Given those constraints the service curve $S(x)$ guaranteed to flow 0 is:

$$S_0^T(x) = \begin{cases} 0, & \text{if } x < T \\ [x(C - \rho_1) - \sigma_1 + \rho_1 T]^+, & \text{if } x \geq T. \end{cases} \quad (2.1)$$

Figure 2.2: $S_0^T(x)$ if $T < \frac{\sigma_1}{C}$

It is easy to rewrite this expression without using the \llbracket^+ expression. It is enough to consider two cases: if $T \leq \frac{\sigma_1}{C}$ then $S_0^T(x) = 0$ for $x < \frac{\sigma_1 - \rho_1 T}{C - \rho_1}$ which is greater than T and $\frac{\sigma_1}{C}$ (if $T < \frac{\sigma_1}{C}$) and (2.1) becomes:

$$S_0^T(x) = \begin{cases} 0, & \text{if } x < \frac{\sigma_1 - \rho_1 T}{C - \rho_1} \\ x(C - \rho_1) - \sigma_1 + \rho_1 T, & \text{if } x \geq \frac{\sigma_1 - \rho_1 T}{C - \rho_1}, \end{cases} \quad (2.2)$$

Figure 2.2 shows the graph of $S_0^T(x)$ if $T < \frac{\sigma_1}{C}$.

If $T \geq \frac{\sigma_1}{C}$, (2.1) becomes:

$$S_0^T(x) = \begin{cases} 0, & \text{if } x < T \\ x(C - \rho_1) - \sigma_1 + \rho_1 T, & \text{if } x \geq T. \end{cases} \quad (2.3)$$

Note that whenever $S_0^T(x)$ is non-zero it is a straight line with slope $C - \rho_1$, independent of the value of T . On the other hand, the value of the service curve for $x = T$ is $S_0^T(T) = TC - \sigma_1$ and does depend on T . Figure 2.4 shows the graph of $S_0^T(x)$ if $T > \frac{\sigma_1}{C}$. Note that in this case there is a discontinuity at $x = T$ and that if T increases $S_0^T(T)$ will move on a line with slope C (more precisely this line is $Cx - \frac{\sigma_1}{C}$ and it is the dashed line in the figure).

Finally Figure 2.3 shows the graph of $S_0^T(x)$ if $T = \frac{\sigma_1}{C}$. The graph is very similar to the one in Figure 2.2; the only difference is that there is no discontinuity for $x = T$. This the “boundary” case between the two previous as (2.2) and (2.3) are identical when $T = \frac{\sigma_1}{C}$.

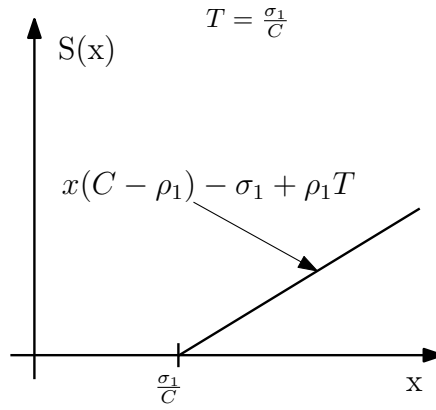


Figure 2.3: $S_0^T(x)$ if $T = \frac{\sigma_1}{C}$

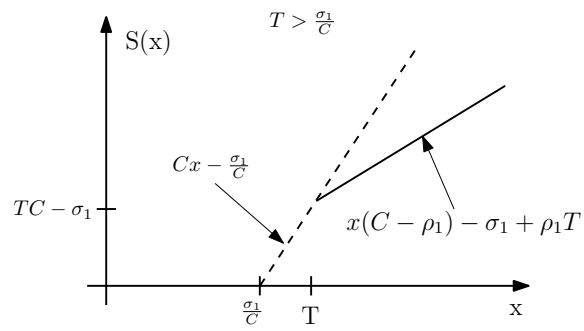


Figure 2.4: $S_0^T(x)$ if $T > \frac{\sigma_1}{C}$

The following proposition will be useful to simplify the search for QoS bounds:

Proposition 2.2. *Let $S_0^T(x)$ be defined as in (2.1) and let T_1 be a value for the parameter T such that $T_1 < \frac{\sigma_1}{C}$. For any such value it is possible to upper bound the corresponding service curve $S_0^{T_1}(x)$ with $S_0^{T_2}(x)$ where $T_2 = \frac{\sigma_1}{C}$, that is $S_0^{T_1}(x) \leq S_0^{T_2}(x)$ for all x .*

Proof. The fact that $T_1 < \frac{\sigma_1}{C}$ implies that $S_0^{T_1}(x)$ is defined as in (2.2). Similarly $T_2 = \frac{\sigma_1}{C}$ implies that $S_0^{T_2}(x)$ is defined as in (2.3). Obviously $S_0^{T_1}(x) = S_0^{T_2}(x) = 0$ for $x \leq T_1$. Also, as noted above, $T_1 < \frac{\sigma_1}{C}$ implies that $\frac{\sigma_1 - \rho_1 T_1}{C - \rho_1} \geq \frac{\sigma_1}{C} = T_2$ so that $S_0^{T_1}(x) = S_0^{T_2}(x) = 0$ for $T_1 \leq x \leq T_2$ as well. For $T_2 \leq x \leq \frac{\sigma_1 - \rho_1 T_1}{C - \rho_1}$ from (2.2) we have $S_0^{T_1}(x) = 0$. From (2.3) we have $S_0^{T_2}(x) \geq 0$ (to see why this is true observe that $S_0^{T_2}(T_2) = 0$ and that $S_0^{T_2}(x)$ is a straight line with slope $C - \rho_1 > 0$). Hence $S_0^{T_1}(x) \leq S_0^{T_2}(x)$ in this case as well. Finally for $x \geq \frac{\sigma_1 - \rho_1 T_1}{C - \rho_1}$ both $S_0^{T_1}(x)$ and $S_0^{T_2}(x)$ are straight lines with slope $C - \rho_1$ but $S_0^{T_2}(\frac{\sigma_1 - \rho_1 T_1}{C - \rho_1}) = 0$ while $S_0^{T_1}(\frac{\sigma_1 - \rho_1 T_1}{C - \rho_1}) > 0$ as $\frac{\sigma_1 - \rho_1 T_1}{C - \rho_1} > \frac{\sigma_1}{C}$. Therefore in this case we also have that $S_0^{T_1}(x) \leq S_0^{T_2}(x)$. \square

Combining the fact that $f \leq \tilde{f}$ and $g \leq \tilde{g}$ implies $f * g \leq \tilde{f} * \tilde{g}$ (see, for example II.8 in [11]) with Proposition 2.2 we have that for any input process $R_{0,\text{in}}(t)$ the output bound obtained with a value for the parameter $T_1 \leq \frac{\sigma_1}{C}$ can be improved by using $T_2 = \frac{\sigma_1}{C}$ given that $R_{0,\text{in}} * S_0^{T_2}(t) \geq R_{0,\text{in}} * S_0^{T_1}(t)$ for any t .

2.2.1 Maximum Delay

It is known that the maximum delay for a flow with envelope $E(x)$ served by a network element that offers a minimum service curve S is given by (see for example Proposition 5 of [1]):

$$\hat{d} = \inf\{\Delta : \Delta \geq 0 \text{ and } S(x) \geq E * \delta_\Delta(x) \forall x \geq 0\}$$

that is the maximum delay is upper bounded by the minimum Δ by which we need to shift the input envelope E in order for $S(x)$ to be greater than E for all x . This is equivalent to the maximum horizontal distance between the input envelope and the service curve. From Proposition 2.2 it follows that it is enough to consider only service curves with a value of $T \geq \frac{\sigma_1}{C}$ given that smaller values of T would give a bigger delay bound.

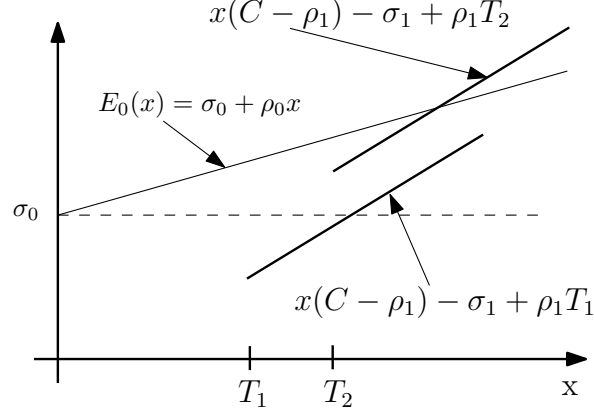


Figure 2.5: Maximum horizontal distance between E_0 and S_0^T

Assuming that the system is stable (i.e. $\rho_0 + \rho_1 \leq C$) it is easy to see that the maximum horizontal distance between the input envelope and the service curve is the biggest between T and the x -value where $S(x) = \sigma_0$. That is $x(C - \rho_1) - \sigma_1 + \rho_1 T = \sigma_0$. Which implies $x = \frac{\sigma_0 + \sigma_1 - \rho_1 T}{C - \rho_1}$. In one formula:

$$\hat{d} = \max \left\{ T, \frac{\sigma_0 + \sigma_1 - \rho_1 T}{C - \rho_1} \right\}. \quad (2.4)$$

We need to take the maximum between the x -value and T because if $T > \frac{\sigma_0 + \sigma_1 - \rho_1 T}{C - \rho_1}$ then the maximum distance is not the x -value but T . Figure 2.5 shows the input envelope and two sample service curves. Note how for the second one the maximum horizontal distance is T_2 and not the x -value.

Of the two arguments of the maximum in (2.4), the first one is an increasing function of T while the second one is a decreasing function of T . Therefore the smallest possible value for \hat{d} is obtained when

$$T = \frac{\sigma_0 + \sigma_1 - \rho_1 T}{C - \rho_1},$$

so that

$$T = \frac{\sigma_0 + \sigma_1}{C}.$$

Figure 2.6 shows the service curve $S_0^T(x)$ when $T = \frac{\sigma_0 + \sigma_1}{C}$. Obviously in this case the worst case delay bound is $\hat{d} = \frac{\sigma_0 + \sigma_1}{C}$ which is the same as in [14].

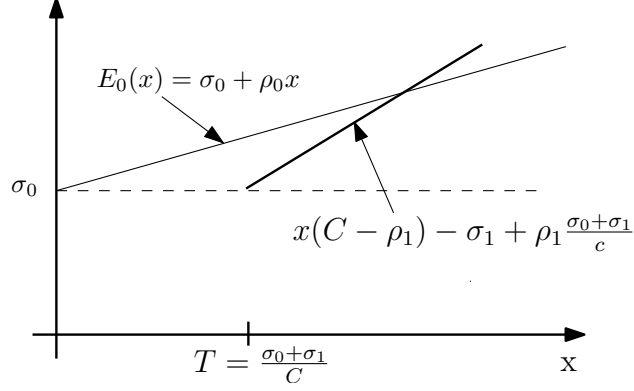


Figure 2.6: Service curve when $T = \frac{\sigma_0 + \sigma_1}{C}$

2.2.2 Output Envelope

Using Proposition 8 of [1] we can calculate the output envelope for flow 0 as:

$$E_{0,\text{out}} = (E_{0,\text{in}} * \bar{S}) \oslash S$$

where $E_{0,\text{in}}$ is the input envelope, \bar{S} is the maximum service curve and S is the minimum service curve guaranteed to flow 0. In our case $E_{0,\text{in}}(x) = \rho_0 x + \sigma_0$ and $\bar{S}(x) = Cx$ as the server operates at rate C . Therefore for any t, τ such that $t > \tau$ $R_{0,\text{out}}(t) - R_{0,\text{out}}(\tau) \leq C(t - \tau)$. Similarly to what we did for the maximum delay we can use the fact that the deconvolution is such that if $f \leq g$ then $h \oslash f \geq h \oslash g$ (see, for example, Theorem 3.1.12 in [23]). Combining this with Proposition 2.2 we have that $(E_{0,\text{in}} * \bar{S}) \oslash S_0^{T_1} \geq (E_{0,\text{in}} * \bar{S}) \oslash S_0^{T_2}$ if $T_1 < \frac{\sigma_1}{C} = T_2$. Given that a smaller envelope is preferable we can concentrate our attention on the case when $T \geq \frac{\sigma_1}{C}$ so that the minimum service curve S is the same as in (2.3).

First we need to calculate $E_{0,\text{in}} * \bar{S}$. Taking advantage of the fact that both functions are concave for $x \geq 0$. We can use Theorem 3.1.6 from [23] and define $F(x)$ as:

$$F(x) = (E_{0,\text{in}} * \bar{S}) = \min\{Cx, \rho_0 x + \sigma_0\}.$$

By the definition of deconvolution it follows that

$$E_{0,\text{out}} = \sup_{\tau} \{F(t + \tau) - S_0^T(\tau)\}.$$

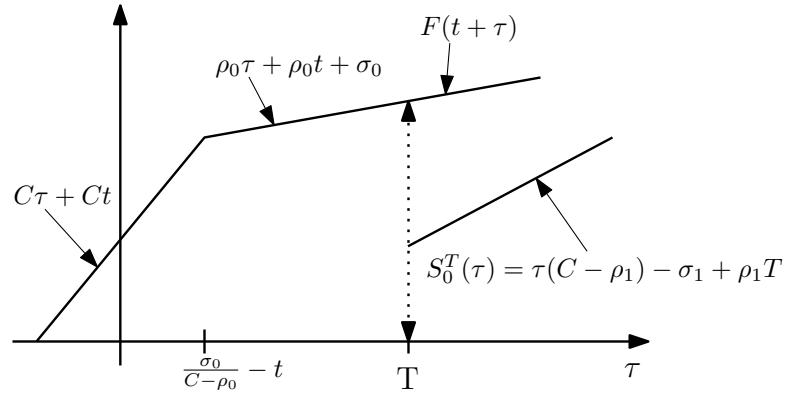


Figure 2.7: $F(t + \tau)$ and $S_0^T(\tau)$ when $T \geq \frac{\sigma_1}{C}$ and $\frac{\sigma_0}{C - \rho_0} - t < T$

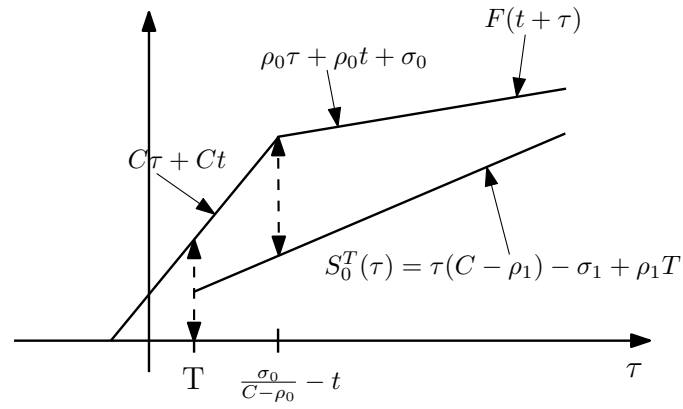


Figure 2.8: $F(t + \tau)$ and $S_0^T(\tau)$ when $T \geq \frac{\sigma_1}{C}$ and $\frac{\sigma_0}{C - \rho_0} - t \geq T$

From Section 2.2 we know that $S_1^T()$ has different forms depending on whether $T > \frac{\sigma_1}{C}$ or $T \geq \frac{\sigma_1}{C}$. First we are going to consider the case where $T \geq \frac{\sigma_1}{C}$.

To calculate the supremum we have to further consider two cases, the first one when $\frac{\sigma_0}{C-\rho_0} - t < T$. Figure 2.7 shows the graph of $F(t+\tau)$ and $S_0^T(\tau)$. In this case $t > \frac{\sigma_0}{C-\rho_0} - T$ and the supremum of the difference between $F(t+\tau)$ and $S_0^T(\tau)$ is $F(t+T)$. To see why this is true consider that $F(t+\tau)$ is an increasing function, $S_0^T(\tau) = 0$ for $\tau < T$ and that the slope of $S_0^T(\tau)$ is $C - \rho_1$ for $\tau \geq T$ which is smaller than the slope of $F(t+\tau)$ which is ρ_0 (recall that we are assuming a stable system, i.e. $C \geq \rho_0 + \rho_1$). Therefore in this first case we have:

$$\sup_{\tau} \{F(t+\tau) - S_0^T(\tau)\} = F(t+T) = \rho_0 t + \sigma_0 + \rho_0 T. \quad (2.5)$$

The second case is when $\frac{\sigma_0}{C-\rho_0} - t > T$. Figure 2.8 shows the graph of $F(t+\tau)$ and $S_0^T(\tau)$. In this case $t \leq \frac{\sigma_0}{C-\rho_0} - T$ and the supremum is achieved either for $\tau = T$ or for $\tau = \frac{\sigma_0}{C-\rho_0} - t$. To see why this is true consider that for $\tau \leq T$ we have $S(\tau) = 0$ and that $F(t+\tau)$ is an increasing function. These facts imply that the supremum of the difference between these two functions when $\tau \leq T$ is achieved when $\tau = T$. On the other hand, if $T \leq \tau \leq \frac{\sigma_0}{C-\rho_0} - t$ the slope of $F(t+\tau)$ is C , which is greater than the slope of $S(\tau)$ which is $C - \rho_1$ (again we are assuming a stable system). For $\tau \geq \frac{\sigma_0}{C-\rho_0} - t$ the opposite is true given that the slope of F is ρ_0 and the slope of S is still $C - \rho_1$. So that we have:

$$\begin{aligned} \sup_{\tau} \{F(t+\tau) - S_0^T(\tau)\} &= \max \left\{ F(T) - S(T), F\left(\frac{\sigma_0}{C-\rho_0} - t\right) - S\left(\frac{\sigma_0}{C-\rho_0} - t\right) \right\} \\ &= \max \left\{ CT + Ct, t(C - \rho_1) + \sigma_1 - \rho_1 T + \frac{\rho_1 \sigma_0}{C - \rho_0} \right\} \\ &= \begin{cases} t(C - \rho_1) + \sigma_1 - \rho_1 T + \frac{\rho_1 \sigma_0}{C - \rho_0}, & \text{if } t \leq t_1 \\ CT + Ct, & \text{if } t_1 \leq t \leq \frac{\sigma_0}{C-\rho_0} - T \end{cases} \end{aligned} \quad (2.6)$$

where $t_1 = \frac{\sigma_1}{\rho_1} + \frac{\sigma_0}{C-\rho_0} - T(1 + \frac{C}{\rho_1})$.

Note that if $\frac{\sigma_1}{C} \geq \frac{\sigma_0}{C-\rho_0}$ the second case is not possible (recall that $T_1 \geq \frac{\sigma_1}{C}$). In this case from (2.5) we have:

$$E_{0,\text{out}}(t) = \begin{cases} 0, & \text{if } t < 0 \\ \rho_0 t + \sigma_0 + \rho_0 T, & \text{if } t \geq 0, \end{cases}$$

so that the bigger is T the bigger the output envelope. Given that a smaller output envelope gives a better (smaller) output bound, we would like T to be as small as possible,

that is $T = \frac{\sigma_1}{C}$. In this case the previous expression becomes:

$$E_{0,\text{out}}(t) = \begin{cases} 0, & \text{if } t < 0 \\ \rho_0 t + \sigma_0 + \rho_0 \frac{\sigma_1}{C}, & \text{if } t \geq 0. \end{cases}$$

If $\frac{\sigma_1}{C} < \frac{\sigma_0}{C-\rho_0}$ then both cases are possible and the output envelope is:

$$E_{0,\text{out}}(t) = \begin{cases} 0, & \text{if } t < 0 \\ t(C - \rho_1) + \sigma_1 - \rho_1 T + \frac{\rho_1 \sigma_0}{C - \rho_0}, & \text{if } 0 \leq t \leq t_1 \\ CT + Ct, & \text{if } t_1 \leq t \leq \frac{\sigma_0}{C - \rho_0} - T \\ \rho_0 t + \sigma_0 + \rho_0 T, & \text{if } t \geq \frac{\sigma_0}{C - \rho_0} - T. \end{cases} \quad (2.7)$$

Note that the final part of the envelope (when $t \geq \frac{\sigma_0}{C - \rho_0} - T$) is an increasing function of T . That is greater values of T will give bigger and hence less desirable envelopes. Just as in the previous case this means that $T = \frac{\sigma_1}{C}$ gives the smallest envelope. When $T = \frac{\sigma_1}{C}$ we have $t_1 = \frac{\sigma_0}{C - \rho_0} - \frac{\sigma_1}{C}$ and:

$$E_{0,\text{out}}(t) = \begin{cases} 0, & \text{if } t < 0 \\ t(C - \rho_1) + \sigma_1 - \rho_1 \frac{\sigma_1}{C} + \frac{\rho_1 \sigma_0}{C - \rho_0}, & \text{if } 0 \leq t \leq \frac{\sigma_0}{C - \rho_0} - \frac{\sigma_1}{C} \\ \rho_0 t + \sigma_0 + \rho_0 \frac{\sigma_1}{C}, & \text{if } t \geq \frac{\sigma_0}{C - \rho_0} - \frac{\sigma_1}{C}. \end{cases} \quad (2.8)$$

Using the same argument as in [14] we can take the minimum between (2.8) and Ct given that the output of a FIFO server operating at rate C can grow only as fast as Ct . Using this extra constraint we have:

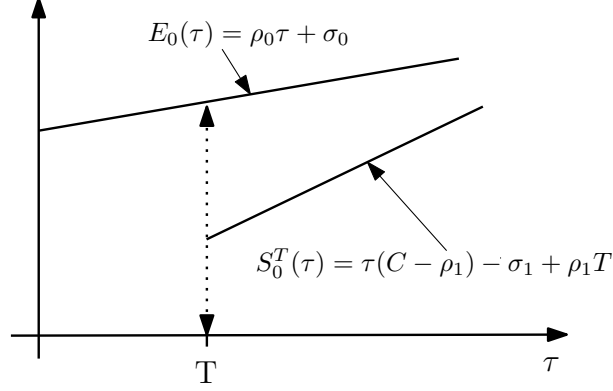
$$E_{0,\text{out}}(t) = \begin{cases} Ct, & \text{if } 0 \leq t \leq \frac{C\sigma_0 + \rho_0 \sigma_1}{C(C - \rho_0)} \\ \rho_0 t + \sigma_0 + \frac{\rho_0 \sigma_1}{C}, & \text{if } t > \frac{C\sigma_0 + \rho_0 \sigma_1}{C(C - \rho_0)}, \end{cases}$$

which is the same as in [14]. Note that this way, we do not have to worry about the fact that the second case in (2.7) is inversely proportional to T so that, at least for this case, we would like T to be as big as possible. On the other hand in the other cases we would like T to be as small as possible. But these cases are never used in the final solution.

2.2.3 Maximum Backlog

Using Proposition 7 from [2] we can compute an upper bound for the backlog for flow 0 as:

$$b_{0,\text{max}} = \sup \{E_0(\tau) - S_0^T(\tau)\}.$$

Figure 2.9: $E_0(\tau)$ and $S_0^T(\tau)$

As in the case of the output envelope we can use Proposition 2.2 and Theorem 3.1.12 in [23] to show that we only need to consider the case when $T \geq \frac{\sigma_1}{C}$.

Figure 2.9 shows the graph of $E_0(\tau)$ and $S_0^T(\tau)$: the supremum of the difference of the two functions is $E_0(T)$ given that for $x \leq T$, $S_0^T(x) = 0$. On the other hand for $x > T$ the slope of $S_0^T(x)$ is $C - \rho_1$ which is greater than ρ_0 (the slope of E_0) by the stability assumption. Therefore we have:

$$b_{0,\max} = \rho_1 T + \sigma_1 ,$$

which is an increasing function of T so the smallest possible bound is when $T = \frac{\sigma_2}{C}$. In this case:

$$b_{0,\max} = \frac{\rho_1 \sigma_2}{C} + \sigma_1 .$$

2.3 Delay Bounds for two FIFO Servers in Tandem

In this section we are going to consider a very simple network of two FIFO queues in tandem where all the inputs have sigma-rho envelopes. We are interested in the worst case delay for a flow that goes through both nodes. One possible solution is to consider each node in isolation and use the bounds derived in the previous sections. For the first server we know all the input envelopes so we can find the worst case delay for the first hop and we can also find the output envelope for the flow that goes on to the second server. Using this envelope and the one for the traffic that enters the network at the second hop

we can find the worst case delay at this hop and then add the two worst case delays (one for each hop). If we could find an arrival pattern that does achieve this end-to-end delay bound we would be done. Unfortunately nobody has ever been able to find such an arrival pattern. Of course this does not mean that such a pattern does not exist but it leads us to wonder about the existence of tighter bounds.

Another possible approach is to use the service curves presented at the beginning of this chapter. It is known (for example, see Proposition 10 in [2]) that if the same flow goes through two network elements, guaranteeing service curves S_1 and S_2 , respectively, then the entire network can be modeled as a single node guaranteeing the service curve obtained by convolving the service curves at each hop: $S_1 * S_2$. It is often the case that the delay bounds derived using the single service curve obtained by the convolution are smaller than the same bounds derived by considering each node in isolation.

In our example the problem is slightly more complicated because each node offers an infinite number of service curves, each one characterized by a different value of the parameter T . In order to find the smallest possible delay bound we will need to find the optimal value for the parameter T for both servers.

At this point it is useful to introduce some definitions and give a precise description of the problem we are addressing. Consider a simple two node system, as the one in Figure 2.10. Two flows enter the first server: flows 0 and 1. Flow 0 then goes to the second server while flow 1 goes to another server, so that flow 1 can be viewed as the “cross traffic” at the first server. Similarly at the second server flow 2 is the cross traffic.

Given that each flow is present at multiple points in the system (input and output links) we will use the notation R_{ij} where i is the “name” of the flow. That is a number uniquely identifying each flow and where j is the number of hops that flow i has gone through, so R_{00} represents flow 0 at the input link of the first node that it goes through and R_{01} represents flow 0 at the output of the first node it uses. Note that this flow is also the input flow for the next node. Similarly R_{02} represents flow 0 at the output of the second node that it uses. It is also assumed that each input flow has a sigma-rho envelope, that is for every $t \geq s$ $R_i(t_1, t_2) \leq \sigma_i + \rho_i(t - s)$, we will use E_i to indicate the envelope of the i -th input flow.

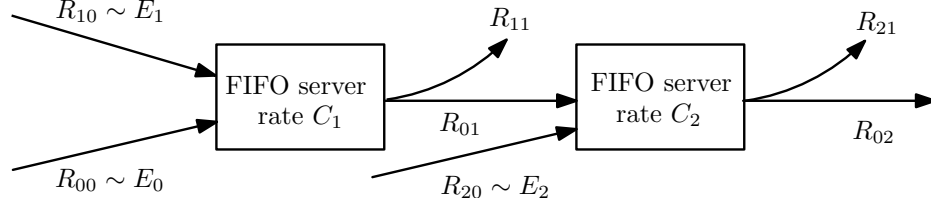


Figure 2.10: A simple two node system

2.3.1 Considering Each Node in Isolation

For the first node we have seen in the previous section that the worst case delay bound is:

$$\hat{d}_1 = \frac{\sigma_0 + \sigma_1}{C_1} \quad (2.9)$$

(again this is the same whether we use the services curves or the results in [14]). Before we can analyze the second node we need to find the output envelope for flow 0. Using Theorem 4.4 from [14] it is easy to see that:

$$E_{01}(t) = \begin{cases} C_1 t, & \text{if } 0 \leq t \leq \frac{C_1 \sigma_0 + \rho_0 \sigma_1}{C_1(C_1 - \rho_0)} \\ \rho_0 t + \sigma_0 + \frac{\rho_0 \sigma_1}{C_1} & \text{if } t > \frac{C_1 \sigma_0 + \rho_0 \sigma_1}{C_1(C_1 - \rho_0)}. \end{cases}$$

Combining this with the fact that $E_{20}(t) = \sigma_2 + \rho_2 t$ we have that:

$$\hat{d}_2 = \begin{cases} \frac{\sigma_2}{C_2}, & \text{if } C_1 + \rho_2 \leq C_2 \\ \frac{\sigma_2}{C_2} + \frac{(C_1 + \rho_2 - C_2)(C_1 \sigma_0 + \rho_0 \sigma_1)}{C_1 C_2 (C_1 - \rho_0)}, & \text{if } C_1 + \rho_2 > C_2. \end{cases} \quad (2.10)$$

Adding (2.10) and (2.9) we have:

$$\hat{d}_{\text{sn}} = \begin{cases} \frac{\sigma_0 + \sigma_1}{C_1} + \frac{\sigma_2}{C_2}, & \text{if } C_1 + \rho_2 \leq C_2 \\ \frac{\sigma_0 + \sigma_1}{C_1} + \frac{\sigma_2}{C_2} + \frac{(C_1 + \rho_2 - C_2)(C_1 \sigma_0 + \rho_0 \sigma_1)}{C_1 C_2 (C_1 - \rho_0)}, & \text{if } C_1 + \rho_2 > C_2 \end{cases} \quad (2.11)$$

where the subscript sn stands for Single Nodes.

2.3.2 Convolution of two service curves

From section 2.2 we know that each node in Figure 2.10 can be characterized with a service curve $S_i^T(x)$ (defined in (2.3)) characterized by the parameters $a_i = T_i C_i - \sigma_i$ and $m_i = C_i - \rho_i$, $i = 1, 2$. Also, from Proposition 2.2 we know that we can limit our attention to the case where $T_i \geq \frac{\sigma_i}{C_i}$.

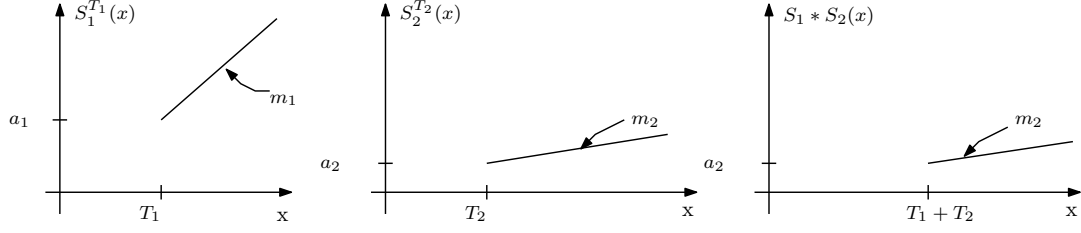


Figure 2.11: Convolution: Case 1

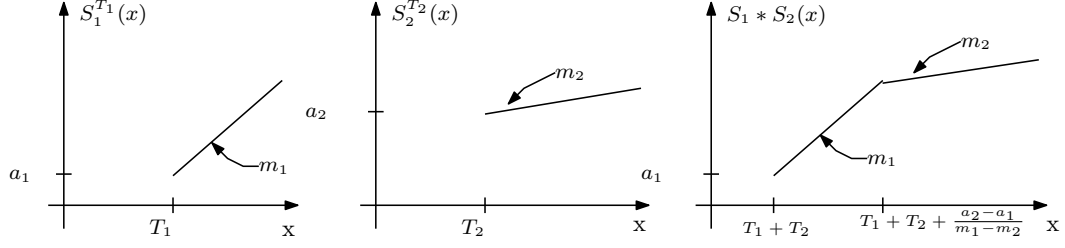


Figure 2.12: Convolution Case 2

As we have mentioned, it is known that the end to end service curve offered to flow 0 is the (min-plus) convolution of the service curve for each node. As discussed above, the service curve for each server depends on the parameter T , which can take any value greater than $\frac{\sigma_i}{C}$, where σ_i is the burstiness parameter for the cross traffic.

In this section we are going to compute the convolution of the two service curves. There are four possible cases depending on the relationship between a_1 , a_2 , m_1 and m_2 :

1. $a_1 \geq a_2$ and $m_1 \geq m_2$;
2. $a_1 < a_2$ and $m_1 \geq m_2$;
3. $a_1 \geq a_2$ and $m_1 < m_2$;
4. $a_1 < a_2$ and $m_1 < m_2$;

Note that, as the convolution is commutative, only the first two cases need to be examined because cases 3 and 4 can be reduced to cases 1 and 2 by exchanging the two service curves. Also note that when $a_1 = a_2$ Cases 1 and 2 are identical therefore the equal sign can put in either one or both.

Claim 2.3. For Case 1 the convolution of the two service curves is:

$$S_1 * S_2(x) = \begin{cases} 0, & \text{if } x < T_1 + T_2 \\ a_2 + m_2(x - T_1 - T_2), & \text{if } x \geq T_1 + T_2, \end{cases} \quad (2.12)$$

and for Case 2 is:

$$S_1 * S_2(x) = \begin{cases} 0, & \text{if } x < T_1 + T_2 \\ a_1 + m_1(x - T_1 - T_2), & \text{if } T_1 + T_2 \leq x \leq T_1 + T_2 + B \\ a_2 + m_2(x - T_1 - T_2), & \text{if } x \geq T_1 + T_2 + B \end{cases} \quad (2.13)$$

where $B = \frac{a_2 - a_1}{m_1 - m_2}$, and $(i = 1, 2)$

$$S_i = \begin{cases} 0, & \text{if } x < T_i \\ a_i + m_i(x - T_i), & \text{if } x \geq T_i. \end{cases}$$

Figures 2.11 and 2.12 show the service curves for the two different cases. Note that for $a_1 = a_2$ the service curves in (2.12) and (2.13) are equal.

Proof. Recall that the definition of convolution of $f(t)$ and $g(t)$ is:

$$f * g(t) = \inf_{0 \leq s \leq t} \{f(t - s) + g(s)\}.$$

As noted above this operator is commutative therefore, without loss of generality, we can choose $S_1 = f$:

$$S_1(t - s) = \begin{cases} a_1 + m_1(t - s - T_1), & \text{if } s \leq t - T_1 \\ 0, & \text{if } s > t - T_1, \end{cases}$$

and this holds for both cases. The rest of the proof is different depending on whether we are in Case 1 or 2.

Suppose we are in Case 1 (i.e. $a_1 \geq a_2$ and $m_1 \geq m_2$). To find the value of $\inf\{S_1(t - s) + S_2(s)\}$ we have two different sub-cases depending on the value of t (see Figure 2.13):

1. $0 \leq t < T_1 + T_2$: (i.e. $t - T_1 < T_2$) in this case the infimum is 0.

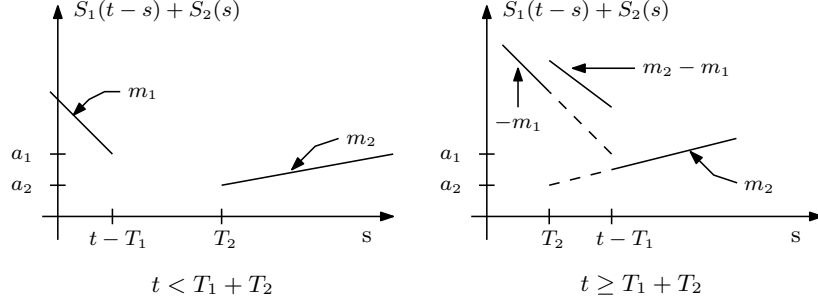


Figure 2.13: Argument of the infimum

2. $t \geq T_1 + T_2$: (i.e. $t - T_1 \geq T_2$) let $g(s) = S_1(t - s) + S_2(s)$ then:

$$g(s) = \begin{cases} a_1 + m_1(t - s - T_1), & \text{if } 0 \leq s < T_2 \\ s(m_2 - m_1) + a_1 + m_1(t - T_1) + a_2 - m_2 T_2, & \text{if } T_2 \leq s \leq t - T_1 \\ a_2 + m_2(s - T_2), & \text{if } s > t - T_1. \end{cases} \quad (2.14)$$

Note that for $0 \leq s < t - T_1$ $g(s)$ is a decreasing function therefore the infimum over this range will be achieved for $s = T_2^-$. On the other hand for $s > t - T_1$ $g(s)$ is increasing. Therefore the infimum, in this range, will be achieved for $s = t - T_1^+$. At the same time $g(T_2^+) \geq g(T_2^-)$, $g(t - T_1^-) \geq g(t - T_1^+)$ and $g(s)$ is non increasing (as $m_1 \geq m_2$) for $T_2 \leq s \leq t - T_1$. Therefore the infimum is achieved either at $s = T_2^-$ or at $s = t - T_1^+$. On the other hand we have:

$$\begin{aligned} g(T_2^-) &= a_1 + m_1(t - T_2 - T_1) \\ &\geq a_2 + m_2(t - T_2 - T_1) \\ &= g(t - T_1^+), \end{aligned}$$

(recall that in this case $a_1 \geq a_2$ and $m_1 \geq m_2$). Hence

$$\begin{aligned} \inf_{0 \leq s \leq t} \{S_1(t - s) + S_2(s)\} &= g(t - T_1^+) \\ &= a_2 + m_2(t - T_2 - T_1), \end{aligned}$$

and combining the two sub-cases we have (2.12).

Now suppose we are in Case 2 (i.e. $a_1 < a_2$ and $m_1 \geq m_2$). Again depending on the value of t we three sub-cases:

1. $0 \leq t < T_1 + T_2$: (i.e. $t - T_1 < T_2$) In this case the infimum is 0.
2. $T_1 + T_2 \leq t \leq T_1 + T_2 + B$, where $B = \frac{a_2 - a_1}{m_1 - m_2}$: (in this case $t - T_1 \geq T_2$) let $g(s)$ be defined as in (2.14). The same argument shows that again the infimum is achieved either at $s = T_2^-$ or at $s = t - T_1^+$. In this case $a_1 < a_2$ and $t \leq T_1 + T_2 + B$. Therefore:

$$\begin{aligned}
g(T_2^-) &= a_1 + m_1(t - T_2 - T_1) \\
&\leq a_2 + m_2(t - T_2 - T_1) \\
&= g(t - T_1^+),
\end{aligned}$$

and

$$\begin{aligned}
\inf_{0 \leq s \leq t} \{S_1(t - s) + S_2(s)\} &= g(T_2^-) \\
&= a_1 + m_1(t - T_2 - T_1).
\end{aligned}$$

3. $t \geq T_1 + T_2 + B$: Again let $g(s)$ be defined as in (2.14). The same argument shows that the infimum is achieved either at $s = T_2^-$ or at $s = t - T_1^+$. In this case $a_1 < a_2$ and $t \geq T_1 + T_2 + B$ therefore:

$$\begin{aligned}
g(T_2^-) &= a_1 + m_1(t - T_2 - T_1) \\
&\geq a_2 + m_2(t - T_2 - T_1) \\
&= g(t - T_1^+),
\end{aligned}$$

and

$$\begin{aligned}
\inf_{0 \leq s \leq t} \{S_1(t - s) + S_2(s)\} &= g(t - T_1^+) \\
&= a_2 + m_2(t - T_2 - T_1).
\end{aligned}$$

Combining these three sub-cases we have (2.13). □

2.3.3 Optimal Values of T_1 and T_2 for Worst Case Delay Computation

Given a service curve $S(x)$ and an input envelope $E_0(x)$ the worst case delay \hat{d} is the “minimum horizontal distance” between the two functions, formally:

$$\hat{d} = \inf \{ \Delta : \Delta \geq 0 \text{ and } S(x) \geq E_0 * \delta_\Delta(x) \forall x \geq 0 \}.$$

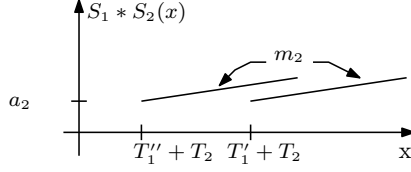


Figure 2.14: Two service curves for Case 1

In this case $S_{\text{net}}(x) = S_1(x) * S_2(x)$ is the service curve offered to flow 0 by the network comprised of the two FIFO queues with cross traffic R_1 and R_2 . As discussed in the previous section, the convolution of the two service curves depends on the parameters T_1 and T_2 that can take any non negative value. Therefore different values of T_1 and T_2 will result in different worst case delay bounds. In this section we are going to find the values that give the smallest bound. Note that $m_i = C_i - \rho_i$ does not depend on T_i while $a_i = T_i C_i - \sigma_i$ does, so that by changing the value of T_i one can only change the value of a_i but not that of m_i .

Now assume, without loss of generality, that $m_1 \geq m_2$. If this is not true we can switch S_1 and S_2 given that the convolution is commutative. From claim 2.3 we know that the service curve is represented by different expressions depending on the values of a_i . We consider each case separately.

1. First consider the case where $a_1 \geq a_2$. In this case we have:

$$T_1 \geq T_2 \frac{C_2}{C_1} + \frac{\sigma_1 - \sigma_2}{C_1},$$

and that the S_{net} is the same as in (2.12). Note that (2.12) is a decreasing function of T_1 . Therefore if T_1' and T_1'' are such that $T_1'' \leq T_1'$ then $S_1^{T_1'} \leq S_1^{T_1''}$. If we let $S'_{\text{net}} = S_1^{T_1'} * S_2^{T_2}$ and $S''_{\text{net}} = S_1^{T_1''} * S_2^{T_2}$ then $S_1^{T_1'} \leq S_1^{T_1''}$ implies $S'_{\text{net}} \leq S''_{\text{net}}$ so that the delay bound obtained by using S''_{net} is smaller (better) than the one obtained by using S'_{net} (recall that the delay bound is the maximum horizontal distance between E_0 and S_{net} , see Figure 2.14).

Therefore for any (fixed) value of T_2 a smaller value of T_1 will give a smaller delay bound. In this case T_1 has to be at least $T_2 \frac{C_2}{C_1} + \frac{\sigma_1 - \sigma_2}{C_1}$ so if we pick $T_1'' = T_2 \frac{C_2}{C_1} + \frac{\sigma_1 - \sigma_2}{C_1}$ we will have the best possible bound for the case where $a_1 \geq a_2$. Now suppose we choose a particular value for T_1 , call it T_1' , such that $T_1' > T_2 \frac{C_2}{C_1} + \frac{\sigma_1 - \sigma_2}{C_1}$. It is then possible to find a “better” service curve if we choose $T_1 = T_1'' = T_2 \frac{C_2}{C_1} + \frac{\sigma_1 - \sigma_2}{C_1}$, which

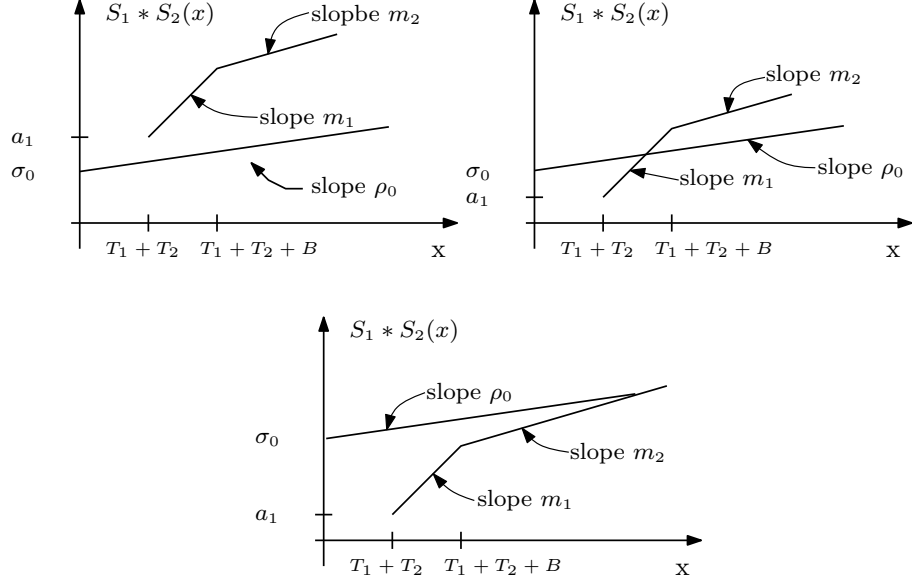


Figure 2.15: Sub-cases for Case 2

is equivalent to $a_1 = a_2$. As noted above, when $a_1 = a_2$ cases 1 and 2 have the same service curve, that is $a_1 = a_2$ is the “boundary” between the two cases and it belongs to both. Therefore it suffices to consider only Case 2, as the optimal value of \hat{d} for Case 2 will not be bigger than the one for Case 1.

2. Now consider the case where $a_1 \leq a_2$, Figure 2.15 shows the three possible sub-cases depending on the values of a_1 and c where $c = (S_1 * S_2)(T_1 + T_2 + B) = a_1 + m_1 B = \frac{m_1 a_2 - m_2 a_1}{m_1 - m_2}$. Note that $c \geq a_1$ (as $m_1 \geq 0$).

- (a) $a_1 \geq \sigma_0$: That is $T_1 C_1 - \sigma_1 \geq \sigma_0$. In this sub-case $\hat{d} = T_1 + T_2$ and hence the minimum in this sub-case is achieved when T_1 and T_2 are as small as possible. In order to be in this case $a_1 = T_1 C_1 - \sigma_1 \geq \sigma_0$, so that $T_1 \geq \frac{\sigma_1 + \sigma_0}{C_1}$. Similarly for T_2 we have that in order to be in this case $a_2 \geq a_1$ and therefore $a_2 = T_2 C_2 - \sigma_2 \geq T_1 C_1 - \sigma_1 = a_1$. Combining this with the fact that $T_1 \geq \frac{\sigma_1 + \sigma_0}{C_1}$ we have $T_2 \geq \frac{\sigma_2 + \sigma_0}{C_2}$. Hence in this sub-case the minimum value of \hat{d} is achieved for $T_1 = \frac{\sigma_0 + \sigma_1}{C_1}$ and $T_2 = \frac{\sigma_0 + \sigma_2}{C_2}$ so that

$$\hat{d} = \frac{\sigma_0 + \sigma_1}{C_1} + \frac{\sigma_0 + \sigma_2}{C_2}. \quad (2.15)$$

(b) $c \geq \sigma_0 \geq a_1$: In this sub-case

$$\begin{aligned}\hat{d} &= (S_1 * S_2)^{-1}(\sigma_0) \\ &= T_1 + T_2 + \frac{\sigma_0 - a_1}{m_1} \\ &= T_1 \left(-\frac{\rho_1}{m_1} \right) + T_2 + \frac{\sigma_0 + \sigma_1}{m_1}.\end{aligned}\tag{2.16}$$

The coefficient of T_1 is negative, while the coefficient of T_2 is positive, therefore smaller values of T_2 give better bounds. At the same time in this case $c \geq \sigma_0$, that is $a_2 \geq \frac{m_2}{m_1}a_1 + \sigma_0 - \frac{m_2}{m_1}\sigma_0$ and (using the definitions of a_1 and a_2):

$$T_2 \geq \frac{m_2 C_1}{m_1 C_2} T_1 + \frac{\sigma_0 + \sigma_2}{C_2} - \frac{m_2}{m_1} \left(\frac{\sigma_0 + \sigma_1}{C_2} \right).\tag{2.17}$$

Combining (2.16) and (2.17) we have:

$$\hat{d} \geq T_1 \left(\frac{m_2 C_1}{m_1 C_2} - \frac{\rho_1}{m_1} \right) + \frac{\sigma_0 + \sigma_2}{C_2} - \frac{m_2}{m_1} \left(\frac{\sigma_0 + \sigma_1}{C_2} \right) + \frac{\sigma_0 + \sigma_1}{m_1}.\tag{2.18}$$

This bound depends only on T_1 and not on T_2 , but the coefficient of T_1 can be either positive or negative. If $\frac{m_2 C_1 - \rho_1 C_2}{m_1 C_2} > 0$ the minimum value of \hat{d} is achieved when a_1 is as small as possible that is for $a_1 = 0$. If $a_1 = 0$ we have $T_1 = \frac{\sigma_1}{C_1}$ and substituting $T_1 = \frac{\sigma_1}{C_1}$ in (2.18) we have:

$$\hat{d} = \frac{\sigma_1}{C_1} + \frac{\sigma_0 + \sigma_2}{C_2} + \frac{\sigma_0 \rho_2}{C_2(C_1 - \rho_1)}.\tag{2.19}$$

On the other hand if $\frac{m_2 C_1 - \rho_1 C_2}{m_1 C_2} < 0$ the minimum value of \hat{d} is achieved for $a_1 = \sigma_0$, that is $T_1 = \frac{\sigma_1 + \sigma_0}{C_1}$. Again substituting $T_1 = \frac{\sigma_1 + \sigma_0}{C_1}$ in (2.18) we have:

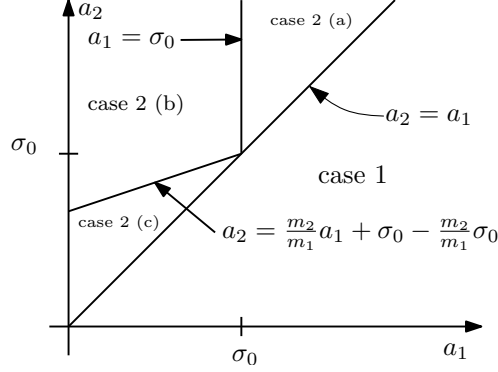
$$\hat{d} = \frac{\sigma_0 + \sigma_1}{C_1} + \frac{\sigma_0 + \sigma_2}{C_2}\tag{2.20}$$

which is the same value obtained in the previous sub-case. If $\frac{m_2 C_1 - \rho_1 C_2}{m_1 C_2} = 0$ we have that:

$$\hat{d} = \frac{\sigma_0 + \sigma_2}{C_2} + \frac{\rho_2(\sigma_0 + \sigma_1)}{m_1}.$$

Combining (2.19) and (2.20) we have that in this sub-case (provided $\frac{m_2 C_1 - \rho_1 C_2}{m_1 C_2} \neq 0$):

$$\hat{d} = \min \left\{ \frac{\sigma_1}{C_1} + \frac{\sigma_0 + \sigma_2}{C_2} + \frac{\sigma_0 \rho_2}{C_2(C_1 - \rho_1)}, \frac{\sigma_0 + \sigma_1}{C_1} + \frac{\sigma_0 + \sigma_2}{C_2} \right\}.$$

Figure 2.16: The different cases in the a_1, a_2 plane

(c) $\sigma_0 \geq c \geq a_1$: In this sub-case

$$\begin{aligned}
 \hat{d} &= (S_1 * S_2)^{-1}(\sigma_0) \\
 &= T_1 + T_2 + B + \frac{\sigma_0 - c}{m_2} \\
 &= T_1 + T_2 \left(-\frac{\rho_2}{m_2} \right) + \frac{\sigma_0 + \sigma_2}{m_2}. \tag{2.21}
 \end{aligned}$$

The coefficient of T_1 is positive while the coefficient of T_2 is negative, therefore bigger values of T_2 will give a better bound. At the same time we know that in this sub-case $c = \frac{m_1 a_2 - m_2 a_1}{m_1 - m_2} \leq \sigma_0$ so that

$$T_2 \leq \frac{m_2 C_1}{m_1 C_2} T_1 + \frac{\sigma_0 + \sigma_2}{C_2} - \frac{m_2}{m_1} \left(\frac{\sigma_0 + \sigma_1}{C_2} \right).$$

Combining this with (2.21) we have:

$$\begin{aligned}
 \hat{d} &\geq T_1 + \left[\frac{m_2 C_1}{m_1 C_2} T_1 + \frac{\sigma_0 + \sigma_2}{C_2} - \frac{m_2}{m_1} \left(\frac{\sigma_0 + \sigma_1}{C_2} \right) \right] \left(1 - \frac{C_2}{m_2} \right) + \frac{\sigma_0 + \sigma_2}{m_2} \\
 &= T_1 \left(\frac{m_2 C_1}{m_1 C_2} - \frac{\rho_1}{m_1} \right) + \frac{\sigma_0 + \sigma_2}{C_2} - \frac{m_2}{m_1} \left(\frac{\sigma_0 + \sigma_1}{C_2} \right) + \frac{\sigma_0 + \sigma_1}{m_1}. \tag{2.22}
 \end{aligned}$$

Note that (2.22) is the same as (2.18) but in this sub-case $0 \leq a_1 \leq c \leq \sigma_0$ while in the previous sub-case $0 \leq a_1 \leq \sigma_0$ (recall that $a_1 = T_1 C_1 - \sigma_1$), therefore the smallest possible value for (2.22) cannot be any smaller than the bounds obtained in the previous sub-case.

Finally note that Case 1 and the three sub-cases of Case 2 cover the entire a_1, a_2 plane (for $a_1 \geq 0$ and $a_2 \geq 0$), as shown in Figure 2.16, therefore we can conclude that

the optimal (smallest) delay bound, which we can obtain by using the service curves, is (assuming $m_1 \geq m_2$):

$$\hat{d}_{\text{opt}} = \min \left\{ \frac{\sigma_1}{C_1} + \frac{\sigma_0 + \sigma_2}{C_2} + \frac{\sigma_0 \rho_2}{C_2(C_1 - \rho_1)}, \frac{\sigma_0 + \sigma_1}{C_1} + \frac{\sigma_0 + \sigma_2}{C_2} \right\}. \quad (2.23)$$

If $m_1 < m_2$ the previous argument holds if we switch S_1 and S_2 (again recall that the convolution is commutative) and we have:

$$\hat{d}_{\text{opt}} = \min \left\{ \frac{\sigma_2}{C_2} + \frac{\sigma_0 + \sigma_1}{C_1} + \frac{\sigma_0 \rho_1}{C_1(C_2 - \rho_2)}, \frac{\sigma_0 + \sigma_1}{C_1} + \frac{\sigma_0 + \sigma_2}{C_2} \right\}. \quad (2.24)$$

We have compared the values of (2.23) and (2.24) with those of the same bounds obtained when the each node is considered in isolation (2.11) by computing their values in several specific instances: whenever $C_1 + \rho_2 \leq C_2$ (2.11) was smaller but if $C_1 + \rho_2 > C_2$ then (2.23) or (2.24) (depending on whether $m_1 \geq m_2$ in the specific case) were smaller than (2.11). We conjecture that this is true in general and that the service curves approach gives a smaller delay bound if $C_1 + \rho_2 > C_2$.

At the same time we were not able to find an arrival pattern that does achieve this smaller delay bound. This led us to look for a smaller bound. In the next chapter we will show that indeed there exist such a smaller (and achievable) delay bound for this simple network. This might be surprising given that in the first part of this chapter we showed that for a single node service curves do give tight delay and burstiness bounds for FIFO queues whose input flows have “sigma-rho” envelopes. But for two nodes the result in the next chapter demonstrates that the approach in this chapter fails to give tight bounds.

3

Service Mappings

3.1 Introduction

Over the past several years, a deterministic theory for analysis of networks of queues has been developed, now commonly termed as “network calculus.” We now briefly review some of the important developments in this area of research. The interested reader is referred to [9] and [23], which documents much of this work in detail.

Networks of queues operating with FIFO or fixed priority scheduling were analyzed by Cruz using a deterministic traffic model in [12, 14]. Subsequently, Parekh and Gallager [25, 26] analyzed networks of queues operating with generalized processor sharing using this deterministic traffic model, and thereby stimulated a large body of work by other researchers on “fair queuing” scheduling algorithms [31]. A key concept in Parekh and Gallager’s work was that of a “universal service curve.” This concept was extended to a *service abstraction* for general network elements by Cruz in [13]. This service model was refined independently by Le Boudec [22] and Sariowan [19]. Le Boudec formalized the notion of the convolution and deconvolution operators, and C. S. Chang [8] first observed that a service curve is analogous to an “impulse response” in the theory of linear time invariant systems. Chang exploited this analogy to provide simple explanations for results on traffic regulators developed in [14].

The insight obtained from these developments are apparent in a framework [5, 7] for deterministic quality of service guarantees proposed for the Internet by the *IntServ* work-

ing group of the Internet Engineering Task Force (IETF). In the IntServ model, traffic is managed on a per-flow basis, leading many researchers to question its capability to scale to large networks. Another working group of the IETF, *DiffServ*, has aimed to address scalability by developing a framework where traffic is managed at a coarser granularity than the level of flow [24]. This led to a renewed interest in performance results for FIFO queuing, since FIFO queues do not require per flow traffic management.

In [15], Cruz presented a service curve characterization for a FIFO multiplexer. In this chapter, we generalize and explore this characterization further. As we will see, this leads to a simplified understanding of previous results in network calculus, as well as the tightening of bounds for delay in networks of FIFO queues for deterministic traffic models.

The remainder of this chapter is organized as follows. In the next section, we define a general service abstraction for network elements. We derive performance bounds in the context of this service abstraction. These performance bounds have a simple unified graphical interpretation that we illustrate. We see that the service curve model is a special case of this this general service abstraction, and the performance bounds we obtain reduce to previously known results. In Section 3.4, we apply our service abstraction to the context of FIFO multiplexing. We will see in Section 3.4.1 that our performance bounds for a single FIFO multiplexer reduce to those tight bounds previously obtained in [14]. In Section 3.4.2 we apply the general service abstraction to analyze a tandem configuration of FIFO multiplexers. We will see that this results in improved bounds to end-to-end delay which are achievable.

3.2 Service Models

Consider a network element, which is an abstraction of a queuing system. For example, the queuing system might represent a packet switch, or more generally an entire network. In this chapter, a network element is an abstraction defined for the purposes of describing a single stream of information passing through the associated queuing system. Specifically, the network element has an arrival process and a departure process, described by two functions of time $R_{in}(\cdot)$ and $R_{out}(\cdot)$, respectively. The value of $R_{in}(t)$ is defined as the number of bits that have arrived to the network element up to time t , and similarly $R_{out}(t)$ is the number of bits that have departed the network element up to time t . The *backlog*

$B(t)$ of the network element at time t is defined as $B(t) = R_{\text{in}}(t) - R_{\text{out}}(t)$, i.e. it is the number of bits stored inside the network element at time t . The *virtual delay* of the network element at time t is defined as $D(t) = \inf\{d : d \geq 0 \text{ and } R_{\text{out}}(t + d) \geq R_{\text{in}}(t)\}$. For example, if the arriving bits depart the network element in first-in first-out (FIFO) order, then a bit that arrives at time t waits no longer than $D(t)$ seconds before departing the network element.

In general, the departure process $R_{\text{out}}(\cdot)$ may not be determined solely by the arrival process R_{in} , but also by external events in the associated queuing system. However, we can partially characterize the network element by bounding the departure process in terms of the arrival process. In general, we assume that the arrival process $R_{\text{in}}(\cdot)$ and departure process $R_{\text{out}}(\cdot)$ can be arbitrary non-decreasing functions. Formally, $R_{\text{in}}(\cdot)$ and $R_{\text{out}}(\cdot)$ are elements of \mathcal{M} , which is defined as the set of all non-decreasing functions whose domain is the set of all real numbers \mathcal{R} and whose range is the extended set of real numbers $\mathcal{R} \cup \{+\infty\}$.

3.2.1 Minimum Service Mappings

Let $\mathcal{S} : \mathcal{M} \rightarrow \mathcal{M}$ be a given operator, which maps elements of \mathcal{M} into elements of \mathcal{M} . Given functions of time $F(\cdot)$ and $G(\cdot)$, we use the notation $F \leq G$ if $F(t) \leq G(t)$ for all t . We say that \mathcal{S} is *monotone* if $F \leq G$ implies that $\mathcal{S}(F) \leq \mathcal{S}(G)$ for all F and G .

Definition 3.1. *Suppose a network element is such that $R_{\text{out}} \geq \mathcal{S}(R_{\text{in}})$ for all possible arrival processes R_{in} , for some monotone operator \mathcal{S} . In this case, say that \mathcal{S} is a (minimum) service mapping for the network element, and we write $R_{\text{in}} \rightarrow \mathcal{S} \rightarrow R_{\text{out}}$.*

The service model in the previous definition is composable in the sense that if several network elements are configured in tandem, each being described by a service mapping, then the composition of the service mappings is a service mapping of the entire system. This is stated formally in the following theorem for the case of two network elements in tandem.

Theorem 3.2. *[Network Elements in Tandem] Suppose $R_0 \rightarrow \mathcal{S}_1 \rightarrow R_1$ and $R_1 \rightarrow \mathcal{S}_2 \rightarrow R_2$. Then $R_0 \rightarrow (\mathcal{S}_1 \circ \mathcal{S}_2) \rightarrow R_2$, where $(\mathcal{S}_1 \circ \mathcal{S}_2)(F) = \mathcal{S}_2(\mathcal{S}_1(F))$.*

Proof. Fix any t . We have

$$\begin{aligned} R_2 &\geq \mathcal{S}_2(R_1) \\ &\geq \mathcal{S}_2(\mathcal{S}_1(R_0)) \\ &= (\mathcal{S}_1 \circ \mathcal{S}_2)(R_0) . \end{aligned}$$

The second inequality above follows since \mathcal{S}_2 is monotone. \square

3.2.2 Shift Invariant Service Mappings

An operator \mathcal{S} is said to be *time invariant* if $\mathcal{S}(F) = G$ implies that $\mathcal{S}(F_\Delta) = G_\Delta$ for all $\Delta \in \mathcal{R}$, where $F_\Delta(t) = F(t - \Delta)$ and $G_\Delta(t) = G(t - \Delta)$ for all t . An operator \mathcal{S} is said to be *space invariant* if $\mathcal{S}(F) = G$ implies that $\mathcal{S}(k + F) = k + G$ for all constants k . An operator \mathcal{S} that is both time invariant and space invariant is called *shift invariant*.

An operator \mathcal{S} is called *additive* if $\mathcal{S}(F_1) = G_1$ and $\mathcal{S}(F_2) = G_2$ imply that $\mathcal{S}(F_1 \wedge F_2) = G_1 \wedge G_2$, where we use the notation $H_1 \wedge H_2$ to denote the function defined by $(H_1 \wedge H_2)(t) = \min\{H_1(t), H_2(t)\}$. An operator that is both space invariant and additive is said to be *linear*.

As an example, suppose that $S(\cdot)$ is a minimum service curve [15, 19, 22], as we now define. The convolution of two functions $F(\cdot)$ and $G(\cdot)$, $F * G$, is first defined as

$$(F * G)(t) = \inf_{\tau} \{F(\tau) + G(t - \tau)\}$$

for all t . The function $S(\cdot)$ is said to be a (minimum) service curve for the network element if for any arrival process R_{in} we have $R_{\text{out}} \geq R_{\text{in}} * S$. This service model is special case of a service mapping where the associated operator is both linear and time invariant. Conversely, it can be seen that any service mapping that is both linear and time invariant can be equivalently be described in terms of a service curve.

Although the space of service models which are linear is adequate for analyzing many queuing systems of interest, in this chapter we make the proposition that non-linear service models are useful as well. In particular, we will demonstrate how they can be used to analyze networks of FIFO queues. For example, suppose that an arrival process R_{in} is multiplexed in a FIFO manner with another arrival process R_{in}^x , such that the aggregate arrival process $R_{\text{in}} + R_{\text{in}}^x$ results in the aggregate departure process $R_{\text{out}} + R_{\text{out}}^x$ with a

service curve of G , i.e. $(R_{\text{out}} + R_{\text{out}}^x)(t) \geq ((R_{\text{in}} + R_{\text{in}}^x) * G)(t)$ for all t . If $R_{\text{in}}^x \leq R_{\text{in}}^x * E^x$, it is known [15] that

$$R_{\text{out}} \geq R_{\text{in}} * S_T$$

for all $T \geq 0$, where

$$S_T(t) = \begin{cases} [G(t) - E^x(t - T)]^+ & \text{if } t \geq T \\ 0 & \text{otherwise.} \end{cases} \quad (3.1)$$

where we use the notation $x^+ = \max\{x, 0\}$. In this case, in fact we have $R_{\text{in}} \rightarrow \hat{\mathcal{S}} \rightarrow R_{\text{out}}$, where the operator $\hat{\mathcal{S}}$ is defined for $F \in \mathcal{M}$ as

$$\hat{\mathcal{S}}(F)(t) = \sup_{T: T \geq 0} [(F * S_T)(t)] . \quad (3.2)$$

It can be verified that $\hat{\mathcal{S}}$ is shift invariant, but not necessarily linear.

This motivates us to look at the class of service models characterized by shift invariant service mappings. For generality, we do not necessarily assume that service mappings are of the form given in (3.2). We shall obtain bounds on delay, backlog, and a traffic envelope (defined below) for the departure process, in the context of service models characterized by shift invariant service mappings.

To begin with, we first define the notion of a traffic envelope [14]. Given a function $E(\cdot)$, and an arrival process R , we say that R has envelope E if $R \leq R * E$. Note that the inequality $R \leq R * E$ is equivalent to

$$R(u) \geq R(t) - E(t - u) \text{ for all } u \quad (3.3)$$

for any fixed value of t . In other words, if R has envelope E , then for any fixed t we have

$$R \geq R_{E,t} , \quad (3.4)$$

where $R_{E,t}(u) = R(t) - E(t - u)$ for all u . Note that in general $E(x)$ may be non-zero for negative values of x , although it is common to assume that $E(x) = 0$ for $x < 0$. For $x < 0$, the value of $-E(x)$ represents a lower bound on the increments of a process over any interval of length $-x$.

Before proceeding further, let us make a few definitions. First, given any function $E(\cdot)$, define the “tilde” operator as follows:

$$\tilde{E}(t) = -E(-t) \text{ for all } t. \quad (3.5)$$

Given any $R \in \mathcal{M}$, define $D_{t,k}(R) = \inf\{d : d \geq 0 \text{ and } R(t+d) \geq k\}$. Note that the virtual delay in a system with arrival and departure processes R_{in} and R_{out} is $D(t) = D_{t,R_{\text{in}}(t)}(R_{\text{out}})$. Finally, for any $R \in \mathcal{M}$, define $D_0(R) = D_{0,0}(R)$.

3.3 Quality of Service Guarantees for Shift Invariant Service Mappings

In this section we consider a single network element whose arrival process has envelope E . We suppose the network element has a shift invariant service mapping and derive bounds on virtual delay and backlog. We also find an envelope for the departure process.

Theorem 3.3. *Suppose \mathcal{S} is a shift invariant service mapping for a network element. Suppose the arrival process to the network element has envelope E . Then the virtual delay $D(t)$ is upper bounded according to*

$$D(t) \leq D_0(\mathcal{S}(\tilde{E})) \text{ for all } t. \quad (3.6)$$

Proof. Fix any t . We have

$$\begin{aligned} D(t) &= \inf\{d : d \geq 0 \text{ and } R_{\text{out}}(t+d) \geq R_{\text{in}}(t)\} \\ &= D_{t,R_{\text{in}}(t)}(R_{\text{out}}) \\ &\leq D_{t,R_{\text{in}}(t)}(\mathcal{S}(R_{\text{in}})) \\ &\leq D_{t,R_{\text{in}}(t)}(\mathcal{S}((R_{\text{in}})_{E,t})) \\ &= D_{t,R_{\text{in}}(t)}(\mathcal{S}(R_{\text{in}}(t) - E(t - \cdot))) \\ &= D_{t,R_{\text{in}}(t)}(R_{\text{in}}(t) + \mathcal{S}(-E(t - \cdot))) \\ &= D_{t,0}(\mathcal{S}(-E(t - \cdot))) \\ &= D_{t,0}(\mathcal{S}(-E(-(\cdot - t)))) \\ &= D_{0,0}(\mathcal{S}(-E(-(\cdot)))) \\ &= D_0(\mathcal{S}(\tilde{E})) . \end{aligned}$$

In the above sequence the first equality follows from the definition of virtual delay, the second equality follows from the definition of $D_{t,k}(R)$, the third inequality follows since \mathcal{S} is a minimum service mapping and $D_{t,k}(R)$ is monotone decreasing in R . The fourth

inequality follows from (3.4) and the monotonicity of \mathcal{S} . The remaining equalities follow from the shift invariance of \mathcal{S} .

Suppose \mathcal{S} is a shift invariant service mapping for a network element. Suppose the arrival process to the network element has envelope E . Then the backlog $B(t)$ is upper bounded according to

$$B(t) \leq \tilde{\mathcal{S}}(\tilde{E})(0) \text{ for all } t. \quad (3.7)$$

Fix any t . We have

$$\begin{aligned} B(t) &= R_{\text{in}}(t) - R_{\text{out}}(t) \\ &\leq R_{\text{in}}(t) - \mathcal{S}(R_{\text{in}})(t) \\ &\leq R_{\text{in}}(t) - \mathcal{S}((R_{\text{in}})_{E,t})(t) \\ &= R_{\text{in}}(t) - \mathcal{S}(R_{\text{in}}(t) - E(t - \cdot))(t) \\ &= -\mathcal{S}(-E(t - \cdot))(t) \\ &= -\mathcal{S}(-E(-(\cdot - t)))(t) \\ &= -\mathcal{S}(-E(-(\cdot)))(0) \\ &= -\mathcal{S}(\tilde{E})(0) \\ &= \tilde{\mathcal{S}}(\tilde{E})(0). \end{aligned}$$

In the above sequence the first equality follows from the definition of backlog, the second equality follows since \mathcal{S} is a minimum service mapping. The third inequality follows from (3.4) and the monotonicity of \mathcal{S} . The remaining equalities follow from the shift invariance of \mathcal{S} . \square

We say a network element is *conservative* if we always have $R_{\text{out}} \leq R_{\text{in}}$.

Theorem 3.4. *Suppose \mathcal{S} is a shift invariant service mapping for a conservative network element. Suppose the arrival process to the network element has envelope E . Then the departure process has envelope E_{out} where*

$$E_{\text{out}} = \tilde{\mathcal{S}}(\tilde{E}). \quad (3.8)$$

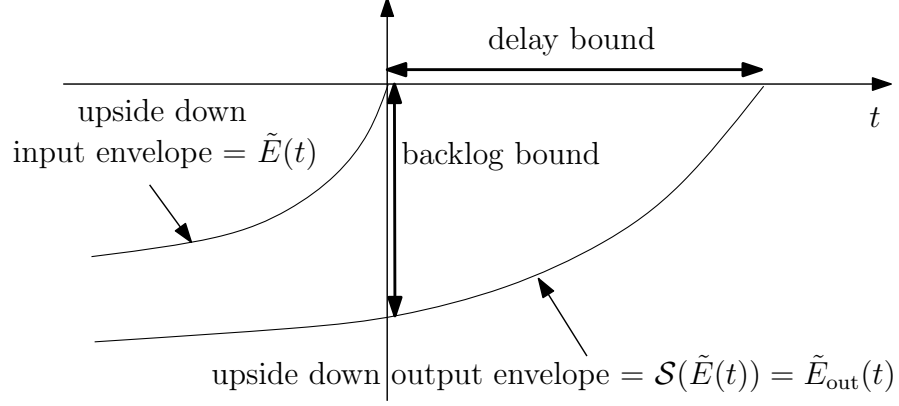


Figure 3.1: \mathcal{S} -mapping: delay, backlog and output bounds

Proof. Fix any s, t . We have

$$\begin{aligned}
 R_{\text{out}}(t) - R_{\text{out}}(s) &\leq R_{\text{in}}(t) - R_{\text{out}}(s) \\
 &\leq R_{\text{in}}(t) - \mathcal{S}(R_{\text{in}})(s) \\
 &\leq R_{\text{in}}(t) - \mathcal{S}((R_{\text{in}})_{E,t})(s) \\
 &= R_{\text{in}}(t) - \mathcal{S}(R_{\text{in}}(t) - E(t - \cdot))(s) \\
 &= -\mathcal{S}(-E(t - \cdot))(s) \\
 &= -\mathcal{S}(-E(-(\cdot - t)))(s) \\
 &= -\mathcal{S}(-E(-(\cdot)))(s - t) \\
 &= -\mathcal{S}(\tilde{E})(s - t) \\
 &= \tilde{\mathcal{S}}(\tilde{E})(t - s) .
 \end{aligned} \tag{3.9}$$

In the above sequence the first equality follows since the network element is conservative, the second equality follows since \mathcal{S} is a minimum service mapping. The third inequality follows from (3.4) and the monotonicity of \mathcal{S} . The remaining equalities follow from the shift invariance of \mathcal{S} . \square

Note that (3.8) can be rewritten as $\tilde{E}_{\text{out}} = \mathcal{S}(\tilde{E})$, which has a simple intuitive appeal. The theorems of this section are illustrated graphically in Figure 3.1.

3.3.1 Linear Time Invariant Service Mappings

We conclude this section by considering a service mapping \mathcal{S} that is linear and time invariant. In other words, the service mapping corresponds to a service curve guarantee, i.e.

$$\mathcal{S}(R) = R * S$$

where S is a service curve for the network element. In this case we show that the theorems of the previous subsection reduce to previously known results. We assume that the arrival process conforms to the envelope E .

Observe that

$$\begin{aligned} \mathcal{S}(\tilde{E})(-x) &= (\tilde{E} * S)(-x) \\ &= \inf_y \{ \tilde{E}(-x - y) + S(y) \} \\ &= \inf_y \{ -E(y + x) + S(y) \} \\ &= -\sup_y \{ E(y + x) - S(y) \} \\ &= -(E \oslash S)(x) , \end{aligned} \tag{3.10}$$

where we use “ \oslash ” to denote the deconvolution operator, i.e. $(F \oslash G)(x) = \sup_y \{ F(x + y) - G(y) \}$ for all x .

First, consider the result of Theorem 3.4. The departure process has envelope E_{out} where $E_{out} = \tilde{\mathcal{S}}(\tilde{E})$. In view of (3.10), we have $E_{out} = E \oslash S$, which agrees with the result in [2].

Second, consider the result of Theorem 3.3. The backlog is upper bounded by $-\mathcal{S}(\tilde{E})(0)$. In view of (3.10), we thus have $B(t) \leq (E \oslash S)(0) = \sup_y \{ E(y) - S(y) \}$, which agrees with the result in [2].

Third, consider the result of Theorem 3.3. Using (3.10), the virtual delay $D(t)$ is upper

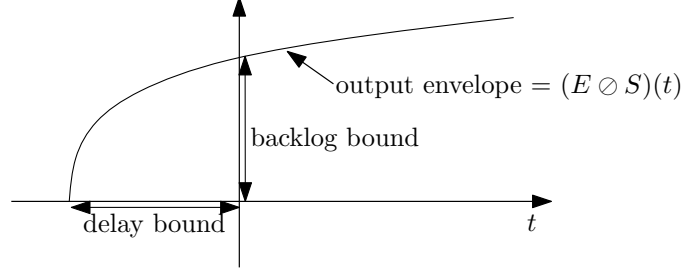


Figure 3.2: Service curves: delay, backlog and output bounds

bounded as follows:

$$\begin{aligned}
 D(t) &\leq D_0(\mathcal{S}(\tilde{E})) \\
 &= \inf\{d : d \geq 0 \text{ and } \mathcal{S}(\tilde{E})(d) \geq 0\} \\
 &= \inf\{d : d \geq 0 \text{ and } -(E \oslash S)(-d) \geq 0\} \\
 &= \inf\{d : d \geq 0 \text{ and } -\sup_y \{E(y-d) - S(y)\} \geq 0\} \\
 &= \inf\{d : d \geq 0 \text{ and } \sup_y \{E(y-d) - S(y)\} \leq 0\} \\
 &= \inf\{d : d \geq 0 \text{ and } E(y-d) \leq S(y) \text{ for all } y\} ,
 \end{aligned}$$

which agrees with the result in [2], i.e. the delay is upper bounded by the “maximum horizontal distance” between the graphs of E and S .

These results are illustrated graphically in Figure 3.2 in the context of the graph of $E \oslash S$.

3.4 FIFO Multiplexers

In this section, we apply the results in the previous section to the case in which network elements correspond to FIFO multiplexers. First, in the next section, we consider a single FIFO multiplexer.

3.4.1 A Single FIFO Multiplexer

Consider two arrival streams incident on a FIFO multiplexer, described by R_{in} and R_{in}^x , with traffic envelopes E and E^x , respectively. The multiplexer serves data in a

FIFO manner as fast as possible with a maximum service rate of C bits per second. From the point of view of the aggregate arrival stream, $R_{in} + R_{in}^x$, a service curve of G is provided, where $G(x) = Cx$ if $x \geq 0$ and $G(x) = 0$ otherwise. In other words, if the corresponding departure streams are denoted as R_{out} and R_{out}^x , then we have $R_{out} + R_{out}^x \geq G * (R_{in} + R_{in}^x)$.

If we assume that packets are served non-preemptively, then if we have non-zero packet sizes (i.e. a non-fluid model), then bits might not depart in exactly FIFO order. Therefore, for simplicity we assume a fluid model, which corresponds to “ $L = 0$ ” as discussed in [14].

In this case, from Theorem 4.1 of [14], it is known that the delay for stream R_{in} is upper bounded by $\bar{D}_{FCFSMUX}$, i.e. $R_{out}(t + \bar{D}_{FCFSMUX}) \geq R_{in}(t)$ for all t , where

$$\bar{D}_{FCFSMUX} = \frac{1}{C} \max_{u \geq 0} [E(u) + E^x(u) - Cu] . \quad (3.11)$$

Since the system is FIFO, we have using the result from [15] that $R_{out} \geq R_{in} * S_T$ holds for any $T \geq 0$, with S_T given in (3.1). Thus we have $R_{in} \rightarrow \hat{S} \rightarrow R_{out}$ holds, with \hat{S} given in (3.2). It can be shown that $D_0(\hat{S}(\tilde{E})) \leq \bar{D}_{FCFSMUX}$. In fact, equality holds here since the delay bound from [14] is the best possible. Thus, in some sense Theorem 4.1 of [14] is a special case of Theorem 3.3.

Furthermore, from Theorem 4.4 of [14], it is known that R_{out} has envelope E_{out} , where

$$E_{out}(x) = \max_{\Delta \geq 0, D \geq 0} [\min\{E(x + D), E(x + D + \Delta) + E^x(\Delta) - C(\Delta + D)\}] . \quad (3.12)$$

We can show that in fact Theorem 3.4 reduces to this result in this case. For brevity we do not include the details here, but we note that we assumed continuity of $E(x)$ and $E^x(x)$ for $x > 0$. We conjecture that this assumption is unnecessary, however.

Thus, we assert that Theorems 3.3 and Theorem 3.4 here are more general than Theorems 4.1 and Theorem 4.4 of [14].

3.4.2 FIFO Multiplexers in Tandem

In this section we consider the system illustrated in Figure 3.3, where each arrival processes R_i has an envelope E_i of the form $E_i(t) = \sigma_i + \rho_i t$. We are interested in an upper bound for the total delay for flow 0, which is the sum of the delay through each node. We assume that the system is stable, that is $\rho_0 + \rho_1 \leq C_1$ and $\rho_0 + \rho_2 \leq C_2$,

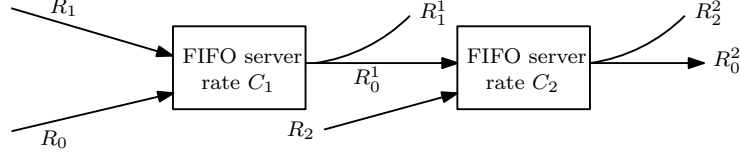


Figure 3.3: Two nodes in tandem

which ensures that the delay is bounded. We use the notation $\hat{\mathcal{S}}_1$ and $\hat{\mathcal{S}}_2$ to denote the corresponding minimum service mappings for flow 0 at the first and second multiplexer, respectively, as implied by (3.1) and (3.2).

It can be shown that

$$\hat{\mathcal{S}}_1(\tilde{E}_0(t)) = \begin{cases} \rho_0 t - \sigma_0 - \rho_0 \frac{\sigma_1}{C_1}, & \text{if } t \leq \frac{\sigma_1}{C_1} \\ C_1 t - \sigma_1 - \sigma_0, & \text{if } \frac{\sigma_1}{C_1} \leq t \leq \frac{\sigma_1 + \sigma_0}{C_1} \\ 0, & \text{if } t \geq \frac{\sigma_1 + \sigma_0}{C_1}. \end{cases}$$

Moreover, it can be shown that if $C_2 - \rho_2 \geq C_1$ we have

$$\hat{\mathcal{S}}_2(\hat{\mathcal{S}}_1(\tilde{E}_0(t))) = \begin{cases} \rho_0 \left(t - \frac{\sigma_2}{C_2} - \frac{\sigma_1}{C_1} \right) - \sigma_0, & \text{if } t \leq \frac{\sigma_1}{C_1} + \frac{\sigma_2}{C_2} \\ C_1 \left(t - \frac{\sigma_2}{C_2} - \frac{\sigma_1}{C_1} \right) - \sigma_0, & \text{if } \frac{\sigma_1}{C_1} + \frac{\sigma_2}{C_2} \leq t \leq \frac{\sigma_1 + \sigma_0}{C_1} + \frac{\sigma_2}{C_2} \\ 0, & \text{if } t \geq \frac{\sigma_1 + \sigma_0}{C_1} + \frac{\sigma_2}{C_2}, \end{cases}$$

whereas if $C_2 - \rho_2 \leq C_1$ we have

$$\hat{\mathcal{S}}_2(\hat{\mathcal{S}}_1(\tilde{E}_0(t))) = \begin{cases} \rho_0 \left(t - \frac{\sigma_2}{C_2} - \frac{\sigma_1}{C_1} \right) - \sigma_0, & \text{if } t \leq \frac{\sigma_1}{C_1} + \frac{\sigma_2}{C_2} \\ \frac{C_2 C_1}{C_1 + \rho_2} \left(t - \frac{\sigma_2}{C_2} - \frac{\sigma_1}{C_1} \right) - \sigma_0, & \text{if } \frac{\sigma_1}{C_1} + \frac{\sigma_2}{C_2} \leq t \leq \frac{\sigma_1}{C_1} + \frac{\sigma_2}{C_2} + \frac{\sigma_0(C_1 + \rho_2)}{C_2 C_1} \\ 0, & \text{if } t \geq \frac{\sigma_1}{C_1} + \frac{\sigma_2}{C_2} + \frac{\sigma_0(C_1 + \rho_2)}{C_2 C_1}. \end{cases}$$

The upper bound on the end-to-end delay as given by Theorem 3.2 and Theorem 3.3 is: $D = D_0((\hat{\mathcal{S}}_1 \circ \hat{\mathcal{S}}_2)(\tilde{E}_0)) = D_0(\hat{\mathcal{S}}_2(\hat{\mathcal{S}}_1(\tilde{E}_0)))$. Carrying out this calculation, it can be verified that our upper bound D on end to end delay is given by

$$D = \begin{cases} \frac{\sigma_0 + \sigma_1}{C_1} + \frac{\sigma_2}{C_2}, & \text{if } C_2 - \rho_2 \geq C_1 \\ \frac{\sigma_1}{C_1} + \frac{\sigma_2}{C_2} + \frac{\sigma_0(C_1 + \rho_2)}{C_2 C_1} = \frac{\sigma_1}{C_1} + \frac{\sigma_2}{C_2} + \frac{\sigma_0}{C_2} + \frac{\sigma_0 \rho_2}{C_2 C_1}, & \text{if } C_2 - \rho_2 \leq C_1. \end{cases} \quad (3.13)$$

These bounds are indeed achievable. To see why this is true when $C_2 - \rho_2 \leq C_1$ consider the following arrival pattern: $R_0(t) = R_1(t) = 0$ for $t < 0$; $R_0(t) = \sigma_0$ and

$R_1(t) = \sigma_1$ for $t \geq 0$. That is both flows at the first server have a burst at time 0. Suppose that flow 1 is served before flow 0, then suppose that $R_2(t) = 0$ for $t < \frac{\sigma_1}{C_1}$ and that $R_2(t) = \rho_2(t - \frac{\sigma_1}{C_1}) + \sigma_2$ otherwise. Under these assumptions the last bit of the burst from flow 0 will experience a total delay given by (3.13). If $C_2 - \rho_2 \leq C_1$ the arrival pattern for flow 0 and 1 is the same while in this case $R_2(t) = 0$ for $t < \frac{\sigma_1 + \sigma_0}{C_1}$ and $R_2(t) = \sigma_2$ otherwise; again the last bit of the burst from flow 0 will experience a total delay given by (3.13).

4

Worst Case Average Delay for a Single FIFO Queue

4.1 Introduction

In the previous chapters we have dealt with point-wise bounds, that is bounds that hold at any given time. They can never be violated no matter what the inputs are (as long as they satisfy the corresponding envelopes). At the same time, in most cases, these bounds cannot be achieved over arbitrary time *intervals*. They can only be achieved for a limited number of specific time instants within each interval. In other words given any arbitrary value t it is often possible to find an arrival pattern so that the delay bound, for example, is achieved exactly at that time. But, if we are interested in a time interval $[t_1, t_2]$, it is not possible, at least in general, to find an arrival pattern such that the delay bound is achieved for all $t \in [t_1, t_2]$.

In some specific cases this might be possible. For example if a single FIFO queue with two inputs with sigma-rho envelopes is such that $C = \rho_0 + \rho_1$, where C is the capacity of the server and ρ_0 and ρ_1 are the long term rates for the two inputs, then the output and delay bounds can be achieved over arbitrary time intervals. But this is somewhat of an extreme case on the verge of instability and the bounds can be achieved if the inputs are greedy, that is they follow the envelope sending as much traffic as possible, causing only one unbounded busy period.

Note that these issues are not caused by the fact that some of these bounds are not tight. The reason is that they are point-wise bounds, as such they have to be true for any t . As an example consider a FIFO queue with inputs that have sigma-rho envelopes (for the sake of simplicity assume that all the envelopes have the same parameters): the worst case delay bound can be achieved at an arbitrary time instant, but after this bound is achieved it will take at least σ/ρ units of time before the input traffic can send another maximum size burst without violating the envelope. Therefore if the delay bound is achieved at time t it cannot be achieved anywhere else in the interval $[t - \frac{\sigma}{\rho}, t + \frac{\sigma}{\rho}]$.

One bound that does not have this shortcoming is the *worst case average delay*, where the average is taken over time. In this case the bound can be achieved over arbitrary time intervals, note that this is not a point-wise bound but rather a bound that has to be true for any time interval.

Theorem 4.7 in [14] gives a bound for the worst case average delay for a FIFO queue serving a single input flow satisfying an envelope. We would like to extend this result to the case where there are two input flows and not just one. A similar problem has been analyzed in [18] but in a probabilistic setting using Palm probabilities.

4.2 Problem Statement and Some Definitions

Consider a single FIFO server with capacity C serving two flows $R_{0,\text{in}}$ and $R_{1,\text{in}}$ with envelopes E_0 and E_1 respectively, such that the system is stable¹ and such that E_0 and E_1 are concave and E_1 is piecewise linear. We are interested in finding an upper bound on the average delay for flow R_0 , where the average is taken over time. In order to rigorously define this quantity we first define the “ingress” and “egress” time for the bit of the i -th input flow arriving after y units of traffic have arrived from the same input:

$$t_{i,\text{in}}(y) = \inf \{t : R_{i,\text{in}}(t) \geq y\}, \quad t_{i,\text{out}}(y) = \inf \{t : R_{i,\text{out}}(t) \geq y\}$$

where $R_{i,\text{in}}$ and $R_{i,\text{out}}$ are generic input and output pairs. Note that $R_{\text{in}}(t) \geq R_{\text{out}}(t) \forall t$ implies that $t_{\text{out}} \geq t_{\text{in}}$. The delay for this bit is defined as:

$$d_i(y) = t_{i,\text{out}}(y) - t_{i,\text{in}}(y),$$

¹That is the delay is bounded.

and the average delay as:

$$d_{i,\text{avg}} = \frac{1}{B} \int_0^B d_i(y) dy$$

where B is the total amount of data sent by flow i .

We will show that there does exist a worst case arrival pattern for $R_{0,\text{in}}$ and $R_{1,\text{in}}$ so that we can use the average delay for this specific case as an upper bound. The main idea is that, exploiting the concavity of E_1 and E_0 , we can start from an arbitrary arrival pattern for both flows and then modify them in such a way that the average delay can only increase (or stay the same). Dealing with the cross traffic it is fairly easy: we will show that flow 1 should simply send as much traffic as the envelope allows starting from the beginning of each busy period.

For flow 0 the argument is more complicated but the basic idea is fairly simple; given that E_1 (the cross traffic envelope) is concave there exist a time α at which the backlog due to flow 1 is maximized if flow 1 were the only input to the server (we will give a rigorous definition for all the quantities involved in the remainder of this chapter). Flow 0 should try to send all its traffic as close as possible to time α without violating its envelope. We will show that “best” way of sending this traffic (i.e. the arrival pattern with the worst case average delay) is to use the biggest slope of E_0 around time α (E_0 is piecewise linear), then use the second biggest slope in two different intervals one to the left of α and one to the right of α , and so on for the other slopes of E_0 . We will show that this optimal solution has a certain property that we will call “water-filling”, because the length of each interval is the solution of a problem that is somewhat equivalent to pouring water in a convex bowl and letting the water reach an uniform level.

4.3 Preliminary Results

Before proving the main theorem that gives a bound for the average delay we are going to present some preliminary results that will be used later on. First we consider an optimization problem that will appear in the proof of the main theorem and that introduces the water filling condition. Next we turn our attention to the problem at hand and we show how we can modify arbitrary arrival patterns for $R_{0,\text{in}}$ and $R_{1,\text{in}}$ in order to increase the average delay. The basic idea is to consider the “cross traffic” first (R_1) and then the “through traffic” (R_0). In a first lemma we show that for any (fixed) $R_{0,\text{in}}$

it is possible to upper bound $d_0(y)$ by considering the case where $R_{1,\text{in}}$ is greedy (that is it follows the envelope) starting at the beginning of the system busy period containing $t_{0,\text{in}}(y)$. Throughout the proofs we will use roman numeral superscripts ($R_{0,\text{in}}^{\text{I}}, R_{0,\text{in}}^{\text{II}}$) to distinguish between specific realizations of the same process corresponding to different scenarios. Similarly let $d_0^{\text{I}}(y)$ (resp. $d_0^{\text{II}}(y)$) be the delay for the bit entering the server at time $t_{0,\text{in}}(y)$ when the inputs are $R_{0,\text{in}}^{\text{I}}$ and $R_{1,\text{in}}^{\text{I}}$ (resp. $R_{0,\text{in}}^{\text{II}}$ and $R_{1,\text{in}}^{\text{II}}$). Furthermore we define a system busy period as each of the maximal-size (longest) time intervals over which the total backlog is non-zero.

4.3.1 An optimization problem

Suppose we are given a function $f(x)$ that is concave for $x > 0$ and such that $f(x)$ is decreasing for all $x \geq \alpha$ for some $\alpha \geq 0$. We will assume the α is the smallest such value, that is

$$\alpha = \inf\{u \geq 0 : f(x_1) \geq f(x_2) \forall x_2 \geq x_1 \geq u\}.$$

Note that the definition of α and the fact that f is concave imply that f is increasing for $x \leq \alpha$ and decreasing for $x \geq \alpha$. We will also assume that $f(x) = 0$ for $x < 0$.

Furthermore assume that we are given a collection of N “weights” $r_n \geq 0$ ($n = 1 \dots N$), and “lengths” $L_n \geq 0$ ($n = 1 \dots N$). Let $T_{nm} = (\tau_{nm}, t_{nm})$ ($t_{nm} > \tau_{nm} \geq 0 \forall n, m$) be a collection of (non overlapping) intervals whose individual length we will define as $l_{nm} = t_{nm} - \tau_{nm}$, let M_n be the number of intervals whose first index is n . Based on these definitions, we would like to solve the following optimization problem:

$$\max_{\tau_{nm}, t_{nm}} \sum_{n=1}^N r_n \int_{\bigcup_m T_{nm}} f(x) dx, \quad (4.1)$$

subject to the constraints:

$$T_{ij} \cap T_{nm} = \emptyset \forall i, j, n, m \quad (4.2)$$

$$\sum_{m=1}^{M_n} l_{nm} = L_n \quad n = 1 \dots N. \quad (4.3)$$

One possible interpretation of this problem is that we would like to maximize the weighted sum of integrals (r_n are the weights) over a certain collection of intervals. Each interval is associated with a weight and we can place these intervals anywhere we want as long

as they are non-overlapping and the sum of the lengths of all the intervals associated with wight n is exactly L_n . Note that the the fact that $f(x) \leq f(\alpha)$ for any x implies that f is bounded so that $\int_{(\tau_{nm}, t_{nm})} f = \int_{[\tau_{nm}, t_{nm}]} f$. Note as well that the definition of α and the fact that f is concave imply that $f(\alpha) = \sup\{f(x)\}$, unless $\alpha = 0$ and f is discontinuous at 0, in this case $\sup\{f(x)\} = \lim_{x \rightarrow 0^+} f(x)$. It is also useful to define the following quantities:

$$\begin{aligned}\tau_{\min} &= \min_{n,m} \{\tau_{nm}\} \\ t_{\max} &= \max_{n,m} \{t_{nm}\} \\ \xi &= \min\{f(\tau_{\min}), f(t_{\max})\} \\ L &= \sum_{n=1}^N L_n.\end{aligned}$$

Before we examine the general case it is useful to point out that there is one trivial case. Let

$$\beta = \sup\{u \geq \alpha : f(u) \geq f(\alpha)\},$$

it is possible that $\beta > \alpha$, this happens if f is “flat” for values of x between α and β , if $\beta > \alpha$ and $L \leq \beta - \alpha$ then we can distribute the weights in any way we want as long as $\alpha \leq \tau_{\min} < t_{\max} \leq \beta$, any solution satisfying this condition will have the same cost and will achieve the maximum.

The next three lemmas show that the optimal solution for (4.1) needs to satisfy three properties. The first one says that there are no “holes” in the collection of intervals, that is for every n and m there exist i and j such that $\tau_{nm} = t_{ij}$ (except for the very first interval). The second one deals with the location of τ_{\min} and t_{\max} in the optimal solution and the third one shows that in the optimal solution weights are distributed around the point α according to a specific property.

Lemma 4.1. *The optimal solution of (4.1) is such that for every n and m there exist i and j satisfying $\tau_{nm} = t_{ij}$ unless $\tau_{nm} = \tau_{\min}$.*

Proof. By contradiction: suppose there exist an optimal solution $\{T_{nm}\}$ with i, j, k, p such that

$$(t_{ij}, \tau_{kp}) \cap \left(\bigcup_{n,m} T_{nm} \right) = \emptyset,$$

that is there are no intervals between t_{ij} and τ_{kp} . Assuming $t_{ij} \geq \alpha$, it is then possible to construct a new set of intervals T'_{nm} as follows:

$$\tau'_{nm} = \begin{cases} \tau_{nm}, & \text{if } \tau_{nm} \leq \tau_{kp} \\ \tau_{nm} - (\tau_{kp} - t_{ij}) & \text{if } \tau_{nm} > \tau_{kp} \end{cases}$$

$$t'_{nm} = \begin{cases} t_{nm}, & \text{if } t_{nm} < t_{kp} \\ t_{nm} - (\tau_{kp} - t_{ij}) & \text{if } t_{nm} \geq t_{kp}. \end{cases}$$

That is all the intervals after τ_{kp} are moved to the left (closer to α) by $\tau_{kp} - t_{ij}$. Let $A = \{(n, m) : \tau_{nm} \leq \tau_{ij}\}$ and $B = \{(n, m) : \tau_{nm} \geq \tau_{kp}\}$ so that $(\bigcup_{a \in A} T_a) \cup (\bigcup_{b \in B} T_b) = \bigcup_{n,m} T_{nm}$ and taking the difference between the two solutions we have:

$$\int_{\bigcup_m T_{nm}} f(x) dx - \int_{\bigcup_m T'_{nm}} f(x) dx = \int_{\bigcup_{a \in A} T_a} + \int_{\bigcup_{b \in B} T_b} - \int_{\bigcup_{a \in A} T'_a} - \int_{\bigcup_{b \in B} T'_b} \quad (4.4)$$

$$= \int_{\bigcup_{b \in B} T_b} - \int_{\bigcup_{b \in B} T'_b} \quad (4.5)$$

$$= \sum_{(n,m) \in B} \int_{\tau_{nm}}^{t_{nm}} f(x) dx - \sum_{(n,m) \in B} \int_{\tau'_{nm}}^{t'_{nm}} f(x) dx \quad (4.6)$$

$$= \sum_{(n,m) \in B} \int_{\tau_{nm}}^{t_{nm}} f(x) dx \quad (4.7)$$

$$- \sum_{(n,m) \in B} \int_{\tau_{nm} - (\tau_{kp} - t_{ij})}^{t_{nm} - (\tau_{kp} - t_{ij})} f(x) dx \quad (4.8)$$

$$\leq 0. \quad (4.9)$$

In (4.4) we did not include the weights r_n because T_{nm} and T'_{nm} have the same weight assignment. The only difference between the two solutions is the some of the intervals in T'_{nm} are shifted to the left. The first equality (4.4) follows from the definitions of A and B . Similarly it follows from the definition of A that $\int_{\bigcup_{a \in A} T_{nm}} = \int_{\bigcup_{a \in A} T'_{nm}}$ as $\forall a \in A$ $T_a = T'_a$ (all the intervals with $\tau_{nm} \leq \tau_{ij}$ are not moved) and hence (4.5). As f is continuous we can rewrite (4.5) as the sum of the integrals over each interval to obtain (4.6). From the definition of B we have that $\forall (n, m) \in B$ $t'_{nm} = t_{nm} - (\tau_{kp} - t_{ij})$ and $\tau'_{nm} = \tau_{nm} - (\tau_{kp} - t_{ij})$ and hence (4.7). Finally the fact that $t_{ij} \geq \alpha$ and the definition of B imply that $\forall (n, m) \in B$ $\tau_{nm} \geq \alpha$ and $\tau_{nm} - (\tau_{kp} - t_{ij}) \geq \alpha$ so that $f(x) \leq f(x - (\tau_{kp} - t_{ij}))$ for any $x \in (\tau_{nm}, t_{nm})$ such that $(n, m) \in B$, as f is a decreasing (non-increasing) function

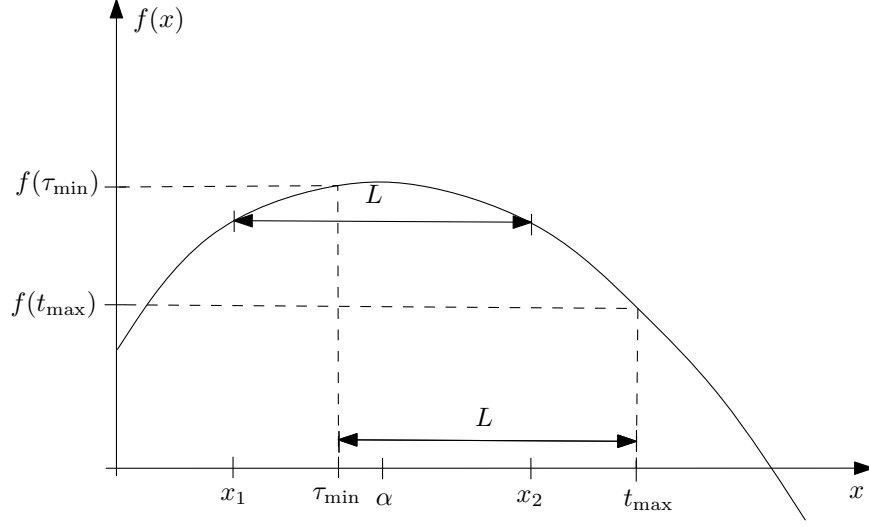


Figure 4.1: How to construct a better solution

for $x \geq \alpha$, hence

$$\int_{\tau_{nm}}^{t_{nm}} f(x) dx \leq \int_{\tau_{nm} - (\tau_{kp} - t_{ij})}^{t_{nm} - (\tau_{kp} - t_{ij})} f(x) dx \quad \forall (n, m) \in B,$$

from this (4.9) follows immediately which is a contradiction because we were assuming that $\{T_{nm}\}$ is an optimal solution but we have constructed a better one.

So far we were assuming that $t_{ij} \geq \alpha$, if this is not the case a similar argument holds. If $t_{ij} \leq \alpha$ simply shift all the intervals before t_{ij} to the right by $\tau_{kp} - t_{ij}$ and leave all the others unchanged. if $t_{ij} \leq \alpha \leq \tau_{kp}$ shift all the intervals before t_{ij} to the right by $\tau_{kp} - t_{ij}$ and all those after t_{ij} to the left by the same amount. In both cases a similar argument shows that the value of the integral over each interval is going to increase given that f is increasing for $x \leq \alpha$ and decreasing for $x \geq \alpha$. \square

Lemma 4.2. *Let $f(0^+) = \lim_{x \rightarrow 0^+} f(x)$, if $f(0^+) \leq f(L)$ the optimal solution of (4.1) is such that*

$$f(\tau_{min}) = f(t_{max}),$$

while if $f(0^+) > f(L)$ the optimal solution is such that

$$\tau_{min} = 0, t_{max} = L.$$

Proof. First let us consider the case where $f(0^+) \leq f(L)$ and suppose that there exists an optimal solution $\{T_{nm}\}$ such that $f(\tau_{min}) > f(t_{max})$ (see Figure 4.1). From Lemma 4.1

we know that the optimal solution is such that $t_{\max} - \tau_{\min} = L$. Given that $f(0^+) \leq f(L)$, and that f is continuous there exist x_1 and x_2 such that $x_1 \leq \alpha \leq x_2$, $x_2 - x_1 = L$ and $f(x_1) = f(x_2)$, note that we also have $x_1 \leq \tau_{\min}$ and $x_2 \leq t_{\max}$.

Let $A = \{(n, m) : \tau_{nm} \leq x_2\}$ and $B = \{(n, m) : \tau_{nm} \geq x_2\}$ that is all the intervals to the left and to the right of x_2 , respectively, in the original solution. Next construct a new solution $\{T'_{nm}\}$ by moving all the intervals in B to the left of τ_{\min} without changing their relative ordering. If $\tau_{ij} < x_2 < t_{ij}$ for some i, j then divide this interval into two new intervals: (τ_{ij}, x_2) and (x_2, t_{ij}) and leave the first one in the set A and add the second one to the set B , this way we will have one more interval than the original solution and $M'_i = M_i + 1$ and $B' = B \cup (\tau_{iM'_i} = x_2, t_{iM'_i} = t_{ij})$. Formally:

$$\tau'_{nm} = \begin{cases} \tau_{nm}, & \text{if } (n, m) \in A \\ \tau_{nm} - (t_{\max} - x_1), & \text{if } (n, m) \in B \\ x_2 - (t_{\max} - x_1), & \text{if } n = i \text{ and } m = M'_i \end{cases}$$

$$t'_{nm} = \begin{cases} t_{nm}, & \text{if } (n, m) \in A \text{ and } (n, m) \neq (i, j) \\ t_{nm} - (t_{\max} - x_1), & \text{if } (n, m) \in B \\ x_2, & \text{if } n = i \text{ and } m = j \\ t_{ij} - (t_{\max} - x_1) & \text{if } n = i \text{ and } m = M'_i. \end{cases}$$

Note that $\tau'_{\min} = x_1$ and $t'_{\max} = x_2$ so that $f(\tau'_{\min}) = f(t'_{\max})$.

Taking the difference between the two solutions and letting $\delta = (t_{\max} - x_1)$ we have:

$$\int_{\bigcup_m T_{nm}} f(x)dx - \int_{\bigcup_m T'_{nm}} f(x)dx = \int_{\bigcup_{a \in A} T_{nm}} + \int_{\bigcup_{b \in B} T_{nm}} - \int_{\bigcup_{a \in A} T'_{nm}} - \int_{\bigcup_{b \in B'} T'_{nm}} \quad (4.10)$$

$$= \int_{\bigcup_{b \in B} T_{nm}} + \int_{(x_2, t_{ij})} - \int_{\bigcup_{b \in B} T'_{nm}} - \int_{(x_2 - \delta, t_{ij} - \delta)} \quad (4.11)$$

$$= \sum_{(n,m) \in B} \int_{\tau_{nm}}^{t_{nm}} f + \int_{x_2}^{t_{ij}} f \quad (4.12)$$

$$- \sum_{(n,m) \in B} \int_{\tau'_{nm}}^{t'_{nm}} f - \int_{x_2 - \delta}^{t_{ij} - \delta} f$$

$$= \sum_{(n,m) \in B} \int_{\tau_{nm}}^{t_{nm}} f + \int_{x_2}^{t_{ij}} f \quad (4.13)$$

$$- \sum_{(n,m) \in B} \int_{\tau_{nm} - \delta}^{t_{nm} - \delta} f - \int_{x_2 - \delta}^{t_{ij} - \delta} f$$

$$\leq 0. \quad (4.14)$$

Similarly to what we did for the proof of Lemma 4.2 in (4.10) we did not include the weights r_n because T_{nm} and T'_{nm} have the same weight assignment. The only difference between the two solutions is the some of the intervals in T'_{nm} are shifted to the left. The first equality (4.10) follows from the definitions of A and B . For (4.11) it follows from the definition of A that

$$\int_{\bigcup_{a \in A} T_{nm}} = \int_{\bigcup_{a \in A \setminus (i,j)} T_{nm}} + \int_{(\tau_{ij}, x_2)} + \int_{(x_2, t_{ij})} = \int_{\bigcup_{a \in A} T'_{nm}} + \int_{(x_2, t_{ij})},$$

as $\forall a \in A \setminus (i,j)$ $T_a = T'_a$ (all the intervals with $\tau_{nm} \leq \tau_{ij}$ are not moved) and the interval (τ_{ij}, t_{ij}) is changed to (τ_{ij}, x_2) in T'_{nm} . From the definitions of B and B' we have that

$$\int_{\bigcup_{b \in B'} T'_{nm}} = \int_{\bigcup_{b \in B} T'_{nm}} + \int_{(x_2 - (t_{\max} - x_1), t_{ij} - (t_{\max} - x_1))}.$$

As f is continuous we can rewrite (4.11) as the sum of the integrals over each interval to obtain (4.12). From the definition of B we have that $\forall (n,m) \in B$ $t'_{nm} = t_{nm} - \delta$ and $\tau'_{nm} = \tau_{nm} - \delta$ and hence (4.13). The concavity of f and the definition of α and the facts that $x_1 \leq \tau_{\min}$ and that $x_2 \leq t_{\max}$ imply that $\forall u \in (x_1, \tau_{\min})$ and $\forall v \in (x_2, t_{\max})$

$f(u) \geq f(v)$ so that:

$$\begin{aligned} \int_{\tau_{nm}}^{t_{nm}} f(x) dx &\leq \int_{\tau_{nm}-\delta}^{t_{nm}-\delta} f(x) dx \quad \forall (n, m) \in B \\ \int_{x_2}^{t_{ij}} f(x) dx &\leq \int_{x_2-\delta}^{t_{ij}-\delta} f(x) dx. \end{aligned}$$

From this we have (4.14) and a contradiction.

So far we have been assuming that $f(0^+) \leq f(L)$ and $f(\tau_{\min}) > f(t_{\max})$; if $f(0^+) \leq f(L)$ and $f(\tau_{\min}) < f(t_{\max})$ a similar proof works with the only difference that now all the intervals to the right of x_1 are in the set A and all those to the left are in set B and these will be shifted to the right by $x_1 - \tau_{\min}$. If $f(0^+) > f(L)$ a similar proof works as well with $x_1 = 0$ and $x_2 = f(L)$, in this case all the intervals between x_2 and t_{\max} are in the set B and they will be shifted to the left by $t_{\max} - x_1$ to construct a new solution with a bigger value than the original one. \square

Before the next lemma we need to introduce the definition of the “water-filling” condition: $\forall w \in [\xi, f(\alpha)]$ define

$$x_1 = \begin{cases} \sup\{u : f(u) \leq w \text{ and } u \leq \alpha\}, & \text{if } w \geq f(0^+) \\ 0, & \text{if } w < f(0^+) \end{cases} \quad (4.15)$$

$$x_2 = \begin{cases} \inf\{u : f(u) \leq w \text{ and } u \geq \alpha\}, & \text{if } w \geq f(0^+) \\ f^{-1}(w), & \text{if } w < f(0^+). \end{cases} \quad (4.16)$$

Note that these two quantities are always well defined and that $f(x_1) = f(x_2)$ if $w \geq f(0^+)$.

Also, define

$$K = \{n : \exists m \text{ such that } x_1 \leq \tau_{nm} \text{ and } t_{nm} \leq x_2\}$$

$$Q = \{n : \exists m \text{ such that } t_{nm} \leq x_1 \text{ or } \tau_{nm} \geq x_2\}.$$

We say that a solution $\{T_{nm}\}$ does satisfy the water-filling condition if $\forall k \in K$ and $\forall q \in Q$ we have $r_k \geq r_q$. Intuitively if this condition is true it means that the weights r_n are distributed around α in such a way that the heavier weights are closer to α (see Figure 4.2). The name for this condition was inspired by the fact that f can be thought as an “upside down” bowl and in the case of a single weight the optimal solution is the same as if a certain amount of water was poured in the bowl until the water level was such

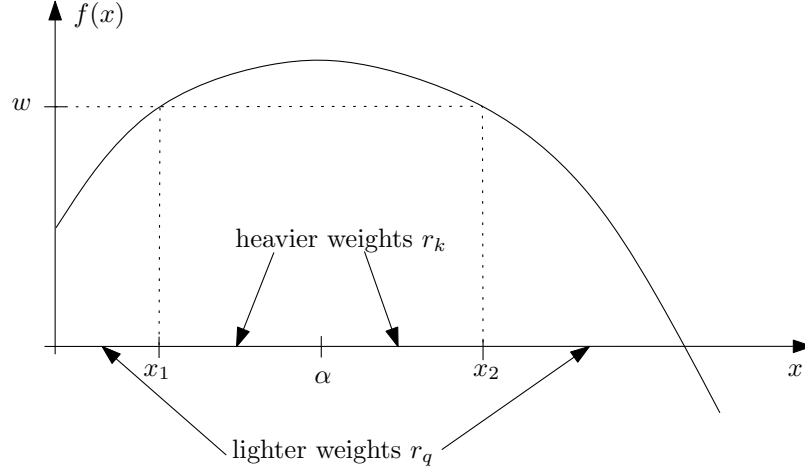
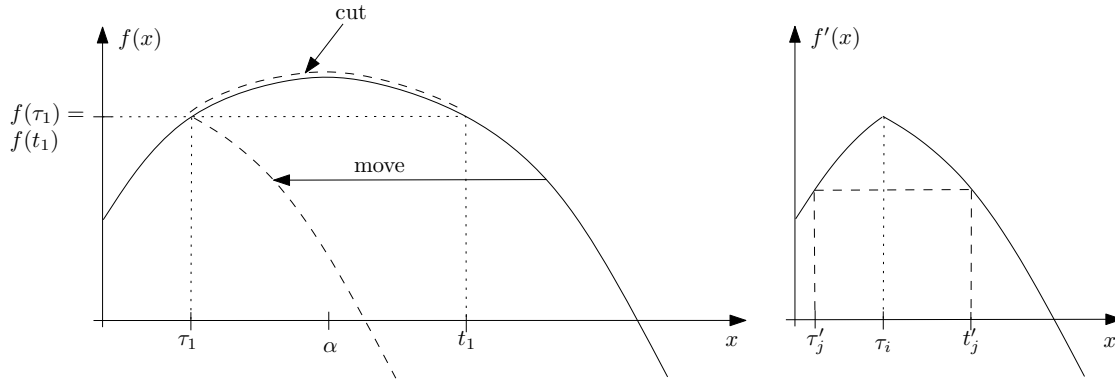


Figure 4.2: Water-filling condition

Figure 4.3: How to construct $f'(x)$

that $f(\tau_{\min}) = f(t_{\max})$ with “gravity” pulling the water up. Note that the analogy is not perfect because the amount (volume) of water is not fixed a priori, what it actually fixed is the length of the stable solution $L = t_{\max} - \tau_{\min}$. In the case of multiple weights the analogy is not perfect as well but we can construct an iterative algorithm that considers one weight at a time. Starting with heavier weight r_i find the optimal solution if r_i was the only weight, this will give τ_i and t_i appropriately centered around α , next consider the second heaviest weight r_j and construct a new “bowl” $f'(x)$ built as follows (see Figure 4.3):

$$f'(x) = \begin{cases} f(x), & \text{if } 0 \leq x \leq \tau_i \\ f(x + t_i - \tau_i), & \text{if } x > \tau_i, \end{cases}$$

then use this new “bowl” to find the water filling solution assuming r_j was the only weight.

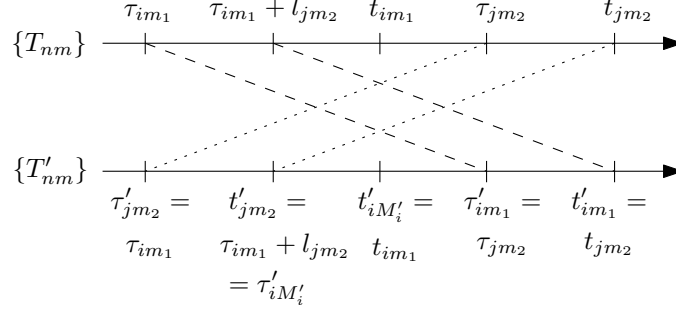


Figure 4.4: Constructing $\{T'_{nm}\}$ by switching T_{im_1} and T_{jm_2}

This solution gives τ'_j and t'_j , such that $f'(\tau'_j) = f'(t'_j)$. We can use these values to construct a solution for the original f by dividing $[\tau_j, t_j]$ in two intervals: $[\tau_{j1} = \tau'_j, t_{j1} = \tau_i]$ and $[\tau_{j2} = t_i, t_{j2} = t_i + t'_j - \tau_i]$, in other words every weight, other than the biggest one, will have two different intervals associated with it, one to the left of α and one to the right.

Using this definition we can introduce the next lemma about the properties of the optimal solution.

Lemma 4.3. *The optimal solution of (4.1) is such that the water-filling condition does hold.*

Proof. Again by contradiction: suppose that there exist an optimal solution $\{T_{nm}\}$ that violates the water-filling condition, therefore there must exist a $\bar{w} \in [\xi, f(\alpha)]$ with the corresponding x_1, x_2, K and Q such that for some $i \in K$ and $j \in Q$ $r_i \leq r_j$. Let (i, m_1) and (j, m_2) be the indices corresponding to the intervals that violate the water-filling condition. First we will consider the case where $l_{im_1} \geq l_{jm_2}$ and $\tau_{im_1} \geq \alpha$. Once more we will construct a new solution $\{T'_{nm}\}$, which is better than $\{T_{nm}\}$, moving the “heavier” interval T_{jm_2} closer to α by replacing with it the first part of the interval T_{im_1} . As this interval is going to be split in two the new solution has one more interval than the original one. We are going to switch the first part of T_{im_1} , specifically $(\tau_{im_1}, \tau_{im_1} + l_{jm_2})$, with T_{jm_2} while the second part $(\tau_{im_1} + l_{jm_2}, t_{im_1})$ will be the new extra interval in $\{T'_{nm}\}$ so

that $M'_i = M_i + 1$ (see Figure 4.4). Formally:

$$\tau'_{nm} = \begin{cases} \tau_{nm} & \text{if } n \neq i, j \text{ and } m \neq m_1, m_2, M'_i \\ \tau_{im_1} & \text{if } n = j \text{ and } m = m_2 \\ \tau_{jm_2} & \text{if } n = i \text{ and } m = m_1 \\ \tau_{im_1} + l_{jm_2} & \text{if } n = i \text{ and } m = M'_i \end{cases} \quad (4.17)$$

$$t'_{nm} = \begin{cases} t_{nm} & \text{if } n \neq i, j \text{ and } m \neq m_1, m_2, M'_i \\ \tau_{im_1} + l_{jm_2} & \text{if } n = j \text{ and } m = m_2 \\ t_{jm_2} & \text{if } n = i \text{ and } m = m_1 \\ t_{im_1} & \text{if } n = i \text{ and } m = M'_i. \end{cases} \quad (4.18)$$

As we are considering the case where $l_{jm_2} \leq l_{im_1}$ and given the fact that $l_{im_1} = t_{im_1} - \tau_{im_1}$ we have:

$$r_i \int_{\tau_{im_1}}^{t_{im_1}} f(x) dx = r_i \int_{\tau_{im_1}}^{\tau_{im_1} + l_{jm_2}} f(x) dx + r_i \int_{\tau_{im_1} + l_{jm_2}}^{t_{im_1}} f(x) dx. \quad (4.19)$$

Combining (4.19) with the fact that $\{T_{nm}\}$ and $\{T'_{nm}\}$ are identical if $n \neq i, j$ and $m \neq m_1, m_2, M'_i$ it is easy to see that by taking the difference of the two solutions we have (4.20):

$$\sum_n r_n \int_{\bigcup_m T_{nm}} f - \sum_n r_n \int_{\bigcup_m T'_{nm}} f = r_i \int_{\tau_{im_1}}^{\tau_{im_1} + l_{jm_2}} + r_i \int_{\tau_{im_1} + l_{jm_2}}^{t_{im_1}} + r_j \int_{\tau_{jm_2}}^{t_{jm_2}} \quad (4.20)$$

$$\begin{aligned} & - r_i \int_{\tau'_{im_1}}^{t'_{im_1}} - r_i \int_{\tau_{iM'_i}}^{t_{iM'_i}} - r_j \int_{\tau'_{jm_2}}^{t'_{jm_2}} \\ & = r_i \int_{\tau_{im_1}}^{\tau_{im_1} + l_{jm_2}} + r_i \int_{\tau_{im_1} + l_{jm_2}}^{t_{im_1}} + r_j \int_{\tau_{jm_2}}^{t_{jm_2}} \\ & - r_i \int_{\tau_{jm_2}}^{t_{jm_2}} - r_i \int_{\tau_{im_1} + l_{jm_2}}^{t_{im_1}} - r_j \int_{\tau_{im_1}}^{\tau_{im_1} + l_{jm_2}} \end{aligned} \quad (4.21)$$

$$\begin{aligned} & = (r_i - r_j) \int_{\tau_{im_1}}^{\tau_{im_1} + l_{jm_2}} - (r_i - r_j) \int_{\tau_{jm_2}}^{t_{jm_2}} \\ & = (r_i - r_j) \left(\int_{\tau_{im_1}}^{\tau_{im_1} + l_{jm_2}} - \int_{\tau_{jm_2}}^{t_{jm_2}} \right) \\ & \leq 0, \end{aligned} \quad (4.22)$$

in (4.11) we have used the definitions of τ'_{nm} and t'_{nm} from (4.17) and (4.18) respectively. To see why (4.22) is true consider that by the definition of x_1 and x_2 and the concavity

of f , for all $u \in [x_1, x_2]$ and v such that $v \leq x_1$ or $v \geq x_2$ it is true that $f(u) \geq f(v)$; by assumption $(\tau_{im_1}, \tau_{im_1} + l_{jm_2}) \subset [x_1, x_2]$ while (τ_{jm_2}, t_{jm_2}) is outside $[x_1, x_2]$, hence for any $u \in (\tau_{im_1}, \tau_{im_1} + l_{jm_2})$ and $v \in (\tau_{jm_2}, t_{jm_2})$ we have $f(u) \geq f(v)$ and:

$$\int_{\tau_{im_1}}^{\tau_{im_1} + l_{jm_2}} f(x) dx \geq \int_{\tau_{jm_2}}^{t_{jm_2}} f(x) dx.$$

At the same time $r_i \leq r_j$ again by assumption so that (4.22) is true and we have a contradiction because $\{T'_{nm}\}$ is a better than the optimal solution $\{T_{nm}\}$.

Incidentally if T_{im_1} and T_{jm_2} were the only two intervals violating the water-filling condition $\{T'_{nm}\}$ does satisfy the condition. Furthermore if there are other intervals that violate the condition the same construction can be used as the basis of an algorithm that will build the optimal solution starting from any solution satisfying the conditions of Lemmas 4.1 and 4.2.

So far we have considered the case where $l_{im_1} \geq l_{jm_2}$ and $\tau_{im_1} \geq \alpha$, for the case where $l_{im_1} \geq l_{jm_2}$ and $\tau_{im_1} \leq t_{im_1} \leq \alpha$ a similar argument holds with the difference that the interval (τ_{jm_2}, t_{jm_2}) is moved to $(t_{im_1} - l_{jm_2}, t_{im_1})$ that is at the end of the T_{im_1} interval while the beginning of the original interval T_{im_1} is left unchanged, in this case the previous solution (moving T_{jm_2} to the beginning of T_{im_1}) would still give a better solution but this solution would still violate the water-filling condition given that the remaining part of T_{im_1} that is $T'_{iM'_i}$ would have a lighter weight than T'_{jm_2} but it would be closer to α . If $l_{im_1} \geq l_{jm_2}$ and $\tau_{im_1} \leq \alpha \leq t_{im_1}$ T_{jm_2} should be moved to (x_3, x_4) where x_3 and x_4 are such that $x_3 \leq \alpha \leq x_4$, $x_4 - x_3 = l_{jm_2}$ and $f(x_3) = f(x_4)$, it is possible that $x_3 \leq \tau_{im_1}$, in this case T_{jm_2} should be moved to $(\tau_{im_1}, \tau_{im_1} + l_{jm_2})$, similarly if $x_4 \geq t_{im_1}$ move T_{jm_2} to $(t_{im_1} - l_{jm_2}, t_{im_1})$; note that it is never possible that $x_3 \leq \tau_{im_1}$ and at the same time $x_4 \geq t_{im_1}$ given that we are assuming $l_{im_1} \geq l_{jm_2}$.

For the case where $l_{im_1} \leq l_{jm_2}$ and $\alpha \leq \tau_{jm_2}$ a similar argument holds if we exchange $(\tau_{jm_2}, \tau_{jm_2} + l_{im_1})$ with (τ_{im_1}, t_{im_1}) while if $l_{im_1} \leq l_{jm_2}$ and $\alpha \geq t_{jm_2}$ we should switch $(t_{jm_2} - l_{im_1}, t_{jm_2})$ with (τ_{im_1}, t_{im_1}) . Note that $\tau_{jm_2} \leq \alpha \leq t_{jm_2}$ is impossible otherwise $\{T_{nm}\}$ would satisfy the water-filling condition and we are assuming it is not. \square

The last three lemmas have established necessary conditions for a solution to be the optimal one. The next lemma shows that any two solutions satisfying the water filling condition have the same cost and therefore the conditions expressed by the previous three lemmas and necessary and sufficient condition for optimality.

Lemma 4.4. *Let $\{T_{nm}\}$ and $\{T'_{nm}\}$ be two different solutions of (4.1) such that both do satisfy the water filling condition as well as Lemmas 4.1 and 4.2, then:*

$$\sum_{n=1}^N r_n \int_{\bigcup_m T_{nm}} f(x) dx = \sum_{n=1}^N r_n \int_{\bigcup_m T'_{nm}} f(x) dx,$$

that is both solutions have the same (optimal) cost.

Proof. First we will show that for any $w \in [\xi, f(\alpha)]$ both solutions are such that

$$\sum_{n=1}^N r_n \int_{(\bigcup_m T_{nm}) \cap [x_1, x_2]} f(x) dx = \sum_{n=1}^N r_n \int_{(\bigcup_m T'_{nm}) \cap [x_1, x_2]} f(x) dx \quad (4.23)$$

where x_1 and x_2 are defined as in (4.15) and (4.16), in other words the value of the sum of the integrals between x_1 and x_2 is the same for both solutions. Note that this does not imply that the solutions are identical, each solution can have a different number of intervals, but they have to distribute the weights in the same way, more precisely: $\bigcup_m T_{nm} = \bigcup_m T'_{nm}$ for every $n \in [1, N]$.

To see why (4.23) is true suppose it is not, that is the two solutions are such that there exist at least two time intervals such that each solution assigns a different weight to the two intervals. Let \bar{w} be the *biggest* value of w such that the corresponding $[x_1, x_2]$ contains one of the two intervals over which the two solutions differ. Let t_1 and t_2 be the extremes of this interval such that $x_1 \leq t_1 < t_2 \leq x_2$ and

$$\left(\bigcup_m T_{im} \right) \cap [t_1, t_2] = \left(\bigcup_m T'_{jm} \right) \cap [t_1, t_2],$$

that is solution $\{T_{nm}\}$ assigns weight r_i to the interval (t_1, t_2) while solution $\{T'_{nm}\}$ assigns weight r_j to the same interval. For the two solutions to be different it must be true that $r_i \neq r_j$, therefore either $r_i > r_j$ or $r_i < r_j$.

Given that both solutions do satisfy the constraints (4.2) and (4.3) L_i and L_j are the same for both solutions, therefore if $\{T_{nm}\}$ assigns r_i to (t_1, t_2) it means that it has to assign r_j to other intervals, *outside* (x_1, x_2) , whose total length is $t_2 - t_1$ (recall that \bar{w} is the biggest value of w such there is a difference between the two solutions so that the other differences must be *outside* (x_1, x_2)). Now suppose that $r_i < r_j$ then this would contradict the water filling condition because $r_j > r_i$ and r_j is assigned to an interval outside (x_1, x_2) while it should be assigned to one inside (x_1, x_2) . So $\{T_{nm}\}$ cannot satisfy the water filling

condition and we have a contradiction. Similarly if $r_i > r_j$ then it is $\{T'_{nm}\}$ that cannot satisfy the water filling condition. Therefore (4.23) must be true.

Given that (4.23) is true for any $w \in [\xi, f(\alpha)]$ it has to be true for $w = \xi$ but in this case $x_1 \leq \tau_{\min}$ and $x_2 \geq t_{\max}$ so that

$$\begin{aligned} \sum_{n=1}^N r_n \int_{\bigcup_m T_{nm}} f(x) dx &= \sum_{n=1}^N r_n \int_{(\bigcup_m T_{nm}) \cap [x_1, x_2]} f(x) dx \\ &= \sum_{n=1}^N r_n \int_{(\bigcup_m T'_{nm}) \cap [x_1, x_2]} f(x) dx \\ &= \sum_{n=1}^N r_n \int_{\bigcup_m T'_{nm}} f(x) dx. \end{aligned}$$

Combining this with the fact that $\{T_{nm}\}$ and $\{T'_{nm}\}$ satisfy Lemmas 4.1, 4.2 and 4.3 we have that they both have the same optimal cost. \square

4.3.2 Fixing the cross traffic and making some changes to the through traffic

Now we consider again the original problem of maximizing the worst case average delay for flow 0. We first turn our attention to the cross traffic $R_{1,\text{in}}$. In the first lemma we show that the worst case is when $R_{1,\text{in}}$ is greedy (i.e. it follows the envelope) from the beginning of each busy period. Next we turn our attention to $R_{0,\text{in}}$ but, before we can show how to modify $R_{0,\text{in}}^I(t)$ in order to increase the average delay, we need to introduce some new functions that will be used to construct the modified version of $R_{0,\text{in}}^I(t)$. We will also show that these functions have several properties that will be exploited later on.

Lemma 4.5. *Given any (fixed) $R_{0,\text{in}}^I(t)$, $R_{1,\text{in}}^I(t)$ and \bar{y} ; let τ be the beginning of the system busy period containing $t_{0,\text{in}}(\bar{y})$ (if $t_{0,\text{in}}(\bar{y})$ is not contained in any busy period let $\tau = t_{\text{in}}(\bar{y})$) also let $R_{0,\text{in}}^{II}(t) = R_{0,\text{in}}^I(t)$ and*

$$R_{1,\text{in}}^{II}(t) = \begin{cases} R_{1,\text{in}}^{II}(t), & t < \tau \\ E_1(t - \tau) + R_{1,\text{in}}^I(\tau), & \tau \leq t \leq t_{0,\text{in}}(\bar{y}) \end{cases} \quad (4.24)$$

(given that we are interested in $d_0(\bar{y})$ how we define $R_{1,\text{in}}(t)$ for $t > t_{0,\text{in}}(\bar{y})$ is irrelevant). Then $d_0^I(\bar{y}) \leq d_0^{II}(\bar{y})$.

Proof. For the sake of simplicity we will assume that all input process are continuous processes, the following argument holds for the non continuous case as well but the notation is somewhat cumbersome. Let $b^I(t)$ and $b^{II}(t)$ be the total backlog (from both flows) at time t for the two scenarios described above. By definition of backlog for any $\tau \leq t \leq t_{0,\text{in}}(\bar{y})$ we have $b(t) = R_{0,\text{in}}(t) - R_{0,\text{in}}(\tau) + R_{1,\text{in}}(t) - R_{1,\text{in}}(\tau) - C(t - \tau)$ so that

$$\begin{aligned} b^I(t) &= R_{0,\text{in}}^I(t) - R_{0,\text{in}}^I(\tau) + R_{1,\text{in}}^I(t) - R_{1,\text{in}}^I(\tau) - C(t - \tau) \\ &\leq R_{0,\text{in}}^I(t) - R_{0,\text{in}}^I(\tau) + E_1(t - \tau) - C(t - \tau) \\ &= R_{0,\text{in}}^{II}(t) - R_{0,\text{in}}^{II}(\tau) + R_{1,\text{in}}^{II}(t) - R_{1,\text{in}}^{II}(\tau) - C(t - \tau) \\ &= b^{II}(t). \end{aligned} \tag{4.25}$$

The first inequality follows from the fact that $R_{1,\text{in}}$ has envelope E_1 and the last two equalities follow from the definitions of $R_{1,\text{in}}^{II}$ and b^{II} . Under the continuity assumption for $R_{0,\text{in}}$ we have $d_0(\bar{y}) = b(t_{0,\text{in}}(\bar{y}))/C$ so that (4.25) implies $d_0^I(\bar{y}) \leq d_0^{II}(\bar{y})$. \square

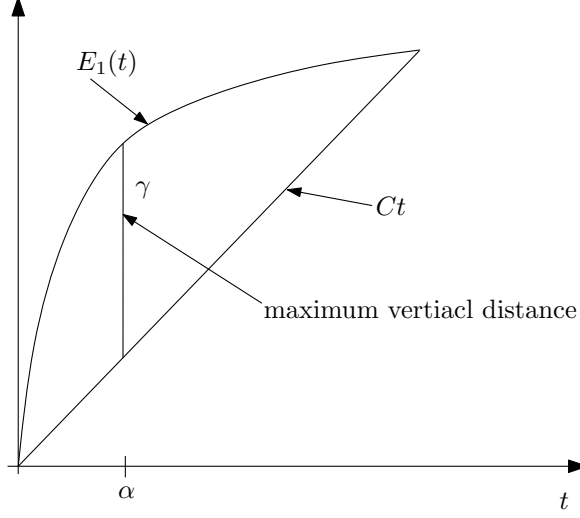
Note that $R_{1,\text{in}}^{II}$ as defined in (4.24) might be inconsistent with the envelope E_1 but it can still be used to obtain a, possibly non-achievable, upper bound on the average delay. Later we will show that it is actually always possible to construct arrival processes such that these bounds are achieved for all values of \bar{y} . Also note that by changing $R_{1,\text{in}}^I$ to $R_{1,\text{in}}^{II}$ we might increase the length of the busy period containing \bar{y} by joining the busy period containing \bar{y} in the original scenario (I) with one or more of the busy periods immediately following it. At the same time we are not changing the beginning of the busy period, only its size. Furthermore given that the system is stable the size of any busy period is bounded.

Next we need to introduce some new definitions. First of all we need to assume that $E_1(t)$ is concave for $t \geq 0$, this is not a terribly restrictive hypothesis considering that some of the most frequently used envelopes like “sigma-rho” envelopes (with or without an additional maximum rate constraint) do satisfy this condition. For concave envelopes such that $\lim_{t \rightarrow \infty} (E_1(t) - Ct) < 0^2$ the following quantities are always well defined (see Figure 4.5):

$$\gamma = \sup_{t \geq 0} \{E_1(t) - Ct\}, \quad \alpha = \inf \{t \geq 0 : E_1(t) - Ct \geq \gamma\}. \tag{4.26}$$

If $E_1(t) \leq Ct$ for $t \geq 0$ let $\gamma = \alpha = 0$.

²That is if E_1 is the only input to a fixed rate server with capacity C the system is stable.

Figure 4.5: Definition of α and γ

Based on the definition of γ and α it is possible to show the following lemma that will be useful later on.

Lemma 4.6. *Let γ and α be defined as in (4.26) and let w be such that $E_1(w) = Cw$ (because the system is stable w is always well defined) then for any t_1, t_2 such that $0 \leq t_1 \leq t_2 \leq \alpha$ we have that $E_1(t_1) - Ct_1 \leq E_1(t_2) - Ct_2$, conversely for any t_3, t_4 such that $\alpha \leq t_3 \leq t_4 \leq w$ we have that $E_1(t_3) - Ct_3 \geq E_1(t_4) - Ct_4$.*

Proof. See Figure 4.6. From the definition of γ it follows that $E_1(\alpha) - C\alpha \geq E_1(t_1) - Ct_1$ and hence

$$E_1(\alpha) - E_1(t_1) \geq C\alpha - Ct_1, \quad (4.27)$$

while from the fact that E_1 is concave we have that $E_1((1-\lambda)t_1 + \lambda\alpha) \geq (1-\lambda)E_1(t_1) - \lambda E_1(\alpha) \forall \lambda \in (0, 1)$. For any t_2 such that $t_1 \leq t_2 \leq \alpha$ we can set $\lambda = \frac{t_2 - t_1}{\alpha - t_1}$ (note that because $t_1 \leq t_2 \leq \alpha$ $\lambda \in (0, 1)$) to obtain:

$$\begin{aligned} E_1(t_2) &\geq E_1(t_1) + \frac{t_2 - t_1}{\alpha - t_1} [E_1(\alpha) - E_1(t_1)] \\ &\geq E_1(t_1) + \frac{t_2 - t_1}{\alpha - t_1} C(\alpha - t_1) \\ &= E_1(t_1) + Ct_2 - Ct_1 \end{aligned} \quad (4.28)$$

where the first inequality follows from the concavity of E_1 and the second inequality follows from (4.27). Finally rearranging (4.28) we have $E_1(t_1) - Ct_1 \leq E_1(t_2) - Ct_2$.

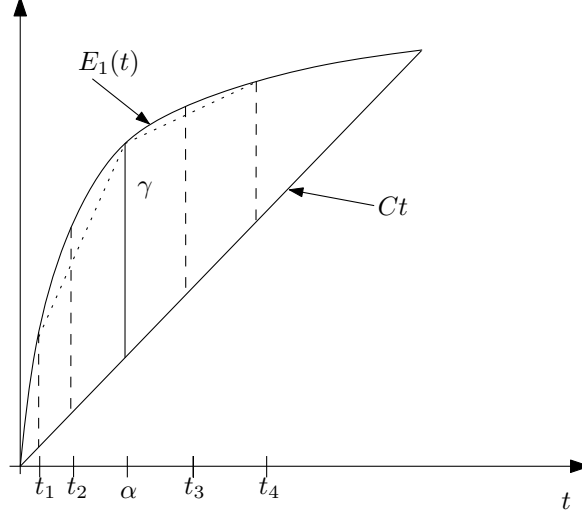


Figure 4.6: $E_1(t_1) - Ct_1 \leq E_1(t_2) - Ct_2$ and $E_1(t_3) - Ct_3 \geq E_1(t_4) - Ct_4$

Similarly from the definition of γ we have (4.27) $E_1(\alpha) - C\alpha \geq E_1(t_4) - Ct_4$ so that

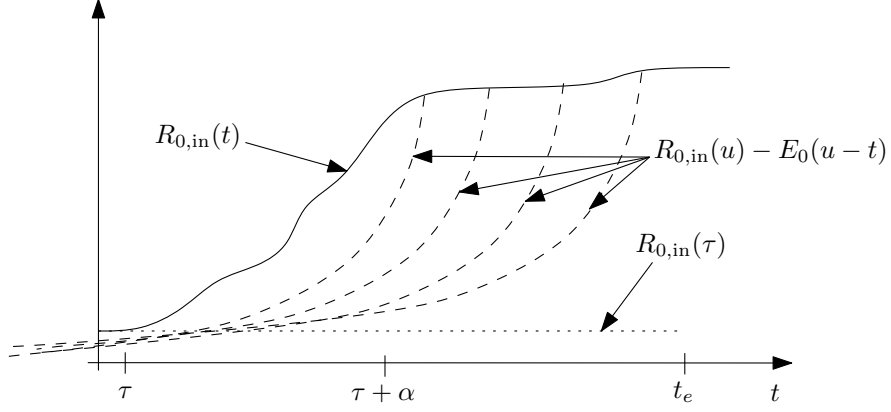
$$E_1(t_4) - E_1(\alpha) \leq Ct_4 - C\alpha, \quad (4.29)$$

from the concavity of E_1 we have that $E_1(\lambda\alpha + (1-\lambda)t_4) \geq \lambda E_1(\alpha) + (1-\lambda)E_1(t_4)$ $\forall \lambda \in (0, 1)$, note that in this case we use λ as the coefficient for the left end-point (α) and $1-\lambda$ as the coefficient for the right end-point (t_4). For any t_3 such that $\alpha \leq t_3 \leq t_4$ we can set $\lambda = \frac{t_4 - t_3}{t_4 - \alpha}$ (note that because $\alpha \leq t_3 \leq t_4$ $\lambda \in (0, 1)$) to obtain:

$$\begin{aligned} E_1(t_3) &\geq E_1(t_4) - \frac{t_4 - t_3}{t_4 - \alpha} [E_1(t_4) - E_1(\alpha)] \\ &\geq E_1(t_4) - \frac{t_4 - t_3}{t_4 - \alpha} C(t_4 - \alpha) \\ &= E_1(t_4) - Ct_4 + Ct_3 \end{aligned} \quad (4.30)$$

where the first inequality follows from the concavity of E_1 and the second inequality follows from (4.29). Finally rearranging (4.30) we have $E_1(t_4) - Ct_4 \leq E_1(t_3) - Ct_3$. \square

Under the concavity assumption for E_1 the next lemmas shows that it is possible to increase the average delay if we force all the traffic from flow $R_{0,\text{in}}$ to concentrate around $\tau + \alpha$, where τ is the beginning of the system busy period being examined. The idea is to keep constant the amount of traffic that arrives before and after time $\tau + \alpha$: the traffic that arrives before $\tau + \alpha$ it is moved so that $R_{0,\text{in}}^{\text{III}}$ follows a “partial upside down inverted

Figure 4.7: Construction of $G(t)$

envelope” rooted at $R_{0,\text{in}}^{\text{II}}(\tau + \alpha)$, while the traffic that arrives after that time it is moved so that $R_{0,\text{in}}^{\text{II}}$ sends as much traffic as possible without violating the envelope. In other words we are not changing the amount of traffic sent by $R_{0,\text{in}}^{\text{II}}$ we are simply moving it around so that it arrives as close as possible to time $\tau + \alpha$ without violating the envelope.

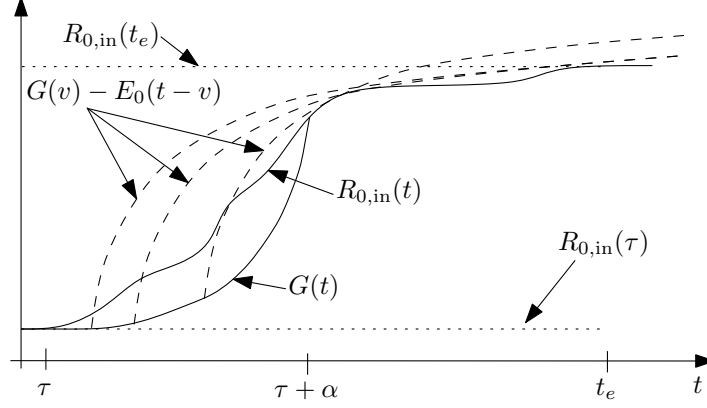
We first need to define two functions that will be used to construct the modified $R_{0,\text{in}}$. The first function deals with the traffic that arrives between τ and $\tau + \alpha$, as such we do not need to define it for any other values of t :

$$G(t) = \max \left\{ \sup_{\tau + \alpha \leq u \leq t_e} \{R_{0,\text{in}}^{\text{II}}(u) - E_0(u - t)\}, R_{0,\text{in}}^{\text{II}}(\tau) \right\} \quad \forall t \in [\tau, \tau + \alpha] \quad (4.31)$$

where t_e is the end of the busy starting at τ , when the inputs are $R_{0,\text{in}}^{\text{II}}$ and $R_{1,\text{in}}^{\text{II}}$, see Figure 4.7.

The idea behind the definition of G is that we can use the values of $R_{0,\text{in}}^{\text{II}}(t)$ *after* time $\tau + \alpha$ to find a lower bound for the values for $R_{0,\text{in}}^{\text{II}}(t)$ *before* time $\tau + \alpha$. Given that $R_{0,\text{in}}^{\text{II}}(t)$ has envelope $E_0(t)$ (recall that $R_{0,\text{in}}^{\text{II}}(t) = R_{0,\text{in}}^{\text{I}}(t)$) we know that for any $t_1 \leq t_2$ we have $R_{0,\text{in}}^{\text{II}}(t_2) - R_{0,\text{in}}^{\text{II}}(t_1) \leq E_0(t_2 - t_1)$ and hence $R_{0,\text{in}}^{\text{II}}(t_1) \geq R_{0,\text{in}}^{\text{II}}(t_2) - E_0(t_2 - t_1)$, letting $t_1 \leq \tau + \alpha \leq t_2$ and taking the supremum over all $t_2 \in [\tau + \alpha, t_e]$ we have (4.31).

Now that we have a lower bound for $R_{0,\text{in}}^{\text{II}}(t)$ for $t \in [\tau, \tau + \alpha]$ we can use it to build an upper bound for $R_{0,\text{in}}^{\text{II}}(t)$ for $t \in [\tau + \alpha, t_e]$. The idea is the same we used for G , given that $R_{0,\text{in}}^{\text{II}}(t)$ has envelope E_0 we can use it to find an upper bound for $R_{0,\text{in}}^{\text{II}}(t)$ using $G(t)$

Figure 4.8: Construction of $H(t)$

as the starting point. For any $t \in [\tau + \alpha, t_e]$ we can define $G(t)$ as (see Figure 4.8):

$$H(t) = \min \left\{ \inf_{\tau \leq v \leq \tau + \alpha} \{G(v) + E_0(t - v)\}, R_{0,\text{in}}^{\text{II}}(t_e) \right\} \quad \forall t \in [\tau + \alpha, t_e]. \quad (4.32)$$

The following lemma shows that, indeed, $G(t)$ is a lower bound for $R_{0,\text{in}}^{\text{II}}(t)$ while $H(t)$ is an upper bound.

Lemma 4.7. *Let $G(t)$ and $H(t)$ be defined as in (4.31) and (4.32), respectively, then $\forall t \in [\tau, \tau + \alpha]$ $G(t) \leq R_{0,\text{in}}^{\text{II}}(t)$ while $\forall t \in [\tau + \alpha, t_e]$ $R_{0,\text{in}}^{\text{II}}(t) \leq H(t)$.*

Proof. Let us first consider the case where $G(t) = R_{0,\text{in}}^{\text{II}}(\tau)$: given that $R_{0,\text{in}}^{\text{II}}(t) \geq R_{0,\text{in}}^{\text{II}}(\tau)$ for $t \in [\tau, t_e]$ the claim is true. If $G(t) \neq R_{0,\text{in}}^{\text{II}}(\tau)$ from the definition of $G(t)$ we have that $\forall \varepsilon > 0$ there exist a $u^* \in [\tau + \alpha, t_e]$ such that $G(t) - \varepsilon \leq R_{0,\text{in}}^{\text{I}}(u^*) - E_0(u^* - t)$ and hence:

$$\begin{aligned} G(t) &\leq R_{0,\text{in}}^{\text{II}}(u^*) - E_0(u^* - t) + \varepsilon \\ &\leq R_{0,\text{in}}^{\text{II}}(t) + \varepsilon. \end{aligned} \quad (4.33)$$

For (4.33) we have used the fact that $R_{0,\text{in}}^{\text{II}}(t)$ has envelope $E_0(t)$ and therefore $R_{0,\text{in}}^{\text{II}}(u^*) - R_{0,\text{in}}^{\text{II}}(t) \leq E_0(u^* - t)$, finally the claim follows from the fact the ε is arbitrary.

Turning our attention to $H(t)$ we have that $R_{0,\text{in}}^{\text{II}}(t) \leq R_{0,\text{in}}^{\text{II}}(t_e)$ for $t \in [\tau, t_e]$ so that the claim is true if $H(t) = R_{0,\text{in}}^{\text{II}}(t_e)$. If this is not the case from the definition of $H(t)$ we have that $\forall \varepsilon > 0$ there exist a $v^* \in [\tau, \tau + \alpha]$ such that $G(v^*) + E_0(t - v^*) \leq H(t) + \varepsilon$ so

that

$$\begin{aligned}
H(t) &\geq G(v^*) + E_0(t - v^*) - \varepsilon \\
&\geq R_{0,\text{in}}^{\text{II}}(t) - E_0(t - v^*) + E_0(t - v^*) - \varepsilon \\
&\geq R_{0,\text{in}}^{\text{II}}(t) - \varepsilon.
\end{aligned} \tag{4.34}$$

To see why (4.34) is true consider that from the definition of G we have

$$\begin{aligned}
G(v^*) &= \sup_{\tau + \alpha \leq u \leq t_e} \{R_{0,\text{in}}^{\text{II}}(u) - E_0(u - v^*)\} \\
&\geq R_{0,\text{in}}^{\text{II}}(t) - E_0(t - v^*),
\end{aligned}$$

given that in this case $t \in [\tau + \alpha, t_e]$. Again the final claim follows from the fact that ε is arbitrary. \square

As in Lemma 4.5 we consider two scenarios (II and III) with the first one (II) equal to the second scenario in Lemma 4.5. In this new scenario we use G and H to define $R_{0,\text{in}}^{\text{III}}(t)$ as follows:

$$R_{0,\text{in}}^{\text{III}}(t) = \begin{cases} R_{0,\text{in}}^{\text{I}}(t), & \text{if } t \leq \tau \\ G(t), & \text{if } \tau \leq t < \alpha \\ R_{0,\text{in}}^{\text{I}}(\alpha), & \text{if } t = \alpha \\ H(t), & \text{if } \alpha < t \leq t_e. \end{cases} \tag{4.35}$$

Before we move to the lemma showing that the average delay increases if we change $R_{0,\text{in}}^{\text{II}}(t)$ into $R_{0,\text{in}}^{\text{III}}(t)$ we will present two lemmas that deal with properties of $R_{0,\text{in}}^{\text{III}}(t)$ that will be used later on. The first lemma shows that, as long as $t \in [\tau, t_e]$, $R_{0,\text{in}}^{\text{III}}(t)$ does satisfy the envelope $E_0(t)$. The second lemma shows that if we divide $R_{0,\text{in}}^{\text{III}}(t)$ for $t \in [\alpha, t_e]$ into several intervals and we consider an arbitrary permutation of these intervals the resulting function will still satisfy the envelope.

Note that this does not imply that $R_{0,\text{in}}^{\text{III}}(t)$ does satisfy the envelope for all $t \geq 0$, given that there might exist $t_1 \leq \tau \leq t_2$ such that $R_{0,\text{in}}^{\text{III}}(t_2) - R_{0,\text{in}}^{\text{III}}(t_1) > E_0(t_2 - t_1)$ but this does not prevent us from using $R_{0,\text{in}}^{\text{III}}(t)$ as a bound as shown in Lemma 4.7. Later on we will use the fact that $R_{0,\text{in}}^{\text{III}}(t)$ does not violate the envelope between τ and t_e to build a specific arrival pattern such that $R_{0,\text{in}}^{\text{III}}(t)$ does conform to the envelope for all values of t .

Lemma 4.8. *Let $R_{0,\text{in}}^{\text{III}}(t)$ be define as in (4.35) then for any t_1 and t_2 such that $\tau \leq t_1 \leq t_2 \leq t_e$ we have $R_{0,\text{in}}^{\text{III}}(t_2) - R_{0,\text{in}}^{\text{III}}(t_1) \leq E_0(t_2 - t_1)$. Furthermore there exist τ' and t'_e such*

that $\tau \leq \tau'$, $t'_e \leq t_e$, $G(t) = R_{0,in}^{II}(t)$ if $\tau \leq t \leq \tau'$, $H(t) = R_{0,in}^{II}(t_e)$ if $t'_e \leq t \leq t_e$ and $R_{0,in}^{III}(t'_e) - R_{0,in}^{III}(\tau') = R_{0,in}^{II}(t_e) - R_{0,in}^{II}(\tau) = E_0(t'_e - \tau')$.

Proof. Let us start with the case where $\tau \leq t_1 \leq t_2 \leq \tau + \alpha$ this implies that $R_{0,in}^{III}(t_i) = G(t_i)$, $i = 1, 2$. Using the definition of $G(t)$ and the fact that $R_{0,in}^{II}(t)$ does satisfy the envelope we have:

$$G(t_2) - G(t_1) \leq R_{0,in}^{II}(u_2) - E_0(u_2 - t_2) + \varepsilon - G(t_1) \quad (4.36)$$

$$\leq R_{0,in}^{II}(u_2) - E_0(u_2 - t_2) + \varepsilon - R_{0,in}^{II}(u_2) + E_0(u_2 - t_1) \quad (4.37)$$

$$\begin{aligned} &= E_0(u_2 - t_1) - E_0(u_2 - t_2) + \varepsilon \\ &\leq E_0(t_2 - t_1) + \varepsilon, \end{aligned} \quad (4.38)$$

where from the definition of $G(t)$ (4.32) we know that for every $\varepsilon > 0$ there exist a u_2 such that $R_{0,in}^{II}(u_2) - E_0(u_2 - t_2) \geq G(t_2) - \varepsilon$ and hence (4.36). From the definition of $G(t)$ we also have that $G(t_1) \geq R_{0,in}^{II}(u_2) - E_0(u_2 - t_1)$ and therefore (4.37). For (4.38) we have used the fact that E_0 is concave and therefore sub-additive so that $E_0(u_2 - t_1) - E_0(u_2 - t_2) \leq E_0(t_2 - t_1)$.

If $t_1 \leq \tau + \alpha \leq t_2$ then $R_{0,in}^{III}(t_1) = G(t_1)$ and $R_{0,in}^{III}(t_2) = H(t_2)$, from the definition of $G(t)$ (4.32) by choosing $v = t_1$ we have that

$$H(t_2) \leq G(t_1) + E_0(t_2 - t_1),$$

which implies that $H(t_2) - G(t_1) \leq E_0(t_2 - t_1)$. Finally the case where $\alpha + \tau \leq t_1 \leq t_2$ is very similar to the first case:

$$H(t_2) - H(t_1) \leq H(t_2) - G(v_1) - E_0(t_1 - v_1) + \varepsilon \quad (4.39)$$

$$\leq G(v_1) + E_0(t_2 - v_1) - G(v_1) - E_0(t_1 - v_1) + \varepsilon \quad (4.40)$$

$$\begin{aligned} &= E_0(t_2 - v_1) - E_0(t_1 - v_1) + \varepsilon \\ &\leq E_0(t_2 - t_1) + \varepsilon \end{aligned} \quad (4.41)$$

where (4.39) follows from the definition of H ($\forall \varepsilon > 0$ there exist a v_1 such that $G(v_1) + E_0(t_1 - v_1) \leq H(t_1) + \varepsilon$), again using the definition of H and choosing $v = v_1$ we have $H(t_2) \leq G(v_1) + E_0(t_2 - v_1)$ and hence (4.40). For (4.41) we have used the fact the E_0 is sub-additive.

For the second part of the claim consider that $\forall u \in [\tau + \alpha, t_e]$ and $\forall t \in [\tau, \tau + \alpha]$ $R_{0,in}^{II}(u) - E_0(u - t)$ is a non-decreasing function; given that $G(t)$ is the supremum of

non-decreasing functions it is non-decreasing as well, a similar argument holds for $H(t)$ as well given that it is the infimum of non-decreasing functions. Given that $G(t)$ and $H(t)$ are monotone functions we can define

$$\tau' = \inf_{\tau \leq u \leq \tau + \alpha} \{G(u) \geq R_{0,\text{in}}^{\text{II}}(\tau)\} \quad (4.42)$$

$$t'_e = \sup_{\tau + \alpha \leq u \leq t_e} \{H(u) \leq R_{0,\text{in}}^{\text{II}}(t_e)\}. \quad (4.43)$$

From the definitions of H and G it follows immediately that $G(t) = R_{0,\text{in}}^{\text{II}}(t)$ if $\tau \leq t \leq \tau'$ and that $H(t) = R_{0,\text{in}}^{\text{II}}(t_e)$ if $t'_e \leq t \leq t_e$. As a consequence of this $\forall v \in [\tau, \tau']$ we have:

$$G(v) + E_0(\tau' + T - v) = G(\tau') + E_0(\tau' + T - v) \quad (4.44)$$

$$\geq G(\tau') + E_0(T) \quad (4.45)$$

where (4.44) follows from the fact that $G(v)$ is constant for $v \in [\tau, \tau']$ while $v \leq \tau'$ and the fact that E_0 is non-decreasing imply (4.45). If $\tau' \leq v \leq \tau + \alpha$ the same is true, to see why this is the case we first need to consider the following inequalities:

$$G(v) - G(\tau') \geq G(v) - R_{0,\text{in}}^{\text{II}}(u^*) + E_0(u^* - \tau') - \varepsilon \quad (4.46)$$

$$\geq R_{0,\text{in}}^{\text{II}}(u^*) - E_0(u^* - v) - R_{0,\text{in}}^{\text{II}}(u^*) + E_0(u^* - \tau') - \varepsilon \quad (4.47)$$

$$= E_0(u^* - \tau') - E_0(u^* - v) - \varepsilon \quad (4.48)$$

$$\geq E_0(\theta) - E_0(\tau' + \theta - v) - \varepsilon. \quad (4.49)$$

From the definition of $G(t)$ (4.32) we know that for every $\varepsilon > 0$ there exist a u^* such that $R_{0,\text{in}}^{\text{II}}(u^*) - E_0(u^* - \tau') \geq G(\tau') - \varepsilon$ and hence (4.46). Again from the definition of $G(t)$ we also have that $G(v) \geq R_{0,\text{in}}^{\text{II}}(u^*) - E_0(u^* - v)$ and therefore (4.47). Given that E_0 is concave for $t > 0$ if $u_1 \geq u_2$ then $E_0(u_1 - \tau') - E_0(u_1 - v) \geq E_0(u_2 - \tau') - E_0(u_2 - v)$, that is the bigger values of u^* give smaller values of (4.48). By the definition of τ' (4.42) it is easy to see that $u^* - \tau' \leq \theta$ where

$$\theta = E_0^{-1}(R_{0,\text{in}}^{\text{II}}(t_e) - R_{0,\text{in}}^{\text{II}}(\tau)), \quad (4.50)$$

if this is not the case, i.e. if $u^* - \tau' > \theta$, then $G(\tau') = R_{0,\text{in}}^{\text{II}}(u^*) - E_0(u^* - \tau') \leq R_{0,\text{in}}^{\text{II}}(\tau)$ (given that $R_{0,\text{in}}^{\text{II}}(u^*) - R_{0,\text{in}}^{\text{II}}(\tau) \leq R_{0,\text{in}}^{\text{II}}(t_e) - R_{0,\text{in}}^{\text{II}}(\tau) \leq E_0(\theta)$) which contradicts the definition of τ' . This implies that (4.48) is minimized when $u^* = \tau' + \theta$ and hence (4.49).

Now that we have established (4.49) we can write:

$$G(v) + E_0(\tau' + \theta - v) \geq E_0(\theta) - E_0(\tau' + \theta - v) + G(\tau') + E_0(\tau' + \theta - v) \quad (4.51)$$

$$= E_0(\theta) + G(\tau') \quad (4.52)$$

where (4.51) follows from (4.49) and (4.52) is the same as (4.45). Therefore we have established that (4.45) holds for all values of $v \in [\tau, \tau + \alpha]$. Combining this with the definition of $H(t)$ we have:

$$H(\tau' + \theta) = \inf_{\tau \leq v \leq \tau + \alpha} \{G(v) + E_0(\tau' + \theta - v)\} \quad (4.53)$$

$$\geq E_0(\theta) + G(\tau') \quad (4.54)$$

$$= E_0(\theta) + R_{0,\text{in}}^{\text{II}}(\tau) \quad (4.55)$$

$$= R_{0,\text{in}}^{\text{II}}(t_e) \quad (4.56)$$

where (4.53) is the definition of $H(t)$, (4.54) follows from (4.45) and (4.52); (4.55) and (4.56) follow from the definitions of τ' (4.42) and θ (4.50) respectively. Combining (4.56) with the definition of t'_e (4.43) we have that $\tau' + \theta \geq t'_e$ that is $t'_e - \tau' \leq \theta$. However, at the beginning of this proof, we have shown that $R_{0,\text{in}}^{\text{III}}(t)$, which is the concatenation of G and H , does not violate the envelope, this implies that $t'_e - \tau' \geq \theta$. Therefore we can conclude that $t'_e - \tau' = \theta$. To show that $R_{0,\text{in}}^{\text{III}}(t'_e) - R_{0,\text{in}}^{\text{III}}(\tau') = R_{0,\text{in}}^{\text{II}}(t_e) - R_{0,\text{in}}^{\text{II}}(\tau)$ we can consider the following equalities:

$$R_{0,\text{in}}^{\text{III}}(t'_e) - R_{0,\text{in}}^{\text{III}}(\tau') = H(t'_e) - G(\tau') \quad (4.57)$$

$$= R_{0,\text{in}}^{\text{II}}(t_e) - G(\tau') \quad (4.58)$$

$$= R_{0,\text{in}}^{\text{II}}(t_e) - R_{0,\text{in}}^{\text{II}}(\tau) \quad (4.59)$$

where (4.57) follows from the definitions of $R_{0,\text{in}}^{\text{III}}(t)$ (4.35), combining (4.56) with the fact that $H(t) \leq R_{0,\text{in}}^{\text{II}}(t_e)$ (this follows immediately from the definition of H (4.31)) we have that $H(\tau' + \theta) = H(t'_e) = R_{0,\text{in}}^{\text{I}}(t_e)$ and hence (4.58). From the definition of τ' it follows that $G(\tau') = R_{0,\text{in}}^{\text{II}}(\tau)$ and hence (4.59). Finally from the definition of θ (4.50) we have $R_{0,\text{in}}^{\text{II}}(t_e) - R_{0,\text{in}}^{\text{II}}(\tau) = E_0(t'_e - \tau')$ and this concludes the proof. \square

For the following lemma we need to consider what would happen if we divide $[\tau, t_e]$ into N intervals $T_n = [\tau_n, t_n]$, associating the corresponding section of $R_{0,\text{in}}^{\text{III}}(t)$ with each interval and then rearrange these sections to construct a new realization of $R_{0,\text{in}}(t)$. The idea is to cut $R_{0,\text{in}}^{\text{III}}(t)$ into different pieces and then recombine them in a different order with the constraint that the beginning of each piece has to coincide with the end of the one that comes immediately before. In other words we can define

$$\tilde{f}_n(t) = \begin{cases} R_{0,\text{in}}^{\text{III}}(t + \tau_n) - R_{0,\text{in}}^{\text{III}}(\tau_n), & \text{if } 0 \leq t \leq \delta_n \\ 0, & \text{otherwise,} \end{cases}$$

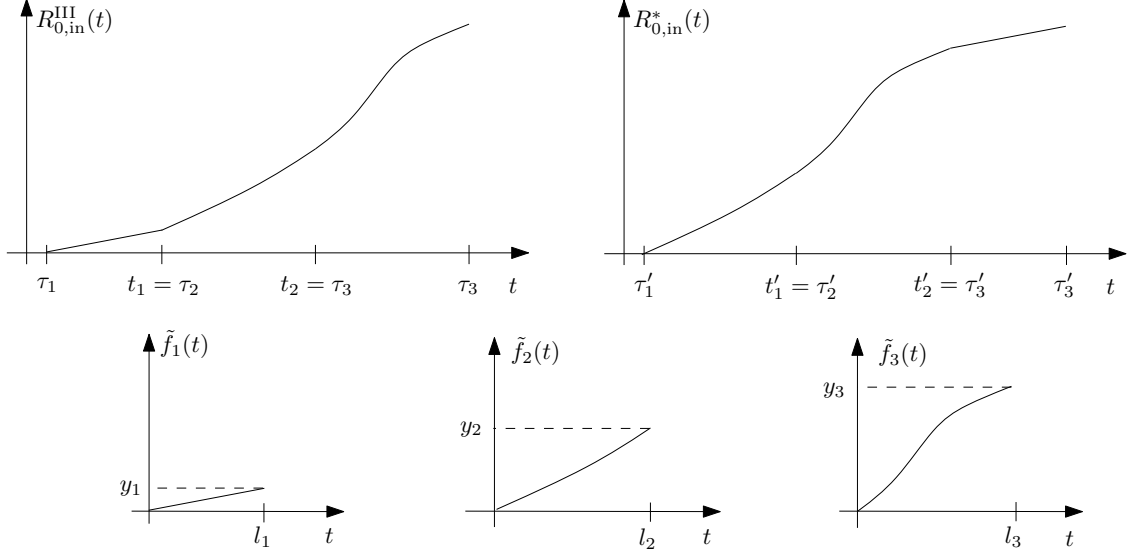


Figure 4.9: A sample $R_{0,\text{in}}^{\text{III}}(t)$ and $R_{0,\text{in}}^*(t)$

where $l_n = t_n - \tau_n$. One way of describing $\tilde{f}_n(t)$ is the n -th section of $R_{0,\text{in}}^{\text{III}}(t)$ shifted vertically so that $\tilde{f}_n(\tau_n) = 0$ and horizontally to the origin. We are also going to assume that $\bigcup_n T_n = [\tau, t_e]$ and define $y_n = R_{0,\text{in}}^{\text{III}}(t_n) - R_{0,\text{in}}^{\text{III}}(\tau_n)$ as the increment of $R_{0,\text{in}}^{\text{III}}(t)$ over the n -th interval. Using this partition we can construct a new version of $R_{0,\text{in}}^{\text{III}}(t)$, let us call it $R_{0,\text{in}}^*(t)$, by changing the order of the intervals with the corresponding \tilde{f}_n . More precisely let $T'_i = [\tau'_i, t'_i]$ ($i = 1, \dots, N$) be such that $\bigcup_i [\tau'_i, t'_i] = [\alpha, t_e]$ and for every i $l_i = l_n$ for some n , that is the new set of intervals is simply a permutation of the original ones with, let $\pi(n) = i$ be this permutation, that is the interval that was in position n in the original partition is now going to be in position $\pi(n)$. We can construct the new process by defining how each $\tilde{f}_n(t)$ is moved, let

$$h_{i=\pi(n)}^* = \begin{cases} \tilde{f}_n(t - \tau_{\pi(n)}) + \sum_{k=1}^{\pi(n)} y_{\pi(k)}, & \text{if } \tau_{\pi(n)} \leq t \leq t_{\pi(n)} \\ 0, & \text{otherwise.} \end{cases}$$

Using h_i^* we can construct $R_{0,\text{in}}^*(t)$ as

$$R_{0,\text{in}}^*(t) = \sum_{i=1}^N h_i^*(t), \quad (4.60)$$

see Figure 4.9 for an example. The next lemma shows that no matter how we rearrange these intervals the resulting $R_{0,\text{in}}^*(t)$ will still satisfy the envelope.

Lemma 4.9. *Let $R_{0,\text{in}}^*(t)$ be defined as in (4.60), if $E_0(t)$ is concave then for any permutation π it is always true that $R_{0,\text{in}}^*(t) \sim E_0(t)$.*

Proof. From the definition of $R_{0,\text{in}}^{\text{III}}(t)$ (4.35) we know that it is the concatenation of $G(t)$ and $H(t)$, furthermore G is the supremum of convex functions (E_0 is concave so that $-E_0$ is convex) and therefore convex itself. While H is concave given that it is the infimum of concave functions. This implies that $R_{0,\text{in}}^{\text{III}}(t)$ is convex for $t \in [\tau, \tau + \alpha]$ and concave for $t \in [\tau + \alpha, t_e]$.

Let t_1 and t_2 be such that $\tau \leq t_1 \leq t_2 \leq t_e$, we would like to show that $R_{0,\text{in}}^*(t_2) - R_{0,\text{in}}^*(t_1) \leq E_0(t_2 - t_1)$. Let $A = \{n : t_n \leq \tau + \alpha\}$ that is the set of all n such that the corresponding interval is to the left of $\tau + \alpha$ in the original permutation. For the sake of simplicity we will assume that $t_j = \tau_{j+1} = \tau + \alpha$ for some $j \in [1, N]$. If this is not the case it is always possible to divide the interval containing $\tau + \alpha$ in two new intervals constructing a new partition that does satisfy this condition. Similarly let $B = \{n : \tau_n \geq \tau + \alpha\}$. All the intervals such that $t_1 \leq t'_i$ and $\tau'_i \leq t_2$ are such that either $\pi^{-1}(i) \in A$ or $\pi^{-1}(i) \in B$, let $A' = \{i : \pi^{-1}(i) \in A\}$ and $B' = \{i : \pi^{-1}(i) \in B\}$ in other words A' contains the indices of all the intervals that were to the left of $\tau + \alpha$ in the original permutation and that now are between t_1 and t_2 , let L_1 be the total length of all these intervals; similarly B' contains the indices of all the intervals that were to the right of $\tau + \alpha$ in the original permutation and that now are between t_1 and t_2 , let L_2 be the their total length. Formally:

$$L_1 = \sum_{k \in A'} l_k$$

$$L_2 = \sum_{k \in B'} l_k.$$

Given that $\tau_{m_1} \leq t_1 \leq t_{m_1}$ and $\tau_{m_2} \leq t_2 \leq t_{m_2}$ for some m_1 and m_2 , we can write:

$$R_{0,\text{in}}^*(t_2) - R_{0,\text{in}}^*(t_1) = \sum_{k \in A' / (m_1, m_2)} y_k + \sum_{j \in B' / (m_1, m_2)} y_j$$

$$+ \tilde{f}_{m_1}(l_{m_1}) - \tilde{f}_{m_1}(t_{m_1} - t_1) + \tilde{f}_{m_2}(t_2 - \tau_{m_2}) - \tilde{f}_{m_2}(\tau_{m_2}),$$

that is $R_{0,\text{in}}^*(t_2) - R_{0,\text{in}}^*(t_1)$ is the sum of all the y_k of the intervals completely between t_1 and t_2 plus a part of the increment over the two extreme intervals, containing t_1 and t_2 themselves. As all the y_k come from intervals that were to the left of $\tau + \alpha$ in the original permutation and whose total length is L_1 we can upper bound their sum by observing

that they all came from G , which is a non-decreasing convex function therefore $\sum y_k \leq G(\tau + \alpha) - G(\tau + \alpha - L_1)$. Similarly we can exploit the fact that all the y_j come from intervals that were to the right of $\tau + \alpha$ and whose total length is L_2 . These intervals all came from H , which is non-decreasing and concave, therefore $\sum y_j \leq H(\tau + \alpha + L_2) - H(\tau + \alpha)$. Note that m_1 and m_2 belong either to A' or B' so that these upper bounds hold for the partial increments at both ends. Therefore, letting $\delta_1 = t_{m_1} - t_1$ and $\delta_2 = t_2 - \tau_{m_2}$ we have:

$$\begin{aligned} R_{0,\text{in}}^*(t_2) - R_{0,\text{in}}^*(t_1) &\leq R_{0,\text{in}}^{\text{III}}(\tau + \alpha + L_2 - \delta_2) - R_{0,\text{in}}^{\text{III}}(\tau + \alpha - L_1 + \delta_1) \\ &\leq E_0(L_2 - \delta_2 + L_1 - \delta_1) \end{aligned} \quad (4.61)$$

$$= E_0(t_2 - t_1) \quad (4.62)$$

where (4.61) follows from Lemma 4.8 ($R_{0,\text{in}}^{\text{III}}(t) \sim E_0(t)$) and (4.62) from the fact that $L_2 - \delta_2 + L_1 - \delta_1 = t_2 - t_1$ by construction. \square

We conclude this section with the lemma that shows how changing $R_{0,\text{in}}^{\text{I}}(t)$ into $R_{0,\text{in}}^{\text{III}}(t)$ does, indeed, increase the average delay.

Lemma 4.10. *Given any (fixed) $R_{0,\text{in}}^{\text{I}}(t)$, $R_{1,\text{in}}^{\text{I}}(t)$ and \bar{y} (such that $t_{0,\text{in}}(\bar{y})$ and $t_{0,\text{out}}(\bar{y})$ are finite), let τ be the beginning of the system busy period containing $t_{0,\text{in}}(\bar{y})$ (if $t_{0,\text{in}}(\bar{y})$ is not contained in any busy period let $\tau = t_{0,\text{in}}(\bar{y})$). Let $E_1(t)$ be concave for $t \geq 0$ so that α can be defined as in (4.26). As in Lemma 4.5 let $R_{0,\text{in}}^{\text{II}}(t) = R_{0,\text{in}}^{\text{I}}(t)$ and let $R_{1,\text{in}}^{\text{II}}(t)$ be as in (4.24). Let y_{tot} be the total amount of traffic sent by $R_{0,\text{in}}^{\text{II}}$ in the busy period containing $t_{0,\text{in}}(\bar{y})$ and let t_e be the end of this busy period when the inputs are $R_{0,\text{in}}^{\text{II}}$ and $R_{1,\text{in}}^{\text{II}}$, so that $y_{\text{tot}} = R_{0,\text{in}}^{\text{II}}(t_e) - R_{0,\text{in}}^{\text{II}}(\tau)$. If $R_{0,\text{in}}^{\text{III}}(t)$ is defined as in (4.35), then $\forall y \in [y_0, y_0 + y_{\text{tot}}]$:*

$$d_0^{\text{II}}(y) \leq d_0^{\text{III}}(y)$$

where $y_0 = R_{0,\text{in}}^{\text{I}}(\tau)$.

Proof. For the sake of simplicity we are going to assume that $R_{0,\text{in}}^{\text{II}}(t)$ is a continuous function, if this is not the case the same argument does apply but the notation is somewhat cumbersome.

For any fixed $y \in [y_0, y_0 + y_{\text{tot}}]$ by definition of $b(t)$ we have:

$$b(t_{\text{in}}) = R_{0,\text{in}}(t_{\text{in}}) - R_{0,\text{in}}(\tau) + R_{1,\text{in}}(t_{\text{in}}) - R_{1,\text{in}}(\tau) - C(t_{\text{in}} - \tau).$$

Let $t_{\text{in}}^{\text{II}}(y)$ (resp. $t_{\text{in}}^{\text{III}}(y)$) be the ingress time for the bit arriving after y units of traffic have arrived from flow 0 when the inputs are $R_{0,\text{in}}^{\text{II}}$ and $R_{1,\text{in}}^{\text{II}}$ (resp. $R_{0,\text{in}}^{\text{III}}$ and $R_{1,\text{in}}^{\text{III}}$). We will often use the shorter notation $t_{\text{in}}^{\text{I}} = t_{\text{in}}^{\text{II}}(y)$ and $t_{\text{in}}^{\text{III}} = t_{\text{in}}^{\text{III}}(y)$. Given that $R_{1,\text{in}}(t)$ is the same in both scenarios we have $R_{1,\text{in}}(t_{\text{in}}) - R_{1,\text{in}}(\tau) = E_1(t_{\text{in}} - \tau)$ (from Lemma 4.5) and

$$\begin{aligned} b^{\text{II}}(t_{\text{in}}^{\text{II}}) &= R_{0,\text{in}}^{\text{II}}(t_{\text{in}}^{\text{II}}) - R_{0,\text{in}}^{\text{II}}(\tau) + E_1(t_{\text{in}}^{\text{II}} - \tau) - C(t_{\text{in}}^{\text{II}} - \tau) \\ b^{\text{III}}(t_{\text{in}}^{\text{III}}) &= R_{0,\text{in}}^{\text{III}}(t_{\text{in}}^{\text{III}}) - R_{0,\text{in}}^{\text{III}}(\tau) + E_1(t_{\text{in}}^{\text{III}} - \tau) - C(t_{\text{in}}^{\text{III}} - \tau). \end{aligned}$$

By definition $t_{\text{in}}^{\text{II}}$ and $t_{\text{in}}^{\text{III}}$ are such that $R_{0,\text{in}}^{\text{III}}(t_{\text{in}}^{\text{II}}) = R_{0,\text{in}}^{\text{III}}(t_{\text{in}}^{\text{III}})$, while by definition $R_{0,\text{in}}^{\text{III}}$ is such that $R_{0,\text{in}}^{\text{III}}(\tau) = R_{0,\text{in}}^{\text{II}}(\tau)$, therefore

$$b^{\text{III}}(t_{\text{in}}^{\text{III}}) - b^{\text{II}}(t_{\text{in}}^{\text{II}}) = E_1(t_{\text{in}}^{\text{III}} - \tau) - C(t_{\text{in}}^{\text{III}} - \tau) - [E_1(t_{\text{in}}^{\text{II}} - \tau) - C(t_{\text{in}}^{\text{II}} - \tau)]. \quad (4.63)$$

If we define $y_1 = R_{0,\text{in}}^{\text{II}}(\tau + \alpha) - R_{0,\text{in}}^{\text{II}}(\tau)$ and $y_2 = R_{0,\text{in}}^{\text{II}}(t_e) - R_{0,\text{in}}^{\text{II}}(\tau + \alpha)$ this implies that for any $y \in [y_0, y_0 + y_1]$ $t_{\text{in}}^{\text{III}}(y) \geq t_{\text{in}}^{\text{II}}(y)$, while for any $y \in [y_0 + y_1, y_0 + y_{\text{tot}}]$ $t_{\text{in}}^{\text{III}}(y) \leq t_{\text{in}}^{\text{II}}(y)$, note that all the traffic in $[y_0, y_0 + y_1]$ arrives before time $\tau + \alpha$ both in scenario II as well in scenario III, while the traffic in $[y_0 + y_1, y_0 + y_{\text{tot}}]$ arrives after time $\tau + \alpha$ again in both scenarios. Therefore we can use Lemma 4.6 to conclude that:

$$E_1(t_{\text{in}}^{\text{III}} - \tau) - C(t_{\text{in}}^{\text{III}} - \tau) \geq E_1(t_{\text{in}}^{\text{II}} - \tau) - C(t_{\text{in}}^{\text{II}} - \tau). \quad (4.64)$$

Combining (4.64) with (4.63) we have that $b^{\text{II}}(t_{\text{in}}^{\text{II}}) \leq b^{\text{III}}(t_{\text{in}}^{\text{III}})$. Therefore $d_0^{\text{II}}(y) \leq d_0^{\text{III}}(y)$ for any y such that $y_0 \leq y \leq y_0 + y_{\text{tot}}$, given that under the continuity assumption $d_0(y) = b(t_{\text{in}}(y))/C$. \square

4.4 Main Result

We will now use the previous results to find an upper bound for the worst case average delay. First by using Lemma 4.5 we can fix the cross traffic; then to find the worst possible $R_{0,\text{in}}$ we divide the original arrival pattern in small intervals so that during each interval $R_{0,\text{in}}$ can be well approximated with a linear function. This approximation will allow us to use the optimization problem that we have introduced in section 4.3.1.

As previously mentioned we are assuming the E_0 is piecewise linear. Let r_n and L_n

be the slopes and lengths of its linear segments. That is E_0 can be written as:

$$E_0(t) = \begin{cases} 0 & \text{if } t \leq 0 \\ r_1 t & \text{if } 0 \leq t \leq L_1 \\ r_2 t + r_1 L_1 & \text{if } L_1 \leq t \leq L_2 \\ \dots & \\ r_n t + \sum_{i=1}^{n-1} r_i L_i & \text{if } L_{n-1} \leq t \leq L_n \\ \dots & \\ r_N t + \sum_{i=1}^{N-1} r_i L_i & \text{if } t \geq L_N. \end{cases} \quad (4.65)$$

Theorem 4.11. *Let $R_{0,in}(t) \sim E_0(t)$ and $R_{1,in}(t) \sim E_1(t)$ be the inputs to a FIFO server with capacity C such that the system is stable, if $E_1(t)$ is concave for $t \geq 0$ then, for any $B > 0$:*

$$\frac{1}{B} \int_0^B d_0(y) dy \leq D^*$$

where

$$D^* = \sup_{\beta > 0} \left\{ \frac{1}{C\beta} \left(\sum_n r_n \int_{\bigcup_m T_{nm}^{IV}} [E_1(x) - Cx] dx + \frac{\beta^2}{2} \right) \right\} \quad (4.66)$$

and $\{T_{nm}^{IV}\}$ is such that:

$$f(\tau_1) = f(t_1)$$

$$\tau_1 \leq \alpha \leq t_1$$

$$t_1 - \tau_1 = L_1$$

$$\tau_{i,1} = t_{i+1,1}$$

$$t_{i,2} = \tau_{i+1,2}$$

$$f(\tau_{i,1}) = f(t_{i,2})$$

$$f(t_{i,1}) = f(\tau_{i,2})$$

$$\tau_{i,1} \leq t_{i,1} \leq \alpha$$

$$\alpha \leq \tau_{i,2} \leq t_{i,2}$$

$$t_{\max} - \tau_{\min} = E_1^{-1}(\beta)$$

$$t_{i,1} - \tau_{i,1} + t_{i,2} - \tau_{i,2} = L_i$$

and all the r_n and L_n are those from E_0 , see Figure 4.10.

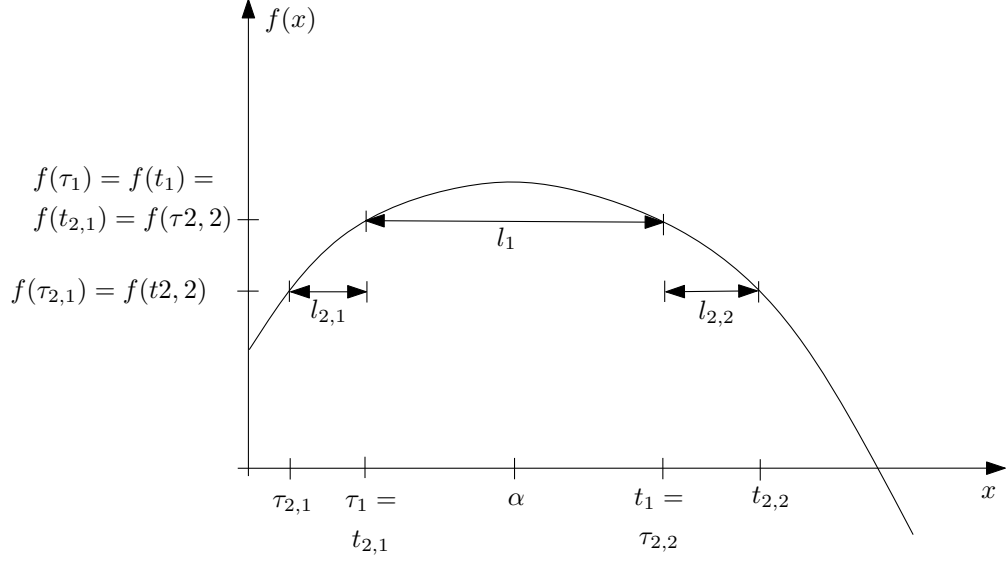


Figure 4.10: The intervals for $\{T_{nm}^{\text{IV}}\}$ with two weights: $r_1 \geq r_2$

The idea behind $\{T_{nm}^{\text{IV}}\}$ is that it satisfies Lemmas 4.1, 4.2 and 4.3. In particular the water filling condition implies that each segment of length L_i is split in two different pieces whose length is dictated by the water filling condition. One piece is placed to the left of α while the other one is placed to the right of α . Furthermore given that E_0 is concave $\rho_{i+1} \leq \rho_i$ so that ρ_1 is the biggest slope and will be centered around α .

Proof. Let a' be the set of all system busy periods and $T > 0$ such that $R_{0,\text{in}}(T) = B$. Define:

$$a = \{A \cap [0, T] : A \in a'\}.$$

Note that the cardinality of the set a' is at most countably infinite. Let J_1, J_2, \dots be the elements of a . Based on this definition J_k is a busy period for all k except for possible one value of k as one of the J_k may consist of only the initial portion of a busy period (the one containing T). Since $R_{i,\text{in}}(t) = 0$ for $t < 0$ it is not possible that one of the J_k consists of only the final part of a busy period. Let $B_k = [y_k^s, y_k^e]^3$ be the amount of traffic sent during the k -th busy period, note that $\sum_k B_k \leq B$.

Based on this decomposition of the interval $[0, B]$ we will first look at a single busy period and find an upper bound for the average delay during this (arbitrary) busy period.

³The superscripts s and e are meant to refer, respectively, the start and the end of the busy period.

For any J_k letting $y_0 = y_k^s$ and $y_{k,\text{tot}} = y_k^e - y_k^s$ we have that:

$$\int_{B_k} d_0(y) dy = \frac{1}{C} \int_{B_k} b(y) dy \quad (4.67)$$

$$= \frac{1}{C} \int_{B_k} [R_1^I(t_{\text{in}}^I(y)) - R_1^I(\tau) + y - R_0^I(\tau) - C(t_{\text{in}}^I(y) - \tau)] dy \quad (4.68)$$

$$= \frac{1}{C} \int_{B'_k} [R_1^I(t_{\text{in}}^I(y')) + y' - C t_{\text{in}}^I(y')] dy' \quad (4.69)$$

$$= \frac{1}{C} \left\{ \int_{B_k} [R_1^I(t_{\text{in}}^I(y)) - C t_{\text{in}}^I(y)] dy + \frac{y_{k,\text{tot}}^2}{2} \right\} \quad (4.70)$$

$$\leq \frac{1}{C} \left\{ \int_{B_k} [E_1(t_{\text{in}}^{\text{II}}(y)) - C t_{\text{in}}^{\text{II}}(y)] dy + \frac{y_{k,\text{tot}}^2}{2} \right\} \quad (4.71)$$

$$\leq \frac{1}{C} \left\{ \int_{B_k} [E_1(t_{\text{in}}^{\text{III}}(y)) - C t_{\text{in}}^{\text{III}}(y)] dy + \frac{y_{k,\text{tot}}^2}{2} \right\} \quad (4.72)$$

$$= \frac{1}{C} \left\{ \int_{B_k} f(t_{\text{in}}^{\text{III}}(y)) dy + \frac{y_{k,\text{tot}}^2}{2} \right\}. \quad (4.73)$$

In (4.67) we have used the fact that $d_0(y) = b(y)/C$, that is the delay for the y -th bit from flow 0 is equal to the total backlog seen by that bit divided by C . In (4.68) we have used the definition of backlog: $b(y) = R_1(t_{\text{in}}(y)) - R_1(\tau) + R_0(t_{\text{in}}(y)) - R_0(\tau) - C(t_{\text{in}}(y) - \tau)$ and the fact that $R_1(t_{\text{in}}(y)) = y$ by definition of $t_{\text{in}}(y)$. Given that we are considering the k -th busy period of a FIFO server what happened before the beginning of this busy period (τ) it is irrelevant, therefore, without loss of generality, we can assume that the k -th busy period started at time 0 and that there was no traffic before that time. To this effect in (4.69) we introduce the new variable $y' = y - R_0(\tau)$, $t'_{\text{in}}(y) = t_{\text{in}}(y) - t_{\text{in}}(y_0)$ and $B'_k = [0, y_{\text{tot}}]$, from (4.70) on we rename y' , $t'_{\text{in}}(y)$ and B'_k as y , $t_{\text{in}}(y)$ B_k in order to simplify the notation. From Lemma (4.5) we know that the delay and backlog are maximized when R_1 is greedy starting at the beginning of the busy period. Hence (4.71) where we have changed t_{in}^I into $t_{\text{in}}^{\text{II}}$ to indicate that we are now considering scenario II, recall that in scenario II R_0 unchanged while R_1 is greedy starting from the beginning of the busy period. Similarly from Lemma 4.10 we know that if we change $R_{0,\text{in}}^{\text{II}}$ into $R_{0,\text{in}}^{\text{III}}$ the average delay is going to increase, hence (4.72). Finally, defining $f(x) = E_1(x) - Cx$, we have (4.73).

At this point we would like to use the optimization problem introduced in section 4.3.1 but before we can do that we need to approximate the arrival process $R_0(t)$ with a piecewise linear function, note that this is always possible and that by increasing the

number of segments it is possible to obtain an arbitrarily good approximation.

As in (4.67)-(4.73) we will focus our attention on the k -th busy period. Let $\varepsilon = y_{\text{tot}}/N$ where N is some fixed number. In other words we are dividing the traffic arriving in the k -th busy period into N intervals each of size ε . Using this subdivision we can write:

$$\int_{B_k} f(t_{\text{in}}(y))dy = \sum_{n=1}^N \int_{(n-1)\varepsilon}^{n\varepsilon} f(t_{\text{in}}^{\text{III}}(y))dy. \quad (4.74)$$

For sufficiently large N , $R_0(t)$ is going to be well approximated by a linear function in each interval $y_n = [(n-1)\varepsilon, n\varepsilon]$. Furthermore from Lemma 4.8 we know that $R_{0,\text{in}}^{\text{III}}(t)$ does satisfy the envelope E_0 therefore we can choose the slope of each segment from one of the slopes of E_0 (recall that E_0 is piecewise linear, see (4.65)). If we let r_n be the slope of this linear approximation in each subinterval then, again from Lemma 4.8, we also know that the total length of all the segments that use a certain slope (say r_i) is at most L_i so that the linear approximation will satisfy the envelope as well.

We can define the endpoints of each interval as follows:

$$\begin{aligned} \tau_n^{\text{III}} &= \lim_{\Delta \rightarrow 0^+} t_{\text{in}}^{\text{III}}((n-1)\varepsilon + \Delta) \\ t_n^{\text{III}} &= t_{\text{in}}^{\text{III}}(n\varepsilon). \end{aligned}$$

We need to use the limit in the definition of τ_n^{III} to deal with the case when the traffic arriving in two consecutive intervals y_{n-1} and y_n is separated by a period of inactivity⁴ so that $t_n^{\text{III}} < \tau_n^{\text{III}}$. Using this linear approximation we have that $\forall y \in [(n-1)\varepsilon, n\varepsilon]$:

$$t_{\text{in}}^{\text{III}}(y) = t_{n-1}^{\text{III}} + \frac{y - (n-1)\varepsilon}{r_n},$$

so that we can define $z = t_{\text{in}}^{\text{III}}(y)$ and operate a change of variable in each element of the sum in (4.74). Given that $dy = r_n dz$ we have

$$\int_{(n-1)\varepsilon}^{n\varepsilon} f(t_{\text{in}}^{\text{III}}(y))dy = r_n \int_{\tau_n^{\text{III}}}^{t_n^{\text{III}}} f(z)dz. \quad (4.75)$$

Combining (4.74) with (4.75) we obtain:

$$\int_{B_k} f(t_{\text{in}}(y))dy = \sum_{n=1}^N r_n \int_{\tau_n^{\text{III}}}^{t_n^{\text{III}}} f(z)dz. \quad (4.76)$$

⁴This is not the case for $R_{0,\text{in}}^{\text{III}}$: given that it is greedy there are no periods of inactivity; in a more general case, though, this could happen.

Given that the right hand side of (4.76) has the same formulation as the optimization problem (4.1), presented in section 4.3.1, and that all the other hypothesis are met, we can upper bound $\int_{B_k} f(t_{\text{in}}(y))dy$ by using the optimal solution of (4.1). Let $T^{\text{IV}} = \{\tau_n^{\text{IV}}, t_n^{\text{IV}}\}$ be the optimal solution of the optimization problem associated with (4.76). This solution corresponds to a unique arrival pattern $R_{0,\text{in}}^{\text{IV}}(t)$ which is going to be piecewise linear. Furthermore T^{IV} and therefore $R_{0,\text{in}}^{\text{IV}}(t)$ satisfy Lemmas 4.1, 4.2, 4.3. From Lemma 4.9 we know that $R_{0,\text{in}}^{\text{IV}}(t)$ does satisfy the envelope $E_0(t)$. Formally we have:

$$\begin{aligned} \int_{B_k} f(t_{\text{in}}^{\text{III}}(y))dy &= \sum_{n=1}^N r_n \int_{\tau_n^{\text{III}}}^{t_n^{\text{III}}} f(z)dz \\ &\leq \sup_{\tau_n, t_n} \left\{ \sum_{n=1}^N r_n \int_{\tau_n}^{t_n} f(z)dz \right\} \\ &= \sum_{n=1}^N r_n \int_{\tau_n^{\text{IV}}}^{t_n^{\text{IV}}} f(z)dz. \end{aligned} \quad (4.77)$$

The only parameter that we need to know in order to explicitly compute $R_{0,\text{in}}^{\text{IV}}(t)$ (other than E_0 and E_1) is the size of the busy period B_k . In order to solve this problem we can take the supremum over all possible busy period sizes:

$$\int_{B_k} d_0(y)dy \leq \frac{1}{C} \left\{ \int_{B_k} f(t_{\text{in}}^{\text{III}}(y))dy + \frac{y_{k,\text{tot}}^2}{2} \right\} \quad (4.78)$$

$$= \left\{ \frac{1}{C y_{k,\text{tot}}} \left(\int_0^{y_{k,\text{tot}}} f(t_{\text{in}}^{\text{III}}(y))dy + \frac{y_{k,\text{tot}}^2}{2} \right) \right\} y_{k,\text{tot}} \quad (4.79)$$

$$\leq \sup_{\beta > 0} \left\{ \frac{1}{C\beta} \left(\int_0^\beta f(t_{\text{in}}^{\text{III}}(y))dy + \frac{\beta^2}{2} \right) \right\} y_{k,\text{tot}} \quad (4.80)$$

$$\leq \sup_{\beta > 0} \left\{ \frac{1}{C\beta} \left(\sum_{n=1}^N r_n \int_{\tau_n^{\text{IV}}}^{t_n^{\text{IV}}} f(z)dz + \frac{\beta^2}{2} \right) \right\} y_{k,\text{tot}} \quad (4.81)$$

$$= \sup_{\beta > 0} \left\{ \frac{1}{C\beta} \left(\sum_n r_n \int_{\bigcup_m T_{nm}^{\text{IV}}} [E_1(x) - Cx]dx + \frac{\beta^2}{2} \right) \right\} y_{k,\text{tot}} \quad (4.82)$$

$$= D^* y_{k,\text{tot}} \quad (4.83)$$

where the first inequality (4.78) is the same as (4.73). In (4.79) we have simply multiplied and divided by $y_{k,\text{tot}}$ (recall that $y_{k,\text{tot}}$ is the size of the k -th busy period $y_{k,\text{tot}} = R_{0,\text{in}}(t_e) - R_{0,\text{in}}(\tau)$). In (4.80) we take the supremum over all positive β to upper bound (4.79). (4.81) follows immediately from (4.77); (4.82) and (4.83) follow immediately from the definition of $t_{\text{in}}^{\text{IV}}$ and $D^*(4.66)$.

Using the bounds that we have derived so far we can write:

$$\int_0^B d_0(y)dy \leq \int_{\bigcup B_k} d_0(y)dy \quad (4.84)$$

$$= \sum_k \int_{B_k} d_0(y)dy \quad (4.85)$$

$$\leq \sum_k D^* y_{k,\text{tot}} \quad (4.86)$$

$$= D^* \sum_k y_{k,\text{tot}} \quad (4.87)$$

$$= D^* B \quad (4.88)$$

the first inequality (4.84) follows from the definition of B_k . Equation (4.85) follows from the fact that the cardinality of a is at most countably infinite. (4.86) follows from (4.83) and (4.81) follows from the fact that D^* does not depend on k . Finally using the definition of B we have (4.88). \square

Corollary 4.12. *There exist arrival patterns $R_{0,\text{in}}^*$ and $R_{1,\text{in}}^*$ that achieve the average delay D^* defined in (4.66).*

Proof. For the sake of simplicity assume that both envelopes are such that the quantity

$$\lambda_i = \lim_{t \rightarrow \infty} \frac{E_i(t)}{t}$$

is well defined for both flows ($i = 0, 1$). Note that if E_0 is piecewise linear λ_0 is simply r_N . If this is not the case it is still possible to construct $R_{0,\text{in}}^*$ and $R_{1,\text{in}}^*$ but their analytical representation it is going to be more complicated. Let β^* be the optimal busy period size, that is the β that achieves the supremum in (4.66). If the supremum is not achieved the same reasoning applies but again the notation is more complicated so we are going to assume that this is not the case. Clearly every β^*/λ_0 units of time flow 0 can send a burst of size β^* without violating the envelope. At the same time, in order to achieve the worst case, we also need the cross traffic ($R_{1,\text{in}}$) to be able to be greedy for the duration of the busy period. It is easy to obtain an upper bound for the amount of time needed by $R_{1,\text{in}}$ to be able to send a burst of the appropriate size. From the definition of D^* in (4.66) we know that the flow 0 will send its traffic around α . We also know that the burst from flow 0 will last exactly $E_0^{-1}(\beta^*)$ units of time. The worst it can happen is for the optimal solution to be such that flow 0 will start sending traffic at α (we know it cannot

start any later) and stop at time $\alpha + E_0^{-1}(\beta^*)$ in this case flow 1 will have to burst for the same amount of time, sending a total of $E_1^{-1}(\alpha + E_0^{-1}(\beta^*))$ units of traffic. Therefore we can define:

$$\varphi = \max \left[\frac{\beta^*}{\lambda_0}, \frac{E_1^{-1}(\alpha + E_0^{-1}(\beta^*))}{\lambda_1} \right]$$

so that if both sources do not send any traffic for φ units of time both of them will be “fresh.” That is they will be able to burst as much as needed in order to achieve D^* no matter what they did in the past (before the resting period).

Based on this observation we can construct a periodic version of $R_{i,\text{in}}(t)$, where φ is the period. Starting at time 0 (assuming there was no traffic before that time) both streams are “fresh” and can follow the envelope until the times dictated by the definition of D^* . By solving the optimization problem (4.1), when the size of the busy period is β^* , we can find τ_{\min} and t_{\max} . From the proof of Theorem 4.11 we know that flow 0 should start sending traffic according to $R_{0,\text{in}}^{\text{IV}}(t)$ at time τ_{\min} until time t_{\max} . Then, if it does not send any traffic until $\alpha + E_0^{-1}(\beta^*) + \varphi$, we know that it can send another burst of size β^* without violating the envelope. Flow 1 should start sending traffic at time 0 until time t_{\max} following the envelope E_1 then, just like flow 0, it will be ready to start a new period at time $\alpha + E_0^{-1}(\beta^*) + \varphi$. Given that both flow are periodic with the same period φ we have $R_{i,\text{in}}(t) = R_{i,\text{in}}(t + \varphi)$ and over the first period:

$$R_{0,\text{in}}(t) = \begin{cases} 0, & \text{if } 0 \leq t < \tau_{\min} \\ R_{0,\text{in}}^{\text{IV}}(t), & \text{if } \tau_{\min} \leq t \leq t_{\max} \\ R_{0,\text{in}}^{\text{IV}}(t_{\max}), & \text{if } t_{\max} < t \leq \alpha + E_0^{-1}(\beta^*) + \varphi \end{cases}$$

$$R_{1,\text{in}}(t) = \begin{cases} E_1(t), & \text{if } 0 \leq t \leq t_{\max} \\ E_1(t_{\max}), & \text{if } t_{\max} < t \leq \alpha + E_0^{-1}(\beta^*) + \varphi. \end{cases}$$

If both flows keep following this pattern every φ units of time the resulting busy periods will all have average delay D^* and hence the average delay for the whole process will be D^* . \square

Example 4.13. When both E_0 and E_1 are “sigma-rho” envelope (that is $E_i(t) = \sigma_i + \rho_i t$)

it can be shown that the upper bound on the average delay given by Theorem 4.11 is:

$$D_{avg} = \frac{\rho_1 \sigma_0}{C \rho_0} \left(\sqrt{\frac{C - (\rho_0 + \rho_1)}{C - \rho_1}} - 1 \right) + \frac{\sigma_0}{\rho_0} \left(1 - \sqrt{\frac{C - (\rho_0 + \rho_1)}{C - \rho_1}} \right) + \frac{\sigma_1}{C}. \quad (4.89)$$

In this case the notation used in Theorem 4.11 does not apply because E_0 is not continuous at time 0 and does not fit the model in (4.65) but the same argument does hold. In this case $\alpha = 0$ so that the function G is not used and $H(t) = \sigma_0 + \rho_0 t$. The water filling condition is trivial because E_1 has only one slope and $\alpha = 0$.

More precisely the arrival pattern is as follows: both flows are greedy starting at time 0 but flow 1 sends its burst of size σ_0 right after the cross traffic does so that the cross traffic will be served first. Then both flows send traffic at rate ρ_i until the end of the busy period. Just as in (4.66) we have to find the optimal busy period size β^* . In this specific case it is easy to see that:

$$d_0(y) = \begin{cases} \frac{y}{C} + \frac{\sigma_1}{C}, & \text{if } 0 \leq y \leq \sigma_0 \\ \frac{1}{C \rho_0} [y(\rho_0 + \rho_1) - \rho_1 \sigma_0 + C(\sigma_0 - y) + \rho_0 \sigma_1], & \text{if } \sigma_0 \leq y \leq \sigma_0 + \rho_0 \frac{\sigma_0 + \sigma_1}{C - \rho_0 - \rho_1} \end{cases} \quad (4.90)$$

where $\frac{\sigma_0 + \sigma_1}{C - \rho_0 - \rho_1}$ is the maximum length of a system busy period so that $\sigma_0 + \rho_0 \frac{\sigma_0 + \sigma_1}{C - \rho_0 - \rho_1}$ is the maximum amount of traffic that $R_{0,in}(t)$ can send in a single busy period. Using (4.90) we can find the optimal size of the busy period by solving the following optimization problem:

$$\sup_{\beta > 0} \frac{1}{\beta} \int_0^\beta d_0(y) dy. \quad (4.91)$$

Given that $d_0(y)$ is a linear function of y we can calculate the value of the integral in (4.91), which will be a quadratic function of β . Then we can find the optimal value of β by taking the first derivative and setting it equal to zero. Using this value for β we can compute $D_{avg} = \frac{1}{\beta} \int_0^\beta d_0(y) dy$ and the corresponding value is (4.89).

We conclude this chapter with another corollary to Theorem 4.11. It is easy to see that all the proofs that we have presented continue to hold if we are interested in finding an upper bound for $g(d_0(y))$ where g is a nondecreasing function.

Corollary 4.14. *Let $R_{0,in} \sim E_0$, $R_{1,in} \sim E_1$, as in Theorem 4.11. Let g be a nondecreasing function then for any $B > 0$:*

$$\frac{1}{B} \int_0^B g(d_0(y)) dy \leq D^*$$

where

$$D^* = \sup_{\beta > 0} \left\{ \frac{1}{\beta} g \left(\frac{1}{C} \sum_n r_n \int_{\bigcup_m T_{nm}^{IV}} [E_1(x) - Cx] dx + \frac{\beta^2}{2C} \right) \right\}.$$

If we pick g as:

$$g_\delta(x) = \begin{cases} 0, & \text{if } x \leq \delta \\ 1, & \text{if } x > \delta, \end{cases}$$

we can use this corollary to obtain a lower bound on the delay distribution for flow 0. Similarly to what we did for the delay we are going to consider the specific case where both envelopes are sigma-rho envelopes. Again the notation used in Theorem 4.11 and corollary 4.14 does not apply because E_0 does not fit the model in 4.65. But we can use the same argument to show that in this case the worst case arrival pattern is the same and the expression for $d_0(y)$ in the worst case and it is the same as in (4.90).

In this case we are interested in finding the values of y such that $d_0(y) > \delta$ for any $\delta \geq 0$. It is easy to see that if $\delta \leq \frac{\sigma_1}{C}$ then $d_0(y) > \delta$ if $0 \leq y < \min\{\beta, \xi\}$ where

$$\xi = \frac{C\sigma_0 + \rho_0\sigma_1 - C\rho_0\delta - \rho_1\sigma_0}{C - \rho_0 - \rho_1},$$

and β is the size of the busy period. While, if $\frac{\sigma_1}{C} \leq \delta \leq \frac{\sigma_0 + \sigma_1}{C}$ then $d_0(y) > \delta$ if $\delta C - \sigma_1 < y < \min\{\beta, \xi\}$. We still have to find the optimal value for β :

$$\sup_{\beta > 0} \int g_\delta(d_0(y)) dy = \begin{cases} \sup_{\beta} \int_0^{\min\{\beta, \xi\}} dy, & \text{if } 0 \leq \delta \leq \frac{\sigma_1}{C} \\ \sup_{\beta} \int_{\delta C - \sigma_1}^{\min\{\beta, \xi\}} dy, & \text{if } \frac{\sigma_1}{C} \leq \delta \leq \frac{\sigma_0 + \sigma_1}{C}. \end{cases}$$

Given that $\int_0^{\min\{\beta, \xi\}} dy = \min\{\beta, \xi\}$ and that $\sup_{\beta} [\min\{\beta, \xi\}] = \xi$ we have:

$$\frac{1}{B} \int_0^B g_\delta(d_0(y)) dy \leq \begin{cases} 1, & \text{if } 0 \leq \delta \leq \frac{\sigma_1}{C} \\ \frac{C^2\delta - C\delta\rho_1 - C\sigma_0 + \rho_1\sigma_0 - C\sigma_1 + \rho_1\sigma_1}{C\delta\rho_0 - C\sigma_0 + \rho_1\sigma_0 - \rho_0\sigma_1}, & \text{if } \frac{\sigma_1}{C} \leq \delta \leq \frac{\sigma_0 + \sigma_1}{C}. \end{cases} \quad (4.92)$$

Given that $\frac{1}{B} \int_0^B g_\delta(d_0(y)) dy$ represents the fraction of bits that have a delay of at least δ , $1 - \frac{1}{B} \int_0^B g_\delta(d_0(y)) dy$ is the fraction of bits that have a delay of at most δ . Therefore $1 - \frac{1}{B} \int_0^B g_\delta(d_0(y)) dy$ is a lower bound for the delay distribution for an arbitrary sample path. Combining this fact with (4.92) we have a lower bound for the delay distribution:

$$F(\delta) = \Pr[\text{delay} \leq \delta] \geq \frac{\delta C(\rho_0 - C + \delta\rho_1) + C\sigma_1 - \rho_0\sigma_1 - \rho_1\sigma_1}{C\delta\rho_0 - C\sigma_0 + \rho_1\sigma_0 - \rho_0\sigma_1}.$$

5

Conclusions

Using the previously known fact that a single FIFO queue is characterized by an infinite family of service curves we have shown that in the case of a single FIFO queue with inputs that have sigma-rho envelopes it is possible to recover previously known and tight QoS bounds. Furthermore this approach allowed us to derive a previously unavailable upper bound on the backlog of a single flow. Using the same approach for two FIFO queues in tandem (with all the inputs having sigma-rho envelopes) we were able to obtain an end-to-end bound that, at least in some cases, it is better than the one obtained by considering each node in isolation. At the same time we were not able to find an arrival pattern that does achieve this bound. Subsequently we derived a tighter bound, which implies that the first bound is not achievable.

In order to find an achievable end-to-end delay bound we have introduced a new service abstraction that generalizes the widely used service curve framework. This new model is defined in terms of a “service mapping” which is a monotone operator that maps an arrival process to a lower bound on the corresponding output process. We considered service mappings that are shift invariant and we have shown how to obtain QoS bounds (worst case delay, maximum backlog and output envelope) for any network element offering a shift invariant service mapping.

Using this new service model we have been able to obtain a new end-to-end delay bound for two FIFO queues in tandem. In this case we were also able to find a set of arrival processes (one for each input) such that this worst case bound is indeed achieved, proving that it is a tight bound.

In both cases (single and multiple queues) we now have an achievable delay bound, at the same time only a fraction of the traffic will experience such a delay (unless we are dealing with some extreme cases on the verge of instability). This led us to consider the worst case average delay, where the average is taken over time.

We considered a single FIFO queues with two input flows, one with a piecewise linear and concave envelope, the other with a concave envelope. For this setting we derived an achievable bound for the worst case average delay for the flow with the piecewise linear envelope. Exploiting the concavity of the cross traffic envelope we showed that the through traffic should concentrate its traffic around specific points in time. Furthermore how the traffic should be distributed around these points in time is dictated by a condition that we called “water filling,” because it is somewhat similar to the way water would distribute itself in a convex bowl. We were also able to show that it is always possible to construct an arrival pattern that does achieve this worst case average delay, proving that this bound is tight.

These new results have also brought our attention to some new and interesting problems and we conclude this dissertation with a brief discussion about some of them.

5.1 Open Problems

As we have mentioned in the introduction one of the motivations for this work was the work of the DiffServ group of the IETF. Yet, as a first step, we only considered FIFO queues. As such, even the two FIFO queues in tandem cannot be considered a non-trivial DiffServ network. At the same time this problem might help us shed some light on a related problem where we have multiple aggregates being served according to some other scheduling algorithm and where each packets belonging to the same aggregate are served in a FIFO manner. For example if the algorithm used to serve the aggregates offers a service curve to the aggregate the service mapping approach should apply as well. Recall that the theorem in [15] holds whenever the aggregate of the two (or more) flows is offered an arbitrary service curve. In our case we used Ct as the service curve but it should be easy to extend these results to other service curves. In particular if PGPS is used the modifications should be minimal: this algorithm has a so called “rate latency” service curve which is basically the same as Ct shifted to the right by the latency term [23].

Another possible extension deals with the network topology: the example with two FIFO queues can be trivially extended to the case where there are more queues with one flow going through all of them (the through traffic) and a series of cross traffic flows, one per queue, each sharing only one queue with the through traffic. But there are other similar scenarios that present more challenges. For example, if some of the cross traffic flows share multiple queues with the through traffic we could partition the nodes in multiple subsets so that each subset can be analyzed using one of the known cases. Most likely, though, we would not be able to obtain tight delay bounds this way. This is similar to what happened when we tried analyzing the two node case as two nodes in isolation. Hopefully the results we have derived can be used as a starting point and should give some insight on similar, but more complicated, scenarios.

In the case of the worst case average delay there are many related problems as well. Probably the first one that comes to mind, similar to what we did in the first part, is to consider the multiple node case, again each node can be considered in isolation but most likely this approach will, once more, fail to give an achievable bound.

Another variation of this problem is when the through traffic is given (i.e. known a priori) but the cross traffic is arbitrary. In this case it would still be interesting to obtain a bound on the worst case average delay for the specific realization of the through traffic. As an analogy consider the service mapping (and service curves) model. In that case if $R_{\text{in}}(t)$ is known it is possible to obtain a lower bound for the output, and different realizations of $R_{\text{in}}(t)$ will give different bounds. The idea is that $R_{\text{in}}(t)$ might be such that it is not the worst case, therefore it is not possible to achieve the worst case with this specific realization so it would be useful to know what the worst case average delay would be in this case. This is similar to the “adversarial queuing theory” [4], where it is assumed that the cross traffic is under the control of an adversary trying to delay the through traffic as much as possible. The best case for the adversary is when he knows what the through traffic will do (even in the future). Of course this is not always possible but, being the best the adversary can hope for, it is still an interesting question.

Finally, as we mentioned in the introduction, the worst case average delay bound can be used to better characterize the set of all possible output processes: this value can be used to bound the area between the arrival and departure processes. Therefore any feasible output process is such that the area between the input and output is less than or equal

to this bound. Unfortunately it can still be the case that a certain output process does satisfy all the known constraints (output lower bound, output envelope and area between input and output) and still not be feasible, in the sense that no arrival pattern (for the through and cross traffic) would ever produce this specific output realization.

It might be possible to extend this idea to construct a new service and traffic characterization that would give a “better” description of the set of all possible outputs. Given that the average delay can be achieved over an arbitrary time interval, and not only point-wise as the service mappings (and service curves) output bounds, the hope is that this approach could give bounds that are representative of the output process over longer time periods.

These are just some of the possible extensions of the work presented in this dissertation and by no mean an exhaustive list. Even though FIFO queues are so widespread and the algorithm itself is simple their analysis of their behavior and performance is far from trivial. Even though several problems have been solved there are many interesting and relevant open problems.

Bibliography

- [1] R. Agrawal, R. L. Cruz, C. M. Okino, and R. Rajan. A framework for adaptive service guarantees. In *Proceedings Allerton Conf. on Comm, Control, and Comp.*, Monticello, IL, Sept. 1998.
- [2] R. Agrawal, R.L. Cruz, C. Okino, and R. Rajan. Performance bounds for flow control protocols. *IEEE/ACM Transactions on Networking*, 7(3):310–323, 1999.
- [3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Service. RFC 2475 (Informational), December 1998. Updated by RFC 3260.
- [4] Allan Borodin, Jon Kleinberg, Prabhakar Raghavan, Madhu Sudan, and David P. Williamson. Adversarial queuing theory. *J. ACM*, 48(1):13–38, 2001.
- [5] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: an Overview. RFC 1633 (Informational), June 1994.
- [6] R. Braden and L. Zhang. Resource ReSerVation Protocol (RSVP) – Version 1 Message Processing Rules. RFC 2209 (Informational), September 1997.
- [7] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification. RFC 2205 (Proposed Standard), September 1997. Updated by RFCs 2750, 3936, 4495.
- [8] C.S. Chang. On deterministic traffic regulation and service guarantee: A systematic approach by filtering. *IEEE Transactions on Information Theory*, 44:1096–1107, 1998.
- [9] C.S. Chang. *Performance Guarantees in Communication Networks*. Springer Verlag, New York, 2000.
- [10] C.S. Chang, Y.T. Chen, and D. S. Lee. Constructions of optical fifo queues. *IEEE/ACM Transactions on Networking*, 14(SI):2838–2843, 2006.
- [11] C.S. Chang and R.L. Cruz. A time varying filtering theory for constrained traffic regulation and dynamic service guarantees. In *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 63–70, New York, NY, 1999.

- [12] R. L. Cruz. A calculus for network delay, part II: Network analysis. *IEEE Transactions on Information Theory*, 37, 1:132–141, 1991.
- [13] R. L. Cruz. Quality of service guarantees in virtual circuit switched networks. *IEEE Journal on Selected Areas in Communications*, 13(6):1048–1056, 1995.
- [14] R.L. Cruz. A calculus for network delay, part I. network elements in isolation. *IEEE Transactions on Information Theory*, 37(1):114–131, 1991.
- [15] R.L. Cruz. SCED+: efficient management of quality of service guarantees. In *INFOCOM '98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 625–634, San Francisco, CA, 1998.
- [16] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. In *SIGCOMM '89: Symposium proceedings on Communications architectures & protocols*, pages 1–12, New York, NY, USA, 1989. ACM Press.
- [17] D. Grossman. New Terminology and Clarifications for Diffserv. RFC 3260 (Informational), April 2002.
- [18] F.M. Guillemin, N. Likhanov, R.R. Mazumdar, and C. Rosenberg. Extremal traffic and bounds for the mean delay of multiplexed regulated traffic streams. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 985–993, June 2002.
- [19] H. Sariowan. *A Service-Curve Approach to Performance Guarantees in Integrated Service Networks*. PhD thesis, University of California, San Diego, 1996.
- [20] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. Assured Forwarding PHB Group. RFC 2597 (Proposed Standard), June 1999. Updated by RFC 3260.
- [21] V. Jacobson, K. Nichols, and K. Poduri. An Expedited Forwarding PHB. RFC 2598 (Proposed Standard), June 1999. Obsoleted by RFC 3246.
- [22] J. Y. Le Boudec. Application of network calculus to guaranteed service networks. *IEEE Transactions on Information Theory*, 44(3), May 1998.
- [23] J.Y. Le Boudec and P. Thiran. *Network Calculus*. Springer Verlag LNCS 2050, 2001.
- [24] K. Nichols, S. Blake, F. Baker, and D. Black. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. RFC 2474 (Proposed Standard), December 1998. Updated by RFCs 3168, 3260.
- [25] 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, 1993.
- [26] 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 (TON)*, 2(2):137–150, 1994.

- [27] H. Sariowan, R. L. Cruz, and G. C. Polyzos. Sced: a generalized scheduling policy for guaranteeing quality-of-service. *IEEE/ACM Trans. Netw.*, 7(5):669–684, 1999.
- [28] S. Shenker, C. Partridge, and R. Guerin. Specification of Guaranteed Quality of Service. RFC 2212 (Proposed Standard), September 1997.
- [29] S. Shenker and J. Wroclawski. General Characterization Parameters for Integrated Service Network Elements. RFC 2215 (Proposed Standard), September 1997.
- [30] J. Wroclawski. Specification of the Controlled-Load Network Element Service. RFC 2211 (Proposed Standard), September 1997.
- [31] H. Zhang. Service disciplines for guaranteed performance service in packet-switching networks. In *Proceedings of the IEEE*, 83(10), October 1995.
- [32] L. Zhang. Virtual clock: a new traffic control algorithm for packet switching networks. In *SIGCOMM '90: Proceedings of the ACM symposium on Communications architectures & protocols*, pages 19–29, New York, NY, USA, 1990. ACM Press.