Large Deviations and Internet Congestion

Damon Wischik

Large Deviations and Internet Congestion

Damon Wischik

Dissertation submitted for the Ph.D. degree
at the University of Cambridge

Contents

Preface

How to read this thesis

The progression of chapters in this thesis mirrors the progression in the title, from abstract probability to applied modelling of the Internet. Accordingly, each chapter concludes with a summary of what is used in those that follow. The queueing model in Chapter 3 should be read before the remaining chapters, but otherwise they can be read in any order, and indeed it might be more interesting to start with the applications and read backwards.

Acknowledgements

I am grateful above all to Professor Frank Kelly, my supervisor, whose enthusiasm and insight have made this work enjoyable. I have also benefited greatly from my time with Dr Neil O'Connell at Hewlett-Packard/BRIMS and with Dr Nick Duffield at AT&T Labs.

Damon Wischik Trinity College, September 1999

Chapter 1

Introduction

If written five years ago, the title of this thesis would probably have been *Large Deviations and Queueing Theory*. However, the Internet is one of the most important queueing networks there is today. Last year it was directly involved in hundreds of billions of dollars of economic activity, and each year its size more than doubles. It should be of interest to mathematicians because it raises interesting mathematical questions, and because good and timely answers to those questions can feed back into better design.

Congestion is currently a major problem in the Internet. It leads to unreliable performance, and it is holding back the deployment of new services. If the Internet is to evolve into a high-performance network, suitable for forms of communication that are richer than simple file-transfers, we must understand how congestion arises and find ways to keep the network operating within its capacity. These are our topics in this thesis, and our main tool is the mathematical theory of large deviations.

1.1 Internet Congestion

The Internet is bewilderingly vast, and draws on the expertise of designers at all levels from physicists studying fibre-optics to legislators regulating access. We will be concerned with the level of traffic generation, transmission, and control.

Here are some figures, obtained from transmitting this thesis across the Atlantic to `www.wischik.com` in July 1999. There are roughly 248000 characters to transmit, grouped into *packets* of 1500 characters. Each packet takes 50-60 milliseconds to reach its destination, travelling through 20 different way-stations or *routers*. In the evening, when the Internet is lightly loaded, the entire operation takes just under 2 seconds. In the afternoon, when there is congestion, it can take over 40 seconds. The reason it takes so long is that the source computer waits between sending packets, so as not to overload the network.

The router, connected by cables to other routers and to users, is the basic building block of the Internet. Each packet of data from a user is labelled with its destination and sent out to the first router on its path; at a router, each incoming packet is examined and sent out on the appropriate cable, either to the next router in its path or to its final destination.

When too many packets arrive at a router they are *queued* until they can

be processed. But a router only has enough buffer space for a limited number of packets to queue, and when the buffer is full further incoming packets are *dropped*, that is, discarded. An end-system will eventually detect the drop, and would typically respond by reducing its transmission rate (and by resending the dropped packet). The frequency of packet drops is thus the primary measure of congestion.

For the past decade Internet congestion has been controlled in this way, relying on users' computers to detect congestion and to back off. As the Internet becomes more commercially important, this consensus arrangement is likely to fail. Engineers have recently proposed new mechanisms for signalling congestion, and economists have begun to look at usage-sensitive pricing schemes. But without a clear *mathematical* understanding of the phenomenon of congestion, it is hard to see how these approaches can be understood and integrated. This is where large deviations theory can help.

1.2 Large deviations

Since the Internet operates as a network of queues, the tools of queueing theory can be used to study it. This is not a simple question of applying well-understood mathematical results, getting an answer, and rewording it to refer to the Internet. Rather, there is an ongoing development of the mathematics, driven by the particular needs of this application.

In this thesis we develop the *large deviations* theory of queueing networks. Large deviations theory is a modern branch of probability, concerned with estimating the probabilities of rare events. This makes it well-suited to studying high-performance communications networks, in which dropping a packet should be a rare event. The Internet is not always a high-performance network, as anyone who has tried to use it in the early afternoon will know, but we believe that techniques like those described here will help improve things!

More precisely, large deviations theory is concerned with *limiting regimes*. In queueing problems, while precise equations can be written down, it is very rare that they can be solved exactly: so instead one seeks limiting results. A typical result would be that for a router used by L traffic flows of a specified type, with capacity to serve CL packets a millisecond, the probability of dropping a packet is roughly $e^{-\kappa L}$, where κ can be calculated, and where the approximation is accurate in the limit as L tends to infinity. (In other words, its accuracy improves as the number of traffic sources increases.) This estimate is called a *large deviations principle* for the probability of dropping a packet, and κ is called its *rate*.

Large deviations estimates are governed by the *principle of the largest term*, which means that if a rare event occurs, it is overwhelmingly likely that it occurs in just one way. If we can calculate which is the most likely way, we know the typical behaviour of the system. This means that many of the details which make it hard to obtain exact answers to queueing problems disappear. With this generality comes, of course, some loss of accuracy; we will not investigate that here.

Large deviations theory has been widely studied, and much work has been done on large deviations in queueing theory. What makes this work different is the limiting regime we look at. We consider the *many sources* limiting regime,

exemplified above, in which the number of traffic flows increases. This limiting regime is well-suited to the Internet, which has many thousands of simultaneous traffic flows in its core.

1.3 Large deviations and Internet congestion

We will use large deviations queueing theory to study congestion in networks. Our study has four parts. In the first, we use large deviations to model traffic coming into a router. In the second, we find the way in which a router's buffer overflows. In the third, we model traffic travelling through the network. In the fourth, we analyse algorithms for signalling and controlling congestion.

Large deviations and traffic modelling

Before one can begin to analyse congestion, one must be able to model traffic; and we do this using large deviations theory and the many sources limiting regime.

The rate at which data is sent by a computer program typically varies with time. For example, in sending a video clip, action sequences take more data than static shots. The natural way to model this variability is to take the traffic flow to be a random process. Data is sent in myriad different ways, from many different types of computer application, so we make only very weak assumptions about the characteristics of the process.

Others have already developed a comprehensive theory to describe large deviations for random processes, under the *large buffer* limiting regime, in which buffer size of a router increases but the number of flows stays fixed. In Chapter 2 we develop a full theory for the many sources regime. We also illustrate how this theory is more applicable than the large buffer theory to flows which exhibit the long-range dependence characteristics seen in real Internet traffic.

Formally speaking, we establish a Large Deviations Principle for random processes under the many sources limiting regime. This Principle gives estimates for the probability of any event associated with the aggregate of many traffic flows.

This chapter is mathematically involved. But the work is largely technical, and it is summarized by a single theorem.

How queues fill up

Congestion happens when buffers overflow, so in Chapter 3 we use large deviations to study overflow. We can estimate the probability of overflow using the Contraction Principle, as follows: we rewrite 'the queue overflows' as 'the incoming traffic is such as to make the queue overflow'. This is an event associated with the aggregate input traffic, the probability of which we can estimate using the Large Deviations Principle from the previous chapter. We will give estimates for several other events associated with overflow, some of which have been found before and others of which are new.

We can do more than estimate the probability that a queue overflows: we can also calculate *how* overflows occur. The Principle of the Largest Term says that all that matters is the most likely path to lead to overflow, and that when

overflow occurs it is overwhelmingly likely that it occurs in this way. The idea of the most likely path will play a vital part in our analysis of congestion-signalling mechanisms in Chapter 5.

Networks of queues

Chapter 2 explains how to model traffic as it enters a network—but what really matters is how traffic behaves as it travels through the network. In Chapter 4 we prove the new and surprising result that *the statistical characteristics of a flow of traffic are essentially unchanged as it passes through a router.* This makes it meaningful to talk about the intrinsic characteristics of, say, video traffic as opposed to audio traffic: it is not necessary to consider either the routers that a flow passes through, or the other flows that it interacts with.

Earlier work on networks has reached different conclusions. The reason for the difference is that these are all limiting results, and earlier work has considered different limiting regimes. For example, under the large buffer regime, the characteristics of a flow of traffic change along its route in ways which are complicated and do not lend themselves to general principles. Our choice of limiting regime allows a much cleaner conclusion. We consider the many sources regime, in which the number of flows of traffic increases, and make the additional assumption that different flows follow diverse routes through the network.

Our result has a straightforward mathematical formulation, but its implications are significant and merit a good deal of interpretation. It dramatically simplifies the analysis of congestion in networks.

Signalling congestion

While the results of the previous chapters are framed with the Internet in mind, they apply in principle to any queueing network. In the last chapter however we address a question which has arisen specifically from the needs of the Internet engineering community: How should routers signal congestion to users? There are actually two parts to this: What should be the goals of a congestion-signalling algorithm? and What sort of algorithm can achieve these goals?

Both of these questions have been looked at, though mainly in isolation: economists have considered the first, and engineers the second. But without a mathematical model of the phenomenon of congestion, economists cannot devise pricing structures to prevent it; and without a theory which explains how congestion occurs, engineers cannot analyse their algorithms. Only recently have mathematicians begun to study these issues. In Chapter 5 we address both questions, using the large deviations tools developed in the previous chapters.

First, we define what it means for a router to signal congestion *fairly* and *efficiently*. This involves a large deviations analysis of the impact of a user on the network. It also involves economic modelling of how users behave—a congestion signalling mechanism has the same purposes as a pricing scheme in a market economy: to convey information and to direct consumption.

We then study algorithms for signalling congestion. Recently it has been proposed that Internet routers should be able to mark certain packets with a *congestion experienced* tag, and that users should respond to marked packets as they would to drops. The proposal leaves open the question of what marking algorithm a router should use. We analyse several different algorithms, including

one which has been implemented in commercial routers, using the idea of the most likely path. This is, as far as we are aware, the first theoretical analysis of these algorithms. It turns out that they are all unfair and economically inefficient. We go on to suggest improvements based on principles from large deviations theory.

This chapter is more discursive, as it takes some effort to frame an appropriate mathematical question. Once that is done, the tools of queueing theory can be powerfully applied.

1.4 Summary

The Internet raises interesting mathematical issues. Using a limiting regime suggested by the structure of the Internet, we have been able to prove a result which significantly simplifies the analysis of networks of queues.

Mathematical study is also of benefit to the Internet. If the Internet is to fulfil its promise of revolutionising the way we communicate, it needs to evolve new ways of coping with congestion. The first step must be to understand the nature of congestion—how it occurs and how it affects traffic—and the tools of queueing theory can help with this. Then there must be some way to signal congestion. Signalling mechanisms are just beginning to be developed and built into routers, and insights from economics and large deviations can help in their design. A good signalling mechanism will be fundamental to the future of congestion control.

Chapter 2

Traffic

Consider a queue fed by several different input processes. Many quantities of interest in queueing theory, such as the amount of work in the queue, can be expressed as functions of the sequence of variables $(x_t)_{t\in\mathbb{N}}$, where x_t is the total amount of work received t timesteps ago.

The sequence (x_t) will typically live in a space on which the quantity of interest is a continuous function. For example, let $\mathcal{X}_\mu$ be the space of real-valued sequences $\mathbf{x} = (x_t)$ for which $t^{-1}\sum_{i=1}^{t} x_i < \mu$ eventually. Then the amount of work Q in a queue with an infinite buffer and fixed service rate $C > \mu$ is given by

$$Q(\mathbf{x}) = \Big[\sup_{t>0}\Big(\sum_{i=1}^{t} x_i - Ct\Big)\Big]^+$$

The principal result of this chapter is a large deviations principle (LDP) for a sequence of processes $\mathbf{X}^L$, in $\mathcal{X}_\mu$ equipped with a topology which makes Q continuous.

This can be used to understand the large deviations behaviour of a wide range of queueing systems. Consider a sequence of queueing systems, in which the Lth system has input $\mathbf{X}^L$. In Chapter 3 we will use the Contraction Principle to deduce, from the LDP for $\mathbf{X}^L$, LDPs for various quantities of interest such as $Q(\mathbf{X}^L)$.

We will be motivated by one particular limiting regime, in which $\mathbf{X}^L$ is the average of L processes. This is known in queueing theory as the *many sources asymptotic*. It is well-suited to modern telecommunications networks, in which a router may have thousands of different input flows. However, in this chapter, $\mathbf{X}^L$ can be any sequence of processes.

Before proving the result, we introduce our notation, explain what a large deviations principle is, and review related work. After, we give some examples.

2.1 Large deviations for averages of processes

We will be concerned with the set $\mathcal{X}$ of real-valued processes indexed by the natural numbers $\{1, 2, \dots\}$. Throughout this thesis, t will represent a natural number. Denote a process in $\mathcal{X}$ by $\mathbf{x}(0, \infty)$, and its truncation to the set $\{s +$

$1 \dots t\}$ by $\mathbf{x}(s,t]$ for $s < t$. When the meaning is unambiguous, $\mathbf{x}(0,\infty)$ and $\mathbf{x}(0,t]$ may be written $\mathbf{x}$. Let $\mathbf{1}$ be the constant process taking value 1 at each time step. Denote by x_t the value of the process at time t, and by $x(s,t]$ the cumulative process $x(s,t] = \sum_{i=s+1}^{t} x_i$, with $x(t,t] = 0$.

We will prove results about the limit of a sequence of random processes $(\mathbf{X}^L : L = 1 \dots \infty)$. Think of $\mathbf{X}^L$ as the average of L independent, identically distributed processes. The principal result of this chapter is a sample path large deviations principle for $\mathbf{X}^L$.

For a full introduction to the theory of large deviations, and details of the tools and definitions we will be using, see Dembo and Zeitouni [15]. For the moment, we will content ourselves with explaining what is meant by a large deviations principle.

A sequence of random variables X^L in a Hausdorff space $\mathcal{X}$ with Borel σ-algebra $\mathcal{B}$ is said to satisfy a Large Deviations Principle (LDP) with good rate function I if for any $B \in \mathcal{B}$,

$$\begin{aligned} -\inf_{x \in B^\circ} I(X) &\le \liminf_{L\to\infty} \frac{1}{L} \log \mathbb{P}(X^L \in B) \\ &\le \limsup_{L\to\infty} \frac{1}{L} \log \mathbb{P}(X^L \in B) \le -\inf_{x \in \bar{B}} I(X), \end{aligned} \tag{2.1}$$

where $I : \mathcal{X} \to \mathbb{R}^+ \cup \{\infty\}$ has compact level sets. If X is a process, this is called a sample path LDP. The left and right hand sides of this inequality are referred to as the large deviations lower and upper bounds.

2.2 Related work

The many sources limiting regime was described in an early paper of Weiss [57]. It has more recently been studied by Botvich and Duffield [4] and Courcoubetis and Weber [11] and others, whose work will be described in Chapter 3.

Another limiting regime which has been much more widely studied is the *large buffer* asymptotic, in which $\mathbf{X}^L$ is a speeded-up version of a base process $\mathbf{X}$: $X^L(0,t] = L^{-1}X(0,Lt]$. Sample path large deviations for this regime have been described by O'Connell [43]; and the proof of the LDP in this chapter is similar in outline. It turns out, as we will show, that the large buffer LDP arises as a special case of the many sources LDP. Puhalskii and Whitt [47] have also proved a large buffer sample path LDP in a similar setup.

2.3 Proving the LDP

We want to find a sample path LDP for $\mathbf{X}^L$ in a space appropriate for queueing applications. This will be done in four steps. The first step, Section 2.3.1, is to find an LDP for finite truncations of the process. If $\mathbf{X}^L$ is the average of L processes, a finite truncation is just the average of L vectors, and there are standard tools for dealing with this. The next step, Section 2.3.2, is to extend the LDP to the entire process. This is done by taking projective limits, again a standard step. The third step, Section 2.3.3, takes most of the work. Many queueing functions of interest are not continuous with respect to the projective limit topology, so we need to strengthen the LDP to a more appropriate

topology. O'Connell [43] has introduced a suitable topology: that given by the *uniform norm*

$$\|\mathbf{x}\| = \sup_{t>0} \left| \frac{x(0,t]}{t} \right|. \tag{2.2}$$

As well as choosing this finer topology we need to restrict the LDP by incorporating a notion of stability; this is the final step, in Section 2.3.4.

We will find conditions under which $\mathbf{X}^L$ satisfies an LDP, in a subset of $\mathcal{X}$ equipped with the uniform topology, and with good rate function

$$\mathbf{I}(\mathbf{x}) = \sup_{t>0} \sup_{\boldsymbol{\theta} \in \mathbb{R}^t} \boldsymbol{\theta} \cdot \mathbf{x}(0,t] - \boldsymbol{\Lambda}_t(\boldsymbol{\theta}), \tag{2.3}$$

where $\boldsymbol{\Lambda}_t(\boldsymbol{\theta})$ is the moment generating function

$$\lim_{L\to\infty} \frac{1}{L} \log \mathbb{E} \exp\left(L\boldsymbol{\theta} \cdot \mathbf{X}^L(0,t]\right).$$

2.3.1 An LDP for truncated sequences

The following lemma establishes an LDP for any finite truncation of the process. It is a direct restatement of the Gärtner-Ellis theorem for the average of vectors in $\mathbb{R}^t$ (see Dembo and Zeitouni [15] Theorem 2.3.6).

Condition 1 (Finite-time regularity)
Define the logarithmic moment generating function $\boldsymbol{\Lambda}_t^L(\boldsymbol{\theta})$ *for* $\boldsymbol{\theta} \in \mathbb{R}^t$ *by*

$$\boldsymbol{\Lambda}_t^L(\boldsymbol{\theta}) = \frac{1}{L} \log \mathbb{E} \exp\left(L\boldsymbol{\theta} \cdot \mathbf{X}^L(0,t]\right).$$

Assume that for each t *and* $\boldsymbol{\theta}$*, the limiting moment generating function*

$$\boldsymbol{\Lambda}_t(\boldsymbol{\theta}) = \lim_{L\to\infty} \boldsymbol{\Lambda}_t^L(\boldsymbol{\theta})$$

exists as an extended real number, and that the origin belongs to the interior of the effective domain of $\boldsymbol{\Lambda}_t$*. Assume further that* $\boldsymbol{\Lambda}_t$ *is an essentially smooth, lower semicontinuous function.*

Lemma 2.1 *Under Condition 1, for any fixed* t*, the sequence* $\mathbf{X}^L(0,t]$ *satisfies an LDP with good rate function*

$$\boldsymbol{\Lambda}_t^*(\mathbf{x}(0,t]) = \sup_{\boldsymbol{\theta} \in \mathbb{R}^t} \boldsymbol{\theta} \cdot \mathbf{x}(0,t] - \boldsymbol{\Lambda}_t(\boldsymbol{\theta}).$$

Throughout this thesis, we have in mind the following example.

Example 2.1 (Many Sources)
Let $\mathbf{X}^L$ be the average of L independent copies of the process $\mathbf{X}$. Then

$$\boldsymbol{\Lambda}_t(\boldsymbol{\theta}) = \boldsymbol{\Lambda}_t^L(\boldsymbol{\theta}) = \log \mathbb{E} e^{\boldsymbol{\theta} \cdot \mathbf{X}(0,t]},$$

and so Condition 1 is automatically satisfied. $\diamond$

2.3.2 The Projective Limit

Now we extend the LDP from finite truncations $\mathbf{X}(0,t]$ to the full process $\mathbf{X}(0,\infty)$. We need a little more care than this in stating the result, because the definition of large deviations principle relies on open and closed sets and there are several useful topologies on the space of processes $\mathcal{X}$. We will use the topology of projective limits, i.e. the topology of pointwise convergence of sequences. The following lemma is a direct application of the Dawson-Gärtner theorem for projective limits (see Dembo and Zeitouni [15] Theorem 4.6.1).

Lemma 2.2 *Under Condition 1, the sequence $\mathbf{X}^L$ satisfies an LDP in $\mathcal{X}$ under the topology of pointwise convergence, with good rate function*

$$\mathbf{I}(\mathbf{x}) = \sup_t \mathbf{\Lambda}_t^*(\mathbf{x}(0,t]). \tag{2.4}$$

The topology of pointwise convergence is however not directly useful for many queueing applications. For example, if x_t is the amount of work arriving at a queue at time $-t$, and the queue is served at constant rate C, then the queue size at time 0 is

$$Q(\mathbf{x}) = \sup_{t\geq 0} x(0,t] - Ct$$

and this function is not continuous with respect to the topology of pointwise convergence. To see this, set $x_t^L = C$ for $t < L$, $x_L^L = C+1$, and $x_t^L = 0$ for $t > L$. Then $\mathbf{x}^L$ converges pointwise to the constant process of rate C, for which $Q = 0$, but $Q(\mathbf{x}^L) = 1 \not\to 0$.

We need to show that the LDP holds in a finer topology, one which will make Q continuous. This is done in the next section.

2.3.3 Strengthening the topology

The uniform topology (2.2) defined above allows one to analyse a wide range of queueing problems. The idea is that it controls what happens over very large timescales. We will show that the sample path LDP of Lemma 2.2 can be extended to it, under an additional assumption on the large timescale behaviour of the process $\mathbf{X}^L$.

The results in the following chapters do not actually need a topology as strong as the uniform topology. The only properties of the topology they use are that it is stronger than the projective limit topology, and that it makes the queue size function continuous. There are weaker topologies that have these two properties, such as the *weak queue topology*, defined by the metric

$$d(\mathbf{x},\mathbf{y}) = |Q(\mathbf{x}) - Q(\mathbf{y})| + \sum_{t=1}^{\infty} \frac{1 \wedge |x_t - y_t|}{2^t}.$$

This will be useful in Chapter 4. But the uniform topology makes it easier to follow the proofs in this chapter, so we will stick with it for now.

Condition 2 (Large timescale characteristics)
A scaling function *is a function $v : \mathbb{N} \to \mathbb{R}$ for which $v(t)/\log t \to \infty$. For some scaling function v, define the scaled cumulant moment generating function*

$$\Lambda_t^L(\theta) = \frac{1}{v(t)} \mathbf{\Lambda}_t^L(\mathbf{1}\theta v(t)/t),$$

for $\theta \in \mathbb{R}$. From Condition 1, for each t there is an open neighbourhood of the origin in which the limit

$$\Lambda_t(\theta) = \lim_{L\to\infty} \Lambda_t^L(\theta)$$

exists. Assume that there is an open neighbourhood of the origin in which these limits and the limit

$$\Lambda(\theta) = \lim_{t\to\infty} \Lambda_t(\theta)$$

exist uniformly in θ.

We also know from Condition 1 that for θ in some open neighbourhood of the origin, the limit $\Lambda_t^L(\theta) - \Lambda_t(\theta) \to 0$ is uniform as $L \to \infty$. Assume that for θ in some open neighbourhood of the origin, the limit

$$\sqrt{\frac{v(t)}{\log t}} \Big(\Lambda_t^L(\theta) - \Lambda_t(\theta) \Big) \to 0 \tag{2.5}$$

is uniform in θ as $t, L \to \infty$: that is, given $\varepsilon > 0$ there is a t_0 and a L_0 such that for $t \geq t_0$ and $L \geq L_0$ and θ in the neighbourhood of the origin, expression (2.5) *is within ε of* 0.

Theorem 2.3 (Sample-path LDP for process averages)
Suppose $\mathbf{X}^L$ satisfies Conditions 1 and 2. Then it satisfies an LDP in the space of real-valued sequences $\mathcal{X}$ equipped with the uniform topology (2.2)*, with good rate function $\mathbf{I}$ given by* (2.4)*.*

Example 2.2 (Many Sources)
In the case of Example 2.1, when $\mathbf{X}^L$ is the average of L independent processes with common distribution $\mathbf{X}$, the uniformity of the limit (2.5) is guaranteed, since $\mathbf{\Lambda}_t^L = \mathbf{\Lambda}_t$. $\diamond$

Proof of Theorem 2.3. The processes $\mathbf{X}^L$ take values in the space $\mathcal{X}$ of real-valued sequences. Write $(\mathcal{X}, p)$ for $\mathcal{X}$ equipped with the projective limit topology, and $(\mathcal{X}, \|\cdot\|)$ for $\mathcal{X}$ equipped with the uniform topology. The identity map from $(\mathcal{X}, \|\cdot\|)$ to $(\mathcal{X}, p)$ is continuous; and we know that $\mathbf{X}^L$ satisfies an LDP in $(\mathcal{X}, p)$ with rate function $\mathbf{I}$. So, by the Inverse Contraction Principle (see Dembo and Zeitouni [15] Theorem 4.2.4), if $\mathbf{X}^L$ is exponentially tight in $(\mathcal{X}, \|\cdot\|)$, then it satisfies an LDP in $(\mathcal{X}, \|\cdot\|)$ with the same rate function.

It remains to show that $\mathbf{X}^L$ is exponentially tight in $(\mathcal{X}, \|\cdot\|)$: in other words that there exist compact sets K_α in $(\mathcal{X}, \|\cdot\|)$ such that

$$\lim_{\alpha\to\infty} \limsup_{L\to\infty} \frac{1}{L} \log \mathbb{P}(\mathbf{X}^L \notin K_\alpha) = -\infty.$$

Choose the sets K_α as follows. For each t, let $\mu_t = \Lambda_t'(0)$, let $d_t = \sqrt{\log t/v(t)}$, let

$$K_\alpha(t) = \left\{ \mathbf{x} \in \mathcal{X} : \frac{x(0,t]}{t} \in [\mu_t - \alpha d_t, \mu_t + \alpha d_t] \right\},$$

and choose

$$K_\alpha = \bigcap_{t \in \mathbb{N}} K_\alpha(t).$$

Exponential tightness with these K_α will be shown in the following two lemmas. □

Lemma 2.4 *The sets K_α are compact in the uniform topology.*

Proof. Because we are working in a metric space, it suffices to show that the sets K_α are sequentially compact. So, let $\mathbf{x}^k$ be a sequence of processes. Since the T-dimensional truncation of $\bigcap_{t \le T} K_\alpha(t)$ is compact in $\mathbb{R}^T$, the intersection K_α is compact under the projective topology. That is, there is a subsequence $\mathbf{x}^{j(k)}$ which converges pointwise, say to $\mathbf{x}$. It remains to show that $\mathbf{x}^j \to \mathbf{x}$ under the uniform topology.

Given any ε, since $d_t \to 0$ as $t \to \infty$, we can find t_0 such that for $t \ge t_0$, $2d_t\alpha < \varepsilon$. And since $\mathbf{x}$ and all the $\mathbf{x}^j$ are in K_α,

$$\sup_{t \ge t_0} \left| \frac{x^j(0,t]}{t} - \frac{x(0,t]}{t} \right| < \varepsilon.$$

Also, since the $\mathbf{x}^j$ converge pointwise, there exists a j_0 such that for $j \ge j_0$,

$$\sup_{t < t_0} \left| \frac{x^j(0,t]}{t} - \frac{x(0,t]}{t} \right| < \varepsilon.$$

Putting these two together gives the result. □

Lemma 2.5

$$\lim_{\alpha \to \infty} \limsup_{L \to \infty} \frac{1}{L} \log \mathbb{P}(\mathbf{X}^L \notin K_\alpha) = -\infty.$$

Proof. First, note that if

$$\lim_{\alpha \to \infty} \limsup_{L \to \infty} L^{-1} \log y_\alpha^L = -\infty,$$

and the same is true of z_α^L, then it is also true of $y_\alpha^L + z_\alpha^L$, by the principle of the largest term. Also note that

$$\mathbb{P}(\mathbf{X}^L \notin K_\alpha) \le \sum_t \mathbb{P}(X^L(0,t]/t > \mu_t + \alpha d_t) + \sum_t \mathbb{P}(X^L(0,t]/t < \mu_t - \alpha d_t).$$

We will adopt the strategy of breaking the infinite sums up into several parts: several finite timescale parts, and a long-timescale infinite part. Finite timescale parts are easy to deal with individually, and with the uniform topology we can control the behaviour of $\mathbf{X}^L$ over long timescales. This strategy is also at the core of proofs for related large deviations results, proved directly by Courcoubetis and Weber [11] and Botvich and Duffield [4].

First, fix t and consider $\limsup_L L^{-1} \log \mathbb{P}(X^L(0,t]/t > \mu_t + \alpha d_t)$. By Chernoff's bound,

$$\mathbb{P}(X^L(0,t]/t > \mu_t + \alpha d_t) \leq \exp\Big[-Lv(t)\big(\theta(\mu_t + \alpha d_t) - \Lambda_t^L(\theta)\big)\Big]$$

for any $\theta > 0$. So the expression we are interested in is bounded above by $\limsup_L -v(t)(\theta(\mu_t + \alpha d_t) - \Lambda_t^L(\theta))$. Choosing any θ for which $\Lambda_t(\theta)$ is finite, it is clear that this quantity tends to $-\infty$ as $\alpha \to \infty$.

Now for the remaining terms. We have assumed that the limits $\Lambda_t^L(\theta) \to \Lambda_t(\theta)$ and $\Lambda_t(\theta) \to \Lambda(\theta)$ exist uniformly in θ in an open neighbourhood of the origin. Since Λ_t^L is a cumulant moment generating function it has a power series expansion, and so the coefficients in the power series also converge. Let $\Lambda_t^L(\theta) = \theta\mu_t^L + \frac{1}{2}\theta^2 s_t^L + O(\theta^3)$, and denote the coefficients of Λ_t and Λ by dropping the superscripts and subscripts appropriately.

For fixed t_0, consider the remaining terms

$$\lim_{\alpha\to\infty} \limsup_{L\to\infty} \frac{1}{L} \log \sum_{t \geq t_0} \exp\Big[-Lv(t)\big(\theta(\mu_t + \alpha d_t) - \Lambda_t^L(\theta)\big)\Big]. \tag{2.6}$$

Assume for the moment that $s > 0$, and pick θ depending on L and t: $\theta_t^L = (d_t + \varepsilon_t^L)/s_t^L$, where $\varepsilon_t^L = \mu_t - \mu_t^L$. This gives as the typical exponent

$$-Lv(t)\left[\left\{\frac{(d_t + \varepsilon_t^L)^2}{2s_t^L} + O(d_t + \varepsilon_t^L)^3\right\} + \frac{\alpha - 1}{s_t^L} d_t(d_t + \varepsilon_t^L)\right].$$

Because of our assumption on the uniformity of convergence (2.5), there exists a t_0 and L_0 such that for $t \geq t_0$ and $L \geq L_0$, θ_t^L is positive; and because $d_t \to 0$, the term in brackets $\{\cdot\}$ is also positive. (If $s = 0$, pick $\theta_t^L = d_t + \varepsilon_t^L$; then the same conclusion holds.)

So the typical exponent in (2.6) is bounded above by

$$-Lv(t)\left[\frac{\alpha - 1}{s_t^L} d_t(d_t + \varepsilon_t^L)\right]$$

for sufficiently large t and L. Indeed, for sufficiently large t and L we can bound it by $-Lv(t)\kappa(\alpha - 1)d_t^2$ for some constant $\kappa > 0$. Therefore, by our choice of d_t, for t_0 sufficiently large, expression (2.6) is bounded above by

$$\lim_{\alpha\to\infty} \limsup_{L\to\infty} \frac{(\alpha - 1)\kappa}{L} \log \sum_{t \geq t_0} t^{-L}.$$

It is easy to check that this is equal to $-\infty$. □

2.3.4 Stability

We have achieved the goal of a sample path LDP for averages of processes. But it is still not directly useful for queueing applications, because the queue size function is still not continuous on $\mathcal{X}$, even with respect to the finer topology. The problem is that there is no notion of stability. If the mean arrival rate is higher than the service rate, the queue will be unstable. Mathematically speaking, the queue size function is only continuous on the subspace of processes for which the mean arrival rate is less than the service rate. Similar stability conditions crop up again and again, so it will be useful to give the following theorem, which shows that the sample path LDP holds in this restricted space of processes.

Definition 3 (Stability) *Define the* mean rate *of the* $\mathbf{X}^L$ *to be the derivative* $\Lambda'(0)$. *Say that* $\mathbf{X}^L$ *is* stationary *if the limiting moment generating functions* $\mathbf{\Lambda}_t$ *correspond to a stationary process.*

Note that if $\mathbf{X}^L$ is stationary, then the mean rate is simply $\lim_{L\to\infty} \mathbb{E}X_1^L$.

Theorem 2.6 *Under Conditions 1 and 2, the LDP of Theorem 2.3 holds on the space* $\mathcal{X}_\mu$, *which has the uniform topology and is given by*

$$\mathcal{X}_\mu = \Big\{ \mathbf{x} \in \mathcal{X} : \frac{x(0,t]}{t} \le \mu \quad \textit{eventually} \Big\},$$

for any μ *greater than the mean rate of the* $\mathbf{X}^L$.

Proof. By Dembo and Zeitouni [15] Lemma 4.1.5, it suffices to show that $\{\mathbf{x} : I(\mathbf{x}) < \infty\} \subset \mathcal{X}_\mu$, and for L sufficiently large, $\mathbb{P}(\mathbf{X}^L \in \mathcal{X}_\mu) = 1$.

Recall that $I(\mathbf{x}) = \sup_t \mathbf{\Lambda}_t^*(\mathbf{x}(0,t])$. Let $\mu = \Lambda'(0) + \varepsilon$, and pick $\theta > 0$ such that $\Lambda(\theta) < \theta(\mu - \frac{1}{2}\varepsilon)$. Now if $x(0,t]/t > \mu$, then for sufficiently large t,

$$\mathbf{\Lambda}_t^*(\mathbf{x}(0,t]) = \sup_{\boldsymbol{\theta}} \boldsymbol{\theta} \cdot \mathbf{x}(0,t] - \mathbf{\Lambda}_t(\boldsymbol{\theta}) \ge \theta v(t) \Big(\frac{x(0,t]}{t} - (\mu - \tfrac{1}{2}\varepsilon) \Big) \ge \tfrac{1}{2}\theta v(t)\varepsilon.$$

So if $\mathbf{x} \notin \mathcal{X}_\mu$ then this inequality holds for infinitely many t, and since $v(t)$ is unbounded, $I(\mathbf{x}) = \infty$.

Second, since $\Lambda_t^L(\theta) \to \Lambda_t(\theta)$ uniformly for t sufficiently large, and $\Lambda_t(\theta) \to \Lambda(\theta)$, there exists $\theta > 0$ such that for L and t sufficiently large, $\Lambda_t^L(\theta) < \theta(\mu - \frac{1}{2}\varepsilon)$. Then, by Chebychev's inequality,

$$\sum_{t=1}^{\infty} \mathbb{P}\Big(\frac{X^L(0,t]}{t} > \mu \Big) \le \sum_{t=1}^{\infty} \exp\Big(-Lv(t)(\theta\mu - \Lambda_t^L(\theta)) \Big)$$

which is finite for L sufficiently large. So, by the Borel-Cantelli lemma, $\mathbb{P}(\mathbf{X}^L \in \mathcal{X}_\mu) = 1$. □

2.4 Examples

We have already given the example of the many sources asymptotic, in which $\mathbf{X}^L$ is the average of L independent processes. We now give three more examples. The first shows how large buffer results can be obtained from the same theorems (though they usually turn out to have a less rich structure).

Example 2.3 (Large Buffer)
Given a base process $\mathbf{X}$, let $X^L(0,t] = f(L)^{-1}X(0,f(L)t]$. This is the *large buffer asymptotic* regime. For a variety of processes $\mathbf{X}$ it is possible to choose a normalising function $f(L)$ such that Condition 1 is satisfied. Often, the normalising function is just $f(L) = L$, and the limit $\mathbf{\Lambda}_t$ has the simple linear form $\mathbf{\Lambda}_t(\boldsymbol{\theta}) = \sum_{i=1}^t \mathbf{\Lambda}_1(\theta_i)$. For an account of conditions under which this occurs, see Dembo and Zajic [14]. In Example 2.5 below, the normalising function is not linear and $\mathbf{\Lambda}_t$ has a more complicated form.

Suppose for now that $\mathbf{\Lambda}_t$ has the simple linear form: this gives as the rate function $\mathbf{I}(\mathbf{x}) = \sum_t \mathbf{\Lambda}_1^*(x_t)$. Then Condition 2 is satisfied. To see this, choose $v(t) = t$, so that $\Lambda(\theta) = \mathbf{\Lambda}_1(\theta)$. Since $\Lambda_t^L(\theta)$ is given by

$$\Lambda_t^L(\theta) = \frac{1}{Lt}\log\mathbb{E}\exp\big(\theta X(0,Lt]\big),$$

and we have assumed that this converges as $L \to \infty$, we can by choosing t and L sufficiently large make $\Lambda_t^L(\theta) - \Lambda_t(\theta)$ arbitrarily small. Thus the limit (2.5) is uniform as $t, L \to \infty$. O'Connell [43] describes sample path large deviations under the large buffer asymptotic in more detail. $\diamond$

The second example is of fractional Brownian motion, a process with long-range dependence, by which we mean that the sum of covariance coefficients $\sum_{i=0}^{\infty}\mathrm{Cov}(X_0, X_i)$ is infinite. This makes it both appealing as a model for Internet traffic, since this phenomenon has been observed empirically by Leland et al. [33] and others, and also a problem for the standard large buffer asymptotic. But under the many sources asymptotic, it looks just like any other process.

Example 2.4 (Fractional Brownian Motion with Many Sources)
As an illustration of the many sources asymptotic, let $\mathbf{X}^L$ be the average of L independent copies of the process $\mathbf{X}$, defined by $X(0,t] = \lambda t + \sigma Z_t$ where Z_t is a fractional Brownian motion with Hurst parameter H. Then $\mathbf{\Lambda}_t(\boldsymbol{\theta}) = \lambda\boldsymbol{\theta}\cdot\mathbf{1} + \frac{1}{2}\sigma^2\boldsymbol{\theta}\cdot S_t\boldsymbol{\theta}$, where the $t\times t$ matrix S_t is given by $(S_t)_{ij} = \frac{1}{2}(|j-i-1|^{2H} + |j-i+1|^{2H} - 2|j-i|^{2H})$, and so $\mathbf{\Lambda}_t(\theta\mathbf{1}) = \lambda\theta t + \frac{1}{2}\sigma^2\theta^2 t^{2H}$.

To check that Condition 2 is satisfied, choose the scaling function $v(t) = t^{2(1-H)}$, so that $\Lambda_t^L(\theta) = \lambda\theta + \frac{1}{2}\sigma^2\theta^2$. This does not depend on L or t, so it is also equal to $\Lambda_t(\theta)$ and $\Lambda(\theta)$. $\diamond$

Example 2.5 (Fractional Brownian Motion with Large Buffer)
To contrast the many sources and the large buffer asymptotic, consider the large buffer version of fractional Brownian motion. Let $\mathbf{X}$ be a fractional Brownian motion with Hurst parameter H, as in the previous example. Choose the scaling $X^L(0,t] = f(L)^{-1}X(0,f(L)t]$ with $f(L) = L^{1/2(1-H)}$. This gives $\mathbf{\Lambda}_t^L(\theta\mathbf{1}) = \mathbf{\Lambda}_t(\theta\mathbf{1}) = \lambda\theta t + \frac{1}{2}\sigma^2\theta^2 t^{2H}$, the same expression as before. This is not linear in t, so $\mathbf{\Lambda}_t(\boldsymbol{\theta})$ does not have the simple linear form described in Example 2.3.

For Condition 2, as with any large buffer example the limit (2.5) is uniform for any scaling function v, and as in Example 2.4 we can choose $v(t) = t^{2(1-H)}$.

Applying the results in Chapter 3 to the LDP we obtain from this, we can rederive a result of Duffield and O'Connell [19] for the workload in a queue fed by a single fractional Brownian motion source. $\diamond$

The final example is of Moderate Deviations. This refers to a family of results somewhere between the Central Limit Theorem and Large Deviations. There is not yet a standard reference for moderate deviations; see de Acosta [12] and Deo and Babu [16] for related results.

Suppose that $\mathbf{X}^L$ is the average of L independent processes distributed like $\mathbf{X}$, and let $\boldsymbol{\mu} = \mathbb{E}\mathbf{X}$. The central limit theorem looks at the limiting behaviour of $L^{1/2}(\mathbf{X}^L - \boldsymbol{\mu})$; it produces estimates based on the normal distribution and involving only the mean and covariance. Large deviations on the other hand can be thought of as looking at the limiting behaviour of $(\mathbf{X}^L - \boldsymbol{\mu})$; the estimates it produces involve the entire distribution, but they are simple because they depend only on the most likely path.

Moderate deviations sits between these: it looks at the limiting behaviour of $L^{\gamma/2}(\mathbf{X}^L - \boldsymbol{\mu})$ for $0 < \gamma < 1$, and produces estimates involving only the mean and covariance and depending only on the most likely path. To be precise, let us say that $\mathbf{X}^L$ satisfies a moderate deviations principle if

$$\frac{1}{L^{1-\gamma}} \log \mathbb{P}(L^{\gamma/2}(\mathbf{X}^L - \boldsymbol{\mu}) \in B) \tag{2.7}$$

satisfies the upper and lower large deviations bounds (2.1), with a good rate function which depends only on the covariance structure.

Example 2.6 (Moderate Deviations)
Let

$$\mathbf{Y}^N = \sqrt{N^{\gamma/(1-\gamma)}}(\mathbf{X}^{N^{1/(1-\gamma)}} - \boldsymbol{\mu}).$$

If $\mathbf{Y}^N$ satisfies the conditions of Theorem 2.3 we obtain estimates of the quantity (2.7), where $L = N^{1/(1-\gamma)}$.

Further, we know that the log moment generating function for $\mathbf{X}^L(0,t]$ does not depend on L since $\mathbf{X}^L$ is assumed to be the average of independent copies of $\mathbf{X}$. Let it be

$$\boldsymbol{\Lambda}_t^L(\boldsymbol{\theta}) = \boldsymbol{\theta} \cdot \boldsymbol{\mu}_t + \tfrac{1}{2}\boldsymbol{\theta} \cdot \Gamma_t \boldsymbol{\theta} + \cdots .$$

Then the log moment generating function for $\mathbf{Y}^L(0,t]$ is

$$\mathbf{M}_t^L(\boldsymbol{\theta}) = \tfrac{1}{2}\boldsymbol{\theta} \cdot \Gamma_t \boldsymbol{\theta} + O\big(1/\sqrt{L}\big)$$

and so the rate function, which depends on the limit $\mathbf{M}_t$, only involves the covariance structure Γ_t. $\diamond$

We shall revisit this example in the next chapter, to see what moderate deviations tells us about queue size.

2.5 Summary

For most of the rest of this thesis, all that matters from this chapter is the following, a restatement of Theorem 2.6. In Chapter 4 we need to pay a little more attention to the conditions under which this theorem is satisfied, but for the rest all we need is the notation and the result.

Consider the space $\mathcal{X}$ of real-valued processes $\mathbf{x} = (x_1, x_2, \dots)$ indexed by the natural numbers. Write $\mathbf{x}(0,t]$ for $(x_1, \dots x_t)$, and $x(0,t]$ for $x_1 + \cdots + x_t$. Consider a sequence of random processes $\mathbf{X}^L$ in $\mathcal{X}$.

Theorem 2.7 (Sample path LDP) *Under Conditions 1 and 2 (on pages 8 and 10),* $\mathbf{X}^L$ *satisfies a large deviations principle with good rate function*

$$\mathbf{I}(\mathbf{x}) = \sup_{t>0} \sup_{\boldsymbol{\theta} \in \mathbb{R}^t} \boldsymbol{\theta} \cdot \mathbf{x}(0,t] - \boldsymbol{\Lambda}_t(\boldsymbol{\theta}),$$

where $\boldsymbol{\Lambda}_t(\boldsymbol{\theta})$ *is the moment generating function*

$$\lim_{L\to\infty} \frac{1}{L} \log \mathbb{E} \exp\left(L\boldsymbol{\theta} \cdot \mathbf{X}^L(0,t]\right),$$

in the space

$$\mathcal{X}_\mu = \left\{ \mathbf{x} \in \mathcal{X} : \frac{x(0,t]}{t} \leq \mu \quad \textit{eventually} \right\},$$

equipped with the uniform norm

$$\|\mathbf{x}\| = \sup_{t>0} \left| \frac{x(0,t]}{t} \right|,$$

for any μ *greater than the mean rate of the* $\mathbf{X}^L$ *(Definition 3 on page 13).*

This result will be used to study the large deviations behaviour of a variety of queueing systems. It lets us estimate the probabilities of events we are interested in, and also gives a good idea of how those events are likely to occur. Some of the systems can easily be studied directly—but the indirect route, via this sample path LDP, can give more insight. It also means there is less additional work for each different application.

Chapter 3

Queues

In this chapter we use the sample path LDP of Chapter 2 to study large deviations in three different queueing problems: in Section 3.3 we study overflow in standard first-in–first-out queues with finite and infinite buffers; in Section 3.4 we study the sample paths that lead to overflow; and in Section 3.5 we study overflow in queues which give some flows priority over others. There are many other possible applications: for example, in Chapter 5 we use it to analyse control algorithms for routers.

The common approach will be to take the sample path LDP and then apply the Contraction Principle to find an LDP for the quantity of interest. The contraction principle says that if $\mathbf{X}^L$ satisfies the sample path LDP in $\mathcal{X}_\mu$ with rate function $\mathbf{I}$, and if f is a continuous function on $\mathcal{X}_\mu$, then $f(\mathbf{X}^L)$ satisfies a LDP with good rate function $I(y) = \inf\{\mathbf{I}(\mathbf{x}) : \mathbf{x} \in \mathcal{X}_\mu, f(\mathbf{x}) = y)\}$. See Dembo and Zeitouni [15] Theorem 4.2.1 for a proof.

In Section 3.6 we describe our results in the more practical language of effective bandwidths. First, though, we relate the abstract setting of the last section to queueing models, and describe the limiting regime.

3.1 The queueing model

Consider a sequence of queues, indexed by L, in which the Lth queue has service rate C and buffer size B. Let X_t^L be the total amount of work arriving at the Lth queue at time $-t$. (Depending on the context, $\mathbf{X}$ will variously be called an input process, a source, or a traffic flow.)

There are several ways in which we can interpret this, depending on what $\mathbf{X}^L$ represents, though none of the results in this section rely on a particular interpretation. Here are three possibilities, corresponding to three examples from the previous chapter.

The first example is the one we have in mind throughout this thesis: when the total input flow is the aggregate of many independent input flows. This sort of scaling is well-suited to modern telecommunications networks, in which a router may have thousands of inputs.

Example 3.1 (Many Sources)
In the many sources asymptotic, $\mathbf{X}^L$ is the average of L independent identically distributed flows. So the Lth queue can be thought of as multiplexing together

L different flows, with its resources growing in proportion: it has service rate LC and buffer size LB. $\diamond$

The next example has been much more widely studied. For Markov modulated fluid sources and for many others, the probability of loss decays exponentially in buffer size, so a good way to reduce loss is to make the buffers larger; and it is natural to study asymptotic regimes in which the buffer size increases. The observation that this is largely inaccurate when there are many input flows or when the sources exhibit long-range dependence (see Choudhury et al. [7] and Leland et al. [33] for example) has prompted some of the work on the many sources asymptotic.

Example 3.2 (Large Buffer)
In the large buffer asymptotic, described in Example 2.3, $\mathbf{X}^L$ is a speeded up version of a base process: $X^L(0,t] = f(L)^{-1}X(0, f(L)t]$. So the Lth queue can be thought of as having a single input $\mathbf{X}$ and fixed service rate C, but increasing buffer size $f(L)B$. $\diamond$

The final example looks at a different sort of limit, in which the impacts of the mean arrival rate and burstiness are treated differently. It has some appealing features: the probability of overflow depends on the input processes only through their mean and covariance structure, which makes calculations easier.

Example 3.3 (Moderate Deviations)
Moderate deviations theory, described in Example 2.6, lies between large deviations theory and the central limit theorem. Let $\mathbf{X}^L = \sqrt{M}(\mathbf{Y}^M - \boldsymbol{\mu})$ where $\mathbf{Y}^M$ is the average of M independent stationary sources distributed like $\mathbf{Y}$, $\boldsymbol{\mu} = \mu\mathbf{1} = \mathbb{E}\mathbf{Y}$, and $M = L^{\gamma/(1-\gamma)}$. So the Lth queue can be thought of as having L independent input flows each distributed like $\mathbf{Y}$, service rate $L\mu + L^{(1-\gamma/2)}C$ and buffer size $L^{(1-\gamma/2)}B$. $\diamond$

3.2 Related work

The work in this chapter and the one preceding was motivated by the results of Courcoubetis and Weber [11] and Botvich and Duffield [4], who find large deviations rate functions for the amount of work in a queue and the event of overflow. Duffield [17] has treated separately the case of nonlinear scaling functions. These authors proved their results directly, but we will start with the sample path LDP and apply the contraction principle. Ours is a more general method, and it lets us fill in some gaps: in particular, we give the large deviations rate function for workload in a queue with a finite buffer. Simonian and Guibert [52] obtain similar results for a special class of input processes, Markov-modulated fluid sources. Botvich and Duffield describe both continuous time and discrete time models, but we will restrict our attention to the discrete case.

The large deviations estimates for overflow probability can be refined using the Bahadur-Rao improvement as described by Likhanov and Mazumdar [34]. Their techniques involve a lot more technical details and give only a little extra insight, so we stick with large deviations.

Another benefit of the sample path LDP approach is that it tells us the most likely sample path to overflow. Weiss [57], who introduced the many sources asymptotic, obtained similar results for the special case of an on-off Markov source using direct methods; and Mandjes and Ridder [40] have too for Markov-modulated sources and periodic sources. Our results hold much more generally.

The contraction principle approach has been applied widely to the large buffer asymptotic, described in Example 3.2. See O'Connell [42, 43], Duffield and O'Connell [19] and Paschalidis [46] for examples. We will see that under the many sources regime, large deviations often possess a richer structure.

The final queueing problem studied in this chapter, that of the priority queue, has been studied by Berger and Whitt [2], who independently obtained similar results for the large buffer asymptotic. Related queueing models under that asymptotic are described by Kulkarni et al. [32] and O'Connell [45].

3.3 Buffer size in a queue

In this section we look at a standard queue with a constant service rate. The following results have previously been proved directly; but it is instructive to see the techniques used in deriving them from the sample path LDP, as these same techniques will be used in the following sections.

Consider a queue with constant service rate C fed with input process $\mathbf{x}$. The amount of work in the queue at time $-s$ may be defined to be $\lim_{t\to\infty} Q_s(\mathbf{x}(0,t])$, where $Q_s(\mathbf{x}(0,t])$ is given by the Lindley recursion

$$Q_{s-1} = \left(Q_s + x_s - C\right)^+, \quad Q_t = 0.$$

If the input is a stationary process, the stationary queue size may be written as

$$Q(\mathbf{x}) = \sup_t x(0,t] - Ct.$$

Lemma 3.7 shows that this function is continuous on $\mathcal{X}_\mu$ for any $\mu < C$. By the Contraction Principle, this immediately gives Corollary 3.1: an LDP for workload in queues with infinite buffers, which when simplified duplicates the results of Botvich and Duffield [4] for linear scaling functions $v(t)$, of Duffield [17] for general scaling functions, and of Simonian and Guibert [52] for the special case of Markov-modulated fluid sources.

Corollary 3.1 *Under the conditions of Theorem 2.7, if $\mathbf{X}^L$ has mean rate less than C then $Q(\mathbf{X}^L)$ satisfies an LDP with good rate function*

$$I(b) = \inf_{\mathbf{x}\in\mathcal{X}_C : Q(\mathbf{x})=b} \mathbf{I}(\mathbf{x}).$$

Proof. The only point to note is that the infimum is taken over $\mathcal{X}_C$. But it might as well have been taken over $\mathcal{X}_\mu$ for any μ greater than the mean rate and less than C, since the rate function will be infinite on $\mathcal{X}_C \setminus \mathcal{X}_\mu$ by Corollary 2.6. □

We can do the same thing for queues with finite buffers. The queue size $\bar{Q}$ in a queue with a finite buffer B is defined similarly to Q, except that it cannot

fill to greater than B and any excess work is discarded. This is expressed by the recursion

$$\bar{Q}_{s-1} = \left(\bar{Q}_s + x_s - C\right)^+ \wedge B, \quad \bar{Q}_t = 0.$$

Lemma 3.7 also shows that $\bar{Q}$ is a continuous function of the input process, and so we obtain Corollary 3.2: an LDP for workloads in queues with finite buffers.

Corollary 3.2 *Under the conditions of Theorem 2.7, if $\mathbf{X}^L$ has mean rate less than C then $\bar{Q}(\mathbf{X}^L)$ satisfies an LDP with good rate function*

$$\bar{I}(b) = \inf_{\mathbf{x}\in\mathcal{X}_C : \bar{Q}(\mathbf{x})=b} \mathbf{I}(\mathbf{x}).$$

These expressions for the rate functions are not very informative, and so Theorem 3.3 gives a more manageable expression for $I(b)$. In fact, if the process is stationary, then for $b \leq B$, $\bar{I}(b)$ and $I(b)$ are identical (and for $b > B$, $\bar{I}(b) = \infty$); this is shown in Theorem 3.4. The proofs of these theorems are deferred to the end of this section.

Theorem 3.3 *Under the conditions of Theorem 2.7, if $\mathbf{\Lambda}'_t(\theta\mathbf{1}) < Ct$ at $\theta = 0$ for all t, then $I(b)$ is increasing in b and is given by*

$$I(b) = \inf_{\mathbf{x}\in\mathcal{X}_C : Q(\mathbf{x})=b} \mathbf{I}(\mathbf{x}) \tag{3.1}$$

$$= \inf_t \inf_{\mathbf{x}\in\mathbb{R}^t : x(0,t]=b+Ct} \mathbf{\Lambda}^*_t(\mathbf{x}(0,t]) \tag{3.2}$$

$$= \inf_t \sup_\theta \theta(b + Ct) - \mathbf{\Lambda}_t(\theta\mathbf{1}). \tag{3.3}$$

Theorem 3.4 *If $I(b)$ is finite, then the optimal timescale $\hat{t}$ and the optimizing path $\hat{\mathbf{x}}(0,\hat{t}]$ are both attained; and if the optimal spacescale $\hat{\theta}$ is attained then*

$$\hat{\mathbf{x}}(0,\hat{t}] = \nabla\mathbf{\Lambda}_{\hat{t}}(\hat{\theta}\mathbf{1}).$$

For a queue with a finite buffer B and stationary input whose mean rate is less than C, if $b \leq B$ then $\bar{I}(b) = I(b)$ and the same path $\hat{\mathbf{x}}$ is optimal.

The optimal $\hat{\theta}$ and $\hat{t}$ appearing in Theorems 3.3 and 3.4 are called the *operating point* or the *critical spacescale* and *timescale* of the queue. Courcoubetis et al. [10] give a detailed account, with simulation results, of how they are affected by the traffic mix and the queue parameters under the many sources asymptotic regime.

Examples

To illustrate the different forms that this rate function can take, we will go back to the three examples of Section 3.1—the many sources asymptotic, the large buffer asymptotic, and moderate deviations—paying particular attention to the interpretation of the critical timescale.

Example 3.4 (Many Sources)
As in Example 3.1, consider a sequence of queues indexed by L in which the Lth queue Q^L is fed by an aggregate $L\mathbf{X}^L$ of L independent inputs and has service rate LC, and suppose the event of interest is that the queue size reaches Lb. As in Example 2.4, let each source be a fractional Brownian motion with mean rate λ and Hurst parameter H. We can calculate the critical spacescale and timescale:

$$\hat{\theta} = \frac{b + (C-\lambda)\hat{t}}{\sigma^2 \hat{t}^{2H}} \quad \text{and}$$
$$\hat{t} = \frac{b}{C-\lambda}\frac{H}{1-H}$$

(or rather, $\hat{t}$ is an integer close to this value; but we will ignore this minor complication.) This gives rate function

$$I(b) = \frac{1}{2\sigma^2} b^{2(1-H)} (C-\lambda)^{2H} \left(\frac{H}{1-H}\right)^{2(1-H)} \frac{1}{H^2}$$

and large deviations approximation

$$\log \mathbb{P}\big(Q^L(L\mathbf{X}^L) = Lb\big) \approx -LI(b) \quad \text{for large } L.$$

◇

Under the large buffer asymptotic the rate function is exactly the same, but it has a very different interpretation, as we now illustrate.

Example 3.5 (Large Buffer)
Instead of a sequence of queues we will consider a single queue with fixed service rate C and fed by a single input flow $\mathbf{X}$, as in Example 3.2. Let the input flow again be a fractional Brownian motion, and consider the event that the queue size reaches $f(L)b$, where $f(L) = L^{1/2(1-H)}$.

As we saw in Example 2.5, the moment generating function $\mathbf{\Lambda}_t$ is exactly the same as for the many sources asymptotic, and so the rate function $I(b)$ is the same too. This similarity disguises the fact that the results have very different interpretations. To see this, note that b is just a scaling factor so we may as well set $b = 1$, and let $\beta = f(L)$. Then the large deviations approximation amounts to

$$\log \mathbb{P}\big(Q(\mathbf{X}) = \beta\big) \approx -\beta^{2(1-H)} I(1) \quad \text{for large } \beta.$$

Notice that when $H = \frac{1}{2}$ the decay is exponential in β: many other sources including Markov-modulated fluid sources share this exponential decay. But when $H > \frac{1}{2}$ the source has long-range dependence and the decay is less than exponential, which means that increasing the buffer size does not give as much of a reduction in loss probability. This phenomenon was observed in real network traffic by Leland et al. [33], and it has stimulated much interest in long-range dependent traffic models. But as we saw in the last example, it makes no difference to the many sources approximation whether $H = \frac{1}{2}$ or $H > \frac{1}{2}$. ◇

There are some noteworthy differences between the many sources and large buffer asymptotics as regards the critical timescale $\hat{t}$ identified in Theorem 3.3. We illustrate the differences in the next example.

Example 3.6 (Timescales)
In the many sources asymptotic, the timescale $\hat{t}$ is easy to interpret: it is the length of time which the buffer is most likely to take to fill from empty to a given level Lb. In the large buffer asymptotic, $\hat{t}$ has a slightly different interpretation. It is a scaling parameter which relates the buffer level $f(L)b$ to the time taken to reach that level, $f(L)\hat{t}$.

In the latter case, the time taken to fill the buffer tends to infinity and so the rate function $I(b)$ depends only on the infinite-time characteristics of the source. For Markov-modulated fluid sources (and many other sources which satisfy conditions described by Dembo and Zajic [14]), it is appropriate to take $f(L) = L$ and so $\mathbf{\Lambda}_t(\theta\mathbf{1}) = t\lim_{L\to\infty} L^{-1}\log\mathbb{E}\exp(\theta X(0,L])$. Then the rate function $I(b)$ simplifies to $I(b) = \sup_\theta \theta b$, where the supremum is taken over all θ such that $\mathbf{\Lambda}_1(\theta) \leq C$.

By contrast, under the many sources asymptotic the rate function depends on the characteristics of the source $\log\mathbb{E}\exp(\theta X(0,t])$ over all timescales t. $\diamond$

Our final example is a moderate deviations result for fractional Brownian motion. The distinguishing feature of moderate deviations results is that the rate function $I(b)$ depends only on the mean and covariances of the source, but since Gaussian sources are completely characterized by their means and covariances this feature is not apparent here. We wish instead to draw attention to the way that in moderate deviations the mean and the covariances are treated differently.

Example 3.7 (Moderate deviations)
As in Example 3.3 consider a sequence of queues indexed by L, where the Lth queue is fed by L independent sources and has service rate $L\lambda + L^{(1-\gamma/2)C}$, and suppose the event of interest is that the queue size reaches $L^{(1-\gamma/2)}b$. As before, let each source be a fractional Brownian motion of mean rate λ and Hurst parameter H.

As noted in Example 2.6 the generating function $\mathbf{\Lambda}_t$ depends only on the covariance structure, and one can calculate $\mathbf{\Lambda}_t(\theta\mathbf{1}) = \frac{1}{2}\sigma^2\theta^2t^{2H}$. This gives a rate function $I(b)$ similar to that in Example 3.4, but without the λ.

The reason for this difference is that in setting the service rate to $L\lambda + L^{(1-\gamma/2)}C$ we are assuming that the queue is already provisioned to cope with the mean rate, and so any loss is attributable to the variance of the input. $\diamond$

More LDPs

There are actually three more LDPs which are useful, but which are easily confused with Corollaries 3.1 and 3.2. The first gives the probability that a queue with an infinite buffer is non-empty. At first sight, we can find this from Corollary 3.1: just consider the event $b > 0$. But the large deviations upper bound we get is useless, because it involves the closure of this set—which is $b \geq 0$, the entire space. So for a better bound, we can go back to the sample path LDP and look at the closure of the set of sample paths for which $Q(\mathbf{x}) > 0$, now not the entire space. The same technique can be used for the events that a queue with a finite buffer is non-empty or overflows. The infinite buffer result has been proved by Botvich and Duffield [4], and the finite buffer results have been proved by Courcoubetis and Weber [11]. The proof of Corollary 3.5 is

deferred to the end of this section. The proof of Corollary 3.6 is similar, and is omitted.

Corollary 3.5 *Under the conditions of Theorem 2.7, if $\mathbf{X}^L$ has mean rate less than C, then the event $\{Q > 0\}$ has large deviations lower bound $-I(0^+)$ and upper bound $-I^+(0)$. If in addition $B > 0$ then the event $\{\bar{Q} > 0\}$ has the same large deviations bounds. Here, $I(b^+) = \lim_{a \downarrow b} I(b)$ and $I^+(0)$ is given by*

$$I^+(0) = \sup_\theta \theta C - \mathbf{\Lambda}_1(\theta \mathbf{1}).$$

Corollary 3.6 *Under the conditions of Theorem 2.7, if $\mathbf{X}^L$ is stationary and has mean rate less then C, then the event that $\bar{Q}$ overflows has large deviations lower bound $-I(B^+)$ and upper bound $-I(B)$ (or $-I^+(0)$ if $B = 0$).*

Proofs

The rest of this section is given over to proofs.

Lemma 3.7 *The queue size functions Q and $\bar{Q}$ are continuous on $\mathcal{X}_\mu$, if $\mu < C$.*

Proof. Consider a sequence of processes $\mathbf{x}^k \to \mathbf{x}$ in $\mathcal{X}_\mu$ under the uniform topology. That is, given ε, there is a k_0 such that for $k \geq k_0$,

$$\sup_t \left| \frac{x^k(0,t]}{t} - \frac{x(0,t]}{t} \right| < \varepsilon.$$

And since $\mathbf{x} \in \mathcal{X}_\mu$, there is a t_0 such that for $t \geq t_0$,

$$x(0,t]/t < \mu.$$

Then for $k \geq k_0$ and $t \geq t_0$, choosing $\varepsilon = C - \mu$,

$$x^k(0,t]/t < C$$

and the same holds for $\mathbf{x}$. So the expression for queue size Q simplifies: for $k \geq k_0$, $Q(\mathbf{x}^k) = Q(\mathbf{x}^k(0,t_0])$, and the same holds for $\mathbf{x}$. Thus for $k \geq k_0$,

$$|Q(\mathbf{x}^k) - Q(\mathbf{x})| = |\sup_{t \leq t_0}(x^k(0,t] - Ct) - \sup_{t \leq t_0}(x(0,t] - Ct)|$$

which tends to 0 as $k \to \infty$.

Now for $\bar{Q}$. Since $Q(\mathbf{x}) = Q(\mathbf{x}(0,t_0])$, the infinite-buffer queue must empty at some time in $[-t_0, 0]$. For suppose it does not. Let $s \leq t_0$ be the last time at which the queue, started from empty at $-t_0$, is empty; then $Q(\mathbf{x}(0,t_0]) = Q(\mathbf{x}(0,s]) = x(0,s] - Cs$. But $Q(\mathbf{x}) = q + x(0,s] - Cs$ where $q > 0$ is the queue size at time $-s$, leading to a contradiction.

So Q empties at some time in $[-t_0, 0]$. So too must $\bar{Q}$, because $\bar{Q} \leq Q$. In other words, $\bar{Q}(\mathbf{x}) = \bar{Q}(\mathbf{x}(0,t_0])$. The same holds for $\mathbf{x}^k$ for k sufficiently large, and so we deduce that $\bar{Q}$ is also continuous. □

Proof of Theorem 3.3. If $b = 0$, then (3.2) and (3.3) take the value 0 at $t = 0$. Now consider the sample path given by $\mathbf{x}(0,t] = \nabla \mathbf{\Lambda}_t(\mathbf{0})$. This is constant,

taking the value of the mean arrival rate, so $Q(\mathbf{x}) = 0$. And it has rate $\mathbf{I}(\mathbf{x}) = 0$, so (3.1) also takes the value 0. So restrict attention to the case $b > 0$.

Note that because $b + Ct$ is greater than $\mathbf{\Lambda}'_t(\theta\mathbf{1})$ at $\theta = 0$, we may take the supremum only over $\theta \geq 0$; thus (3.3) is increasing in b.

First, (3.2) = (3.3). Fix t. Then $\mathbf{X}^L(0,t] \cdot \mathbf{1}$ is just a real-valued random variable, and from Condition 1 it satisfies an LDP with good rate function given by the expression in (3.3). Another way of finding this is by contracting from the sample path LDP for $\mathbf{X}^L(0,t]$, which gives as rate function the expression in (3.2). By the uniqueness of the rate function, these are equal.

Next, (3.1) $\geq$ (3.2). It will be helpful to introduce some new notation. For a finite process $\mathbf{x}$ and an infinite process $\mathbf{y}$, write $\mathbf{x} :: \mathbf{y}$ for the concatenation of the two. And recall that we may replace $\mathcal{X}_C$ in (3.1) with $\mathcal{X}_\mu$ for any μ greater than the mean arrival rate and less than C, because by Theorem 2.6 the sample path rate function is infinite on $\mathcal{X}_C \setminus \mathcal{X}_\mu$.

Suppose that (3.1) is finite (otherwise the inequality is trivial). The sample path rate function $\mathbf{I}$ is good, so an optimal path $\hat{\mathbf{x}}$ is attained. Now $Q(\hat{\mathbf{x}}) = \sup_t \hat{x}(0,t] - Ct = b$, and this supremum must be attained since otherwise there is a sequence t_n for which $\hat{\mathbf{x}}(0,t_n]/t_n \to C$, which cannot happen in $\mathcal{X}_\mu$. So $\hat{\mathbf{x}} = \hat{\mathbf{x}}(0,\hat{t}] :: \hat{\mathbf{y}}$ for some $\hat{\mathbf{y}}$, with $\hat{x}(0,\hat{t}] = b + C\hat{t}$ and $Q(\hat{\mathbf{y}}) = 0$. Clearly $\mathbf{\Lambda}^*_t(\mathbf{x}(0,t])$ is increasing in t for any $\mathbf{x}$, so

$$\mathbf{I}(\hat{\mathbf{x}}) = \sup_s \mathbf{\Lambda}^*_{\hat{t}+s}(\hat{\mathbf{x}} :: \hat{\mathbf{y}}(0,s]) \geq \mathbf{\Lambda}^*_{\hat{t}}(\hat{\mathbf{x}}(0,\hat{t}]).$$

Taking the infimum over t and $\mathbf{x}(0,t]$ gives the result.

Finally, (3.1) $\leq$ (3.2). Assume that (3.2) is finite (since otherwise the inequality is trivial). For a given t, an optimal $\hat{\mathbf{x}}(0,\hat{t}]$ is attained by goodness of the rate function $\mathbf{\Lambda}^*_t$. And an optimal $\hat{t}$ is also attained. For suppose not, and take a sequence $t_n \to \infty$ and $\mathbf{x}^n(0,t_n]$ with $x^n(0,t_n]/t_n \to C$ and $\mathbf{\Lambda}^*_{t_n}(\mathbf{x}^n)$ bounded above by K say. By the contraction principle and the goodness of the rate function $\mathbf{I}$, we can extend $\mathbf{x}^n(0,t_n]$ to $\mathbf{x}^n(0,\infty)$, with $\mathbf{I}(\mathbf{x}^n) < K$. Since $\mathbf{I}$ is good it has compact level sets, so the $\mathbf{x}^n$ have a convergent subsequence, say $\mathbf{x}^k \to \mathbf{x}$, also with $\mathbf{I}(\mathbf{x}) < K$. But then $x(0,t_k]/t_k \to C$ also, and so $\mathbf{I}(\mathbf{x}) = \infty$, giving a contradiction.

By the contraction principle and the goodness of the rate function, we can extend $\hat{\mathbf{x}}(0,\hat{t}]$ to $\hat{\mathbf{x}} = \hat{\mathbf{x}}(0,\infty)$, where $\mathbf{I}(\hat{\mathbf{x}}(0,\hat{t}]) = \mathbf{I}(\hat{\mathbf{x}})$. If $Q(\hat{\mathbf{x}}) = b$ the inequality is proved. So suppose $Q(\hat{\mathbf{x}}) = b' > b$. Then there is some $s > \hat{t}$ with $\hat{x}(0,s] = b'$. But then

$$\inf_t \inf_{\mathbf{x}:x(0,t]=b+Ct} \mathbf{\Lambda}^*_t(\mathbf{x}) \geq \inf_{s>t} \inf_{\mathbf{x}:x(0,s]=b'+Cs} \mathbf{\Lambda}^*_s(\mathbf{x}) \geq \inf_{s>t} \inf_{\mathbf{x}:x(0,s]=b+Cs} \mathbf{\Lambda}^*_s(\mathbf{x}),$$

where the last inequality is because for fixed t, (3.3) is increasing in b. The inequalities must then both be equalities. We can repeatedly apply this argument until we find an optimal $\hat{\mathbf{x}}$ such that $Q(\hat{\mathbf{x}}) = b$. For otherwise, as in the previous paragraph, there are arbitrarily large optimal $\hat{t}$, leading to a contradiction. □

Proof of Theorem 3.4. First, we prove that $\bar{I}(b) = I(b)$. If $I(b)$ is infinite then $\bar{I}(b)$ must certainly be infinite, as any path which makes $\bar{Q}(\mathbf{x}) = b$ makes $Q(\mathbf{x}) \geq b$. So suppose $I(b)$ is finite, and let the optimizing path in Theorem 3.3 be $\hat{\mathbf{x}}(0,\hat{t}]$. We may assume that this path never causes the buffer to exceed level

b. For suppose that under $\hat{\mathbf{x}}$ the buffer reaches level $b' > b$ at time $-s$. Consider the truncated process $\tilde{\mathbf{x}}(0,s] = \mathbf{x}(\hat{t}-s, \hat{t}]$. By stationarity, $\mathbf{\Lambda}_{\hat{t}}^*(\hat{\mathbf{x}}) \geq \mathbf{\Lambda}_s^*(\tilde{\mathbf{x}})$. And

$$\mathbf{\Lambda}_s^*(\tilde{\mathbf{x}}) \geq \inf_{\mathbf{x}\in\mathbb{R}^s : x(0,s]=b'+cs} \Lambda_s^*(\mathbf{x}) \geq \inf_{\mathbf{x}\in\mathbb{R}^s : x(0,s]=b+cs} \Lambda_s^*(\mathbf{x}),$$

where the second inequality follows because (3.3) is increasing in b. Because the optimal path does not cause the buffer to exceed level b, it is also optimal for the finite buffer case; and so $\bar{I}(b) = I(b)$.

Now fix t and suppose that $\hat{\theta}$ is optimal in (3.3). By Condition 1, $\mathbf{\Lambda}_t$ must be differentiable at $\hat{\theta}\mathbf{1}$. Set $\hat{\mathbf{x}} = \nabla\mathbf{\Lambda}_t(\hat{\theta}\mathbf{1})$. Differentiating (3.3) gives $\hat{\mathbf{x}}\cdot\mathbf{1} = b + Ct$. But by Dembo and Zeitouni [15] Lemma 2.3.9, $\mathbf{\Lambda}_t^*(\hat{\mathbf{x}})$ is equal to (3.3), and so $\hat{\mathbf{x}}$ is optimal. □

Proof of Corollary 3.5. Let F be the event that $Q > 0$. For the large deviations lower bound we will prove that $\inf_{\mathbf{x}\in F}\mathbf{I}(\mathbf{x}) = \lim_{b\downarrow 0} I(b)$, and for the large deviations upper bound,

$$\inf_{\mathbf{x}\in\bar{F}}\mathbf{I}(\mathbf{x}) = \inf_{t>0}\inf_{\mathbf{x}:x(0,t]=Ct}\mathbf{I}(\mathbf{x}). \tag{3.4}$$

This reduces to

$$\inf_{t>0}\sup_{\theta}\theta Ct - \mathbf{\Lambda}_t(\theta\mathbf{1})$$

as in Theorem 3.3. By convexity, $\mathbf{\Lambda}_t(\theta\mathbf{1}) \leq \mathbf{\Lambda}_1(\theta\mathbf{1})$, so the optimum is attained at $t = 1$ and we are left with $I^+(0)$.

Since $F = \cup_{b>0}\{Q = b\}$, $\inf_{\mathbf{x}\in F}\mathbf{I}(\mathbf{x}) = \inf_{b>0} I(b)$. But because $I(b)$ is increasing, this is $\lim_{b\downarrow 0} I(b)$.

LHS ≤ RHS in (3.4). Suppose $x(0,t] = Ct$ for some $t > 0$. For $\varepsilon > 0$, let $\mathbf{x}^\varepsilon = (x_1 + \varepsilon, x_2, \dots)$. Then $Q(\mathbf{x}^\varepsilon) > 0$ so $\mathbf{x}^\varepsilon \in F$. But as $\varepsilon \to 0$, $\mathbf{x}^\varepsilon \to \mathbf{x}$, so $\mathbf{x} \in \bar{F}$. Thus $\{\mathbf{x} : \exists t > 0, x(0,t] = Ct\} \subset \bar{F}$. Taking the infimum of $\mathbf{I}$ over these sets gives the result.

LHS ≥ RHS in (3.4). Let $\mathbf{x} \in \bar{F}$. Then there exist $\mathbf{x}^n \to \mathbf{x}$ in F, and $Q(\mathbf{x}^n) \to Q(\mathbf{x})$ by Lemma 3.7. If $Q(\mathbf{x}) > 0$ then

$$\mathbf{I}(\mathbf{x}) \geq \inf_{b>0} I(b) \geq \inf_{t>0}\sup_{\theta}\theta Ct - \mathbf{\Lambda}_t(\theta\mathbf{1})$$

because the optimal $\hat{t}$ in (3.3) must be strictly positive for $b > 0$.

So suppose $Q(\mathbf{x}^n) \to 0$. As in Lemma 3.7, there exist an n_0 and t_0 such that for $n \geq n_0$,

$$Q(\mathbf{x}^n) = \sup_{t\leq t_0} x^n(0,t] - Ct.$$

And because $Q(\mathbf{x}^n) > 0$, the supremum must be attained at $t > 0$. Some t must be repeated infinitely often as $n \to \infty$; for that t, $x(0,t] = Ct$. Taking the infimum over such $\mathbf{x}$ gives the result.

Now for $\{\bar{Q} > 0\}$. If $\bar{Q}(\mathbf{x}) > 0$ then $Q(\mathbf{x}) > 0$ also, so the same upper bound works. And as for $Q > 0$, the lower bound is straightforward. □

3.4 Paths to Overflow

The expression for the rate function in Corollary 3.1 tells us more than just the probability that the queue size reaches a certain level: it tells us *how* the queue reaches that level. Because the rate function $\mathbf{I}$ is good, the infimum in

$$I(b) = \inf_{\mathbf{x} \in C : Q(\mathbf{x}) = b} \mathbf{I}(\mathbf{x})$$

is attained. And Theorems 3.3 and 3.4 tell us what that sample path looks like: $\hat{\mathbf{x}}$ is the path most likely to make the queue fill from empty to level b, and it takes time $\hat{t}$ to do so. Furthermore, the sample path LDP tells us the likelihood of any deviation from this path.

The problem of most likely paths to overflow under the many sources asymptotic has been studied before using direct methods. Weiss [57] solves it for two-state Markov-modulated fluid sources, and Mandjes and Ridder [40] solve it for general Markov sources and for periodic sources. The advantage of our sample path LDP method is that it can be applied very easily to general input processes.

Example 3.8 (Markov-modulated fluid source)
Let $\mathbf{X}^L$ be the average of L independent sources distributed like $\mathbf{X}$, where $\mathbf{X}$ is a Markov chain which produces an amount of work h each timestep while in the on state and no work while in the off state, and which flips from on to off with probability p and from off to on with probability q. If θ and t are the critical space and time scales, then the most likely path to overflow is given by

$$x_s = \nabla \mathbf{\Lambda}_t(\theta \mathbf{1}) = \frac{\mathbb{E}(X_s e^{\theta X(0,t]})}{\mathbb{E}(e^{\theta X(0,t]})}.$$

We may compute $\mathbb{E}(e^{\theta X(0,t]} | X_0)$ by conditioning on X_1. By reversibility, this is equal to $\mathbb{E}(e^{\theta X[-t,0)} | X_0)$, and by stationarity it is equal to $\mathbb{E}(e^{\theta X[0,t) | X_t})$. This lets us compute $\mathbb{E}(X_s e^{\theta X(0,t]} | X_s)$ and hence x_s. For $s \in (0, t]$ the solution is

$$x_s = \frac{qhe^{\theta h} A_{t-s} A_{s-1}}{qA_t + pB_t}$$

where

$$\begin{pmatrix} A_t \\ B_t \end{pmatrix} = \begin{pmatrix} (1-p)e^{\theta h} & p \\ qe^{\theta h} & 1-q \end{pmatrix}^t \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

If $p+q < 1$ the path to overflow $s \mapsto x_s$ is concave over $s \in (0, t]$: the sources start slowly, then conspire to produce lots of work in the middle of the critical timeperiod, then slow down again at the end. (If $p + q > 1$ it is convex.) An example is illustrated in Figure 3.1. The parameters of the process are $p = 0.4$, $q = 0.2$, $h = 2$. The service rate for the queue is $C = 1$ and the buffer size is $B = 1.3$, giving critical spacescale $\theta = 0.282$ and critical timescale $t = 6$.

Multistate Markov models exhibit more varied behaviour. $\diamond$

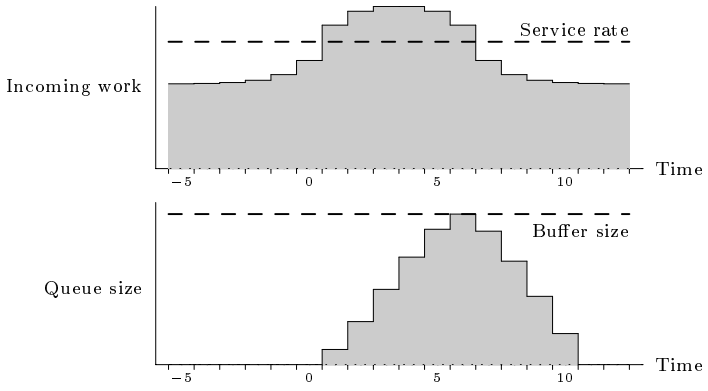

Figure 3.1: A sample path to overflow. The source is a Markov-modulated on/off source, as described in Example 3.8. The most likely path to overflow is convex: the sources start slowly, then conspire to produce lots of work in the middle of the critical timeperiod, then slow down again at the end.

Example 3.9 (Gaussian sources)
Suppose $\mathbf{X}^L$ is the average of L independent Gaussian processes, each with mean λ and covariance structure $\mathrm{Cov}(X_0, X_i) = \gamma_i$. It is easy to work out the optimal path: $\nabla \mathbf{\Lambda}_t(\theta \mathbf{1}) = \lambda \mathbf{1} + \theta V \mathbf{1}$, where $V_{ij} = \gamma_{|i-j|}$.

Consider the earlier fractional Brownian motion example, Example 2.4. For this process, $\gamma_i = \frac{1}{2}\sigma^2\big((i-1)^{2H} - 2i^{2H} + (i+1)^{2H}\big)$, and so the most likely path to overflow is given by

$$x_i = \lambda + \tfrac{1}{2}\theta\sigma^2\Big(i^{2H} - (i-1)^{2H} + (t-i+1)^{2H} - (t-i)^{2H}\Big).$$

If $H > \frac{1}{2}$, the source exhibits long-range dependence, and the most likely input path $\mathbf{x}$ leading to overflow is concave; whereas if $H < \frac{1}{2}$, the path to overflow is convex.

Now let $\mathbf{X}$ be a single-step autoregressive process: $X_t = \lambda + a(X_{t-1} - \lambda) + (1-a^2)\varepsilon_t$, where $\varepsilon_t \sim \mathrm{N}(0, \sigma^2)$ and $|a| < 1$. Then $\gamma_t = \sigma^2 a^t$, and the most likely path to overflow is

$$x_i = \lambda + \theta\sigma^2\left(1 + \frac{1-a^i}{1-a} + \frac{1-a^{t-i+1}}{1-a}\right).$$

If $a > 0$ then path to overflow is concave; whereas if $a < 0$, it starts and finishes at a high rate and in between it oscillates. An example is illustrated in Figure 3.2. The parameters of the process are $\lambda = 0.7$, $a = -0.5$, and $\sigma^2 = 1$. The service rate for the queue is $C = 0.8$ and the buffer size is $B = 0.9$, giving critical spacescale $\theta = 0.575$ and critical timescale $t = 7$. $\diamond$

Example 3.10 (Large Buffer)
By contrast, in the large buffer asymptotic it is often the case that the process $\mathbf{X}$ leads to a limiting generating function $\mathbf{\Lambda}_t$ with the simple linear form $\mathbf{\Lambda}_t(\boldsymbol{\theta}) = \sum \mathbf{\Lambda}_1(\theta_i)$. (See Dembo and Zajic [14] for conditions under which this is so.) Then, because $\mathbf{\Lambda}_1$ is convex, the most likely path $\mathbf{x}$ to overflow is constant, and so the queue fills up at a steady rate. $\diamond$

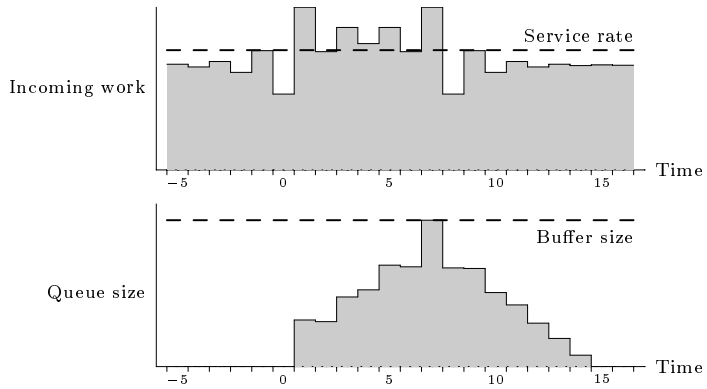

Figure 3.2: A sample path to overflow. The source is a first-order autoregressive process with negative correlation coefficient, as described in Example 3.9. This means that the most likely sample path to overflow is oscillatory and the queue fills up in an irregular fashion, over the critical time period $(0, 7]$.

3.5 Priority Queues

The sample path LDP for the average of processes can be applied to a wide variety of queueing models. We have seen in the last two sections how it gives overflow probabilities and sample paths to overflow for a standard queue. As a further illustration of the power of the technique, in this section we look at another queueing discipline: the priority queue. This has been studied under the large buffer regime by Berger and Whitt [2], and related queueing models have been studied by Kulkarni et al. [32] and O'Connell [45].

Consider a sequence of priority queues, indexed by L. The Lth queue has two inputs, $L\mathbf{X}^L$ and $L\mathbf{Y}^L$, and service rate LC. Think of $\mathbf{X}^L$ and $\mathbf{Y}^L$ as the averages of L processes. The two flows are assumed to be independent. The first flow $\mathbf{X}^L$ has high priority; the second flow $\mathbf{Y}^L$ has low priority. Let Q^L and R^L be respectively the stationary amounts of high-priority and low-priority work waiting to be served.

Kelly [29] notes that the amount of high-priority traffic Q is exactly the amount of work in a standard queue with service rate C and only the high-priority input $\mathbf{X}$, and that the total amount of work $Q + R$ is the amount of work in a standard queue with service rate C and the aggregate input $\mathbf{X} + \mathbf{Y}$. Therefore, results from Section 3.3 can be applied directly to find the high-priority loss probability and the aggregate loss probability. But this leaves some open questions, such as how much low-priority work there is in the queue. Such questions can be answered with methods very similar to those of Section 3.3.

Theorem 3.8 *Suppose that $\mathbf{X}^L$ and $\mathbf{Y}^L$ satisfy the conditions of Theorem 2.7 with limiting moment generating functions $\mathbf{\Lambda}_t$ and $\mathbf{M}_t$ respectively. Suppose also that the sum of the mean arrival rates for $\mathbf{X}^L$ and $\mathbf{Y}^L$ is strictly less than*

C. Then the pair (Q^L, R^L) satisfies an LDP with good rate function

$$I(q,r) = \inf_{\substack{\mathbf{x}\in\mathcal{X}_C,\mathbf{y}\in\mathcal{X}_C:\\ Q(\mathbf{x})=q,R(\mathbf{x},\mathbf{y})=r}} \sup_t \mathbf{\Lambda}_t^*(\mathbf{x}(0,t]) + \sup_t \mathbf{M}_t^*(\mathbf{y}(0,t]). \tag{3.5}$$

This is bounded below by

$$\inf_t \inf_{s\le t} \sup_{\theta,\phi} \theta(q+Cs) + \phi(r+C(t-s)) - \mathbf{\Lambda}_t(\theta\mathbf{1}(0,s] + \phi\mathbf{1}(s,t]) - \mathbf{M}_t(\phi\mathbf{1}). \tag{3.6}$$

Let $I(\cdot,r) = \inf_{q\ge 0} I(q,r)$. This is bounded below by

$$\inf_t \sup_\theta \theta(r+Ct) - \mathbf{\Lambda}_t(\theta\mathbf{1}) - \mathbf{M}_t(\theta\mathbf{1}). \tag{3.7}$$

Proof. Let $\mathbf{I}_X(\mathbf{x}) = \sup_t \mathbf{\Lambda}_t^*(\mathbf{x})$, and define $\mathbf{I}_Y$ similarly. Because $\mathbf{X}^L$ and $\mathbf{Y}^L$ are independent, the pair $(\mathbf{X}^L, \mathbf{Y}^L)$ satisfies an LDP with good rate function $\mathbf{I}(\mathbf{x},\mathbf{y}) = \mathbf{I}_X(\mathbf{x}) + \mathbf{I}_Y(\mathbf{y})$. Let λ and μ be the mean rates for $\mathbf{X}^L$ and $\mathbf{Y}^L$. Since $\lambda+\mu<C$, we can pick an $\varepsilon>0$ such that $\lambda+\mu+2\varepsilon<C$: then by Theorem 2.6, $(\mathbf{X}^L,\mathbf{Y}^L)$ satisfies the LDP on $(\mathcal{X}_{\lambda+\varepsilon},\mathcal{X}_{\mu+\varepsilon})$, and the rate function $\mathbf{I}$ is infinite outside this space. So if we can show that (Q,R) is continuous on this space, then using the Contraction Principle we can deduce the LDP for the priority queue.

Now Q depends only on the high priority process: it is defined as though there were no other inputs to the queue. So by Lemma 3.7, it is continuous on $\mathcal{X}_{\lambda+\varepsilon}$. Also, $Q+R$ is the aggregate workload, and does not depend on the structure of the queue: so again by Lemma 3.7, $Q+R$ is continuous on $\mathcal{X}_{\lambda+\varepsilon}\times\mathcal{X}_{\mu+\varepsilon}$. Thus (Q,R) is continuous.

The bound on the rate function $I(q,r)$ may be obtained by noting a few properties of the optimal paths to overflow. If $I(q,r)$ is finite the optimal paths must be attained, because the rate function is good. As in Theorem 3.3, there must be a last time $-t$ at which the high priority and low priority queues are both empty. And there must be a last time $-s\ge -t$ at which the high priority queue is empty. Because $Q(\mathbf{x})=q$, it must be that $x(0,s]=q+Cs$. And because $R(\mathbf{x},\mathbf{y})=r$, it must be that $x(s,t]+y(0,t]=r+C(t-s)$. So

$$I(q,r) \ge \inf_t \inf_{s\le t} \inf_{\substack{\mathbf{x},\mathbf{y}\in\mathbb{R}^t:\\ x(0,s]=q+Cs,\\ x(s,t]+y(0,t]=r+C(t-s)}} \mathbf{\Lambda}_t^*(\mathbf{x}) + \mathbf{M}_t(\mathbf{y}). \tag{3.8}$$

Now fix s and t. As in Theorem 3.3, the pair $(X^L(0,s], X^L(s,t]+Y^L(0,t])$ is just an $\mathbb{R}^2$-valued random variable, and by Assumption 1 it satisfies an LDP with a good rate function which simplifies to the expression in (3.6). Another way of finding this LDP is by contracting from the sample path LDP for $(\mathbf{X}^L(0,t],\mathbf{Y}^L(0,t])$ which gives as rate function the expression in (3.8). By the uniqueness of the rate function, these are equal.

We can obtain the lower bound on $I(\cdot,r)$ in a similar way, by noting that if $R(\mathbf{x},\mathbf{y})=r$ then there exists a last time $-t$ at which both queues were empty, and since then $x(0,t]+y(0,t]\ge r+Ct$. The argument of the previous paragraph can be applied to paths for which $x(0,t]+y(0,t]=q+r+Ct$. The resulting expression is increasing in q (it is a special case of (3.3) which is increasing in

b), and setting $q = 0$ yields the result. □

To help interpret this result, we will give an alternative description in terms of the service seen by the low priority flow. A sensible first guess would be that the service is a random amount, the service rate C less a random amount of high priority work. More thought would throw up various complications about queue sizes and leftover workloads. In fact, it turns out that in some cases the first guess is correct and in other cases these complications do arise, and a system can switch from one regime to the other as its parameters change. We will give an example to illustrate this transition.

In making precise the idea of the service seen by the low priority flow, we will use the theory of effective bandwidths. The effective bandwidth $\lambda(\theta, t)$ of a flow is a measure of the impact it has at a queue. It lies between the mean and peak bandwidths, and is defined by

$$\lambda(\theta, t) = \frac{1}{\theta t}\mathbf{\Lambda}_t(\theta \mathbf{1}).$$

Effective bandwidths are described more fully in Section 3.6, where we show the following: if a queue is fed by many input flows of effective bandwidth $\lambda(\theta, t)$ and has critical point $(\hat{\theta}, \hat{t})$, then replacing a small number of the input flows by flows of constant rate $\lambda(\hat{\theta}, \hat{t})$ does not affect the loss probability.

Effective bandwidths can also describe the service seen by the low priority flow. Consider a single queue fed by a process with effective bandwidth $\mu(\theta, t)$, but where the service is an independent stochastic process $\tilde{\mathbf{C}}$ with effective bandwidth $\tilde{C}(\theta, t)$. As above, if the critical space and time scales are $\hat{\theta}$ and $\hat{t}$, replacing a small part of the service with constant service of rate $\tilde{C}(\hat{\theta}, \hat{t})$ does not affect the operation of the queue, and so we will call $\tilde{C}(\theta, t)$ the *effective service rate*. Before we use this idea to describe the service seen by the low priority flow, we had better check that it actually exists: that is, that the appropriate cumulant moment generating functions converge.

Lemma 3.9 (Effective Service) *Under the assumptions of Theorem 3.8, the service seen by the low priority queue has an effective service rate.*

Proof. O'Connell [44] shows that the departure map (which maps the aggregate input process to the aggregate departure process) is continuous under the uniform topology. Let $\mathbf{d}$ be the departure process from the high priority queue. The service seen by the low priority queue is $\tilde{\mathbf{C}}$ where $\tilde{C}_t = C - d_t$. Since the departure map is continuous, the service map is also continuous. Therefore the service process satisfies a large deviations principle, say with good rate function $\mathbf{J}$.

Applying Varadhan's Integral Lemma (Dembo and Zeitouni [15] Theorem 4.3.1), and using the fact that the service process is bounded, we find that

$$\lim_{L\to\infty} \frac{1}{L} \log \mathbb{E} e^{L\boldsymbol{\theta}\cdot\tilde{\mathbf{C}}(0,t]} = \sup_{\mathbf{c}\in\mathbb{R}^t} \boldsymbol{\theta}\cdot\mathbf{c} - \mathbf{J}(\mathbf{c}).$$

In particular, the limit exists. □

We are now in a position to make precise the earlier claim about the service seen by the low priority queue. The effective service rate is difficult to deal with analytically, but fortunately we can avoid doing so by using Theorem 3.8. The following corollary is a restatement of the bound (3.7). The terminology is due to Berger and Whitt [2], who independently obtained the corresponding result for the large buffer asymptotic regime. As noted in Example 2.3, large buffer results can be deduced from a special case of the corresponding many sources results.

Corollary 3.10 (Empty Buffer Approximation) *The effective service rate seen by the low priority queue is bounded below by the* empty buffer approximation *to the service rate, $\tilde{C}(\theta,t) = C - \lambda(\theta,t)$, in the following sense:*

$$I(\cdot,r) \geq EI(r) = \inf_t \sup_\theta \theta(r + t\tilde{C}(\theta,t)) - \theta t\mu(\theta,t),$$

where $\mu(\theta,t)$ is the effective bandwidth of the low priority source.

This is just the usual rate function (3.3) for overflow in a single queue, but with the service rate C replaced by the effective service rate $\tilde{C}$. It is called the *empty buffer approximation* because it is the rate function for the event that total workload reaches r—so if the most likely way for this to happen leaves the high priority buffer empty, then $EI(r)$ will agree with $I(\cdot,r)$.

Berger and Whitt stress the point that this approximation gives a simple admission control region. It is also interesting to consider the conditions under which the inequality is strict. When there is equality, the two queues operate essentially independently. But when the inequality is strict, the low priority queue gets extra benefit from the sharing arrangement. Such an arrangement seems desirable from an engineering perspective. The following example illustrates how the queue and traffic parameters control whether or not there is extra benefit to the low priority traffic.

Example 3.11 (Phase transition in priority queues)
It is often hard to simplify rate functions like $I(q,r)$ because the queue could overflow over any timescale. But for periodic processes, the queue can only overflow over timescales less than the period, so the calculations are easier.

Consider a sequence of priority queues indexed by L. Let the high priority flow $\mathbf{X}^L$ be the average of L independent copies of a stationary periodic process of random phase, which produces 4 units of work every second timestep. Let the low priority flow $\mathbf{Y}^L$ be the average of L independent copies of the process that independently at each timestep produces 1 unit of work with probability p and no work with probability $1-p$. Let the service rate C be in the range $[3,4)$.

These figures are chosen so that the entire queue empties every second timestep, so that if it overflows it must do so in a single timestep. This means that the only sample paths we need consider in (3.5) are those over a single timestep. So

$$\begin{aligned} I(0,r) &= \inf_{0\leq x\leq C} \mathbf{\Lambda}_1^*(x) + \mathbf{M}_1^*(r+C-x) \\ I(q,r) &= \mathbf{\Lambda}_1^*(q+C) + \mathbf{M}_1^*(r) \quad (\text{for } q>0). \end{aligned}$$

Since $q + C$ is greater than the mean rate of $\mathbf{\Lambda}$, $\mathbf{\Lambda}_1^*(q + C) \geq \mathbf{\Lambda}_1^*(C)$, and so $I(\cdot, r) = I(0, r)$. Now for the empty buffer approximation. Since $EI(r)$ is the rate function of the sample path most likely to give total queue size r,

$$EI(r) = \inf_{0 \leq x \leq C+r} \mathbf{\Lambda}_1^*(x) + \mathbf{M}_1^*(r + C - x).$$

Clearly $I(\cdot, r) \geq EI(r)$. When is this inequality strict? Let $g(x) = \mathbf{\Lambda}_1^*(x) + \mathbf{M}_1^*(r + C - x)$. It is easy to calculate that for $r < 1$,

$$g(x) = h(x/4 \,|1/2) + h(r + C - x \mid p),$$

where $h(x|p) = x \log(x/p) + (1 - x) \log(1 - x)/(1 - p)$, and to show that $g(x)$ is convex. So $I(\cdot, r) > EI(r)$ if and only if $g'(C) < 0$, where

$$g'(C) = \frac{1}{4} \log \frac{C}{4 - C} - \log \frac{r}{1 - r} + \log \frac{p}{1 - p}.$$

In other words, there is extra benefit to the low priority traffic when the service rate is small, or when the low priority buffer is large, or when there is little low priority work. $\diamond$

3.6 Effective Bandwidths

In this section we will not prove any new results about queues. Instead, we will express the results of the earlier sections in a different way. The effective bandwidth of a flow is a convenient and intuitive description of its impact on a queue.

Kelly [29] gives a comprehensive survey of effective bandwidth results. Here we do not attempt to be comprehensive. Rather we extend the definition of effective bandwidth and suggest a new way of looking at it. This lets us explain various results in the following chapters more conveniently.

As usual, let $\mathbf{X}$ be a real-valued random process indexed by the natural numbers. For $t \geq 1$ and $\boldsymbol{\theta} \in \mathbb{R}^t$, define the *effective bandwidth* of $\mathbf{X}$ at $\boldsymbol{\theta}$ to be

$$\alpha_X(\boldsymbol{\theta}) = \frac{1}{\boldsymbol{\theta} \cdot \mathbf{1}(0, t]} \log \mathbb{E} \exp\left(\boldsymbol{\theta} \cdot \mathbf{X}(0, t]\right).$$

It is trivially true that all queueing behaviour depends on the effective bandwidths of the input flows, because the effective bandwidth encodes the entire distribution. What is less obvious is that often only a small part of the distribution matters.

First we will explain how effective bandwidths arise in admission control at queues, and why they are so called. Then we will describe other circumstances in which they are useful. Finally we discuss their use as approximations.

Effective bandwidths and loss probability

Consider first a standard queue with service rate C and buffer size B, as described in Section 3.3, fed by input process $\mathbf{X}$. From Corollary 3.6 we know that the rate function for overflow is

$$I = \inf_t \sup_\theta \theta(B + Ct) - \theta t \alpha_X\left(\theta \mathbf{1}(0, t]\right).$$

(This is shorthand for the following: Consider a sequence of queues indexed by L, where the Lth queue has service rate LC and buffer size LB, and is fed by $L\mathbf{X}^L$, where $\mathbf{X}^L$ satisfies the conditions of Theorem 2.7. Let α_X be the effective bandwidth function arising from the limiting moment generating function $\mathbf{\Lambda}$. Then the large deviations upper bound for the event that the queue overflows is $-I$. We will often use this shorthand.)

Consider replacing a small proportion δ of the L input flows by flows which produce work at a constant rate a; these have effective bandwidth a. The rate function for overflowing is now

$$I(\delta) = \inf_t \sup_\theta \theta(B + Ct) - \theta t\Big((1-\delta)\alpha_X\big(\theta \mathbf{1}(0,t]\big) + \delta a\Big). \tag{3.9}$$

If the optimizing parameters in I are $\hat{\theta}$ and $\hat{t}$, and under appropriate differentiability conditions, the value of a that makes $I'(0) = 0$ is $a = \alpha_X(\hat{\boldsymbol{\theta}})$ where $\hat{\boldsymbol{\theta}} = \hat{\theta}\mathbf{1}(0,\hat{t}]$. In other words, an input flow has the same effect on the queue as would a constant flow of rate $\alpha_X(\hat{\boldsymbol{\theta}})$. This is why α is called the effective bandwidth function. The value $\hat{\boldsymbol{\theta}}$ is called the *operating* or *critical point* of the queue.

The standard definition of effective bandwidth is $\alpha_X(\theta, t) = \alpha_X(\theta \mathbf{1}(0,t])$. This is because the operating point for overflow is always of the special form $\hat{\boldsymbol{\theta}} = \hat{\theta}\mathbf{1}(0,\hat{t}]$.

If there are multiple input flows of different types, then the effective bandwidth function measures the tradeoff between different types. For example, if at the operating point the effective bandwidth of a flow of type A is twice that of a flow of type B, then replacing a small number of flows of type A by twice that number of flows of type B will not affect the probability of overflow.

Effective bandwidths for other purposes

Suppose we are interested not in the event that the queue overflows but in some general event E. The large deviations rate for this event is

$$I(E) = \inf_{\mathbf{x}\in E} \mathbf{I}(\mathbf{x}) = \inf_{\mathbf{x}\in E} \sup_t \sup_{\boldsymbol{\theta}\in\mathbb{R}^t} \boldsymbol{\theta}\cdot\mathbf{x}(0,t] - \boldsymbol{\theta}\cdot\mathbf{1}(0,t]\alpha_X(\boldsymbol{\theta}).$$

All the events we are interested in in this thesis have the form $E = \bigcup_t E_t$ where E_t is of the form $\mathbf{x}\cdot\mathbf{w}_t = a_t$ for some $\mathbf{w}_t \in \mathbb{R}^t$. For example, the event *the queue overflows* can be written *there exists a time t such that $x(0,t] > B + Ct$*. In such cases, the rate function is

$$\begin{aligned} I(E) &= \inf_t \inf_{\mathbf{x}\in E_t} \sup_{\boldsymbol{\theta}\in\mathbb{R}^t} \boldsymbol{\theta}\cdot\mathbf{x} - \boldsymbol{\theta}\cdot\mathbf{1}(0,t]\alpha_X(\boldsymbol{\theta}) \\ &= \inf_t \sup_\theta \theta a_t - \theta\mathbf{w}_t\cdot\mathbf{1}(0,t]\alpha_X(\theta\mathbf{w}_t). \end{aligned}$$

(The proof of the second equality is just the same as that of (3.2) = (3.3) in Theorem 3.3.)

If the event E is closed and $I(E)$ is finite, as will typically be the case, then since $\mathbf{I}$ is a good rate function the optimal path $\hat{\mathbf{x}}$ will be attained and so will the optimal $\hat{t}$. Usually the optimal $\hat{\boldsymbol{\theta}}$ is attained too.

When the critical point $\hat{\boldsymbol{\theta}}$ is attained, as in Theorem 3.3 the optimal path $\hat{\mathbf{x}}$ is given by

$$\hat{\mathbf{x}}(0,t] = \nabla\boldsymbol{\theta} \cdot \mathbf{1}(0,t]\alpha_X(\boldsymbol{\theta}), \tag{3.10}$$

where the derivative is taken at $\theta_s = \hat{\theta}_s$ for $0 < s \leq \hat{t}$ and $\theta_s = 0$ otherwise.

We will use these ideas extensively in Chapter 5, to analyse the behaviour of various queue algorithms. The event E will be *some mechanism is triggered* and (3.10) will tell us the most likely way for it to be triggered.

Effective bandwidth as an approximation

We have described how effective bandwidths can be interpreted formally, in a limiting regime in which the number of independent identically distributed input flows increases and the service rate and buffer size increase in proportion. But the intention is that they should be thought of as approximating finite systems. For example, if a queue has buffer size B and service rate C and is fed by input flows $\mathbf{X}(1) \cdots \mathbf{X}(n)$, then we approximate

$$-\log \mathbb{P}(\text{overflow}) \approx \inf_t \sup_\theta \theta(B + Ct) - \theta t \sum_{i=1}^{n} \alpha_{X(i)}(\theta, t).$$

This approximation should be good when C and n are large. (To measure how good, one can find finer approximations [34] or perform simulations [4, 10, 11].)

What the effective bandwidth approximation does is pick out the most important part of the distribution, the critical point, and make a zero-order approximation to the flow at the critical point, $\alpha(\hat{\boldsymbol{\theta}} + \boldsymbol{\varepsilon}) = \alpha(\hat{\boldsymbol{\theta}}) + \cdots$, much as one would approximate a real-valued function $f(x)$ by $f(x + \varepsilon) = f(x) + \cdots$. The other terms do matter, but the zero-order term is most important; and in the large deviations limit, it is only the zero-order term that matters.

3.7 Summary

A sample path large deviations principle is an LDP factory: it makes it easy to study large deviations in a wide range of queueing problems. First we recall the queueing model, and then we describe the results that will be used in the rest of this thesis.

Consider a sequence of queues indexed by L. Let the Lth queue have service rate LC and buffer size LB, where $B > 0$, and let it be fed by $L\mathbf{X}^L$, which will typically be the aggregate of L independent copies of some base flow $\mathbf{X}$. Let $\bar{Q}(\mathbf{X}^L)$ be the amount of work in the queue. Assume that $\mathbf{X}^L$ satisfies the conditions for the sample path LDP (Theorem 2.7), and has limiting log moment generating functions $\boldsymbol{\Lambda}_t$. Using the Contraction Principle we can show the following.

Theorem 3.11 (Rate functions for queues) *Suppose that $\mathbf{X}^L$ satisfies the conditions of Theorem 2.7 and is stationary with mean rate strictly less than C. Let*

$$I(b) = \inf_{t \geq 0} \sup_{\theta \in \mathbb{R}} \theta(b + Ct) - \boldsymbol{\Lambda}_t(\theta\mathbf{1}),$$

let $I(b^+) = \lim_{a \downarrow b} I(a)$*, and let* $I^+(0) = \sup_{\theta \in \mathbb{R}} \theta C - \mathbf{\Lambda}_1(\theta \mathbf{1})$.

Then $\bar{Q}(\mathbf{X}^L)$ *satisfies an LDP with good rate function* $I(b)$*. Also, the event* $\{\bar{Q} > 0\}$ *has large deviations lower bound* $-I(0^+)$ *and large deviations upper bound* $-I^+(0)$*, and the event that* $\bar{Q}$ *overflows has large deviations lower bound* $-I(B^+)$ *and large deviations upper bound* $-I(B)$*.*

Theorem 3.12 (Path to overflow) *Under the conditions of the previous theorem it is also the case that*

$$\begin{aligned} I(b) &= \inf_{\mathbf{x} \in \mathcal{X}_C : \bar{Q}(\mathbf{x}) = b} \mathbf{I}(\mathbf{x}) \\ &= \inf_t \inf_{\mathbf{x} \in \mathbb{R}^t : x(0,t] = b + Ct} \sup_{\boldsymbol{\theta} \in \mathbb{R}^t} \boldsymbol{\theta} \cdot \mathbf{x} - \mathbf{\Lambda}_t(\boldsymbol{\theta}). \end{aligned}$$

If $I(b)$ *is finite, the optimal* $\hat{\mathbf{x}}$ *and* $\hat{t}$ *are attained. If the optimal* $\hat{\boldsymbol{\theta}}$ *is attained, it has the form* $\hat{\theta}\mathbf{1}$ *and*

$$\hat{\mathbf{x}}(0,t] = \nabla \mathbf{\Lambda}_t(\boldsymbol{\theta}) \tag{3.11}$$

where the derivative is taken at $\theta_s = \hat{\theta}_s$ *for* $s \in (0,t]$ *and* $\theta_s = 0$ *otherwise.*

The optimal $\hat{\boldsymbol{\theta}}$ (or the optimal $(\hat{\theta}, \hat{t})$ pair) is called the *critical point* of the queue.

These are limiting results, and they are intended to be used as approximations for finite systems. We will talk about a queue with service rate C and buffer size B fed by aggregate input $\mathbf{X}$, and we will approximate

$$\mathbb{P}(\text{overflow}) \approx \exp\Big(- \inf_t \sup_\theta \theta(B + Ct) - \mathbf{\Lambda}_t(\theta \mathbf{1}) \Big).$$

(Often the large deviations upper and lower bounds agree). This approximation should be good when $\mathbf{X}$ is the aggregate of many independent flows.

Chapter 4

Networks

A *router* or *switch* is a device that routes traffic. A router has several input flows of traffic, each of which is routed to a specified destination; and inside the router, work from all the input flows is queued together. Routers are the building blocks of the Internet. It is by describing their behaviour that queueing theory can tell us about telecommunications networks.

The behaviour of isolated queues has been much studied. The preceding chapters have used large deviations to characterize the input traffic, to estimate the probability that the queue overflows, and to study different queueing regimes. In this chapter, we study networks of routers. The fundamental result is that, under the many sources limiting regime, the statistical characteristics of a flow of traffic are not changed by passing through a router.

This result dramatically simplifies the analysis of networks. It means that the techniques for describing isolated queues can be applied inductively to feed-forward networks. It also means that it is useful to talk about the characteristics of a type of traffic, without bothering how many routers the flow has passed through or what other flows it has interacted with.

The theory of large deviations is concerned with limiting regimes, and it is our choice of limiting regime which makes possible such clean results for networks. We study the *many sources* limiting regime, in which the number of independent flows coming into a router increases, and the buffer size per flow and service rate per flow stay fixed. We suppose that of the different flows coming into a router, only a small number stay together when they leave—it is after all the function of a router to route traffic to different destinations.

The rest of this chapter is arranged as follows. In Section 4.2 we describe the network model and set up the notation. In Section 4.3 we prove the fundamental result, that the large deviations characteristics of a flow are not changed by passing through a router. In Section 4.4 we extend this to networks. In Section 4.5 we describe these results in terms of effective bandwidths, and discuss limitations and extensions. First we review related work.

4.1 Related work

Kelly [26] describes queueing networks in which all input traffic flows are Poisson and service times are exponential. These networks admit a very simple solution:

at any instant in time, the different queues are independent, and the distribution of queue sizes can be written down explicitly. Furthermore the distribution of a flow when it leaves the network is the same as when it entered (though inside the network, the distribution may be different). These are best-possible results, and they break down when the input processes are more general. Our results are weaker, in that they only concern limiting regimes, but they do cope with general input processes. And they are complementary in a curious way: we can calculate the distribution of a flow at any point inside the network and show that it is the same as when the flow entered, though we cannot calculate queue size distributions (except in feedforward networks) because traffic flows within the network are not independent.

Traffic limits

The limiting regime we are interested in is the many sources limiting regime, in which the number of independent inputs to a router increases. Another limiting regime—one which has been more widely studied—is the large buffer regime, described in Example 3.2, in which the number of input flows is fixed and the buffer size increases. We call this a *traffic limit*, because it is mathematically equivalent to a limiting regime in which the router is fixed but the traffic is speeded up. In the large buffer regime, clean results for networks are hard to come by.

It is possible to prove a large deviations principle for the aggregate output of a router, simply by noting that the map from aggregate input to aggregate output is continuous and using the contraction principle. It can also be shown that the map from the set of inputs to an individual output is continuous, and in this way O'Connell [42, 45] and Majewski [39], find LDPs under the large buffer limiting regime for the individual output processes. Most of the work in obtaining this sort of result is in defining the queueing model and in proving continuity. The outcome is an LDP with a rate function which is the solution to a complicated variational formula.

It can be seen from this variational formula that the aggregate output is smoother than the aggregate input. It is hard to draw any other general conclusions. For example, when there are several input flows, it sometimes happens that some are made burstier while others are made smoother. This has been investigated further by de Veciana et al. [13]. They found that if the service rate is sufficiently high then the outputs *decouple*. By this they mean that the effective bandwidth of a flow is the same when it leaves the router as when it came in, at least for low values of the spacescale θ. In other words, as long as we are not interested in extreme behaviour, the statistical characteristics of a flow of traffic are not altered by passing through the router. Unfortunately, their arguments only apply to the first router in a network, because the output flows do not satisfy the conditions that would enable their results to be applied inductively. We will prove a much stronger form of decoupling. We will show that the effective bandwidth of the output is the same as that of the input whatever the spacescale or timescale, for any service rate larger than the mean input rate; and that this is true throughout the network; and further that in the limit the different output flows are independent.

There has also been some work on the output of a router under the many sources limiting regime. Duffield and Low [18] give a large deviations principle

for the aggregate output using the contraction principle, just as has been done for the large buffer regime. Because the many sources regime has a richer structure than the large buffer regime, it produces variational formulae that are even more complicated.

A different way of looking at networks, which does not face these problems, is taken by Paschalidis [46] and Bertsimas et al. [3]. They still have complicated optimization formulae for the output processes, but they eliminate the problem of coupling by assuming that any work leaving a router chooses its destination randomly. However, we will assume that each flow is routed to a specific destination.

Network limits

What all these approaches have in common is that they take a fixed network and look at various sorts of traffic limit. We have been able to prove much cleaner results by looking at a different sort of limit, one in which both the traffic and the structure of the network change.

Many different sorts of network limit have been looked at in the past, though mostly this has been to help answer questions about routing rather than about traffic characteristics, and the analysis has mostly involved tools other than large deviations theory. There has nonetheless been some work on how the characteristics of traffic change in various network limits: for example, Mountford and Prabhakar [41] have studied the limiting form of traffic as it passes through more and more queues, and found conditions under which it converges to a fixed point.

In our network limit the number of traffic flows and the number of routers both increase, but along the path of a single flow the number of routers stays fixed. This seems well-suited to the Internet, in which the number of users has increased dramatically but the length of a typical path has not.

All of this work, like our own, deals only with feedforward networks. Feedback raises considerable theoretical challenges.

4.2 The network model

We are still using the queueing model described in Chapter 3. Consider a standard first-in–first-out queue with constant service rate LC and finite buffer size LB, and let any work that arrives when the queue is full be lost. We will be concerned with the behaviour of a queue fed with input process $L\mathbf{X}^L$, where $\mathbf{X}^L$ satisfies the conditions for the sample path large deviations principle, Theorem 2.7.

Suppose that $\mathbf{X}^L$ is the average of L independent identically distributed flows. Let $\mathbf{X}^{(L)}$ be a typical input flow, and let $\tilde{\mathbf{X}}^{(L)}$ be the corresponding output flow. (In later sections we will allow $\mathbf{X}^L$ to be the aggregate of several such averages.) The generating function for the aggregate input is therefore

$$\mathbf{\Lambda}_t^L(\boldsymbol{\theta}) = \log \mathbb{E} \exp (\boldsymbol{\theta} \cdot \mathbf{X}^{(L)}).$$

Similarly, the moment generating function for the aggregate of *independent copies* of a typical output is

$$\tilde{\mathbf{\Lambda}}_t^L(\boldsymbol{\theta}) = \log \mathbb{E} \exp (\boldsymbol{\theta} \cdot \tilde{\mathbf{X}}^{(L)}).$$

These are the mathematical quantities we will be dealing with, and it is worth explaining what they represent. We consider the moment generating function $\tilde{\mathbf{\Lambda}}^L$ because it is the natural way to describe the behaviour of a *single* output from an upstream queue. But large deviations does not tell us about the behaviour of a single output, so in order to understand the following theorems it can be helpful to think of $\tilde{\mathbf{\Lambda}}^L$ as describing the aggregate of independent copies of $\tilde{\mathbf{X}}^{(L)}$. One could think of L different upstream queues each fed by L independent copies of $\mathbf{X}^{(L)}$ and each contributing a single output flow.

Our fundamental result is that if $\mathbf{\Lambda}^L$ satisfies the conditions of the sample path LDP, Theorem 2.7, then $\tilde{\mathbf{\Lambda}}^L$ also satisfies the sample path LDP, with the same rate function. We will discuss the meaning and applications of this result in Sections 4.4 and 4.5. But first we must prove it.

4.3 The output of a router

Suppose that $\mathbf{\Lambda}^L$ satisfies Conditions 1 and 2 of Theorem 2.7, and that $\mathbf{\Lambda}^L \to \mathbf{\Lambda}$. Then the theorem tells us that $\mathbf{X}^L$ satisfies a sample path LDP with a good rate function which can be calculated from $\mathbf{\Lambda}$.

What we would like to show is that that $\tilde{\mathbf{\Lambda}}^L$ satisfies the same conditions and converges to the same limit. If this were true, $\tilde{\mathbf{X}}^L$ would satisfy exactly the same LDP as $\mathbf{X}^L$: in other words, the statistical characteristics of a flow of traffic would be unchanged by passing through the router.

The first condition can be proved. The key idea in its proof is this: that the probability that the queue is empty over a fixed interval tends to one, by Theorem 3.11, and so the probability that the amount of work arriving in that interval is equal to the amount of work leaving in that interval tends to one also. There is not only convergence in probability but also convergence in expectation; this is shown in Theorem 4.1.

We would also like to show that $\tilde{\mathbf{\Lambda}}^L$ satisfies Condition 2, which is a technical condition on the uniformity of convergence. In fact that condition is not satisfied, and we have not been able to establish Theorem 2.7 for the output. This is not actually a problem! A sample path LDP still holds, under a weaker topology which we call the weak queue topology, and as noted in Section 2.3.3 this is sufficient to obtain all the results of Chapter 3. We prove the sample path LDP in Theorem 4.2.

In the same way that in Section 2.3.4 we restricted the sample path LDP to take account of the mean arrival rate, so we do here for the output process, in Theorem 4.3.

Before proving the theorems, we give simulation results to illustrate them. Figure 4.1 shows two cases: in the first a router handles a single traffic flow, and in the second a router with three times the capacity handles three identical and independent flows. In the first case the flow is significantly smoothed by passing through the router, but in the second the smoothing is negligible. The flows illustrated are periodic, sending one unit of work every fourth timestep. The service rate per flow is 0.4 and the buffer size per flow is 1.5. (The figure shows the effective bandwidth of the flow before it enters the router and after it leaves. The effective bandwidth function $\alpha(\theta, t)$ is a convenient representation of the moment generating function: $(\theta t)^{-1}\mathbf{\Lambda}_t(\theta \mathbf{1})$. Effective bandwidths are described more fully in Section 3.6.)

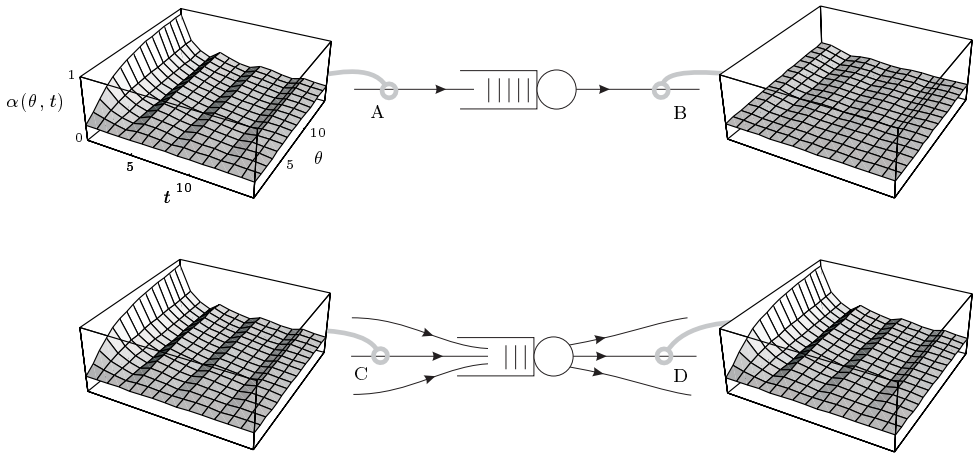

Figure 4.1: The output of a router. A single flow (A) passes through a router and is made smoother (B). But when the router is fed by three identical flows (C) and its service rate and buffer size are increased in proportion, any smoothing is negligible (D). The graphs plot the effective bandwidth functions of the flows, all to the same scale. The effective bandwidth of a flow is a convenient representation of its burstiness over different timescales and spacescales.

Theorem 4.1 (Finite-time characteristics of the output)
If the input $\mathbf{X}^{(L)}$ satisfies Conditions 1 and 2, and is stationary with mean rate strictly less than C, then the output $\tilde{\mathbf{X}}^{(L)}$ satisfies Condition 1, with the same limiting moment generating function as $\mathbf{X}^{(L)}$. In other words,

$$\lim_{L\to\infty} \log \mathbb{E} \exp\left(\boldsymbol{\theta} \cdot \tilde{\mathbf{X}}^{(L)}(0,t]\right) = \boldsymbol{\Lambda}_t(\boldsymbol{\theta}).$$

Proof. First note that $\tilde{X}^{(L)}(0,t] \leq X^{(L)}(0, t + \lfloor B/C \rfloor]$, since any work arriving before $-\lfloor B/C \rfloor$, even if it finds the queue full, must have left by time 0. In what follows, we drop the $\lfloor \cdot \rfloor$ notation.

For fixed t, the collection $\{\exp(\boldsymbol{\theta} \cdot \tilde{\mathbf{X}}^{(L)}(0,t])\}$ is uniformly integrable, since $0 \leq \boldsymbol{\theta} \cdot \tilde{\mathbf{X}}^{(L)}(0,t] \leq \max|\theta_i| X^{(L)}(0, t+B/C]$, and $X^{(L)}(0, t+B/C]$ is L^p-bounded for some $p > 1$ (because the limiting moment generating function exists, by Condition 1).

For any $1 \leq s \leq t$, $\mathbb{P}(\tilde{X}_s^{(L)} \neq X_s^{(L)})$ is bounded by the probability that the queue is non-empty at either $s-1$ or s. By Theorem 3.11, this tends to 0. So $\exp(\boldsymbol{\theta} \cdot \tilde{\mathbf{X}}^{(L)}(0,t]) - \exp(\boldsymbol{\theta} \cdot \mathbf{X}^{(L)}(0,t])$ converges to 0 in probability.

Thus $\mathbb{E}\exp(\boldsymbol{\theta} \cdot \tilde{\mathbf{X}}^{(L)}(0,t]) - \mathbb{E}\exp(\boldsymbol{\theta} \cdot \mathbf{X}^{(L)}(0,t]) \to 0$, and taking logarithms gives the result. □

Definition 4 (Weak queue topology)
Recall the weak queue topology *wq defined on $\mathcal{X}$ by the metric*

$$d(\mathbf{x}, \mathbf{y}) = |Q(\mathbf{x}) - Q(\mathbf{y})| + \sum_{t=1}^{\infty} \frac{1 \wedge |x_t - y_t|}{2^t} \tag{4.1}$$

and $d(\mathbf{x}, \mathbf{y}) = \infty$ *if* $Q(\mathbf{x}) = \infty$ *or* $Q(\mathbf{y}) = \infty$.

Theorem 4.2 (Large timescale characteristics of the output)
If the input $\mathbf{X}^{(L)}$ *satisfies Conditions 1 and 2, and is stationary with mean rate strictly less than* C*, then the output* $\tilde{\mathbf{X}}^{(L)}$ *satisfies satisfies an LDP in* $(\mathcal{X}, wq)$ *with good rate function* $\mathbf{I}$ *as in Theorem 2.7.*

Proof. First, by the Dawson-Gärtner theorem for projective limits (see Dembo and Zeitouni [15] Theorem 4.6.1), the finite time LDPs of Theorem 4.1 can be extended to the full space $\mathcal{X}$ equipped with the projective limit topology, with good rate function $\mathbf{I}$. The projective limit topology corresponds to pointwise convergence of sequences, and can be made into a metric space with the metric given by the second term in (4.1). Denote this topology by p.

We want to strengthen this LDP from $(\mathcal{X}, p)$ to $(\mathcal{X}, wq)$. To do this we will use the Inverse Contraction Principle (Dembo and Zeitouni [15] Theorem 4.2.4). Since wq is stronger than p, the identity map from $(\mathcal{X}, wq)$ to $(\mathcal{X}, p)$ is continuous. And $\tilde{\mathbf{X}}^{(L)}$ satisfies an LDP in $(\mathcal{X}, p)$ with rate function $\mathbf{I}$. So if $\tilde{\mathbf{X}}^{(L)}$ is exponentially tight in $(\mathcal{X}, wq)$ then it satisfies an LDP in $(\mathcal{X}, wq)$ with the same rate function, and that rate function is good.

It remains to show that $\tilde{\mathbf{X}}^{(L)}$ is exponentially tight in $(\mathcal{X}, wq)$: in other words that there exist compact sets K_α in $(\mathcal{X}, wq)$ such that

$$\lim_{\alpha \to \infty} \limsup_{L \to \infty} \log \mathbb{P}(\tilde{\mathbf{X}}^{(L)} \notin K_\alpha) = -\infty. \tag{4.2}$$

Let μ be the mean rate of the $\mathbf{X}^L$, let $d_t = \sqrt{\log t / v(t)}$ where $v(t)$ is the scaling function from Condition 2, and choose the sets

$$K_\alpha = \left\{ \mathbf{x} : 0 \leq \frac{x(0,t]}{t + B/C} \leq \mu + \alpha d_{t+B/C} \ \forall\, t \right\}.$$

First, to show that K_α is compact. Since $\mathcal{X}$ is a metric space, it suffices to show that it is sequentially compact. So let $\mathbf{x}^k$ be a sequence of processes. Since the T-dimensional truncation of K_α is compact in $\mathbb{R}^t$, the intersection K_α is compact under the projective topology. That is, there is a subsequence $\mathbf{x}^j(k)$ which converges pointwise, say to $\mathbf{x}$. It remains to show that $\mathbf{x}^j \to \mathbf{x}$ under the weak queue topology. But if $\mathbf{x} \in K_\alpha$, there exists a t_0 such that for $t > t_0$, $x(0,t]/t < C$, and this t_0 can be chosen independently of $\mathbf{x}$. Therefore the queue size is just $Q(\mathbf{x}^j) = \sup_{t \leq t_0} x^j(0,t] - Ct$, which converges because the $\mathbf{x}^j$ converge pointwise. Thus K_α is compact.

Next, to show the equation (4.2). Since $\tilde{\mathbf{X}}^{(L)}(0,t] \leq \mathbf{X}^{(L)}(0, t + B/C]$, the left hand side is bounded above by the expression in the statement of Lemma 2.5, which is there shown to equal $-\infty$. □

Theorem 4.3 (Output stability)
If the input $\mathbf{X}^{(L)}$ *satisfies Conditions 1 and 2, and is stationary with mean rate strictly less than* C*, then for any* μ *greater than the mean rate, the output process* $\tilde{\mathbf{X}}^{(L)}$ *satisfies a sample path LDP in* $\mathcal{X}_\mu$ *equipped with the weak queue topology, with good rate function* $\mathbf{I}$.

Proof. We want to restrict the LDP of Theorem 4.2 to $\mathcal{X}_\mu$. By Dembo and Zeitouni [15] Lemma 4.1.5. it suffices to show that $\mathbf{I}(\mathbf{x}) = \infty$ if $\mathbf{x} \notin \mathcal{X}_\mu$, and that $\mathbb{P}(\tilde{\mathbf{X}}^L \in \mathcal{X}_\mu) = 1$. The proof of the first is identical to Theorem 2.6. For the second, that theorem also shows that for ε sufficiently small, $\mathbb{P}(\mathbf{X}^L(0,t]/t \leq \mu - \varepsilon \text{ eventually}) = 1$, and since $\tilde{\mathbf{X}}^L(0,t] \leq \mathbf{X}^L(0, t + B/C]$, we obtain the result. □

Before using these results to describe more interesting network models, we make a brief note about speed of convergence. We have shown that the input and output have essentially the same statistical characteristics, for large L, and it is interesting to know how large L needs to be for this to be accurate.

The idea behind the proof of Theorem 4.1 is that the probability that the queue is empty tends to one, and so $\mathbf{X}^{(L)}(0,t] - \tilde{\mathbf{X}}^{(L)}(0,t]$ converges to zero in probability. And large deviations gives us an estimate for the probability that the queue is nonempty. If I is the rate function for this event, as given in Theorem 3.11, then for any $\varepsilon > 0$ there exists an L_0 such that for $L \geq L_0$, $\mathbb{P}(Q^L > 0) \leq \exp -L(I - \varepsilon)$. Therefore

$$\mathbb{P}(\mathbf{X}^{(L)}(0,t] \neq \tilde{\mathbf{X}}^{(L)}(0,t]) \leq (t+1)e^{-L(I-\varepsilon)}.$$

For fixed θ and t, the difference in log moment generating functions $\mathbf{\Lambda}_t(\theta\mathbf{1})$ and $\tilde{\mathbf{\Lambda}}_t(\theta\mathbf{1})$ can be bounded similarly. So the error decays exponentially in LI at least.

4.4 Traffic mixes, decoupling, and networks

The results of the previous section can tell us a great deal about networks. Those results only applied to a single traffic class at a single router, but in this section we extend them to describe multiple traffic classes on multiple paths. The most significant result is *decoupling*, which means that different traffic flows sharing a router do not influence each other.

4.4.1 Traffic Mixes

In Section 4.3 we assumed that the aggregate input $\mathbf{X}^L$ to the router was the average of L independent identically distributed input processes. This was used in two ways. First, it gave a large deviations estimate for the probability that the queue is non-empty. Second, it let us describe a typical input using the moment generating function for the aggregate, $\mathbf{\Lambda}_t^L$.

We can still estimate the probability that the queue is non-empty and describe a typical input, even when the aggregate input is not made up of independent identically distributed flows. Let $\mathbf{Y}^L$ be the aggregate input, and let $\mathbf{X}^{(L)}$ be the single input we are interested in. Define the moment generating functions $\mathbf{M}_t^L(\boldsymbol{\theta}) = \frac{1}{L}\log \mathbb{E}\exp(\boldsymbol{\theta} \cdot \mathbf{Y}^L)$ and $\mathbf{\Lambda}_t^L(\boldsymbol{\theta}) = \log \mathbb{E}\exp(\boldsymbol{\theta} \cdot \mathbf{X}^{(L)})$. Suppose that $\mathbf{M}$ and $\mathbf{\Lambda}$ satisfy the conditions of Theorem 2.7 and are stationary, and that the mean rate of the aggregate input is less than the service rate. Then $\mathbf{M}$ gives a large deviations estimate for the event that the queue is non-empty, and $\mathbf{\Lambda}$ describes the input we are interested in. Theorems 4.1–4.3 go through unchanged, except that the rate I will depend on $\mathbf{M}$ rather than on $\mathbf{\Lambda}$.

There are many different ways of scaling the system to meet these conditions, with different numbers of inputs of different types. For example, let the aggregate input be made up of a mix of traffic types: $L\rho(j)$ copies of $\mathbf{X}^{(L)}(j)$ for $j = 1 \ldots J$, each traffic type satisfying the conditions of Theorem 2.7. Then $\mathbf{M}$ is just a linear combination of the moment generating functions for the different traffic types.

Another example is when the aggregate input is made up of L flows that were independent and identical when they entered the network, but which have passed through several queues before reaching the queue Q we are considering. Allow each flow to follow a different route, possibly involving feedback and interaction with other flows. This is interesting because it makes the flows neither independent nor identical. Let the maximum delay that each flow can incur before reaching Q be less than $D < \infty$. Let the aggregate input to Q be $Y^L(0,t]$; this is less than the original aggregate input $X^L(0,t+D]$ over a longer time interval. From Theorem 3.11 we find that if the mean rate of the $\mathbf{X}^L$ is less than C/D, then a queue with service rate C fed with $X^L(0,t+D]$ still empties with high probability, and so Q empties with high probability, and the results of the last section apply. (Unfortunately, since the inputs to a queue are not independent, we cannot use this to find an LDP for $\mathbf{Y}^L$ and thereby estimate the probability of overflow.)

4.4.2 Decoupling of Flows

Consider two independent inputs $\mathbf{X}$ and $\mathbf{Y}$ to a router whose aggregate input satisfies the conditions of Theorem 2.7, and is stationary with mean rate less than the service rate. (The (L) notation has been dropped here.) We know from the previous sections that in the limit, $\tilde{\mathbf{X}}$ has the same distribution as $\mathbf{X}$, and that $\tilde{\mathbf{Y}}$ has the same distribution as $\mathbf{Y}$. We can also view $\mathbf{X}+\mathbf{Y}$ as a single input to the queue, note that $\tilde{\mathbf{X}}+\tilde{\mathbf{Y}}$ has the same distribution as $\mathbf{X}+\mathbf{Y}$, and deduce that in the limit $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{Y}}$ are independent.

It might be expected that traffic flows would influence each other. For example, if $\mathbf{X}$ is very bursty and $\mathbf{Y}$ is smooth, one might expect $\tilde{\mathbf{X}}$ to be less bursty and $\tilde{\mathbf{Y}}$ to be less smooth, and indeed this can happen when the router only has a small number of inputs. But we have seen that in the many sources limiting regime it is not the case. In other words, $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{Y}}$ do *not* depend on the traffic mix at the router (so long as the total mean input rate is less than the service rate). This is known as *decoupling*.

Simulation results to illustrate decoupling are shown in Figure 4.2 on page 47. There are two classes of traffic: one bursty, producing one unit of work every third timestep, and the other smooth, producing 0.2 units of work every timestep. The service rate is 0.6 and the buffer size is 1.1. When there is only one traffic flow from each class, the outputs are coupled; but when there are three traffic flows from each class, and the buffer size and service rate are scaled up in proportion, the outputs are decoupled. As before we plot the effective bandwidth functions of the various traffic flows.

4.4.3 Feedforward networks of routers

A feedforward network of routers is one in which the routers may be ordered in such a way that for every flow the sequence of routers through which it passes

is strictly increasing. In this section so far we have shown that a flow passing through a router is essentially unchanged, even if several different types of flows use the router. This can be applied to a feedforward network of routers, as long as the network is scaled also.

Consider, for example, a simple network of two routers in tandem. Let the first router have L independent inputs, each distributed like $\mathbf{X}^{(L)}$. Let one of the outputs $\tilde{\mathbf{X}}^{(L)}$ be fed into the downstream router, along with a further $L-1$ independent copies of $\tilde{\mathbf{X}}^{(L)}$ from other upstream routers. Then the aggregate input to the downstream router satisfies a sample path LDP with the same rate function as that appearing in the LDP for the aggregate input to the upstream router, so we can estimate the overflow probability of the downstream queue with standard techniques.

For routers which are further downstream in the network, the proofs of Section 4.3 still work, if the maximum delay incurred by a flow at a router, B/C, is replaced by the maximum delay incurred by a flow in reaching the router under consideration.

4.5 Discussion

The fundamental result in Section 4.3 was simply stated, and the proof was not too long. But its consequences for networks, described in Section 4.4, are far-reaching. In this section we elaborate, describing our results in the more practical language of effective bandwidths. We also discuss their limitations and extensions.

Effective bandwidths for networks

The idea of effective bandwidth from Section 3.6 will help us with the interpretation. Recall that if a random traffic flow has effective bandwidth $\alpha(\theta, t)$ then it can be replaced by a constant flow of rate $\alpha(\hat{\theta}, \hat{t})$ without affecting the loss rate at a router, where $(\hat{\theta}, \hat{t})$ is the operating point of that router. We have shown in this chapter that a flow has the same effective bandwidth function at all points in a network (though the different routers will typically have different operating points, so the values of the function will be different).

This means, for example, that the effective bandwidth of a flow in queueing networks plays a similar role to the bandwidth of a call in loss networks. This encourages the hope that well-understood techniques and insights from loss networks (reviewed by Kelly [28]) can be applied to queueing networks.

It also makes it easier to understand feedback and rate control for adaptive traffic—that is, traffic which can alter its rate in response to congestion-indicating signals from the network. It is natural to believe that feedback from a router to a user should depend on the characteristics of the traffic from that user, *as seen by the router*. If the effective bandwidth function changed along the route, depending on interactions with other flows at other routers, then the user might have difficulty in making effective use of the feedback signals, because she would not know how her traffic had been shaped by the intervening routers. But it does not change, and so she can better interpret feedback.

The key idea is that we can meaningfully talk about the characteristics of, say, video traffic, because the flow retains these characteristics regardless of its

interactions with other flows in various routers throughout the network.

The network limit

The results in this chapter are considerably cleaner than earlier large deviations results for networks. This is because we have taken limits as the structure of the network changes. Most previous work, on the other hand, has kept the structure of the network fixed and looked at limits where the traffic changes.

Neither approach is intrinsically better (except insofar as one gives cleaner results). What matters practically is under what circumstances each is accurate. Our limit seems better-suited to networks with what we call *diverse routing*, by which we mean that many of the inputs at any router are reasonably independent, though it is difficult to make such a vague claim precise.

We have not dwelt on the question of how many input processes are needed for our limiting result to be accurate. Numerical simulation, illustrated in Figures 4.1 and 4.2, shows that in some cases only a small number of independent inputs are needed to make the input and output look nearly identical. The real question, though, is: how many input processes are needed for reasonable convergence *over the scale of interest*? If we are interested in the probability of overflow at a downstream router, we want reasonable convergence of the moment generating function at the critical timescale and spacescale for that router. For fixed θ and t, we noted in Section 4.3 that the difference between the moment generating functions for the input and output is bounded by a term which decays exponentially in LI, where L is the number of inputs and I is the rate function for the event that the upstream queue is nonempty. The accuracy of the large deviations estimate of Theorem 3.11 must also be taken into account; this has been studied by Likhanov and Mazumdar [34].

Limitations and extensions

The core of the argument is Theorem 4.1, which proves that the limiting moment generating function of the output process is the same as that of the input. It relies on the fact that when there are many independent sources, the queue empties regularly, with high probability. That it empties regularly is a reasonable engineering constraint for high-performance networks, in which delay and cell loss probabilities should be small. This constraint is satisfied by any work-conserving queue (that is, any queue which does not idle when there is work waiting).

The theorem is proved for the case of a queue with a finite buffer. It seems likely that the result still holds for queues with infinite buffers and for other regimes like priority queues. The finiteness of the buffer is used to bound the amount of work that can leave the queue over a period of time, to give uniform integrability; for those other cases some other way of proving uniform integrability would be needed.

Closely related to this is the problem of continuous time. In the continuous time formulation, it is not true that at any instant in time the queue is overwhelmingly likely to be empty—even in the simplest example of Poisson arrivals and exponential service times there are likely to be small fluctuations in queue size. What is true though is that at any instant in time it is overwhelmingly

likely that the queue will shortly be empty, and so the queueing delay experienced by any incoming work should be extremely small. Unfortunately we are again left with the problem of uniform integrability: while this queueing delay is extremely small with high probability it is nonetheless unbounded, so we cannot use it directly to bound the amount of work that can leave the queue over a period of time.

When feedforward networks are so simple, it is tempting to conjecture that similar results might hold in networks with feedback. There are numerous examples of pathological behaviour in finite networks. But in large networks, under this many sources regime, we expect that queues will still empty sufficiently often, and the main result will still hold.

4.6 Summary

The conclusions of this chapter are very simple to state, at least in an imprecise way. In a network with diverse routing, by which we mean that most of input flows at a router are reasonably independent, the statistical characteristics of a flow of traffic are the same at all points in the network.

More precisely, the distribution of a flow of traffic is preserved by passage through a router, in the limit where the number of independent input flows to that router increases and the service rate and buffer size increase in proportion.

This is a limiting result. But simulation suggests that it can still be reasonably accurate even for a handful of independent sources. And the theory is useful at least as much for the insights it gives as for numerical estimates.

It dramatically simplifies the analysis of networks of routers with different classes of traffic.

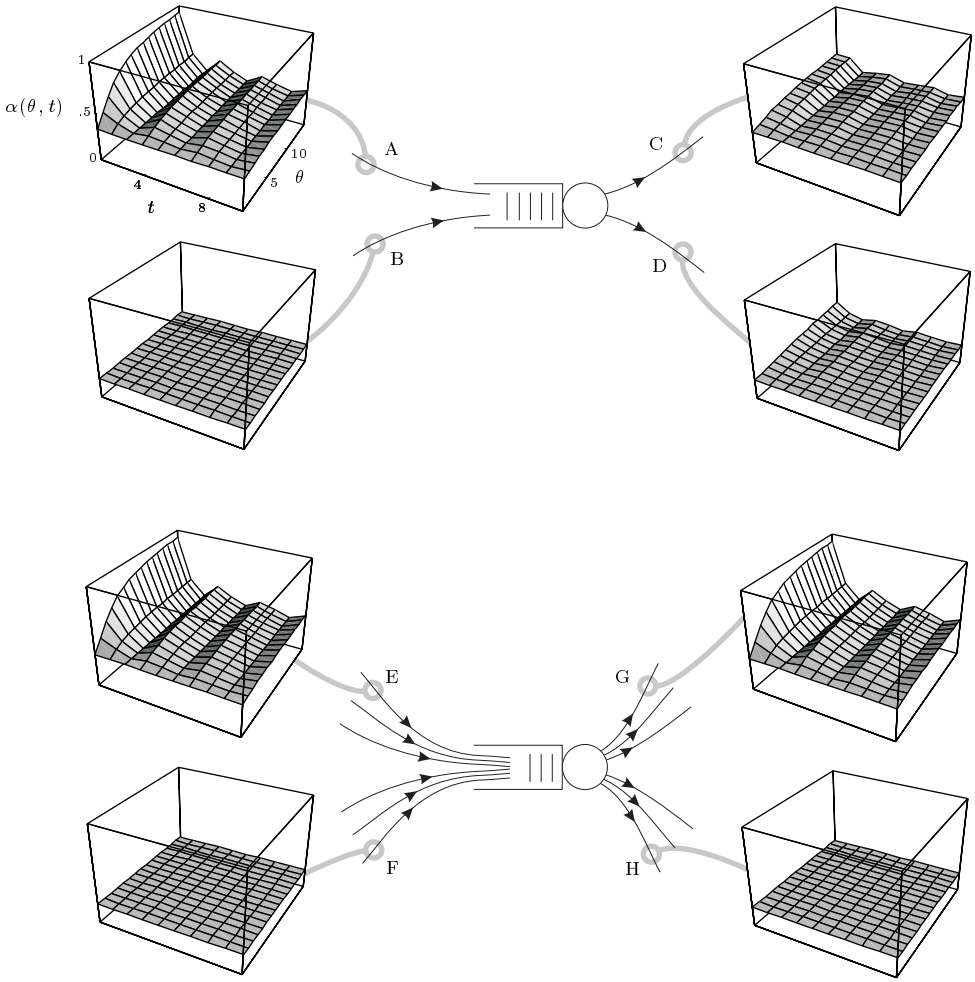

Figure 4.2: Decoupling. A router has two inputs flows, one bursty (A) and the other smooth (B). In passing through the switch together they become coupled: the bursty one is made smoother (C), and the smooth one is made burstier (D). However, if the system is scaled up so that there are three input flows of the bursty type (E) and three input flows of the smooth type (F), and the service rate and buffer size are increased in proportion, then the corresponding output flows (G) and (H) are decoupled. The graphs plot the effective bandwidth functions of the flows, all to the same scale. The effective bandwidth of a flow is a convenient representation of its burstiness over different timescales and spacescales.

Chapter 5

Congestion

In the preceding chapters, we have developed theory to model traffic as it travels through a network and to explain how congestion arises. Now we use these ideas to address a specific problem facing the Internet today: how routers should respond to congestion.

Commonly, a router simply drops incoming packets when there is no space for them in its buffer, and end-systems try to gauge from the frequency of drops the rate at which they should be transmitting. But dropping packets in this way is a very blunt sort of signal: it tends to give the wrong amount of feedback to the wrong end-users; and anyway, it would be better if congestion could be signalled before it became a problem.

The technical groundwork for fixing these problems has been laid by the Internet engineering community with an RFC [48] which proposes a scheme called Explicit Congestion Notification, or ECN. (RFC stands for Request For Comments, the name given to documents proposing and specifying Internet standards. This and other Internet acronyms are listed in the Glossary.) Under the ECN proposal, routers can *mark* packets instead of dropping them, and end-systems are expected to respond to marks as they would to drops. The proposal leaves open the problem of what marking algorithm routers should use.

Marks can be thought of as a technological solution to the problem of congestion, but they can also be thought of economically as a pricing mechanism. Prices in a market economy have a similar role to marks in the Internet: to convey information and to direct consumption. So economic theory plays a significant part in the study of marking algorithms.

In this chapter we consider the problem of what a marking algorithm should do, paying particular attention to what it means to mark fairly. There have been many different suggestions for marking and pricing schemes, and in Section 5.1 we describe some of them, before explaining in Section 5.2 what we think should be the goals of a marking algorithm.

In Sections 5.3–5.5 we describe three different ways to define fairness and efficiency in marking. In Section 5.3 we propose the EB definition, drawing on effective bandwidth theory; in Section 5.4 we propose the ΔL definition, drawing on economic efficiency theory, and in Section 5.5 we propose the SPSP definition, drawing on economic ideas of fairness. In Section 5.6 we compare and reconcile the three different definitions, and indicate why we believe SPSP is the most

appropriate.

Having explained how a marking algorithm ought to work, we go on in Section 5.7 to study various algorithms that have been proposed, including the RED algorithm designed by Floyd and Jacobson [22], and point out how they can be unfair. The principal tool is the sample path large deviations principle of Chapter 2, and the idea of the most likely path. It turns out that a few simple changes to RED can make it significantly fairer, and we summarise them in a new algorithm called ROSE. We conclude in Section 5.8 by making specific comparisons between our results and what others have found.

5.1 Related work

Most traffic in the Internet today is controlled by the TCP algorithm. It controls the rate at which packets are sent, as follows: when there is congestion and packets are dropped, the rate is reduced; when no packets are dropped, suggesting that it is lower than necessary, the rate is cautiously increased. The algorithm was designed in 1988 by Jacobson [24] in response to congestion collapse in the Internet, caused by end-systems which did not back off enough. It has been extremely successful, and has lasted over a decade with only minor modifications. But a decade is many generations in Internet time, and TCP is beginning to show its age in two ways.

TCP was designed to work well when nothing is known about the network beyond the trivial fact that it drops packets when overloaded. However, networks are becoming slightly more intelligent than they used to be, and this raises the possibility of new and better ways of signalling congestion and of responding to it. In the past, routers have only dropped incoming packets when there is no space for them to be queued. It takes time for an end-system to detect the drop and reduce its rate—so end-systems are only notified of congestion when the time to prevent it has passed. It has long been recognised that routers are well-placed to detect congestion and signal it before it becomes a problem, though as Ramakrishnan and Jain [49] describe, early schemes did not catch on. More recently, various router algorithms have been proposed which signal incipient congestion by dropping packets before the buffer is full. The RED algorithm by Floyd and Jacobson [22] has received much attention and has even been implemented in commercial routers [8]. And ECN will mean that routers can signal congestion without dropping packets, by marking them instead.

TCP is becoming dated in another way. It is a one-size-fits-all algorithm: the rate-adaptation algorithm leads to one particular allocation of network capacity. Applications which need more bandwidth have no way of indicating this (though by disabling the rate-reducing part of TCP or by using multiple simultaneous connections one can unscrupulously get a larger share). And applications for which TCP is not appropriate, like streaming multimedia, may use an entirely different sort of rate-adaptation algorithm—or none at all—and can compete unfairly with TCP.

A lot of work has been done on how the network can provide different levels of service—that is, how limited network resources should be divided between competing users with differing requirements. This problem is natural material for economic analysis. The economic approach to congestion control began with an influential but impractical market-based proposal by MacKie-Mason

and Varian [37] in which each user attaches prices to individual packets and routers hold auctions to decide which packets get served. Since this there have been many more proposals, all aiming to turn the technological problem of congestion into an economic one of prices for users.

Typically it is assumed that each user sends work at some rate which he can change in response to charges. For example, in the model of Low and Lapsley [36], each user chooses a rate according to his preferences, and is charged, and the charges are chosen so that social welfare is maximized subject to capacity constraints. Chen and Park [6] let each user allocate his total rate among a class of services and seek to maximize social welfare, measured in terms of constraints on a fixed class of quality of service indicators such as average delay or loss.

The problem with these approaches is that they ignore the random bursty nature of traffic, which is what causes most of the problem of congestion. By contrast, Courcoubetis et al. [9] explicitly take random traffic flows into account by using effective bandwidth as a basis for charging. Their model of user behaviour is well-suited to telephony-like networks with a fixed range of services, but not so well-suited to networks like the Internet, in which users have complete freedom to send their traffic however they like.

An elegant approach to the problems of marking and pricing has recently been proposed by Gibbens and Kelly [23]. This chapter follows on from their work, which we describe in more detail in the following sections.

We are not aware of any analysis of marking algorithms other than by simulation, and hence believe that our use of large deviations for this purpose is unique.

5.2 The goals of marking

Most of this chapter is given to trying to define fairness in marking algorithms. The ideas of fairness and justice in allocating resources and setting prices have occupied thinkers since the beginning of civilisation; more recent thinkers range from Sen [50] to John Paul II [25]. Fairness has been taken to mean very different things even in the limited arena of bandwidth allocation—and the very need for fairness is not always recognised. We must therefore explain carefully what we hope to achieve. *We want marking algorithms to allocate marks according to the amount of capacity that each flow consumes.* This brief statement needs considerable elaboration.

Why mark fairly?

The first concern of engineers who design congestion control mechanisms is whether they are efficient: that is, whether better use could be made of the available resources. Efficiency too is the first thing that a modern microeconomist looks for: the standard textbook on microeconomics by Varian [56] has much to say about efficiency and nothing at all about fairness (though Varian himself has made many contributions to the theory of superfairness [55]).

And yet nearly every paper proposing a new marking algorithm or a modification to TCP asks whether it is fair (though often with a simplistic idea of what fairness means). In economics too, regulators and the public are often at least as interested in fairness as in efficiency. The authors of two main economic

books on fairness, Baumol [1] and Zajac [61], were both involved in US government investigations of AT&T's pricing policy. So at the very least we want to know what it means to mark fairly.

Zajac describes very many cases in which fairness and efficiency are opposed. Happily, in the problem of bandwidth allocation they are mostly aligned, and this chapter is as much a study of efficiency as of fairness. In fact, the reason we focus on fairness is because it turns out to be *easier* to define than efficiency. We will give three different definitions of what a marking algorithm should achieve, based on three different models for user behaviour. From these three definitions we will distill a single notion of fairness, but it does not seem possible to do the same for efficiency.

What fairness should *not* involve

Congestion control is performed in two places: at the periphery of a network (the end users and their access points) and in its core; and it is crucially important to properly divide responsibility between them.

In TCP all the responsibility rested with end-users, because the core was assumed not to be intelligent enough to do anything more than drop packets. Floyd and Jacobson in the design of RED sought a better division of responsibility. They had the goal that their algorithm should mark flows fairly, and expected that well-behaved flows at least should react accordingly. Lin and Morris [35] go further in their design of the FRED algorithm. Their explicit goal is to mark in such a way as to give a fair *allocation* of bandwidth, taking into account that some flows respond less quickly than others.

The problem with this last approach is that routers are badly placed to decide what users value and how they will react: only users know that. What routers are well-placed for is measuring utilization and congestion—so the focus of this chapter is on routers, and how they can respond to congestion by marking packets. We do not assume that users should be given an equal share of bandwidth: we merely mark in proportion to the amount they have taken, as we believe that trying to make routers do anything more would result in an inflexible network with a limited range of services.

Of course, users ought to respond in some way to marks. We will not go as far as the ECN proposal [48] in dictating the form of this response. For example, if marks form the basis of a usage-sensitive pricing scheme, users may be safely left to respond as they see fit. We postpone further discussion of how users should be encouraged to respond until Section 5.8.

What fairness should involve

Floyd and Jacobson set the goal that RED should mark flows fairly. They note that fairness is not well-defined, and design the algorithm to mark roughly in proportion to a flow's average bandwidth. Lin and Morris with FRED are less circumspect, and explicitly seek an equal allocation of average bandwidth. While it is certainly true that if the average bandwidth coming into a router is higher than the service rate there will be congestion, the problems come mainly from bursts in the traffic. We therefore seek to mark each flow in proportion to how much of the resource it uses, taking account of its fluctuations.

Another aspect of marking which has received only a little attention [27] is its impact on routing. Ideally, a router should generate marks in proportion to its congestion, so that users have a way to measure and an incentive to choose the route with the least impact on the network. In other words, it is only fair that a user using an uncongested resource should be marked less than a similar user on a congested resource.

The marks given by a router to a flow should reflect
- *how much of the capacity it uses, and*
- *the congestion at the router.*

The hard part is in finding the right measure of how much capacity a flow uses and of how congested the router is. There will inevitably be some degree of judgement in trying to define such concepts, especially as there are several different candidate definitions. In the next three sections we will give three different definitions, EB, ΔL, and SPSP, drawing on effective bandwidth theory and economics. We then reconcile them in Section 5.6.

5.3 Effective bandwidths and marking: EB

What Baumol describes as the 'crudest but most direct approach ... to determine the fair set of prices' is called *full allocation of costs*. To determine fair prices, the total cost to a company is entirely divided between the products it makes, and the fair price for a product is its allocated cost. He calls it crude because the allocation of costs to products is generally arbitrary, and because no account is taken of consumer preferences.

In this section we will give a definition of fairness and efficiency in marking based on effective bandwidth theory. Our definition, which we will call EB, is a way of fully allocating the costs of congestion to users. In the limited domain of bandwidth allocation there are sound reasons for doing this, for example as in the model of Courcoubetis et al. [9]. First we will recall the theory, which was described in Section 3.6. For the purposes of fairness, what matters is the following summary.

5.3.1 Effective bandwidth theory

The effective bandwidth function $\alpha(\theta, t)$ of a random traffic flow is a measure of the capacity it consumes, somewhere between the mean and peak rates, encoding all the important information about the flow's burstiness. The convenient feature is that the rate function for loss probability at a router is governed by the sum of effective bandwidths of the input flows. So if a router has several input flows of different types, then the effective bandwidth function measures the tradeoff between them. For example, suppose that a router has inputs of types A and B and at the operating point $(\hat{\theta}, \hat{t})$ of the queue, $\alpha_A(\hat{\theta}, \hat{t}) = 2\alpha_B(\hat{\theta}, \hat{t})$. Then replacing one flow of type A by two flows of type B will not affect the loss probability.

We do not need the next result immediately, but it will be useful in Section 5.6. Recall that the most likely path to overflow is given by $\hat{\mathbf{x}}$ in equation (3.11), and that the amount of work produced by $\hat{\mathbf{x}}$ in the busy period leading

to overflow is

$$\hat{x}(0,\hat{t}] = \frac{\partial}{\partial\theta}\theta\hat{t}\alpha(\theta,\hat{t}), \tag{5.1}$$

where the derivative is taken at $\hat{\theta}$.

5.3.2 Fairness

Effective bandwidth measures the impact of a flow at a resource, so the first point of our goals of fairness in Section 5.2 would suggest marking in proportion to effective bandwidth—or, equivalently, marking in proportion to $\hat{t}\alpha(\hat{\theta},\hat{t})$ which has the right units—and we shall say that such a marking scheme satisfies the EB definition of fairness. If one user of type A can be replaced by two users of type B without affecting loss probability, it is fair that a user of type A be charged twice as much as a user of type B. (We shall revisit this definition in Section 5.6.)

We can also address the second point. The ECN proposal [48] requires that one mark be equivalent to one dropped packet. We might loosen this a little, and say that one dropped packet should be worth a fixed number of marks. In either case, the large deviations interpretation is that the rate function for overflow should be equal to the rate function for marking. To see this, let I_M be the rate function for marking and I_O the rate function for overflow. This means that when the system is scaled up to have L users and the service rate and buffer size are scaled up by L, the probability of marking is roughly e^{-LI_M} while that of overflow is e^{-LI_O}. If the rate functions are not equal, then as the system scales up the number of marks per dropped packet tends to either zero or infinity.

We saw in Chapter 4 that the effective bandwidth of a flow is preserved as it travels through a network, at least as long as routing is diverse. This makes it easy to see that marking according to effective bandwidth is reasonable in networks, not just in isolated routers, and we do not need to worry about flows being made smoother or more bursty as they progress through the network.

5.3.3 Efficiency

Courcoubetis et al. [9] describe an economic model of user behaviour, under which a social optimum is attained by charging in proportion to effective bandwidth. We will not repeat their model here, as we look at social optima in much more detail in the next section. We will simply note for the moment that social optima are always economically efficient, so that in this model fairness and efficiency are both served by charging in proportion to effective bandwidth.

5.3.4 Summary of EB

Large deviations and effective bandwidth theory suggest a full allocation of costs, in which flows are marked according to their effective bandwidths.

Large deviations can give us a great deal of information. With it, for example, we can model nearly any sort of random traffic (including long-range dependent sources like fractional Brownian motion, Example 2.4 on page 14);

we can calculate quantities such as the loss rate and the most likely path to overflow; and we can analyse the behaviour of traffic in a network.

This comes at the price of loosing some details. For example, it does not distinguish precisely how many marks correspond to a dropped packet. To give a different perspective, we now take the economic view. This gives more precise answers, but cannot answer as many questions.

5.4 Economics and efficiency

This and the following section describe an economic approach to marking. Economists have developed ways to model the problem of individuals competing for limited resources, which is exactly our problem of congestion control—they treat prices as a mechanism for directing consumption, and we will treat marks in just the same way. The difference with standard economic theory is that the technological infrastructure of the Internet may, according to MacKie-Mason and Varian [37], allow 'breakthroughs ... in the area of in-line distributed accounting.' The breakthrough that we are looking for is the ability to charge users in a way which precisely reflects their actions, using only the very simple mechanism of marking packets.

In this section we will look at the problem of efficiency. An allocation of goods and prices is said to be *efficient* if there is no change that would simultaneously benefit someone and harm no-one, as measured by their utility functions. In fact, we will concentrate on one particular sort of efficient allocation: the social welfare optimum, in which the sum of everyone's utility functions is maximised.

This is a very simplistic approach to efficiency, and modern economists try to steer clear of interpersonal comparisons of utility. Yet, as Baumol [1] and others note, this sort of comparison is inherent in defining fairness. And in this chapter we are as interested in fairness as we are in efficiency.

First, we briefly discuss in Section 5.4.1 the relationship between marking and charging. In Section 5.4.2 we review the problem of marking when each user sets the rate at which they send work, largely following Gibbens and Kelly [23]. They go on to consider how users should respond to such a marking scheme, and Tan [53, 54] analyses the stability of the whole system. We however will stay with the topic of marking algorithms, and in Section 5.4.3 we describe how marking should work when users send random flows. This leads to a definition of fair and efficient marking, which we call ΔL. (Here we only prove efficiency; in Section 5.5 we explain why it can also be taken to define fairness.) Finally in Section 5.4.4 we discuss some limitations of this definition.

5.4.1 Dropping, marking, and charging

First, a note on marking and charging. We will mainly refer to charging rather than marking in the rest of this section, so it is important to make clear the relationship between the two ideas.

Perhaps the most apparent costs in the Internet are infrastructure costs. It is easy to put a price on a new fibre-optic cable or a new router. We are not concerned here with this sort of cost: we are interested instead in costs associated with congestion. Even when all the infrastructure has been paid for,

congestion can still be a problem. The standard economic way of coping with congestion is to levy extra charges on people who use congested resources.

Marking in the Internet is intended to achieve exactly the same things as congestion-pricing in economics, which is why we will use the term *charging* rather than *marking*. However, while people will naturally respond to monetary charges, it is less clear what incentives there might be for responding to marks. If users were charged say a millionth of a pence for each mark, the incentives would be obvious. But even if the Internet is not yet ready for full-blown congestion-based pricing, economic theory can still help us understand what the cost of congestion is to users of the network, and how users' demands for more bandwidth can be reconciled with the network's capacity constraints. We will postpone further discussion of how to encourage and enforce good behaviour until Section 5.8.

A user's response to marks will be governed by what the marks signify. The ECN proposal [48] specifies that users must respond to marks in essentially the same way as they respond to dropped packets. The reasons for this are largely historical; and while our discussion of marking refers the ECN mechanism, it is based on very different premises. Nonetheless, we too will treat marks as akin to drops. We will take the frequency with which a user's packets are dropped to be the primary measure of his dissatisfaction, and so it will be natural to measure his charge in the same units.

In the rest of this section we will discuss pricing structures rather than marking algorithms. In translating from charges into marks, it should be borne in mind that a user 'feels the cost' of both marks and drops. For example, a user who should incur charge P, of whose packets L are dropped, need only have $P - L$ of his remaining packets marked.

5.4.2 Efficiently marking fluid flows

Consider a network with a set $\mathcal{R}$ of resources and a set $\mathcal{U}$ of users. Identify a user $u \in \mathcal{U}$ with the set of resources $u \subset \mathcal{R}$ he wants to use. Suppose he sends work at constant deterministic rate x_u and has utility $U_u(x_u)$ in doing so. We will take one dropped packet to be our unit of utility. We also need a utility term to indicate the cost of congestion: let $C_{ru}(x)$ be the average loss at resource r experienced by user u when the total load in that resource is x. (The idea of average loss is left intentionally ambiguous for now. It will be made clear when we go on to consider random flows in Section 5.4.3.) Write $\mathbf{x}$ for the vector $(x_u)_{u\in\mathcal{U}}$. Then each user will seek to

$$\max_{x_u} U_u(x_u) - \sum_{r\in u} C_{ru}(y_r) \quad \text{where} \quad y_r = \sum_{u:r\in u} x_u.$$

Let us consider the social welfare problem: to maximise the net utility. In other words,

$$\max_{\mathbf{x}} \sum_{u\in\mathcal{U}} U_u(x_u) - \sum_{r\in\mathcal{R}} C_r(y_r) \quad \text{such that} \quad x_u \geq 0 \ \ \forall \ \ u \in \mathcal{U} \tag{5.2}$$

where

$$y_r = \sum_{u:r\in u} x_u \quad \text{and} \quad C_r(y_r) = \sum_{u:r\in u} C_{ru}(y_r).$$

This can be solved with normal Lagrangian techniques. Define $\mathcal{L}$ by

$$\mathcal{L} = \sum_{u \in \mathcal{U}} U_u(x_u) - \sum_{r \in \mathcal{R}} C_r(y_r) + \sum_{r \in \mathcal{R}} \lambda_r \left(y_r - \sum_{u: r \in u} x_u \right) \tag{5.3}$$

and solve $\partial \mathcal{L}/\partial y_r = 0$ and $\partial \mathcal{L}/\partial x_u = 0$ (or $x_u = 0$ and $\partial \mathcal{L}/\partial x_u \leq 0$). This gives

$$\begin{aligned} \lambda_r &= \frac{dC_r}{dy_r} \quad \text{and} \\ \frac{dU_u}{dx_u} &= \sum_{r \in u} \lambda_r \quad \text{if } x_u > 0. \end{aligned} \tag{5.4}$$

This solution can be written in an intuitively appealing way. Suppose that each user can adjust his rate x_u, and for sending x_u is receives $P_u(\mathbf{x})$ marks. Then, if he ignores the other users, he would act to maximise $U_u(x_u) - P_u(\mathbf{x})$. Let us choose the shadow price

$$P_u(\mathbf{x}) = x_u \sum_{r \in u} \lambda_r. \tag{5.5}$$

Then the solution to the system of equations (5.4) coincides with the solution to the welfare problem (5.2). (Actually, this charge should be reduced for users who experience drops. It will be easier to see how when we go on to look at random processes in the next section.)

The pricing structure (5.5) leads to a decentralised solution, in the following sense. Each resource computes its own price per unit flow $dC_r(y_r)/dy_r$, and that price is communicated to everyone using that resource. Each user observes the total price he is charged, and adjusts his bandwidth accordingly. By this choice of prices, the interests of users are harnessed to achieve a social optimum.

One example, first described by Gibbens and Kelly [23], is especially worth noting, as it leads to a very simple marking algorithm.

Example 5.1
As usual, assume a slotted time traffic model. Also assume for simplicity that all packets are the same size. Consider a bufferless resource fed by Poisson flows of packets. Specifically, suppose that each user u sends a Poisson flow of packets of rate x_u, and that $C_r(y_r)$ is the expected number of dropped packets at a bufferless resource of service rate C when fed with an input Y_r which is Poisson with parameter y_r (i.e., $C_r(y_r) = \mathbb{E}(Y_r - C)^+$). Then it can be shown that the correct expected charge given in (5.5) is attained by the following marking algorithm: in a timeslot in which overflow occurs, mark every packet that arrived in that timeslot (except for dropped packets, which do not need to be marked.) $\diamond$

5.4.3 Efficiently marking random flows: ΔL

The last section assumed fluid traffic flows, or at least traffic flows parameterized by a scalar rate. But the optimization (5.2) can be interpreted another way, to say how general random traffic flows should be marked. This will enable us to draw links with effective bandwidth theory.

Consider again a slotted time model in which all packets are the same size, and a network of bufferless resources. Suppose that each user u transmits a random amount of work at each timestep. Each user will have a probability distribution controlling how much work is sent, and it is over these distributions that we wish to optimise. So let x_u in (5.2) be a distribution over the nonnegative integers, rather than a scalar as in the last section. This means that y_r is also a distribution, the distribution of the total amount of work arriving at resource r in a single timestep. (To avoid problems with what happens upstream, we could restrict attention to a single resource. It is easiest to deal what happens upstream using effective bandwidths and the results of Chapter 4.) We can now be clear about how we measure the cost of congestion: let $C_{ru}(x)$ be the expected number of packets belonging to user u which are dropped at resource r when the total load is x.

The notation becomes a little more complicated here, but the argument is just the same as in the last section. Let us write Z for the random variable with distribution z, and $z(n)$ for $\mathbb{P}(Z = n)$. Let $L_r(Y)$ be the number of packets dropped at resource r when fed with Y. Then $C_r(y_r) = \mathbb{E}L_r(Y_r)$, which expands to $\sum_n L_r(n)y_r(n)$. Now the multipliers λ_r are measures on the nonnegative integers, and the Lagrangian (5.3) becomes

$$\mathcal{L} = \sum_u U_u(x_u) - \sum_r \mathbb{E}L_r(Y_r) + \sum_r \sum_n \lambda_r(n)\Big(\mathbb{P}(Y_r = n) - \mathbb{P}(\sum_{u:r\in u} X_u = n)\Big).$$

Solving $\partial\mathcal{L}/\partial y_r(n) = 0$ gives

$$\lambda_r(n) = \frac{\partial \mathbb{E}L_r(Y_r)}{\partial y_r(n)} = L_r(n)$$

and solving $\partial\mathcal{L}/\partial x_u(n) = 0$ gives

$$\begin{aligned}\frac{\partial U_u(x_u)}{\partial x_u(n)} &= \sum_{r,m} \lambda_r(m) \frac{\partial \mathbb{P}(\sum_{v:r\in v} X_v = m)}{\partial x_u(n)} \\ &= \sum_{r\in u,m} \lambda_r(m)\mathbb{P}(\sum_{v:r\in v} X_v = m | X_u = n) \\ &= \sum_{r\in u} \mathbb{E}\big(L_r(Y_r)|X_u = n\big).\end{aligned}$$

Really, we should include constraints that $\sum_n x_u(n) = 1$ and $0 \le x_u(n) \le 1$. But by parameterizing the distribution of X_u differently, it can be shown that these constraints do not affect the solution.

We can again construct the shadow prices which make the solutions to the user problems coincide with the social optimum:

$$P_u(\mathbf{x}) = \sum_n x_u(n) \sum_{r\in u,m} \lambda_r(m)\mathbb{P}(Y_r = m|X_u = n) = \sum_{r\in u} \mathbb{E}L_r(Y_r).$$

In fact, this is a little bit silly, because even when the user sends nothing (i.e. $\mathbb{P}(X_u = 0) = 1$) he is still charged. This has happened because the space of probability measures for X_u over which we are optimizing is affine, not linear.

So we might as well assert that when a user sends nothing he should be charged nothing, which leads to the price

$$P_u(\mathbf{x}) = \sum_{r \in u} \mathbb{E}L_r(Y_r) - \mathbb{E}L_r(Y_r - X_u).$$

This pricing scheme is naturally attained by charging $L_r(Y_r) - L_r(Y_r - X_u)$ in each instance. We will explain in the Section 5.5 why this can be considered to be fair. We shall call it the ΔL pricing scheme, and say that any marking algorithm which achieves it satisfies the ΔL definition of fairness.

It is true much more widely that this sort of pricing structure (total cost with an individual minus total cost without that individual) will lead to a social optimum. The only distinguishing feature of our probability model is that this charge arises as a shadow price. Normally the shadow price comes out as a derivative, as in (5.4) and (5.5).

So far we have assumed a bufferless model. The same argument works for queues, though with a slight technical difficulty. The problem is that a queue can overflow over any timescale, and so we would need to consider x_u to be a distribution of a stationary process indexed by the positive integers. This has more than countably many sample points, so a more intricate analysis would be needed. To avoid these problems, we can note that real queues only overflow over a finite timescale, and only consider marginal distributions over this timescale. This means that $\mathbb{E}L_r(Y_r) - \mathbb{E}L_r(Y_r - X_u)$ is still the right charge to levy, where Y_r and X_u are to be seen as entire processes. Henceforth we drop the r subscript for simplicity and talk about single resources, remembering that marks from different resources should be summed.

Recall that $L(m)$ is the number of packets dropped at a queue when the aggregate input is m. So ΔL says that the charge assigned to a user should equal the difference in the total number of packets dropped between the case where the user is present and the case where he is not. Over a long enough time period, this gives the right expected charge.

5.4.4 Problems with ΔL

There are several concerns with ΔL, which we now note.

We have simplistically taken the social welfare function (5.2) to be the sum of utilities of each of the users. This is an arbitrary way to balance the needs of different users (though it is reasonable from the point of view of fairness). A more general concept is the idea of Pareto efficiency: a Pareto efficient allocation is one in which there is no change which harms no-one and strictly benefits someone, as measured by their utilities

$$U_u(x_u) - \sum_{r \in u} \mathbb{E}L_{ru}(Y_r).$$

The problem with trying to characterize Pareto efficient allocations is that they depend in detail on the loss function $L_{ru}(Y_r)$, which depends on the exact order in which packets arrive. This would require more precise assumptions than are justified by the queueing models used in this thesis.

A more pressing concern is about strategic play. We have assumed that each user will try to maximize his own utility, independent of other users. But

we would expect that a strategic user would anticipate the effect of his actions on prices and adjust his behaviour, leading away from the social optimum. The idea of a Nash equilibrium describes what would happen when users play strategically; but to find these equilibria we have to make further assumptions about the options open to each user. Gibbens and Kelly [23] give some examples of what might happen. A user who takes up a large fraction of the capacity and who does not anticipate the effect of his actions would back off a certain amount; if he did anticipate, he would back off more. Nonetheless, when there are many small users, this should not be much of a problem.

There are also problems with defining what we mean by a user. The optimization argument took a user to be an entity that values what it sends and can shape its traffic in response to charges, and supposed that different users shape their traffic independently. But what is a user? Is it an institution? a person sitting at a computer? an application program? a flow of traffic from an application? an individual packet? Sometimes each of these levels should be considered a user, and sometimes they act together. Some preliminary discussion about how these levels interact is given by Key et al. [31].

5.4.5 Summary of economics and efficiency

We have found a pricing scheme, ΔL, which maximises social welfare (and is therefore efficient) assuming a particular model of user behaviour—namely, that users have total freedom in choosing the distribution of the traffic they send, and that their cost is measured by their expected loss. The pricing scheme we found is that user u should be charged the shadow price $\mathbb{E}L(Y) - \mathbb{E}L(Y - X_u)$. This rule is summarized by *make each user feel any loss he causes as though it were his own.* A pricing scheme like this is called a Pigovian tax. It is the standard economic prescription for achieving a socially desirable outcome in the presence of social costs.

This has several problems. The most significant is the problem of whom to take to be a user. In the next section we go on to consider economic views of fairness, and indicate how the problem may be remedied.

5.5 Economics and fairness

In Section 5.4 we found that the pricing scheme ΔL leads to an efficient allocation of bandwidth (at least under the model of user behaviour given in that section). It has the further virtue that it is fair by definition, or at least by one of the definitions of fairness that economists have proposed. In Section 5.3 we suggested charging in proportion to effective bandwidth, which is fair according to another definition.

In this section we will review some of the different definitions of fairness that economists have given. We will describe superfairness, the burden test, incremental fairness and anonymous equity, and game-theory. And we will introduce another pricing scheme, called SPSP. The principal references are Baumol [1] and Zajac [61].

5.5.1 Superfairness

Perhaps the most mathematically developed idea of fairness is the theory of superfairness. An individual A is said to *envy* individual B if he would rather have B's goods than his own. An allocation is *fair* if no-one envies anyone else, and *superfair* if everyone strictly prefers their own goods.

Unfortunately this theory is of no use in congestion pricing, and we only mention superfairness to dismiss it. Any pricing scheme would be fair by this definition, because if A envies B then A can just start sending traffic with the same distribution as B. We however want the price for a user to reflect the amount of congestion he causes.

5.5.2 The burden test

The idea of a fair price arises in monopoly trials, when a company is charged with cross-subsidising a product it sells in a competitive market by increasing the cost of a different product in which it has a monopoly. One way of testing if there is cross-subsidy is with the burden test, which says that product P constitutes no burden on consumers of other products supplied by the same company, if the total income from P exceeds the extra cost incurred by producing P. (Actually, economists use two closely related tests: the burden test and the incremental cost test. The distinction is not important for our purposes).

Standard economic models of companies and products do not fit very well with the problem of bandwidth allocation, because it is hard to decide what the product is. The fit is, however, good enough to describe the ΔL pricing scheme as fair according to the burden test. The extra cost of carrying a user's traffic is precisely what ΔL charges, so we can say that ΔL fair. (But we shall revise this conclusion in Section 5.6.)

5.5.3 Game theory and fairness

The standard way to apply game theory to fairness is with the idea of a *core*. Suppose that a company supplies products to several consumers. Let the stand-alone cost for a group of those consumers be the cost of supplying only them. Then if any group is being charged more than its stand-alone cost, it has an incentive to withdraw and take its custom elsewhere. The core is the set of allocations and prices where there is no such group, and it is reasonable to call the core fair. There are other closely related definitions of fairness, such as the Shapley value.

These ideas are not appropriate for the problem of bandwidth allocation, because there is no meaningful idea of stand-alone cost. But the inherent idea of social equilibrium is useful. The core expresses the idea that a group of individuals could form a coalition and act in their own interest as a group. In the context of bandwidth allocation, a group of users could band together and transmit their packets through a proxy to make it look as if they all came from a single user. With the pricing scheme ΔL, a group of users who band together (but do not otherwise alter their traffic characteristics) may lower but never increase their net charge.

We would not want a pricing structure that encouraged users to band together and use proxies in this way to hide the characteristics of their traffic,

because that would lead to complicated arrangements and extra traffic to control them. We would therefore describe ΔL as *socially unstable*. Further, if many users banded together then they would constitute a significant proportion of the traffic, and the problem of strategic play described at the end of Section 5.4 would become serious.

These problems in reaching social equilibrium are well-known. In economic systems with external diseconomies (such as congestion, which is a problem for all users) Shapley and Shubik [51] have shown that the core may not coincide with the set of socially desirable outcomes, and in some cases it may not even exist.

5.5.4 Incremental fairness: SPSP

The difficulties about users banding together, and also the problem described in Section 5.4 of whom we should consider to be a user, arise because ΔL is not incrementally fair, in the following sense: Suppose that a user sends some packets in addition to what he sends normally. Then the extra price charged is typically less than if a separate user had sent those additional packets. In other words, increments are not charged a fair price. In this section we introduce another pricing scheme, SPSP, which is *incrementally fair*.

Incremental fairness is closely related to the economic idea of anonymous equity, described by Baumol in the context of stand-alone prices (which are not meaningful in the problem of bandwidth allocation). We can define it in another way though, as a generalisation of the burden test, which says that an individual is not benefiting from cross-subsidisation if the amount he is charged is enough to cover the incremental cost he causes. We may say that a pricing scheme is anonymously equitable if no individual *or part thereof* benefits from cross-subsidisation. In other words, each increment should be charged at least its fair price.

We can now introduce our final fair pricing scheme, called Sample Path Shadow Pricing (or SPSP), first described by Gibbens and Kelly [23]. It works as follows: mark a packet if removing it would result in one less packet being dropped. In other words, when there is an overflow, mark every packet that arrived between the start of the current busy period and that overflow; and when there is more than one overflow in a busy period, mark every packet that arrived between the start of the busy period and the last overflow. It is illustrated in Figure 5.1. Clearly SPSP satisfies the condition of anonymous equity, since it charges each individual packet its incremental cost.

This is not a proposal for a marking algorithm: after all, a packet may have left the queue before we know whether or not it should be marked. So we will simply say that a marking algorithm satisfies the SPSP definition of fairness if it marks the same number of packets from each flow as SPSP.

It is interesting to note that this is precisely the marking scheme described in Example 5.1 on page 56. There it arose as the efficient pricing scheme for Poisson flows using a bufferless resource. So SPSP can lead to an efficient allocation of bandwidth, at least for certain models of user behaviour.

It is not surprising that incremental fairness (SPSP) and fairness (ΔL) disagree. There is an example from superfairness theory, known as the Feldman-Kirman consistency result, which stresses the difference: Starting from an allocation which is fair, a change which is incrementally fair and beneficial to all

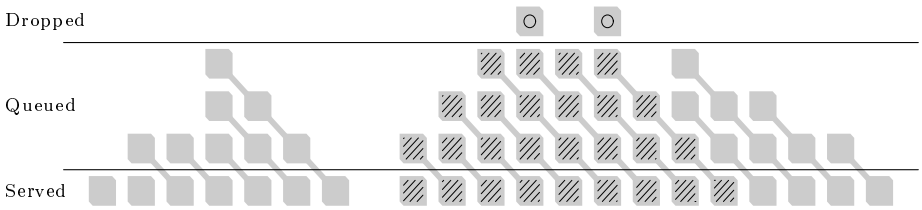

Figure 5.1: Sample path shadow price marking. The squares represent packets, and the grey diagonal lines indicate the progress of a packet through the queue. Shaded packets are those that would be marked by SPSP. This rule marks each packet whose removal would result in one less packet being dropped.

parties may result in an allocation which is unfair to all parties.

5.5.5 Summary of economics and fairness

The two most important ideas in this section are *fairness according to the burden test* and *incremental fairness*. The burden test says that it is fair to charge a user the extra cost of carrying his traffic, which is precisely what ΔL specifies. Incremental fairness says that each individual packet should be charged its fair price (according to the burden test), and this is what SPSP specifies. In addition to these two we have the *full allocation of costs* definition of fairness, described in Section 5.3, which suggests charging according to EB.

In the next section we compare these three definitions and explain how they relate.

5.6 Different definitions of fairness

So far we have seen three different definitions of fairness in marking: EB, ΔL, and SPSP. Each can lead to an efficient allocation of bandwidth, with an appropriate model for user behaviour. The situation is however not as confusing as it might seem. In this section we will explain why the three definitions differ, and why SPSP seems to be the most appropriate definition for marking algorithms for routers.

Even if we decide to allow all three definitions of fairness, it is still possible to point out what is unfair, since the three definitions agree for certain traffic mixes. We call these traffic mixes *anonymous scenarios*, and we will describe them in this section. Zajac suggests ten fairness maxims for aggrieved persons, the first of which is 'frame your initiative as a concrete unfairness issue'. We will use anonymous scenarios heavily in Section 5.7, in pointing out how various proposed marking algorithms can be unfair.

5.6.1 The different definitions

Recall the three definitions of fairness: EB, ΔL, and SPSP.

- EB *says that flows should be marked in proportion to their effective bandwidth* $\hat{t}\alpha(\hat{\theta}, \hat{t})$. *This is fair in that it achieves a full allocation of*

costs, and efficient for the user model mentioned in Section 5.3.3.

- ΔL *says that flows should be marked according to the number of extra drops they cause,* $L(Y) - L(Y - X)$*. This is fair according to the burden test, and efficient for the user model of Section 5.4.3.*
- SPSP *says that a packet should be marked if removing it would lead to one less drop. This is incrementally fair, and efficient for the user model of Example 5.1 in Section 5.4.2.*

These three definitions are different. First, EB is different to ΔL because effective bandwidth is additive over independent flows, so EB would mark the aggregate of two independent flows according to the sum of their individual marks, while ΔL would typically give the aggregate fewer marks. Second, SPSP marks every packet that arrives in the critical period before overflow, and expression (5.1) shows that this is related to the derivative of the effective bandwidth, which is typically not in proportion to the effective bandwidth. Finally, ΔL gives fewer marks than SPSP, for example when a single packet is dropped and some flow contributed two packets in the busy period leading up to the drop.

5.6.2 Anonymity

For a range of traffic mixes these three definitions agree, giving a single clear-cut standard of fairness. While the range is very limited, it is broad enough to show that certain algorithms like RED fail the standard. We call these traffic mixes *anonymous*. We will first define anonymity in terms of effective bandwidth, which is how we will use it in Section 5.7, then give the more natural interpretation in terms of packets.

Anonymity is based on the requirement that at the critical point each flow $\mathbf{X}$ looks as if it is made up of a number of independent copies of some base flow $\mathbf{P}$. Specifically, call a traffic mix *anonymous* if for each flow $\mathbf{X}$ there is a multiple k_x such that the effective bandwidth satisfies $\alpha_X(\hat{\theta}, \hat{t}) = k_x \alpha_P(\hat{\theta}, \hat{t})$ and the most likely path to overflow satisfies $\hat{\mathbf{x}}(0, t] = k_x \hat{\mathbf{p}}(0, t]$, where $(\hat{\theta}, \hat{t})$ is the critical point. One might think of $\mathbf{P}$ as a Poisson flow of very low rate, representing an isolated packet. Since EB marks in proportion to effective bandwidth, and SPSP marks each copy of the $\hat{\mathbf{p}}$ sample path identically, these two definitions of fairness agree.

Now we interpret this definition in terms of packets. Think of $\mathbf{P}$ as representing an isolated packet. At the critical point, i.e. in the busy period leading up to overflow, each aggregate flow $\mathbf{X}$ looks as if it is made up of independent copies of $\mathbf{P}$, i.e. of independent packets belonging to different users. This gives a more natural way of expressing the assumption of anonymity: that all packets arriving in the critical interval leading up to overflow are independent. This means that ΔL marks them all, and so agrees with SPSP. No two packets belong to the same user, so there is no point classifying them, which is why we call this scenario *anonymous*.

Another way of understanding anonymity is through the *formal principle of distributive justice*: that equal cases should be treated equally, and unequals unequally, in proportion to relevant similarities and differences. This is very vague. But in anonymous scenarios, when each user is indistinguishable from an aggregate of independent copies of a base flow, it is clear what the equal cases and the relevant differences are.

5.6.3 SPSP is best

Traffic mixes will rarely be anonymous, and the three definitions of fairness will rarely agree. One way to cope with this would be to recognise that it is technologically difficult to classify packets according to which flow they belong to (at least in very high speed backbone routers), decide that since we cannot classify packets we should just act as though the traffic mix were anonymous, and be satisfied with any algorithm which is fair in anonymous scenarios.

We propose instead a different way of looking at the results of Sections 5.3–5.5 which suggests that SPSP is the right thing to do even when the traffic mix is not anonymous.

First an analogy. I am sharing a cake (which represents capacity-when-there-is-congestion) with several people. The others insist on having a certain size piece which leaves me with half, which is what I want, though I am very prepared to take less if necessary. Now if someone else were to come along, the others would insist on keeping their share, but I would give up some of my share. Should I be charged for taking half? Or should I be given a small discount, to reflect the fact that I will be more flexible than the others if the situation changes?

The first approach is taken by SPSP, and the second by EB and ΔL. Indeed, Gibbens and Kelly [23] introduced SPSP for the very reason that it is the straightforward measure of resource usage. Given a packet trace, we can easily work out which packets used the resource when it was limited—they are exactly the packets that SPSP marks.

How EB differs from SPSP

Marking according to EB tries to achieve something different. The whole idea of effective bandwidths is to capture what happens when the system changes: we say that two flows have the same effective bandwidth if replacing one by the other does not *change* the loss probability. This is the right thing to study for the purposes of controlling admission to the network, but it is not the same as measuring resource usage.

However, the effective bandwidth theory of Section 3.6 tells us about resource usage as well. It identifies the critical timescale $\hat{t}$, and hence the limited capacity $B + C\hat{t}$ available over that timescale, such that the probability of overflow is governed by the likelihood that the sources will consume that limited capacity. When overflow does occur, expression (5.1) gives us $\hat{x}(0, \hat{t}]$, which is the amount of limited capacity consumed by source X. We can suggestively rewrite that expression as

$$\hat{t}\alpha_X(\hat{\theta}, \hat{t}) = \hat{x}(0, \hat{t}] - \hat{\theta}\hat{t}\frac{\partial}{\partial\hat{\theta}}\alpha_X(\hat{\theta}, \hat{t}). \tag{5.6}$$

In words, the effective bandwidth measures the amount of limited capacity consumed by a source, less a derivative term indicating how that source behaves when the system changes.

In Section 3.6 we showed that loss probability is not changed when one flow is replaced by another of the same effective bandwidth. The same equations can tell us what happens to resource usage when this replacement is made. In (3.9), a fraction δ of the sources are replaced by constant rate sources of rate

equal to the effective bandwidth of the sources they are replacing. The optimal θ does change, by $O(\delta)$, but because the loss rate involves a supremum over θ it only changes by $O(\delta^2)$, and so the derivative of the loss rate $I'(0)$ is zero. Nonetheless, since the optimal θ changes by $O(\delta)$, the allocation of the limited resource $B + C\hat{t}$ does change by a nontrivial amount.

The fact that loss probability is not changed by this substitution makes effective bandwidth the appropriate measure in certain circumstances. For example, in admission control the aim is to only accept a call if doing so would not increase the loss probability above a certain threshold. Courcoubetis et al. [9] show how this leads to charging according to effective bandwidth. But if we are only interested in measuring resource consumption, we should charge according to $\hat{x}(0, \hat{t}]$ instead.

How ΔL differs from SPSP

The differences between ΔL and SPSP also arise from whether we take into account how a user would respond to small changes. In our economic model, if the system changes then users can change their behaviour too, potentially reshaping their traffic or changing the amount they send, according to their utility functions. The shadow pricing scheme ΔL charges them so that they have the right incentives to reshape their traffic in a way that fits in which the social optimum. Like EB, ΔL considers what would happen if the system were to change slightly, and it charges accordingly. We can write the ΔL charge as

$$\mathbb{E}L(Y) - \mathbb{E}L(Y - X) = \mathbb{E}A1_{D>0} - \mathbb{E}(A - D)^+$$

where A is the number of packets belonging to X that arrive in the critical interval and D is the number of packets dropped. Again, the first term $A1_{D>0}$ is the sample path shadow price, and the last term concerns reaction on the part of the user: if $A > D$ then there is no point reacting as much as if $A \leq D$.

(The difference between EB and ΔL is in their assumptions about what will happen when the system changes slightly. The former assumes that the traffic will not change but the critical point will shift slightly, whereas the latter assumes that users will reshape their traffic flows.)

When EB, ΔL and SPSP agree

As we have already noted, if all the packets arriving in the interval leading up to overflow belong to different users, i.e. there is some worth attached to each individual packet and they are sent independently, then the three definitions of fairness agree. This is because there is only limited scope for reshaping (you either send the packet or you do not), and so the flexibility term does not come into the price.

It is worth noting another case where they agree: when the queue is overloaded. In terms of effective bandwidths, suppose that the mean input rate is very close to the service rate. This means that the optimal spacescale $\hat{\theta}$ will be very small, and so the second term in (5.6) will be small and SPSP and EB will roughly agree. In terms of economics, suppose that the queue is overloaded in that each user only sends a small number of packets compared to the total number dropped. This means that removing the n packets belonging to a single user would result in n fewer packets being dropped, and so SPSP and ΔL agree.

This case of overloading is akin to the cake analogy in the situation where there is not enough cake to even meet everyone's minimum demand, so flexibility does not come into the price.

5.6.4 Summary of the different definitions

In this section we have described how and why the three measures of fairness differ. In anonymous scenarios they agree, and so there is a clear-cut standard of fairness. In other scenarios, they differ because they are trying to measure different things: SPSP purely measures use-when-there-is-congestion, while EB and ΔL also take into account how the user might react and how elastic the demand is.

A user's reaction will depend on what he wants and what he is prepared to do, and routers are badly placed to predict this. There is no single right user model, and any algorithm that predicts how users react will eventually be mistaken. We therefore suggest that SPSP is the best way to define fairness for routers.

Deciding on efficiency is rather harder. Marking according to each of the three definitions can lead to an efficient allocation for an appropriate user model, and indeed it is impossible to define efficiency without modelling user behaviour. So we shall content ourselves with having found a definition of fair marking.

Unfortunately the implementation of SPSP would require predicting the future behaviour of the queue, since it is often unclear whether a packet should be marked until after it has left the queue. In the next section we look at algorithms for marking, and see how well they approximate SPSP.

5.7 Marking algorithms

In this section we will use the economical and mathematical insights we have found in the first part of this chapter to design and analyse marking algorithms. The goal will be to mark fairly. The best of our three definitions of fairness is SPSP. But we will point out unfairness in anonymous scenarios, when all of them agree. We are all sensitive to being treated *unfairly*, even when we have no definitive idea of what *fair* means!

We will illustrate the two main fairness pitfalls, then go on to show how RED falls into both of them. Other algorithms we analyse include BLUE [20], FRED [35] and Adaptive RED [21]. There are some simple modifications to RED which make it perfectly fair in anonymous scenarios and approximately fair in others, and we summarize them in a new marking strategy we call ROSE.

The main mathematical idea in analysing these algorithms is that of the most likely sample path. Suppose $\mathbf{X}$ is a random input to a queue, and that a rare event occurs. Then, in the many sources large deviations limit, the most likely way for this to happen is if $\mathbf{X}$ had sample path $\hat{\mathbf{x}}$ given in Theorem 3.12: and this path is exponentially more likely than any other. We will calculate and plot examples of these paths. See Chapter 3 and especially Examples 3.8 and 3.9 for details of the theory.

5.7.1 Mark After Loss

The ideal marking algorithm SPSP is impossible to implement, as it requires knowledge of future events. It is easy to describe what it would do, though: in every busy period containing an overflow, mark every packet that arrives between the beginning and the last overflow of that busy period. The problem is that when a packet arrives at a router, we do not know if the queue will overflow before it next idles.

To get around the problem, Gibbens and Kelly [23] suggest the following marking algorithm. When the buffer overflows, mark everything inside the buffer. Also keep track of how many packets should be marked according to SPSP, and continue marking after the overflow so that in total the right number of packets are marked. We will call this algorithm MAL.

They also suggest an even simpler approach, which is to mark all packets leaving the queue from the time of packet loss until the queue becomes empty. This is essentially very similar to the BLUE algorithm designed by Feng et al. [20] which marks packets with a probability which is incremented whenever the queue overflows and decremented when it idles. This mechanism was actually designed with a very different goal to MAL—BLUE's goal is to smooth out the flow of marks, not to approximate SPSP—but in the many sources large deviations limit this goal is not apparent, and BLUE simply amounts to marking a fixed proportion of those packets that arrive after a queue overflows and before it next idles.

(In our slotted time model, it is not clear whether we should mark packets that arrive in the timeslot in which overflow occurs or in the one after. The problem is that real routers operate in continuous time, or at least as close as their timing circuits allow. In fact, at this level of detail they do not even behave entirely like queues. It is interesting to consider how accurate the slotted time queueing model is, but hardly appropriate here. We will assume for simplicity that work arrives evenly distributed throughout a timeslot, and that the marking algorithm parameters are updated at the end of a timeslot.)

The problem with these algorithms is that they close the stable doors after the horse has bolted, and then blame the horses left inside for running away! The packets that arrived before overflow are the ones that caused the problem, while the packets that arrive after are innocent. Hopefully there will be enough of the guilty packets left in the buffer when the queue overflows, and not too many innocent packets marked afterwards, for MAL not to be too bad. But if for example the buffer is small and the most likely time to overflow is large, then most of the guilty packets will have escaped.

The problem of marking innocent packets is illustrated in Figure 5.2, which shows most likely path to overflow and indicates which packets are marked by SPSP, MAL and BLUE. There are two traffic flows: one (the darker) which produces an independent Normal amount of work each timestep, with mean 0.114 and variance 0.161; and another (the lighter) which is periodic and produces 2 units of work every 20 timesteps. The queue has service rate 0.34 and buffer size 1.683. This leads to critical spacescale $\hat{t} = 1.4$ and critical timescale $\hat{t} = 5$. These parameters were chosen so as to make the scenario anonymous.

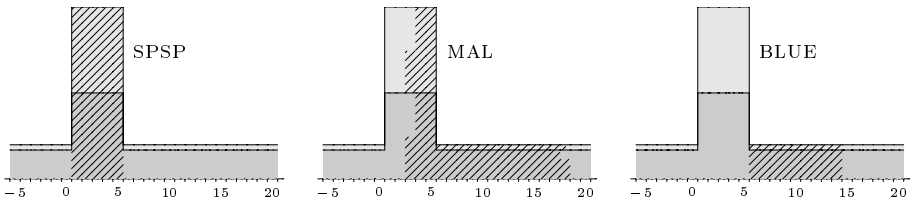

Figure 5.2: Sample paths and marks. The graphs plot the most likely path to lead to overflow, giving the amount of incoming work at each timestep, for a queue fed by two different traffic flows. The shaded regions indicate how the different algorithms would mark. The scenario is anonymous, so SPSP is perfectly fair, and it marks the two flows equally. With MAL the darker flow gets 67% of the marks, and BLUE does even worse, giving it 85%.

5.7.2 Mark in Virtual Queue

Gibbens and Kelly [23] suggest the following virtual queue algorithm, which tries to detect congestion before it becomes a problem, and thereby avoid the problem of marking after a loss occurs. The algorithm runs a *virtual queue* of smaller buffer size and service rate in addition to the real queue, and feeds it a copy of each incoming packet. Specifically, if the real queue has buffer size B and service rate C then let the virtual queue have buffer size κB and service rate κC. From when the virtual queue overflows until it idles, mark all arriving packets. The idea is that the virtual queue will overflow before the real queue, and so the packets that cause overflow in the real queue might be marked. It is appealing because it leaves some space in reserve for bursty flows.

The virtual queue algorithm starts marking after an overflow (in the virtual queue), so it suffers from the same problem as MAL and BLUE. But there is another problem which we wish to highlight, and to do this we will consider an idealized version: instead of marking after the virtual queue overflows, we will suppose that packets are marked in the virtual queue according to SPSP, even though it is impossible to implement. Call this VIRTQ.

Even VIRTQ can still be unfair. This is because the critical point for overflow in the virtual queue is not the same as the critical point for overflow in the real queue. Therefore the most likely path to lead to marking by VIRTQ will be different to the most likely path to lead to overflow in the real queue. Since VIRTQ and SPSP allocate marks in proportion to how much work each flow contributes in these different paths, the two algorithms will mark flows in different proportions. And since SPSP is fair, VIRTQ must be unfair.

The two algorithms will even have different marking frequencies: the virtual queue is more likely to overflow than the real queue, so VIRTQ generates more marks that SPSP. However, this is not so clearly an issue of unfairness.

We illustrate this problem in Figure 5.3 with the same anonymous scenario as in the last section. The mean arrival rate is 0.134 and the real service rate is 0.34, and to stress the problem we will set $\kappa = 0.7$, giving a virtual service rate of 0.238. The (darker) Gaussian source has a much higher mean rate than the (lighter) periodic source, but a smaller variance. In order for the real queue to overflow, both flows have to put on a burst, and the high variance source

will put on a bigger burst. So flows should be marked roughly in proportion to their variances. The virtual queue has a lower service rate, so a small burst in addition to the mean rate is sufficient to make it overflow. So in the virtual queue, mean rates are more important in determining marking ratios.

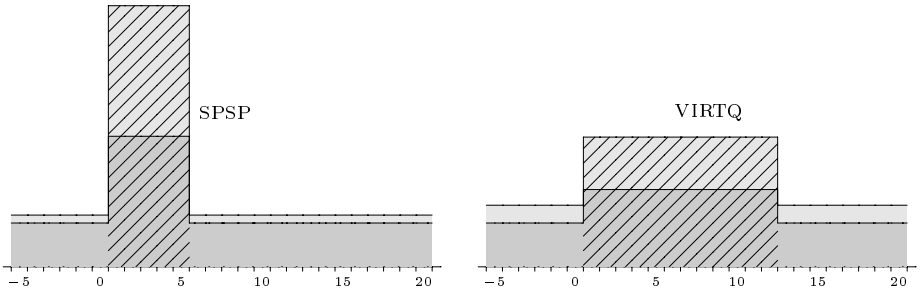

Figure 5.3: Virtual queues. The left graph shows the most likely path to lead to overflow in the real queue: it plots the amount of incoming work at each timestep. The shading indicates the marks that SPSP would give. The right graph shows the most likely path to lead to overflow in the virtual queue, and the marks that VIRTQ would give. The scenario is anonymous, so SPSP is perfectly fair, and it marks the two flows equally; but VIRTQ gives the darker source 60% of the marks. The problem is that overflow occurs in essentially different ways, so the behaviour of the virtual queue is not a good indication of the behaviour of the real queue.

This is a more subtle problem than that described in Section 5.7.1; and while it is possible to construct scenarios in which it marks totally the wrong flow, it does reasonably well in many cases where κ is close to one.

5.7.3 Random Early Detect

We now consider the Random Early Detect (RED) algorithm. Actually, for convenience, we will look at a version of RED in a slotted time model where all packets are the same size. It may be described as follows. Keep track of the exponentially weighted queue size, $\bar{q}_t = \omega q_t + (1-\omega)\bar{q}_{t-1}$. When this is between a threshold b and the buffer size B, mark arriving packets with a probability which is an increasing piecewise linear function of $\bar{q}_t$.

The real algorithm has a mechanism to ensure that marks are allocated regularly, but for large deviations neither this nor the form of the piecewise linear function matter. Recall that the large deviations limiting regime has the number of sources and the capacity of the resource increasing, and this leads to the probability of overflow decaying to 0. In fact, the probability of reaching level $b+\varepsilon$ conditional on reaching level b decays to 0 exponentially in the size of the system. So while the probability of marking may increase linearly in $\bar{q}_t$, the likelihood of reaching that level decays much faster. So we will only look at paths leading up to $\bar{q}_t = b$, and assume that when this happens packets arriving in the next timestep are marked independently and randomly. Thereafter the queue size decreases.

We do not mean to say that the increasing linear function is not important. We merely claim that it is not as important as ω or b. In this particular limiting

regime only ω and b matter, but real life systems are not arbitrarily large and the other parameters will come into play.

Typical behaviour

Assume that the most likely path to lead to marking leaves the queue empty up to time 0, that in $(0, t]$ the queue does not idle, and that at t there are marks. This assumption is valid for certain sources with positive correlations, such as fractional Brownian motion with $H > \frac{1}{2}$. We will restrict attention to Gaussian sources, to make the calculations easier. The average queue size at time t when the input is $\mathbf{x}$ is given by

$$\bar{q}_t(\mathbf{x}) = \mathbf{w}^\mathsf{T}(\mathbf{x} - C\mathbf{1})$$

where $w_s = 1 - (1-\omega)^{t+1-s}$. It is easy to find the most likely path to marking now: we simply solve

$$\inf_{\mathbf{x}:\bar{q}_t(\mathbf{x})=b} \sup_{\boldsymbol{\theta}} \boldsymbol{\theta}\cdot\mathbf{x} - (\lambda\mathbf{1}\cdot\boldsymbol{\theta} + \tfrac{1}{2}\boldsymbol{\theta}^\mathsf{T}\Gamma\boldsymbol{\theta})$$

which is attained at

$$\hat{\mathbf{x}} = \lambda\mathbf{1} + (b + (C-\lambda)\mathbf{1}^\mathsf{T}\mathbf{w})\frac{\Gamma\mathbf{w}}{\mathbf{w}^\mathsf{T}\mathbf{w}}.$$

Marking happens at critical point $\phi\mathbf{w}$ rather than at $\theta\mathbf{1}$. It happens in this way: the average queue size just reaches b at time t, some packets are marked, and in the very next timestep the average queue size decreases again. So RED marks a fixed proportion of the packets that arrive at time t.

The behaviour of RED is illustrated in Figure 5.4. We could have chosen the same anonymous scenario as in the two previous sections, but calculating the most likely path to lead to marking is difficult for non-Gaussian sources, so instead we consider the following non-anonymous scenario. A queue of service rate 0.6 and buffer size 1 serves two traffic flows. One (the darker) sends work according to a fractional Brownian motion with mean rate 0.3, variance 0.1 and Hurst parameter 0.7. (See Example 2.4 on page 14 for details.) The other flow (the lighter) sends an independent amount of work each timestep, normally distributed with mean 0.1 and variance 1. The RED parameters are $\omega = 0.1$ and $b = 0.5$. This ω is much larger than is advised by Floyd and Jacobson, but as we shall show it is fairer to make ω large.

The RED algorithm falls down in both the ways we have described so far. First, it only marks packets that arrive after the problem has occurred so it misses the packets that actually caused the overflow. Second, its marking is not representative of overflow, because marking and overflow occur in essentially different ways.

Lin and Morris [35] have described a modified version of RED called FRED which is meant to be fairer. In the large deviations limit it works roughly as follows. When the average queue size $\bar{q}_t$ reaches the threshold b, whereas RED would mark a sample of all arriving packets, FRED only marks or drops packets from flows which have more than their fair share of packets in the queue, where 'fair share' means an equal allocation between all flows of the current average queue size. In the example of Figure 5.4, when RED starts marking at time 13, most of the work in the queue belongs to the darker flow: so FRED would only

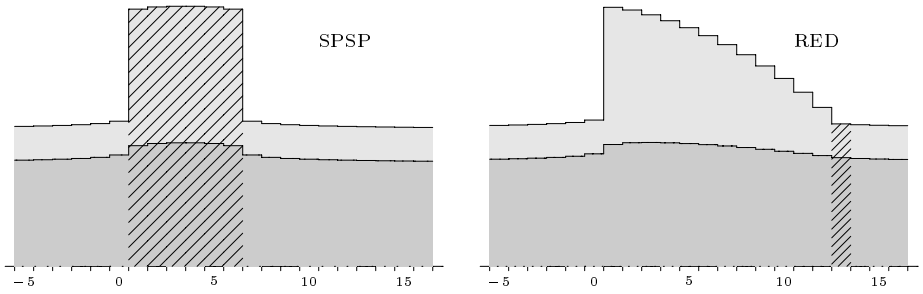

Figure 5.4: How RED marks. The left graph shows the most likely path to lead to overflow: it plots the amount of incoming work at each timestep. The shading indicates the marks that SPSP would give. The right graph shows the most likely path to lead to marking by RED, and indicates how much each flow is likely to be marked. Marking and overflow occur in quite different ways, and anyway, RED starts marking too late to catch the guilty packets. In this example, SPSP would give the darker source 47% of marks, but RED gives it 76%.

drop that flow's packets. In other words, in this example the unfairness of RED has been exacerbated!

Setting RED parameters

It is widely accepted that the RED parameters must be set to match the traffic characteristics. Feng et al. [21] describe one such scheme: they alter the piecewise linear function that determines marking probability, though as we have noted this will not achieve anything in the large deviations limit.

We have developed enough theory now to tell us at least how ω and b should relate. Recall from Section 5.3 that the rate functions for marking and dropping must be equal, if a drop is to be worth a fixed number of marks. The rate function for marking is just a function $I_M = I_M(\omega, b)$, and we can work out how to choose ω and b to keep I_M fixed, or at least we can for a specific traffic mix.

This is illustrated in Figure 5.5, for a queue with service rate 1.5 fed by an first order autoregressive traffic flow with mean rate 1, autoregression coefficient 0.1 and variance 0.5. There is a tradeoff: the larger ω is, the larger b should be. This is hardly surprising, since if the current queue size is given a large weighting we should accept fairly large fluctuations in the average queue size.

If one does not know the traffic mix then it is natural to set ω and b adaptively. For example, one could fix ω and then adjust b adaptively so that on average the right number of packets are marked.

5.7.4 *R*each *O*verload, *S*end *E*CN

The final algorithm we will look at is called ROSE, and we have designed it to address the pitfalls described so far. It is basically a special case of RED with some minor modifications.

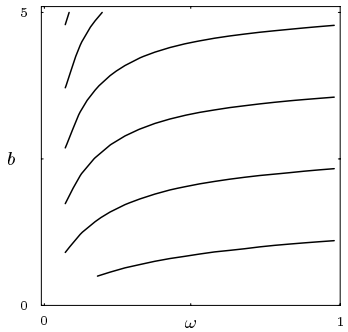

Figure 5.5: How to set some RED parameters. Each line indicates a family of choices of ω and b that lead to the same frequency of marking, for a specific traffic distribution. To change the way the system responds, without changing the value of a mark, ω and b should be changed together along one of these lines.

It is not intended as a concrete proposal. It is simply a demonstration that it is possible to design algorithms which scale properly to large networks and which are fair, at least in anonymous traffic mixes, and approximately fair in many others. There are many such algorithms, and engineering judgement is required in deciding between them. For example, the virtual queue algorithm described in Section 5.7.2 would be fair if the virtual queue scaling factor κ was set adaptively.

The ROSE algorithm

The ROSE algorithm works as follows. Whenever the queue size exceeds a threshold b, mark everything in the queue. Adjust the threshold b as follows. For every packet that would be marked by SPSP, decrease b by $\kappa\epsilon$. For every packet that is marked, increase b by ϵ. Here, ϵ is a fixed small quantity, and κ is a fixed quantity which indicates how many marks correspond to one drop. (As we discussed in Section 5.4.1, the ECN proposal indicates that one drop should be worth one mark. But it may be that the whole network can be made more robust if one mark is only worth a fraction of a drop.)

This is rather like RED with $\omega = 1$, with an adaptive mechanism to set b, and the modification that rather than just marking arriving packets, everything in the queue is marked as well.

The two pitfalls

To see that ROSE addresses the issues raised so far, we need to answer two questions. Does it marks packets that caused overflow, or does it mark innocent packets that arrived later? Does it mark in essentially the same way that overflow occurs—in other words, does marking have the same critical point as overflow? The answer to both of these is Yes.

We will deal with the second point first. At a large deviations level, the adaptive algorithm must settle on a value of b equal to the buffer size. We know this because each drop is worth κ marks, so the rate function for marking is equal to the rate function for overflow, and the only value of b that would achieve this

is $b = B$. This seems at first to be inconsistent with the adaptive mechanism, which would set $b < B$. To explain the apparent inconsistency, recall that large deviations is only concerned with limiting behaviour. This means that while b will actually fluctuate and be a little smaller than B, this difference does not grow as the network grows. This means that the most likely path to exceed the threshold b is just the same as the most likely path to overflow, and therefore the critical point for ROSE is the same as that for overflow. It is now easy to deal with the first point. All the packets that ROSE marks did indeed contribute to overflow, because it marks everything in the buffer when overflow occurs.

Fairness of ROSE

Thus ROSE addresses the problems we have described in the other algorithms. Not only does it address those problems, but it is also perfectly fair in anonymous scenarios, and approximately fair in many others.

First, we need to check that it marks in proportion to congestion. It does indeed mark exactly the number of packets that SPSP marks (or a constant multiple thereof) by construction. Next, we need to see if it marks flows in the correct proportion. We consider anonymous scenarios first, then non-anonymous scenarios.

Recall the effective bandwidth definition of anonymity: that at the critical point, we treat each flow as if it were made up of a certain number of copies of some base flow $\mathbf{P}$. The number of copies of $\mathbf{P}$ that make up a flow $\mathbf{X}$ is proportional to the effective bandwidth of $\mathbf{X}$ at the critical point. Now, since they are identical, each copy of the base flow will leave the same amount of work in the queue at the time of overflow. This means that, under anonymity, we can treat the amount of work belonging to $\mathbf{X}$ caught in the queue at the time of overflow as proportional to the effective bandwidth of $\mathbf{X}$. In other words, under anonymity, ROSE is fair.

When the traffic is not anonymous, we have to ask if ROSE agrees with SPSP. By construction it marks the same number of packets in total. But it does not always mark flows in the same proportions as SPSP, and one can construct examples where it does arbitrarily badly by choosing sources with peculiar paths to overflow. However, it will agree whenever the sample paths are such that the contents of the buffer at overflow are representative of the work that arrived during the critical congestion interval. This will often be approximately true, and there is an important class of scenarios where it is precisely true: large buffer asymptotics.

The large buffer asymptotic was described in Example 3.2. It refers to the limiting regime in which the sources and the service rate are fixed and the buffer size grows. It used to be a standard tool for estimating overflow probability; it has since been superseded by the many sources asymptotic, but it is still a good approximation for queues with large buffers. Importantly for us, it has the property that the most likely sample paths to overflow are constant rate—this is called having linear geodesics. This means that buffer contents at the time of overflow precisely reflect the arrival rates of the different flows during the critical time period, and so ROSE marks flows in the same proportion as SPSP.

5.7.5 Summary of marking algorithms

Strictly speaking, all we have found is a collection of negative results. We have several different definitions of fairness, which agree only in certain circumstances, so while we can decide if one algorithm is unfair we cannot firmly say that another is fair.

And large deviations too only allows us to find negative results. Large deviations is a good tool for modelling certain sorts of networks, in which there are many independent users and correspondingly large amounts of resources and in which overflow is rare. All we can decide with our analysis is whether an algorithm is unfair in this regime.

These tests are enough to decide that most of the algorithms that have been proposed are unfair. In fact, only a small class of algorithms (including ROSE) pass them both. We expect that studying the characteristics of this class will be of considerable help in designing better marking algorithms.

5.8 Frequently Asked Questions

A FAQ is a frequently asked (or answered) question, and a list of FAQs and their answers is the canonical form of Internet document for collecting and storing information on a given topic. In that spirit, we compare our findings to previous work by listing FAQs.

What modelling assumptions do you make?
We make no assumptions about the nature of the sources, except for some very minor mathematical restrictions which will be satisfied by most sources that average out in the long run, including bursty sources like fractional Brownian motion. Most importantly, we do not assume that the sources use TCP. Our definition of fairness makes no modelling assumptions at all. The large deviations analysis of marking algorithms assumes that the system is large, with many independent flows.

To avoid bias against bursty sources, should not the marking algorithm use a weighted average, as RED does?
There are two ideas behind this claim, and they are both wrong. The first is that sources should be marked in proportion to their mean rates, and weighted averaging is needed to achieve this. But it is not the mean rate that causes queue overflow, rather it is the bursts; and so the marking algorithm ought to penalise bursts. The second idea is that short-term fluctuations in bursty traffic which do not cause overflow should be accommodated, and the way to achieve this is to use a weighted average. But there are other ways to achieve this, for example by increasing the marking threshold b when the traffic is bursty, as ROSE does.

Since these algorithms mark everyone, will they not lead to synchronization and instability?
Many of the algorithms we have suggested mark a group of successive packets. If the users to whom these packets belong all respond at the same time by reducing their rate, there might be a much larger decrease in aggregate rate than is necessary, followed by a collective increase in rate, and so on. This is

called synchronization, and it makes the network see-saw unstably. But the general issue of stability is much more complicated than this, and so far there are only preliminary results. Tan [53] gives cases in which, with reasonable user behaviour, algorithms similar to SPSP are stable. The issue here is that stability depends on how users behave. If they are reasonable, and do not respond to marks too suddenly, any decent marking scheme should be stable. If they are perverse, any marking scheme can be unstable.

How does ROSE scale?
The large deviations underpinning these arguments are *designed* to work in large networks, and indeed the larger the network the better the approximation. It is in small networks that the approximations may break down.

Are there simulation results to support your claims?
We are proposing not merely an improved mechanism but a better *definition* of fairness, so it would be premature to report simulation results. There are ongoing experiments [5, 30, 58] to see how users might respond if faced with fair marking, and anyone with access to the Internet can take part.

How do you make marking fair for users with long round trip times?
This question is based on what we call a social idea of fairness. This says that certain classes of users, such as those who cannot respond quickly because of long delays, or even those from troubled social backgrounds, ought to receive fewer marks because they are less able to compete or deserve more bandwidth. Our definition, which might be called technical fairness, says that users should be marked in proportion to the impact they have. The issue of social fairness is a genuine one, but routers are absolutely the wrong part of the network to deal with it.

How do you account for the fact that the number of marks given can be wildly different from the number of drops?
To make the objection concrete, we give an example due to Kelly. Suppose there are two routers: router A is fed by smooth traffic flows, so a small increase in traffic causes a large increase in loss; and router B is fed by fluctuating flows, so a small increase in traffic does not cause such a large increase in loss. Then it is reasonable to run A at a lower loss rate than B, for example if the goal is to minimize loss rate per unit throughput. Marking according to SPSP would encourage this, because A would have a critical timescale that is longer than that for B, and so more marks would be generated at A; whereas marking in proportion to loss would mean that A generates fewer marks than B. In general, marks reflect marginal costs (and thus how users should respond) rather than average costs (which are only relevant to the router).

How does your definition of fairness compare to max-min fairness?
The idea of max-min fairness can be traced back to Rawls, and further. He proposed that social and economic inequalities be arranged to the greatest benefit of the least advantaged. It is easy to say what this means when considering a simple allocation of capacity subject to a constraint on the total, and assuming that benefit is measured simply by mean bandwidth: everyone should be allocated the same bandwidth. But it is unclear how to extend it to incorpo-

rate demand for different services, and to cope with random traffic flows—the objects of study for this thesis—where the idea of mean bandwidth is not very relevant. The algorithms we have suggested owe more to proportional fairness, described by Gibbens and Kelly [23].

How do you enforce responsiveness from unresponsive flows?
Some router algorithms have been designed to drop packets from flows that do not respond to marking, or even from flows that do not respond as quickly as TCP does. It is hard to see what else can be done in the Internet today. The problem with this is that it does not take account of different preferences: some users might want to pay more so that they do not have to back down, while others would happily take a smaller share of the bandwidth. In a private intranet, users can be expected to cooperate and so marking should be sufficient incentive. In the Internet, the obvious solution would be to charge a user for every marked packet he receives; but this sort of pricing is a long way off. A more workable solution might be for Internet Service Providers to police traffic flows, reducing the rate at which the user can send when he receives very many marked packets. The problem of unresponsiveness should if at all possible be dealt with at the boundary of a network, close to users, and not in the network core. See the ECN proposal [48] for some more discussion of incentives.

How could users be encouraged to respond to marks?
Suppose a user is charged for every marked packet he receives. This is appealing, since it fits so well with the economic model of Section 5.5. Internet Service Providers could collect charges from users for marked packets, and could in turn pay upstream network operators according to how many marked packets they receive. There are problems with this, as with all Internet pricing mechanisms around today. For example, sometimes it should be the sender who pays rather than the receiver, such as in viewing advertisements. Some users might also be reluctant to put up with a variable bill, even though most cope well enough with variable telephone and electricity bills. Even if users demanded fixed prices, this could be achieved through intermediaries who take on the risk and charge a premium, just like insurance agents. Key et al. [31] discuss further the use of marks as a pricing mechanism, and MacKie-Mason and Varian [38] discusses usage-based pricing in general.

5.9 Summary

In this chapter we have sought to define what is meant by marking fairly, taking into account the average bandwidth and the burstiness of each traffic flow. We have found several candidate definitions of fairness: SPSP, EB and ΔL, from effective bandwidth theory and economics. They all measure resource usage, but the latter two additionally take into account how the user might behave when the system changes; and they differ because they have different models of how users behave. When the traffic mix is what we call anonymous, the three definitions agree. Otherwise, we choose SPSP as the most useful definition, because it is intrinsically difficult for routers to model user behaviour.

We have used large deviations to model the behaviour of marking algorithms.

We have seen that RED can be unfair, even in anonymous scenarios. We have described a variant, called ROSE, which is fair in anonymous scenarios and approximately fair in many others.

Glossary

Admission control. In order to keep congestion within fixed bounds, some networks can decide whether or not to accept a new flow based on current traffic levels. This is called admission control.

Bandwidth. The bandwidth of a traffic flow is a measure of the rate at which data is transmitted, measured in bits per second.

Congestion collapse. When there is congestion and packets are dropped, end-systems typically retransmit the dropped packets. If they do this too suddenly they cause more congestion, leading to more drops. This vicious circle is called congestion collapse, and it was first noticed in the Internet in October 1986 [24].

Drop. A packet that is discarded inside the network is said to have been dropped. Packets are dropped when they arrive at a router which has no space to store them. See page 1.

ECN. Explicit Congestion Notification. This is a scheme whereby a router can mark packets to indicate that it is experiencing congestion. See page 48.

Effective bandwidth. The effective bandwidth of a random traffic flow is a measure of the impact it has, lying between its mean and peak bandwidth, and measured in the same units. See page 32.

Efficiency. An economic system is said to be Pareto-efficient if there is no change which would simultaneously make someone better off and no-one worse off.

End-system. This refers to any sort of device that can generate and receive Internet traffic. Most end-systems are computers, but the term can also cover telephones, video cameras, and many other appliances.

Externality. An externality is an economic factor that affects your welfare but is under the control of someone else. An external diseconomy is an externality which detracts from your welfare. Pollution is an example of an external diseconomy.

FAQ. Frequently Asked Question. Also, a list of such questions.

Intranet. An intranet is a network which is internal to an organisation but operates in the same way as the global Internet.

LDP. Large Deviations Principle. A particular type of probability estimate. See page 2 for an example and page 7 for a full definition.

Marginal cost. The marginal cost of a good is the cost of producing one extra unit.

Mark. A mark on a packet is an indication that it has passed through a congested router. Marks are set by ECN algorithms such as RED.

Packet. A packet is the basic unit in which data is sent through the Internet. See page 1.

Pigovian tax. Pigovian taxes, proposed at the beginning of this century by the economist Pigou, are taxes on externalities, designed to lead to socially desirable outcomes.

Rate function. A rate function is part of a large deviations principle (LDP). Informally, we say that an event has rate κ if in a system scaled up in a specified way by factor L, the probability of that event is $e^{-\kappa L}$.

RED. Random Early Detect. This is an algorithm which tells ECN-enabled routers how to mark packets. See page 69.

RFC. Request for Comments. This is the democratic name given to notes about the Internet. Specifications of the Internet Engineering Taskforce are published as RFCs. For a full list, see `http://www.rfc-editor.org/`.

Router. A router or switch is a device that routes packets. See page 1.

Streaming traffic. For certain sorts of traffic, such as live audio, packets are transmitted as soon as they are generated. This is called streaming.

Switch. See Router.

TCP. Transmission Control Protocol. This is an algorithm that end-systems can use to control the rate at which they send packets, so as not to cause too much congestion. See page 49.

Bibliography

[1] William J. Baumol. *Superfairness: applications and theory.* MIT Press, 1986.

[2] Arthur W. Berger and Ward Whitt. Effective bandwidths with priorities. *IEEE/ACM Transactions on Networking*, 6(4), August 1998.

[3] Dimitris Bertsimas, Ioannic Ch. Paschalidis, and John N. Tsitsiklis. On the large deviations behaviour of acyclic networks of G/G/1 queues. Tech. Report LIDS-P-2278, MIT Laboratory for Information and Decision Systems, December 1994. URL `http://justice.mit.edu/pubs/2278.html`.

[4] D.D. Botvich and N.G. Duffield. Large deviations, the shape of the loss curve, and economies of scale in large multiplexers. *Queueing Systems*, 20: 293–320, 1995.

[5] Microsoft Research Cambridge. Congestion pricing and a distributed game. Available on the Internet, 1999. URL `http://www.research.microsoft.com/research/network/disgame.html`.

[6] Shaogang Chen and Kihong Park. An architecture for noncooperative QoS provision in many-switch systems. In *Proceedings of IEEE Infocom*, 1999. URL `http://yake.ecn.purdue.edu/~shaogang`.

[7] Gagan L. Choudhury, David M. Lucantoni, and Ward Whitt. On the effectiveness of effective bandwidths for admission control in ATM networks. In *Proceedings of the 14th International Teletraffic Congress — ITC 14*, pages 411–420. Elsevier Science, 1994.

[8] *Cisco IOS Release 12.0, Configuring WRED.* Cisco, 1999. URL `http://www.cisco.com/`.

[9] Costas Courcoubetis, Frank Kelly, and Richard Weber. Measurement-based usage charges in communications networks. Research Report 1997-19, University of Cambridge, Statistical Laboratory, 1997. URL `http://www.statslab.cam.ac.uk/Reports/1997/1997-19.html`.

[10] Costas Courcoubetis, Vasilios A. Siris, and George D. Stamoulis. Application of the many sources asymptotic and effective bandwidth to traffic engineering. URL `http://www.ics.forth.gr/netgroup/publications/`. To appear in Telecommunication Systems, 1999.

[11] Costas Courcoubetis and Richard Weber. Buffer overflow asymptotics for a buffer handling many traffic sources. *Journal of Applied Probability*, 33:886–903, 1996. URL `http://www.statslab.cam.ac.uk/~rrw1/research/nsource2.ps`.

[12] A. de Acosta. Moderate deviations and associated Laplace transformations for sums of independent random vectors. *Transactions of the American Mathematical Society*, 329(1):357–375, January 1992.

[13] G. de Veciana, C. Courcoubetis, and J. Walrand. Decoupling bandwidths for networks: a decomposition approach to resource management for networks. In *Proceedings of IEEE Infocom*, volume 2, pages 466–474, 1994. URL `http://walrandpc.eecs.berkeley.edu/Papers/dec.pdf`.

[14] Amir Dembo and Tim Zajic. Large deviations: From empirical mean and measure to partial sums process. *Stochastic processes and their applications*, 57:191–224, 1995.

[15] Amir Dembo and Ofer Zeitouni. *Large Deviations Techniques and Applications*. Jones and Bartlett, 1993.

[16] Chandrakant M. Deo and Gutti Jogesh Babu. Probabilities of moderate deviations in Banach spaces. *Proceedings of the American Mathematical Society*, 83(2):392–397, October 1981.

[17] N. G. Duffield. Economies of scale in queues with sources having power-law large deviation scalings. *Journal of Applied Probability*, 33:840–857, 1996.

[18] N. G. Duffield and S. Low. The cost of quality in networks of aggregate traffic. In *Proceedings of IEEE Infocom*, 1998. URL `http://www.research.att.com/~duffield/pubs/cost.ps`.

[19] N. G. Duffield and N. O'Connell. Large deviations and overflow probabilities for the general single-server queue, with applications. *Mathematical Proceedings of the Cambridge Philosophical Society*, 118:363–374, 1995.

[20] Wu-chang Feng, Dilip D. Kandlur, Debanjan Saha, and Kang G. Shin. BLUE: a new class of active queue management algorithms. Technical report CSE-TR-387-99, University of Michigan, 1999. URL `http://www.eecs.umich.edu/~wuchang/blue/`.

[21] Wu-chang Feng, Dilip D. Kandlur, Debanjan Saha, and Kang G. Shin. A self-configuring RED gateway. In *Proceedings of IEEE Infocom*, 1999. URL `http://www.eecs.umich.edu/~wuchang/work/infocom99.ps.Z`.

[22] Sally Floyd and Van Jacobson. Random Early Detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, August 1993. URL `http://www.aciri.org/floyd/papers/red/red.html`.

[23] R. J. Gibbens and F. P. Kelly. Resource pricing and the evolution of congestion control. *Automatica*, 35, 1999. URL `http://www.statslab.cam.ac.uk/~frank/evol.html`.

[24] V. Jacobson. Congestion avoidance and control. In *Proceedings of SIGCOMM*. ACM, August 1988. URL `http://www-nrg.ee.lbl.giv/papers/congavoid.pdf`.

[25] John Paul II. Centisimus annus. Encyclical letter, 1991. URL `http://www.vatican.va/holy_father/john_paul_ii/encyclicals/`.

[26] F. P. Kelly. *Reversibility and Stochastic Networks*. Wiley, 1978.

[27] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49:237–252, 1998. URL `http://www.statslab.cam.ac.uk/~frank/rate.html`.

[28] F.P. Kelly. Loss networks. *The Annals of Applied Probability*, 1(3): 319–378, 1991. URL `http://www.statslab.cam.ac.uk/~frank/PAPERS/loss.html`.

[29] Frank Kelly. Notes on effective bandwidths. In F. P. Kelly, S. Zachary, and I. Ziedins, editors, *Stochastic Networks: Theory and Applications*, Royal Statistical Society Lecture Note Series, chapter 8, pages 141—168. Oxford, 1996. URL `http://www.statslab.cam.ac.uk/~frank/eb.html`.

[30] Peter Key and Derek McAuley. Differential pricing and QoS in networks: where flow-control meets game theory. *IEE Proceedings – Software*, 146(2), 1999. URL `http://www.research.microsoft.com/research/network/disgame.html`.

[31] Peter Key, Derek McAuley, Paul Barham, and Koenraad Laevens. Congestion pricing for congestion avoidance. Technical Report MSR-TR-99-15, Microsoft Research Cambridge, 1999. URL `http://www.research.microsoft.com/research/network/disgame.html`.

[32] V. G. Kulkarni, L. Gün, and P. F. Chimento. Effective bandwidth vectors for multiclass traffic multiplexed in a partitioned buffer. *IEEE Journal of Selected Areas in Communications*, 13(6):1039–1047, 1995.

[33] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, 2(1):1–15, 1994.

[34] N. Likhanov and R. R. Mazumdar. Cell loss asymptotics for buffers fed with a large number of independent stationary sources. *Journal of Applied Probability*, 1999. To appear.

[35] Dong Lin and Robert Morris. Dynamics of Random Early Detection. In *Proceedings of SIGCOMM*. ACM, 1997. URL `http://www.acm.org/sigcomm/sigcomm97/papers/p078.html`.

[36] S. H. Low and D. E. Lapsley. Optimization flow control, I: Basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, 1999. URL `http://www.ee.mu.oz.au/staff/slow/research/internet.html`.

[37] J. K. MacKie-Mason and H. R. Varian. Pricing the Internet. In B. Kahin and J. Keller, editors, *Public Access to the Internet*. Prentice-Hall, 1994. URL `http://www.sims.berkeley.edu/~hal/people/hal/papers.html`.

[38] Jeffrey K. MacKie-Mason and Hal R. Varian. Some FAQs about usage-based pricing. Available on the Internet, 1994. URL `http://www.sims.berkeley.edu/~hal/people/hal/papers.html`.

[39] Kurt Majewski. *Large Deviations of Feedforward Queueing Networks*. PhD thesis, Ludwig-Maximilians-Universität München, May 1996.

[40] Michel Mandjes and Ad Ridder. Optimal trajectory to overflow in a queue fed by a large number of sources. *Queueing Systems*, 31:137–170, 1999.

[41] Tom Mountford and Balaji Prabhakar. The Cesaro limit of departures from certain ./G/1 queueing tandems. Technical Report HPL-BRIMS-96-017, Hewlett Packard, 1996.

[42] Neil O'Connell. Queue lengths and departures at single-server resources. In F. P. Kelly, S. Zachary, and I. Ziedins, editors, *Stochastic Networks: Theory and Applications*, chapter 5. Oxford, 1996. URL `ftp://hplose.hpl/hp.com/pub/noc/papers/9604.ps`.

[43] Neil O'Connell. A large deviation principle with queueing applications. Technical Report HPL-BRIMS-97-05, BRIMS, Hewlett Packard Labs, Bristol, March 1997. URL `ftp://hplose.hpl/hp.com/pub/noc/papers/9705.ps`.

[44] Neil O'Connell. Large deviations for departures from a shared buffer. *Journal of Applied Probability*, 34:753–766, 1997. URL `ftp://hplose.hpl/hp.com/pub/noc/papers/9603.ps`.

[45] Neil O'Connell. Large deviations for queue lengths at a multi-buffered resource. *Journal of Applied Probability*, 35:240–245, 1998. URL `ftp://hplose.hpl/hp.com/pub/noc/papers/9610.ps`.

[46] Ioannis Ch. Paschalidis. *Large Deviations in High Speed Communications Networks*. PhD thesis, MIT Laboratory for Information and Decision Systems, Cambridge, MA, USA, May 1996.

[47] A. A. Puhalskii and W. Whitt. Functional large deviation principles for waiting and departure processes. *Probability in the Engineering and Informational Sciences*, pages 479–7507, 1998.

[48] K. Ramakrishnan and S. Floyd. A proposal to add Explicit Congestion Notification (ECN) to IP. RFC 2481, The Internet Society, January 1999. URL `http://www.aciri.org/floyd/papers/rfc2481.txt`.

[49] K. K. Ramakrishnan and Raj Jain. A binary feedback scheme for congestion avoidance in computer networks. *ACM Transactions on Computer Systems*, 8:158–181, 1990.

[50] Amartya Sen. *On Ethics and Economics*. Blackwell, 1987.

[51] Lloyd S. Shapley and Martin Shubik. On the core of an economic system with externalities. *American Economic Review*, 59(4):678–684, September 1969.

[52] A. Simonian and J. Guibert. Large deviations approximation for fluid sources fed by a large number of on/off sources. *IEEE Journal of Selected Areas in Communications*, 13:1017–1027, 1995.

[53] D. K. H. Tan. Rate control and user behaviour in communication networks. In *4th INFORMS Telecommunications Conference*, 1998. URL `http://www.statslab.cam.ac.uk/~dkht2/conf.ps`.

[54] David Tan. *Mathematical models of rate control for communication networks*. PhD thesis, University of Cambridge, 1999.

[55] William Thomson and Hal R. Varian. Theories of justice based on symmetry. In Leonid Hurwicz, David Schmeidler, and Hugo Sonnenschein, editors, *Social goals and social organization*. Cambridge University Press, 1985.

[56] Hal R. Varian. *Microeconomic Analysis*. Norton, third edition edition, 1992.

[57] A. Weiss. A new technique for analyzing large traffic systems. *Advances in Applied Probability*, 18:506–532, 1986.

[58] Damon Wischik. Pricing the Internet: an experiment. Available on the Internet, 1998. URL `http://www.statslab.cam.ac.uk/~djw1005/Compete/`.

[59] Damon Wischik. The output of a switch, or, effective bandwidths for networks. URL `http://www.statslab.cam.ac.uk/~djw1005/Stats/Research/output.html`. To appear in Queueing Systems, 1999.

[60] Damon Wischik. Sample path large deviations for queues with many inputs. URL `http://www.statslab.cam.ac.uk/~djw1005/Stats/Research/sampleldp.html`. Submitted to Annals of Applied Probability, 1999.

[61] Edward E. Zajac. *Political economy of fairness*. MIT Press, 1995.