

How to mark fairly

Damon Wischik, Statistical Laboratory, Cambridge

Final Draft—January 2001

Abstract

A network router that marks packets to signal congestion should do so fairly. We propose a definition of fairness, using ideas from effective bandwidth theory and economics. Our definition measures both the level of congestion at a router and the contribution of each flow to that congestion, taking into account the flow's burstiness. We then use large deviations to analyse marking algorithms such as RED, point out how they can be unfair, and suggest how they can be made fairer.

1 Introduction

This paper attempts to bring together three strands of research: engineering mechanisms for end-to-end congestion control in the Internet, economic study of pricing structures for networks, and the mathematical theory of queueing and effective bandwidth.

1.1 Engineering mechanisms. 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 the sending rate is lower than necessary, it is cautiously increased. The algorithm was designed in 1988 by Jacobson [10] 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. 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 [21] which proposes a scheme called Explicit Congestion Notification (ECN). In this scheme 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 a router should use. One such algorithm which has achieved much attention (and even been implemented in commercial routers [4]) is the RED algorithm by Floyd and Jacobson [8].

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. Any new marking scheme must therefore cope with a wide range of types of traffic.

1.2 Economic study. 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. 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 limited capacity.

Economists have paid a good deal of attention to the problem of how limited network resources should be divided between competing users with differing requirements. An influential early paper by MacKie-Mason and Varian [18] proposes a market-based model 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 [17], 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 [3] 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.

These approaches assume that traffic is parameterized solely by its mean rate. By contrast, Courcoubetis et al. [5] 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 been proposed by Gibbens and Kelly [9]. This paper follows on from their work, which we describe in more detail in the following sections.

1.3 Queueing theory. It is clear that before we can understand how to charge for congestion, we must understand the *nature* of congestion. It is natural to measure the level of congestion by the frequency of dropped packets, but it is much harder to measure traffic levels and to relate them to congestion. The problem is that it is *bursts* of traffic that fill up buffers and give rise to drops—and there is no generally accepted way to measure burstiness.

One could measure the mean rate of a traffic flow and simply assume this

is proportional to burstiness; one could assume that traffic of a certain type, say video traffic, has a typical burstiness that can be measured by simulation; or one could allow the bursts in a traffic flow to have an arbitrary distribution. We study these notions of burstiness, and the charging schemes they lead to, in Sections 3–5.

Such different ideas about how to quantify traffic inevitably lead to different ideas about how queues build up, and hence to different ideas about how to charge for congestion. In Section 6 we consider these marking schemes from the perspective of fairness. Perhaps surprisingly, we are led to conclude that there is a single charging scheme which is the most appropriate to take as a model for marking algorithms in routers.

Having explained what a marking algorithm ought to achieve, we go on in Section 7 to analyse the RED algorithm designed by Floyd and Jacobson [8], using the mathematical theory of large deviations to calculate its typical behaviour. It turns out that a few simple changes to RED can make it much fairer, and we summarize them in a new algorithm called ROSE.

2 The goals of marking

Most of this paper 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 [22] to John Paul II [11]. 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.

2.1 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 [26] has much to say about efficiency and nothing at all about fairness (though Varian himself has made many contributions to the no-envy theory of fairness [25]).

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 [29], 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 paper 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 notions of burstiness. From these three definitions we

will distill a single notion of fairness, but it does not seem possible to do the same for efficiency.

2.2 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 [16] 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 paper 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 [21] 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.

2.3 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 problem of congestion is largely attributable to 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 burstiness.

Another aspect of marking which has received only a little attention [12] 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.*

As we have noted, the hard part is finding the right measure of how much capacity a flow uses.

2.4 Dropping, marking, and charging. Before we continue, a note on marking and charging. We will mainly refer to charging rather than marking, so it is worth making explicit 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 thousandth of a penny 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 8.

A user's response to marks will be governed by what the marks signify. The ECN proposal [21] 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 to 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 most of this paper, 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 should incur charge P , of whose packets L are dropped, need only have $P - L$ of his remaining packets marked.

3 Effective bandwidths and marking

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. [5]. First we will recall the theory, which is described more fully by Kelly [13] and Wischik [28]. For the purposes of fairness and efficiency, what matters is the following summary.

3.1 Effective bandwidth theory. Informally, the effective bandwidth of a random traffic flow is a measure of the bandwidth it consumes, somewhere between the mean and peak rates, encoding all important information about the burstiness. The neat feature is that the loss probability at a queue is governed by the sum of the effective bandwidths of the input flows. This means that if two flows have the same effective bandwidth they have the same impact on loss probability, which suggests that they should be marked equally.

To make this more precise, and to describe the analytical tools which will be used extensively in Section 7, we must give a more precise explanation of the theory.

Let \mathcal{X} be the space of real-valued processes indexed by the positive integers. Let X be a stationary random process in \mathcal{X} , $X = (X_1, X_2 \dots)$, where X_i is the amount of work generated by a source at time i . Write $X(0, t]$ for $X_1 + \dots + X_t$. Define the effective bandwidth of X to be

$$\alpha_X(\theta, t) = \frac{1}{\theta t} \log \mathbb{E} \exp(\theta X(0, t])$$

for $t \in \mathbb{N}$ and $\theta \in \mathbb{R}_+$.

It is shown in Wischik [28] that the probability of overflow in a queue with service rate C and buffer size B , fed by traffic flow X , is given by¹

$$\log \mathbb{P}(\text{overflow}) \approx -I$$

where the *rate function* I is given by

$$I = \inf_{t \in \mathbb{N}} \sup_{\theta \in \mathbb{R}_+} \theta(B + Ct) - \theta t \alpha_X(\theta, t).$$

The optimizing t^* and θ^* are called the operating point of the queue.

We do not need the next result immediately, but it will be important in analysing marking algorithms. It says that conditional on overflow, the amount of work produced by X in the busy period leading up to overflow is

$$x(0, t^*] = \frac{\partial}{\partial \theta^*} \theta^* t^* \alpha(\theta^*, t^*). \quad (1)$$

Suppose that the queue has many inputs with total effective bandwidth α and we replace a small proportion δ of them by flows which produce work at constant rate a (these have effective bandwidth a). The rate function is now

$$I(\delta) = \inf_t \sup_{\theta} \theta(B + Ct) - \theta t((1 - \delta)\alpha(\theta, t) + \delta a). \quad (2)$$

If the optimizing parameters for I are θ^* and t^* , and under appropriate differentiability conditions, the value of a that makes $I'(0) = 0$ is $a = \alpha(\theta^*, t^*)$. In other words, an input flow has the same effect on the system as would a constant flow of rate $\alpha(\theta^*, t^*)$. This is why α is called the effective bandwidth function.

¹This can be made precise using large deviations theory. The precise statement is a limit theorem concerning a queue with service rate LC and buffer size LB , fed by the aggregate of L copies of X : for such a system, under reasonable conditions on X ,

$$\frac{1}{L} \log \mathbb{P}(\text{overflow}) \rightarrow -I.$$

If the switch has multiple input flows of different types, then the effective bandwidth function measures the tradeoff between different flows. For example, suppose that a router has inputs of types A and B and at the operating point (θ^*, t^*) of the queue, $\alpha_A(\theta^*, t^*) = 2\alpha_B(\theta^*, t^*)$. Then replacing one flow of type A by two flows of type B will not affect the loss probability.

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 2 would suggest marking in proportion to effective bandwidth—or, equivalently, in proportion to $t^* \alpha_X(\theta^*, t^*)$ —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 6.)

We can also address the second point. The ECN proposal [21] 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.

It is shown by Wischik [27] 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.

3.3 Efficiency. Courcoubetis et al. [5] 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.

3.4 Summary of effective bandwidth: 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); 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 losing 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.

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—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 [18], 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: a 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, some sort of interpersonal comparison is inherent in defining fairness. And in this paper we are at least as interested in fairness as we are in efficiency.

4.1 How to efficiently mark 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.) 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 a simple social welfare problem: to maximise the net utility. In other words,

$$\max_{\mathbf{x}} \sum_{u \in \mathcal{U}} U_u(x_u) - \sum_{r \in \mathcal{R}} C_r(y_r) \quad \text{such that} \quad x_u \geq 0 \quad \forall u \in \mathcal{U} \quad (3)$$

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

$$\mathcal{L} = \sum_{u \in \mathcal{U}} U_u(x_u) - \sum_{r \in \mathcal{R}} C_r(y_r) + \sum_{r \in \mathcal{R}} \lambda_r \left(y_r - \sum_{u:r \in u} x_u \right) \quad (4)$$

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

$$\begin{aligned} \lambda_r &= \frac{dC_r}{dy_r} \quad \text{and} \\ \frac{dU_u}{dx_u} &= \sum_{r \in u} \lambda_r \quad \text{if } x_u > 0. \end{aligned} \tag{5}$$

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 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

$$P_u(\mathbf{x}) = x_u \sum_{r \in u} \lambda_r. \tag{6}$$

Then the solution to the system of equations (5) coincides with the solution to the welfare problem (3).

The pricing structure (6) 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.

4.2 How to mark fluid-random flows. These results for fluid flows apply also to random flows parameterized by a scalar quantity. One example, first described by Gibbens and Kelly [9], is especially worth noting, as it leads to a very simple marking algorithm.

As usual, assume a slotted time traffic model. Also assume for simplicity that all packets are the same size. Consider a bufferless resource which can serve C packets in every timeslot, 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 when the aggregate input is a Poisson flow of rate y_r . That is,

$$C_r(y_r) = \mathbb{E}(Y_r - C)^+$$

where Y_r is Poisson with parameter y_r . Then it can be shown that the correct expected charge given in (6) is achieved 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).

4.3 How to efficiently mark bursty flows. So far we have assumed fluid traffic flows, or at least traffic flows parameterized by a scalar rate. But the optimization (3) 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 a network of bufferless resources, operating as before in slotted time, and assume that all packets are the same size. Suppose that each user u transmits a random amount of work at each timestep. Suppose that each user can choose a probability distribution controlling how much work is sent;

it is over these distributions that we wish to optimise. So let x_u in (3) be a distribution over the nonnegative integers, rather than a scalar as in the last two sections. 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 Wischik [27].) 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 (4) becomes

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

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

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

Really, we should include constraints that $0 \leq x_u(n) \leq 1$ and $\sum_n x_u(n) = 1$. 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:

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

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

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

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 why this can be considered to be fair. We 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) and (6).

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 levied on 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.

4.4 Problems with efficient marking. There are several concerns about our efficient pricing scheme ΔL .

We have simplistically taken the social welfare function (3) 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. Gibbens and Kelly [9] give a useful description Pareto optimality: if we treat the network as a player, the Lagrangian (4) characterizes Pareto efficient allocations.

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 [9] give some examples of what might happen. 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 applica-

tion? 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. [15].

4.5 Summary of economics and efficiency: ΔL . 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 Economics and fairness

In Section 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 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; and we will introduce another pricing scheme, called SPSP. The principal references are Baumol [1] and Zajac [29].

5.1 No-envy fairness. Perhaps the most mathematically developed idea of fairness is the theory of envy-free allocations. An individual A is said to *envy* individual B if he would rather have B 's goods than his own. An allocation is *envy-free* if no-one envies anyone else. Such allocations can be deemed to be fair. We will call the resulting definition of fairness *no-envy fairness*. Some authors use the term *superfair* to refer to allocations in which everyone strictly prefers his own goods.

Unfortunately this theory is of no use in congestion pricing, and we only mention no-envy fairness here to reject it. Any pricing scheme would lead to an envy-free allocation, because if A envies B then A can just start sending traffic with the same distribution as B . We however want the charge levied on a user to reflect the amount of congestion he causes.

5.2 The burden test. The idea of a fair price arises in monopoly trials, in which a company may face charges of 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 is fair. (But we shall revise this conclusion in Section 6.)

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, because that would lead to extra control traffic and thus greater congestion. We 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 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 [23] have shown that the core may not coincide with the set of socially desirable outcomes, and in some cases it may not even exist.

5.4 Incremental fairness. The difficulties about users banding together, and also the problem described in Section 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 different 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. The burden test 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 shall say that a pricing scheme is *incrementally fair*, or 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 [9]. 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 1. Clearly SPSP satisfies the condition of anonymous equity, since it charges each individual packet its incremental cost.

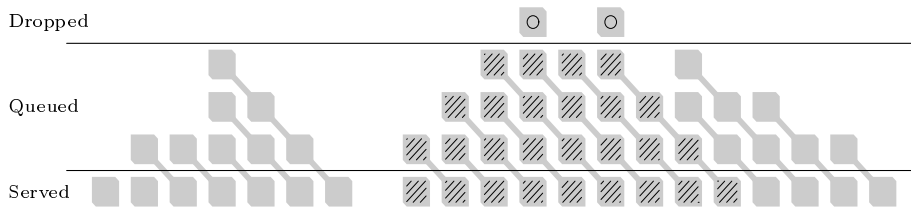


Figure 1: Sample path shadow price marking. The squares represent packets, and the grey diagonal lines indicate the progress of a packet through the queue. Shaded packets are those that would be marked by SPSP. This rule marks each packet whose removal would result in one less packet being dropped.

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 Section 4.2. 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 no-envy theory 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 parties may result in an allocation which is unfair to all parties.

5.5 Summary of economics and fairness: SPSP. 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 3, which suggests charging according to EB.

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

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 in Section 7, in pointing out how various proposed marking algorithms can be unfair.

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 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 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 Section 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 (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.

6.2 Anonymity. For a certain 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 7, then give the more natural interpretation in terms of packets.

Anonymity is based on the requirement that at the critical point each flow X looks as if it is made up of a number of independent copies of some base flow P .

Specifically, call a traffic mix *anonymous* if for each flow X there is a multiple k_X such that the effective bandwidth satisfies $\alpha_X(\theta^*, t^*) = k_X \alpha_P(\theta^*, t^*)$ and the most likely paths to overflow satisfy $x_t^* = k_X p_t^*$ for $0 < t < t^*$, where (θ^*, t^*) is the critical point. One might think of 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 p^* sample path identically, these two definitions of fairness agree.

Now let us interpret this definition in terms of packets. Think of P as representing an isolated packet. At the critical point, i.e. in the busy period leading up to overflow, each aggregate flow X looks as if it is made up of independent copies of 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.

6.3 Why 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 3–5 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 [9] introduced SPSP for the very reason that it is the most 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 3.1 tells us about resource usage as well. It identifies the critical timescale t^* , and hence the limited capacity $B + Ct^*$ 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 (1) gives us $x^*(0, t^*)$, which is the amount of limited capacity consumed by source X . We can suggestively rewrite that expression as

$$t^* \alpha_X(\theta^*, t^*) = x^*(0, t^*) - \theta^* t^* \frac{\partial}{\partial \theta^*} \alpha_X(\theta^*, t^*). \quad (7)$$

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 3.1 we saw 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 (2), 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 $B + Ct^*$ 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. [5] 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 $x^*(0, t^*)$ 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 θ^* will be very small, and so the second term in (7) 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.

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. These differences arise because the three measures are derived from three different notions of burstiness.

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.

7 Marking algorithms

How well does a given marking algorithm achieve SPSP-fairness? In this section, we will use large deviations theory to study the performance of the RED marking algorithm, and find example traffic flows for which it is unfair. We will see that some small modifications can make it fairer (indeed, perfectly fair in anonymous scenarios), and we summarise these changes in a new algorithm called ROSE.

7.1 Mathematical theory. Our main mathematical tool will be the following result about most likely paths. Suppose that a queue is fed by a random traffic flow X . Write $\mathbf{X}(0, t]$ for the vector (X_1, \dots, X_t) . Let E be some event of interest, such as ‘the queue overflows’ or ‘a packet is marked’. Then

$$\log \mathbb{P}(E) \approx -I$$

where

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

and

$$\Lambda_t(\boldsymbol{\theta}) = \log \mathbb{E} \exp(\boldsymbol{\theta} \cdot \mathbf{X}(0, t]).$$

Let t^* and $\boldsymbol{\theta}^*$ be the optimizing parameters. Then conditional on event E occurring, say at time s , the most likely time at which the queue was last empty is $s - t^*$, and the most likely path to lead to E is given by

$$x_{s-t^*+i}^* = \frac{\partial}{\partial \theta_i} \Lambda_{t^*}(\boldsymbol{\theta}^*)$$

where the derivative is taken at $\theta_i = \theta_i^*$ for $0 < i \leq t^*$ and $\theta_i = 0$ otherwise.

As we mentioned in Section 3, this theory is a limiting theory, and it is accurate in the limit in which the number of flows (i.e. aggregation level) increases but the service rate and buffer size per flow stays fixed. If L is the number of flows, then $L^{-1} \log \mathbb{P}(E) \rightarrow -I$, and the probability of any deviation from the most likely path x^* decays to zero exponentially quickly in L . See Wischik [28] for details of the theory and calculations.

7.2 The RED algorithm. The Random Early Detect (RED) algorithm was designed by Floyd and Jacobson [8] and has been implemented in commercial routers [4]. We will, for convenience, deal with a version of RED operating in slotted-time, and assume that 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 in the large deviations limiting regime, the number of sources and the capacity of the resource increasing; 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.²

²In 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 our simple queueing model is only an approximation to the behaviour of an Internet router. 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.

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.

7.3 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. Suppose that the input process has mean rate μ and covariance matrix Γ . The average queue size at time t when the input is \mathbf{x} is given by

$$\bar{q}_t(x) = \mathbf{w}^\top (\mathbf{x}(0, t] - C\mathbf{1})$$

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

$$\inf_{x: \bar{q}_t(x)=b} \sup_{\boldsymbol{\theta}} \boldsymbol{\theta}^\top \mathbf{x}(0, t] - (\mu \mathbf{1}^\top \boldsymbol{\theta} + \frac{1}{2} \boldsymbol{\theta}^\top \Gamma \boldsymbol{\theta})$$

which is attained at

$$\mathbf{x}^*(0, t] = \mu \mathbf{1} + (b + (C - \mu) \mathbf{1}^\top \mathbf{w}) \frac{\Gamma \mathbf{w}}{\mathbf{w}^\top \mathbf{w}}.$$

Marking happens at critical point of the form $\boldsymbol{\theta}^* = \theta^* \mathbf{w}$ (whereas overflow typically happens at a critical point of the form $\phi^* \mathbf{1}$). 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 it decreases again. So RED marks a fixed proportion of the packets that arrive at time t .

The behaviour of RED is illustrated in Figure 2. We could have chosen an anonymous scenario, 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. 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. Note that these pictures only illustrate most likely paths: they do not tell us about the relative frequencies of overflow and marking, so we cannot conclude directly from the picture that SPSP marks more or fewer packets than RED.

7.4 Modes of failure. The first problem with RED is that it closes the stable doors after the horse has bolted—and then blames the horses left inside for running away! The packets that arrived before overflow occurred are the ones that caused the problem, and Figure 2 shows that overflow is attributable in roughly equal measure to the light and dark flows. But RED only starts marking after overflow has occurred, and so it marks the flows roughly in proportion to their mean rates rather than in proportion to their burstiness: it gives the lighter flow too few marks.

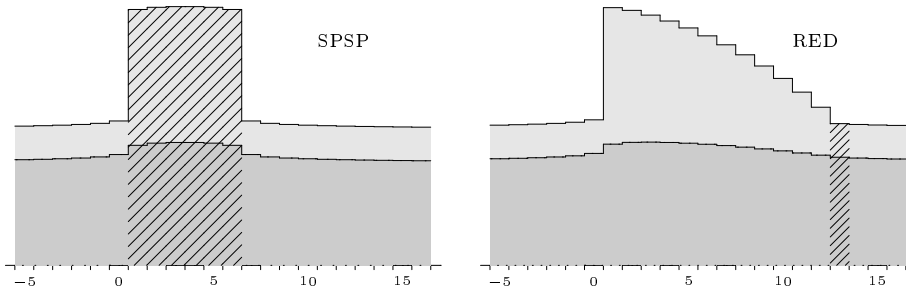


Figure 2: How RED marks. The left graph shows the most likely path to lead to overflow: it plots the amount of incoming work at each timestep. The shading indicates the marks that SPSP would give. The right graph shows the most likely path to lead to marking by RED, and indicates how much each flow is 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%.

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. But if for example the threshold b is small and the most likely time to marking, t , is large, then most of the guilty packets will have escaped.

Various authors have proposed modifications to RED. Feng et al. [6] describe BLUE, which has the goal of smoothing out the flow of marks. In the large deviations limit this goal is not apparent, and BLUE suffers from exactly the same problem as RED. Lin and Morris [16] have describe another modification 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 2, when RED starts marking at time 13, most of the work in the queue belongs to the darker flow: so FRED would only drop that flow’s packets. In other words, in this example the unfairness of RED has been exacerbated!

A more basic problem with RED is that marking is not representative of overflow, in that they occur in essentially different ways. This is because the critical point for marking, $\theta^* \mathbf{1}$, is not the same as the critical point for overflow, $\phi^* \mathbf{1}$. Therefore the most likely path to lead to marking will be different from the most likely path to lead to overflow. This difference is clearly seen in 2. Even if RED were able to start marking at time 0 and continue marking throughout the critical time-period, it would mark packets in the wrong proportions.

7.5 Setting RED parameters It is widely accepted that the RED parameters must be set to match the traffic characteristics. Feng et al. [7] 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 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 some 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 3, for a queue with service rate 1.5 fed by a 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: if the current queue size is given a large weighting, we should accept fairly large fluctuations in the average queue size.

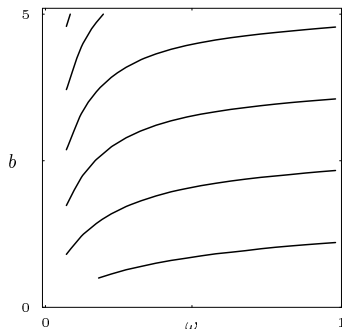


Figure 3: How to set some RED parameters. Each line indicates a family of choices of ω and b that lead to the same frequency of marking, for a specific traffic distribution. To change the way the system responds, without changing the value of a mark, ω and b should be changed together along one of these lines.

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. We now go on to describe such an adaptive algorithm.

7.6 Reach Overload, Send ECN. 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.

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 of Gibbens and Kelly [9] would be fair if the virtual queue scaling factor were set adaptively.

The 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 2.4, 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.

Fairness of ROSE. The ROSE algorithm addresses both of the problems identified in RED. Furthermore, it is perfectly fair in anonymous scenarios, and approximately fair in many others.

We will first deal with the second problem with RED. At the large deviations scale, the adaptive algorithm must settle on a value of b equal to the buffer size. We know this because for every overflow event, a fixed number of marks are given; thus the rate function for marking is equal to the rate function for overflow; and the only value of b that achieves 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 exactly the same as that for overflow.

It is now easy to deal with the first problem with RED. All the packets that ROSE marks did indeed contribute to overflow, since it marks everything that is in the buffer when overflow occurs.

Thus ROSE does not suffer from either of the two specific flaws identified in RED. Is it fair in general? In other words, does it mark in proportion to congestion at a queue; and does it mark individual flows in proportion to their contribution to congestion?

It is easy to check that ROSE generates marks in proportion to congestion: by construction, it marks exactly the number of packets that SPSP says should be marked (or a constant multiple κ thereof).

It is not the case that ROSE is fair in all scenarios. It is, however, fair in anonymous scenarios (that is, in scenarios where the three candidate definitions of fairness agree). To see this, recall the effective bandwidth definition of anonymity: that at the critical point, each flow can be treated as if it is made up of a certain number of copies of some base flow P . The number of copies of P that make up a flow X is proportional to the effective bandwidth of X at the queue's operating point. Now, since the copies of P are identical, each will leave the same amount of work in the queue at the time of overflow. This means that the amount of work belonging to X that is caught in the queue at the time of overflow is proportional to the effective bandwidth of X . Thus, in anonymous scenarios, ROSE is fair.

When the traffic mix is not anonymous, ROSE may not agree with SPSP. One can construct examples where the two are arbitrarily different, by choosing sources with peculiar paths to overflow. However, they 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, for example in queues with large buffers. Large deviations in queues with large buffers have been described, for example, by O'Connell [20]. In the large-buffer limit, the most likely paths to overflow are constant rate—that is, the most likely way for the queue to fill up over an interval of length t , is if each source produces work at a rate that higher than its mean rate, and constant throughout that interval. 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.

7.7 Summary of marking algorithms Strictly speaking, all we have found are 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 tools too only allow 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.

We nonetheless hope that these negative tests will be of considerable help in designing better marking algorithms.

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 SPSP and ROSE mark groups of successive packets, will it not lead to synchronization and instability?

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 [24] 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 the analysis of ROSE 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 [2, 14] 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?

Much of the attention given to fairness in the engineering community has focussed on *distributive fairness*, that is, on how to achieve a fair allocation of capacity. Max-min fairness is an example of this. It is an idea that can be traced back to Rawls, and further. He proposed that social and economic inequalities be arranged to the greatest benefit of the last 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 incorporate demand for different services, and to cope with random traffic flows—the objects of study for this paper—where the idea of mean bandwidth is not very relevant. Therefore we have approached fairness from a different perspective, the perspective of *fair pricing*. This has allowed us to treat the problem of marking on its own, without regard for user behaviour or the allocation of capacity that results.

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 [21] 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. 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. [15] discuss further the use of marks as a pricing mechanism, and MacKie-Mason and Varian [19] discusses usage-based pricing in general.

9 Summary

In this paper 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 user burstiness. 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.

I am very grateful to Frank Kelly for many fruitful discussions.

References

- [1] William J. Baumol. *Superfairness: applications and theory*. MIT Press, 1986.
- [2] Microsoft Research Cambridge. Congestion pricing and a distributed game. Available on the Internet, 1999. URL <http://www.research.microsoft.com/research/network/disgame.html>.
- [3] 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>.
- [4] *Cisco IOS Release 12.0, Configuring WRED*. Cisco, 1999. URL <http://www.cisco.com/>.
- [5] 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>.
- [6] 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/>.
- [7] 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>.
- [8] 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>.
- [9] 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>.
- [10] V. Jacobson. Congestion avoidance and control. In *Proceedings of SIGCOMM*. ACM, August 1988. URL <http://www-nrg.ee.lbl.gov/papers/congavoid.pdf>.
- [11] John Paul II. Centesimus annus. Encyclical letter, 1991. URL http://www.vatican.va/holy_father/john_paul_ii/encyclicals/.

- [12] 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>.
- [13] 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>.
- [14] 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>.
- [15] 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>.
- [16] 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>.
- [17] 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>.
- [18] 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>.
- [19] 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>.
- [20] 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.hp1/hp.com/pub/noc/papers/9604.ps>.
- [21] 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>.
- [22] Amartya Sen. *On Ethics and Economics*. Blackwell, 1987.
- [23] Lloyd S. Shapley and Martin Shubik. On the core of an economic system with externalities. *American Economic Review*, 59(4):678–684, September 1969.
- [24] 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>.

- [25] 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.
- [26] Hal R. Varian. *Microeconomic Analysis*. Norton, third edition edition, 1992.
- [27] Damon Wischik. The output of a switch, or, effective bandwidths for networks. *Queueing Systems*, 32:383–396, 1999. URL <http://www.statslab.cam.ac.uk/~djw1005/Stats/Research/output.html>.
- [28] Damon Wischik. Sample path large deviations for queues with many inputs. URL <http://www.statslab.cam.ac.uk/~djw1005/Stats/Research/sampleldp.html>. To appear in *Annals of Applied Probability*, 2001.
- [29] Edward E. Zajac. *Political economy of fairness*. MIT Press, 1995.