A Processor-Sharing Heuristic for Multipath Congestion Control

Jonatha Anselmi[1], Bernardo D'Auria[2] and Neil Walton[3]

Abstract— **Protocols are currently being designed where a connection can simultaneously send traffic across multiple routes (or paths) of a communication network; for example, the IETF's MP-TCP protocol. Traditional single-path congestion-control protocols, which increase and decrease transmission rates depending on the level of congestion, can be argued to implicitly solve a network-wide utility optimization in a decentralized manner. It is well-known that this optimization implies desirable properties such as fairness. However, it is not really clear in the literature how to achieve such desirable optimization when a connection can use multiple paths. In this case, a connection aims at balancing traffic between the available routes striving a tradeoff between congested and uncongested routes. In this multipath framework, we propose a simple load-balancing heuristic for splitting traffic among multiple routes. Our load-balancer is inspired by the modus operandi of processor-sharing queues and, essentially, operates as follows: every time a packet is acknowledged on a route, increase the congestion window of that route by one and decrease the congestion window of a route selected with probability proportional to the size of its congestion window. In our main result, we argue that this simple end-to-end policy achieves a desirable network-wide utility optimization for multipath routing, when the network is congested. Variants of our heuristic and its implications are discussed.**

I. INTRODUCTION

Multipath routing is an appealing mechanism for improving performance and resilience in communication networks. The idea behind multipath routing is simple: by communicating along multiple paths, the parallelism and path diversity that are inherent within a network can be exploited better.

In the context of Internet communications, there are immediate benefits. Firstly, a multipath protocol would react to link failures by shifting load to functioning routes on a time-scale that is much quicker than those obtained by updating the routing tables of network routers. Further, path diversity already is inherent in many Internet devices: a smart-phone usually has both 3G and WiFi capabilities. By allowing multipath routing between these media and between cells within these media, transmission can be maintained as one transfers between different areas of connectivity. Finally and somewhat more abstractly, the use of multiple routes increases resource pooling – that is where multiple network resources can collectively be seen as a single *pooled* resource. Resource pooling has been achieved in telecommunication networks with the implementation of Dynamic Alternative Routing policies [1], [2]. These potential benefits have lead to the research and development of versions of the transmission control protocol (TCP) which allow for multipath routing.

Current TCP implementations are single-path protocols which avoid congestion by an additive increase multiplicative decrease rule [3], [4]. It was later argued that one reason for the success of this congestion avoidance protocol is that the resulting rates (or throughputs) allocated to network users implicitly maximize, in a decentralized manner, a network-wide utility function subject to capacity constraints [5], [6], [7], [8], [9]. This optimization implies convenient properties such as fairness and follows because of a decomposition occurring between the network and the protocols accessing that network [10]. This original decomposition applies to both singlepath and multipath cases. Thus, a similar mathematical analysis for multipath congestion-control protocols has been also developed [11], [12], [13], [14]. Based on earlier insights [1], it is then argued that multipath routing can pool together the resources of a network [15].

Due to the potential benefits of multipath routing, the Internet Engineering Task Force (IETF) MP-TCP working group is currently developing a working multipath TCP implementation.[1] For a discussion on the implementation of these protocols, see [16]. Issues have been observed in the development of these protocols. In particular, when research studies based on differential equation models of congestion control were implemented as a discrete packet-based system, they were found to be highly prone to *route flap* [17]. Here, rather than balancing flows between routes, the sender rapidly switches states between transmitting all packets on one route, then on another route and so forth. Due to this disconnection between the model and practical implementation, it is of interest to investigate *discrete* congestion windowing mechanisms. In this setting, the additive increase and multiplicative decrease rule, as used by (single-path) TCP, is well accepted. However, it remains unclear how to balance traffic across multiple routes in an end-to-end manner while ensuring fairness. This issue will be the focus of this paper.

In this article, we propose a simple heuristic policy that balances a fixed amount of traffic among multiple routes. The key idea is that a congestion window, which records the number of packets traversing each of the available routes, should mimic the behavior of a processor-sharing queue as follows:

[1] Basque Center for Applied Mathematics (BCAM), Alameda de Mazarredo 14, E-48009 Bilbao, Basque Country, Spain, anselmi@bcamath.org

[2] Universidad Carlos III de Madrid, Avda. Universidad 30, 28911 Leganés, Madrid, Spain, bernardo.dauria@uc3m.es

[3] University of Amsterdam, Science Park 904, 1098 XH, Amsterdam, The Netherlands, n.s.walton@uva.nl

[1] A Linux implementation of MP-TCP can be found at http://nrg.cs.ucl.ac.uk/mptcp/implementation.html.

Processor-Sharing Load Balancer. *Every time a packet is acknowledged on a route: with a fixed probability, increase the number of packets on that route by one and remove one packet from a route that is selected with probability proportional to the number of packets on that route.*

This policy only uses local information and is easy to implement. The analogy with the processor-sharing queue is intuitively driven by the following "coupling" argument. Each acknowledgement increases by one the number of packets in the route where the acknowledgement occurred and corresponds to an arrival event at the queue. Each packet removed from its route is interpreted as a service event at the queue. As in the processor-sharing queue, service occurs at a rate proportional to the number of packets in each class/route.

We develop a mathematical model for the dynamics induced by our heuristic on the randomly-varying number of packets in the network. The model assumes that the congestion-window updates of our heuristic occur at a time-scale that is slower than the one where packets are acknowledged. This is motivated by the fact that such updates occur with some small probability every time a packet is acknowledged and thus not for each acknowledged packet. Within this assumption, the rates (or throughputs) given by the network can be modelled as the optimizer of a concave utility-maximization problem [5], [6], [7], [8], [9]. This leads to the definition of a Markov chain with the following surprising property: under a congested limiting regime, the rates allocated to the users solve the utility maximization problem that corresponds to the multipath extension of the proportionally-fair allocation. In this way, as the network becomes congested, our load-balancing heuristic will balance traffic to achieve optimal and fair flow rates.

The remainder of the paper is organized as follows. In Section II, we introduce the modeling framework and the multipath proportionally-fair optimization, that is the design objective we are interested in. In Section III, we present our heuristic, a Markov chain that models its dynamics, and our main result (Theorem 1), which consists in showing that the throughput functions of such Markov chain solve, in the limit, the multipath proportional-fair optimization. In Section IV, we show some numerical results supporting the claim that the throughputs of our Markov chain are close to the desired ones even outside the limit. Finally, Section V discusses possible modifications to our proportionally-fair load balancer and how these might be actualized.

II. Modelling Framework

A. Network Structure and Notation

Let the set $\mathcal{J}$, indexed by j, be the set of *queues* (or *links*). These will be the resources of the network affected by congestion. Packets circulate in the network following given *routes* (or *paths*). A route is an ordered set of queues, $r = (j_1, \ldots, j_{k_r})$ with $j_1, \ldots, j_{k_r} \in \mathcal{J}$. Thus, k_r is the number of queues in route r. The set $\mathcal{R}$ is the set of routes. For notational convenience, we assume that this set is ordered. A *source-destination*, s, is a set of routes $s \subset \mathcal{R}$. Each source-destination contains the set of routes available to the hosts (or network users) for communication. The set $\mathcal{S}$ gives the set of source-destinations. We assume that there are no source-destinations that share a common route, i.e., $s \cap s' = \emptyset$ when $s \neq s'$. Thus, we can define $s(r) \in \mathcal{S}$ as the source-destination to which route r belongs. For conciseness, in this paper we assume that there is exactly one host that uses source-destination s, for each $s \in \mathcal{S}$. Thus, there are $|\mathcal{S}|$ hosts in total.

We now describe the notation related to the state of packets. Let $m_{jr} \in \mathbb{Z}_+$ denote the number of packets at queue $j \in \mathcal{J}$ traversing route $r \in \mathcal{R}$. Accordingly, $m = (m_{jr} : r \in \mathcal{R}, j \in \mathcal{J})$ is the number of packets from each route at each queue. Further, as a shorthand, we will let m_j be the total number of packet at each queue. Let n_r denote the number of packets on route $r \in \mathcal{R}$. Correspondingly, $n = (n_r : r \in \mathcal{R})$ is the number of packets on each route. Let w_s denote the number of packets on source-destination $s \in \mathcal{S}$. Accordingly, $w = (w_s : s \in \mathcal{S})$ is the number of packets on each source-destination. Thus, we have

$$n_r = \sum_{j \in r} m_{jr}, \qquad r \in \mathcal{R}, \tag{1}$$

$$w_s = \sum_{r \in s} n_r, \qquad s \in \mathcal{S}. \tag{2}$$

We now describe the notation related to the processing of packets. The strictly positive vector $C = (C_j : j \in \mathcal{J})$ gives the *capacity* (or *service rate*) of each queue: the number of packets served per unit of time in a non-empty queue. Let Λ_r denote the *throughput* (or *flow rate* or *bandwidth*) of packets on route $r \in \mathcal{R}$, and let Γ_s denote the throughput on source-destination $s \in \mathcal{S}$. Thus, we have

$$\Gamma_s = \sum_{r \in s} \Lambda_r, \qquad s \in \mathcal{S}, \tag{3}$$

$$\sum_{r \ni j} \Lambda_r \leq C_j, \qquad j \in \mathcal{J}. \tag{4}$$

B. Utility Maximization

When the congestion control at a given host has a single route choice, it is known that the throughputs $(\Lambda_r)_{r \in \mathcal{R}}$ can be modeled by the unique solution of the strictly-concave optimization

$$\text{maximize} \qquad \sum_{r \in \mathcal{R}} n_r U_r(\Lambda_r / n_r) \tag{5a}$$

$$\text{subject to} \qquad \sum_{r \ni j} \Lambda_r \leq C_j, \qquad j \in \mathcal{J} \tag{5b}$$

$$\text{over} \qquad \Lambda_r \geq 0, \qquad r \in \mathcal{R}, \tag{5c}$$

where $U_r(\cdot)$ is some strictly-concave function associated to each route, [5], [7], [9]. The value of function $U_r(\cdot)$ is interpreted as the *utility* of packets sent on route r.

On the other hand, when the congestion control at a given host *can* distribute packets among multiple routes, one wishes to design a congestion-control protocol ensuring that

the resulting throughputs $(\Lambda_r)_{r\in\mathcal{R}}$ solve

$$\begin{aligned}
\text{maximize} \quad & \sum_{s\in\mathcal{S}} w_s U_s(\Gamma_s/w_s) && && \text{(6a)}\\
\text{subject to} \quad & \Gamma_s = \sum_{r\in s} \Lambda_r, && s\in\mathcal{S} && \text{(6b)}\\
& \sum_{r\ni j} \Lambda_r \le C_j, && j\in\mathcal{J} && \text{(6c)}\\
\text{over} \quad & \Lambda_r \ge 0, && r\in\mathcal{R} && \text{(6d)}
\end{aligned}$$

where $U_s(\cdot)$ is some strictly-concave function associated to each source-destination. In the above optimization, we notice that the throughputs on different routes are combined to give the throughput between a source and a destination, (6b).

There are several desirable properties in designing decentralized protocols achieving the optimization (6), and these are explained in [14], [18]. We could consider both a singlepath optimization or a multipath optmization to obtain the same form of fairness, if hosts optimize the same objective function.

C. Proportional Fairness and an Approximation.

For this paper, we assume the above optimizations have utilities

$$U_r(\Lambda_r/n_r) = \log(\Lambda_r/n_r), \qquad r\in\mathcal{R}, \qquad (7)$$
$$U_s(\Gamma_s/w_s) = \log(\Gamma_s/w_s), \qquad s\in\mathcal{S}. \qquad (8)$$

The throughputs achieved with such particular choice of the utility functions are called *proportionally-fair*. Several economic and game-theoretic interpretations are known in this case [10], [5], [6], [8], [13], [9]. Communication protocols achieving such optimization objective have also been designed and implemented [19], [20], [21]. Furthermore, proportionally-fair throughputs have connections with product-form queueing networks: proportionally fair throughputs are achieved by Kelly or BCMP networks when the number of packets on each route is large but fixed [22], [23], [24] (this is stated more precisely in the following).

Summarizing so far, our goal is to design a congestion control mechanism ensuring that the resulting throughputs solve

$$\begin{aligned}
\text{maximize} \quad & \sum_{s\in\mathcal{S}} w_s \log\Gamma_s && && \text{(9a)}\\
\text{subject to} \quad & \Gamma_s = \sum_{r\in s} \Lambda_r, && s\in\mathcal{S} && \text{(9b)}\\
& \sum_{r\ni j} \Lambda_r \le C_j, && j\in\mathcal{J} && \text{(9c)}\\
\text{over} \quad & \Lambda_r \ge 0, && r\in\mathcal{R}. && \text{(9d)}
\end{aligned}$$

We refer to problem (9) as the multipath proportionally-fair optimization.

Let us consider singlepath routing and the proportionally-fair optimization. Although this proportionally-fair policy has several desirable features, it may not be tractable for direct modeling analysis (as we show later). The idea is that $\Lambda(n)$ can be also interpreted as the asymptotic throughput of a closed queueing network. More precisely, given the set of queues $\mathcal{J}$ with capacities $C = (C_j : j\in\mathcal{J})$ and the set of routes $\mathcal{R}$ as described, we consider a queueing network where there are a fixed number of packets circulating along each route, $n = (n_r : r\in\mathcal{R})$. That is, every time a packet completes its route, a new packet is injected into the network. We can interpret this as a lossless communication protocol which maintains a fixed congestion window. Under general assumptions [25], [26], it is known that the equilibrium distribution of the number of packets on each route and at each queue $m = (m_{jr} : j\in r)$ is

$$\eta(m) = \frac{1}{B(n)} \prod_{j\in\mathcal{J}} \binom{m_j}{m_{jr} : r\ni j} \left(\frac{1}{C_j}\right)^{m_j}, \qquad (10)$$

where $m_j = \sum_{r\in j} m_{jr}$ and

$$B(n) = \sum_{m\in n} \prod_{j\in\mathcal{J}} \binom{m_j}{m_{jr} : r\ni j} \left(\frac{1}{C_j}\right)^{m_j},$$

is a normalizing constant. For this closed network, the completion rate (or, again, throughput) of route-r jobs can be defined by quantity $B(n-e_r)/B(n)$ [26]. As proven in [23], these quantities are proportionally-fair.

Proposition 1: For any $r\in\mathcal{R}$ and (finite) $\kappa_r\in\mathbb{N}^{|\mathcal{R}|}$,

$$\frac{B(cn+\kappa_r-e_r)}{B(cn+\kappa_r)} \xrightarrow[c\to\infty]{} \Lambda_r(n), \qquad (11)$$

where $\Lambda(n)$ is the optimizer of (5) when (7) holds.

Thus an allocation policy of the form

$$\Lambda_r(n) = \frac{B(n)}{B(n+e_r)}, \qquad r\in\mathcal{R} \qquad (12)$$

can be used to approximate the singlepath proportionally fair allocation or, for that matter, the throughputs of a network with a fixed number of packets circulating each route. This policy is a minor modification of the store-forward policy in [27].

III. Main Result

In this section, we develop a simple heuristic rule that balances traffic among multiple routes of a network. In our main result, we argue that the throughputs achieved by our heuristic have the appealing property of solving the *multipath* proportionally fair optimization (9), provided that the number of packets between each source-destination is large.

A. Processor-Sharing Load Balancer

We consider a load balancer which, in short, sends incoming packets along a given route with a rate proportional to the number of packets in transfer on that route. This modus operandi is similar to how a Processor-Sharing (PS) queue serves different customer classes [28]. In a PS queue, the proportion of service devoted to a given class is proportional to the number of jobs of that class. Given that our heuristic is inspired by the functioning of PS queues, we refer to it as "Processor Sharing Load Balancer" (PSLB). More precisely, for each s, PSLB operates with information that is available

locally, uses tuning parameters $p_s \in (0,1)$ for $s \in \mathcal{S}$, discussed below, and works as described in Algorithm 1.

Algorithm 1: PSLB on host s

Data: $(n_r)_{r\in s}, p_s$

Result: $(n_r)_{r\in s}$ # *updated congestion windows for* s

begin

wait for an acknowledged packet;

$r =$ "route of the acknowledged packet";

if *rand*$([0,1]) \le p_s$ **then**

$n_r = n_r + 1$;

$q =$ rand$([0,1])$;

Let r' be the route satisfying

$$\frac{\sum_{\bar{r}\in s:\bar{r}<r'}(n_{\bar{r}}+1)}{\sum_{\bar{r}\in s}(n_{\bar{r}}+1)} \le q < \frac{\sum_{\bar{r}\in s:\bar{r}\le r'}(n_{\bar{r}}+1)}{\sum_{\bar{r}\in s}(n_{\bar{r}}+1)} \tag{13}$$

$n_{r'} = n_{r'} - 1$;

return $(n_r)_{r\in s}$;

end

In words, every time a packet is acknowledged on path $r \in s$, with probability $1-p_s$, we keep the congestion windows on s unchanged, and with probability p_s, we

1) increase by one the congestion window of path r,
2) generate the random number q between zero and one, and decrease by one the congestion window of path r' where r' is the unique route such that (13) holds.

It is clear that PSLB is simple and easy to implement on end-to-end hosts which maintain a congestion window for each route choice.

Notice the probability of decreasing the congestion window on each route (13) is, approximately, proportional to the congestion window on each route.

Remark 1: The main tuning parameter in this policy is p_s, the probability of load balancing. In the following section, we will assume that p_s is suitably small so that the network is "in equilibrium", given the number of flows present in each route. However, we will see that the throughputs of this system are not a function parameter p_s. This would suggest balancing updates could occur on a far quicker timescale. The authors believe that load-balancing updates could be applied for every packet acknowledgement with probability one and still obtaining the same qualitative properties that follow.

B. Load Balancing Model

We develop a mathematical model describing the randomly varying number of packets in the network when PSLB is adopted. We model such dynamics by a continuous-time Markov chain with the following transition rates

$$T'(n, n+e_r-e_{\tilde{r}}) \stackrel{\text{def}}{=} p_s\,\Lambda_r(n)\frac{n_{\tilde{r}}+1}{\sum_{r'\in s}(n_{r'}+1)} \tag{14}$$

for each $r, \tilde{r} \in s$ and $T'(n, n+e_r-e_{\tilde{r}}) \stackrel{\text{def}}{=} 0$ otherwise. For each $s \in \mathcal{S}$, p_s is the tuning parameter of PSLB; $\Lambda_r(n)$ is the throughput allocated to route $r \in s$ given the number of flows across all routes, n; and e_r is unit size-$|\mathcal{R}|$ vector in direction r.

Rates (14) are interpreted as follows: given the number of packets in flight on each route, n, packets are acknowledged at rate $\Lambda(n) = (\Lambda_r(n) : r \in \mathcal{R})$. Assuming the load balancing probability is reasonably small. Each of these has some small probability of leading to a load balancing step, leading to an exponentially distributed amount of time until the next load balancing step. This occurs on each route at rate $p_s\Lambda_r(n)$, $r \in s$. The final term represents the probability, under the PSLB policy, that the load added to route r is taken from route $\tilde{r}$.

Although our model considers a fixed number of file transfers between hosts, it closely relates to other flow-level Internet models, such as the ones in [29], [30]. Indeed, our model can be interpreted as a closed version of the latters.

Remark 2: The rates $\Lambda(n)$ represent equilibrium throughputs of our network [7], and the PSLB policy is executed on each packet acknowledgment. However, we stress that the *updates* of PSLB does not necessarily occur at the timescale of packets: if the tuning parameter p_s is small for all $s \in \mathcal{S}$, then the updates of the congestion window are not very frequent and we can assume that between two consecutive updates the throughputs achieved by the network have reached their equilibrium value $\Lambda(n)$. Even so, one might expect that smooth (adiabatic) load balancing steps would allow connections to maintain a close to equilibrium state.

This technical remark helps to motivate our model. Further investigations would be required to understand the performance of our load-balancer when p_s is close to one.

The Markov chain (14) defined with proportionally fair rates (9) is not easy to study directly. However, as previously discussed, these rates can be approximated by (12). So, we can consider a continuous-time Markov chain with transition rates

$$T(n, n+e_r-e_{\tilde{r}}) \stackrel{\text{def}}{=} p_s\frac{B(n)}{B(n+e_r)}\frac{n_{\tilde{r}}+1}{\sum_{r'\in s}(n_{r'}+1)} \tag{15}$$

for each $r, \tilde{r} \in s$ and $T(n, , n+e_r-e_{\tilde{r}}) \stackrel{\text{def}}{=} 0$ otherwise for each $s \in \mathcal{S}$. Rather straightforwardly, it turns out that the Markov chain defined by rates (15) brings convenient properties such as reversibility.

C. Equilibrium Analysis

We analyze the stationary behavior of the Markov chain defined by rates (15). The following theorem is our main result.

Theorem 1: (a) The Markov chain with rates (15) is reversible and with invariant measure

$$\pi(n) = \frac{1}{B(n)}\prod_{s\in\mathcal{S}}\binom{w_s+|s|}{n_r+1 \ : r\in s}, \tag{16}$$

where e is the $\mathcal{R}$-vector of all ones.
(b) The asymptotics of $\pi(n)$ are as follows

$$\lim_{c\to\infty}\frac{1}{c}\log\pi(cn) = -\min_{\Lambda\geq 0}\sum_{s\in\mathcal{S}}\sum_{r\in s} n_r\log\left(\frac{n_r}{w_s\Lambda_r}\right) \quad (17a)$$

$$\text{subject to } \sum_{r\ni j}\Lambda_r \leq C_j,\ \forall j\in\mathcal{J}. \quad (17b)$$

(c) The most likely state of this system is characterized through the multipath routing optimization, i.e.,

$$\max_{n\in w}\lim_{c\to\infty}\frac{1}{c}\log\pi(cn) = \max_{\Lambda\geq 0}\quad \sum_{s\in\mathcal{S}} w_s\log\Gamma_s \quad (18a)$$

$$\text{subject to}\quad \Gamma_s = \sum_{r\in s}\Lambda_r,\quad s\in\mathcal{S} \quad (18b)$$

$$\sum_{r\ni j}\Lambda_r \leq C_j,\quad j\in\mathcal{J}\ . \quad (18c)$$

where $n\in w$ means that the conditions (2) are satisfied.
(d) If we let $\Gamma'_s(w)$ be the mean throughput between source-destination pair s for equilibrium π with weights w, that is

$$\Gamma'_s(w) \stackrel{\text{def}}{=} \mathbb{E}_\pi[\sum_{r\in\mathcal{R}}\frac{B(n)}{B(n+e_r)}], \quad (19)$$

then

$$\Gamma'_s(cw)\xrightarrow[c\to\infty]{}\Gamma_s(w)$$

where $\Gamma_s(w)$ maximizes the multipath proportionally fair objective (9).

Part (a) shows that our Markov chain is reversible and gives its stationary distribution, which is reminiscent of the stationary distribution of $\mathcal{S}$ independent processor sharing queues. Part (b) characterizes the asymptotic likelihood of different states in a limit where the number of packets on each route is increased. This likelihood function is both a form of singlepath proportionally fair optimization and the likelihood function of a multiclass processor sharing queue. Part (c) shows that the most likely state of this system is one that solves the *multipath* proportionally fair optimization problem. Part (d) states that the asymptotic throughputs of this Markov chain converge to a solution of the multipath proportionally fair optimization.

Parts (a-c) are proven in this extended abstract; a proof of part (d) would follow in a similar manner to Theorem 7.2 of [23] but is not currently provided in this extended abstract.

IV. Numerical results

The goal of this section is to compare the behavior of the Markov chain defined by rates (14), which solves the (singlepath) proportionally fair optimization, with the behavior of the Markov chain defined by rates (15), which has stationary distribution π given in Theorem 1. Both chains have been discussed in previous section, and we recall that the latter is motivated by the approximation

$$\Lambda_r(n)\approx\frac{B(n)}{B(n+e_r)},\ r\in\mathcal{R}, \quad (20)$$

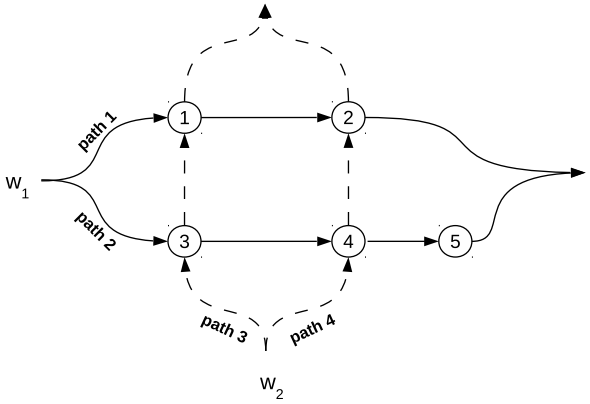

Fig. 1. Topology of the toy-example network.

shown in Proposition 1. In particular, we aim at understanding how load balanced proportionally fair model:

$$\Lambda'_r \stackrel{\text{def}}{=} \mathbb{E}_{\pi'}[\Lambda_r(n)],\ r\in\mathcal{R}, \quad (21)$$

is close to quantity

$$\Lambda_r \stackrel{\text{def}}{=} \mathbb{E}_\pi[\frac{B(n)}{B(n+e_r)}],\ r\in\mathcal{R} \quad (22)$$

where π' and π are the stationary distributions of the chains with rates (14) and (15) under proportional fairness, respectively.

When the overall number of packets w populating the network is large, both Λ and Λ' should be close each other in view of Proposition 1. We now show that both quantities are close each other even when w is relatively small. We investigate this property numerically by considering the simple network in Figure 1, which is characterized by two source-destinations, two paths for each source-destination and five queues. In Tables I and II, we show Λ' and Λ with respect to different capacity vectors and by proportionally increasing the number of customers in each source-destination. In Table I, we use the capacity vector $C=(1,1,1,1,1)$, which implies that all queues except the fifth are bottlenecks. In Table II, we use the capacity vector $C=(1,1,1,1,0.25)$, which implies that the bottlenecks are queue one, two and five. Throughputs have been computed by Monte Carlo

w_1, w_2	Λ_1, Λ'_1	Λ_2, Λ'_2	Λ_3, Λ'_3	Λ_4, Λ'_4
10, 5	.613, .666	.560, .666	.372, .333	.372, .333
20, 10	.633, .666	.617, .666	.360, .333	.360, .333
30, 15	.642, .666	.634, .666	.354, .333	.354, .333
40, 20	.649, .666	.643, .666	.350, .333	.350, .333
∞,∞	.666, .666	.666, .666	.333, .333	.333, .333

TABLE I
Comparison between Λ and Λ' fixing $C=(1,1,1,1,1)$.

simulation, and the initial states have been chosen such that packets were equally divided in each path. In each table, we observe that the throughputs of both chains are very close each other for any value of w. More surprisingly, they are also close to their asymptotic values (shown in the last row

w_1, w_2	Λ_1, Λ_1'	Λ_2, Λ_2'	Λ_3, Λ_3'	Λ_4, Λ_4'
10, 5	.636, .540	.218, .249	.394, .454	.394, .454
20, 10	.625, .577	.243, .250	.401, .422	.401, .422
30, 15	.615, .582	.249, .250	.406, .417	.406, .417
40, 20	.607, .583	.250, .250	.410, .416	.410, .416
∞, ∞	.583, .583	.250, .250	.416, .416	.416, .416

TABLE II
COMPARISON BETWEEN Λ AND Λ' FIXING $C = (1, 1, 1, 1, 0.25)$.

of both tables), which suggests that the speed of convergence can be fast for both chains (when w increases). These insights have been also obtained with respect to a wider set of parameters, which we omit in the current presentation.

V. CONCLUSIONS

We discuss some future research directions:

- The proposed heuristic is shown to mimic the behavior of a processor-sharing queue. As a result, the proportionally-fair multipath optimization is achieved. If our heuristic aims at mimicing the behaviour of a discriminatory processor-sharing queue, mutatis mutandis, we may conjecture that the weighted proportionally-fair multipath optimization is achieved. It is interesting also to understand which modifications will recover the whole set of weighted α-fair multipath optimization [7].
- By placing a weight on each route proportional to the number of packets on that route, our model attempts to perform a weighted prediction on each route and, in this way, learn the best performing routes. It would be interesting to compare the performance of our policy with respect to other weighted schemes, [31], [32].
- Finally, although multipath routing versions of TCP act as a motivation for this work, our load-balancing principle can be considered at the application (instead of transport) layer. For instance, peer-to-peer file-transfer protocols [33], [34] and mobile Apps operate at the application level and here content may be downloaded/uploaded over multiple routes even though they rely on single-path TCP. Other examples are content-delivery or content-centric networks and mobile ad-hoc networks. Again, routes must be evaluated and traffic balanced between the receiver and sender.

APPENDIX

We first prove the following proposition that contains some preliminary results used in the main theorem. This results are already known, see for example [23], [24], but we decided to include their proof here for sake of completeness.

Proposition 2: (a) For $n = (n_r : r \in \mathcal{R}) \in \mathbb{R}_+^{|\mathcal{R}|}$ and $m = (m_{jr} : j \in \mathcal{J}, r \in \mathcal{R}) \in \mathbb{R}_+^{|\mathcal{J}|\times|\mathcal{R}|}$ satisfying (1),

$$\lim_{c\to\infty} \frac{1}{c} \log\left(B(cn)\eta(cm)\right) = -\sum_{\substack{j\in\mathcal{J}\\ r\in\mathcal{R}}} m_{jr} \log \frac{m_{jr}C_j}{m_j}. \quad (23)$$

(b) The normalizing constant $B(n)$ scales as follows

$$\lim_{c\to\infty} \frac{1}{c} \log B(cn) = -\min_{m\geq 0} \sum_{\substack{j\in\mathcal{J}\\ r\in\mathcal{R}}} m_{jr} \log \frac{m_{jr}C_j}{m_j} \quad (24a)$$

$$\text{subject to } \sum_{j\in r} m_{jr} = n_r,\ r \in \mathcal{R}. \quad (24b)$$

(c) The dual of the optimization (24) is the proportionally fair optimization,

$$\max_{\Lambda\geq 0} \quad \sum_{r\in\mathcal{R}} n_r \log \Lambda_r \quad (25a)$$

$$\text{subject to } \sum_{r\ni j} \Lambda_r \leq C_j,\ \forall j \in \mathcal{J}. \quad (25b)$$

Moreover, for any optimal solution to (24), m^*,

$$\frac{m_{jr}^*}{m_j^*} C_j = \Lambda_r^*(n) \quad (26)$$

for all $r \in \mathcal{R}$, $j \in \mathcal{J}$ with $m_{jr}^* > 0$, where $\Lambda^*(n)$ is the proportionally fair solution for (25).

Proof: (a)

$$\begin{aligned}
&\lim_{c\to\infty} \frac{1}{c} \log\left(B(cn)\eta(cm)\right) \\
&= \sum_{j\in\mathcal{J}} \frac{1}{c} \log \left(\binom{cm_j}{cm_{jr} : r \ni j} \prod_{r\ni j} \left(\frac{1}{C_j}\right)^{cm_{jr}} \right) \\
&= -\sum_{j\in\mathcal{J}} \sum_{r\ni j} m_{jr} \log \frac{m_{jr}C_j}{m_j},
\end{aligned}$$

where the last equation follows from an application of the Stirling formula.

(b) Since $\eta(cm) \leq 1$, we can lower bound $B(cn)$ as follows

$$\begin{aligned}
\lim_{c\to\infty} \frac{1}{c} \log B(cn) &\geq \lim_{c\to\infty} \frac{1}{c} \log\left(B(cn)\eta(cm)\right) \\
&= -\sum_{j\in\mathcal{J}} \sum_{r\ni j} m_{jr} \log \frac{m_{jr}C_j}{m_j}.
\end{aligned}$$

Maximizing over $m \geq 0$ gives the lower-bound

$$\lim_{c\to\infty} \frac{1}{c} \log B(cn) \geq -\min_{m\geq 0} \sum_{\substack{j\in\mathcal{J}\\ r\in\mathcal{R}}} m_{jr} \log \frac{m_{jr}C_j}{m_j} \quad (27)$$

$$\text{subject to } \sum_{j\in r} m_{jr} = n_r,\ r \in \mathcal{R}.$$

Now we must provide an upper-bound for $B(n)$. Writing $m \in n$ for the states m such that

$$\sum_{j\in r} m_{jr} = n_r, \qquad r \in \mathcal{R}, \quad (28)$$

we have that $B(n)$ is achieved by summing combinatorial terms over $m \in n$

$$B(n) = \sum_{m\in n} \prod_{j\in\mathcal{J}} \binom{m_j}{m_{jr} : r \ni j} \prod_{r\ni j} \left(\frac{1}{C_j}\right)^{m_{jr}}.$$

For any set of positive constants $(\Lambda_r : r \in \mathcal{R}) > 0$, we can apply the bound

$$\begin{aligned}\mathbb{I}[m \in n] &= \prod_{r\in\mathcal{R}} \mathbb{I}[\sum_{j\in r} m_{jr} = n_r] \le \prod_{r\in\mathcal{R}} \Lambda_r^{-n_r+\sum_{j\in r} m_{jr}} \\ &= \prod_{r\in\mathcal{R}} \Lambda_r^{-n_r} \times \prod_{j\in\mathcal{J}} \prod_{r\ni j} \Lambda_r^{m_{jr}}.\end{aligned}$$

This gives the following upper-bound for $B(n)$

$$\begin{aligned}B(n) &= \sum_{m\ge 0} \mathbb{I}[m \in n] \prod_{j\in\mathcal{J}} \binom{m_j}{m_{jr} : r \ni j} \prod_{r\ni j} \Big(\frac{1}{C_j}\Big)^{m_{jr}} \\ &\le \prod_{r\in\mathcal{R}} \Lambda_r^{-n_r} \times \sum_{m\ge 0} \prod_{j\in\mathcal{J}} \binom{m_j}{m_{jr} : r \ni j} \prod_{r\ni j} \Big(\frac{\Lambda_r}{C_j}\Big)^{m_{jr}}\end{aligned}$$

By the multinomial theorem we have that

$$\begin{aligned}&\sum_{m\ge 0} \prod_{j\in\mathcal{J}} \binom{m_j}{m_{jr} : r \ni j} \prod_{r\ni j} \Big(\frac{\Lambda_r}{C_j}\Big)^{m_{jr}} \\ &= \prod_{j\in\mathcal{J}} \sum_{m_j\ge 0} \Big(\sum_{r\ni j} \frac{\Lambda_r}{C_j}\Big)^{m_j} = \prod_{j\in\mathcal{J}} \Big(1 - \sum_{r\ni j} \frac{\Lambda_r}{C_j}\Big)^{-1}\end{aligned}$$

It follows that

$$B(n) \le \prod_{r\in\mathcal{R}} \Lambda_r^{-n_r} \times \prod_{j\in\mathcal{J}} \Big(1 - \sum_{r\ni j} \frac{\Lambda_r}{C_j}\Big)^{-1} \tag{29}$$

Notice that in equation (29), the right hand side is finite if $\sum_{r\ni j} \Lambda_r \le C_j$, and infinite otherwise. Appropriately scaling and assuming that $\sum_{r\ni j} \Lambda_r \le C_j$, we get the following upper-bound

$$\lim_{c\to\infty} \frac{1}{c} \log B(cn) \le -\sum_{r\in\mathcal{R}} n_r \log \Lambda_r, \tag{30}$$

and minimizing over the region where the inequality is valid, we achieve the bound

$$\begin{aligned}\lim_{c\to\infty} \frac{1}{c} \log B(cn) \le -\max_{\Lambda\ge 0} &\sum_{r\in\mathcal{R}} n_r \log \Lambda_r \\ \text{subject to } &\sum_{r\ni j} \Lambda_r \le C_j,\ \forall j \in \mathcal{J}.\end{aligned} \tag{31}$$

Now, if we can show that the lower bound (27) and the upper bound (31) are equal, then we have shown part (b). This is proved in part (c).

(c) The optimizations (27) and (31) both optimize a strictly convex/concave function over a linear region and so are both *Strong Lagrangian* (i.e. their primal optimization equals their dual). Let's calculate the dual of (27). Given $\Lambda \ge 0$, if we let $(\lambda_r = \log \Lambda_r : r \in \mathcal{R})$ be our Lagrangian multipliers and define, slightly abusing notation, $\Gamma_j = \sum_{r\ni j} \Lambda_r$ and $m_{(j)}$ and $\Lambda_{(j)}$ be the vectors with components $\{r \ni j : m_{jr}\}$ and $\{r \ni j : \Lambda_r\}$ respectively, then the Lagrangian of (27) is

$$\begin{aligned}L(m;\lambda) &= \sum_{\substack{j\in\mathcal{J}\\ r\in\mathcal{R}}} m_{jr} \log \frac{m_{jr} C_j}{m_j} + \sum_{r\in\mathcal{R}} \Big(n_r - \sum_{j\in r} m_{jr}\Big) \lambda_r \\ &= \sum_{\substack{j\in\mathcal{J}\\ r\in\mathcal{R}}} m_{jr} \log \frac{m_{jr} C_j}{m_j \Lambda_r} + \sum_{r\in\mathcal{R}} n_r \log \Lambda_r \\ &= \sum_{j\in\mathcal{J}} m_j \sum_{r\in\mathcal{R}} \frac{m_{jr}}{m_j} \log \frac{m_{jr}/m_j}{\Lambda_r/C_j} + \sum_{r\in\mathcal{R}} n_r \log \Lambda_r \\ &= \sum_{\substack{j\in\mathcal{J}\\ m_j>0}} m_j D\Big(\frac{m_{(j)}}{m_j}\Big|\Big|\frac{\Lambda_{(j)}}{C_j}\Big) + \sum_{r\in\mathcal{R}} n_r \log \Lambda_r \\ &= \sum_{\substack{j\in\mathcal{J}\\ m_j>0}} m_j D\Big(\frac{m_{(j)}}{m_j}\Big|\Big|\frac{\Lambda_{(j)}}{\Gamma_j}\Big) - \sum_{\substack{j\in\mathcal{J}\\ m_j>0}} m_j \log\Big(\frac{\Gamma_j}{C_j}\Big) \\ &\quad + \sum_{r\in\mathcal{R}} n_r \log \Lambda_r.\end{aligned}$$

Here $D(m_{(j)}/m_j || \Lambda_{(j)}/\Gamma_j)$ is the relative entropy between distributions $m_{(j)}/m_j$ and $\Lambda_{(j)}/\Gamma_j$. The relative entropy is positive and minimized at zero when $m_{(j)}/m_j = \Lambda_{(j)}/\Gamma_j$.

We want to minimize the expression above over the space $m \ge 0$. If $\log(\Gamma_j/C_j) \le 0$, the second term in the righthand side is non-negative, so the minimum is attained when the the vector m is null. When $\log(\Gamma_j/C_j) > 0$ the function is lower unbounded. It follows that

$$\min_{m\ge 0} L(m,\lambda) = \begin{cases} \sum_{r\in\mathcal{R}} n_r \log \Lambda_r & \text{if} \quad \Gamma_j \le C_j,\ \ \forall j \in \mathcal{J}, \\ -\infty & \text{otherwise.}\end{cases}$$

The Dual of (27) is given by looking for the maximum of the expression above and therefore it coincides with (31). Using the fact that they are Strong Lagrangian, we have that the upper-bound (27) and lower-bound (31) are tight. ∎

Now we are ready to prove the main result.

Proof of Theorem 1: (a) We check the detailed balance equations for this Markov Chain. Let $\tilde{n} = n + e_r - e_{\tilde{r}}$, with $r, \tilde{r} \in s$ and $r \ne \tilde{r}$, $w_s > 0$, we want to verify that

$$\frac{\pi(n)}{\pi(\tilde{n})} = \frac{T(\tilde{n}, n)}{T(n, \tilde{n})}\ ,$$

where $T(n,\tilde{n})$ is defined in (15). We have that

$$\begin{aligned}B(\tilde{n})\pi(\tilde{n}) &= \prod_{s\in\mathcal{S}} \frac{(w_s + |s|)!}{\prod_{u\in s}(\tilde{n}_u + 1)!} \\ &= \prod_{s\in\mathcal{S}} \frac{(w_s + |s|)!}{\prod_{u\in s}(n_u + 1)!} \frac{n_{\tilde{r}} + 1}{n_r + 2} \\ &= B(n)\pi(n) \frac{n_{\tilde{r}} + 1}{n_r + 2}\ ,\end{aligned}$$

implying that

$$\frac{\pi(n)}{\pi(\tilde{n})} = \frac{B(\tilde{n})}{B(n)} \times \frac{n_r + 2}{n_{\tilde{r}} + 1}\ . \tag{32}$$

Then we have

$$\frac{T(\tilde{n},n)}{T(n,\tilde{n})} = \frac{T(\tilde{n},\tilde{n}+e_{\tilde{r}}-e_r)}{T(n,n+e_r-e_{\tilde{r}})} = \frac{B(\tilde{n})}{B(\tilde{n}+e_{\tilde{r}})}\frac{B(n+e_r)}{B(n)}\frac{\tilde{n}_r+1}{n_{\tilde{r}}+1} = \frac{B(\tilde{n})}{B(n)}\frac{n_r+2}{n_{\tilde{r}}+1} \quad (33)$$

where in the last equation we used the fact that $\tilde{n}+e_{\tilde{r}} = n+e_r$ which implies $\tilde{n}_r+1 = n_r+1$. From the equality of (32) and (33) the result follows.

(b) Using formula (10) we get

$$\lim_{c\to\infty}\frac{1}{c}\log\pi(cn) = \lim_{c\to\infty} -\frac{1}{c}\log B(cn) + \sum_{s\in\mathcal{S}}\lim_{c\to\infty}\frac{1}{c}\log\binom{cw_s+|s|}{cn_r+1 : r\in s}.$$

It can be shown that this limit is the same as

$$\begin{aligned}\lim_{c\to\infty}\frac{1}{c}\log\pi(cn) &= \lim_{c\to\infty} -\frac{1}{c}\log B(cn) + \sum_{s\in\mathcal{S}}\lim_{c\to\infty}\frac{1}{c}\log\binom{cw_s}{cn_r : r\in s}\\ &= \sum_{r\in\mathcal{R}} n_r\log\Lambda_r^*(n) + \sum_{r\in R} n_r\log\frac{w_{s(r)}}{n_r}\\ &= -\sum_{r\in\mathcal{R}} n_r\log\left(\frac{n_r}{w_{s(r)}\Lambda_r^*(n)}\right),\end{aligned}$$

where we have used equations (24) and (25) and with $\Lambda_r^*(n)$ being the solution of the optimization problem (25). Then the result (17) readily follows.

(c) Writing $n\in w$ if the equations (2) are satisfied,

$$\max_{n\in w}\lim_{c\to\infty}\frac{1}{c}\log\pi(cn) = -\min_{\Lambda\geq 0}\min_{n\in w}\sum_{s\in\mathcal{S}}\sum_{r\in s} n_r\log\left(\frac{n_r}{w_s\Lambda_r}\right) \quad \text{subject to } \sum_{r\ni j}\Lambda_r\leq C_j,\ \forall j\in\mathcal{J}$$

and using $\Gamma_s = \sum_{r\in s}\Lambda_r$ we have

$$= -\min_{\substack{\Lambda\geq 0\\ n\in w}}\sum_{s\in\mathcal{S}}\left\{w_s\sum_{r\in\mathcal{R}}\frac{n_r}{w_s}\log\left(\frac{n_r/w_s}{\Lambda_r/\Gamma_s}\right) - w_s\log\Gamma_s\right\} \quad \text{subject to } \sum_{r\ni j}\Lambda_r\leq C_j,\ \forall j\in\mathcal{J}\ .$$

Noticing that the first term in the cost function is a relative entropy and that in the feasible domain it attains its minimal null value, we finally have

$$\max_{n\in w}\lim_{c\to\infty}\frac{1}{c}\log\pi(cn) = \max_{\Lambda\geq 0}\sum_{s\in\mathcal{S}} w_s\log\Gamma_s \quad \text{subject to } \sum_{r\ni j}\Lambda_r\leq C_j,\ \forall j\in\mathcal{J}$$

and the result follows.

■

ACKNOWLEDGMENT

The work of J. Anselmi has been supported by grant MTM2010-17405 (Ministerio de Ciencia e Innovación, Spain), grant SA-2012/00331 of the Department of Industry, Innovation, Trade and Tourism (Basque Government). The work of B. D'Auria has been partially supported by the Spanish Ministry of Education and Science Grants MTM2010-16519 and SEJ2007-64500. The work of Neil Walton is fund by the VENI research programme, which is financed by the Netherlands Organisation for Scientific Research (NWO).

REFERENCES

[1] C. N. Laws, "Resource pooling in queueing networks with dynamic routing," *Advances in Applied Probability*, vol. 24, no. 3, pp. pp. 699–726, 1992.

[2] R. J. Gibbens, F. P. Kelly, and P. B. Key, "Dynamic alternative routing," *Routing in communications networks*, pp. 13–47, 1995.

[3] D.-M. Chiu and R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," *Computer Networks and ISDN systems*, vol. 17, no. 1, pp. 1–14, 1989.

[4] V. Jacobson, "Congestion avoidance and control," in *Symposium proceedings on Communications architectures and protocols*, ser. SIGCOMM '88. New York, NY, USA: ACM, 1988, pp. 314–329. [Online]. Available: http://doi.acm.org/10.1145/52324.52356

[5] 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*, vol. 49, pp. 237–252, 1998.

[6] S. H. Low and D. E. Lapsley, "Optimization flow control, i: Basic algorithm and convergence," *IEEE/ACM TRANSACTIONS ON NETWORKING*, vol. 7, no. 6, pp. 861–874, 1999.

[7] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Trans. Networking*, vol. 8, pp. 556–567, 2000.

[8] R. Srikant, *The Mathematics of Internet Congestion Control*. Birkhauser, 2004.

[9] Y. Yi and M. Chiang, "Stochastic network utility maximisation," *European Transactions on Telecommunications*, vol. 19, no. 4, pp. 421–442, 2008.

[10] F. P. Kelly, "Charging and rate control for elastic traffic," *European Transactions on Telecommunications*, vol. 8, pp. 33–37, 1997.

[11] H. Han, S. Shakkottai, C. V. Hollot, R. Srikant, and D. Towsley, "Multi-path tcp: A joint congestion control and routing scheme to exploit path diversity on the internet," *IEEE/ACM Trans. Networking*, vol. 14, no. 6, pp. 1261–1271, 2006.

[12] F. P. Kelly and T. Voice, "Stability of end-to-end algorithms for joint routing and rate control," *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 2, pp. 5–12, 2005.

[13] X. Lin and N. Shroff, "Utility maximization for communication networks with multipath routing," *Automatic Control, IEEE Transactions on*, vol. 51, no. 5, pp. 766–781, May 2006.

[14] P. Key, L. Massouli, and D. Towsley, "Path selection and multipath congestion control," in *In Proc. IEEE INFOCOM*, 2007.

[15] D. Wischik, M. Handley, and M. B. Braun, "The resource pooling principle," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 5, pp. 47–52, 2008.

[16] C. Raiciu, C. Paasch, S. Barr, A. Ford, M. Honda, F. Duchene, O. Bonaventure, and M. Handley, "How hard can it be? designing and implementing a deployable multipath tcp," in *USENIX Symposium of Networked Systems Design and Implementation (NSDI'12), San Jose (CA)*, 2012.

[17] C. Raiciu, D. Wischik, and M. Handley, "Practical congestion control for multipath transport protocols," *University College of London Technical Report*, 2009.

[18] C. Raiciu, M. Handley, and D. Wischik, "Coupled congestion control for multipath transport protocols," *RFC 6356, http://tools.ietf.org/html/rfc6356*, 2011.

[19] S. H. Low, L. L. Peterson, and L. Wang, "Understanding tcp vegas: a duality model," *Journal of the ACM (JACM)*, vol. 49, no. 2, pp. 207–235, 2002.

[20] P. Viswanath, D. Tse, and R. Laroia, "Opportunistic beamforming using dumb antennas," *Information Theory, IEEE Transactions on*, vol. 48, no. 6, pp. 1277–1294, 2002.
[21] H. Kushner and P. Whiting, "Convergence of proportional-fair sharing algorithms under general conditions," *Wireless Communications, IEEE Transactions on*, vol. 3, no. 4, pp. 1250–1259, 2004.
[22] P. J. Schweitzer, "Approximate analysis of multiclass closed networks of queues," *Proceedings of the international conference on stochastic control and optimization*, 1979.
[23] N. S. Walton, "Proportional fairness and its relationship with multi-class queueing networks," *Ann. Appl. Probab.*, vol. 22, no. 6, pp. 2301–2333, 2009.
[24] J. Anselmi, B. D'Auria, and N. S. Walton, "Closed queueing networks under congestion: non-bottleneck independence and bottleneck convergence," *Mathematics of Operations Research (to appear)*.
[25] F. Baskett, K. M. Chandy, R. R. Muntz, and F. G. Palacios, "Open, closed, and mixed networks of queues with different classes of customers," *J. ACM*, vol. 22, no. 2, pp. 248–260, 1975.
[26] F. P. Kelly, *Reversibility and Stochastic Networks*. Chicester: Wiley, 1979.
[27] T. Bonald and A. Proutière, "On performance bounds for balanced fairness," *Performance Evaluation*, vol. 55, pp. 25–50, 2004.
[28] L. Kleinrock, "Time-shared systems: a theoretical treatment," *J. ACM*, vol. 14, no. 2, pp. 242–261, Apr. 1967.
[29] L. Massoulié and J. Roberts, "Bandwidth sharing: Objectives and algorithms," *IEEE Infocom 1999*, vol. 10, no. 3, pp. 320–328, 1999.
[30] T. Bonald and L. Massoulié, "Impact of fairness on internet performance," *Proc. of ACM Sigmetrics*, vol. 29, pp. 82–91, 2001.
[31] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*. New York, NY, USA: Cambridge University Press, 2006.
[32] N. Littlestone and M. Warmuth, "The weighted majority algorithm," in *Foundations of Computer Science, 1989., 30th Annual Symposium on*, oct-1 nov 1989, pp. 256 –261.
[33] B. Cohen, "Incentives build robustness in bittorrent," 2003.
[34] X. Yang and G. de Veciana, "Service capacity of peer to peer networks," in *INFOCOM*, 2004.