On efficient and fair scheduling in resource-sharing systems

I.M. Verloop

BCAM – Basque Center for Applied Mathematics
Bizkaia Technology Park, Building 500, E-48160 Derio, Spain

Abstract. In the context of communication networks and the Internet, resource-sharing systems have been widely studied. In this paper we give an overview of the main results obtained, and the main mathematical techniques used in the analysis of efficient and fair scheduling policies, or, more generally, in the performance evaluation and optimal control of resource-sharing systems. We discuss two main techniques extensively used in the literature: sample-path comparison and stochastic dynamic programming. However, often the models under consideration are extremelly complex, which makes exact analysis not feasible. This has motivated the development of asymptotic regimes. We present two of them: the heavy-traffic regime and a fluid scaling of the system. We then describe the so-called bandwidth-sharing networks, which model the interaction of data transfers in communication networks, and discuss results related to performance evaluation and optimal scheduling.
Keywords: optimal scheduling, stochastic comparison, dynamic programming, fluid scaling, heavy traffic, (discriminatory) processor sharing, α-fair bandwidth-sharing.
AMS 2000 subject classification: Primary 68M20, Secondary 90B15, 90B36

1 Introduction

Sharing resources among multiple users is common in daily life. One may think of resources such as lanes on a highway, agents in a call center, the processing capacity of a computer system, the available bandwidth in communication systems, or the transmission power of wireless base stations. In each of these situations, some scheduling mechanism regulates how the resources are shared among competing users. It is not always clear what the “best” way is to do this. Besides efficient use of the available resources in order to meet the demand, issues like fairness and the performance perceived by the users are important.

Our main motivation for studying scheduling in resource-sharing systems arises from communication networks. Internet traffic is expected to continue to grow, caused by increasing demand for, and availability of, multi-media applications. The available bandwidth may not grow at the same scale due to high cost or physical limitations. Therefore, the analysis of efficient and fair scheduling policies that regulate how resources are shared, is becoming more and more important. A crucial difference with the classical telephone network is that in the

Internet a resource can be shared simultaneously among many users present in the system. This feature triggered the need for new mathematical models. In particular, the random nature of arrivals of new users (data transfers), and of their corresponding service characteristics, motivates the study of queueing-theoretic models. An important contribution concerns the so-called bandwidth-sharing networks introduced in [47], which model the interaction of data transfers in communication networks.

The literature on resource-sharing systems, and bandwidth-sharing networks in particular, has exploded over the last years, which motivated us to write the present overview. We do not aim at providing a comprehensive overview, but rather a snapshot of the most relevant results and approaches used in the context of performance evaluation and optimal control of resource-sharing systems. In particular, we discuss two main techniques extensively used in the literature: stochastic comparison and stochastic dynamic programming. For cases when exact analysis of the stochastic system is not possible, we discuss two asymptotic regimes to which one can resort: the heavy-traffic regime and the fluid scaling. In addition, for the case of bandwidth-sharing networks we then present scheduling results existing in the literature.

The remainder of the paper is organized as follows. In Section 2 we present the essential characteristics of resource-sharing systems and introduce the notions of efficient and fair scheduling. In Section 3 we describe stochastic comparison techniques, stochastic dynamic programming, and two asymptotic regimes. In Section 4 we explain how resource sharing in communication networks motivates the study of bandwidth-sharing networks. In Sections 5 and 6, an overview is given on the scheduling results obtained for single-server systems and bandwidth-sharing networks, respectively.

2 Scheduling in resource-sharing systems

Deciding how to share the resources among users contending for service is a complicated task. This is in particular due to the following two elements. First of all, it is uncertain at what time new jobs arrive to the system and what amount or what kind of service they require. Second, the capacity of the resources is finite and there may be additional constraints on the way the resources can be shared among the various jobs. For example, some types of jobs might be processed faster by certain specialized resources, some types of jobs might need capacity from several resources simultaneously, etc.

In order to mathematically model the dynamic behavior of a resource-sharing system, we investigate queueing-theoretic models that capture the two elements as mentioned above. A queueing model consists of several servers with finite capacity, which can be allocated to users, possibly subject to additional constraints. The arrivals of new users and the amount and type of service they require, are described by stochastic processes.

The evolution of a queueing model is determined by the employed scheduling policy, which specifies at each moment in time how the capacity of the servers is shared among all users contending for it. An important body of the scheduling literature is devoted to seeking a policy that *optimizes the performance* of the queueing model. The latter may be expressed in terms of performance measures such as throughput, holding cost, user's delay, and the number of users in the system. Besides performance, another important notion is *fairness.* This relates to maintaining some level of "social justice", i.e., fairness in treatment of the users. Fairness is a subjective notion and much research has been devoted to developing quantitative measures [4].

In the remainder of this section we introduce in more detail the notions of optimal and fair scheduling. We make a distinction between the static regime and the dynamic regime, which are treated in Sections 2.1 and 2.2, respectively. In the static regime the population of users is fixed, while the dynamic regime allows for departures and arrivals of new users.

2.1 Static setting

For a given population of users, indexed by $i = 1, \dots, I$, we consider different ways to allocate the available capacity among the users. Let x_i be the rate allocated to user i and let $\boldsymbol{x} = (x_1, \dots, x_I)$ be the rate allocation vector. The set consisting of all feasible rate allocation vectors is denoted by S. Besides the fact that the capacity of the servers is finite, the shape of S is determined by additional constraints on the way the capacity of the servers can be shared among the users.

In a static setting it is natural to measure the performance in terms of the throughput $\sum_{i=1}^{I} x_i$. A feasible allocation that maximizes the throughput may be called optimal in the static setting. However, this optimal allocation does not guarantee that all users are allocated a strictly positive rate. It can be the case that some types of users obtain no capacity at all, which is highly unfair.

A commonly used definition of fairness has its origin in microeconomics. It relies on a social welfare function, which associates with each possible rate allocation the aggregate utility of the users [44]. A feasible allocation is called fair when it maximizes the social welfare function, i.e., an $\boldsymbol{x} \in S$ that solves

$$\max_{\boldsymbol{x} \in S} \sum_i U_i(x_i), \tag{1}$$

with $U_i(x_i)$ the utility of allocating rate x_i to user i. When the functions $U_i(\cdot)$ are strictly concave and the set S is convex and compact, the maximization problem has a unique solution. An important class of utility functions, see [50], is described by

$$U_i(x_i) = U_i^{(\alpha)}(x_i) = \begin{cases} w_i \log x_i & \text{if } \alpha = 1, \\ w_i \frac{x_i^{1-\alpha}}{1-\alpha} & \text{if } \alpha \in (0, \infty) \backslash \{1\}, \end{cases} \tag{2}$$

with $w_i > 0$ a weight assigned to user i, $i = 1, \dots, I$. The fact that these functions are increasing and strictly concave forces fairness between users: increasing the rate of a user that was allocated a relatively little amount, yields a larger improvement in the aggregate utility. The corresponding allocation that solves the optimization problem (1) is referred to as a *weighted α-fair allocation*. The resulting performance of this static fairness notion in a dynamic context is discussed in Section 6 for so-called bandwidth-sharing networks.

The class of weighted α-fair allocations contains some popular allocation paradigms when $w_i = 1$ for all i. For example, as $\alpha \to 0$ the resulting allocation achieves maximum throughput. Under suitable conditions, the Proportional Fair (PF) and max-min fair allocations (as defined in [13]) arise as special cases when $\alpha = 1$ and $\alpha \to \infty$, respectively, [50]. These notions of fairness have been widely used in the context of various networking areas.

The max-min fair allocation ($\alpha \to \infty$) is commonly seen as the most fair, since it maximizes the minimum rate allocated to any user. On the other extreme, maximizing the throughput ($\alpha \to 0$) can be highly unfair to certain users. The parameter α is therefore often referred to as the fairness parameter measuring the degree of fairness. Typically, realizing fairness and achieving a high throughput are conflicting objectives.

2.2 Dynamic setting

In practice, users depart upon service completion and new users arrive into the system over time. As mentioned previously, this can by modeled by queueing-theoretic models. In this section we discuss performance and fairness measures to evaluate different scheduling policies.

A key performance requirement in a dynamic setting is stability. Loosely speaking, stability means that the number of users in the system does not grow unboundedly or, in other words, that the system is able to handle all work requested by users. In resource-sharing systems the total used capacity typically depends on the scheduling decisions taken. Hence, stability conditions strongly depend on the policy employed. We therefore distinguish two types of conditions: (i) stability conditions corresponding to a *particular policy* and (ii) maximum stability conditions. The latter are conditions on the parameters of the *model* under which there exists a policy that makes the system stable.

Besides stability, another important performance measure concerns the number of users present in the system. We note that minimizing the total mean number of users is equivalent to minimizing the mean sojourn time, and thus equivalent to maximizing the user's throughput defined as the ratio between the mean service requirement and the mean sojourn time, cf. Little's law. As we will point out in Section 5.3, size-based scheduling policies, e.g. the Shortest Remaining Processing Time (SRPT) policy, are popular mechanisms for improving the performance by favoring smaller service requests over larger ones. However, this does not immediately carry over to resource-sharing systems in general. There are two effects to be taken into account. In the short term, it is preferable to favor "small" users that are likely to leave the system soon. In the long term however, a policy that uses the maximum capacity of the system at every moment in time, can empty the work in the system faster. When the total capacity used depends on the way the resources are shared among the users, the above-described goals can be conflicting.

The objective of optimal scheduling is often contradictory with fair scheduling. For example, giving preference to users based on their size (as is the case with SRPT) may starve users with large service requirements. Similar to the static setting, there is no universally accepted definition of fairness in the dynamic setting. We refer to [4, 72] for an overview on definitions existing in the literature.

In general, it is a difficult task to find fair or efficient policies for the dynamic setting. One may think of a policy as a rule that prescribes a rate allocation for each given population (as the population dynamically changes, the allocation changes as well). It is important to note that the use of fair or efficient allocations from the static setting does not give any guarantee for the behavior of the system in the dynamic setting. For example, maximizing the throughput at every moment in time might unnecessarily render the system unstable, and hence be certainly suboptimal in the dynamic context (e.g., see [15, Example 1]).

3 Techniques and methodology

In this section we describe some of the main techniques and tools used in the performance evaluation and optimization of resource-sharing systems. We first sketch the sample-path comparison and stochastic dynamic programming techniques, which are important tools to obtain comparison and optimality results, see Subsections 3.1 and 3.2. Unfortunately, it is not always within reach to explicitly analyze the original stochastic system. In such cases, one can resort to asymptotic regimes such as for example a fluid scaling and a heavy-traffic regime, which will be described in more detail in Subsections 3.3 and 3.4. Besides the above-mentioned techniques, we like to emphasize that there exist many other approaches that we do not comment on in this paper.

3.1 Sample-path comparison

Sample-path comparison is a useful tool in the control of queueing networks. A sample path corresponds to one particular realization of the stochastic process. As the name suggests, sample-path comparison techniques aim to compare, sample path by sample path, stochastic processes defined on a common probability space. This approach has been successfully applied in order to compare the performance of queueing systems under different policies.

Sample-path comparison is a special case of stochastic ordering. Two real-valued random variables X and Y are stochastically ordered, $X \leq_{st} Y$, when $\mathbb{P}(X > s) \leq \mathbb{P}(Y > s)$ for all $s \in \mathbb{R}$. Equivalently, $X \leq_{st} Y$ if and only if there exist two random variables X' and Y' defined on a common probability space, such that $X \overset{d}{=} X'$, $Y \overset{d}{=} Y'$, and $X' \leq Y'$ with probability one [51, 59]. Stochastic ordering of processes is then defined as follows: processes $\{X(t)\}_t$ and $\{Y(t)\}_t$ are stochastically ordered, $\{X(t)\}_t \leq_{st} \{Y(t)\}_t$, if and only if

$(X(t_1),\ldots,X(t_m)) \leq_{st} (Y(t_1),\ldots,Y(t_m))$ for any m and all $0 \leq t_1 < t_2 < \ldots < t_m < \infty$, [51]. Hence, if there exists a *coupling* $\{X'(t),Y'(t)\}_t$ such that $X'(t)$ and $Y'(t)$ are ordered sample-path wise and satisfy $\{X'(t)\}_t \overset{d}{=} \{X(t)\}_t$ and $\{Y'(t)\}_t \overset{d}{=} \{Y(t)\}_t$, then the processes $\{X(t)\}_t$ and $\{Y(t)\}_t$ are stochastically ordered.

In queueing networks, a rather intuitive way of coupling processes corresponding to different policies is to consider the same realizations of the arrival processes and service requirements. However, often more ingenious couplings are needed in order to obtain the desired comparison. We refer to [22, 39] for an overview on sample-path comparison methods and applications to optimal scheduling in queueing networks. In [45] (see also [41]) necessary and sufficient conditions on the transition rates are given in order for a stochastic order-preserving coupling to exist between two Markov processes.

3.2 Stochastic dynamic programming

Markov decision theory is a useful framework for modeling decision making in Markovian queueing systems. So-called stochastic dynamic programming techniques, based on Bellman's principle of optimality [11], allow to study a wide range of optimization problems. Although these techniques are well developed, only a few special queueing networks allow for an explicit solution of the optimal policy, see the survey on Markov decision problems (MDP's) in the control of queues [64]. Even when not explicitly solvable, characterizations of the optimal policies can often still be obtained. We refer to the textbooks [55, 59] for a full overview on MDP's.

In the simplest setting, an MDP is described as follows. At equidistant points in time, $t = 0, 1, \ldots$, a decision maker observes the state of the system, denoted by x, and chooses an action a from the action space $A(x)$. The state at the next decision epoch, denoted by y, is described by the transition probabilities $p(x, a, y)$ depending on the current state and the action chosen. There is a direct cost $C(x)$ each time state x is visited. The corresponding Markov decision chain can be described by $\{X_t, A_t\}_t$, where X_t and A_t represent the state and action at time t, respectively.

Markov decision theory allows optimization under finite-horizon, infinite-horizon discounted, and average-cost criteria. Here we focus on the latter, that is, we search for a policy that minimizes

$$\text{limsup}_{m\to\infty}\frac{1}{m}\mathbb{E}(\sum_{t=0}^{m-1} C(X_t)).$$

An average-cost optimal policy does not necessarily need to exist when the state space is infinite. There exist, however, sufficient conditions under which existence is guaranteed, see for example [61]. In that case, if $(g, V(\cdot))$ is a solution of the average-cost optimality equations

$$g + V(x) = C(x) + \min_{a\in A(x)} \sum_{y} p(x, a, y)V(y), \text{ for all states } x, \tag{3}$$

then g equals the minimum average cost and a stationary policy that realizes the minimum in (3) is average-cost optimal [59, Chapter V.2]. The function $V(\cdot)$ is referred to as the value function.

There are two main dynamic programming techniques: the policy iteration algorithm and the value iteration algorithm. Value iteration consists in analyzing the functions $V_m(\cdot)$, $m = 0, 1, \dots$, defined as

$$\begin{aligned} V_0(x) &= 0 \\ V_{m+1}(x) &= C(x) + \min_{a \in A(x)} \{\sum_y p(x,a,y) V_m(y)\},\ m = 0, 1, \dots. \end{aligned} \tag{4}$$

The functions $V_{m+1}(x)$ are interesting by themselves. They represent the minimum achievable expected cost over a horizon $m+1$ when starting in state x, i.e., the term $\mathbb{E}(\sum_{t=0}^{m} C(X_t)|X_0 = x)$ is minimized. Under certain conditions it holds that $V_m(\cdot) - mg \to V(\cdot)$ and $V_{m+1}(\cdot) - V_m(\cdot) \to g$ as $m \to \infty$ [30]. In addition, the minimizing actions in (4) converge to actions that constitute an average-cost optimal policy [30, 62]. As a consequence, if properties such as monotonicity, convexity, and submodularity [36] are satisfied for $V_m(\cdot)$, for all $m = 0, 1, \dots$, then the same is true for the value function $V(\cdot)$. Together with (3) this helps in the characterization of an optimal policy.

For a finite state space, the value iteration algorithm is useful to numerically determine an approximation of the average-cost optimal policy. This consists in recursively computing the functions $V_{m+1}(\cdot)$ until the difference between $\max_x(V_{m+1}(x) - V_m(x))$ and $\min_x(V_{m+1}(x) - V_m(x))$ is sufficiently small. When the state space is infinite, value iteration can be applied after appropriate truncation of the state space.

In a Markovian queueing system, without loss of generality, one can focus on policies that make decisions at transition epochs. The times between consecutive decision epochs are state-dependent and exponentially distributed. We can however equivalently consider the uniformized Markov process [55]: After uniformization, the transition epochs (including "dummy" transitions that do not alter the system state) are generated by a Poisson process of uniform rate. As such, the model can be reformulated as a discrete-time MDP, obtained by embedding at transition epochs.

Value iteration is used to find either (characterizations of) average-cost optimal policies (as described above), or stochastically optimal policies. The latter is done by setting the direct cost equal to zero, $C(\cdot) = 0$, and allowing a terminal cost at the end of the horizon, $V_0(\cdot) = \tilde{C}(\cdot)$. In that case, the term $V_{m+1}(x)$ represents the minimum achievable expected terminal cost when the system starts in state x at $m+1$ time units away from the horizon, i.e., the term $\mathbb{E}(\tilde{C}(X_{m+1})|X_0 = x)$ is minimized. Setting $\tilde{C}(\cdot) = \mathbf{1}_{(\tilde{c}(\cdot) > s)}$, with $\tilde{c}(\cdot)$ some cost function, this corresponds to minimizing $\mathbb{P}(\tilde{c}(X_m) > s | X_0 = x)$. The minimizing action in (4) is an optimal action at $m+1$ time units from the horizon. Hence, if the optimal actions do not depend on the time horizon m and on the value for s, then the corresponding stationary policy stochastically minimizes the cost $\tilde{c}(X_t)$ for all t.

3.3 Fluid scaling

The analysis of fluid-scaled processes has proved to be a powerful approach to investigate stability and optimal scheduling in queueing networks. A well-known result is [20], where stability of a multi-class queueing network is linked to that of the corresponding fluid-scaled model. For more details on fluid analysis, we refer to [19, 49, 58] and references therein. In this section we describe one fluid scaling in particular and focus on its application to optimal scheduling.

Consider a sequence of processes, indexed by $r \in \mathbb{N}$, such that $N_k^r(t)$ denotes the number of class-k users at time t in a queueing network with K classes of users when the initial queue lengths equal $N_k^r(0) = rn_k$, $n_k \geq 0$, $k = 1, \dots, K$. The fluid-scaled number of users is obtained when both time and space are scaled linearly, i.e.,

$$\overline{N}_k^r(t) := \frac{N_k^r(rt)}{r},\ k = 1, \dots, K.$$

We assume exponential inter-arrival times and service requirements, and consider non-anticipating policies. More general service requirements are allowed when posing additional conditions on the intra-class policies. Due to the functional strong law of large numbers [19], loosely speaking, each converging subsequence of $\overline{N}^r(t)$ converges to some process $\overline{N}(t)$, which has continuous characteristics and deterministic fluid input [20]. This limit is referred to as a *fluid limit.*

When it does not seem possible to derive optimal policies for the stochastic queueing network, fluid-scaling techniques can help to obtain approximations instead. In order to do so, a deterministic *fluid control model* is considered, which is a first-order approximation of the stochastic network by only taking into account the mean drifts. For example, in a multi-class single-server queue, on average λ_k class-k users arrive per time unit, and on average $\mu_k := 1/\mathbb{E}(B_k)$ class-k users depart when class k is given full priority. Hence, in this case the fluid control model is described by the process $(n_1(t), \dots, n_K(t))$ that satisfies

$$n_k(t) = n_k + \lambda_k t - \mu_k U_k(t), \quad \text{and} \quad n_k(t) \geq 0,\ t \geq 0,\ k = 1, \dots, K,$$

with $U_k(t) = \int_0^t u_k(v)\mathrm{d}v$ and where $u_k(\cdot)$ are feasible control functions, i.e.,

$$\sum_{k=1}^K u_k(v) \leq 1, \quad \text{and} \quad u_k(v) \geq 0,\ k = 1, \dots, K, \quad \text{for all} \quad v \geq 0.$$

We call a fluid control optimal when it minimizes $\int_0^\infty \sum_{k=1}^K c_k n_k(t)\mathrm{d}t$. The optimal trajectories in the fluid control model are denoted by $n_1^*(t), \dots, n_K^*(t)$. In the literature, optimal fluid controls have been obtained by using Pontryagin's maximum principle, see for example [7] or by solving a separated continuous linear program, see for example [71].

Motivated by the close relation between stability of the stochastic queueing network and its associated fluid model [20], researchers became interested in connections between optimal scheduling in the stochastic network and the far simpler fluid control problem [6, 49]. A crucial question is how to make a translation from the optimal control in the fluid model to a stable and efficient policy in the actual stochastic network. The optimal fluid control provides intuition on what a good policy in the stochastic network should try to achieve, however, difficulties can arise around the boundaries of the state space where a straightforward translation is not always adequate, see for example [25, 69].

Once a translation to the stochastic network has been made, one needs to show that this policy is close to optimal. Given that the system is stable, a policy π is called *asymptotically fluid-optimal* when

$$\lim_{r\to\infty} \mathbb{E}\Big(\int_0^D \sum_{k=1}^K c_k \overline{N}_k^{\pi,r}(t)\mathrm{d}t\Big) = \int_0^D \sum_{k=1}^K c_k n_k^*(t)\mathrm{d}t,$$

for all D sufficiently large. The main step to prove that a policy is asymptotically fluid-optimal consists in showing that the fluid limit of the stochastic network under this policy coincides with the optimal trajectories in the fluid control model, $n_1^*(t), \ldots, n_K^*(t)$. We refer to [8, 25, 42, 48, 69] for several examples of multi-class queueing networks for which asymptotically fluid-optimal policies have been derived.

Under suitable conditions, an average-cost optimal policy is asymptotically fluid-optimal [9], [25], [49, Theorem 10.0.5]. Unfortunately, no guarantee exists for the average cost of an asymptotically fluid-optimal policy. In fact, the asymptotically fluid-optimality definition aims at emptying the system efficiently starting from large initial state conditions, while average-cost optimality is concerned with the steady-state behavior of the system. In numerical experiments it has been observed that the average cost under asymptotically fluid-optimal policies is close to optimal. A first step towards a formal connection has been made in [48]. There, asymptotically fluid-optimal policies are proposed for which bounds on the average cost exist. In heavy traffic, these bounds (scaled with $1-\rho$) are tight and coincide with the optimal (scaled) average cost.

3.4 Heavy-traffic regime

Under a heavy-traffic regime the system is investigated as the traffic load approaches the capacity limit of the system. Analyzing the system in this regime can provide useful intuition as to how the system behaves when it is close to saturation. Typical heavy-traffic results relate to optimal control, queue length approximations, and state-space collapse (reduction in dimension of a multi-dimensional stochastic process).

The earliest heavy-traffic result is due to Kingman [35] who considered the steady-state behavior of a single-server queue under a non-preemptive policy (for service requirements with finite second moments). He proved that the steady-state queue length, scaled with $1-\rho$, converges in distribution to an exponential random variable as $\rho \to 1$. For PS or DPS the queue length is of the order $(1-\rho)^{-1}$ as well, see Section 5.2, but this is not true in general. For example, under LAS it can be either smaller or larger than $(1-\rho)^{-1}$, depending on the service requirement distribution [53].

So-called diffusion-scaled processes are commonly studied in a heavy-traffic setting to describe the *transient* behavior. A sequence of traffic parameters, indexed by r, is considered that converges at an appropriate rate to a heavily-loaded system. Let $N_k^r(t)$ denote the number of class-k users in the r-th system and define the diffusion-scaled number of users by

$$\hat{N}_k^r(t) := N_k^r(rt)/\sqrt{r}.$$

Due to the functional central limit theorem, the limit of such a diffusion-scaled process typically involves a reflected Brownian motion [19, 37]. We refer to [14, 32, 43] for several examples of queueing networks where diffusion-scaled processes have been analyzed.

For the single-server queue the diffusion scaling consists in letting

$$\lim_{r\to\infty} \rho^r = 1 \quad \text{such that} \quad \lim_{r\to\infty} \sqrt{r}\mu^r(\rho^r - 1) = \theta \in \mathbb{R}.$$

It is known that the diffusion-scaled number of users in a non-preemptive single-server system converges to a reflected Brownian motion with negative drift [19, 37]. Note that the stationary distribution of the latter process is exponential [19, Theorem 6.2], which coincides with the exponential distribution as mentioned earlier for the scaled steady-state process. For general networks, it is not obvious whether this interchange of the heavy-traffic and steady-state limits is allowed, and it has only been proved for some special cases, see for example [26, 68].

Optimal scheduling in heavy traffic is a well-studied field, typically focusing on policies with non-preemptive intra-class policies. Optimality results relate to diffusion-scaled networks, where the goal is to find a policy that minimizes some diffusion-scaled cost (either sample-path wise or on average) as $r \to \infty$ [10, 43, 75]. Asymptotically optimal policies in heavy traffic can serve as useful approximations for the optimal policy in the original system when the load is high.

4 Resource-sharing systems in communication networks

In this section we describe how resource sharing in communication networks naturally motivates the study of queueing systems. These systems are discussed in more detail in Sections 5 and 6.

The Internet is a packet-switched network, carrying data from source to destination. Each data transfer (flow) is split into several chunks (packets) that are routed individually over a common path from source to destination. Along this path, packets traverse various switches and routers that are connected by links. As a result, data flows contend for bandwidth on these links for the duration of the transfer. Data flows can be broadly categorized into streaming and elastic traffic. Streaming traffic, corresponding to real-time connections such as audio and video applications, is extremely sensitive to packet delays. It has an intrinsic rate requirement that needs to be met as it traverses the network in order to guarantee satisfactory quality. On the other hand, elastic traffic, corresponding to the transfer of digital documents like Web pages, e-mails, and data files, does not have a stringent rate requirement. Most of the elastic data traffic in the Internet nowadays is regulated by the Transmission Control Protocol (TCP) [31]. This end-to-end control dynamically adapts the transmission rate of packets based on the level of congestion in the network.

Typically, a given link is transmitting packets generated by several data flows. When viewing the system on a somewhat larger time scale (flow level), it can be argued that each data flow is transmitted as a continuous stream through the link, using only a certain fraction of the bandwidth. In case of homogeneous data flows and routers this implies that the bandwidth is equally shared among the data flows, i.e., the throughput of each data flow is C/n bits per second when there are n flows present on a link in isolation with bandwidth C.

Since the dynamics at the packet level occur at a much faster time scale than the arrivals and departures of data flows, it is reasonable to assume that the bandwidth allocation is adapted instantly after a change in the number of flows. Under this *time-scale separation*, the dynamic bandwidth sharing under TCP coincides with the so-called Processor Sharing (PS) queue, where each flow (or user) receives a fraction $1/n$ of the total service rate whenever there are n active flows [12, 52]. The actual bandwidth shares may in fact significantly differ among competing flows, either due to the heterogeneous end-to-end behavior of data flows or due to differentiation among data flows in routers. An appropriate model for this setting is provided by the Discriminatory Processor Sharing (DPS) queue where all flows share the bandwidth proportional to certain flow-dependent weights.

Instead of one link in isolation, a more realistic scenario is to consider *several* congested links in the network. Even though individual packets travel across the network on a hop-by-hop basis, when we view the system behavior on a somewhat larger time scale, a data flow claims roughly equal bandwidth on each of the links along its source-destination path *simultaneously.* A mathematical justification for the latter can be found in [70]. The class of weighted α-fair allocations, as will be described in Section 2.1, is commonly accepted to model the flow-level bandwidth allocation as realized by packet-based protocols. For example, the α-fair allocation with $\alpha = 2$ and weights w_k inversely proportional to the source-destination distance, has been proposed as an appropriate model for TCP [54]. In addition, for any α-fair allocation (defined at flow level) there exists a distributed mechanism at packet level that achieves the α-fair allocation [34, 50, 63].

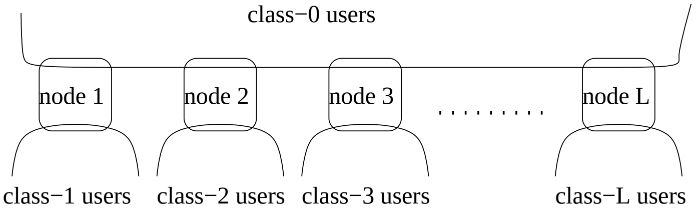

Fig. 1. Linear bandwidth-sharing network with $L + 1$ classes of data flows.

Under the time-scale separation assumption, bandwidth-sharing networks as considered in [47] provide a natural way to describe the *dynamic* flow-level interaction among elastic data flows. See also [33, 70], where bandwidth-sharing networks are obtained as limits of packet-switched networks. In bandwidth-sharing networks, a flow requires simultaneously the same amount of capacity from all links along its source-destination path. An example of a bandwidth-sharing network is depicted in Figure 1. Here flows (or users) of class 0 request the same amount of bandwidth from all links simultaneously and in each link there is possibly cross traffic present from other routes. This interaction between active flows can cause inefficient use of the available capacity. For example, when there are flows of class 0 present, the capacity of a certain link with no cross traffic may not be fully used when the capacity of another link is already exhausted.

5 The single-server system

The classical single-server system (G/G/1 queue) consists of a single queue and a single server with capacity equal to one. This may model for instance a link in isolation in a communication network. Users arrive one by one in the system and each user requires a certain amount of service after which it leaves the system. Let $1/\lambda$ denote the mean inter-arrival time. The service requirement of a user, B, represents the amount of time that the server needs to serve the user when it would devote its full capacity to this user. The capacity of the server may be shared among multiple users at the same time.

In a single-server queue the focus is on work-conserving scheduling policies, that is, policies that always use the full capacity of the server whenever there is work in the system. Obviously, the total unfinished work in the system, the workload, is independent of the work-conserving policy employed. In addition, any work-conserving policy in a G/G/1 queue is stable as long as the traffic load $\rho := \lambda\mathbb{E}(B)$ is strictly less than one.

While the workload process and the stability condition are independent of the employed work-conserving policy, this is not the case for the evolution of the queue length process and, hence, for most performance measures. There is a vast body of literature on the analysis of scheduling policies in the single-server queue. In the remainder of this section we first give a description of two time-sharing policies: PS and DPS, which provide a natural approach for modeling the flow-level performance of TCP. We conclude this section with an overview of optimal size-based scheduling in the single-server queue.

5.1 Processor sharing

As described in Section 4, PS is a useful paradigm for evaluating the dynamic behavior of elastic data flows competing for bandwidth on a single link. When there are n users in the system, each user receives a fraction $1/n$ of the capacity of the server.

In the literature, there is a large body of analytical results available for the PS queue. For full details and references on the PS queue we refer to [52]. One of the most important results states that when the arrival process is Poisson and $\rho < 1$, the stationary distribution of the queue length exists and is insensitive to the service requirement distribution apart from its mean. More precisely, the queue length in steady state has a geometric distribution with parameter ρ, i.e., the probability of having n users in the queue is equal to $(1-\rho)\rho^n$, $n = 0, 1, \dots$.

5.2 Discriminatory processor sharing

DPS, introduced by Kleinrock, is a multi-class generalization of PS. By assigning different weights to users from different classes, DPS allows class-based differentiation. Let K be the number of classes, and let w_k be the weight associated with class k, $k = 1, \dots, K$. Whenever there are n_k class-k users present, $k = 1, \dots, K$, a class-l user is served at rate

$$\frac{w_l}{\sum_{k=1}^{K} w_k n_k}, \quad l = 1, \dots, K.$$

In case of unit weights, the DPS policy reduces to the PS policy. Despite the similarity, the analysis of DPS is considerably more complicated compared to PS. The geometric queue length distribution for PS does not have any counterpart for DPS. In fact, the queue lengths under DPS are sensitive with respect to higher moments of the service requirements [17]. Despite this fact, in [5] the DPS model was shown to have finite mean queue lengths regardless of the higher-order moments of the service requirements.

The seminal paper [24] provided an analysis of the mean sojourn time conditioned on the service requirement by solving a system of integro-differential equations. As a by-product, it was shown that a user's slowdown behaves like the user's slowdown under PS $(1/(1-\rho))$, as its service requirement grows large, see also [5]. Another asymptotic regime under which the DPS policy has been studied is the heavy-traffic regime, i.e., $\rho \uparrow 1$. For Poisson arrivals and phase-type distributed service requirements, in [68] the authors showed a state space collapse result for the scaled joint queue length vector, see also [56]. Let N_k denote the number of class-k users in steady state, then

$$(1-\rho)(N_1, N_2, \dots, N_K) \overset{d}{\to} X \cdot (\frac{\hat{\rho}_1}{w_1}, \frac{\hat{\rho}_2}{w_2}, \dots, \frac{\hat{\rho}_K}{w_K}), \quad \text{as} \quad \rho \uparrow 1,$$

where $\overset{d}{\to}$ denotes convergence in distribution, $\hat{\rho}_k := \lim_{\rho\uparrow 1} \rho_k$, $k = 1, \dots, K$, and X is an exponentially distributed random variable. For more results on DPS under heavy traffic and several other limiting regimes we refer to the overview paper [3].

5.3 Optimal scheduling

There exists a vast amount of literature devoted to optimal scheduling in single-server systems. A well-known optimality result concerns the Shortest Remaining

Processing Time (SRPT) policy, which serves at any moment in time the user with the shortest remaining service requirement. In [60] it is proved that SRPT minimizes sample-path wise the number of users present in the single-server system.

SRPT relies on the knowledge of the (remaining) service requirements of the users. Since this information might be impractical to obtain, a different strand of research has focused on finding optimal policies among the so-called non-anticipating policies. These policies do not use any information based on the (remaining) service requirements, but they do keep track of the attained service of users present in the system. Popular policies like First Come First Served (FCFS), Least Attained Service (LAS), PS and DPS are all non-anticipating. Among all non-anticipating policies, the mean number of users is minimized under the Gittins rule [2, 28], which is proved by stochastic dynamic programming. The Gittins rule simplifies to LAS and FCFS for particular cases of the service requirements [2].

The LAS policy, also known as Foreground-Background, which serves at any moment in time the user(s) with the least attained service, has been extensively studied. For an overview we refer to [53]. In case of Poisson arrivals, LAS stochastically minimizes the number of users in the system if and only if the service requirement distribution has a decreasing failure rate (DFR) [2, 57]. This sample-path comparison is obtained with an interchange argument. It uses the fact that under the DFR assumption, as a user obtains more service, it becomes less likely that it will leave the system soon. Therefore, prioritizing the newest users is optimal.

For a service requirement distribution with an increasing failure rate (IFR), any non-preemptive policy, in particular FCFS, stochastically minimizes the number of users in the system [57]. A policy is non-preemptive when at most one user is served at a time and once a user is taken into service this service will not be interrupted. This result can be understood from the fact that under the IFR assumption, as a user obtains more service, it becomes more likely that it will leave the system soon.

We finish this section with an important result for the multi-class single-server system. We associate with each user class a cost c_k and let $\mu_k := 1/\mathbb{E}(B_k)$, where B_k denotes the class-k service requirement. A classical result states that the so-called $c\mu$-rule, the policy that gives strict priority to classes in descending order of $c_k\mu_k$, minimizes the mean holding cost $\mathbb{E}(\sum_k c_k N_k)$. In the literature this has been proved with several approaches, e.g. sample-path comparison techniques, the achievable region approach, and interchange arguments. It holds for the M/G/1 queue among all non-preemptive non-anticipating policies [27] and for the G/M/1 queue among all non-anticipating policies [39]. The optimality of the $c\mu$-rule can be understood from the fact that $1/\mu_k$ coincides in both settings with the expected remaining service requirement of a class-k user *at a scheduling decision epoch.* Hence, at every moment in time, the user with the smallest weighted expected remaining service requirement is served.

6 Bandwidth-sharing networks

Bandwidth-sharing networks provide a modeling framework for the dynamic interaction of data flows in communication networks, where a flow claims roughly equal bandwidth on each of the links along its path, as described in Section 4. Mathematically, a bandwidth-sharing network can be described as follows. It consists of a finite number of nodes, indexed by $l = 1, \dots, L$, which represent the links of the network. Node l has finite capacity C_l. There are K classes of users. Associated with each class is a route that describes which nodes are needed by the users from this class. Let A be the $L \times K$ incidence matrix containing only zeros and ones, such that $A_{lk} = 1$ if node l is used by users of class k and $A_{lk} = 0$ otherwise. Each user requires simultaneously the same capacity from all the nodes on its route. Let s_k denote the aggregate rate allocated to all class-k users. The total capacity used from node l is $\sum_{k=1}^{K} A_{lk} s_k$. Hence, a rate allocation is feasible when $\sum_{k=1}^{K} A_{lk} s_k \leq C_l$, for all $l = 1, \dots, L$.

An example of a bandwidth-sharing network is the so-called linear network as depicted in Figure 1. It consists of L nodes and $K = L+1$ classes, for convenience indexed by $j = 0, 1, \dots, L$. Class-0 users require the same amount of capacity from all L nodes simultaneously while class-i users, $i = 1, \dots, L$, require service at node i only. The $L \times (L+1)$ incidence matrix of the linear network is

$$A = \begin{pmatrix} 1\,1\,0\,0 \dots 0 \\ 1\,0\,1\,0 \dots 0 \\ 1\,0\,0\,1 \dots 0 \\ \vdots\,\vdots\,\vdots\,\vdots\,\ddots\,\vdots \\ 1\,0\,0\,0 \dots 1 \end{pmatrix},$$

hence the capacity constraints are $s_0 + s_i \leq C_i$, $i = 1, \dots, L$.

An inherent property of bandwidth-sharing networks is that, given a population of users, the total used capacity of the network, $\sum_{k=1}^{K} \sum_{l=1}^{L} A_{lk} s_k$, is not necessarily equal to the total available capacity of the network, $\sum_{l=1}^{L} C_l$. This may even be the case when we restrict ourselves to Pareto-efficient allocations, i.e., allocations where the rate allocated to a class cannot be increased without reducing the rate allocated to another class. For example, one may think of the linear network where at a certain moment in time there are no users of class L present. The Pareto-efficient allocation that serves class 0 makes *full* use of the capacity of the network. However, the Pareto-efficient allocation that serves classes 1 until $L-1$ uses only the capacity of the first $L-1$ nodes, and leaves the capacity of node L unused.

The maximum stability conditions of a bandwidth-sharing network are $\sum_{k=1}^{K} A_{lk} \rho_k < C_l$, for all $l = 1, \dots, L$, i.e., the offered load in each node is strictly less than its available capacity. In general, the stability conditions corresponding to a specific policy can be more restrictive than the maximum stability conditions. This becomes for example apparent in the linear network with unit capacities, $C_l = 1$, $l = 1, \dots, L$. The policy that gives preemptive priority to class-0 users is stable under the maximum stability conditions, $\rho_0 + \rho_i < 1$, for all $i = 1, \ldots, L$. However, the Pareto-efficient policy that gives preemptive priority to classes 1 through L is stable if and only if $\rho_0 < \prod_{i=1}^{L}(1-\rho_i)$, which is a more stringent condition. Note that in [29] it is shown that this instability effect can be avoided. It is proved that any Pareto-efficient policy in a bandwidth-sharing network is stable, provided that it is suitably modified when the number of users in a class becomes too small.

The fact that in a bandwidth-sharing network the total capacity used depends on the scheduling decisions taken, complicates the task of optimal scheduling. Consider for example the linear network. Using sample-path approaches and dynamic programming techniques it was proved in [69] that for certain parameter settings simple priority rules minimize the weighted number of users, while for the remaining cases an optimal policy can be characterized by "switching curves", i.e., the policy dynamically switches between several priority rules. An exact characterization of these curves is not possible, however, by studying the corresponding fluid model it was found that policies with linear or square-root switching curves are asymptotically fluid-optimal.

6.1 Weighted α-fair sharing

A popular class of policies studied in the context of bandwidth-sharing networks are weighted α-fair bandwidth-sharing policies. In state $\boldsymbol{n} = (n_1, \ldots, n_K)$, with n_k the number of class-k users, a weighted α-fair policy allocates $s_k(\boldsymbol{n})/n_k$ to each class-k user, with $(s_1(\boldsymbol{n}), \ldots, s_K(\boldsymbol{n}))$ the solution of the utility optimization problem

$$\begin{aligned} \text{maximize} \quad & \sum_{k=1}^{K} n_k U_k^{(\alpha)}\left(\frac{s_k}{n_k}\right), \\ \text{subject to} \quad & \sum_{k=1}^{K} A_{lk} s_k \le C_l, \quad l = 1, \ldots, L, \end{aligned} \tag{5}$$

and $U_k^{(\alpha)}(\cdot)$, $\alpha > 0$, as defined in (2).

For a network consisting of one node, the weighted α-fair policy reduces to the DPS policy with weights $w_k^{1/\alpha},\ k = 1, \ldots, K$. For the linear network with unit capacities, the weighted α-fair rate allocation is given by

$$s_0(\boldsymbol{n}) = \frac{(w_0 n_0^{\alpha})^{1/\alpha}}{(w_0 n_0^{\alpha})^{1/\alpha} + (\sum_{i=1}^{K} w_i n_i^{\alpha})^{1/\alpha}}, \quad s_i(\boldsymbol{n}) = \mathbf{1}_{(n_i>0)} \cdot (1 - s_0(\boldsymbol{n})), \quad i \neq 0,$$

see [15]. For grid and cyclic networks, as described in [15], the weighted α-fair rate allocations can be found in closed form as well.

An important property of weighted α-fair policies in bandwidth-sharing networks concerns stability, which has been investigated by means of the fluid-scaling approach [20]. In [15] it is proved that when the service requirements and the inter-arrival times are exponentially distributed, weighted α-fair bandwidth-sharing policies ($\alpha > 0$) achieve stability under the maximum stability conditions, $\sum_{k=1}^{K} A_{lk}\rho_k < C_l$, for all $l = 1, \ldots, L$, see also [66, 74]. For phase-type distributed service requirements, maximum stability is proved for the Proportional Fair (PF) policy ($\alpha = 1$ and unit weights) [46]. In [16, 40] stability is investigated when the set of feasible allocations is not given by (5). The authors of [16] prove that for any convex set of feasible allocations, PF and the max-min fair policy ($\alpha \to \infty$ and unit weights) provide stability under the maximum stability conditions. In [40] stability is investigated when the set of feasible allocations is non-convex or time-varying. It is shown that the stability condition depends on the parameter α, and that for some special cases the stability condition becomes tighter as α increases.

6.2 Flow-level performance of weighted α-fair sharing

Very little is known about the way α-fair sharing affects the performance perceived by users. Closed-form analysis of weighted α-fair policies has mostly remained elusive, except for so-called hypercube networks (a special case is the linear network) with unit capacities. For those networks, the steady-state distribution of the numbers of users of the various classes under PF is of product form and insensitive to the service requirement distributions [15, 17]. For all other situations, the distributions of the numbers of users under weighted α-fair policies are sensitive with respect to higher moments of the service requirement distributions [17]. In [18], insensitive stochastic bounds on the number of users in any class are derived for the special case of tree networks. A related result can be found in [65] where the authors focus on exponentially distributed service requirements and obtain an upper bound on the total mean number of users under PF. In [67] stochastic comparison results on the workload and the number of users are obtained in the special case of a linear network. When in addition the service requirements are exponentially distributed, monotonicity of the mean total number of users is established with respect to the fairness parameter α and the relative weights.

A powerful approach to study the complex dynamics under weighted α-fair policies is to investigate asymptotic regimes. For example, in [23] the authors study the max-min fair policy under a large-network scaling and give a mean-field approximation. Another asymptotic regime is the heavy-traffic setting where the load on at least one node is close to its capacity. In this regime, the authors of [32, 75] study weighted α-fair policies under fluid and diffusion scalings and investigate diffusion approximations for the numbers of users of the various classes. In addition, when the load on *exactly* one node tends to its capacity, the authors of [75] identify a cost function that is minimized in the diffusion scaling by the weighted α-fair policy. For the linear network, heavy-traffic approximations for the scaled mean numbers of users are derived in [38]. Bandwidth-sharing networks in an overloaded regime, that is when the load on one or several of the nodes exceeds the capacity, are considered in [21]. The growth rates of the numbers of users of the various classes under weighted α-fair policies are characterized by a fixed-point equation.

Motivated by the optimality results in the single-server system, research has focused on improving weighted α-fair policies using performance benefits from size-based scheduling. In [1] the authors propose to deploy SRPT as intra-class policy, instead of PS, in order to reduce the number of users in each class. Another approach is taken in [73], where weighted α-fair policies are studied with dynamic per-user weights that depend on the remaining service requirements. Simulations show that the performance can improve considerably over the standard α-fair policies.

Acknowledgements

The author is grateful to her Ph.D. supervisors S.C. Borst and R. Núñez Queija for many fruitful discussions.

References

1. Aalto, S., and Ayesta, U.: SRPT applied to bandwidth-sharing networks. Annals of Operations Research **170** (2009) 3–19
2. Aalto, S., Ayesta, U., and Righter, R.: On the Gittins index in an M/G/1 queue. Queueing Systems **63** (2009) 437–458
3. Altman, E., Avrachenkov, K.E., and Ayesta, U.: A survey on discriminatory processor sharing. Queueing Systems **53** (2006) 53–63
4. Avi-Itzhak, B., Levy, H., and Raz, D.: Quantifying fairness in queuing systems: principles, approaches, and applicability. Probability in the Engineering and Informational Sciences **22** (2008) 495–517
5. Avrachenkov, K.E., Ayesta, U., Brown, P., and Núñez-Queija, R.: Discriminatory processor sharing revisited. In Proceedings of INFOCOM. Miami FL, USA (2005)
6. Avram, F.: Optimal control of fluid limits of queueing networks and stochasticity corrections. Mathematics of stochastic manufacturing systems. Lectures in Applied Mathematics **33** (1997) 1–36
7. Avram, F., Bertsimas, D., and Ricard, M.: Fluid models of sequencing problems in open queueing networks: An optimal control approach. IMA Volumes in Mathematics and its Applications **71** (1995) 199–234
8. Bäuerle, N.: Asymptotic optimality of tracking policies in stochastic networks. Annals of Applied Probability **10** (2000) 1065–1083
9. Bäuerle, N.: Optimal control of queueing networks: An approach via fluid models. Advances in Applied Probability **34** (2002) 313–328
10. Bell, S.L., and Williams, R.J.: Dynamic scheduling of a system with two parallel servers in heavy traffic with resource pooling: Asymptotic optimality of a threshold policy. Annals of Applied Probability **11** (2001) 608–649
11. Bellman, R.E.: Dynamic Programming. Princeton University Press, 1957
12. Ben Fredj, S., Bonald, T., Proutière, A., Régnié, G., and Roberts, J.W.: Statistical bandwidth sharing: A study of congestion at flow level. In Proceedings of ACM SIGCOMM. San Diego CA, USA (2001) 111–122
13. Bertsekas, D., and Gallager, R.: Data Networks. Prentice-Hall, 1987

14. Bhardwaj, S., and Williams, R.J.: Diffusion approximation for a heavily loaded multi-user wireless communication system with cooperation. Queueing Systems **62** (2009) 345–382
15. Bonald, T., and Massoulié, L.: Impact of fairness on Internet performance. In Proceedings of ACM SIGMETRICS/Performance. Boston MA, USA (2001) 82–91
16. Bonald, T., Massoulié, L., Proutière, A., and Virtamo, J.: A queueing analysis of max-min fairness, proportional fairness and balanced fairness. Queueing Systems **53** (2006) 65–84
17. Bonald, T., and Proutière, A.: Insensitive Bandwidth Sharing in Data Networks. Queueing Systems **44** (2003) 69–100
18. Bonald, T., and Proutière, A.: On Stochastic Bounds for Monotonic Processor Sharing Networks. Queueing Systems **47** (2004) 81–106
19. Chen, H., and Yao, D.D.: Fundamentals of Queueing Networks: Performance, Asymptotics, and Optimization. Springer-Verlag, New York, 2001
20. Dai, J.G.: On positive Harris recurrence of multiclass queueing networks: A unified approach via fluid limit models. Annals of Applied Probability **5** (1995) 49–77
21. Egorova, R., Borst, S.C., and Zwart, A.P.: Bandwidth-sharing networks in overload. Performance Evaluation **64** (2007) 978–993
22. El-Taha, M., and Stidham, S.: Sample-path analysis of queueing systems. Kluwer Academic Publishers, 1999
23. Fayolle, G., Fortelle de la, A., Lasgouttes, J.M., Massoulié, L., and Roberts, J.W.: Best-effort networks: Modeling and performance analysis via large networks asymptotics. In Proceedings of INFOCOM. Anchorage AK, USA (2001)
24. Fayolle, G., Mitrani, I., and Iasnogorodski, R.: Sharing a Processor Among Many Job Classes. Journal of the ACM **27** (1980) 519–532
25. Gajrat, A., and Hordijk, A.: Fluid approximation of a controlled multiclass tandem network. Queueing Systems **35** (2000) 349–380
26. Gamarnik, D., and Zeevi, A.: Validity of heavy traffic steady-state approximations in generalized Jackson networks. Annals of Applied Probability **16** (2006) 56–90
27. Gelenbe, E., and Mitrani, I.: Analysis and Synthesis of Computer Systems. Academic Press, London, 1980
28. Gittins, J.C.: Multi-Armed Bandit Allocation Indices. Wiley, Chichester, 1989
29. Hansen, J., Reynolds, C., Zachary, S.: Stability of processor sharing networks with simultaneous resource requirements. Journal of Applied Probability **44** (2007) 636–651
30. Hernández-Lerma, O., and Lasserre, J.B.: Discrete-Time Markov Control Processes: Basic Optimality Criteria. Springer-Verlag, New york, 1996
31. Jacobson, V.: Congestion avoidance and control. In Proceedings of ACM SIGCOMM (1988) 314–329
32. Kang, W.N., Kelly, F.P., Lee, N.H., and Williams, R.J.: State space collapse and diffusion approximation for a network operating under a fair bandwidth sharing policy. Annals of Applied Probability **19** (2009) 1719–1780
33. Kelly, F.P., Massoulié, L., and Walton, N.S.: Resource pooling in congested networks: Proportional fairness and product form. Queueing Systems **63** (2009) 165–194
34. Kelly, F.P., Maulloo, A., and Tan, D.: Rate control in communication networks: Shadow prices, proportional fairness and stability. Journal of the Operational Research Society **49** (1998) 237–252
35. Kingman, J.F.C.: On queues in heavy traffic. Journal of the Royal Statistical Society: Series B **24** (1962) 383–392

36. Koole, G.M.: Monotonicity in Markov Reward and Decision Chains: Theory and Applications. Foundations and Trends in Stochastic Systems **1** (2006) 1–76
37. Kushner, H.J.: Heavy Traffic Analysis of Controlled Queueing and Communication Networks. Springer-Verlag, New york, 2001
38. Lieshout, P., Borst, S.C., and Mandjes, M.: Heavy-Traffic Approximations for Linear Networks Operating under Alpha-Fair Bandwidth-Sharing Policies. In Proceedings of VALUETOOLS (2006)
39. Liu, Z., Nain, Ph., and Towsley, D.: Sample path methods in the control of queues. Queueing Systems **21** (1995) 293–335
40. Liu, J., Proutière, A., Yi, Y., Chiang, M., and Poor, V.H.: Flow-level stability of data networks with non-convex and time-varying rate regions. In Proceedings of ACM SIGMETRICS. San Diego, USA (2007) 239–250
41. López, F.J., and Sanz, G.: Markovian couplings staying in arbitrary subsets of the state space. Journal Applied Probability **39** (2002) 197–212
42. Maglaras, C.: Discrete-review policies for scheduling stochastic networks: Trajectory tracking and fluid-scale asymptotic optimality. Annals of Applied Probability **10** (2000) 897–929
43. Mandelbaum, A., and Stolyar, A.L.: Scheduling flexible servers with convex delay costs: Heavy-traffic optimality of the generalized cμ-rule. Operations Research **52** (2004) 836–855
44. Mas-Colell, A., Whinston, M.D., and Green, J.R.: Microeconomic Theory. Oxford University Press, New York, 1995
45. Massey, W.A.: Stochastic orderings for Markov processes on partially ordered spaces. Mathematics of Operations Research **12** (1987) 350–367
46. Massoulié, L.: Structural properties of proportional fairness: Stability and insensitivity. Annals of Applied Probability **17** (2007) 809–839
47. L. Massoulié and J.W. Roberts: Bandwidth sharing and admission control for elastic traffic. Telecommunication Systems **15** (2000) 185–201
48. Meyn, S.P.: Dynamic safety-stocks for asymptotic optimality in stochastic networks. Queueing Systems **50** (2005) 255–297
49. Meyn, S.P.: Control Techniques for Complex Networks. Cambridge University Press, New York, 2008
50. Mo, J., and Walrand, J.: Fair end-to-end window-based congestion control. IEEE/ACM Transactions on Networking **8** (2000) 556–567
51. Müller, A.M., and Stoyan, D.: Comparison methods for stochastic models and risks. J. Wiley & Sons, 2002
52. Núñez-Queija, R.: Processor-Sharing Models for Integrated-Services Networks. Ph.D. Thesis Eindhoven University of Technology, 2000
53. Nuyens, M., and Wierman, A.: The foreground-background queue: A survey. Performance Evaluation **65** (2008) 286–307
54. Padhye, J., Firoiu, V., Towsley, D., and Kurose, J.: Modeling TCP Reno performance: A simple model and its empirical validation. IEEE/ACM Transactions on Networking **8** (2000) 133–145
55. Puterman, M.L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley, New York, 1994
56. Rege, K.M., and Sengupta, B.: Queue length distribution for the discriminatory processor-sharing queue. Operations Research **44** (1996) 653–657
57. Righter, R., and Shanthikumar, J.G.: Scheduling multiclass single server queueing systems to stochastically maximize the number of successful departures. Probability in the Engineering and Informational Sciences **3** (1989) 323–333

58. Robert, P.: Stochastic Networks and Queues. Springer-Verlag, New York, 2003
59. Ross, S.M.: Introduction to Stochastic Dynamic Programming. Academic Press, New York, 1983
60. Schrage, L.E., and Miller, L.W.: The Queue M/G/1 with the Shortest Remaining Processing Time Discipline. Operations Research **14** (1966) 670–684
61. Sennott, L.I.: Average reward optimization theory for denumerable state spaces. Handbook of Markov Decision Processes, eds. E.A. Feinberg and A. Shwartz. Kluwer (2002) 153–172
62. Sennott, L.I.: Value iteration in countable state average cost Markov decision processes with unbounded costs. Annals of Operations Research **28** (2005) 261–271
63. Srikant, R.: The Mathematics of Internet Congestion Control. Birkhäuser, Boston, 2004
64. Stidham Jr., S., and Weber, R.R.: A survey of Markov decision models for control of networks of queues. Queueing Systems **13** (1993) 291–314
65. Tan, B., Ying, L., and Srikant, R.: Short-term fairness and long-term QoS. In Proceedings of CISS Conference on Information Sciences and Systems. Princeton University (2008)
66. Veciana De, G., Lee, T.-L., and Konstantopoulos, T.: Stability and performance analysis of networks supporting elastic services. IEEE/ACM Transactions on Networking **9** (2001) 2–14
67. Verloop, I.M., and Ayesta, U., and Borst, S.C.: Monotonicity properties for multiclass queueing systems. To appear in Discrete Event Dynamic Systems (2010), DOI: 10.1007/s10626-009-0069-4
68. Verloop, I.M., and Ayesta, U., and Núñez-Queija, R.: Heavy-traffic analysis of a multiple-phase network with discriminatory processor sharing. To appear in Operations Research (2010)
69. Verloop, I.M., and Núñez-Queija, R.: Assessing the efficiency of resource allocations in bandwidth-sharing networks. Performance Evaluation **66** (2009) 59–77
70. Walton, N.S.: Proportional Fairness and its Relationship with Multi-class Queueing Networks. Royal Statistical Society Lecture Notes Series **19** (2009) 2301–2333
71. Weiss, G.: Optimal Draining of Fluid Re-Entrant Lines: Some Solved Examples. Annals of Applied Probability **4** (1996) 19–34
72. Wierman, A.: Fairness and classifications. Performance Evaluation Review **34** (2007) 4–12
73. Yang, S., Veciana De, G.: Enhancing both network and user performance for networks supporting best-effort traffic. IEEE/ACM Transactions on Networking **12** (2004) 349–360
74. Ye, H.-Q.: Stability of data networks under an optimization-based bandwidth allocation. IEEE Transactions on Automatic Control **48** (2003) 1238–1242
75. Ye, H.-Q., and Yao, D.D.: Heavy-traffic optimality of a stochastic network under utility-maximizing resource allocation. Operations Research **56** (2008) 453–470