

Scheduling Policies for Single-Hop Networks with Heavy-Tailed Traffic

Mihalis G. Markakis, Eytan H. Modiano and John N. Tsitsiklis

Abstract—In the first part of the paper, we study the impact of scheduling, in a setting of parallel queues with a mix of heavy-tailed and light-tailed traffic. We analyze queue-length unaware scheduling policies, such as round-robin, randomized, and priority, and characterize their performance. We prove the queue-length instability of Max-Weight scheduling, in the presence of heavy-tailed traffic. Motivated by this, we analyze the performance of Max-Weight- α scheduling, and establish conditions on the α -parameters, under which the system is queue-length stable. We also introduce the Max-Weight-log policy, which provides performance guarantees, without any knowledge of the arriving traffic. In the second part of the paper, we extend the results on Max-Weight and Max-Weight- α scheduling to a single-hop network, with arbitrary topology and scheduling constraints.

I. INTRODUCTION

The area of control of communication networks is one of the most active and fruitful fields of research in recent years, including several elaborate and interconnected forms of control, such as congestion control, routing, scheduling and power control. Much of the theoretical foundation regarding network stability was laid out in [13], where a communication network was abstracted by using a constrained queueing system. In that study, Tassiulas and Ephremides presented the Max-Weight scheduling and Back-Pressure routing policies, that are guaranteed to stabilize a queueing network, whenever possible. Subsequently, numerous studies have addressed variations or extensions of this policy in different settings.

In this paper we analyze the performance of various scheduling policies, in a single-server system serving two parallel queues, with a mix of heavy-tailed and light-tailed arrivals. We then extend our results, to single-hop networks, with arbitrary topology and scheduling constraints, and some heavy-tailed traffic.

Heavy-tailedness has been studied extensively in related fields, such as queueing theory [15] and task assignment in server farms [6]. In the context of scheduling in queueing networks, related work has been done by [1], [3], [4] and [11]. A not surprising, common conclusion of these studies is that scheduling policies that discriminate against the heavy-tailed arrivals tend to achieve better performance. The present study not only confirms this conclusion, but also determines the extent to which a scheduling policy needs to discriminate against the heavy-tailed arrivals, what performance guaran-

tees can be achieved by doing so, and, most importantly, what can happen otherwise.

The remainder of the paper is organized as follows: Section II contains the model that we use in the first part of this study, namely a single-server system with 2 parallel queues, with a mix of heavy-tailed and light-tailed traffic, and also a performance analysis for some popular scheduling policies in this setting. We begin by analyzing some queue-length unaware scheduling policies, such as round-robin, randomized, and priority, and characterize their performance. We present the Max-Weight scheduling policy, and show that the existence of a heavy-tailed arrival process causes the queue-length instability of both queues. Then, considering the Max-Weight- α scheduling policy, we associate the values of the α -parameters with the tails of the arrival processes, and provide sufficient conditions for the α -moments of the queue-lengths to be finite. Finally, we introduce and analyze the Max-Weight-log scheduling policy, which provides stability guarantees, even when the tails of the arrival processes are completely unknown. In Section III, we extend our analysis of the Max-Weight and Max-Weight- α policies to a single-hop network with arbitrary topology and scheduling constraints, a model general enough to capture a class of real-world communication networks. We conclude with a discussion of the contributions and limitations of this study, and possible future directions of research, in Section IV.

II. SCHEDULING IN PARALLEL QUEUES

A. Model and Definitions

We begin with the model of the queueing network that we consider in the first part of this study, a single-server system of 2 parallel queues. Time is slotted, and arrivals occur at the end of each time-slot. We call the two queues in the system H and L , and assume that they receive heavy-tailed and light-tailed traffic, respectively. We denote their queue-length stochastic processes as $\{Q_H(t); t \geq 0\}$ and $\{Q_L(t); t \geq 0\}$, and their arrival processes as $\{A_H(t); t \geq 0\}$ and $\{A_L(t); t \geq 0\}$, respectively, where $A_H(t)$ is the number of packets that queue H receives at time-slot t , and similarly for queue L . The arrival processes to each queue are assumed to be *independent and identically distributed* over time, and independent of each other. The size of the packets is fixed. The server is assumed to be constantly connected to the two queues, and has deterministic service rate of 1 packet per time-slot.

The scheduling decision that has to be made at each time-slot, is to determine which one of the two queues will be

This work was supported by the NSF Grants CNS-0915988, and CCF-0728554, and by the ARO MURI Grant W911NF-08-1-0238.

M. Markakis, E. Modiano, and J. Tsitsiklis are with the Massachusetts Institute of Technology, Cambridge, MA 02139 (e-mail: mihalis@mit.edu; modiano@mit.edu; jnt@mit.edu).

served. The scheduler has knowledge only of the current length of the queues and of the time-slot index; hence we restrict our analysis to Markovian scheduling policies. We consider a number of members of this class, ranging from queue-length unaware to queue-length aware, and from deterministic to randomized policies. However, all the policies analyzed here are *work-conserving*, in the sense that they allow the server to idle only when the system is empty.

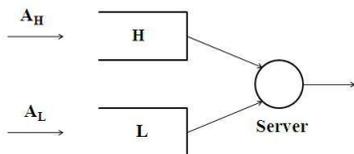


Fig. 1. A system of 2 parallel queues with a mix of heavy and light tailed traffic.

We will focus on the impact of scheduling in queueing networks, in the presence of heavy-tailed traffic. Before we proceed though, we should make this more specific.

Definition 1: The *Tail Coefficient* C of a nonnegative random variable X is a positive constant, defined as

$$C = \inf\{c \in R^+ \mid E[X^c] = \infty\}.$$

The Tail Coefficient is a generalization of the Tail Index [10], and provides a rough measure of ‘how heavy-tailed’ the distribution of a random variable is. As Definition 1 indicates, the Tail Coefficient is determined only through the moments of a random variable, and therefore, can be defined for any probability distribution. In contrast, the Tail Index is defined only for distributions whose tail decays polynomially.

Definition 2: An IID arrival process $\{A(t)\}$ is called *heavy-tailed* if $A(t)$ has Tail Coefficient less than or equal to 2. Otherwise, the arrival process is called *light-tailed*.

Note that an IID arrival process is heavy-tailed if the variance of its underlying distribution is infinite. We assume that the arrival process $\{A_H(t)\}$ is *heavy-tailed*, whereas $\{A_L(t)\}$ is *light-tailed*. Let us now turn to the issues of stability of a queueing network and admissibility of arriving traffic.

Definition 3: A discrete-time queueing system is *stable*, if it is modeled by a Markov Chain which is positive/Harris recurrent.

In our setting, where arrivals and departures of packets are integer quantities, stability is defined as positive recurrence. The stability region of the system we study is determined by the first moments of its arrival processes. So, define $\lambda_H = E[A_H(1)]$ and $\lambda_L = E[A_L(1)]$, and assume, for the problem to be meaningful, that both arrival rates are positive. We call the pair of arrival processes to the two queues *admissible* if: $\lambda_H + \lambda_L < 1$. Finally, we define throughput optimality as follows: a scheduling policy is called *throughput optimal*, if it stabilizes the queueing system for every admissible set of arrival processes.

Clearly, for the system of parallel queues described above, every work-conserving scheduling policy is throughput optimal. Since work-conserving scheduling policies are our focus in this paper, all the queueing systems to be analyzed are positive recurrent and converge to a steady-state distribution. So, we turn our attention to secondary measures of performance, such as the expected queue-length in steady-state. Our basic criterion is whether this expected queue-length, or moments of possibly different order, are finite or not.

Our analysis begins with a negative result: if a queue has heavy-tailed arrivals, then its stationary expected length is infinite, no matter what scheduling policy we apply.

Proposition 1: Consider the system of parallel queues described above, with any admissible, IID traffic. The expected length of queue H in steady-state is infinite, under any work-conserving, scheduling policy.

Proof: Consider a fictitious queue \tilde{H} , which has exactly the same arrivals as queue H , but is served at every time-slot with rate 1 packet per time-slot. We assume that both systems have reached steady-state, and we observe them at a typical time-slot τ . Being in steady-state, implies that there was a time-slot before τ , when both systems were empty. An easy, inductive argument can show that no matter which scheduling policy we apply to the system of parallel queues, the stationary queue-length of H dominates the queue-length of \tilde{H} at time-slot τ , for every sample path. Moreover, \tilde{H} is a discrete-time, stable $Geo^{[B]}/D/1$ queue, a special case of the $Geo^{[B]}/G/1$ queue, discussed in Appendix A, where the service time of each packet is deterministic and equal to 1. It is known that the expected queueing delay of a typical packet arriving to this queue in steady-state, is proportional to the second moment of its arrivals, which in the heavy-tailed case is infinite. Little’s Law (see [14] for a precise statement of the result invoked here), and the deterministic domination between the queue-lengths imply that the expected queue-length of H in steady-state is infinite. ■

Note that the statement of Proposition 1 is true, not only for the class of policies analyzed in this study, but for possibly non-Markovian and/or non-work-conserving scheduling policies, as well as much more general queueing systems.

B. Round-Robin Scheduling

Let us begin by considering scheduling policies that do not take into account the lengths of the queues in their decision, i.e., ‘queue-length unaware’ policies. We analyze the performance of three well-known policies: round-robin, randomized, and priority.

Consider the following version of the Round-Robin scheduling policy: service alternates every time-slot between the two queues, as long as packets are available for service in both of them. Assume, without loss of generality, that the scheduler attempts to serve queue L at even time-slots and queue H at odd time-slots. If a queue is chosen for scheduling and is empty, it loses its turn and the other queue is served instead. We call this scheduling policy ‘work-conserving Round-Robin’; Proposition 2 summarizes its performance.

Proposition 2: Under the ‘work-conserving Round-Robin’ scheduling policy, the expected length of queue L in steady-state is

- 1) finite, if $\lambda_L < \frac{1}{2}$;
- 2) infinite, if $\lambda_L > \frac{1}{2}$.

Proof: see Appendix B. ■

C. Randomized Scheduling

We continue with the analysis of a ‘work-conserving Randomized’ scheduling policy: at every time-slot a biased coin is tossed and if heads come up, an event of probability q , queue L is served; otherwise queue H is served. Coin tosses are IID Bernoulli trials, independent of any other random variable in our probabilistic model. If a queue is chosen for service but is empty, it loses its turn and the other queue is served.

Proposition 3: Under the ‘work-conserving Randomized’ scheduling policy, the expected length of queue L in steady-state is

- 1) finite, if $\lambda_L < q$;
- 2) infinite, if $\lambda_L > q$.

Proof: The proof is very similar to the proof of Proposition 2, and is omitted for brevity. ■

D. Priority Scheduling

The policies presented above, achieve finite expected length of queue L , only for some admissible pairs of arrival processes. Consider now the priority scheduling policy: serve queue L whenever it is nonempty.

Proposition 4: Under the Priority scheduling policy, the expected length of queue L in steady-state is finite.

Proof: Under the Priority scheduling policy, queue L is a discrete-time, stable $Geo^{[B]}/D/1$ queue, a special case of the $Geo^{[B]}/G/1$ queue, where the size of each burst of packets has finite first and second moment, and the service time of each packet is deterministic and equal to 1. The expected queueing delay of a typical packet arriving to queue L in steady-state is given by the formula presented in Appendix A, and is finite. Thus, the expected queue-length of L is finite as well. ■

E. Max-Weight Scheduling

So far, we have shown that we can guarantee finite expected queue-length for L , by giving priority to it. Why would we want to consider other scheduling policies? One reason is that we are ultimately interested in more complex models, involving time-varying connectivity (e.g., ON/OFF channels), as well as queueing networks. In such models, priority policies may not be throughput optimal. A second reason are the fairness issues that arise when priority policies are adopted.

It is well-known that for several more complex models, ‘queue-length aware’ policies are throughput optimal. So, let us consider the Max-Weight policy, which in our setting is

equivalent to ‘serve the longest queue’. Ties are broken in favor of queue L . The following proposition presents a surprising negative result for this policy, which can be interpreted as ‘queue-length instability of Max-Weight scheduling, in the presence of heavy-tailed traffic’.

Proposition 5: Consider the system of parallel queues described above, with any admissible, IID traffic. Under the Max-Weight scheduling policy, the expected length of queue L in steady-state is infinite.

Proof: see Appendix C. ■

F. Max-Weight- α Scheduling

As Proposition 5 suggests, Max-Weight scheduling performs poorly in the presence of heavy-tailed traffic. Driven by the idea of discriminating against heavy queues, we turn to a slightly more general member of the same class, the Max-Weight- α scheduling policy. The analysis of Max-Weight- α scheduling, in the context of an input-queued switch, was done in [7]. The scheduling decision of Max-Weight- α is given by

$$\arg \max_{n \in \{H, L\}} \{Q_n^{\alpha n}(t)\},$$

with ties broken arbitrarily. In contrast to most of the studies that have analyzed this scheduling policy, we use different values of α for each queue, smaller for the heavy queue and larger for the light queue. An appropriate relation between the values of the α -parameters and the Tail Coefficients of the corresponding arrival processes, is given in Proposition 6.

Proposition 6: Consider the system of parallel queues described above, with any admissible, IID traffic. Let C_H and C_L be the Tail Coefficients of the arrival processes to queue H and L , respectively. Under the Max-Weight- α scheduling policy, if $\alpha_H + 1 < C_H$ and $\alpha_L + 1 < C_L$, then the system of parallel queues is stable, and for $n \in \{H, L\}$:

$$E[Q_n^{\alpha n}] < \infty,$$

where the random variables Q_H and Q_L are distributed according to the stationary distributions of the queue-length processes $\{Q_H(t)\}$ and $\{Q_L(t)\}$, respectively.

Proof: It can be shown, that both the arrival processes and the Max-Weight- α scheduling policy, satisfy the conditions of Theorem 1 of [4], and hence the result follows. An alternative, more direct proof, is presented in Appendix D. ■

It is evident from the proof of Proposition 6 that the closer each of the α -parameters is to the value $C-1$ (or equivalently, the stronger the performance guarantee we wish to provide), the looser the upper bound becomes.

Example 1: Let the heavy-tailed arrival process have Tail Coefficient $C_H = 2$, and the light-tailed process have Tail Coefficient $C_L = 12$. The arriving traffic is assumed to be admissible.

- 1) If Max-Weight- α scheduling is applied, with $\alpha_L = 10$ and $\alpha_H = 1 - \epsilon$, for some $\epsilon \in (0, 1]$, then Proposition

6 implies that:

$$E[Q_L^{10}] < \infty;$$

- 2) If Max-Weight scheduling is applied, then Proposition 5 implies that:

$$E[Q_L] = \infty.$$

In fact, regarding the Max-Weight scheduling policy, we can derive an even stronger result. Note that in the given setting, the Max-Weight policy is equivalent to Max-Weight- α scheduling, if we choose $\alpha_H = \alpha_L$. Combining the results of Propositions 5 and 6, we have that every moment of the stationary queue-length Q_L with order less than 1 is finite. In contrast, all the moments of order 1 or higher are infinite. Hence, for this particular example, under the Max-Weight scheduling policy, we have a complete characterization of all polynomial moments of the stationary length of queue L . This, in turn, allows us to characterize the decay rate of the tail of the corresponding distribution: as B becomes large, $P(Q_L > B)$ must decay roughly as $1/B$.

G. Max-Weight-log Scheduling

Even though Proposition 6 gives a complete description of what can be guaranteed, in terms of the moments of the queue-lengths, it has a possible drawback: the proper values for the α -parameters have to be used, and for this to happen, some knowledge of the Tail Coefficients of both arrival processes is required. What if this requirement cannot be met in some cases? Consider the scheduling policy, termed Max-Weight-log, based on:

$$\arg \max_{n \in \{H, L\}} \{\log(1 + Q_n(t))\}.$$

Proposition 7: Consider the system of parallel queues described above, with any admissible, IID traffic. Under the Max-Weight-log scheduling policy, the system is stable, and, in steady-state,

$$E[\log(1 + Q_n)] < \infty, \quad \forall n \in \{H, L\}.$$

Proof: Similar to Proposition 6, it can be shown that both the arrival processes and the Max-Weight-log scheduling policy, satisfy the conditions of Theorem 1 of [4], and hence the result follows. ■

Combinations of the two, into hybrid scheduling policies, are possible too: give polynomial weight to the queues whose Tail Coefficients of the arrivals are known — give logarithmic weight to the queues whose Tail Coefficients are unknown.

Example 2: Consider the system of parallel queues described above, where the stochastic process $\{A_L(t)\}$ is exponentially light, i.e., the tail of the underlying distribution decays exponentially. In contrast, the only fact that is known about the process $\{A_H(t)\}$ is that it is heavy-tailed. Consider a scheduling policy that relies on the maximization:

$$\max\{\log(1 + Q_H(t)), Q_L^{\alpha_L}(t)\}.$$

By choosing appropriately the parameter α_L , we can guarantee the finiteness of any polynomial moment of Q_L .

III. SCHEDULING IN SINGLE-HOP NETWORKS

A. Model and Definitions

Let us now extend our analysis of Max-Weight and Max-Weight- α policies to a single-hop queueing network, with arbitrary topology and scheduling constraints. Again we assume that time is slotted and that arrivals occur at the end of each slot. Our network is represented by a directed graph $G = (\mathcal{N}, \mathcal{L})$, where \mathcal{N} is the set of nodes and \mathcal{L} is the set of links of the network. Our model involves *single-hop traffic flows*, where packets arrive at the source node of a link, for transmission to the node at the other end of the link.

In general, not all links can be activated at every time-slot. A set of links that can be activated concurrently is called a *feasible schedule*. We call \mathcal{S} the set of all feasible schedules, which is an arbitrary subset of the powerset of \mathcal{L} . We assume that all attempted transmissions of packets are successful, and that the transmission/service rates for all queues, and for all feasible schedules, are deterministic, bounded above by a finite constant S_{\max} , and bounded below by a positive constant S_{\min} . A *scheduler* is to determine, at each time-slot, which feasible schedule to activate; similar to the first part of the paper, we assume that the scheduler has knowledge of all the queue-lengths but only at the current time-slot.

Call F the set of all flows supported by the network. Note that F can be identified with \mathcal{L} , since the network has only single-hop traffic. Consider the following quantities:

$Q_f(t)$ is the number of packets in queue f , buffering the traffic of flow f , at time-slot t ;

$A_f(t)$ is the number of arrivals to queue f , at time-slot t ;

$S_f(t)$ is the number of packets that are scheduled for transmission from queue f , at time-slot t (not necessarily the number of packets that are transmitted because the queue may be empty).

We assume that the stochastic processes $\{A_f(t)\}$, $f \in F$ are IID over time and independent of each other, with rates λ_f , $f \in F$. The *Single-Hop Queue Dynamics* for each flow f are given by the expression:

$$Q_f(t+1) = Q_f(t) + A_f(t) - \min\{Q_f(t), S_f(t)\}.$$

The time-varying scheduling vector $S(t) = [S_f(t)] \in \mathcal{S}$ is determined by the scheduling policy that we apply. Let us now state for completeness some standard definitions, that will be useful to our analysis.

Definition 4: The *Stability Region* of the single-hop queueing network described above, is defined as the closure of the set of all arrival rate vectors $(E[A_f(1)]; f \in F)$, for which there exists a scheduling policy that stabilizes the network.

Definition 5: Consider the single-hop queueing network described above, and a set of IID arrival processes $\{A_f(t)\}$, $f \in F$. This set is called *admissible* if

- 1) the Tail Coefficient of every arrival process is strictly greater than 1;
- 2) the vector $(E[A_f(1)]; f \in F)$ lies in the interior of the network's Stability Region.

B. Max-Weight Scheduling

Under the Max-Weight scheduling policy, the scheduling vector $S(t)$ is a solution to the following optimization problem:

$$S(t) \in \arg \max_{[s_f] \in S} \left\{ \sum_{f \in F} Q_f(t) \cdot s_f \right\}.$$

Proposition 8: Consider the single-hop queueing network described above, with any admissible, IID traffic, under the Max-Weight scheduling policy. Call \mathcal{Q} the set of queues of the network, and consider any partition of \mathcal{Q} into two nonempty sets A and A^c . If

- 1) A contains at least one queue with infinite expected length in steady-state;
- 2) no link that carries traffic of A , can be activated concurrently with a link that carries traffic of A^c ,

then A^c contains at least one queue whose stationary expected length is infinite.

Proof: The proof is similar to the proof of Proposition 5, and is omitted for brevity. ■

Notice that at each time-slot, only one of the two sets can be served, so, in a way, the two sets are ‘parallel to each other’. A subtle but important point, is that now we refer to queues with infinite expected length, not necessarily queues with heavy-tailed arrivals. Proposition 8 suggests that there is more than one way for a queue to become queue-length unstable: either it has heavy-tailed arrivals, or it competes with a queue that has heavy-tailed arrivals, or it competes with a queue that has grown unbounded by competing with a queue that has heavy-tailed arrivals, etc. Therefore, in single-hop networks, we can have a very interesting phenomenon: the *propagation of queue-length instability*. Obviously, the extent of this propagation depends on the scheduling constraints of the network. The following example and corollaries illustrate the importance of Proposition 8, with respect to this phenomenon.

Example 3: Consider the wireless ring network of Fig. 2, under the two-hop interference model. The network consists of 6 nodes, each of which transmits packets to the node next to it, in the clockwise direction. The arriving traffic is admissible and IID, and Max-Weight scheduling is applied. The feasible schedules in this network are $\{1, 4\}$, $\{2, 5\}$, and $\{3, 6\}$.

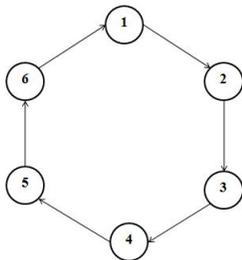


Fig. 2. A wireless ring network, with heavy-tailed traffic.

Let the queue of node 1 have heavy-tailed arrivals, and the rest of the queues have light-tailed arrivals. Consider the partition of the set of queues $\{\{1, 2, 4, 5\}, \{3, 6\}\}$. This partition satisfies the conditions of Proposition 8, so at least one queue in the set $\{3, 6\}$ blows up, in expectation. Repeating this argument for the partition $\{\{1, 3, 4, 6\}, \{2, 5\}\}$, we conclude that at least one queue in the set $\{2, 5\}$ blows up, in expectation. Summarizing, even though only one queue of the network has heavy-tailed arrivals, because of the propagation of queue-length instability, at least three queues have infinite expected length in steady-state.

Corollary 1: Consider a single-server system of N parallel queues, with any admissible, IID traffic, under Max-Weight scheduling. If there exists a queue with heavy-tailed arrivals, then all N queues have infinite expected length in steady-state.

Corollary 2: Consider a $N \times N$ input-queued switch, with any admissible, IID traffic, under Max-Weight scheduling. If there exists a queue with heavy-tailed arrivals, then at least N , out of the total N^2 virtual queues, have infinite expected length in steady-state.

C. Max-Weight- α Scheduling

Under the Max-Weight- α scheduling policy, the scheduling vector $S(t)$ is a solution to the following optimization problem:

$$S(t) \in \arg \max_{[s_f] \in S} \left\{ \sum_{f \in F} Q_f^{\alpha_f}(t) \cdot s_f \right\}.$$

Similar to the case of parallel queues, the values of the nonnegative parameters α_f , $f \in F$, are crucial. Proposition 9 describes the performance that this scheduling policy achieves, when the values of the α -parameters are properly selected.

Proposition 9: Consider the single-hop queueing network described above, with any admissible, IID traffic. Let C_f be the Tail Coefficient of the arrival process $\{A_f(t)\}$. Under the Max-Weight- α scheduling policy, if $\alpha_f + 1 < C_f$ for all $f \in F$, then the queueing network is stable, and

$$E[Q_f^{\alpha_f}] < \infty, \quad f \in F.$$

Proof: The proof is similar to the proof of the Max-Weight- α result for parallel queues (Proposition 6), and is omitted for brevity. ■

IV. DISCUSSION

The vast majority of past studies dealing with scheduling in a queueing network have assumed that all arrival processes are light-tailed, i.e., $E[A^2] < \infty$. In contrast, we dealt with heavy-tailed traffic explicitly. In order to motivate the subsequent work, we analyzed the queue-length stability of some queue-length unaware scheduling policies, such as round-robin, randomized, and preemptive priority, in a simple system of parallel queues, under a mix of heavy-tailed and light-tailed traffic. We then broadened our scope, by considering a

single-hop network, with arbitrary topology and scheduling constraints, a model general enough to capture important, real-world communication networks. We analyzed the queue-length stability of the popular Max-Weight scheduling policy in such a setting, and showed that the existence of a queue with heavy-tailed arrivals can have a significant impact on the performance of the whole network. Finally, we considered the Max-Weight- α scheduling policy, and proved that if the α -parameters are chosen appropriately, then the network is stable, in the positive recurrence sense, and the α -moments of the queue-lengths are finite.

It is straightforward to further extend the results of Proposition 9 to multi-hop networks, by associating the α -parameters of a *Max-Weight- α scheduling and Back-Pressure- α routing policies* with the Tail Coefficients of the arrival flows.

Clearly, the most restrictive assumption of the present study is the IID nature of the arriving traffic, when evidence suggests that strong correlations can appear [8]. In case these correlations are bounded in time, then generalizing our results is possible, by considering a multi-step Lyapunov drift argument. In contrast, if the correlations in time extend to infinity, and particularly if the arriving traffic is long-range dependent, new paths must be investigated. This is an interesting and important direction of future research.

V. APPENDIX A - THE $Geo^{[B]}/G/1$ QUEUE

The $Geo^{[B]}/G/1$ queue is a discrete-time queue, where jobs (packets in our setting) arrive in batches (traffic bursts). The number of jobs in the various batches are independent and identically distributed random variables, and the arrival of batches forms a Bernoulli process (IID, geometric inter-arrival times). The service time of each job is an independent and identically distributed copy of some integer random variable, following a general distribution. We assume that the service discipline is ‘First Come, First Served’ and that arrivals occur at the end of each time-slot. The expected queueing delay of a typical job arriving to a discrete-time, stable $Geo^{[B]}/G/1$ queue in steady-state, is given by:

$$E[D] = \frac{\lambda^2 b_2 - \lambda \rho + \lambda_2 b}{2\lambda(1 - \rho)},$$

where λ and λ_2 are the first and second moments of the number of jobs in each batch, b and b_2 are the first and second moments of the service time of each job, and $\rho = \lambda \cdot b$ is the traffic intensity, which has to be strictly less than 1 for the queue to be stable. More details about the $Geo^{[B]}/G/1$ queue can be found in [12]. What is important for our analysis, is the fact that the expected delay of a $Geo^{[B]}/G/1$ queue scales with the second moment of the batch size, which in the case of heavy-tailed arrivals is infinite.

VI. APPENDIX B - PROOF OF PROPOSITION 2

Suppose that $\lambda_L < \frac{1}{2}$. Consider a fictitious queue F that has the same arrivals as queue L , but is scheduled for service

only at even time-slots. Therefore, it does not get the extra service slots that L does whenever queue H is empty. We assume that both queues are in steady-state, and we observe them at a typical time-slot τ . Arguing similarly to Proposition 1, we can show that the length of queue F dominates the length of L at a typical time-slot τ , for every sample path. Since queue L , and consequently queue F , has light-tailed arrivals, the first and second moment of the size of the bursts of packets it receives are finite. Moreover, queue F is a discrete-time, stable $Geo^{[B]}/D/1$ queue. The expected queueing delay of a typical packet arriving to it in steady-state is given by the formula presented in Appendix A, and is finite. Thus, the expected queue-length of F is also finite. The fact that the expected length of queue L is finite results from the deterministic domination between the stationary lengths of the two queues.

Now suppose that $\lambda_L > \frac{1}{2}$. Consider a fictitious system of two parallel queues, very similar to the original one: queue \tilde{H} has the same arrivals as queue H , but instead of ‘work-conserving Round-Robin’, it receives a deterministic service of 1 packet/time-slot, at every time-slot. Queue \tilde{L} has the same arrivals as queue L , and is served with the same deterministic rate at i) even time-slots and ii) odd time-slots, if queue \tilde{H} is empty.

Suppose that both the actual and the fictitious system are in steady-state, and we observe them at a typical time-slot τ . First, we compute the expected queue-length of \tilde{L} , at time-slot τ . Consider the events $\Gamma(\tau) = \{\tilde{H} \text{ nonempty at time } \tau\}$ and its complement, $\Gamma^c(\tau)$. We can express the expected queue-length of \tilde{L} , viewed at a typical time-slot τ , as follows:

$$E[Q_{\tilde{L}}(\tau)] = E[Q_{\tilde{L}}(\tau) | \Gamma(\tau)] \cdot P(\Gamma(\tau)) + E[Q_{\tilde{L}}(\tau) | \Gamma^c(\tau)] \cdot P(\Gamma^c(\tau)).$$

The fact that \tilde{H} is a stable queue, is sufficient to guarantee that $P(\Gamma(\tau)) > 0$ and $P(\Gamma^c(\tau)) > 0$. Conditioning on the event $\Gamma(\tau)$, call $Z(\tau)$ the age of the busy period of \tilde{H} , at time-slot τ . Clearly, this random variable is bounded below by the age of the traffic burst in service, at time-slot τ (the number of packets from this burst that are already served). From renewal theory [5], we know that the expectation of the latter, is proportional to the second moment of the arrivals to queue \tilde{H} . Since this process is heavy-tailed, the expected age of the traffic burst in service, at time-slot τ , is infinite, which leads to:

$$E[Z(\tau) | \Gamma(\tau)] = \infty.$$

Now call $\tilde{\tau}$ the time-slot that initiates the busy period of \tilde{H} , and consider the stochastic processes $\{A_{\tilde{L}}(t)\}$ and $\{S_{\tilde{L}}(t)\}$, representing the arrivals and departures from queue \tilde{L} . One can see that

$$Q_{\tilde{L}}(\tau) \geq \sum_{k=\tilde{\tau}}^{\tau} A_{\tilde{L}}(k) - \sum_{k=\tilde{\tau}}^{\tau} S_{\tilde{L}}(k). \quad (1)$$

Notice that we have constructed the fictitious system so that conditioning on the event $\Gamma(\tau)$ does not change the

statistics of the arrivals of \tilde{L} , which are identical to the arrivals to queue L . Moreover, if we start from time-slot τ and go backwards in time, time-slot $\tilde{\tau}$ is a stopping time for the process $\{A_{\tilde{L}}(t)\}$; hence, we can invoke Wald's equality. Furthermore, conditioned on the event $\Gamma(\tau)$, queue \tilde{L} is served once every two time-slots. Considering the conditional expectation of Eq. (1), and taking into account these facts, we get:

$$E[Q_{\tilde{L}}(\tau) | \Gamma(\tau)] \geq E[Z(\tau) | \Gamma(\tau)] \cdot (\lambda_L - \frac{1}{2}) - \frac{1}{2},$$

which implies that

$$E[Q_{\tilde{L}}(\tau) | \Gamma(\tau)] = \infty,$$

and thus

$$E[Q_{\tilde{L}}(\tau)] = \infty.$$

Now we come to the second part of the proof, namely showing that if the expected queue-length of \tilde{L} is infinite, then the expected length of L has to be infinite too. The fact that the two systems are in steady-state, guarantees that there was a time-slot before τ , when both systems were empty. For convenience, call this time-slot 0. A simple, inductive argument is sufficient to prove that $Q_H(t) \geq Q_{\tilde{H}}(t)$, for all $t \geq 0$. We now focus on the evolution of queues L and \tilde{L} , whose dynamics are described by the set of equations:

$$\begin{aligned} Q_L(t+1) &= Q_L(t) - S_L(t) \cdot \mathbf{1}_{\{Q_L(t) > 0\}} + A_L(t); \\ Q_{\tilde{L}}(t+1) &= Q_{\tilde{L}}(t) - S_{\tilde{L}}(t) \cdot \mathbf{1}_{\{Q_{\tilde{L}}(t) > 0\}} + A_L(t); \end{aligned}$$

and $Q_L(0) = Q_{\tilde{L}}(0) = 0$. Both queues are served on two occasions i) at even time-slots (common for L and \tilde{L}) and ii) at odd timeslots, whenever queue H/\tilde{H} is empty. Since $Q_H(t) \geq Q_{\tilde{H}}(t)$, for all $t \geq 0$, the set of time-slots when queue H is empty is a subset of the set of time-slots when queue \tilde{H} is empty. So, the service opportunities for the two queues satisfy:

$$S_{\tilde{L}}(t) \geq S_L(t), \quad \forall t \geq 0.$$

An inductive argument can show:

$$Q_{\tilde{L}}(t) \leq Q_L(t), \quad \forall t \geq 0,$$

which implies that

$$E[Q_L(\tau)] \geq E[Q_{\tilde{L}}(\tau)] = \infty.$$

Finally, if Q_L is a random variable, distributed according to the stationary distribution of the queue-length process $\{Q_L(t)\}$, the fact that we observe the system at a typical time in steady-state, implies that

$$E[Q_L] = E[Q_L(\tau)] = \infty.$$

VII. APPENDIX C - PROOF OF PROPOSITION 5

Suppose that the system of parallel queues is in steady-state, and a typical burst of packets arrives to queue L , at time-slot τ . We distinguish between two cases:

Consider first the event $\Gamma(\tau)$, that $Q_L(\tau) < Q_H(\tau)$. Define the queueing delay $W_L(\tau)$ that this burst of packets experiences, as the number of time-slots between its arrival and the time it starts getting served. Consider the following, best-case scenario about $W_L(\tau)$: as soon as this typical burst arrives to queue L , queue H receives no more packets and is served at every time-slot, until its length becomes equal to the queue-length of L . At that point, the scheduling policy begins to serve queue L , until the first packet of the burst is finally served. Call T the number of time-slots until the service switches from queue H to queue L . Our best-case scenario gives a lower-bound on $W_L(\tau)$:

$$W_L(\tau) \geq Q_L(\tau) + T. \quad (2)$$

Under Max-Weight scheduling, for the service to switch, T must satisfy:

$$Q_L(\tau) + \sum_{k=0}^T A_L(\tau+k) \geq Q_H(\tau) - T,$$

which implies that

$$\sum_{k=0}^T [A_L(\tau+k) + 1] \geq Q_H(\tau) - Q_L(\tau).$$

The last inequality, combined with Wald's equality, gives

$$E[T | \Gamma(\tau)] \cdot (\lambda_L + 1) \geq E[Q_H(\tau) | \Gamma(\tau)] - E[Q_L(\tau) | \Gamma(\tau)]. \quad (3)$$

Taking the conditional expectation of both sides of Eq. (2) and combining it with Eq. (3), results in

$$\begin{aligned} E[W_L(\tau) | \Gamma(\tau)] &\geq E[Q_L(\tau) | \Gamma(\tau)] + E[T | \Gamma(\tau)] \\ &\geq \frac{1}{\lambda_L + 1} \cdot E[Q_H(\tau) | \Gamma(\tau)] \\ &\quad + E[Q_L(\tau) | \Gamma(\tau)] - \frac{1}{\lambda_L + 1} \cdot E[Q_L(\tau) | \Gamma(\tau)] \\ &\geq \frac{1}{\lambda_L + 1} \cdot E[Q_H(\tau) | \Gamma(\tau)]. \quad (4) \end{aligned}$$

Consider now the event $\Gamma^c(\tau)$, that $Q_L(\tau) \geq Q_H(\tau)$. Consider also the following, best-case scenario about the delay $W_L(\tau)$ that the typical burst of packets experiences in this case: as soon as the burst arrives, queue L starts getting served constantly, until the first packet of the burst is served. Our best-case scenario indicates a lower-bound on the delay:

$$W_L(\tau) \geq Q_L(\tau) \geq Q_H(\tau) \geq \frac{1}{\lambda_L + 1} \cdot Q_H(\tau),$$

which implies that

$$E[W_L(\tau) | \Gamma^c(\tau)] \geq \frac{1}{\lambda_L + 1} \cdot E[Q_H(\tau) | \Gamma^c(\tau)]. \quad (5)$$

Since the system is assumed to be in steady-state, both events $\Gamma(\tau)$ and $\Gamma^c(\tau)$ have positive probability, so the conditional expectations that we considered are well-defined. Proposition 1 and the BASTA property (see [9] for a precise

statement) imply that

$$E[Q_H(\tau)] = \infty. \quad (6)$$

Combining Eq. (4), (5) and (6), we have:

$$E[W_L(\tau)] \geq \frac{1}{\lambda_L + 1} \cdot E[Q_H(\tau)] = \infty. \quad (7)$$

Little's law and Eq. (7) imply that the expected number of traffic bursts, and hence the expected number of packets in queue L in steady-state, is infinite. So, if Q_L is some random variable, distributed according to the stationary distribution of the queue-length process $\{Q_L(t)\}$:

$$E[Q_L] = \infty.$$

VIII. APPENDIX D - PROOF OF PROPOSITION 6

Define the queue-length vector $Q = (Q_H, Q_L)$ as the state of the system, and consider the Lyapunov function:

$$V(Q) = \sum_{n \in \{H, L\}} \frac{1}{\alpha_n + 1} Q_n^{\alpha_n + 1}.$$

We have:

$$\begin{aligned} & E[V(Q(t+1)) | Q(t)] \\ &= \sum_{n \in \{H, L\}} E \left[\frac{1}{\alpha_n + 1} (Q_n(t) + \Delta_n(t))^{\alpha_n + 1} | Q(t) \right], \end{aligned}$$

where

$$\Delta_n(t) = A_n(t) - S_n(t) \cdot 1_{\{Q_n(t) > 0\}}.$$

Here, $A_n(t)$ and $S_n(t)$ represent the number of arriving and departing packets from queue n at time-slot t . Note that for the system of parallel queues described in Section II, $S_n(t)$ can only take the values 0 and 1. Finally, the indicator function $1_{\{Q_n(t) > 0\}}$ ensures that packets depart only from nonempty queues. Throughout the proof we use the shorthand notation:

$$V_n(Q_n(t)) = \frac{1}{\alpha_n + 1} Q_n^{\alpha_n + 1}(t).$$

We now consider the conditional expectation of the terms $V_n(Q_n(t+1))$, distinguishing between two cases:

Regarding the term(s) $V_n(Q_n(t))$ with $\alpha_n + 1 < 2$ (corresponding to queue H , and possibly queue L , depending on the value of α_L), consider the 0^{th} order Taylor expansion around $Q_n(t)$:

$$\begin{aligned} & \frac{1}{\alpha_n + 1} (Q_n(t) + \Delta_n(t))^{\alpha_n + 1} \\ &= \frac{1}{\alpha_n + 1} Q_n^{\alpha_n + 1}(t) + \Delta_n(t) \cdot \xi^{\alpha_n}, \end{aligned}$$

for some $\xi \in [Q_n(t) - 1_{\{Q_n(t) > 0\}}, Q_n(t) + A_n(t)]$. Thus,

$$V_n(Q_n(t+1)) = V_n(Q_n(t)) + \Delta_n(t) \cdot \xi^{\alpha_n},$$

and

$$\begin{aligned} & E[V_n(Q_n(t+1)) | Q(t)] \\ &= V_n(Q_n(t)) + E[\Delta_n(t) \cdot \xi^{\alpha_n} | Q(t)]. \end{aligned}$$

Consider the event $\Gamma_n(t) = \{\Delta_n(t) < 0\}$ and its complement. Then,

$$\begin{aligned} & E[V_n(Q_n(t+1)) | Q(t)] \leq V_n(Q_n(t)) \\ &+ E[\Delta_n(t) \cdot (Q_n(t) - 1_{\{Q_n(t) > 0\}})^{\alpha_n} | Q(t), \Gamma_n(t)] \\ &\quad \cdot P(\Gamma_n(t) | Q(t)) \\ &+ E[\Delta_n(t) \cdot (Q_n(t) + A_n(t))^{\alpha_n} | Q(t), \Gamma_n^c(t)] \\ &\quad \cdot P(\Gamma_n^c(t) | Q(t)). \end{aligned}$$

Since $Q_n(t)$ and $A_n(t)$ are nonnegative integers, and $0 \leq \alpha_n < 1$, it is not difficult to verify that:

$$\begin{aligned} & (Q_n(t) + A_n(t))^{\alpha_n} \leq Q_n^{\alpha_n}(t) + A_n^{\alpha_n}(t), \\ & (Q_n(t) - 1_{\{Q_n(t) > 0\}})^{\alpha_n} \geq Q_n^{\alpha_n}(t) - 1. \end{aligned}$$

Using these inequalities, we have:

$$\begin{aligned} & E[V_n(Q_n(t+1)) | Q(t)] \leq V_n(Q_n(t)) \\ &\quad + E[\Delta_n(t) | Q(t)] \cdot Q_n^{\alpha_n}(t) \\ &+ E[-\Delta_n(t) | Q(t), \Gamma_n(t)] \cdot P(\Gamma_n(t) | Q(t)) \\ &+ E[\Delta_n(t) \cdot A_n^{\alpha_n}(t) | Q(t), \Gamma_n^c(t)] \cdot P(\Gamma_n^c(t) | Q(t)), \end{aligned}$$

and

$$\begin{aligned} & E[V_n(Q_n(t+1)) | Q(t)] \leq V_n(Q_n(t)) \\ &\quad + E[\Delta_n(t) | Q(t)] \cdot Q_n^{\alpha_n}(t) \\ &+ 1 + E[A_n^{\alpha_n + 1}(t) | Q(t), \Gamma_n^c(t)] \cdot P(\Gamma_n^c(t) | Q(t)). \end{aligned}$$

In the last equation we used the fact that when $\Delta_n(t) < 0$ (event $\Gamma_n(t)$), then $\Delta_n(t) = -1$. Also, when $\Delta_n(t) \geq 0$ (event $\Gamma_n^c(t)$), then $\Delta_n(t) \leq A_n(t)$, so that $\Delta_n(t) \cdot A_n^{\alpha_n}(t) \leq A_n^{\alpha_n + 1}(t)$. Since $\alpha_n + 1 < C_n$, we know that $E[A_n^{\alpha_n + 1}(t)] < \infty$, which implies that $E[A_n^{\alpha_n + 1}(t) | Q(t), \Gamma_n^c(t)] < \infty$. So, there exists a finite constant b_n , such that:

$$\begin{aligned} & E[V_n(Q_n(t+1)) | Q(t)] \leq V_n(Q_n(t)) \\ &\quad + (\lambda_n - S_n(t)) \cdot Q_n^{\alpha_n}(t) + b_n. \end{aligned}$$

This, in turn, implies (trivially) that for every $y_n > 0$, there exists some finite constant $B_n(y_n)$ such that:

$$\begin{aligned} & E[V_n(Q_n(t+1)) | Q(t)] \leq V_n(Q_n(t)) \\ &\quad + (\lambda_n - S_n(t)) \cdot Q_n^{\alpha_n}(t) \\ &\quad + \frac{y_n}{4} \cdot Q_n^{\alpha_n}(t) + B_n(y_n). \end{aligned}$$

Regarding a possible term $V_n(Q_n(t))$ with $\alpha_n + 1 \geq 2$ (corresponding to queue L , if α_L is chosen to be greater than or equal to 1), consider the 1^{st} order Taylor expansion around $Q_n(t)$:

$$\begin{aligned} & \frac{1}{\alpha_n + 1} (Q_n(t) + \Delta_n(t))^{\alpha_n + 1} \\ &= \frac{1}{\alpha_n + 1} Q_n^{\alpha_n + 1}(t) + \Delta_n(t) \cdot Q_n^{\alpha_n}(t) \\ &\quad + \frac{\Delta_n^2(t)}{2} \cdot \alpha_n \cdot \xi^{\alpha_n - 1}, \end{aligned}$$

for some $\xi \in [Q_n(t) - 1_{\{Q_n(t) > 0\}}, Q_n(t) + A_n(t)]$. Thus,

$$\begin{aligned} E[V_n(Q_n(t+1)) | Q(t)] &= V_n(Q_n(t)) \\ &\quad + E[\Delta_n(t) | Q(t)] \cdot Q_n^{\alpha_n}(t) \\ &\quad + E\left[\frac{\Delta_n^2(t)}{2} \cdot \alpha_n \cdot \xi^{\alpha_n-1} | Q(t)\right]. \end{aligned}$$

Since $\alpha_n - 1 \geq 0$, the last term can be bounded above by:

$$\begin{aligned} &E\left[\frac{\Delta_n^2(t)}{2} \cdot \alpha_n \cdot \xi^{\alpha_n-1} | Q(t)\right] \\ &\leq E\left[\frac{\Delta_n^2(t)}{2} \cdot \alpha_n \cdot (Q_n(t) + A_n(t))^{\alpha_n-1} | Q(t)\right]. \end{aligned}$$

It is easy to verify that

$$(Q_n(t) + A_n(t))^{\alpha_n-1} \leq 2^{\alpha_n-1} \cdot (Q_n^{\alpha_n-1}(t) + A_n^{\alpha_n-1}(t)),$$

and also

$$\Delta_n^2(t) \leq A_n^2(t) + 1.$$

Using these inequalities we have:

$$\begin{aligned} &E\left[\frac{\Delta_n^2(t)}{2} \cdot \alpha_n \cdot \xi^{\alpha_n-1} | Q(t)\right] \\ &\leq 2^{\alpha_n-2} \cdot \alpha_n \cdot (E[A_n^2(t)] + 1) \cdot Q_n^{\alpha_n-1}(t) \\ &\quad + 2^{\alpha_n-2} \cdot \alpha_n \cdot (E[A_n^{\alpha_n+1}(t)] + E[A_n^{\alpha_n-1}(t)]). \end{aligned}$$

Since $2 \leq \alpha_n + 1 < C_n$, we know that the terms $E[A_n^2(t)]$, $E[A_n^{\alpha_n+1}(t)]$, and $E[A_n^{\alpha_n-1}(t)]$ are finite. So, for every $y_n > 0$, there exists some finite constant $B_n(y_n)$ such that:

$$\begin{aligned} &E\left[\frac{\Delta_n^2(t)}{2} \cdot \alpha_n \cdot \xi^{\alpha_n-1} | Q(t)\right] \\ &\leq \frac{y_n}{4} \cdot Q_n^{\alpha_n}(t) + B_n(y_n). \end{aligned}$$

Hence,

$$\begin{aligned} E[V_n(Q_n(t+1)) | Q(t)] &\leq V_n(Q_n(t)) \\ &\quad + (\lambda_n - S_n(t)) \cdot Q_n^{\alpha_n}(t) \\ &\quad + \frac{y_n}{4} \cdot Q_n^{\alpha_n}(t) + B_n(y_n). \end{aligned}$$

Summing over $n \in \{H, L\}$, we have an upper bound for the Lyapunov drift:

$$\begin{aligned} E[V(Q(t+1)) | Q(t)] &\leq V(Q(t)) \\ &\quad + \sum_{n \in \{H, L\}} (\lambda_n - S_n(t)) \cdot Q_n^{\alpha_n}(t) \\ &\quad + \sum_{n \in \{H, L\}} \frac{y_n}{4} \cdot Q_n^{\alpha_n}(t) + \sum_{n \in \{H, L\}} B_n(y_n). \end{aligned}$$

The admissibility of the arriving traffic implies the existence of a positive constant γ , such that:

$$\sum_{n \in \{H, L\}} \lambda_n = 1 - \gamma.$$

Moreover, by the definition of the Max-Weight- α scheduling policy, we know that

$$\sum_{n \in \{H, L\}} S_n(t) \cdot Q_n^{\alpha_n}(t) = \max_{n \in \{H, L\}} \{Q_n^{\alpha_n}(t)\}.$$

Recall that y_n can be chosen arbitrarily small. By suitably choosing $y_n = \gamma$ for all n , we have:

$$\sum_{n \in \{H, L\}} (\lambda_n + y_n) \leq 1 - \gamma/2.$$

So, the Lyapunov drift term can be bounded above by:

$$\begin{aligned} E[V(Q(t+1)) | Q(t)] &\leq V(Q(t)) \\ &\quad - \frac{\gamma}{2} \cdot \max_{n \in \{H, L\}} \{Q_n^{\alpha_n}(t)\} + \sum_{n \in \{H, L\}} B_n(\gamma), \end{aligned}$$

which implies that for every $n \in \{H, L\}$:

$$\begin{aligned} E[V(Q(t+1)) | Q(t)] &\leq V(Q(t)) \\ &\quad - \frac{\gamma}{2} \cdot Q_n^{\alpha_n}(t) + \sum_{n \in \{H, L\}} B_n(\gamma). \end{aligned}$$

By Foster's Criterion for irreducible Markov Chains [2], the queue-length processes $\{Q_H(t)\}$ and $\{Q_L(t)\}$ converge in distribution to some random variables Q_H and Q_L respectively. Additionally, Fatou's Lemma, the Law of Iterated Expectations and a telescopic summation of terms, are sufficient to show that

$$E[Q_n^{\alpha_n}] \leq \frac{2}{\gamma} \cdot \sum_{n \in \{H, L\}} B_n(\gamma) < \infty, \quad n \in \{H, L\}.$$

REFERENCES

- [1] V. Anantharam (1999). Scheduling strategies and long-range dependence. *Queueing Systems*, 33, 73-89.
- [2] S. Asmussen (1987). *Applied probability and queues*. Wiley Inc.
- [3] S. Borst, M. Mandjes, M. van Uitert (2003). Generalized processor sharing with light-tailed and heavy-tailed input. *IEEE/ACM Transactions on Networking*, 11, 821-834.
- [4] A. Eryilmaz, R. Srikant, J. Perkins (2005). Stable scheduling policies for fading wireless channels. *IEEE/ACM Transactions on Networking*, 13, 411-424.
- [5] R. Gallager (1996). *Discrete stochastic processes*. Kluwer Academic.
- [6] M. Harchol-Balter, A. Scheller-Wolf, A. Young (2009). Surprising results on task assignment in server farms with high-variability workloads. In: *Proc. Sigmetrics 2009*.
- [7] I. Keslassy, N. McKeown (2001). Analysis of scheduling algorithms that provide 100% throughput in input-queued switches. In: *Proc. Allerton 2001*.
- [8] W. Leland, M. Taqqu, W. Willinger, D. Wilson (1994). On the self-similar nature of ethernet traffic. *IEEE/ACM Transactions on Networking*, 2, 1-15.
- [9] A. Makowski, B. Melamed, W. Whitt (1989). On averages seen by arrivals in discrete time. In: *Proc. CDC 1989*.
- [10] K. Park, W. Willinger (2000). Self-similar network traffic: an overview. In: *Self-similar network traffic and performance evaluation*, K. Park and W. Willinger, editors, Wiley Inc.
- [11] Z. Shao, U. Madhow (2002). Scheduling heavy-tailed data traffic over the wireless internet. In: *Proc. VTC 2002*.
- [12] H. Takagi (1993). *Queueing analysis*, vol. 3: discrete-time systems. North-Holland.
- [13] L. Tassioulas, A. Ephremides (1992). Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 37, 1936-1948.
- [14] J. Walrand (1988). *An introduction to queueing networks*. Prentice Hall.
- [15] W. Whitt (2000). The impact of heavy-tailed service-time distribution upon the M/GI/s waiting time distribution. *Queueing Systems*, 36, 71-87.