MIT Open Access Articles

Max-weight scheduling in networks with heavy-tailed traffic

Citation: Markakis, Mihalis G., Eytan H. Modiano, and John N. Tsitsiklis. "Max-Weight Scheduling in Networks with Heavy-Tailed Traffic." INFOCOM, 2012 Proceedings IEEE, 25-30 March, 2012, Orlando, FL, IEEE, 2012, pp. 2318–26.

As Published: http://dx.doi.org/10.1109/INFCOM.2012.6195619

Publisher: Institute of Electrical and Electronics Engineers (IEEE)

Persistent URL: http://hdl.handle.net/1721.1/112965

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Massachusetts Institute of Technology

Max-Weight Scheduling in Networks with Heavy-Tailed Traffic

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

Abstract—**We consider the problem of packet scheduling in a single-hop network with a mix of heavy-tailed and light-tailed traffic, and analyze the impact of heavy-tailed traffic on the performance of Max-Weight scheduling. As a performance metric we use the delay stability of traffic flows: a traffic flow is delay stable if its expected steady-state delay is finite, and delay unstable otherwise. First, we show that a heavy-tailed traffic flow is delay unstable under any scheduling policy. Then, we focus on the celebrated Max-Weight scheduling policy, and show that a light-tailed flow that conflicts with a heavy-tailed flow is also delay unstable. This is true irrespective of the rate or the tail distribution of the light-tailed flow, or other scheduling constraints in the network. Surprisingly, we show that a light-tailed flow can be delay unstable, even when it does not conflict with heavy-tailed traffic. Furthermore, delay stability in this case may depend on the rate of the light-tailed flow. Finally, we turn our attention to the class of Max-Weight-α scheduling policies; we show that if the α-parameters are chosen suitably, then the sum of the α-moments of the steady-state queue lengths is finite. We provide an explicit upper bound for the latter quantity, from which we derive results related to the delay stability of traffic flows, and the scaling of moments of steady-state queue lengths with traffic intensity.**

I. INTRODUCTION

We consider a single-hop queueing network with a mix of heavy-tailed and light-tailed traffic, and study the impact of heavy-tailed traffic on the performance of Max-Weight scheduling. Single-hop network models have been used extensively to capture the dynamics and scheduling decisions in real-world communication networks, such as wireless uplinks and downlinks, switches, wireless ad hoc networks, sensor networks, and call centers. In all these systems, one cannot serve all queues simultaneously, e.g., due to wireless interference constraints, giving rise to a scheduling problem. Clearly, the overall performance of the network depends critically on the scheduling policy applied.

The focus of this paper is on a well-studied class of scheduling policies, commonly refered to as Max-Weight policies. This class of policies was introduced in the seminal work of Tassiulas and Ephremides [20], and since then numerous studies have analyzed the performance of such policies in different settings, e.g., see [8], and the references therein. A remarkable property of Max-Weight policies is their **throughput optimality**, i.e., their ability to stabilize a queueing network whenever this is possible, without any information on the arriving traffic. Moreover, it has been shown that policies from this class achieve low, or even optimal, average delay for specific network

topologies, when the arriving traffic is light-tailed [7], [13], [19].[1] However, the performance of Max-Weight scheduling in the presence of heavy-tailed traffic is not well understood.

We are motivated to study networks with heavy-tailed traffic by significant evidence that traffic in real-world communication networks exhibits strong correlations and statistical similarity over different time scales. This observation was first made by Leland *et al.* [10] through analysis of Ethernet traffic traces. Subsequent empirical studies have documented this phenomenon in other networks, while accompanying theoretical studies have associated it with arrival processes that have heavy tails; see [14] for an overview. The impact of heavy tails has been analyzed extensively in the context of single or multi-server queues; see the survey papers [1], [3], and the references therein. However, the related work is rather limited in the context of queueing networks, e.g., see the paper by Borst *et al.* [2], which studies the "Generalized Processor Sharing" policy.

This paper aims to fill a gap in the literature, by analyzing the performance of Max-Weight scheduling in a single-hop queueing network with a mix of heavy-tailed and light-tailed traffic. In particular, we study the delay stability of traffic flows: a traffic flow is delay stable if its expected steady-state delay is finite, and delay unstable otherwise. Our previous work [11] gives some preliminary results in this direction, in a system with two parallel queues and a single server. Here we elaborate on the intricate delay stability phenomena that arise in more complex networks. The **main contributions** of this paper include: i) in a single-hop queueing network under the Max-Weight scheduling policy, we show that any light-tailed flow that conflicts with a heavy-tailed flow is delay unstable; ii) surprisingly, we also show that for certain admissible arrival rates, a light-tailed flow can be delay unstable even if it does not conflict with heavy-tailed traffic; iii) we analyze the Max-Weight-α scheduling policy, and show that if the α-parameters are chosen suitably, then the sum of the α-moments of the steady-state queue lengths is finite. We use this result to prove that by proper choice of the α-parameters, all light-tailed flows are delay stable. Moreover, we show that Max-Weight-α achieves the optimal scaling of higher moments of steady-state queue lengths with traffic intensity.

The rest of the paper is organized as follows. Section II contains a detailed presentation of the model that we analyze, namely, a single-hop queueing network. It also defines formally the notions of heavy-tailed and light-tailed traffic, and of delay

[1]On the other hand, when Max-Weight scheduling is combined with Back-Pressure routing in the context of multi-hop networks, there is evidence that delay performance can be poor, see, e.g., the discussion in [4].

stability. In Section III we motivate the subsequent development by presenting, informally and through simple examples, the main results of the paper. In Section IV we analyze the performance of the celebrated Max-Weight scheduling policy. Our general results are accompanied by examples, which illustrate their implications in practical network settings. Section V contains the analysis of the parameterized Max-Weight-α scheduling policy, and the performance that it achieves in terms of delay stability. This section also includes results about the scaling of higher moments of steady-state queue lengths with traffic intensity. We conclude with a discussion of our findings and future research directions in Section VI.

II. MODEL AND PROBLEM FORMULATION

We start with a detailed presentation of the queueing model considered in this paper, together with some necessary definitions and notation.

We denote by $\Re_+$, Z_+, and N the sets of nonnegative reals, nonnegative integers, and positive integers, respectively. The cartesian products of M copies of $\Re_+$ and Z_+ are denoted by $\Re_+^M$ and Z_+^M, respectively.

We assume that time is slotted and that arrivals occur at the end of each time slot. The topology of the network is captured by a directed graph $G = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{E}$ is the set of (directed) edges. Our model involves single-hop traffic flows: data arrives at the source node of an edge, for transmission to the node at the other end of the edge, where it exits the network. More formally, let $F \in N$ be the number of traffic flows in the network. A **traffic flow** $f \in \{1, \ldots, F\}$ consists of a discrete time stochastic arrival process $\{A_f(t); \ t \in Z_+\}$, a source node $s(f)$, and a destination node $d(f)$, with $s(f), d(f) \in \mathcal{N}$, and $(s(f), d(f)) \in \mathcal{E}$. We assume that each arrival process $\{A_f(t); \ t \in Z_+\}$ takes values in Z_+, and is independent and identically distributed (IID) over time. Furthermore, the arrival processes associated with different traffic flows are mutually independent. We denote by $\lambda_f = E[A_f(0)] > 0$ the rate of traffic flow f, and by $\lambda = (\lambda_f; \ f = 1, \ldots, F)$ the vector of the rates of all traffic flows.

Definition 1: (Heavy Tails) A traffic flow f is heavy-tailed if $E[A_f^2(0)] = \infty$, and light-tailed otherwise.

The traffic of flow f is buffered in a dedicated queue at node $s(f)$ (queue f, henceforth.) Our modeling assumptions imply that the set of traffic flows can be identified with the set of edges and the set of queues of the network. The service discipline within each queue is assumed to be "First Come, First Served." The stochastic process $\{Q_f(t); \ t \in Z_+\}$ captures the evolution of the length of queue f. Since our motivation comes from communication networks, $A_f(t)$ will be interpreted as the number of packets that queue f receives at the end of time slot t, and $Q_f(t)$ as the total number of packets in queue f at the beginning of time slot t.

In general, not all edges can be activated simultaneously, e.g., due to interference in wireless networks, or matching constraints in a switch. Consequently, not all traffic flows can be served simultaneously. A set of traffic flows that can be served simultaneously is called a **feasible schedule**. We denote by S the set of all feasible schedules, which is assumed to be an arbitrary subset of the powerset of $\{1, \ldots, F\}$. For simplicity, we assume that all attempted transmissions of data are successful, that all packets have the same size, and that the transmission rate along any edge is equal to one packet per time slot. We denote by $S_f(t) \in \{0, 1\}$ the number of packets that are scheduled for transmission from queue f at time slot t. Note that this is not necessarily equal to the number of packets that are transmitted because the queue may be empty. The time-varying scheduling vector $S(t) = (S_f(t); \ f = 1, \ldots, F) \in S$ is determined by a scheduling policy.

Using the notation above, the **dynamics** of queue f take the form:

$$Q_f(t+1) = Q_f(t) + A_f(t) - S_f(t) \cdot 1_{\{Q_f(t) > 0\}},$$

for all $t \in Z_+$, where $1_{\{Q_f(t) > 0\}}$ denotes the indicator function of the event $\{Q_f(t) > 0\}$. The lengths of the various queues at time slot t are captured by the vector $Q(t) = (Q_f(t); \ f = 1, \ldots, F)$. The vector of initial queue lengths $Q(0)$ is assumed to be an arbitrary element of Z_+^F.

In the context of a communication network, a batch of packets arriving to a queue at any given time slot can be viewed as a single entity, e.g., as a file that needs to be transmitted. We define the **end-to-end delay of a file** of flow f to be the number of time slots that the file spends in the network, starting from the time slot right after it arrives at $s(f)$, until the time slot that its last packet reaches $d(f)$. For $k \in N$, we denote by $D_f(k)$ the end-to-end delay of the kth file of queue f. The vector $D(k) = (D_f(k); \ f = 1, \ldots, F)$ captures the end-to-end delay of the kth files of the different traffic flows.

The following definition gives the precise notion of stability that we use in this paper.

Definition 2: (Stability) The single-hop queueing network described above is stable under a specific scheduling policy, if the vector-valued sequences $\{Q(t); \ t \in Z_+\}$ and $\{D(k); \ k \in N\}$ converge in distribution, and their limiting distributions do not depend on the initial queue lengths $Q(0)$.

Notice that our definition of stability is slightly different than the commonly used definition (positive recurrence of the Markov chain of queue lengths), since it includes the convergence of the sequence of file delays $\{D(k); \ k \in N\}$. The reason is that in this paper we study properties of the limiting distribution of $\{D(k); \ k \in N\}$ and, naturally, we need to ensure that this limiting distribution exists.

Under a stabilizing scheduling policy, we denote by $Q = (Q_f; \ f = 1, \ldots, F)$ and $D = (D_f; \ f = 1, \ldots, F)$ the limiting distributions of $\{Q(t); \ t \in Z_+\}$ and $\{D(k); \ k \in N\}$, respectively. The dependence of these limiting distributions on the scheduling policy has been suppressed from the notation, but will be clear from the context. We refer to Q_f as the steady-state length of queue f. Similarly, we refer to D_f as the steady-state delay of a file of traffic flow f.

The stability of the queueing network depends on the rates of the various traffic flows relative to the transmission rates of the edges and the scheduling constraints. This relation is captured by the stability region of the network.

Definition 3: (Stability Region) [20] The stability region of the single-hop queueing network described above, denoted by Λ, is the set of rate vectors:

$$\left\{\lambda \in \Re_+^F \mid \exists \ \zeta_s \in \Re_+, \ s \in S : \ \lambda \le \sum_{s \in S} \zeta_s \cdot s, \ \sum_{s \in S} \zeta_s < 1\right\}.$$

In other words, a rate vector λ belongs to Λ if there exists a convex combination of feasible schedules that covers the rates of all traffic flows. If a rate vector is in the stability region of the network, then the traffic corresponding to this vector is called **admissible**, and there exists a scheduling policy under which the network is stable.

Definition 4: (Traffic Intensity) The traffic intensity of a rate vector $\lambda \in \Lambda$ is a real number in [0,1) defined as:

$$\rho(\lambda) = \inf\left\{\sum_{s \in S} \zeta_s \mid \lambda \le \sum_{s \in S} \zeta_s \cdot s, \ \zeta_s \in \Re_+, \ \forall s \in S\right\}.$$

Clearly, arriving traffic with rate vector λ is admissible if and only if $\rho(\lambda) < 1$. **Throughout the paper we assume that the arriving traffic is admissible**.

Let us now define the property that we use to evaluate the performance of scheduling policies, namely, the delay stability of a traffic flow.

Definition 5: (Delay Stability) A traffic flow f is delay stable under a specific scheduling policy if the queueing network is stable under that policy and $E[D_f] < \infty$; otherwise, the traffic flow f is delay unstable.

We restrict our attention to scheduling policies that are **regenerative**, i.e., policies under which the evolution of the network is a (possibly delayed) aperiodic and positive recurrent regenerative process. In other words, under a regenerative scheduling policy, the network starts afresh probabilistically at certain time slots (stopping times), and the expected time between consecutive regenerations is finite. We note that for admissible rate vectors, all policies considered in this paper are regenerative. Moreover, the network is stable (in the sense of Definition 2) under any regenerative policy.

Theorem 1: (Delay Instability of Heavy Tails) Consider the single-hop queueing network described above under a regenerative scheduling policy. Every heavy-tailed traffic flow is delay unstable.

Proof: (Sketch) The result follows easily from the Pollaczek-Khinchine formula for the expected delay in a $M/G/1$ queue, and a stochastic comparison argument. The main idea is that in a heavy-tailed traffic flow, the probability that a very big file arrives to the respective queue is relatively high. Combined with the "First Come, First Served" discipline within the queue, this implies that a large number of files,

arriving after the big one, experience very large delays. This is true even if the queue gets served whenever it is nonempty, namely, if the queue is given preemptive priority. Consequently, under any scheduling policy, there is relatively high probability that a large number of files experiences very large delays. This then implies that a heavy-tailed traffic flow is delay unstable. For a formal proof see [12]. ∎

Since there is little we can do about the delay stability of heavy-tailed flows, we turn our attention to light-tailed traffic. The Pollaczek-Khinchine formula for the expected delay in a $M/G/1$ queue implies that the intrinsic burstiness of light-tailed traffic is not sufficient to cause delay instability. However, scheduling in a queueing network couples the statistics of different traffic flows. We will see that this coupling can cause light-tailed flows to become delay unstable, giving rise to a form of **propagation of delay instability**.

III. OVERVIEW OF MAIN RESULTS

In this section we introduce, informally and through simple examples, the main results of the paper and the basic intuition behind them.

Let us start with the queueing system of Figure 1, which consists of two parallel queues and a single server. Traffic flow 1 is assumed to be heavy-tailed, whereas traffic flow 2 is light-tailed. Service is allocated according to the Max-Weight scheduling policy, which is equivalent to "Serve the Longest Queue" in this simple setting. Theorem 1 implies that traffic flow 1 is delay unstable. Our findings imply that **traffic flow 2 is also delay unstable, even though it is light-tailed**. The intuition behind this result is that queue 1 is occasionally very long (infinite, in steady-state expectation) because of its heavy-tailed arrivals. When this happens, and under the Max-Weight policy, queue 2 has to build up to a similar length in order to receive service. A very long queue then implies very large delays for the files of that queue under "First Come, First Served," which leads to delay instability.
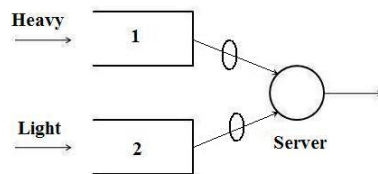

Fig. 1.   Delay instability in parallel queues with heavy-tailed traffic.

Systems of parallel queues have been analyzed extensively in the literature. One of the main reasons is that their simple dynamics often lead to elegant analysis and clean results. However, real-world communication networks are much more complex. In this paper we go beyond parallel queues and analyze queueing networks with more complicated structure. A simple example is the queueing network of Figure 2, where traffic flow 1 is assumed to be heavy-tailed, whereas traffic flows 2 and 3 are light-tailed. The server can serve either queue

1 alone, or queues 2 and 3 simultaneously. This example could represent a wireless network with interference constraints. In this setting the Max-Weight policy compares the length of queue 1 to the sum of the lengths of queues 2 and 3, and serves the "heavier" schedule.
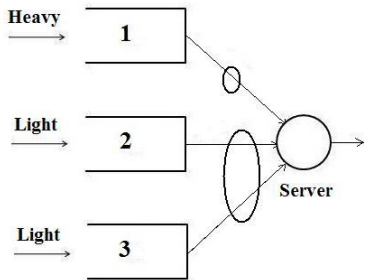

Fig. 2.   Propagation of delay instability: conflicting with heavy-tailed traffic.

The intuition from the previous example suggests that at least one of the queues 2 and 3 has to build up to the order of magnitude of queue 1, in order for these two queues to receive service. In other words, we expect that at least one of the traffic flows 2 and 3 will be delay unstable under Max-Weight. Our findings imply that, in fact, **both traffic flows are delay unstable**. The main idea behind this result is the following: with positive probability, the arrival processes to queues 2 and 3 exhibit their "average" behavior. In that case, the corresponding queues build up slowly and together, which implies that when they claim the server they have both built up to the order of magnitude of queue 1.

The simple networks of Figures 1 and 2 illustrate special cases of a general result: every light-tailed flow that conflicts with a heavy-tailed flow is delay unstable. For more details see Theorem 2 in Section IV.A.
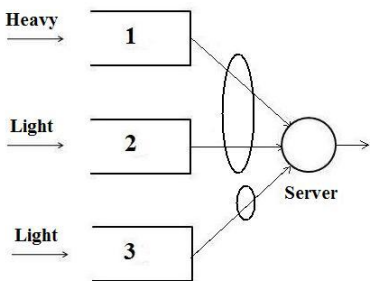

Fig. 3.   Propagation of delay instability: concurring with heavy-tailed traffic.

Going one step further, consider the queueing network of Figure 3. Traffic flow 1 is assumed to be heavy-tailed, whereas traffic flows 2 and 3 are light-tailed. The server can serve either queues 1 and 2 simultaneously, or queue 3 alone. In this setting the Max-Weight policy compares the length of queue 3 to the sum of the lengths of queues 1 and 2, and serves the "heavier" schedule. The intuition from the previous examples suggests that traffic flow 3 is delay unstable, but the real question is the delay stability of traffic flow 2. One would expect that this flow

is delay stable: it is light-tailed itself, and is served together with a heavy-tailed flow, which should result in more service opportunities under Max-Weight. Surprisingly though, we show that **there exist arrival rates within the stability region of this network, such that traffic flow 2 is delay unstable**. The key observation here is that even though traffic flow 2 does not conflict with heavy-tailed traffic, it does conflict with traffic flow 3, which is delay unstable because it conflicts with heavy-tailed traffic. For more details see Propositions 1, 3, and 4 in Sections IV.B and IV.C.

The examples above suggest that in queueing networks with heavy-tailed traffic, delay instability not only appears but propagates through the network under the Max-Weight policy. Seeking a remedy to this situation, we turn to the more general Max-Weight-α scheduling policy. This policy assigns a positive α-parameter to each traffic flow, and instead of using the lengths to calculate the weight of a schedule and serving the heavier one, it uses the respective α powers of the different queue lengths. Our findings imply that in the network of Figure 1, we can guarantee that **traffic flow 2 is delay stable, provided the α-parameter for traffic flow 1 is sufficiently small**. In other words, we prevent the propagation of delay instability. This is a special case of a general result: if the α-parameters of the Max-Weight-α policy are chosen suitably, then the sum of the α-moments of the steady-state queue lengths is finite. For more details see Theorem 3 in Section V.A.

IV. MAX-WEIGHT SCHEDULING

In this section we evaluate the performance of the Max-Weight scheduling policy, with respect to the delay stability of traffic flows. Informally speaking, the "weight" of a feasible schedule is the sum of the lengths of all queues included in it. As its name suggests, the Max-Weight policy activates a feasible schedule with the maximum weight at any given time slot. More formally, under the Max-Weight policy, the scheduling vector $S(t)$ belongs to the set:

$$S(t) \in \arg\max_{(S_f) \in S} \left\{ \sum_{f=1}^{F} Q_f(t) \cdot S_f \right\}.$$

If this set includes multiple feasible schedules, then one of them is chosen uniformly at random. The following lemma states that the network is stable under the Max-Weight policy. Essentially, this result is well-known, e.g., for light-tailed traffic, see [20]; for more general arrivals, see [19]. A subtle point is that in this paper we adopt a somewhat different definition for stability. So, we have to ensure that, apart from the sequences of queue lengths, the sequences of file delays converge as well.

Lemma 1: (Stability under Max-Weight) The single-hop queueing network described in Section II is stable under the Max-Weight scheduling policy.

Proof: Consider the single-hop queueing network of Section II under the Max-Weight scheduling policy. It can be verified that the sequence $\{Q(t);\ t \in Z_+\}$ is a time-homogeneous,

irreducible, and aperiodic Markov chain on the countable state-space Z_+^F. Proposition 2 of [19] implies that this Markov chain is also positive recurrent. Hence, $\{Q(t);\ t \in Z_+\}$ converges in distribution, and its limiting distribution does not depend on $Q(0)$. Based on this, it can be verified that the sequence $\{D(k);\ k \in N\}$ is a (possibly delayed) aperiodic and positive recurrent regenerative process. Therefore, it also converges in distribution, and its limiting distribution does not depend on $Q(0)$; see [18]. ∎

A. Conflicting with Heavy-Tailed Flows

Next, we state one of the main results of the paper, which generalizes our observations from the simple networks of Figures 1 and 2. Before we give the result, though, let us define precisely the notion of conflict between traffic flows.

Definition 6: The traffic flow f conflicts with f', and vice versa, if there exists no feasible schedule in S that includes both f and f'.

Theorem 2: (Conflicting with Heavy Tails) Consider the single-hop queueing network described in Section II under the Max-Weight scheduling policy. Every light-tailed flow that conflicts with a heavy-tailed flow is delay unstable.

Proof: (Sketch) Let h and l be a heavy-tailed and a light-tailed traffic flow, respectively, and suppose that l conflicts with h. Queue h is occasionally very long (infinite, in steady-state expectation), due to the heavy-tailed nature of the traffic that it receives. In order for queue l to get served, the weight of at least one feasible schedule that includes l has to build up to the order of magnitude of queue h. However, with positive probability, the arrival processes of all feasible schedules that include l exhibit their "average" behavior. In that case, queue l builds up at a roughly constant rate, for a time period of the order of magnitude of queue h. This implies that traffic flow l is delay unstable. For a formal proof see Appendix 1. ∎

We emphasize the generality of this result. Namely, a light-tailed flow that conflicts with heavy-tailed traffic is delay unstable, irrespective of: i) its rate; ii) the tail asymptotics of its underlying distribution; iii) whether it is scheduled alone or with other traffic flows. Hence, we view Theorem 2 as capturing a universal phenomenon of instability propagation

B. Non-Conflicting with Heavy-Tailed Flows

So far we have shown that: i) a heavy-tailed traffic flow is delay unstable under any regenerative scheduling policy; and ii) any traffic flow that conflicts with a heavy-tailed flow is delay unstable under the Max-Weight scheduling policy. It seems reasonable, however, that a light-tailed flow that does not conflict with heavy-tailed traffic should be delay stable. Unfortunately, this is not always the case. We demonstrate this by means of simple examples.

Let us come back to the queueing network of Figure 3. The feasible schedules of this network are $\{1,2\}$ and $\{3\}$, and all queues are served at unit rate, whenever the respective schedules are activated. The rate vector $\lambda = (\lambda_1, \lambda_2, \lambda_3)$ is assumed admissible. The following proposition shows that traffic flow 2 is delay unstable if its rate is sufficiently high.

Proposition 1: (Non-Conflicting with Heavy Tails) Consider the single-hop queueing network of Figure 3 under the Max-Weight scheduling policy. If the arriving traffic is admissible and the rates satisfy $\lambda_2 > (1 + \lambda_1 - \lambda_3)/2$, then traffic flow 2 is delay unstable.

Proof: (Sketch) Let us first give the intuition for the special case, where $\lambda_1 = \lambda_3$. Consider sample paths for which a very large file arrives to queue 1; this is a relatively likely event, since traffic flow 1 is heavy-tailed. Queue 3 will build up to the order of magnitude of the large file in queue 1 in order to receive service. Starting from the time slot that their weights become equal, the Max-Weight policy will be draining the weights of the two schedules at the same rate. The period of time until they empty is of the order of magnitude of the large file in queue 1. Now assume that queue 2 stays small throughout this period. If the traffic flows 1 and 3 exhibit their "average" behavior, then each feasible schedule will be activated approximately once every two time slots, since $\lambda_1 = \lambda_3$. However, if $\lambda_2 > 1/2$, queue 2 will build up to the order of magnitude of the large file in queue 1, which is a contradiction.

The intuition for the more general case is based on the following "fluid argument": assume that the arrivals at each queue $f \in \{1,2,3\}$ are a fluid with rate λ_f. The departures from queue f during periods when all queues are nonempty are also assumed to be a fluid with rate μ_f. The Max-Weight policy has the property of draining the weights of the two feasible schedules at the same rate. Hence, when all queues are nonempty, the departure rates are the solution to the following linear system:

$$\lambda_1 + \lambda_2 - \mu_1 - \mu_2 = \lambda_3 - \mu_3$$
$$\mu_1 + \mu_3 = 1$$
$$\mu_1 = \mu_2.$$

The last two equations follow from the facts that Max-Weight is a work-conserving policy, and that queues 1 and 2 are served simultaneously. If the rate at which fluid arrives to queue 2 is greater than the rate at which it departs, i.e.,

$$\lambda_2 > \mu_2 = \frac{1 + \lambda_1 + \lambda_2 - \lambda_3}{3},$$

or, equivalently,

$$\lambda_2 > \frac{1 + \lambda_1 - \lambda_3}{2},$$

then queue 2 builds up over long periods of time, which causes the delay instability of flow 2. A formal proof essentially shows that this fluid model is a faithful approximation of the actual stochastic system (with nonvanishing probability), whenever queue 1 receives a large file; see [12]. ∎

Proposition 1, as well as Propositions 3 and 4 of the next section, capture a "rate-dependent phenomenon" for the

propagation of delay instability.

We conjecture that a converse to Proposition 1 also holds; namely, that queue 2 is delay stable if the arriving traffic is admissible and $\lambda_2 < (1 + \lambda_1 - \lambda_3)/2$.

C. Practical Examples and Implications

We illustrate the implications of the results presented so far in the context of specific network topologies, often used to model real-world communication networks.

Example 1: (Parallel Queues) Consider the network of Figure 4, consisting of n parallel queues and a single server. Networks of parallel queues are often used to model wireless uplinks, downlinks, and call centers. Traffic flow 1 is assumed to be heavy-tailed, whereas the other traffic flows are light-tailed. The scheduling constraints of parallel queues require that no two queues can be served simultaneously. The server is allocated according to the Max-Weight scheduling policy, which in this setting is equivalent to "Serve the Longest Queue."
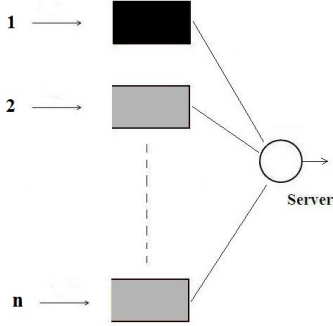

Fig. 4. Delay instability in parallel queues under Max-Weight scheduling: if traffic flow 1 is heavy tailed (black), then all traffic flows are delay unstable (gray.)

Proposition 2: Consider the system of parallel queues depicted in Figure 4, under the Max-Weight scheduling policy. If traffic flow 1 is heavy-tailed, then all traffic flows are delay unstable.

Proof: The result follows easily from Theorems 1 and 2. ∎

Example 2: (Input-Queued Switch) Consider the 2×2 input-queued switch depicted in Figure 5. Input-queued switches are often used to model internet routers. Traffic flow $(1,1)$ is assumed to be heavy-tailed, whereas all other flows are light-tailed. The scheduling constraints of an input-queued switch require that every feasible schedule has to be a matching between the sets of input and output ports. Thus, the feasible schedules of the network are $\{(1, 1), (2, 2)\}$ and $\{(1, 2), (2, 1)\}$. In this setting the Max-Weight scheduling policy activates a matching with the maximum weight.

Proposition 3: Consider the 2×2 input-queued switch depicted in Figure 5, under the Max-Weight scheduling policy. If traffic flow $(1,1)$ is heavy-tailed, then traffic flows $(1,1)$,
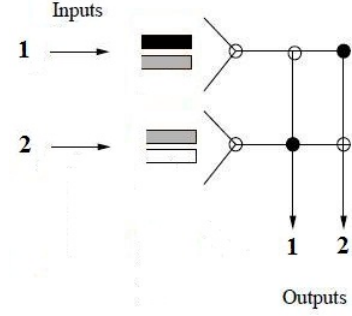

Fig. 5. Delay instability in a data switch under Max-Weight scheduling: if traffic flow (1,1) is heavy tailed (black), then traffic flows (1,2) and (2,1) are delay unstable (gray.) Traffic flow (2,2) is also delay unstable, if its rate is sufficiently high.

$(1,2)$, and $(2,1)$ are all delay unstable. If, additionally, $\lambda_{22} > (2 + \lambda_{11} - \lambda_{12} - \lambda_{21})/3$, then traffic flow $(2,2)$ is also delay unstable.

Proof: The first part of the result follows from Theorems 1 and 2. The second part is proved similarly to Proposition 1; see [12]. ∎

Example 3: (Wireless Ring) Consider the wireless ring network of Figure 6. The network consists of 6 nodes, each of which receives traffic that it transmits to its neighboring node in the clockwise direction. Traffic flow 1 is assumed to be heavy-tailed, whereas all other flows are light-tailed. Due to wireless interference, if a link of the network is activated, then the links within two-hop distance must be inactive; this is the so-called two-hop interference model. Thus, the feasible schedules of the network are $\{1, 4\}$, $\{2, 5\}$, and $\{3, 6\}$.
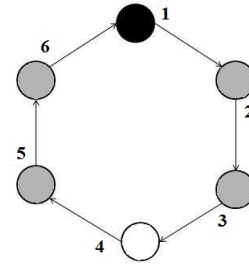

Fig. 6. Delay instability in a wireless ring network under Max-Weight scheduling: if traffic flow 1 is heavy tailed (black), then traffic flows 2, 3, 5, and 6 are delay unstable (gray.) Traffic flow 4 is also delay unstable, if its rate is sufficiently high.

Proposition 4: Consider the wireless ring network depicted in Figure 6, under the Max-Weight scheduling policy. If traffic flow 1 is heavy-tailed, then traffic flows 1, 2, 3, 5, and 6 are all delay unstable. If, additionally, $\lambda_4 > (2 + 2\lambda_1 - \lambda_2 - \lambda_3 - \lambda_5 - \lambda_6)/4$, then traffic flow 4 is also delay unstable.

Proof: The first part of the result follows from Theorems 1 and 2. The second part is proved similarly to Proposition 1; see [12]. ∎

V. Max-Weight-α Scheduling

The results of the previous section suggest that Max-Weight scheduling performs poorly in the presence of heavy-tailed traffic. The reason is that by treating heavy-tailed and light-tailed flows equally, there are very long stretches of time during which heavy-tailed traffic dominates the service. This leads some light-tailed flows to experience very large delays and, eventually, to become delay unstable. Intuitively, by discriminating against heavy-tailed flows one should be able to improve the overall performance of the network, namely to mitigate the propagation of delay instability. One way to do this is by giving preemptive priority to the light-tailed flows. However, priority-based scheduling policies are undesirable because of fairness considerations, and also because they can be unstable in many network settings, e.g., see [9], [15].

Instead, we focus on the Max-Weight-α scheduling policy: given constants $\alpha_f > 0$, for all $f \in \{1, \ldots, F\}$, the scheduling vector $S(t)$ belongs to the set:

$$S(t) \in \arg \max_{(S_f) \in S} \Big\{ \sum_{f=1}^{F} Q_f^{\alpha_f}(t) \cdot S_f \Big\}.$$

If this set includes multiple feasible schedules, one of them is chosen uniformly at random. By choosing smaller values of the α-parameters for heavy-tailed flows and larger values for light-tailed flows, we give a form of partial priority to light-tailed traffic.

A. The Main Result

Let us start with a preview of the **main result** of this section: if the α-parameters of the Max-Weight-α policy are chosen such that $E[A_f^{\alpha_f+1}(0)] < \infty$, for all $f \in \{1, \ldots, F\}$, then the network is stable and the steady-state queue lengths satisfy:

$$E[Q_f^{\alpha_f}] < \infty, \qquad \forall f \in \{1, \ldots, F\}.$$

An earlier work by Eryilmaz *et al.* has given a similar result for the case of parallel queues with a single server; see Theorem 1 of [5]. In this paper we extend their result to a general single-hop network setting. Moreover, we provide an explicit upper bound on the sum of the α-moments of the steady-state queue lengths. Before we do that we need the following definition.

Definition 7: (Covering Number of Feasible Schedules) The covering number k^* of the set of feasible schedules is defined as the smallest number k for which there exist $s^1, \ldots, s^k \in S$ with $\bigcup_{i=1}^{k} s^i = \{1, \ldots, F\}$.

Notice that the quantity k^* is a structural property of the queueing network, and is not related to the scheduling policy or the statistics of the arriving traffic: it is the minimum number of time slots required to serve at least one packet from each flow.

Theorem 3: (Max-Weight-α Scheduling) Consider the single-hop queueing network described in Section II under the Max-Weight-α scheduling policy. Let the intensity of the arriving traffic be $\rho < 1$. If $E[A_f^{\alpha_f+1}(0)] < \infty$, for all

$f \in \{1, \ldots, F\}$, then the queueing network is stable and the steady-state queue lengths satisfy:

$$\sum_{f=1}^{F} E[Q_f^{\alpha_f}] \leq \sum_{f=1}^{F} H\Big(\rho, k^*, \alpha_f, E[A_f^{\alpha_f+1}(0)]\Big),$$

where

$$
\begin{aligned}
&H\Big(\rho, k^*, \alpha_f, E[A_f^{\alpha_f+1}(0)]\Big) \\
&= \begin{cases} \frac{2k^*}{1-\rho} \cdot \Big(E[A_f^{\alpha_f+1}(0)] + 1\Big), & \alpha_f \leq 1, \\ \Big(\frac{2k^*}{1-\rho}\Big)^{\alpha_f} \cdot K^{\alpha_f} + \frac{2k^*}{1-\rho} \cdot K, & \alpha_f > 1, \end{cases}
\end{aligned}
$$

and $K = 2^{\alpha_f - 1} \cdot \alpha_f \cdot \Big(E[A_f^{\alpha_f+1}(0)] + 1\Big)$.

Proof: The proof is based on drift analysis of the Lyapunov function

$$V(Q(t)) = \sum_{f=1}^{F} \frac{1}{\alpha_f + 1} \cdot Q_f^{\alpha_f+1}(t),$$

and use of the Foster-Lyapunov stability criterion and moment bound. For a formal proof see [12]. ∎

B. Traffic Burstiness and Delay Stability

A first corollary of Theorem 3 relates to the delay stability of light-tailed flows.

Corollary 1: (Delay Stability under Max-Weight-α) Consider the single-hop queueing network described in Section II under the Max-Weight-α scheduling policy. If the α-parameters of all light-tailed flows are equal to 1, and the α-parameters of heavy-tailed flows are sufficiently small, then all light-tailed flows are delay stable.

Proof: With the particular choice of α-parameters, Theorem 3 guarantees that the expected steady-state queue length of all light-tailed flows is finite. Little's Law and a stochastic comparison argument relate this result to delay stability. ∎

Combining this with Theorem 1, we conclude that when its α-parameters are chosen suitably, **the Max-Weight-α policy delay-stabilizes a traffic flow, whenever this is possible**.

Max-Weight-α turns out to perform well in terms of another criterion too. Theorem 3 implies that by choosing the α-parameters such that $E[A_f^{\alpha_f+1}(0)] < \infty$, for all $f \in \{1, \ldots, F\}$, the steady-state queue length moment $E[Q_f^{\alpha_f}]$ is finite, for all $f \in \{1, \ldots, F\}$. The following result suggests that this is the best we can do under any regenerative scheduling policy.

Theorem 4: Consider the single-hop queueing network described in Section II under a regenerative scheduling policy. Then,

$$E[A_f^{c+1}(0)] = \infty \implies E[Q_f^c] = \infty, \qquad \forall f \in \{1, \ldots, F\}.$$

Proof: This result is well-known in the context of a M/G/1 queue, e.g., see Section 3.2 of [3]. It can be proved similarly to Theorem 1. ∎

Thus, when its α-parameters are chosen suitably, **the Max-Weight-α policy guarantees the finiteness of the highest possible moments of steady-state queue lengths**.

C. Scaling with Traffic Intensity

Although this paper focuses on heavy-tailed traffic and its consequences, some implications of Theorem 3 are of general interest. In this section we assume that all traffic flows in the network are light-tailed, and analyze how the sum of the α-moments of steady-state queue lengths scales with traffic intensity.

Corollary 2: (Scaling with Traffic Intensity) Let us fix a single-hop queueing network and constants $\alpha \geq 1$ and $B > 0$. The Max-Weight-α scheduling policy is applied with $\alpha_f = \alpha$, for all $f \in \{1, \ldots, F\}$. Assume that the traffic arriving to the network is admissible, and that the $(\alpha + 1)$-moments of all traffic flows are bounded from above by B. Then,

$$\sum_{f=1}^{F} E[Q_f^\alpha] \leq \frac{M(k^*, \alpha, B)}{(1-\rho)^\alpha},$$

where $M(k^*, \alpha, B)$ is a constant that depends only on k^*, α, and B. Moreover, under any stabilizing scheduling policy,

$$\sum_{f=1}^{F} E[Q_f^\alpha] \geq \frac{M'(\alpha)}{(1-\rho)^\alpha},$$

where $M'(\alpha)$ is a constant that depends only on α.

Proof: The first part of the result follows directly from Theorem 3. The second part follows from Theorem 2.1 in [17] and Jensen's inequality. For a formal proof see [12]. ∎

Similar scaling results appear in queueing theory, mostly in the context of single-server queues. More recently, results of this flavor have been shown for particular queueing networks, such as input-queued switches [16], [17]. All the related work, though, concerns the scaling of first moments. Corollary 2 gives the precise scaling of higher order steady-state queue length moments with traffic intensity, and shows that Max-Weight-α achieves the **optimal scaling**.

VI. DISCUSSION

The main conclusion of this paper is that the celebrated Max-Weight scheduling policy performs poorly in the presence of heavy-tailed traffic. More specifically, our findings show that the phenomenon of delay instability not only arises, but can propagate to a significant part of the network. This is somewhat surprising, since Max-Weight is known to perform very well in the presence of light-tailed traffic, at least in single-hop queueing networks.

Another important conclusion is that the Max-Weight-α scheduling policy can be used to alleviate the effects of heavy-tailed traffic, and is even order optimal, if its α-parameters are chosen suitably. However, for Max-Weight-α to perform well, some knowledge of higher order statistics of the traffic flows is required. If the α-parameters are not chosen appropriately, then in light of Theorem 4, this policy may also perform poorly.

Note that our rate-dependent delay instability results (Propositions 1, 3, and 4) concern very simple queueing systems. Despite their simplicity, the proofs of these results are long, technical, and, in some sense, "tailored" to the specific structure of these systems. An interesting direction for future research is a systematic methodology for determining which flows are delay stable, which flows are delay unstable, and over which parts of the stability region.

REFERENCES

[1] S. Borst, O. Boxma, R. Nunez-Queija, B. Zwart (2003). The impact of the service discipline on delay asymptotics. Performance Evaluation, 54, 175-206.

[2] 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.

[3] O. Boxma, B. Zwart (2007). Tails in scheduling. Performance Evaluation Review, 34, 13-20.

[4] L. Bui, R. Srikant, A. Stolyar (2009). Novel architectures and algorithms for delay reduction in back-pressure scheduling and routing. In: Proc. Infocom 2009.

[5] A. Eryilmaz, R. Srikant, J. Perkins (2005). Stable scheduling policies for fading wireless channels. IEEE/ACM Transactions on Networking, 13, 411-424.

[6] R. Gallager (1996). Discrete stochastic processes. Kluwer Academic.

[7] A. Ganti, E. Modiano, J. Tsitsiklis (2007). Optimal transmission scheduling in symmetric communication models with intermittent connectivity. IEEE Transactions on Information Theory, 53, 998-1008.

[8] L. Georgiadis, M. Neely, L. Tassiulas (2006). Resource allocation and cross-layer control in wireless nertworks. Foundations and Trends in Networking, 1, 1-144.

[9] P. R. Kumar, T. Seidman (1990). Dynamic instabilities and stabilization methods in distributed real-time scheduling of manufacturing systems. IEEE Transactions on Automatic Control, 35, 289-298.

[10] 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.

[11] M. Markakis, E. Modiano, J. Tsitsiklis (2009). Scheduling policies for single-hop networks with heavy-tailed traffic. In: Proc. Allerton 2009.

[12] M. Markakis, E. Modiano, J. Tsitsiklis (2011). Max-Weight scheduling in queueing networks with heavy-tailed traffic. In: arXiv e-prints, July 2011.

[13] M. Neely (2008). Order optimal delay for opportunistic scheduling in multi-user wireless uplinks and downlinks. IEEE/ACM Transactions on Networking, 16, 1188-1199.

[14] 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.

[15] A. Rybko, A. Stolyar (1992). Ergodicity of stochastic processes describing the operation of open queueing networks. Probl. Peredachi Inf., 3, 3-26.

[16] D. Shah, J. Tsitsiklis, Y. Zhong (2011). Optimal scaling of average queue sizes in an input-queued switch: an open problem. Queueing Systems, 68, 375-384.

[17] D. Shah, D. Wischik (2008). Lower bound and optimality in switched networks. In: Proc. Allerton 2008.

[18] K. Sigman, R. Wolff (1993). A review of regenerative processes. SIAM Review, 35, 269-288.

[19] A. Stolyar (2004). Maxweight scheduling in a generalized switch: state space collapse and workload minimization in heavy traffic. The Annals of Applied Probability, 14, 1-53.

[20] L. Tassiulas, 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.

[21] D. Williams (1991). Probability with Martingales. Cambridge University Press.

APPENDIX 1 - PROOF OF THEOREM 2

Consider a heavy-tailed traffic flow h, and a light-tailed flow l that conflicts with h. We will show that for admissible traffic flow rates and under the Max-Weight scheduling policy, $E[Q_l]$ is infinite. Then, Little's Law and a stochastic comparison argument will imply that traffic flow l is delay unstable.

The time slots that initiate busy periods of the network constitute renewal times. Denote by X_i the length of the ith renewal period. The random variables $\{X_i; \ i \in N\}$ can be viewed as IID copies of some nonnegative random variable X with finite first moment. This is because the network is stable under the Max-Weight policy and the empty state is accessible from all states, hence recurrent.

We define an instantaneous reward on this renewal process:

$$R^M(t) = \min\{Q_l(t), M\}, \qquad \forall t \in Z_+,$$

where M is a positive integer.

Without loss of generality, assume that a renewal period of the network starts at time slot 0. Consider the set of sample paths where, at time slot 0, queue h receives a file with size B packets, and all other queues receive no traffic; we denote this set of sample paths by $H(B)$. Since the arrival processes of different traffic flows are mutually independent, $P(H(B)) = P(A_h(0) = B) \cdot \prod_{g \neq h} P(A_g(0) = 0)$. For sample paths in $H(B)$, denote by T_B the first time slot when the length of queue h becomes less than or equal to the sum of the lengths of all other queues:

$$T_B = \min \left\{ t > 0 \ \Big| \ \sum_{g \neq h} Q_g(t) \geq Q_h(t) \right\} \cdot 1_{H(B)}.$$

Under the Max-Weight scheduling policy, queue l receives no service until time slot T_B. Moreover, queue h is served at unit rate. So, for sample paths in $H(B)$,

$$B - (T_B - 1) \leq Q_h(T_B) \leq \sum_{g \neq h} Q_g(T_B) = \sum_{g \neq h} \sum_{t=1}^{T_B - 1} A_g(t).$$

A direct consequence of the Strong Law of Large Numbers is the existence of positive constants ϵ and δ, such that the set of sample paths

$$\Delta = \left\{ \Big| \sum_{\tau=1}^{t} A_g(\tau) - \lambda_g \Big| \leq \epsilon \cdot t + \delta, \ \forall t \in N, \ \forall g \neq h \right\},$$

has positive probability. We denote by $\tilde{H}(B)$ the set of sample paths $\Delta \cap H(B)$. Due to the IID nature of the arriving traffic, $P(\tilde{H}(B)) = P(\Delta) \cdot P(H(B))$. For sample paths in $\tilde{H}(B)$, we have

$$T_B - 1 \geq \frac{B - (F - 1) \cdot \delta}{\sum_{g \neq h}(\lambda_g + \epsilon) + 1}.$$

Moreover,

$$Q_l(T_B) = \sum_{t=1}^{T_B - 1} A_l(t) \geq (\lambda_l - \epsilon) \cdot (T_B - 1) - \delta.$$

Consequently, there exist positive constants c and B_0 such that

for every $B \geq B_0$ and any sample path in $\tilde{H}(B)$, we have

$$Q_l(T_B) \geq cB.$$

Since at most one packet from queue l can be served at each time slot, the length of queue l is at least $cB/2$ packets over a time period of length at least $cB/2$ time slots. Hence, the aggregate reward R_{agg}^M, i.e., the reward accumulated over a renewal period, satisfies the lower bound

$$R_{agg}^M \cdot 1_{\{B \geq B_0\}} \cdot 1_{\tilde{H}(B)} \geq \min \left\{ \left(\frac{cB}{2} \right)^2, M^2 \right\} \cdot 1_{\{B \geq B_0\}} \cdot 1_{\tilde{H}(B)}.$$

Then, the expected aggregate reward satisfies

$$E[R_{agg}^M] \geq \sum_{B=1}^{\infty} E[R_{agg}^M \cdot 1_{\{B \geq B_0\}} \cdot 1_{\tilde{H}(B)}]$$

$$\geq P(\Delta) \cdot \prod_{g \neq h} P(A_g(0) = 0)$$

$$\cdot \sum_{B=1}^{\infty} \min \left\{ \left(\frac{cB}{2} \right)^2, M^2 \right\} \cdot 1_{\{B \geq B_0\}} \cdot P(A_h(0) = B).$$

So, there exists a positive constant c', such that

$$c' \cdot E \left[\min \left\{ \left(\frac{cA_h(0)}{2} \right)^2, M^2 \right\} \cdot 1_{\{A_h(0) \geq B_0\}} \right] \leq E[R_{agg}^M]. \quad (1)$$

We have argued that inter-renewal periods have finite expectation, and, clearly, the expected aggregate reward is finite. Then, the Renewal Reward theorem (e.g., see Section 3.4 of [6]) implies that

$$\frac{E[R_{agg}^M]}{E[X]} = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} R^M(t), \qquad \text{w.p.1.} \quad (2)$$

The fact that the reward is a bounded function of an ergodic Markov chain implies that

$$\lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} R^M(t) = E[\min\{Q_l, M\}], \qquad \text{w.p.1.} \quad (3)$$

Eqs. (1)-(3) give

$$\frac{c'}{E[X]} \cdot E \left[\min \left\{ \left(\frac{cA_h(0)}{2} \right)^2, M^2 \right\} \cdot 1_{\{A_h(0) \geq B_0\}} \right] \leq E[\min\{Q_l, M\}].$$

Taking the limit as M goes to infinity on both sides, and using the Monotone Convergence theorem (e.g., see Section 5.3 of [21]), we have that

$$\frac{c' \cdot c^2}{4E[X]} \cdot E[A_h^2(0) \cdot 1_{\{A_h(0) \geq B_0\}}] \leq E[Q_l].$$

Finally, the fact that h is a heavy-tailed traffic flow implies that $E[Q_l]$ is infinite.