

MIT Open Access Articles

*Delay stability of back-pressure policies
in the presence of heavy-tailed traffic*

The MIT Faculty has made this article openly available. **Please share**
how this access benefits you. Your story matters.

Citation: Markakis, Mihalís G., Eytan Modiano, and John N. Tsitsiklis. "Delay Stability of Back-Pressure Policies in the Presence of Heavy-Tailed Traffic." 2014 Information Theory and Applications Workshop (ITA) (February 2014).

As Published: <http://dx.doi.org/10.1109/ITA.2014.6804223>

Publisher: Institute of Electrical and Electronics Engineers (IEEE)

Persistent URL: <http://hdl.handle.net/1721.1/95984>

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

Terms of use: Creative Commons Attribution-Noncommercial-Share Alike



Delay Stability of Back-Pressure Policies in the presence of Heavy-Tailed Traffic

Mihalis G. Markakis
Department of Economics and Business
Universitat Pompeu Fabra
Email: mihalis.markakis@upf.edu

Eytan Modiano
LIDS, MIT
Email: modiano@mit.edu

John N. Tsitsiklis
LIDS, MIT
Email: jnt@mit.edu

Abstract—We study scheduling and routing problems that arise in multi-hop wireline networks with a mix of heavy-tailed and light-tailed traffic. We analyze the delay performance of the widely studied class of Back-Pressure policies, known for their throughput optimality property, using as a performance criterion the notion of delay stability, i.e., whether the expected end-to-end delay in steady state is finite. First, by means of simple examples, we provide insights into how the network topology, the routing constraints, and the link capacities (relative to the arrival rates) may affect the delay stability of the Back-Pressure policy in the presence of heavy-tailed traffic. Next, we illustrate how fluid approximations facilitate the delay-stability analysis of multi-hop networks with heavy-tailed traffic. This approach allows us to derive analytical results that would have been hard to obtain otherwise, and also to build a Bottleneck Identification algorithm, which identifies (some) delay unstable queues by solving the fluid model of the network from certain initial conditions. Finally, we show how one can achieve optimal performance, with respect to the delay stability criterion, by using a parameterized version of the Back-Pressure policy.

I. INTRODUCTION

We study scheduling and routing problems that arise in multi-hop wireline networks with a mix of heavy-tailed (i.e., arrival processes with infinite variance) and light-tailed traffic, and, potentially, multiple source-destination routes for each traffic flow. We analyze the delay performance of the widely studied class of Back-Pressure policies, known for their throughput optimality property. Classical results in queueing theory (e.g., the Pollaczek-Khinchin formula) imply that heavy-tailed flows experience large delays, infinite in steady-state expectation. Thus, we focus on the (policy-dependent) impact of heavy-tailed traffic on light-tailed flows, using as a performance criterion the notion of delay stability, i.e., whether the expected end-to-end delay in steady state is finite.

The class of Back-Pressure policies was introduced in the seminal work of Tassiulas and Ephremides [14] and, since then, numerous studies have analyzed these policies in a variety of settings; see [5] for an overview. A remarkable property of Back-Pressure policies is their throughput optimality, i.e., their ability to stabilize a queueing network whenever this is possible. Moreover, Back-Pressure policies have been combined with congestion control in “cross-layer control”

schemes that are provably stabilizing and utility-optimizing, e.g., see [4], [12].

We are motivated to study heavy-tail phenomena by empirical evidence that traffic in real-world networks exhibits strong correlations and statistical similarity over different time scales. This observation was first made by Leland *et al.* [7] through analysis of Ethernet traffic traces. Subsequent empirical studies have documented this phenomenon in other networks, while accompanying theoretical studies have associated it with arrivals that exhibit high variability. In stochastic models, high variability is typically captured by heavy-tailed probability distributions and/or processes.

The impact of heavy tails has been analyzed extensively in relatively simple queueing systems, e.g., single or multi-server queues; see the survey paper [1]. Moreover, as alluded to above, there is vast literature on the performance of Back-Pressure policies under light-tailed traffic. However, the delay analysis of Back-Pressure policies in networks with a mix of heavy-tailed and light-tailed traffic has only recently attracted attention. Jagannathan *et al.* [6] consider a system with two parallel queues, receiving heavy-tailed and light-tailed traffic while sharing a single server, and determine the queue-length asymptotics under the Generalized Max-Weight policy. In the same setting, Nair *et al.* [11] analyze the role of intra-queue scheduling, i.e., the way that jobs are served within each queue, on queue-length asymptotics.

Closer to the present paper comes our earlier work [9], which studies the delay stability of Max-Weight policies (the single-hop equivalent of Back-Pressure) in networks with a mix of heavy-tailed and light-tailed traffic. There, the decision problem is “one-dimensional,” boiling down to a link-scheduling problem: which subset of servers/communication links to activate at any given time slot. This determines directly which traffic flows are to be served, because in single-hop network models there is, typically, a one-to-one correspondence between links and flows. However, in multi-hop networks, multiple flows may traverse the same communication link. This makes the decision problem “two-dimensional,” comprising of a link-scheduling and a flow-scheduling part. The latter can be roughly stated as follows: given the activated links, which flow to send through each of them. This can be interpreted as a joint scheduling and routing decision. Since the link-scheduling part of the decision problem has been

This work was supported by NSF grants CNS-1217048, CCF-0728554, and CMMI-1234062, ONR grant N00014-12-1-0064, and ARO MURI grant W911NF-08-1-0238.

analyzed extensively in [9], and in the follow-up work [10], here we consider a wireline multi-hop network, where only the flow-scheduling part remains relevant. Thus, in the present paper we focus only on phenomena and insights that arise due to the multi-hop nature of the network and the possibility of multiple source-destination routes.

The main **contributions** of the paper can be summarized as follows.

(i) Through simple examples, we provide insights into how the network topology, the routing constraints, and the link capacities (relative to the arrival rates) may affect the delay performance of the Back-Pressure policy in the presence of heavy-tailed traffic. These insights, in turn, lead to network design principles.

(ii) By extending the results of [10] to the multi-hop setting, we illustrate the use of fluid approximations in delay-stability analysis of multi-hop networks with heavy-tailed traffic. This approach allows us to derive analytical results in cases where purely stochastic arguments would have been hard, e.g., see Proposition 5. Moreover, based on these results, we propose the Bottleneck Identification algorithm, which identifies (some) delay unstable queues/traffic flows by solving the fluid model of the network from certain initial conditions.

(iii) We show how one can achieve optimal performance with respect to the delay stability criterion by using a parameterized version of the Back-Pressure policy, provided the parameters are chosen suitably.

The remainder of the paper is organized as follows. Section II includes a detailed description of a multi-hop wireline network, together with useful definitions and notation. In Section III we show, through simple examples, which “system parameters” may affect the delay performance of the Back-Pressure policy and in what way. In Section IV we illustrate how fluid approximations can be used for proving delay instability results, and present the Bottleneck Identification algorithm. Section V contains the delay-stability analysis of the parameterized Back-Pressure- α policy. We conclude with a brief discussion in Section VI.

II. A MULTI-HOP WIRELINE NETWORK UNDER THE BACK-PRESSURE POLICY

In this section we give a detailed description of a multi-hop switched queueing network, present the Back-Pressure scheduling and routing policy, and provide some useful definitions and notation.

We denote by \mathbb{R}_+ , \mathbb{Z}_+ , and \mathbb{N} the sets of nonnegative reals, nonnegative integers, and positive integers, respectively. Also, $[x]^+$ represents $\max\{x, 0\}$, the positive part of scalar x .

The topology of the network is captured by a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where \mathcal{N} is the set of nodes and \mathcal{L} is the set of directed links. Nodes represent the physical or virtual locations where traffic is buffered before transmission, and edges represent communication links, i.e., the means of transmission. In a queueing context, nodes capture the locations of queues, while each link is a server that carries traffic from queues at

its source node to queues at its destination node. With few exceptions, we use variables i and j to represent nodes, and (i, j) to denote a directed link from node i to node j .

Central to our model is the notion of a traffic flow $f \in \{1, \dots, F\}$, $F \in \mathbb{N}$, which is a long-lived stream of traffic that arrives to the network according to a discrete time stochastic arrival process $\{A_f(t); t \in \mathbb{Z}_+\}$. Each traffic flow $f \in \mathcal{F}$ has a unique source node $s_f \in \mathcal{N}$ where it enters the network, and a unique destination node $d_f \in \mathcal{N}$ where it exits the network. The quantity $A_f(t)$ may be interpreted as the (random) number of packets that flow f brings (exogenously) to s_f at the end of time slot t .

We assume that all arrival processes take values in \mathbb{Z}_+ , and are independent and identically distributed (IID) over time. Furthermore, different arrival processes are independent. We denote by $\lambda_f = \mathbb{E}[A_f(0)] > 0$ the rate of traffic flow f and by $\lambda = (\lambda_f; f = 1, \dots, F)$ the vector of the rates of all traffic flows.

Definition 1: (Heavy/Light Tails) A nonnegative random variable X is heavy-tailed if $\mathbb{E}[X^2]$ is infinite, and is light-tailed otherwise. Moreover, X is exponential-type (light-tailed) if there exists $\theta > 0$ such that $\mathbb{E}[\exp(\theta X)] < \infty$.

A traffic flow is heavy-tailed/light-tailed/exponential-type if the corresponding IID arrival process is heavy-tailed/light-tailed/exponential-type, respectively. We note that there are several definitions of heavy and light tails in the literature. In fact, a random variable is often defined as light-tailed if it is exponential-type, and heavy-tailed otherwise. Definition 1 has been used in the literature on data communication networks, e.g., see [13], due to its close connection to long-range dependence.

For technical reasons we assume the existence of some $\gamma \in (0, 1)$ such that $\mathbb{E}[A_f^{1+\gamma}(0)] < \infty$, for all $f \in \mathcal{F}$.

In the context of data communication networks, a batch of packets arriving to the network 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 $k \in \mathbb{N}$, we denote by $D_f(k)$ the end-to-end delay of the k^{th} file of flow f , and use the vector notation $D(k) = (D_f(k); f = 1, \dots, F)$.

Each traffic flow f has a predetermined set of links $\mathcal{L}_f \subset \mathcal{L}$ that it is allowed to access. We assume that $s_f \neq d_f$ and that there exists at least one directed path from s_f to d_f within the links in \mathcal{L}_f . We also assume that the links in \mathcal{L}_f , together with the associated nodes, form a Directed Acyclic Graph in which nodes s_f and d_f are the only source and sink nodes, respectively. If the set \mathcal{L}_f includes exactly one path from the source to the destination, then we say that flow f has **fixed routing**. On the other hand, if there are more than one source-destination paths, we say that flow f has **dynamic routing**.

Node i belongs to set \mathcal{N}_f if there exists a directed path from s_f to i that includes only links in \mathcal{L}_f . Thus, $\mathcal{N}_f \subset \mathcal{N}$ is the set of nodes that traffic flow f can access. Note that the source

node s_f is trivially included in \mathcal{N}_f , while the destination node d_f is included in \mathcal{N}_f , due to our assumptions on \mathcal{L}_f .

The queueing network operates in discrete time slots, which we index by $t \in \mathbb{Z}_+$. Traffic flow f maintains a queue at every node $i \in \mathcal{N}_f$. We refer to this queue as queue (f, i) and denote its length at the beginning of time slot $t \in \mathbb{Z}_+$ by $Q_{f,i}(t)$. We emphasize that queue (f, i) buffers only packets of flow f . The service discipline within each queue is “First Come, First Served.”

Traffic may arrive to queue (f, i) either exogenously, if i is the source node s_f , or endogenously, through a link in \mathcal{L}_f whose destination node is i . We refer to queue (f, s_f) as the source queue of traffic flow f . We denote by $S_{f,i,j}(t)$ the number of packets that are scheduled for transmission from queue (f, i) through link $(i, j) \in \mathcal{L}_f$. These packets serve as (potential) departures from queue (f, i) and arrivals to queue (f, j) , at time slot t .

We assume that **all links can transmit packets simultaneously**, and that all attempted transmissions are successful. Thus, our queueing model is suitable for wireline applications, but does not capture the interference constraints or the possibility of dropped packets that wireless networks typically exhibit.

Each link can only serve one traffic flow at any given time slot, giving rise to **flow-scheduling constraints**. The set of decisions regarding which flow is scheduled through every link can be interpreted as joint scheduling and routing. For simplicity, we assume that the capacity of all links is equal to one packet per time slot, which implies that $S_{f,i,j}(t) \in \{0, 1\}$, for all $(i, j) \in \mathcal{L}_f$, for all $f \in \mathcal{F}$. We use the shorthand notation $Q(t)$ for the set of queue lengths $\{Q_{f,i}(t); i \in \mathcal{N}_f, f \in \mathcal{F}\}$, and $S(t)$ for the set of scheduling/routing decisions $\{S_{f,i,j}(t); (i, j) \in \mathcal{L}_f, f \in \mathcal{F}\}$, $t \in \mathbb{Z}_+$.

In general, a queue-length-based policy is a sequence of mappings from the history of queue lengths $\{Q(\tau); \tau = 0, \dots, t\}$ to scheduling/routing decisions $S(t)$, $t \in \mathbb{Z}_+$. For much of the paper we focus on a particular stationary and Markovian queue-length-based policy, the **Back-Pressure policy**, which we proceed to describe. At the beginning of each time slot $t \in \mathbb{Z}_+$ set $S(t) = 0$. Then, go through all links in \mathcal{L} in some predetermined order. For each link $(i, j) \in \mathcal{L}$,

(i) compute the maximum differential backlog

$$W_{i,j}(t) = \max_{f:(i,j) \in \mathcal{L}_f} \left\{ [Q_{f,i}(t) - Q_{f,j}(t)]^+ \right\};$$

(ii) if $W_{i,j}(t) = 0$ then set $S_{f,i,j}(t) = 0$, for all f such that $(i, j) \in \mathcal{L}_f$;

(iii) otherwise, pick a flow f^* with maximum differential backlog, i.e.,

$$f^* \in \arg \max_{f:(i,j) \in \mathcal{L}_f} \{Q_{f,i}(t) - Q_{f,j}(t)\}.$$

(If the set on the right-hand side includes multiple flows, f^* is picked uniformly at random.) Then, set $S_{f^*,i,j}(t) = 1$ and $S_{f,i,j}(t) = 0$, for all $f \neq f^*$.

The dynamics of the multi-hop switched queueing network

can be written in the following form:

$$Q_{f,s_f}(t+1) = Q_{f,s_f}(t) - \sum_{j:(s_f,j) \in \mathcal{L}_f} S_{f,s_f,j}(t) + A_f(t), \quad (1)$$

and

$$Q_{f,i}(t+1) = Q_{f,i}(t) - \sum_{j:(i,j) \in \mathcal{L}_f} S_{f,i,j}(t) + \sum_{j:(j,i) \in \mathcal{L}_f} S_{f,j,i}(t), \quad (2)$$

for all $f \in \mathcal{N}_f \setminus \{s_f, d_f\}$. Finally, by convention,

$$Q_{f,d_f}(t) = 0, \quad \forall f \in \mathcal{F}. \quad (3)$$

The initial queue lengths are arbitrary nonnegative integers.

It is possible that not all scheduled transmissions result in actual transmissions, because queues may empty. In those cases, Eqs. (1) and (2) may be violated. In that light, the above set of equations describes the evolution of the network only when the queue lengths are sufficiently large.

As already mentioned above, the main motivation to study the class of Back-Pressure policies is their excellent stability properties. In this paper, the notion of **stability** is defined as convergence in distribution of the sequences of queue lengths and file delays. Moreover, an arrival rate vector λ belongs to the **stability region** Λ if there exists a policy under which the network is stable. An explicit characterization of the stability region of the above multi-hop network can be found in [8].

Lemma 1: (Throughput Optimality of Back-Pressure)

The multi-hop switched queueing network described above is stable under the Back-Pressure policy, for all $\lambda \in \Lambda$.

Proof: For the case of light-tailed traffic, this result follows from the findings in [14]; in the presence of heavy-tailed traffic, it follows from the findings of [3]. For a formal proof the reader is referred to [8]. ■

We denote by $Q_{f,i}$ and $D_{f,i}$ the steady-state length and delay of queue (f, i) , respectively, while we reserve D_f for the end-to-end delay of traffic flow f in steady state.

Definition 2: (Delay Stability) Traffic flow f is delay stable under a specific policy if the network is stable under that policy and $\mathbb{E}[D_f]$ is finite; otherwise, f is delay unstable.

Similarly, queue (f, i) is delay stable if $\mathbb{E}[D_{f,i}]$ is finite, and delay unstable otherwise.

Theorem 1: (Delay Instability of Heavy Tails) Consider the multi-hop switched queueing network described above under any joint scheduling and routing policy. The source queue of every heavy-tailed flow is delay unstable. Consequently, every heavy-tailed flow is delay unstable.

Proof: The result follows easily from the Pollaczek-Khinchin formula for the expected delay in a $M/G/1$ queue, and a stochastic comparison argument. For a formal proof the reader is referred to [8]. ■

Since there is little that can be done regarding the delay stability of heavy-tailed flows, we turn our attention to light-tailed traffic. The Pollaczek-Khinchin formula implies that

light-tailed flows in isolation are delay stable. However, the existence of flow-scheduling constraints couples the evolution of different queues and flows. Below we show that this coupling may cause light-tailed flows to become delay unstable, giving rise to a form of **propagation of delay instability**.

III. DELAY STABILITY ANALYSIS OF BACK-PRESSURE

Which “system parameters” affect the delay performance of the class of Back-Pressure policies in the presence of heavy-tailed traffic? Motivated by this question we study the delay performance of the original Back-Pressure policy described in Section II in terms of the delay stability criterion. By means of simple examples, we investigate the role of the network topology, the routing constraints, and the link capacities (relative to the arrival rates) on the delay stability of queues and flows. Our goal is to derive insights into the type of systems where Back-Pressure is expected to perform well, which can, then, be translated into network design principles.

Our analysis highlights the importance of links that are allowed to serve the source queues of heavy-tailed flows, which we call **bottleneck links**. In mathematical terms, if $f \in \mathcal{F}$ is a heavy-tailed traffic flow, the set of bottleneck links associated with f is defined as

$$\mathcal{B}_f = \{(s_f, i) : (s_f, i) \in \mathcal{L}_f\}.$$

To illustrate the importance of bottleneck links let us consider the simple system of Figure 1, which includes two traffic flows, the heavy-tailed flow 1 and the light-tailed flow 2. Both flows arrive exogenously at node 1, their packets get buffered in the respective queues, eventually get transmitted through link $(1, 0)$, and exit the network as soon as they reach node 0. Link $(1, 0)$ is a bottleneck link, since it is allowed to serve the source queue of flow 1. It is not hard to see that this model is equivalent to a single-server system of two parallel queues, where the Back-Pressure policy reduces to Max-Weight scheduling. Then, Theorem 2 of [9] implies that the light-tailed flow 2 is delay unstable. The main idea behind this result is the following: queue $(1, 1)$ is, occasionally, very long due to the heavy-tailed arrivals that it receives exogenously. During those time periods, flow 1 has very large differential backlog over link $(1, 0)$, which implies that, under the Back-Pressure policy, queue $(2, 1)$ is deprived of service until it builds up a comparable backlog.

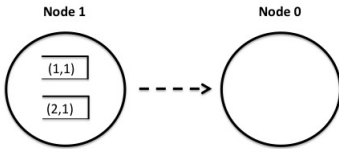


Fig. 1. A single-server system with two parallel queues, cast as a multi-hop network. Traffic flow 1 is heavy-tailed and traffic flow 2 is light-tailed. Since the network has single-hop traffic, the Back-Pressure policy reduces to Max-Weight scheduling. The findings of [9] imply that the light-tailed flow is delay unstable.

More generally, *light-tailed flows experience large delays whenever they have to traverse bottleneck links. Consequently, the delay performance of Back-Pressure depends crucially on the ability of light-tailed flows to avoid bottlenecks, in static or dynamic ways.* This ability is dictated by a number of “system parameters,” as we show below.

A. The Role of Network Topology

We start by illustrating the role of network topology in the delay stability of light-tailed flows. Consider the “line” network depicted in Figure 2. The heavy-tailed flow 1 arrives exogenously at node 1, eventually gets transmitted through link $(1, 0)$, and exits the network as soon as it reaches node 0. The light-tailed flow 2 arrives exogenously at node 2, eventually gets transmitted through link $(2, 1)$ first, and through link $(1, 0)$ next, and exits the network when it reaches node 0. We are interested in the delay stability of flow 2 under the Back-Pressure policy.

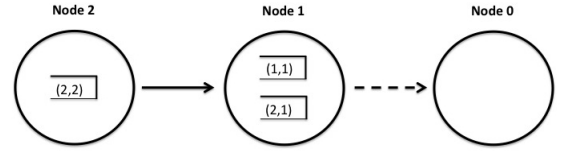


Fig. 2. The heavy-tailed flow 1 enters the network at node 1 and exits at node 0. The light-tailed flow 2 enters the network at node 2, and passing through node 1, it also exits the network at node 0. Traffic flow 2 is delay unstable under the Back-Pressure policy because it has to pass through the bottleneck link $(1, 0)$.

Proposition 1: Consider the network of Figure 2 under the Back-Pressure policy, with an arrival rate vector in the stability region. Traffic flow 2 is delay unstable.

Proof: This result is a special case of Theorem 2. ■

The reason that traffic flow 2 is delay unstable is the topology of the network, and more specifically the fact that the only source-destination path of flow 2 passes through a bottleneck link. We will see shortly that this condition leads to delay instability in more general networks.

B. The Role of Routing Constraints

We continue with the role of routing constraints. Consider the network of Figure 3: the heavy-tailed flow 1 arrives exogenously at node 1, and may reach its destination node 0 through the path $((1, 2), (2, 0))$, or through the path $((1, 3), (3, 0))$. The same applies to the light-tailed flow 2. In other words, both flows have dynamic routing. We are interested in the delay stability of flow 2 under the Back-Pressure policy.

Proposition 2: Consider the network of Figure 3 under the Back-Pressure policy, with an arrival rate vector in the stability region. Traffic flow 2 is delay unstable.

Proof: This result is a special case of Theorem 2. ■

The reason that traffic flow 2 is delay unstable in Figure 3 lies in the routing constraints of the heavy-tailed flow 1, or,

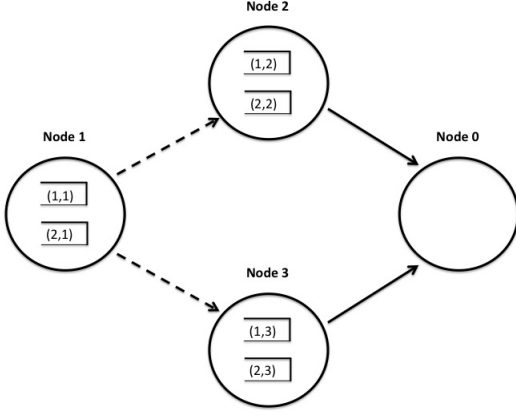


Fig. 3. Both the heavy-tailed flow 1 and the light-tailed flow 2 enter the network at node 1 and exit at node 0. They are both allowed to access all links of the network. Traffic flow 2 is delay unstable under the Back-Pressure policy because it has to pass through, either link (1, 2) or link (1, 3), which are both bottleneck links.

more accurately, the lack of constraints. By not restricting the links that flow 1 is allowed to access, both links (1, 2) and (1, 3) become bottleneck links. In turn, all feasible source-destination paths of flow 2 pass through bottleneck links.

Similar conclusions can be reached if we force both flows 1 and 2 to follow the same fixed route to their destination node.

The insights derived from the simple examples of Figures 1-3 can be unified in a general result. We say that **traffic flow** $f \in \mathcal{F}$ **has to pass through a set of link** $\mathcal{L}' \subset \mathcal{L}$ if every packet arriving at queue (f, s_f) must traverse one of the links in \mathcal{L}' in order to reach d_f . Clearly, whether a traffic flow has to pass through a given set of links or not depends on the network topology, the routing constraints, and the joint scheduling and routing policy applied.

Theorem 2: Consider the multi-hop switched queueing network of Section II under the Back-Pressure policy, with an arrival rate vector in the stability region. Let $f \in \mathcal{F}$ be a light-tailed traffic flow. If there exists a heavy-tailed flow $f' \in \mathcal{F}$ such that f has to pass through the set of bottleneck links $\mathcal{B}_{f'}$, then f is delay unstable.

Proof: Without loss of generality, suppose that the network starts empty. We track the evolution of the network along sample paths of the arrivals where:

- (i) $A_{f'}(0) = b$, for sufficiently large b in the support of $A_{f'}(0)$;
- (ii) $A_f(0) = 0$, for all $f \neq f'$;
- (iii) $\left\{ \left| \sum_{\tau=1}^t A_g(\tau) - \lambda_g \right| \leq \epsilon t + \delta, \forall t \in \mathbb{N}, \forall g \in \mathcal{F} \right\}$, for sufficiently small $\epsilon > 0$ and some $\delta > 0$.

Let H_b be the set of these sample paths. The probability $\mathbb{P}(H_b)$ is bounded away from zero because the arrival processes are mutually independent and IID over time slots.

At time slot zero, the differential backlog of flow f' over every link in $\mathcal{B}_{f'}$ is b , while the differential backlog of flow f over any of those links is zero. Moreover, the differential backlog of flow f' can decrease at rate no more than $2|\mathcal{B}_{f'}|$

packets per time slot (since the capacity of all links is equal to one), while the differential backlog of flow f can increase at rate no more than $(\lambda_f + \epsilon)$ along the sample paths in H_b .

So, for sample paths in H_b , there exist $b_0, k > 0$ for which $Q_{f',i}(t) - Q_{f',j}(t) > Q_{f,i}(t) - Q_{f,j}(t)$, $\forall t < kb$, $\forall b \geq b_0$, for all $(i, j) \in \mathcal{B}_{f'}$.

Consequently, for sample paths in H_b and under the Back-Pressure policy, no packets of flow f are transmitted through any of the links in $\mathcal{B}_{f'}$ during an order $\Omega(b)$ time period.

Now it is useful to keep track of the total number of packets of flow f between the source node s_f and the bottleneck node $s_{f'}$, and to view them as one fictitious queue. Let us denote the length of that queue at time slot t by $\tilde{Q}_f(t)$. The argument above implies that this queue has arrivals at rate no less than $(\lambda_f - \epsilon) > 0$ and no departures, during an order $\Omega(b)$ time period. Hence, there exist constants $c, c' > 0$ such that time slot $cb \in \mathbb{Z}_+$ is in the same busy period as slot zero, and

$$\tilde{Q}_f(cb) = c'b, \quad \forall b \geq b_0.$$

Thus, the aggregate length of this fictitious queue during a busy period is $\Omega(b^2)$ with positive probability. Then, renewal theory and Little's Law imply that the fictitious queue is delay unstable because b is drawn from a heavy-tailed distribution. This also implies the delay instability of traffic flow f , since the delay experienced in the fictitious queue bounds from below the end-to-end delay. ■

C. The Role of Link Capacities

In this section we illustrate the impact of link capacities, relative to the arrival rates, on the delay stability of light-tailed flows. Let us consider a variation of the network of Figure 3, where the heavy-tailed flow 1 has to reach node 0 through the path $((1, 2), (2, 0))$, whereas the light-tailed flow 2 can access all links.

Let us first look at the case where $\lambda_1, \lambda_2 < 1$. The importance of this assumption lies in the fact that it allows flow 2 to route all its traffic through the path $((1, 3), (3, 0))$ whenever the path of the heavy-tailed flow is congested.

Proposition 3: Consider the network of Figure 3 under the Back-Pressure policy, where flow 1 has fixed routing and flow 2 has dynamic routing. If the arrival rates satisfy $\lambda_1, \lambda_2 < 1$, then traffic flow 2 is delay stable.

Proof: See [8]. ■

Now let us consider the case where $\lambda_2 > 1$. It is intuitively clear that, irrespective of the specific routing decisions made at each time slot, a nonvanishing fraction of the traffic of flow 2 has to pass through the bottleneck link (1, 2). This fraction of the traffic experiences large delays under the Back-Pressure policy, which implies that the delays of flow 2 are, on average, large as well.

Proposition 4: Consider the network of Figure 3 under the Back-Pressure policy, where flow 1 has fixed routing and flow

2 has dynamic routing. If $\lambda_2 > 1$ then traffic flow 2 is delay unstable.

Proof: See [8]. ■

D. The Impact of Heavy Tails on Cross-Traffic

Consider the multi-hop network of Figure 4, which includes three traffic flows: the heavy-tailed flow 1, and the light-tailed flows 2 and 3. The source of flow 1 is node 2, whereas the source of flows 2 and 3 is node 1. The destination of flows 1 and 2 is node 3, whereas the destination of flow 3 is node 4.

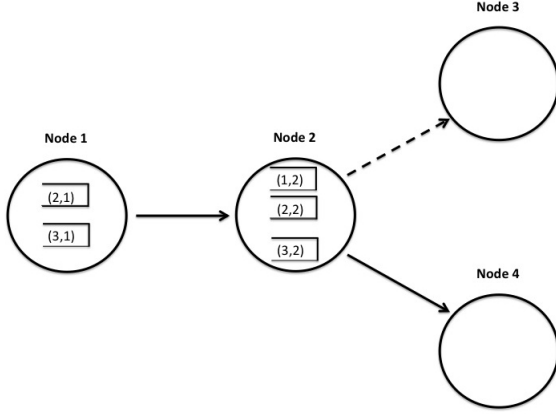


Fig. 4. The heavy-tailed flow 1 enters the network at node 2 and exits at node 3. The light-tailed flow 2 enters the network at node 1 and exits at node 3. The light-tailed flow 3 enters the network at node 1 and exits at node 4. Traffic flow 3 is delay unstable under the Back-Pressure policy if its arrival rate is sufficiently high.

Clearly, traffic flow 2 is delay unstable because it has to pass through the bottleneck link (2,3). So, the real question concerns the delay stability of flow 3, which serves as cross-traffic to flow 2. The following result establishes that flow 3 may or may not be affected by heavy tails, depending on its arrival rate.

Proposition 5: Consider the network of Figure 4 under the Back-Pressure policy, with an arrival rate vector in the stability region. If $\lambda_3 > (2 + \lambda_1 - 2\lambda_2)/3$, then traffic flow 3 is delay unstable. On the other hand, if $\lambda_3 < (2 + \lambda_1 - 2\lambda_2)/3$ and flows 2 and 3 are exponential-type light-tailed, then traffic flow 3 is delay stable.

Proof: (Outline) The first part of the result is based on a sample-path argument. Similarly to the proof of Theorem 2, we track the evolution of the network along sample paths of the arrival processes where (i) a busy period of the network starts with the source queue of the heavy-tailed flow receiving a large batch of size b packets; and (ii) from that point on, all arrival processes exhibit their average behavior. We distinguish between two phases in the evolution of the network along those sample paths.

In the first phase, the length of queue (1,2) is greater than the length of queue (2,2), which implies that link (2,3) transmits only packets of flow 1. Consequently, queues (2,1) and (2,2) build up together and at a constant rate throughout

this phase. This phase terminates when queues (1,2) and (2,2) have the same length, which happens after $\Omega(b)$ time slots. At that point, queues (1,2), (2,1), and (2,2) are all $\Omega(b)$ long.

In the second phase, link (2,3) transmits packets from both flows 1 and 2, and lasts until one of the two queues empties. In order to build some intuition on how the network behaves during this phase, let us assume that both arrivals and departures are fluids with constant rate. Let $\mu_{f,i,j}$ be the rate at which traffic of flow f is transmitted through link (i,j) . Of course, λ_f is the arrival rate at the source queue of flow f . The arrival and departure rates satisfy the linear system

$$(\lambda_2 - \mu_{2,1,2}) - (\mu_{2,1,2} - \mu_{2,2,3}) = \lambda_3 - \mu_{3,1,2}, \quad (4)$$

$$\lambda_1 - \mu_{1,2,3} = \mu_{2,1,2} - \mu_{2,2,3}, \quad (5)$$

$$\mu_{2,1,2} + \mu_{3,1,2} = 1, \quad (6)$$

$$\mu_{1,2,3} + \mu_{2,2,3} = 1. \quad (7)$$

Eq. (4) is due to the fact that the Back-Pressure policy tries to keep the differential backlogs of flows 2 and 3 over link (1,2) the same. In order to achieve this, it determines service rates for the various queues such that the differential backlogs of the link are drained at the same rate. We note that queue (3,2) remains zero throughout both phases, so that the rate of change of its length is also zero. Eq. (5) follows from a similar argument for link (2,3). Eqs. (6) and (7) result from the fact that the service rate of all links is equal to one and Back-Pressure is a work-conserving policy.

The above equations and some simple algebra imply that

$$\mu_{3,1,2} = \frac{2 + \lambda_1 - 2\lambda_2 + 2\lambda_3}{5}.$$

Therefore,

$$\lambda_3 > \mu_{3,1,2} \iff \lambda_3 > \frac{2 + \lambda_1 - 2\lambda_2}{3}.$$

Thus, if $\lambda_3 > (2 + \lambda_1 - 2\lambda_2)/3$, then arrival rate to queue (3,1) exceeds its service rate throughout the second phase. Therefore, queue (3,1) builds up to a length of $\Omega(b)$, which can be shown to imply delay instability. This heuristic argument can be formalized through the use of fluid approximations, in particular, Theorem 3 of Section IV.

The second part of the result is based on drift analysis of the piecewise linear Lyapunov function

$$V(t) = \max \left\{ [Q_{3,1}(t) - Q_{3,2}(t)]^+, [Q_{2,1}(t) - Q_{2,2}(t)]^+ \right\} + [Q_{2,2}(t) - Q_{1,2}(t)]^+,$$

over a sufficiently long time interval. ■

E. The Role of Intersecting Paths

Finally, consider the network of Figure 5: the heavy-tailed flow 1 enters the network at node 1 and exits the network as soon as it reaches node 5. Flow 1 is allowed to access all links, so packets can get to node 4, either through the path $((1,2), (2,4))$, or through the path $((1,3), (3,4))$. After they reach node 4, though, they have to pass through link (4,5) in

order to reach their destination. In that sense, the two paths of flow 1 intersect.

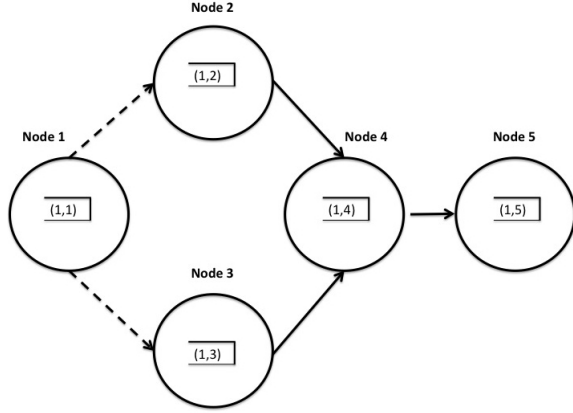


Fig. 5. The heavy-tailed flow 1 enters the network at node 1 and exits the network after it gets transmitted from node 4. Flow 1 is allowed to access all links in the network. Queues (1, 2), (1, 3), and (1, 4) are delay unstable under the Back-Pressure policy because the two alternative paths of flow 1 intersect.

Theorem 1 implies that queue (1, 1) is delay unstable but provides no information regarding the other queues of flow 1, namely queues (1, 2), (1, 3), and (1, 4). Since all links have finite capacities, the endogenous arrivals to those queues are, by definition, light-tailed. So, one might argue that these queues are delay stable, since there are no link-activation constraints and no other traffic flows to compete for service. Somewhat surprisingly, we show that these queues are also delay unstable. This is due to the dynamics induced by the Back-Pressure policy, and the fact that multiple paths intersect. In more detail, the queue at node 4 builds up, and this effect propagates backwards to cause the buildup of queues 2 and 3.

Proposition 6: Consider the network of Figure 5 under the Back-Pressure policy, with an arrival rate vector in the stability region. All queues are delay unstable, apart from queue (1, 5) which is always empty by definition.

Proof: (Outline) Suppose that a busy period of the network starts with a large batch of size b packets arriving to queue (1, 1). From that point on, and throughout an $\Omega(b)$ time period, flow 1 has positive differential backlog over both links (1, 2) and (1, 3). Thus, the set of queues $\{(1, 2), (1, 3), (1, 4)\}$ receive traffic at an aggregate rate of two packets per time slot during that time period. On the other hand, traffic departs from this set of queues at a rate of one packet per time slot, which is the capacity of the outgoing link from node 4. So, the aggregate length of this set of queues builds up at a constant rate over an $\Omega(b)$ time period. The Back-Pressure policy forces the queues to build up together so that, eventually, they all build up to $\Omega(b)$. This can be translated to delay instability since b is drawn from a heavy-tailed distribution. For a formal proof the reader is referred to [8]. ■

Theorem 1 states that the traffic of flow 1 experiences large delays overall, and definitely at the source queue. Whether

these large delays are experienced only at the source queue, or at several other queues as well, may not be as interesting. What is interesting, though, is the case of intersecting paths in networks with multiple flows. There, the delay unstable queues that are created by the intersecting paths may cause cross-traffic light-tailed flows to be delay unstable, similarly to the network of Figure 4. We conjecture that, again, the delay stability of cross-traffic flows depends on the exact values of the arrival rates.

IV. DELAY STABILITY ANALYSIS VIA FLUID APPROXIMATIONS

The findings of the previous section suggest that the delay stability of queues and flows under the Back-Pressure policy depends on a number of parameters, ranging from the network topology and the routing constraints, to the capacities of the various links and the arrival rates. Moreover, this dependence could come in subtle ways, as in the network of Figure 4. So, how do we analyze the delay stability of complex multi-hop networks under the Back-Pressure policy, and in the presence of heavy-tailed traffic?

Direct stochastic analysis of complex networks is typically lengthy and involved, if not intractable. Furthermore, Monte Carlo methods are very slow to converge, or may even fail to converge at all, due to the very nature of heavy-tailed traffic.

Our approach to this methodological challenge relies on the use of fluid approximations: although we cannot track the evolution of sample paths of complex networks directly, we are still able to do it approximately, through the solutions to their fluid models from certain initial conditions (large initial conditions for the source queues of heavy-tailed flows). Then, we can use renewal theory and Little's law to translate sample-path arguments to delay instability results.

The **Fluid Model (FM)** of the multi-hop network of Section II, under the Back-Pressure policy, is a deterministic dynamical system that aims to capture the evolution of its stochastic counterpart on longer time scales. It is defined by the following relations and differential equations, for every time $t \geq 0$ that the derivatives exists (such t is often called a regular time):

$$\begin{aligned} \dot{q}_{f,i}(t) &= - \sum_{j:(i,j) \in \mathcal{L}_f} \dot{s}_{f,i,j}(t) + \sum_{j:(j,i) \in \mathcal{L}_f} \dot{s}_{f,j,i}(t) + \lambda_f \cdot 1_{\{i=s_f\}}, \\ \dot{s}_{f,i,j}(t) &\geq 0, \\ \exists f' : q_{f',i}(t) - q_{f',j}(t) > 0 &\implies \sum_{f:(i,j) \in \mathcal{L}_f} \dot{s}_{f,i,j}(t) = 1, \\ q_{f',i}(t) - q_{f',j}(t) < \max_{f:(i,j) \in \mathcal{L}_f} \left\{ [q_{f,i}(t) - q_{f,j}(t)]^+ \right\} &\implies \dot{s}_{f',i,j}(t) = 0. \end{aligned}$$

The quantity $q_{f,i}(t)$ represents the length of queue (f, i) at time t , and $s_{f,i,j}(t)$ represents the total amount of time that link $(i, j) \in \mathcal{L}_f$ has been serving queue (f, i) up to time t . Thus, the derivative $\dot{s}_{f,i,j}(t)$ is the corresponding service rate

at time t . The last expression is the analogue of Back-Pressure in the fluid domain.

Our convention regarding zero queue lengths in destination nodes provides a final equation for the FM:

$$q_{f,d_f}(t) = 0, \quad \forall f \in \mathcal{F}, \quad \forall t \in [0, T].$$

Henceforth, we use the shorthand notation $q(t)$ for the set of queue lengths $\{q_{f,i}(t); i \in \mathcal{N}_f, f \in \mathcal{F}\}$, and $s(t)$ for the set of scheduling decisions $\{s_{f,i,j}(t); (i,j) \in \mathcal{L}_f, f \in \mathcal{F}\}$.

Fix arbitrary $T > 0$. A **Fluid Model Solution (FMS)** from initial condition $q(0) = q$ is a Lipschitz continuous function $x(\cdot) = (q(\cdot), s(\cdot))$ that satisfies: (i) $x(0) = (q, 0)$; (ii) $q(t) \geq 0$, for all $t \in [0, T]$; (iii) the equations above over $[0, T]$.

A discussion on the formal connection between the fluid model and the original stochastic system can be found in [8].

Fluid approximations of queueing networks under the Back-Pressure policy have been employed by previous studies in order to show stability results, e.g., see [3]. Next, we show how fluid approximations can be used for proving delay instability results in the presence of heavy-tailed traffic.

Theorem 3: Consider the multi-hop network of Section II under the Back-Pressure policy, and its natural FM described above. Let $h \in \mathcal{F}$ be a heavy-tailed traffic flow, and $q^*(\cdot)$ be the (unique) queue-length part of a FMS from initial condition $q_{h,s_h}^*(0) = 1$ and zero for every other queue. If there exists $\tau \in [0, T]$ such that $q_{f,i}^*(\tau) > 0$, then queue (f, i) is delay unstable.

Proof: (Outline) Suppose that there exists $\tau \in [0, T]$ such that $q_{f,i}^*(\tau) > 0$. Then, the existence of a fluid limit, which also guarantees the existence of a FMS, and the uniqueness of the queue-length part of a FMS imply that, after a big arrival to queue (h, s_h) , queue (f, i) builds to the order of magnitude of the heavy-tailed queue with high probability. In turn, renewal theory and Little's Law provide the desired delay instability result. For a formal proof the reader is referred to [8]. ■

As a consequence, we can systematically test for delay instability through the following **Bottleneck Identification algorithm**.

INITIALIZATION: $U = \emptyset$

REPEAT

For every heavy-tailed traffic flow $h \in \mathcal{F}$,

(i) solve the FM with initial condition one for queue (h, s_h) , and zero for all other queues;

(ii) find the set of queues that become positive at any point before the FMS drains, U_h ;

(iii) set $U = U \cup U_h$;

END

Clearly, upon termination of the algorithm, all queues included in U are delay unstable, which, in turn, can be used to identify delay unstable flows.

We illustrate the use of the above algorithmic procedure in the multi-hop network of Figure 4. We consider the set of arrival rates $\lambda_1 = 0.2$, $\lambda_2 = 0.1$, and $\lambda_3 = 0.8$. It can be easily verified that they are in the stability region, so that the

network is stable under the Back-Pressure policy. Also, notice that $\lambda_3 > (1 + \lambda_1 - 2\lambda_2)/3$, which implies that flow 3 is delay unstable according to Proposition 5. Figure 6 shows the FMS for the particular set of rates, and with initial condition one for queue (1,2) and zero for all other queues. The length of queue (3,1) becomes positive before the FMS drains, so Theorem 3 implies that traffic flow 3 is delay unstable, confirming the findings of Proposition 5.

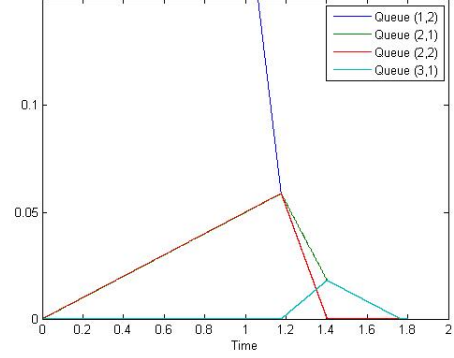


Fig. 6. The FMS of the multi-hop network of Figure 4 from initial condition one for queue (1,2) and zero for the other queues, and arrival rates $\lambda_1 = 0.2$, $\lambda_2 = 0.1$, $\lambda_3 = 0.8$. Since all three traffic flows have fixed routing, Theorem 3 implies that they are all delay unstable.

V. THE BACK-PRESSURE- α POLICY

The results and discussion presented above suggest that the Back-Pressure policy may perform poorly in the presence of heavy-tailed traffic. The reason is that by treating heavy-tailed and light-tailed flows “equally,” there are long stretches of time during which the source queues of heavy-tailed flows dominate the service. This creates bottleneck links, which, in turn, may affect the delay stability of light-tailed flows directly or indirectly.

Intuitively, by discriminating against heavy-tailed flows, one should be able to eliminate bottlenecks and improve the overall performance of the network. One way to do this would be by giving preemptive priority to light-tailed flows. However, priority policies are undesirable because of fairness considerations, and also because they can be unstable in many network settings.

Motivated by the Max-Weight- α scheduling policy, studied in [9] in the context of single-hop networks, here we consider the Back-Pressure- α policy: instead of comparing the differential backlogs of the various flows, we compare the differential backlogs raised to different α -powers, smaller for heavy-tailed flows and larger for light-tailed flows. In that way we give partial priority to light-tailed flows.

More specifically, fix $\alpha_f > 0$, for every traffic flow $f \in \mathcal{F}$. The **Back-Pressure- α policy** makes decisions as follows: at the beginning of each time slot $t \in \mathbb{Z}_+$ set $S(t) = 0$. Then, go through all links in \mathcal{L} in some predetermined order. For each link $(i, j) \in \mathcal{L}$:

(i) compute the maximum α -weighted differential backlog

$$\tilde{W}_{i,j}(t) = \max_{f:(i,j) \in \mathcal{L}_f} \left\{ \left[Q_{f,i}^{\alpha_f}(t) - Q_{f,j}^{\alpha_f}(t) \right]^+ \right\};$$

(ii) if $\tilde{W}_{i,j}(t) = 0$ then set $S_{f,i,j}(t) = 0$, for all $f : (i,j) \in \mathcal{L}_f$;

(iii) otherwise, pick a flow f^* with maximum α -weighted differential backlog, i.e.,

$$f^* \in \arg \max_{f:(i,j) \in \mathcal{L}_f} \left\{ Q_{f,i}^{\alpha_f}(t) - Q_{f,j}^{\alpha_f}(t) \right\}.$$

(If the set on the right-hand side includes multiple flows, f^* is picked uniformly at random.) Then, set $S_{f^*,i,j}(t) = 1$ and $S_{f,i,j}(t) = 0$, for all $f \neq f^*$.

Theorem 4: Consider the multi-hop switched queueing network of Section II under the Back-Pressure- α policy. If $\mathbb{E}[A_f^{\alpha_f+1}(0)]$ is finite, for all $f \in \mathcal{F}$, then the network is stable and

$$\sum_{f \in \mathcal{F}} \sum_{i \in \mathcal{N}_f} \mathbb{E}[Q_{f,i}^{\alpha_f}] < \infty.$$

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

$$V(Q(t)) = \sum_{f \in \mathcal{F}} \sum_{i \in \mathcal{N}_f} \frac{1}{\alpha_f + 1} Q_{f,i}^{\alpha_f+1}(t),$$

and subsequent use of the Foster-Lyapunov stability criterion and moment bound. For a formal proof the reader is referred to [8]. ■

Corollary 1: (Delay Stability under Back-Pressure- α)

Consider the multi-hop network of Section II under the Back-Pressure- α policy. If the α -parameters of all light-tailed flows are equal to one, and the α -parameters of heavy-tailed flows are sufficiently small, then all light-tailed flows are delay stable.

Proof: We recall our standing assumption that all traffic flows have $(1 + \gamma)$ moments, for some $\gamma > 0$. If the α -parameters of all light-tailed flows are equal to one, and the α -parameters of heavy-tailed flows are less than γ , then Theorem 4 and Little's Law imply that every queue of every light-tailed flow is delay stable. The linearity of expectations implies the delay stability of all light-tailed flows. ■

Combining Corollary 1 with Theorem 1, we conclude that *the Back-Pressure- α policy is optimal with respect to delay stability, provided the α -parameters are suitably chosen.*

A special case of the Back-Pressure- α policy has been considered by Bui *et al.* [2], where all α -parameters take the same value. We note that their setting includes just light-tailed traffic and, additionally, the existence of congestion controllers. Thus, the insight that smaller parameter values should be used for heavy-tailed flows, so that light-tailed flows are given some form of priority, does not arise in their model.

VI. CONCLUDING REMARKS

The main objective of this paper was to obtain insights on the delay performance of multi-hop networks with heavy-tailed traffic under Back-Pressure policies. Our analysis highlighted the significance of “bottleneck links,” i.e., links that are allowed to serve the source queues of heavy-tailed traffic flows. The fundamental insight was that traffic flows that have to pass through bottleneck links experience large delays under Back-Pressure. We then investigated reasons that may force a light-tailed flow to pass through a bottleneck link, identifying the following: (i) the network topology, i.e., the source-destination paths that the network offers to the given flow; (ii) the routing constraints, i.e., the a priori decisions regarding which links the particular flow is allowed to traverse; (iii) the link capacities relative to the arrival rates, i.e., whether the combined capacity of non-bottleneck paths is sufficient to support the arrival rate of the flow.

The insights that we derived can be translated into **network design principles**. In particular, heavy-tailed flows should be relatively constrained in terms of the links that they are allowed to access, whereas the network should provide multiple source-destination paths to light-tailed flows; the latter flows should be left unconstrained to dynamically find their way around heavy-tailed traffic. Moreover, these alternate paths should have enough capacity to support the rates of light-tailed traffic. In contrast, leaving heavy-tailed flows unconstrained while forcing light-tailed flows to compete with them could be detrimental to the overall performance of the network.

In terms of **policy design**, we proposed the parameterized Back-Pressure- α policy, and showed that it can delay stabilize all light-tailed flows in the network, provided that its α -parameters are chosen suitably. In order to pick appropriate parameter values, though, some knowledge of higher order moments of the different traffic flows is required.

REFERENCES

- [1] O. Boxma, B. Zwart (2007). Tails in scheduling. *Performance Evaluation Review*, 34(4), pp. 13-20.
- [2] L. Bui, S. Sanghavi, R. Srikant (2009). Distributed link scheduling with constant overhead. *IEEE/ACM Transactions on Networking*, 17(5), pp. 1467-1480.
- [3] J.G. Dai, W. Lin (2005). Maximum pressure policies in stochastic processing networks. *Operations Research*, 53(2), pp. 197-218.
- [4] A. Eryilmaz, R. Srikant (2006). Joint congestion control, routing and MAC for stability and fairness in wireless networks. *IEEE Journal on Selected Areas in Communications*, 24(8), pp. 1514-1524.
- [5] L. Georgiadis, M. Neely, L. Tassiulas (2006). Resource allocation and cross-layer control in wireless networks. *Foundations and Trends in Networking*, 1(1), pp. 1-144.
- [6] K. Jagannathan, M.G. Markakis, E. Modiano, J.N. Tsitsiklis (2012). Queue-length asymptotics for generalized Max-Weight scheduling in the presence of heavy-tailed traffic. *IEEE/ACM Transactions on Networking*, 20(4), pp. 1096-1111.
- [7] W. Leland, M. Taqqu, W. Willinger, D. Wilson (1994). On the self-similar nature of ethernet traffic. *IEEE/ACM Transactions on Networking*, 2(1), pp. 1-15.
- [8] M.G. Markakis (2013). Scheduling in switched queueing networks with heavy-tailed traffic. *PhD Thesis, Massachusetts Institute of Technology*. Available online at: http://www.econ.upf.edu/~mmarkakis/markakis_thesis.pdf.

- [9] M.G. Markakis, E. Modiano, J.N. Tsitsiklis (2013). Max-Weight scheduling in queueing networks with heavy-tailed traffic. *IEEE/ACM Transactions on Networking*, 22(1).
- [10] M.G. Markakis, E. Modiano, J.N. Tsitsiklis (2013). Delay analysis of the Max-Weight policy under heavy-tailed traffic via fluid approximations. *Proceedings of the 51st Allerton Conference*.
- [11] J. Nair, K. Jagannathan, A. Wierman (2013). When heavy-tailed and light-tailed flows compete: the response time tail under generalized Max-Weight scheduling. *Proceedings of the 32nd IEEE Infocom Conference*.
- [12] M.J. Neely, E. Modiano, C. Li (2008). Fairness and optimal stochastic control for heterogeneous networks. *IEEE/ACM Transactions on Networking*, 16(2), pp. 396-409.
- [13] K. Park, W. Willinger (2000). Self-similar network traffic: an overview. *Self-Similar Network Traffic and Performance Evaluation*, Wiley Inc.
- [14] 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(12), pp. 1936-1948.