# Adversarial Queuing Theory

Allan Borodin[*] Jon Kleinberg[†] Prabhakar Raghavan[‡] Madhu Sudan[§]
David P. Williamson[¶]

August 20, 2001

**Abstract**

We consider packet routing when packets are injected continuously into a network. We develop an adversarial theory of queuing aimed at addressing some of the restrictions inherent in probabilistic analysis and queuing theory based on time-invariant stochastic generation. We examine the stability of queuing networks and policies when the arrival process is adversarial, and provide some preliminary results in this direction. Our approach sheds light on various queuing policies in simple networks, and paves the way for a systematic study of queuing with few or no probabilistic assumptions.

## 1 Introduction

Motivated by the study of dynamic continuous (or dynamic) packet routing, we introduce a new approach to the analysis of queuing systems. In the context of packet routing, our objective is to study stability and bounds on routing delay for various networks and scheduling policies. Our approach is based on an adversarial generation of packets (i.e., jobs) so that positive results (e.g., stability and upper bounds on queue sizes) are more robust in that they do not depend on particular probabilistic assumptions about the input sequences. Negative results (e.g., instability) can also be used to suggest corresponding results for more traditional stochastic queuing model assumptions. Moreover, the adversarial model may be a better (or at least safer) model of arrival processes in applications such as heterogeneous ATM networks.

New applications in communications networks and complex manufacturing systems have recently led to significant advances in queuing theory. In particular, the issue of stability for "open multiclass

[*]Department of Computer Science, University of Toronto, Toronto, Canada M5S 3G4. Part of this work was performed while visiting the IBM T.J. Watson Research Center.

[†]Department of Computer Science, Cornell University, Ithaca, New York. Part of this work was performed while visiting the IBM T.J. Watson Research Center. Currently supported in part by an Alfred P. Sloan Research Fellowship and by NSF Faculty Early Career Development Award CCR-9701399.

[‡]IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120.

[§]Laboratory for Computer Science, MIT, Cambridge, MA 02139. Part of this work was performed while this author was at the IBM T.J. Watson Research Center.

[¶]IBM T.J. Watson Research Center, Box 218, Yorktown Heights, NY 10598.

queuing networks" is now much better understood. In a preliminary version of our work (see [9]), we did not fully take into account the dramatic progress in this area. For example, using fluid model limits (see [18, 20, 47]), Bramson [14] has recently shown that for independent and time-invariant input distributions (say, for example, Poisson arrivals), FIFO scheduling is stable for *any* class-independent service time distribution (including constant service time, the standard assumption in packet routing) as long as the necessary load conditions (i.e., total expected arrival rate at any server is less than the expected service rate) are satisfied. Hence queuing theory clearly provides a general methodology for studying continuous packet routing. However, our initial results coupled with the subsequent compelling results of Andrews, Awerbuch, Fernández, Kleinberg, Leighton and Liu [2], and other more recent results (for example, see [1, 3, 4]) demonstrate that the adversarial approach provides another useful perspective to an important and fast evolving field. In fact, as one might expect in such a well-studied field, our "new approach" is not entirely new. The "leaky bucket" model studied by Cruz [16, 17] should also be considered as a (restricted) adversarial model.

We introduce our adversarial model and for definiteness present this model in the context of packet routing. More generally this adversarial approach can be applied to any queuing network; see Tsaparas [51]. Surprisingly, in spite of the power of the adversary, certain networks and scheduling disciplines can be proven to be "universally stable". That is, certain networks are stable under any greedy scheduling rule and certain scheduling rules are stable for any network. In contrast, we will also see that some natural scheduling disciplines can be quite ill-behaved.

## 2 Related work

Queuing theory and analysis is, of course, a well developed and highly utilized field of study. We cannot endeavor to survey all the applicable literature. We will only attempt to briefly review the most relevant results concerning multiclass[1] queuing networks. We will also briefly review some relevant results from the field of packet routing. For basic definitions in queuing theory we refer the reader to standard texts such as Kelly [29], Kleinrock [28] and Walrand [52]. The reader may wish to skip this section and proceed directly to the new definitons in Section 3.

### 2.1 Multiclass queuing networks

We can view dynamic packet routing as one restrictive (but still quite important and non-trivial) type of multiclass queuing network. In particular, while oblivious routing necessitates different classes[2] (i.e., one for each path), the service time distributions are all identical and independent of the class as well as independent of the number of jobs (i.e., packets) waiting for service at any edge.

In a series of papers beginning with the work of Lu and Kumar [35] and Rybko and Stolyar [47], it was shown that the load conditions are not in general sufficient to guarantee stability. These initial "counterexamples" (to what seems a very natural conjecture) were first derived for scheduling rules based on the priority of a class. Seidman [48] (for deterministic arrivals and service times) and then Bramson [12] for Poisson arrivals and exponential service times, showed that even FIFO can be

[1]In a multiclass queuing network, jobs are partitioned into different classes with the understanding that all jobs in a class are indistinguishable in the sense that they have the same service requirements.

[2]Note that from a queuing theory perspective, the Bernoulli (or Markovian) network assumption discussed below avoids the need for multiple classes.

made to be unstable. Indeed Bramson [13] showed that FIFO can be unstable at an arbitrarily small ratio of arrival rate to service rate. All these examples require that the service time distribution at a given server is class dependent.

Rybko and Stolyar [47] were able to analyze the stability of a particular network by first arguing about stability in a "fluid model abstraction" of this network. This approach was then formalized as a general approach in the work of Dai [18] and Dai and Meyn [20]. They show a "meta theorem" whereby stability in the fluid model abstraction of a queuing network implies stability in the given queuing network. We note that the fluid model abstraction only considers the mean rate of service disregarding the nature of the service distribution. (Since the fluid model must necessarily depend on the scheduling rule and there is neither a formal definition of a scheduling rule nor a procedure showing how to transform the scheduling rule into the corresponding equations, we have chosen to call this a meta theorem. But the meaning and applicability of this result is clear.) A Kelly-type network is a possibly multiclass network but one in which the service time distribution at a server is class independent. Using the fluid model approach, Bramson [14] has shown that FIFO scheduling in a Kelly-type network using FIFO scheduling is stable. In particular then, under mild assumptions on the nature of the (mutually independent) input distributions of the classes, continuous packet-routing networks (having constant time service) using FIFO scheduling are stable networks. The Bramson FIFO stability result for the fluid model stands in direct contrast to the result of Andrews *et al.* [2] which shows that FIFO can be unstable in the adversarial model.

## 2.2 The Cruz permanent session model

Cruz [16] introduces an input model designed to capture the burstiness of inputs in communication networks. Cruz defines a $(\sigma_i, \rho_i)$ regulator as an input process that controls the rate of a particular input session (i.e. a path in the network) so that during any time interval $[t_1, t_2]$, the input traffic (for this path) is bounded by $\sigma_i + \rho_i(t_2 - t_1)$ units of traffic. The rate $\rho_i$ corresponds to the long term average rate of flow while the constant $\sigma_i$ bounds the burstiness of the input. Such an input process can be viewed as the output of a "leaky bucket model" of flow control where packets are dropped whenever the above constraints cannot be satisfied. The total induced traffic rate $\rho(e)$ on any edge $e$ is defined as $\sum_{i:e \in P_i} \rho_i$. Andrews [3] refers to this input model as a $(\sigma, \rho)$ regulated session model. Following Andrews and Zhang [4], we will refer to this input model as the permanent session model (in contrast to the adversarial model in which sessions are viewed as temporary). Cruz [16, 17] develops some basic properties of such an input model including the study of stability for acyclic networks relative to a class of greedy scheduling rules.

Tassiulas and Georgiadis [50] adopt the permanent session model for the analysis of routing packets [3] in a (unidirectional) ring $G$. They show that for any greedy scheduling rule $\mathcal{S}$ that the system $(G, \mathcal{A}, \mathcal{S})$ is stable for any adversary $\mathcal{A}$ that corresponds to the $(\sigma, \rho)$ permanent session model under the load constraint that $\rho(e) < 1$ for every edge $e$. Although the concept of an adversary is implicit, these results by Cruz, and Tassiulas and Georgiadis, can be viewed as initial results in the development of an adversarial queuing theory. Following the adversarial FIFO instability result of Andrews *et al.* a recent result of Andrews [3] proves that FIFO can also be unstable in the Cruz permanent session model.

[3]To be more precise, the scheduling policies in [50] also allow processor-sharing rules and the transmission of packets viewed as streams of traffic that can be routed in a cut-through mode as well as discrete packets that are transmitted in a store-and-forward mode. This added generality is not important for our purposes.

For $\rho < 1$, we can model the permanent session model as a $(w, \rho')$ adversary for any $\rho' > \rho$ and a sufficiently big window $w$. Note that this corresponds to a (restricted) adversary which is controlling individual input streams (i.e., particular paths in the network) rather than an adversary which is globally controlling the entire input process. The more general $(w, \rho)$ adversary thus intuitively seems to provide a better model for ATM networks having heterogeneous and frequently changing rates of traffic. Recently Andrews and Zhang [4] have shown that there are known scheduling rules (namely, Generalized Processor Sharing and Weighted Fair Queuing) which are unstable (for the "baseball graph" introduced by Andrews *et al.* [2]) in the full adversarial model but which (by results of Parekh and Gallager [41, 42]) are universally stable in the permanent session model. Our DAG result and the unidirectional ring result of Andrews *et al.* therefore provide (respectively) a significant generalization of the Cruz, and Tassiulas and Georgiadis results.

### 2.3 Static routing

There has been recent work showing how stability results and delay bounds for continuous packet routing may be derived from analogous results in the *static* setting. In a static routing problem, we are given a finite set of packets, each with an assigned path, and we wish to schedule the motion of each packet along its respective path so as to minimize the maximum arrival time. A seminal result in this regard is the well-known bound of Leighton, Maggs, and Rao [32], that every static packet routing problem in which the maximum path length is $D$ and the maximum number of packets using an edge is $C$ has a schedule in which each packet reaches its destination within $O(C + D)$ units of time. A centralized, polynomial-time algorithm to compute such a schedule is given by Leighton, Maggs, and Richa in [33]. Rabani and Tardos [46] and Ostrovsky and Rabani [40] developed distributed algorithms for the static problem of Leighton, Maggs, Rao with bounds that approach the $O(C + D)$ achieved by the centralized methods of [32, 33].

The work of Rabani and Tardos [46] establishes a connection with the adversarial model developed in this paper, in the following sense. Consider any algorithm for the $n$-packet static problem with parameters $C$ and $D$ defined above that produces a schedule with completion time $(1 + \delta)C + g(n)D + f(n)$, where $\delta > 0$ is a constant and $f(n) \geq \log n$. Then it can be converted into an algorithm for continuous packet routing that is stable against an arbitrary $(w, 1 - \varepsilon)$ adversary, where $\varepsilon$ is a function of $\delta$ and each packet has an inverse polynomial probability of being "dropped" before reaching its destination. This latter condition, that an algorithm may drop packets with small probability, is not present in our model.

Broder, Frieze, and Upfal [11] also present a general method for transforming static routing algorithms into continuous ones. They focus on a model in which the switching nodes in the network have bounded buffers. Here a distinction is drawn between *input nodes* in the network, where packets are generated, and all other *intermediate nodes* of the network. The queue at an input node may grow arbitrarily large, but the queue at an intermediate node $v$ can grow no larger than an absolute bound $B$, after which packets cannot enter $v$. Broder *et al.* show that any static packet routing algorithm in this model that satisfies some technical conditions can be transformed into a stable algorithm for continuous packet routing, against an arbitrary adversary of sufficiently small constant rate.

The connections between the static and continuous settings appears implicitly in the development of other protocols in adversarial queuing theory. The analysis showing a polynomial bound on queue size for the randomized universally stable protocol of Andrews *et al.* [2], for example, builds on techniques used in the proof of the static result of Leighton, Maggs, and Rao [32]; the protocol, however, itself has a simple description that is independent of the analysis.

## 2.4 Continuous packet routing

Without any explicit use of queuing theory results, Leighton [31] analyzes one-bend routing on $n \times n$ arrays; the paths in one-bend routing are acyclic and in fact one-bend routing on arrays turns out to be easier to analyze than routing on cyclic networks such as rings. Leighton considers the case where each injected packet has a random destination[4] and packets are injected at each node according to a Bernoulli distribution with rate $\alpha < 4/n$.

These assumptions (on the routing scheme, on the injection rate at each node, and on the fact that packets have random destinations) imply that the induced traffic rate on any edge is less than one (the service rate of each edge). Leighton is able to provide a detailed analysis that shows that for the "farthest-to-go" scheduling discipline at each edge queue, the network is stable and "with high probability" queue sizes are bounded by a small constant. Moreover, the expected delay of a packet is bounded by a constant with high probability. These results are strengthened in Kahale and Leighton [30] where the same results are proven for *any* scheduling discipline. Somewhat weaker results are derived when the underlying graph is a ring. For the ring, stability and bounded delay results are shown for FTG scheduling. The precise nature of these packet delay results seems beyond what we can hope to derive from applying general queuing theory results. On the other hand, the Leighton and Kahale-Leighton results are specific to certain networks and utilize the assumption of random destinations.

Stamoulis and Tsitsiklis [49] consider the case of layered networks under the assumption of Bernoulli routing. In queuing networks with Bernoulli routing, jobs are indistinguishable (i.e. a single class network) and the next server taken is a Markov process (i.e., a probabilistic function of the last server and independent of previous history). In the context of packet routing, the Bernoulli assumption means that the next edge to be traversed is a random function of the last edge traversed and is independent of the packet identity. Stamoulis and Tsitsiklis consider such networks under the assumption of Poisson arrivals. They observe that for layered Bernoulli routing networks, the distribution on the network states (i.e., the queue size at each edge) that results from a network with constant time edge traversal and FIFO scheduling is statistically dominated by the state distribution obtained using a processor (i.e., edge) sharing scheduling discipline. They apply this observation to random destination routing in layered networks (e.g., the butterfly) and in networks that can be layered (e.g., dimension-by-dimension routing to random destinations in the hypercube).[5] Thus bounds on expected queue sizes for constant-time edge traversal can be inferred from results about the analogous network which assumes processor sharing for edge traversal. Since the latter assumption results in a product form network, standard queuing theory analysis can yield constant bounds on expected queue sizes and from this follow (by Little's Theorem) bounds on the expected time in the network.

Following similar experiments by Mitra and Cieslak [38] for the Omega network, Harchol-Balter and Black [6] simulate packet routing on array networks and conjecture that the queue sizes that

[4]We note that continuous packet routing results to date are restricted to the case in which packets have random destinations. One of our main goals is to extend such results to the case where a packet is given a specified destination and path at the time of its injection.

[5]Stamoulis and Tsitsiklis allow a more general, non-uniform, selection of random destinations than in Leighton and Kahale and Leighton; e.g., nearby destinations can be more probable.

obtain under the assumption of exponentially-distributed edge-traversal times are an upper bound for the queue sizes obtained with constant-time edge traversal. Mitzenmacher [39] proves this conjecture for one bend routing on arrays. It is tempting to believe that the experiments and conjecture of Harchol-Balter and Black apply to a much wider context. Indeed, independent of the results in packet routing, there are many queuing-theoretic results pertaining to generalized Jackson networks. In particular, under the appropriate load conditions and some very general assumptions on the arrival and service time distributions, Meyn and Down [37] establish stability for such networks. These results apply to the more general case that each server has its own service time distribution. However, Harchol-Balter and Wolfe [7] give evidence that it will not be a simple task to apply the approach in [49] to the general study of dynamic packet routing. First they show that the "layered" assumption in [49] is not necessary as they are able to derive the same results for any Bernoulli routing network. But they also show that without the Bernoulli network assumption, it is no longer necessarily true that the set of delays for FIFO with constant-time servers is stochastically dominated by processor-sharing servers.

## 3 The adversarial model

We begin with an informal discussion of packet routing, adversaries and scheduling policies. A *routing network* is a directed graph. Time proceeds in discrete *steps*. A *packet* is an atomic entity that resides at a node at the end of any step. A packet must travel along a path in the network from its *source* to its *destination*, both of which are nodes in the network. When the packet reaches its destination, we say that it is *absorbed*. During each step, a packet may be sent from its current node along one of the outgoing edges from that node. At most one packet may travel along any edge of the network in a step. Any packets that wish to travel along an edge $e$ at a particular time step but are not sent wait in a *queue* for edge $e$. The *delay* of a packet is the number of steps which the packet spends waiting in queues.

At each step, an *adversary* generates a set of requests. In this paper a *request* is a path specifying the route followed by a packet. We say that the adversary *injects* a set of packets when it generates a set of requested paths. We restrict ourselves to the case in which the path traversed by each packet is fixed at the time of injection (i.e., non-adaptive routing), so as to be able to focus on the queuing rather than routing aspects of the problem. (Recently, Aiello *et al.* [1] have successfully extended the adversarial model to adaptive routing.)

Clearly an unrestricted adversary can flood the network with packets, demanding more bandwidth than available. Therefore we need to introduce a restriction analogous to the load condition imposed in queuing theory. Let $w$ be an arbitrary positive integer, $e$ any edge in the network and $\tau$ any sequence of $w$ consecutive time steps. We define $N(\tau, e)$ to be the number of paths injected by the adversary during time interval $\tau$ that traverse edge $e$. It is common in packet routing to assume that all paths are simple paths. However, in more general queuing applications, one may not want to assume simple paths and hence in the definition of $N(\tau, e)$ we need not assume simple paths; thus we would count the multiplicity of the number of times a particular path traverses $e$. For any $\rho > 0$, we define a $(w, \rho)$ *adversary* which injects paths subject to the following *load condition*: for every sequence $\tau$ of $w$ consecutive time steps and for every edge $e$, $N(\tau, e)/w \leq \rho$. We say that such a $(w, \rho)$ adversary injects packets at *rate* $\rho$ with *window size* $w$. A rate $\rho$ adversary is a $(w, \rho)$ adversary for some $w$ [6].

[6]We follow the definition of a $(w, \rho)$ adversary as it appears in Andrews *et al.* [2]. In our original paper[9], we

Clearly if the rate $\rho$ were greater than one, an adversary (or any input generation process) would congest some edge (since the service rate of every edge is assumed to be 1) and the network would be unstable. Hence we will hereafter assume that $\rho \leq 1$. We note that in its full generality, there are no computational requirements on how the adversary chooses its requests in any given step and (as will be formalized below). The adversary's choice of input packets is simply a function of the history of the packet routing that has taken place thus far. One can of course place further restrictions on the nature of the adversary. For example, a $(w, \rho)$ *path-packing adversary* is further restricted so that the paths requested at any step must be edge disjoint. A more stringent restriction (a *single path adversary*) requires that at most one packet be injected at each step. In general, we have a *request collection* of permissible request sets; at each step, the adversary must pick one request set (= set of paths) from this request collection.

Additionally, we consider stochastic adversaries. A $(w, \rho)$ *stochastic adversary* is also specified by a request collection, a rate and a window size; now, however, at each step the adversary has a probability distribution (possibly different for each step) over its request collection, and at each step draws a set of requests from the specified distribution. Now, the quantity $N(\tau, e)$ is a random variable induced by the distributions used by the adversary during the time steps $\tau$. If $\tau$ denotes the $w$ time steps $t+1, \ldots, t+w$, we let $H_t$ denote the entire history of packet arrivals up to (and including) step $t$. The appropriate load condition[7] is that for every sequence $\tau$ of $w$ consecutive time steps and for every edge $e$, $E[N(\tau, e)|H_t]/w \leq \rho$. In order to achieve the greatest degree of generality, we would like to impose the fewest conditions on a stochastic adversary and still be able to derive stability results. Once again, it is clear that $\rho \leq 1$ is necessary to avoid flooding some edge. Moreover, it is not hard to see that even for the simplest network consisting of one edge, a rate 1 stochastic adversary can be unstable. Simply consider an adversary which injects zero packets with probability $\frac{1}{2}$ and injects two packets with probability $\frac{1}{2}$ ; after $t$ steps the expected queue size (which is acting like a random walk on the line $[0, \infty)$) is approximately $\sqrt{t}$. Hence we will be concerned with stochastic adversaries having rate $\rho < 1$. However, this bound on the expectation (i.e., the first moment) is not sufficient for our purposes and we will also have to impose a bound on the $p$th moment (see Lemma 2) for $p > 2$. We will then say that a $(w, \rho)$ stochastic adversary is *properly bounded* if there exist constants $p > 2$, $\epsilon > 0$ and $V$ such that for all sequences $\tau$ of $w$ consecutive time steps $t+1, \ldots t+w$ and all edges $e$, $E[N(\tau, e)|H_t]/w \leq 1-\epsilon$ and $E[N(\tau, e)^p|H_t] \leq V$. Note that this model is quite general as it allows the adversary to adaptively modify the distribution at each time step. In particular, it includes the special case (as is often assumed in queuing theory) of a fixed, time-invariant input distribution (e.g., say a Poisson or constant rate arrival process) for each possible request. It also subsumes the case of oblivious packet routing for packets that are generated at each node with randomly chosen destinations (e.g., as studied in [30, 31, 49]). To differentiate such stochastic adversaries from the non-stochastic adversaries defined above, we will refer to the non-stochastic adversaries as *deterministic* adversaries. Clearly, deterministic adversaries are a special case of stochastic adversaries.

To the best of our knowledge, these adversarial models provide the first framework for studying queuing models where we do not essentially assume independent input streams. Moreover, the analysis of these models provide not only results about stability but also quantitative bounds on

only defined and proved results for path-packing adversaries (to be defined) although we implicitly suggested a more general adversary. In subsequent discussions, we independently adopted the general $(w, \rho)$ adversarial framework. Stability proofs for the general adversary that are presented in this paper are extended versions of our earlier proofs given for path-packing adversaries.

[7]In our conference paper[9], we incorrectly stated the load condition as the unconditional expectation $E[N(\tau, e)]/w \leq \rho$.

queue lengths and delays.

A *scheduling policy* specifies, for each edge $e$ and each time step, which packet (amongst those waiting) is to be moved along edge $e$. A *greedy* scheduling policy (called *a work-conserving policy* in the terminology of queuing theory) always specifies some packet to move along edge $e$ if there are packets waiting to use edge $e$. In this paper we restrict ourselves to deterministic greedy scheduling policies. Examples of natural greedy scheduling policies include the following:

- FIFO (First-In-First-Out). This is also called FCFS (First-Come-First-Served).
- LIS (Longest-In-System). A packet originates at a specified time, and priority is given to the packet which has been in the network for the longest amount of time.
- NIS (Newest-In-System). Similar to LIS but now priority is given to the packet that has spent the least amount of time in the network.
- FTG (Furthest-To-Go). Each packet has a prescribed path, and priority is given to the packet which has the largest number of edges still to be traversed.
- NTG (Nearest -To-Go). Similar to FTG but now priority is given to the packet that has the smallest number of edges still to be traversed.

All of these scheduling policies require some tie-breaking rule in order to be unambiguously defined. For our purposes we can assume that whenever we are proving a positive (e.g., stability) result, the adversary can break the tie. For a negative result, we can assume any well-determined tie-breaking rule. It should also be clear that all of these scheduling policies can be extended to more general queuing networks. For example, FTG would be extended to MTTG (Most-service-Time-To-Go) where priority is given to that job which has the most service time remaining (summing the required service over all servers still scheduled to be visited).

In order to make the above discussion more precise we need some additional definitions. A *packet* $P$ is a triple $(ID, p, \tilde{t})$ where $ID$ is some unique identifier, $p$ is the path (from the origin to the destination of the packet) that the packet must traverse, and $\tilde{t}$ is the time that the packet was injected into the network (at its origin). The $state_t(P)$ of a packet $P$ at time $t$ is the vector $(P, i, t_1, \ldots, t_i)$ where $i$ is the number of time steps in which $P$ has traversed an edge in its path and $t_j$ is the time of the $j^{\text{th}}$ such traversal. We define the *configuration* $config_t(G)$ of a network $G$ at time $t$ as $config_t(G) = \{state_t(P_k) | P_k$ is a packet that is present in the network at time $t$ $\}$. Note that the configuration implicitly specifies the states of packets in each (edge) queue of the network. In particular, we will assume that time begins at step $t = 0$ and thus $config_0(G)$ denotes the initial configuration of the network. We can then formally define a *deterministic adversary* $\mathcal{A}$ as a function $\mathcal{A} : \bigcup_{j=0}^{\infty}[j \times \prod_{t=0}^{j} config_t(G)] \to \{P_k | P_k$ is a new packet$\}$ ; that is, $\mathcal{A}$ $(\tilde{t}, \prod_{t=0}^{\tilde{t}} config_t(G))$ specifies the new packets that the adversary $\mathcal{A}$ injects into the network at time $\tilde{t}$ and it does so based on the entire history of the routing up to this point of time. A *stochastic adversary* $\mathcal{A}_S$ is similarly defined as a function $\mathcal{A}_S : \bigcup_{j=0}^{\infty}[j \times \prod_{t=0}^{j} config_t(G)] \to \{D_k | D_k$ is a distribution on new packets$\}$. That is, a stochastic adversary chooses a distribution for the next set of requests and this distribution is a function of the the history of the network configurations. At any time $t$, once the distribution has been chosen, the inputs are then randomly generated according to this distribution. A *deterministic scheduling policy* $\mathcal{S}$ is defined as having the same domain as a deterministic adversary with its range being the set of packets that exist in the system at time $\tilde{t}$; that is, in its full generality we can consider global scheduling rules based on the entire history to date. However, in practice and for the purpose of this paper we will mainly be concerned with distributed scheduling policies where the packet selected to traverse any given edge $e$ is determined by the states of the packets that are presently in the queue associated with edge $e$. We note that all the known deterministic scheduling rules fit into this framework. We could also easily define randomized scheduling policies but will not do so since we do not consider such policies in this paper. However, we note that randomized scheduling policies are often utilized in packet routing; see, for example, Rabani and Tardos [46].

Our main goal will be to prove the *stability* of arbitrary and particular greedy scheduling policies $\mathcal{S}$ on various networks $G$ and against various classes $\Upsilon$ of adversaries[8]. Similar to Andrews *et al.* [2], we define a *network system* as the tuple $(G, \Upsilon, \mathcal{S})$. We will say that a network system $(G, \Upsilon, \mathcal{S})$ is *stable*, if for every initial configuration $C_0(G)$ there is a constant $M$ (which may depend on the size of the network $G$, the initial configuration and the rate and window parameters of the adversary class $\Upsilon$) such that for every adversary $\mathcal{A}$ in the class of adversaries $\Upsilon$, when the network system is executed with initial configuration $C_0(G)$ against adversary $\mathcal{A}$ (i.e., is executed using the deterministic adversary $\mathcal{A}$ to generate packets and the scheduling policy $\mathcal{S}$ to route packets), the maximum number of packets in any queue is bounded by $M$.[9] For stochastic adversaries, we say that the network system $(G, \Upsilon, \mathcal{S})$ is stable if for every initial configuration $C_0(G)$ there is a constant $M$ as above such that for every stochastic adversary $\mathcal{A}_S$ in the class $\Upsilon$, when the network system is executed with initial configuration $C_0(G)$ against adversary $\mathcal{A}_S$ then at all times the expected number of packets in any queue is bounded by $M$. We will say that a network $G$ is *universally stable* with respect to a class $\Upsilon$ of adversaries if $(G, \Upsilon, \mathcal{S})$ is stable for all greedy scheduling policies, and similarly a scheduling policy $\mathcal{S}$ is universally stable if $(G, \Upsilon, \mathcal{S})$ is stable for all networks $G$[10]. Andrews, Awerbuch, Fernández, Kleinberg, Leighton and Liu [2] show that for undirected networks (i.e. in the sense that each undirected edge represents two directed edges) it is decidable if a packet routing network is universally stable. Based on Andrews *et al.*, Goel [25] gives a simple characterization for the universal stability of directed and undirected networks. Bertsimas, Gamarnik, and Tsitsiklis [8] use linear programming to decide universal stability for all 2-station fluid model networks and conjecture that such a test exists for all fluid model queuing networks.

Rather than define stability in terms of maximum queue size, one could also define stability in terms of the maximum delay incurred by any packet. Clearly, if no packet is delayed by more than $M$ steps then the maximum queue size is also bounded by $M$. Conversely, for any $(w, \rho)$ adversary with rate $\rho < 1$, if the maximum queue size is bounded by $M$, then a packet can be delayed in any queue by at most $cw$ steps where $c$ is large enough that $cw > M/(1-\rho)$. (In $cw$ steps, at most $cw\rho$ packets enter the queue and $cw$ packets leave the queue while it remains nonempty.) Clearly, when $\rho = 1$, it is possible to have a given packet delayed forever even though the system is stable under the given definition.

[8]For example, certain networks $G$ are stable against the class of stochastic adversaries of rate $\rho < 1$ for any greedy scheduling rule.

[9]Equivalently, we can say that the maximum number of packets in the system is bounded by some constant $M$. In queuing theory one sometimes defines stability in terms of the existence of a limiting stationary distribution for the state of the network. Our definition is consistent with other standard uses of the term and furthermore one cannot hope for a limiting distribution in an adversarial model.

[10]For the study of the universal stability of a scheduling rule, we can assume that the initial configuration is empty. Informally, we argue that any network can be modified by appending "input trees" to each node and using these input trees to adversarily construct any desired initial configuration.

## 4 Results

It is perhaps tempting to conjecture that for every network $G$, every greedy queuing discipline is stable for a deterministic adversary with injection rate $\rho < 1$ (or even $\rho = 1$). Our first result shows that every directed acyclic network is universally stable for rate 1 deterministic adversaries.

**(1)** *If $G$ is a DAG, and $\Upsilon$ is the class of deterministic rate 1 adversaries, then $(G, \Upsilon, \mathcal{S})$ is stable for every greedy queuing discipline $\mathcal{S}$. That is, every DAG is universally stable against the class of rate 1 deterministic adversaries.*

Unfortunately universal stability against rate 1 adversaries does not extend to the case of graphs with directed cycles, as our next result shows.

**(2)** *Let $G$ be a unidirectional ring network on $n \geq 3$ nodes and let $\Upsilon$ be the class of deterministic rate 1 single path adversaries . Then the network system $(G, \Upsilon, LIS)$ is not stable. Indeed for every initial configuration, there is a rate 1 adversary $\mathcal{A}$, injecting a single path on every step, which will force the network to have an unbounded number of packets. Similarly, FIFO is unstable for rate 1 single path adversaries.*

This instability result is in contrast with the next result.

**(3)** *If $G$ is a unidirectional ring and $\Upsilon$ is the class of deterministic rate 1 adversaries, then $(G, \Upsilon, FTG)$ is stable.*

All our stability results also hold for stochastic adversaries of bounded variance for injection rates bounded away from 1.

In our conference paper [9] we asked if our stability results for the ring and DAGs could be extended to all networks and all greedy scheduling rules. Our results were soon significantly extended by Andrews, Awerbuch, Fernández, Kleinberg, Leighton and Liu [2]. They show that the ring is universally stable with respect to the class of rate $\rho < 1$ deterministic adversaries. They also show that certain scheduling policies $\mathcal{S}$ (such as Newest-in-System (NIS), LIS and FTG) are *universally stable* against the class of rate $\rho < 1$ deterministic adversaries. However, they demonstrate that certain common scheduling policies (such as FIFO and Nearest To Go (NTG)) are *not* universally stable. Specifically, for FIFO (respectively NTG) scheduling, they show that there is some network and initial configuration for which some rate $\rho > .798$ (respectively, rate $\rho > .62$) path-packing adversary causes instability.

In hindsight, the instability of NTG (packet routing) is well motivated by the queuing instability examples[11] of Lu and Kumar [35] and Rybko and Stolyar [47]. Note that our instability result for FIFO on the ring with rate 1 and the FIFO instability result of Andrews *et al.* stands in contrast to Bramson's [14] fluid stability result for class-independent service time networks with FIFO scheduling. We strengthen the Andrews *et al.* NTG instability result by showing the following.

**(4)** *For* every *rate $\rho > 0$, there exists a network and initial configuration for which some rate $\rho$ adversary causes NTG to be unstable. Indeed, although motivated by initial adversarial results, this turns out not to be an adversarial result rather, the "adversary" will only set the initial configuration and then will be injecting all paths at a constant rate!*

[11] These queuing network examples do not show that NTG is unstable for packet routing (where we assume identical service time distributions) but can be easily modified to show packet routing instability for some priority based scheduling policy.

Thus in addition to providing a framework for obtaining very general stability results, the adversarial model is able to distinguish between different scheduling policies and to suggest instability results in more classical settings. This sensitivity to the scheduling policy is not unnatural, yet prior queuing theoretic results do not seem to highlight this.

The remainder of the paper is structured as follows. Section 5 discusses stability bounds for acyclic networks $G$. Section 6 discusses results for the ring. Our NTG instability result is outlined in Section 7. We conclude in Section 8 with a number of open problems.

## 5 DAGs and Meshes

Perkins and Kumar [45] show that for constant rate arrivals there is a class of scheduling rules (called "clear a fraction") with respect to which all acyclic networks are stable. Using a fluid model, Down and Meyn [21] consider a specific acyclic network and show that it is universally stable. Again using a fluid model, Dai [18] shows that every acyclic network is universally stable for rate less than 1. These stability results for fluid models then can be applied to obtain stability results for the more standard "stochastic queuing networks" (i.e., with time-invariant input distributions). We extend these results by showing that for packet routing, acyclic networks are universally stable in the (deterministic and stochastic) adversarial setting. Recently, Gamarnik [23] showed how to use fluid models to show that if a packet routing network is universally stable in the fluid model, then it is universally stable in the deterministic adversarial model. However, the fluid model results do not seem to provide quantitative bounds on queue lengths.

**Theorem 1** *Let $G$ denote an arbitrary directed acyclic graph, $\mathcal{S}$ an arbitrary greedy protocol, and $\Upsilon$ the class of deterministic rate 1 adversaries. Then $(G, \Upsilon, \mathcal{S})$ is stable.*

**Proof:** For $e$ an edge of $G$, let $Q_t(e)$ denote the queue at edge $e$ at time $t$, and let $A_t(e)$ denote the number of packets (not already absorbed) that have arrived by time $t$ and are eventually destined to cross edge $e$. For any adversary $\mathcal{A}$ that injects at rate 1, there exists some window size $w$, such that for any window of time $(t-w, t]$ and for any edge $e$ the adversary can inject at most $w$ packets during this window that are destined to cross the edge $e$.

We define a function $\psi(\cdot)$ inductively on the edges as follows. For an edge $e$, suppose $f_1, \ldots, f_k$ are edges entering the tail of $e$. (Notice there may be no such elements and then we just take $k$ to be zero.)

$$\psi(e) = \max\{2w, Q_0(e)\} + \sum_{i=1}^{k} \psi(f_i).$$

We claim that for all $t = l \cdot w \geq 0$ and all $e \in G$, we have

$$A_t(e) \leq \psi(e) \tag{1}$$

Note that for any time $t'$ such that $t - w < t' < t$, $A_{t'}(e) \leq A_{t-w}(e) + w$. The theorem will then follow, since $\sum_e \psi(e)$ gives an absolute upper bound on the number of packets in the system, in terms of the initial configuration.

The proof of this claim proceeds by induction on $l$. The claim clearly holds when $l = 0$. Now let $t = lw$ for $l \geq 1$. Suppose $e$ is an edge whose tail is entered by edges $f_1, \ldots, f_k$. We consider two cases.

Case 1: $A_{t-w}(e) \leq w + \sum_{i=1}^{k} \psi(f_i)$. In this case, we use the fact that in $w$ time steps the number of new packets inserted that wish to cross the edge $e_j$ is at most $w$. Thus, in this case

$$A_t(e) \leq A_{t-w}(e) + w \leq 2w + \sum_{i=1}^{k} \psi(f_i) \leq \psi(e).$$

Case 2: $A_{t-w}(e) > w + \sum_{i=1}^{k} \psi(f_i)$. By the inductive assertion we have that $A_{t-w}(f_i) \leq \psi(f_i)$. But notice that $A_{t-w}(e)$ is at most $Q_{t-w}(e) + \sum_{i=1}^{k} A_{t-w}(f_i)$. Thus we have

$$\begin{aligned} Q_{t-w}(e) &\geq A_{t-w}(e) - \sum_{i=1}^{k} A_{t-w}(f_i) \\ &> w + \sum_{i=1}^{k} \psi(f_i) - \sum_{i=1}^{k} A_{t-w}(f_i) \\ &\geq w. \end{aligned}$$

In other words, the number of packets queued at edge $e$ must be at least $w$ at the beginning of time step $t - w$. Thus in the next $w$ time steps at least one packet crosses the edge $e$ in every step. But the number of packets inserted which wish to cross this edge may go up by at most $w$ in $w$ time steps. Thus, in this case also, we have $A_t(e) \leq A_{t-w}(e) \leq \psi(e)$. ▮

The upper bound on the number of packets in the system that follows from this proof is exponential in the number of edges (more precisely, the depth) of $G$. Indeed, Andrews *et al.* show that for every $m$, there is an $O(m^2)$-node DAG $G$ and a $(1 - 1/(m+2))$ rate path-packing adversary for which the scheduling rule NIS (or FTG) can be forced to have queue size $2^{m-1}$. Another example of exponentially long (in the depth but not the size) queues for DAGs can be found in Cruz [17]. For the special case of tree networks, the proof of Theorem 1 gives a much better bound; in this case, all the sets $Q(e)$ have linear (in the depth) size.

It is now reasonably easy to obtain essentially the same stability result for stochastic adversaries (satisfying some minimal conditions as previously indicated). We need the following Martingale type lemma due to Pemantle and Rosenthal [44]:

**Lemma 2** *Let $X_1, X_2, \ldots$ be a sequence of nonnegative random variables satisfying the following properties:*

1. *There exists positive constants $\alpha$ and $\beta$ such that for all $x_1, \ldots, x_n$ with $x_n > \beta$,*

$$E[X_{n+1} - X_n \mid X_1 = x_1, ..., X_n = x_n] \leq -\alpha$$

2. *There exists a positive constant $\theta$ and a $p > 2$ such that for all $x_1, \ldots, x_n$*

$$E[|X_{n+1} - X_n|^p \mid X_1 = x_1, ..., X_n = x_n] \leq \theta.$$

*Then there exists $M$ ($M$ is a function of $X_0, \alpha, \beta$ and $\theta$) and $t_0$ such that for all $t \geq t_0$, $E[X_t] \leq M$.*

An example in [44] shows that condition (ii) above cannot be replaced by a bounded second moment. We also note that by bounding higher moments we immediately obtain improved results on the tail probabilities (i.e., the probability that $X_n$ will exceed $c \cdot M$). Indeed, Hajek [26] had previously shown that a bound on exponential moments (replacing condition (ii) above) yields exponentially decreasing bounds on the tail probabilities of the $X_n$ and assuming an absolute bound on $X_{n+1} - X_n$ allows us to use basic results concerning super-Martingales (see, for example, the text by Durrett [22]) to determine bounds on $E[X_n]$ and the tail probabilities.

**Theorem 3** *Let $G$ denote an arbitrary directed acyclic graph, $\mathcal{S}$ an arbitrary greedy protocol, and $\Upsilon$ the class of properly bounded stochastic adversaries with rate $1-\epsilon$ for some $\epsilon > 0$. Then $(G, \Upsilon, \mathcal{S})$ is stable.*

**Proof:** Let $\mathcal{A}_S$ be a rate $(1-\epsilon)$ adversary and let $w$ be an appropriate window size; that is, in any window of $w$ consecutive time steps and for any history of packet injections preceding this window, for every edge $e$, the expected number of packets injected by $\mathcal{A}_S$ that need to cross $e$ is bounded by $(1-\epsilon)w$. For every edge $e$ in $G$, define $Q_t(e)$, $A_t(e)$ as in the proof of Theorem 1. We also define $\psi(e)$ in a manner similar to the proof of Theorem 1. Let $f_1, \ldots, f_k$ be edges entering $e$. Then $\psi(e) = \max\{2w, Q_0(e)\} + \sum_{l=1}^{k}(\psi(f_l) + \epsilon w)$. (Notice that for $\epsilon = 0$ this is exactly what we had in the previous proof.)

Unlike the deterministic adversary case, we will not be able to claim that $A_t(e) \leq \psi(e)$. Instead we use a potential function and show that if the potential function is larger than a specified quantity, the potential is expected to decrease in the next $w$ time steps.

To define the potential, let $e_1, \ldots, e_m$ be a numbering of the edges in topological order (i.e., if $i < j$, then no directed path in $G$ contains the edge $e_j$ followed by the edge $e_i$). The potential function associated with $e_i$ is defined as:

$$\phi_t(e_i) = \max\{A_t(e_i), \psi(e_i)\}.$$

The potential associated with the whole network is:

$$\Phi_t = \sum_i \lambda^{m-i} \phi_t(e_i),$$

where $\lambda$ is a positive real number greater than 1 whose exact value will be chosen later.

Following the notation used in the definition of a stochastic adversary, let $H_{t-w}$ denote the entire history of packet injections up to and including step $t-w$. In order to prove that the expectation of this potential function must decrease if it is too large, we prove the following bounds.

**Claim 4** *1. $A_t(e_i) \geq A_{t-w}(e_i) - w$*

*2. $E[A_t(e_i)|H_{t-w}] \leq A_{t-w}(e_i) + (1-\epsilon)w$.*

*3. Let $i$ be the smallest index (if it exists) such that for every $j < i$, $A_{t-w}(e_j) \leq \psi(e_j)$ and $A_{t-w}(e_i) > \psi(e_i)$. (If such an $i$ does not exist then say $i = m+1$.) Then for every $j < i$, $E[A_t(e_j)] \leq \phi_{t-w}(e_j)$ and (if $i \leq m$) $E[A_t(e_i)|H_{t-w}] \leq A_{t-w}(e_i) - \epsilon w$.*

**Proof:** Part (1) follows from the fact that at most one packet can cross any edge in a given time step. Part (2) follows from the fact that the expected number of packets that are injected in time $(t-w, t]$ and wish to cross the edge $e_i$ is at most $(1-\epsilon)w$. To show part (3) we use an induction argument similar to that in the proof of Theorem 1. Suppose for some $j \le i$ that $e_j$ is an edge whose tail is entered by edges $f_1, \ldots, f_k$. We consider two cases.

Case 1: $A_{t-w}(e_j) \le w + \sum_{l=1}^{k}(\psi(f_l) + \epsilon w)$. Notice first that in this case that $A_{t-w}(e_j) \le \psi(e_j) - w$ and (since $j < i$) we only need to show that $E[A_t(e_j)] \le \phi_{t-w}(e_j)$. We use the fact that in $w$ time steps the expected number of newly inserted packets that wish to cross the edge $e$ is at most $(1-\epsilon)w \le w$. Thus, in this case

$$E[A_t(e_j)|H_{t-w}] < A_{t-w}(e_j) + w \le \psi(e_j) - w + w = \psi(e_j) \le \phi_{t-w}(e_j).$$

Case 2: $A_{t-w}(e_j) > w + \sum_{l=1}^{k}(\psi(f_l) + \epsilon w)$. We will show in this case that $E[A_t(e_j)|H_{t-w}] \le A_{t-w}(e_j) - \epsilon w$. Notice that this is sufficient to prove the assertion for both cases $j < i$ and $j = i$.

Since $f_l = e_{j'}$ for some $j' < i$, by the definition of $i$, we have that $A_{t-w}(f_l) \le \psi(f_l) < \psi(f_l) + \epsilon w$. But $A_{t-w}(e_j)$ is at most $Q_{t-w}(e_j) + \sum_{l=1}^{k} A_{t-w}(f_l)$. Thus we have

$$\begin{aligned} Q_{t-w}(e_j) &\ge A_{t-w}(e_j) - \sum_{l=1}^{k} A_{t-w}(f_l) \\ &> w + \sum_{l=1}^{k}(\psi(f_l) + \epsilon w) - \sum_{l=1}^{k} A_{t-w}(f_l) \\ &\ge w. \end{aligned}$$

In other words, the number of packets queued at edge $e$ must be at least $w$ at the beginning of time step $t-w$. Thus in the next $w$ time steps one packet crosses the edge $e$ in every step. But the expected number of packets inserted which wish to cross this edge may go up by at most $(1-\epsilon)w$ in $w$ time steps. Thus, in this case, we have $E[A_t(e_j)|H_{t-w}] \le A_{t-w}(e_j) - \epsilon w$. ∎

We now conclude by observing that if $\Phi_{t-w} > \beta = \sum_{i=1}^{m} \lambda^{m-i}(\psi(e_i) + \epsilon w)$, then there must exist a smallest $i$ such that $\phi_{t-w}(e_i) > \psi(e_i) + \epsilon w$. Using the claim above, we conclude that

$$\begin{aligned} E[\Phi_t | H_{t-w} \wedge \Phi_{t-w} > \beta] - \Phi_{t-w} &= \sum_{j<i} \lambda^{m-j}(E[\phi_t(e_j)|H_{t-w}] - \phi_{t-w}(e_j)) \\ &\quad + \lambda^{m-i}(E[\phi_t(e_i)|H_{t-w}] - \phi_{t-w}(e_i)) + \sum_{j>i} \lambda^{m-j}(E[\phi_t(e_j)|H_{t-w}] - \phi_{t-w}(e_j)) \\ &\le 0 - \epsilon w \lambda^{m-i} + \sum_{j=i+1}^{m} (1-\epsilon) w \lambda^{m-j} \\ &\le -\epsilon w \lambda^{m-i} + (1-\epsilon) w \lambda^{m-i}/(\lambda - 1) \\ &= \frac{w\lambda^{m-i}}{\lambda - 1}(-\epsilon\lambda + \epsilon + 1 - \epsilon) \\ &\le -w\epsilon/2 \ \ (\text{provided } \lambda \ge 2/\epsilon) \end{aligned}$$

The final inequality above determines our choice of $\lambda$, which we set to $2/\epsilon$. Thus we conclude that if the potential $\Phi_{t-w}$ is high enough, then after $w$ time steps the potential $\Phi_t$ is expected to decrease by at least $\alpha = \epsilon w/2$. It should also be clear that by assuming $\mathcal{A}_S$ is properly bounded, we also know that $E[|\Phi_t - \Phi_{t-w}|^p]$ is bounded by some constant $\theta$ for $p > 2$ since $\Phi_t$ is linear in the $A_t(e)$. Letting $X_i = \Phi_{i \cdot w}$, the theorem follows from Lemma 2 since the potential $\Phi_t$ is an upper bound on the number of packets in the system at time $t$. ▮

**Corollary 5** *Consider any (say two dimensional) mesh as a routing network and consider the case of one bend routing. That is, packets are first sent along their originating row to the destination column and then traverse along this column until reaching the destination. Then for any scheduling rule $\mathcal{S}$, and for the class $\Upsilon$ of rate $1 - \epsilon$ stochastic (and hence deterministic) one bend adversaries, the network $(G, \Upsilon, \mathcal{S})$ is stable.*

**Proof:** We sketch the proof for a two dimensional $N$ by $N$ mesh and deterministic adversaries. The proof is similar to the argument used by Kahale and Leighton [30] when they consider one bend routing in the context of Bernoulli distributed inputs destined to random destinations. Let $\epsilon > 0$ and consider a $(w_1, 1-\epsilon)$ adversary for any window $w_1$. We can consider packets traversing a row as if they are traversing a one dimensional line (i.e., in each direction a very restricted DAG) with inputs generated by the adversary. By Theorem 1, all packets reach their column destination within $c(w_1 + N)$ steps for some constant $c$. Now consider the packets that enter a particular column (say column $j$) during any interval $[t, t+w_2)$ of $w_2$ time steps. Any such packet has either arrived during this window of time or was generated at a step $t' \in [t - c(w_1 + N), t)$. There are therefore at most $c(w_1+N)(1-\epsilon) + w_2(1-\epsilon)$ packets in the network that are destined to traverse any particular edge in column $j$ during the interval $[t, t+w_2)$. Now for any $\epsilon' < \epsilon$, $c(w_1+N)(1-\epsilon)+w_2(1-\epsilon) < w_2(1-\epsilon')$ for $w_2$ sufficiently large. With regard to column $j$ , we can think of the $(w_1, 1-\epsilon)$ adversary as a $(w_2, 1-\epsilon')$ adversary. Applying Theorem 1 again (for column $j$ thought of as a line) , every packet that enters column $j$ at time $t$ will reach its destination by time $t + c(w_2 + N)$.

▮

This proof easily generalizes to any dimensional array and also to stochastic adversaries. The same idea can be applied to toroidal networks using the Andrews *et al.* [2] universal stability results for unidirectional rings. However without the one bend routing assumption, the instability results of Andrews *et al.* [2] show that (two dimensional) meshes are not universally stable networks.

## 6 The ring

There are now a number of independent results (for different queuing model assumptions) showing that the (unidirectional) ring is universally stable for rate $< 1$ (i.e., the total rate of service required at any server is less than 1). For class independent service rates, Dai and Weiss [19] prove universal stability for fluid models (and hence for time-invariant stochastic queuing networks under the same assumption of class independent service rates). Tassiulas and Georgiadis [50] establish the analogous result for packet routing using the Cruz [16] leaky bucket model. Andrews *et al.* [2] show that the ring is universally stable with respect to deterministic adversaries. We shall now show that the scheduling rule FTG is stable at rate 1 for the ring. In contrast, neither LIS (which is a universally stable scheduling rule at any rate less than 1) nor FIFO are stable at rate 1 for the ring. We then extend the FTG proof to stochastic adversaries with rate $1 - \epsilon$.

Throughout this section, our underlying graph $G$ will be the $n$-node unidirectional cycle, with vertices numbered $0, \ldots, n-1$. For the purpose of packet routing the unidirectional assumption is not usually a restriction as we most often assume simple one directional paths and clearly paths in the clockwise direction will not interfere with paths in the counterclockwise direction. For definiteness let us assume packets are being routed in a clockwise direction. Even this case is quite non-trivial, both from the point of view of classical queuing theory and within our adversarial setting. Indeed as demonstrated in Andrews *et al.*, a "slight" extension of the ring network (allowing two edges between each pair of adjacent nodes) is enough to show that common scheduling policies like FIFO and NTG (nearest-to-go) can be unstable.

## 6.1 Instability of LIS and FIFO at rate 1

We show in Section 6.2 that the FTG protocol is stable at injection rate 1 on the ring. On the other hand, we now exhibit simple adversaries with injection rate 1 that cause instability on the ring for the Longest-in-System protocol (priority to the packet that was injected longest ago) and the FIFO protocol (queues are maintained in First–Come–First–Served fashion).

**Theorem 6** *There is a deterministic adversary $\mathcal{A}$ (respectively, an adversary $\mathcal{A}'$) that injects single paths onto the ring at rate* 1*, such that $\mathcal{A}$ (respectively, $\mathcal{A}'$) will force the scheduling rule LIS (respectively, the scheduling rule FIFO) to have unbounded size queues.*

**Proof:** We first describe the adversary $\mathcal{A}$ for LIS. For simplicity of presentation, assume that each path requested by $\mathcal{A}$ will be a "self-loop" — a path which traverses all the edges of the ring in sequence. It is not difficult to refine this argument so that the adversary injects shorter paths.

$\mathcal{A}$ works as follows:

- For $k = 1, 2, 3, \ldots$
  - Inject $kn$ self-loops in sequence at node 1.
  - Inject $kn$ self-loops in sequence at node 0.

It is easy to verify by induction on $k \geq 1$ that at the end of iteration $k$ of this process, there will be one packet at node 1 (destined for node 0) and $kn - 1$ packets queued at node 0 (also destined for node 0). Thus the number of packets becomes unbounded.

The adversary for the FIFO case is similar but a bit more complicated. To simplify the presentation, we prove this case by contradiction. Say there is an absolute bound $M$ such that the number of packets in the system is bounded by $M$ for every adversary. Let us first show how to construct an adversary that contradicts this bound $M$. The adversary, $\mathcal{A}'$ first does an "initial" phase 0 and then works in phases indexed by $k$ as follows:

- For $k = 1, 2, 3, \ldots$
  - Inject $(M + 1)n$ self-loops in sequence at node $k(\bmod n)$.

The invariant that will be established is that at the end of phase $k$ all packets in the network are destined for node $k$ with one packet at each queue other than the queue on the edge from $k \to k+1$, which has $Q_k$ packets in its queue where the $Q_k$'s form a monotone increasing sequence in $k$. Thus by the end of $M$ phases we derive a contradiction.

We assume an empty initial configuration. For phase 0, the adversary simply injects $n$ self loops at node 0. Thus the invariant is initially satisfied for $k = 0$ with $Q_0 = 1$. Assume that the invariant is true at the end of phase $k-1$. We make some observations on the transient behavior in phase $k$. We start by observing that once the invariant is established for some time step during phase $k$, it continues to hold in all subsequent time steps in phase $k$. Next notice that throughout this phase there is at most one packet queued in every queue other than queues $k$ and $k-1$. The number of packets queued at $k-1$ is monotone non-increasing and the number of packets queued at $k$ is monotone non-decreasing. Furthermore, since the queues at nodes $k$ and $k-1$ are of length at most $M$, no packet waits at any queue for more than $M$ time steps. Thus after at most $Mn$ time steps in phase $k$, every packet injected in phase 0 to $k-1$ has reached its destination. In at most $n$ more time steps, we reach the invariant that all the queues other than $k-1$ and $k$ have exactly one packet in their queue. To conclude we need to show that the queue at node $k-1$ does eventually drain down to having 0 and then 1 packet; and that the queue size at $k$ at the end of phase $k$ is strictly larger than the queue size at $k-1$ at the beginning of this phase.

To verify the first part notice that the queue size at node $k-1$ (which consists of both phase $k-1$ packets destined for node $k-1$ as well as phase $k$ packets destined for node $k$) is upper bounded by the number of packets in the system with destination $k-1$. This observation is easily verified by induction on time. Moreover, since the queue size goes down every time a packet with destination $k-1$ reaches its destination, the queue becomes empty (for one time step). Finally we need to verify that $Q_k$, the number of packets at queue $k$ at the end of phase $k$, is greater than $Q_{k-1}$. Assume otherwise. Then this implies that the total work remaining in the system has not increased. But this can not be the case, since there is at least one queue, namely at node $k-1$, that was idle for one time step (immediately after the last packet destined for $k-1$ reached its destination) during phase $k$. Since $n$ units of work are added at each time step, the only way the workload does not go up is if every queue remains non-idle in every time step in phase $k$. This concludes the proof for the bound $M$.

In order to show that there is (one) adversary that defeats every bound $M$, we simply keep changing the goal of the adversary so that after it defeats a given $M$, it resets all queues to be empty (by not doing any injections) and then proceeds to defeat $M+1$, etc. ▌

## 6.2 The Furthest-to-Go Protocol

In this section, we prove that the FTG protocol is stable for the ring, first (in contrast to the LIS and FIFO instability result at rate 1) for a deterministic adversary at injection rate 1, and then for a stochastic adversary at injection rate $1-\epsilon$. We are assuming that all packets are traveling a simple path (say in a clockwise direction).

We first define a quantity $A(i,j,t)$ for $0 \leq i,j \leq n-1$. (Throughout this section, all arithmetic on node names is $\bmod n$.) The quantity $A(i,j,t)$ denotes the number of packets at time $t$ in the queues in nodes $i, i+1, \ldots, j$ (inclusive) which need to cross the edge from node $i-1$ to node $i$. We assume that this quantity is measured at the end of time step $t$ (that is, at time step $t$, packets are inserted, then moved, after which the value of $A$ is determined). Let $\mathcal{A}$ be a deterministic $(w,\rho)$ adversary with $\rho = 1$. We next define an appropriate potential function $\phi(i,t) = \max\{\phi_1(i,t), w+n-1\}$ where $\phi_1(i,t) = \max\{A(i,k,t) + (i+n-1-k) : i \leq k \leq i+n-1\}$. The crux of the argument is the following lemma:

**Lemma 7** *Let $\mathcal{A}$ be a $(w,\rho)$ adversary with $\rho = 1$. Then for all $t \geq 0$, $\phi(i,t+w) \leq \phi(i,t)$.*

**Proof:** Clearly the only way $A(i,j,t)$ can increase is due to the insertion of packets and can only go up by one per insertion of a packet that needs to cross edge $(i-1,i)$. (Note that by assuming simple paths any packet entering the queue at node $i$ from the queue at $i-1$ will not need to traverse the edge $(i-1,i)$ again, and thus is not counted in $A(i,j,t)$.) Thus after the $w$ time steps $t+1, t+2, \ldots, t+w$, no $A(i,j,t)$ can increase by more than $w$ and hence $\phi(i,t)$ will increase by at most $w$ due to all the packet insertions during these steps.

If $\phi_1(i,t+\ell) \le n-1$ at any step $t+\ell$ with $0 \le \ell \le w-1$ then $\phi_1(i,t+w)$ can be at most $w+n-1$ and hence $\phi(i,t+w) = w+n-1 \le \phi(i,t)$. If $\phi_1(i,t+\ell) > n-1$ throughout these $w$ time steps then we argue that each routing step causes $\phi_1$ to decrease by one (offsetting any increase due to the injection step). To see this note that there must be at least one packet in the system at every time step by the definition of $\phi_1$. Let $k$ be any index which maximizes the quantity $B = \max\{A(i,k,t) + (i+n-1-k) : i \le k \le i+n-1\}$. If $k > i$, the queue for edge $(k,k+1)$ must be non-empty and contain a packet destined to cross edge $(i-1,i)$ or else the index $k-1$ would yield a larger value for $B$. If $k = i$, the queue for edge $(i,i+1)$ must be nonempty since $\phi_1(i,t+\ell) > n-1$. By the FTG protocol some such packet must traverse the edge $(k,k+1)$ during this routing step and hence the quantity $B$ must decrease. ▌

**Theorem 8** *Let $G$ denote the $n$-node cycle and $\Upsilon$ the class of rate 1 deterministic adversaries (say with window size $w$). Define $Q_0(G)$ to be the total number of packets initially in the network (i.e. the sum of all edge queue sizes) and let $Q_0 = \max\{Q_0(G), w\}$. Then $(G, \Upsilon, FTG)$ is stable and furthermore there are never more than $n(Q_0-1)+w$ packets in the system.*

**Proof:** Consider any time $t = mw$ for $m$ a non-negative integer. By Lemma 7, we have $\phi(i,mw) \le \phi(i,0) \le Q_0+n-1$. Hence at time $t = mw$, $A(i,i-1,mw) \le Q_0-1$. The total number of packets in the system at time $t = mw$ can then be upper bounded by $\sum_{i=0}^{n-1} A(i,i-1,mw)$, which will be no greater than $n(Q_0-1)$. Finally, for any time $t$ with $mw < t \le (m+1)w$, we have $\sum_{i=0}^{n-1} A(i,i-1,t) \le w + \sum_{i=0}^{n-1} A(i,i-1,mw)$. ▌

We now extend the above theorem to the case of a stochastic adversary with injection rate $1-\epsilon$.

**Theorem 9** *Let $G$ denote the $n$-node cycle and $\Upsilon$ the class of properly bounded stochastic adversaries with rate $1-\epsilon$ for some $\epsilon > 0$. Then $(G, \Upsilon, FTG)$ is stable.*

The proof here is analogous to the proof for deterministic adversaries. We need an analogue of Lemma 7 so as to apply Lemma 2.

**Lemma 10** *For all $i$, and for all $t \ge 0$, $E[\phi(i,t+w) | \phi(i,t) \ge n+2w] - \phi(i,t) \le -w\epsilon$.*

**Proof:** We argue as in the deterministic case, that as packets are inserted during steps $t+1, \ldots, t+w$, the potential increases by at most the number of inserted packets during these time steps which wish to cross the edge $i-1$ to $i$. By the definition of a $(w, 1-\epsilon)$ stochastic adversary, this implies that the expected increase in $\phi$ due to insertions during time steps $t+1, \ldots, t+w$ is bounded by $w(1-\epsilon)$. It remains to observe (as in the deterministic case) that subject to the condition that $\phi(i,t) \ge n+2w$, the decrease in $\phi$ during these steps due to packet routing is exactly $w$.

We combine these facts (about the expected increase due to insertions and the decrease due to routing steps) to obtain $E[\phi(i,t+w)|H_t] \leq \phi(i,t) + w(1-\epsilon) - w$ provided $\phi(i,t) \geq n + 2w$. ▍

Given Lemma 10 we can now apply Lemma 2 so as to complete the proof of the Theorem.

## 7 NTG

Two natural scheduling rules for packet routing are the Nearest-To-Go (NTG) and Furthest-To-Go (FTG) policies. Andrews *et al.* [2] prove that FTG is stable for any packet routing network at rate $\rho < 1$. Tsaparas [51] generalizes the Andrews *et al.* stability result for FTG to more general queuing networks by proving the universal stability of Most-Time-To-Go (MTTG) scheduling. (When all service times are identical as we assume in packet routing networks, then MTTG becomes FTG.)

In contrast to the stability of FTG, Andrews *et al.* show that there is a simple 6-node network such that NTG scheduling is unstable for a rate $\rho = .62$ path-packing adversary.

We extend this result in two ways. First we show for every $\rho > 0$ there is a queuing network for which NTG is unstable at rate $\rho$. Moreover, this instability will occur (for some initial configuration) even if packets are generated with constant rate $\rho$. Thus this instability result is not an adversarial result (except for the setting of the initial configuration) but was motivated by the adversarial approach. (Some experimental evidence suggests that this instability would still occur for Poisson arrivals and an empty initial configuration.)

We shall now describe the network G and only sketch the intuitive reason for instability. (The details of the instability proof can be found in Tsaparas.) Let $\rho > 0$ be given and without loss of generality assume that $1/\rho$ is an integer. We first construct a "toroidal queuing network" $\tilde{G}$ (where the servers are the nodes rather than the edges) and describe the instability result in terms of $\tilde{G}$. It is then easy to convert $\tilde{G}$ to a packet routing network $G$ for which the same instability phenomenon occurs.

Let $\tilde{G}$ be an $n \times n$ torus with $n\rho > 2$ and $n$ even. There will be $n$ classes of jobs. A class $i$ job initiates at node $(i, i+1)$ and follows the following path of (node) servers $(i,i+1), (i,i+2), \ldots, (i,i-1), (i+1,i), (i+2,i), \ldots, (i-1,i)$. (See Figure 1. All node addresses are computed mod $n$.)

For every $i$, a new class $i$ job will be generated every $1/\rho$ steps. For every node, exactly two job classes will pass through any given node so that the total induced load per step on any server is exactly $2\rho$.

For the NTG scheduling rule, jobs moving along a column have priority over jobs moving along a row. Thus class $i$ jobs when moving along column $i$ will intersect and have priority over all other job classes. The intuitive idea to achieve instability is to initially establish two "walls" of packets, say in column $i$ and column $j = i + n/2$. By a wall of packets on column $i$ we mean that every node $(k,i)$ with $k \neq i$ is occupied and in addition there may be a large number of class $i$ jobs queued in node $(i, i-1)$ [12] which will continue to keep column $i$ occupied. Such a wall will prevent other jobs from making progress. Eventually, the walls at columns $i$ and $j$ will drain out but at the same time a pair of walls will have been formed at columns $i+1$ and $i+1+n/2$. Moreover, we claim that the total number of jobs in the system has increased during this "phase" where the walls are

[12] To be more precise, we consider this queue of jobs at node $(i, i-1)$ to include other class $i$ packets which will reach $(i, i-1)$ during the present "phase" when a wall exists at column $i$.

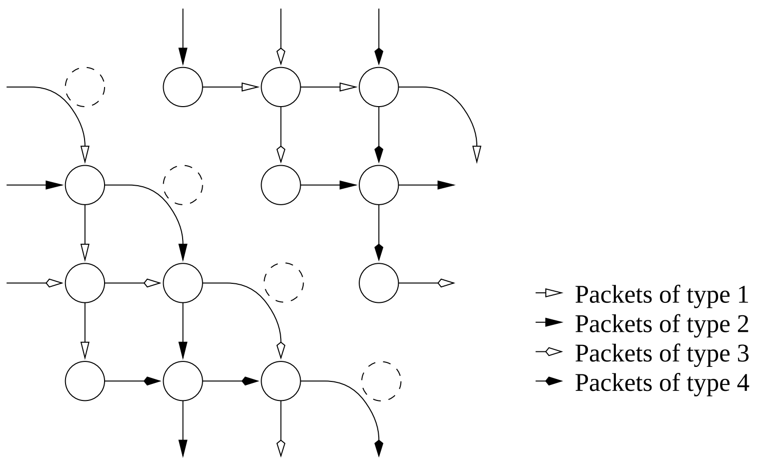

Figure 1: Bad network for NTG scheduling

moving over one column. To see this suppose that there are $M$ class $i$ jobs in the wall at the start of the phase (and the same for class $j = i + n/2$). Now assume that there is a total of $T$ jobs in the system at the start of the phase. The phase lasts $M$ steps during which time exactly $nM\rho$ new jobs have entered the system and $2M$ jobs have left the system. Thus at the end of the phase (and the start of the next phase) there are $T' = T + nM\rho - 2M > T$ jobs in the system since $n\rho > 2$.

Tsaparas [51] provides a careful proof of this intuitive idea. In particular, his proof requires an initial configuration which is very "symmetric" and then shows that this symmetry is preserved throughout the process so that new walls are formed without any breaks occurring in the walls.

Finally we need to indicate how to convert $\tilde{G}$ to a packet routing network $G$. For every node $\kappa = (i, j)$ in $\tilde{G}$, we have two nodes $u_\kappa$ and $v_\kappa$ in $G$. The edge set of $G$ consists of a "server" edge $e_\kappa = (u_\kappa, v_\kappa)$ for every node $\kappa$ in $\tilde{G}$ and a "connecting edge" $f_{(\kappa,\kappa')} = (v_\kappa, u_{\kappa'})$ for every pair of nodes $\kappa, \kappa'$ in $\tilde{G}$ for which some job traverses from $\kappa$ to $\kappa'$ in the queuing network; i.e., $\kappa = (i, j)$ and $\kappa' = (i', j')$ and either $i' = i + 1$ and $j' = j$ or $i' = i$ and $j' = j + 1$. In the obvious way, each job class in the queuing network will determine a packet class (and its path) in the packet routing network. We claim that for $n > 3\rho$ that the packet routing network system $(G, NTG, \mathcal{A})$ is unstable where $\mathcal{A}$ is an "adversary" which is generating packets in each class at constant rate $\rho$. Again the proof of this claim can be found in [51].

## 8 Conclusion and open problems

We have introduced a new approach for the study of queuing networks. Although our motivation comes from continuous packet routing, we note that the adversarial approach can be applied more generally to the wider field of queuing theory (as is done in Tsaparas [51]). Clearly more general queuing networks (in which jobs can revisit the same server many times and/or where service time distributions are class dependent) offer additional challenges. It remains an interesting open question to find some set of general conditions (depending, for example, on the rate, queuing discipline, and underlying network) that are sufficient to guarantee stability. Further work is also needed to better bound queue sizes and packet delays, and to understand specific networks such as arrays and hypercubes. We mention a few of the many open problems:

- For packet routing, can FIFO be made unstable for arbitrarily small positive rates of injection in the adversarial model? More generally, does stability at some rate $\rho_1 > 0$ imply stability for all $\rho < 1$ (for a "natural" scheduling policy)?

- Considering the ring as a queuing network allowing arbitrary (i.e., reversing) paths, is every scheduling policy stable?

- Is NTG unstable for a Poisson input model for packet routing and for arbitrarily small rates? In particular, does the instability result of Section 7 extend from constant rate arrivals to Poisson arrivals? More generally, when does adversarial instability from some given initial configuration imply instability with Poisson arrivals (and say an empty initial configuration). Note that while FIFO is unstable in an adversarial setting (a special case of class independent service times), it is stable for Poisson arrivals and class independent service times.

- Is there a generic transformation of stability results from deterministic to stochastic adversaries?

- Our queue size bound for DAGs is exponential in the length $d$ of the longest path in the network. Indeed Andrews *et al.* [2] show that NIS and FTG can be made to suffer such exponential size queues on DAGs. What is the best bound on queue size in DAGs that holds for the FIFO scheduling rule? Recently Adler and Rosén (personal communication) show that LIS has polynomial size queues (as a function of $d$) on any DAG but in contrast Andrews and Zhang [4] show that LIS can have exponential size queues (in $d$) queues for certain networks. But these networks have size exponential in $d$ and hence it remains open if the queue size for LIS can be polynomially bounded as a function of the size of the network. Andrews *et al.* construct a randomized ("LIS based") scheduling rule which has (with high probability) polynomially-sized queues and note that this randomized policy can be converted to a centralized deterministic scheduling policy with polynomially-sized queues. The obvious question is whether there exists a (natural) decentralized, deterministic scheduling rule that has polynomially-sized (as a function of either $d$ or the size of the network) queues for all networks.

- What can be said about stability of non-adaptive routing in a network where each edge $e$ is capable of simultaneously transmitting some number $n_e$ packets in one step? Specifically, if $(G, \mathcal{A}, \mathcal{S})$ is universally stable for any $(w, \rho)$ adversary $\mathcal{A}$, does it follow that $(G', \mathcal{A}', \mathcal{S})$ is also universally stable for any $(w, \rho)$ adversary $\mathcal{A}$' where $G$ is a graph where all edges have bandwidth 1 and $G'$ is the same graph with arbitrary edge bandwidths $\{n_e\}$. Here the load condition is modified in the obvious way so that in any window $\tau$ of $w$ time steps and for any edge $e$, the adversary injects at most $N(\tau, e) \leq \rho \cdot w \cdot n_e$ packets which cross edge $e$. (This can also be viewed as a very special case of adaptive routing by viewing a "high bandwidth" edge as a set of edges.) More generally, what "network transformations" preserve stability? Recent results along these lines have been derived by Borodin, Ostrovsky and Rabani [10].

- Which (natural or efficient) scheduling rules are universally stable for deterministic rate 1 adversaries? Subsequent to asking this question, David Gamarnik [24] has proven that the "nearest to origin" NTO scheduling rule (which gives priority to packets that have traversed the fewest edges thus far) is universally stable for deterministic rate 1 adversaries. It is easy to adapt Gamarnik's proof to show that FTG is also universally stable for deterministic rate 1 adversaries. These proofs yield an exponential bound on the total number of packets in the network. Gamarnik also observes that using either NTO or FTG, a rate 1 adversary can prevent some packets from ever reaching their destination. He asks if there is a scheduling rule (perhaps a modification of NTO or FTG) that can guarantee bounded or at least finite delivery time with respect to deterministic rate 1 adversaries.

**Acknowledgments**

We thank Nabil Kahale, Tom Leighton, Baruch Schieber, and Ken Sevcik for valuable discussions on queuing theory and routing. We are also indebted to the referees for their very constructive suggestions.

## References

[1] W. Aiello, E. Kushilevitz, R. Ostrovsky, A. Rosén Adaptive Packet Routing for Bursty Adversarial Traffic. In *Proc. of the 30th Ann. ACM Symposium on the Theory of Computing* (STOC), 359-368, 1998.

[2] M. Andrews, B. Awerbuch, A. Fernández, J. Kleinberg, F.T. Leighton, Z. Liu. Universal Stability Results for Greedy Contention-Resolution Protocols. *Proceedings of the Thirty-Seventh Annual IEEE Symposium on Foundations of Computer Science*, 380–389, 1996.

[3] M. Andrews Instability of FIFO in Session Oriented Networks. *To appear in Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*, 2000.

[4] M. Andrews and L. Zhang The Effects of Temporary Sessions on Network Performance. *To appear in Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*, 2000.

[5] B. Awerbuch and F. T. Leighton. Improved approximations for the multicommodity flow problem and local competitive routing in networks. *Proceedings of the Twenty–Sixth Annual ACM Symposium on Theory of Computing*, 487-496, 1994.

[6] M. Harchol-Balter and P. E. Black. Queuing analysis of oblivious packet-routing algorithms. *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1994.

[7] M. Harchol-Balter and D. Wolfe. Bounding delays in packet-routing networks. *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing*, 248-257, 1995.

[8] D. Bertsimas, D. Gamarnik and J. N. Tsitsiklis. Stability conditions for multiclass fluid queuing networks. *IEEE Transactions on Automatic Control*, vol 41, Nov 1996, 1618–1631.

[9] A. Borodin, J. Kleinberg, P. Raghavan, M. Sudan, and D. Williamson. Adversarial Queuing Theory. *Proceedings of the Twenty–Eighth Annual ACM Symposium on Theory of Computing*, 376-385, 1996.

[10] A. Borodin, R. Ostrovsky, Y. Rabani. Routing Networks with Speeds and Capacities. *In preparation*, November, 1999.

[11] A. Broder, A. Frieze, E. Upfal. A General Approach to Dynamic Packet Routing with Bounded Buffers. *Proceedings of the Thirty-Eighth Annual IEEE Symposium on Foundations of Computer Science*, 390-399, 1997.

[12] M. Bramson. Instability of FIFO Queuing Networks. *The Annals of Applied Probability*, vol 4, 1994, 414-431.

[13] M. Bramson. Instability of FIFO Queuing Networks with Quick Service Times. *The Annals of Applied Probability*, vol 4, 1994, 693-718.

[14] M. Bramson. Convergence to equilibria for fluid models of FIFO queuing networks. *Queuing Systems*, vol 22, 1996, 5-45.

[15] K. Mani Chandy, J. Howard and D. Towsley. Product form and local balance in queuing networks. *Journal of the ACM*, April 1977, 250-263.

[16] R. L. Cruz. A Calculus for Network Delay, Part I. *IEEE Transactions on Information Theory*, vol 37, January, 1991, 114-131

[17] R. L. Cruz. A Calculus for Network Delay, Part II. *IEEE Transactions on Information Theory*, vol 37, January 1991, 132-141

[18] J. G. Dai. On Positive Harris Recurrence of Multiclass Queuing Networks: a Unified Approach via Fluid Limit Models. *The Annals of Applied Probability* vol 5, 1995, 49-77.

[19] J. G. Dai and G. Weiss. Stability and Instability of Fluid Models for Reentrant Lines. *Mathematics on Operations Research*, vol 21, Feb 1996.

[20] J. G. Dai and S. P. Meyn. Stability and Convergence of Moments for Multiclass Queuing Networks via Fluid Limit Models. *IEEE Transactions on Automatic Control*, vol 40, Nov 1995, 1889-1904.

[21] D. D. Down and S. P. Meyn. Stability of acyclic multiclass queuing networks. *IEEE Transactions on Automatic Control*, vol 40, no 5, 1995, 916-920.

[22] R. Durrett. Probability: theory and examples, 2nd edition. Duxbury Press, Belmont, CA.

[23] D. Gamarnik. Stability of Adversarial Queues via Fluid Models. *Proceedings of the 39th Symposium on the Foundations of Computer Science*, 60-70, 1998.

[24] D. Gamarnik. Stability of Adaptive and Non-Adaptive packet Routing Policies in Adversarial Queuing Networks. *Proceedings of the Thirty–First Annual ACM Symposium on Theory of Computing*, 206-214, 1999.

[25] A. Goel. Stability of Networks and Protocols in the Adversarial Queuing Model for Packet Routing. Stanford University Technical Note STAN-CS-TN-97-59, June 1997.

[26] B. Hajek. 'Hitting-time and occupation-time bounds implied by drift analysis with applications. *Advances in Applied Probability*, vol 14, 1982, 502-525.

[27] J. R. Jackson. Jobshop-like queuing systems. *Management Science*, vol. 10, 1963, 131-142.

[28] L. Kleinrock. Queuing systems. Wiley, New York, 1975.

[29] F. P. Kelly. Reversibility and Stochastic Networks. Wiley, New York, 1979.

[30] N. Kahale and F.T. Leighton. Greedy dynamic routing on arrays. *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1995.

[31] F.T. Leighton. Average case analysis of greedy routing algorithms on arrays. *Proceeding of the Second Annual ACM Symposium on Parallel Algorithms and Architectures*, 1990.

[32] F.T. Leighton, B.M. Maggs, S.B. Rao. Packet routing and job-shop scheduling in O(congestion+dilation) steps. *Combinatorica*, 14(1994), 167-180.

[33] F. T. Leighton, B. M. Maggs and A. W. Richa. Fast algorithms for finding O(congestion+dilation) packet routing schedules. *Combinatorica*, to appear.

[34] M. Loève. *Probability Theory*, volume 2. Springer-Verlag, New York, 1978.

[35] S. H. Lu and P. R. Kumar. Distributed scheduling based on due dates and buffer priorities. *IEEE Trans. Automatic Control*, vol 36, no. 12, 1991, 1406-1416.

[36] Y. Mansour, B. Patt-Shamir. Greedy packet scheduling on shortest paths. *Proceedings of the Tenth Annual ACM Symposium on Principles of Distributed Computing*, 1991.

[37] S. P. Meyn and D. Down Stability of generalized Jackson networks *Annals of Applied Probability*, vol 4, 1994, 124-148.

[38] D. Mitra and R. Cieslak. Randomized parallel communications. *Proceedings of the International Conference on Parallel Processing*, 1986.

[39] M. Mitzenmacher. Bounds on the Greedy Routing Algorithm for Array Networks. *Proceedings of the Sixth Annual ACM Symposium on Parallel Algorithms and Architectures*, 1994.

[40] R. Ostrovsky and Y. Rabani. Universal $O(\text{congestion} + \text{dilation} + \log^{1+\epsilon} N)$ Local Control Packet Switching Algorithms. *Proceedings of the Twenty–Ninth Annual ACM Symposium on Theory of Computing*, 644-653, 1997.

[41] A. K. Parekh and R. G. Gallager. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case. *IEEE/ACM Transactions on Networking*, vol 1(3), 344-357, 1993.

[42] A. K. Parekh and R. G. Gallager. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Multiple-Node Case. *IEEE/ACM Transactions on Networking*, vol 2(2), 137-150, 1994.

[43] R. Pemantle and J. S. Rosenthal. Moment Conditions for a Sequence with Negative Drift to be Uniformly Bounded in $L^r$. Technical Report 9814, Department of Statistics, University of Toronto, August 1998.

[44] R. Pemantle and J. S. Rosenthal. Moment Conditions for a Sequence with Negative Drift to be Uniformly Bounded in $L^r$. Stochastic Processes and Their Applications, vol 82, 1999, 143-155.

[45] J. R. Perkins and P. R. Kumar. Stable distributed real-time scheduling of flexible manufacturing/assembly/disassembly systems. *IEEE Trans. on Automatic Control*, vol 34, 1989, 139-148.

[46] Y. Rabani and E. Tardos. Distributed Switching in Arbitrary Networks. *Proceedings of the Twenty–Eighth Annual ACM Symposium on Theory of Computing*, 366-375, 1996.

[47] A. N. Rybko and A. L. Stolyar. Ergodicity of stochastic processes describing the operation of open queuing networks. *Problems of Information Transmission*, vol 28, 1992, 199-220.

[48] T. I. Seidman. 'First come, first serve' can be unstable. *IEEE Trans. Automatic Control*, vol 39, 1994, 2166-2171

[49] G. D. Stamoulis and J. N. Tsitsiklis. The efficiency of greedy routing in hypercubes and butterflies. *Proceedings of the Third Annual ACM Symposium on Parallel Algorithms and Architectures*, 1991.

[50] L. Tassiulas and L. Georgiadis. Any work-conserving policy stabilizes the ring with spatial re-use. *IEEE/ACN Trans. on Networking*, vol. 4, no. 2, 1996, 205-208.

[51] P. Tsaparas. Stability in Adversarial Queuing Theory *M.Sc Thesis, Department of Computer Science, University of Toronto*, 1997.

[52] J. Walrand. An introduction to queuing networks. Prentiss Hall, Englewood Cliffs, New Jersey, 1988.