

Adversarial Queueing Theory

Allan Borodin*

Jon Kleinberg[†]

Prabhakar Raghavan[‡]

Madhu Sudan[§]

David P. Williamson[§]

1 Introduction

We introduce a new approach to the study of dynamic (or continuous) packet routing, where packets are being continuously injected into a network. Our objective is to study what happens to packet routing under continuous injection as a function of network load, for various queueing policies. Our approach is based on the adversarial generation of packets, so that the results are more robust in that they do not hinge upon particular probabilistic assumptions.

In suggesting a new approach to studying a classical phenomenon, it is important to give careful consideration to all the relevant previous work in packet routing, queueing theory and probabilistic analysis. We give a more detailed account of previous work in Appendix A, to permit comparison with our work. Here we summarize the salient features of prior work in order to motivate our model. Most prior work on packet routing has been in the *static model* in which there is a fixed initial set of packet routing requests; when these packets are delivered, the problem is considered solved and the analysis stops there. Static packet routing is a basic problem in the context of parallel computation models, but for the setting of communications networks it is essential to study the case of continuous injection of packets. While it is possible to try modelling such continuous problems statically, by delaying the entry of packets using synchronization barriers, a much more natural approach is to analyze standard (local-control) routing algorithms in this fully dynamic setting. Nearly all previous work in this regard has used probabilistic models for the generation (and sometimes, delivery) of packets. Such work can broadly be classified into:

1. Queueing-theoretic approaches, where packets are generated by a Poisson injection process; frequently, each packet is assumed to have a random destination. A common assumption in queueing theory is that the time for a packet to pass through a server (i.e., an edge) is exponentially distributed whereas for packet routing this time is a constant. This apparently slight difference poses a world of subtle difficulties in adapting queueing theory to continuous packet routing.
2. Probabilistic analyses in which a Poisson or Bernoulli process generates packets bound for random destinations; the passage time through an edge is a constant as in packet routing (see [9], for example). The resulting analyses are technically difficult, and yield results that are specific to particular networks and queueing disciplines; moreover, as in queueing-theoretic approaches, they rely heavily on the underlying probabilistic assumptions which determine the injection process.

We present our new model in the following section. In contrast to the two classes above, we concentrate on constant service times and adversarial packet generation and choice of destination. Section 2 summarizes our results. We see that even in such a worst-case setting we are able to obtain strong results for a variety of networks and queueing policies.

1.1 The New Model

A *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. 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 set of paths, each 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

*Department of Computer Science, University of Toronto, Toronto, Canada M5S 1A4. Part of this work was performed while visiting the IBM T.J. Watson Research Center.

[†]Laboratory for Computer Science, MIT, Cambridge, MA 02139. Supported by an ONR Graduate Fellowship. Part of this work was performed while visiting the IBM T.J. Watson Research Center.

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

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

by each packet is fixed at the time of injection, so as to be able to focus on the queueing aspects of the problem.

Clearly an unrestricted adversary can flood the network with packets, demanding more bandwidth than available. One approach is to restrict the set of paths generated by the adversary at each step to be a *path packing* — the paths requested at a step must be edge-disjoint. A more stringent restriction might stipulate that at most one packet be injected at each step. In general, we have a collection of *allowed requests*; at each step, the adversary must pick one request (= set of paths) from this collection. In this paper, we will concentrate on the cases of path packings and single path requests.

Our model permits an adversary that can demand network bandwidth up to a prescribed *load factor*, or *injection rate*. For $\epsilon > 0$, we say that an adversary as above *injects at rate* $1 - \epsilon$ if it generates requests subject to the following additional restriction: during any r consecutive steps, the adversary can make at most $\lceil r(1 - \epsilon) \rceil$ requests. As a special case, we say that an adversary injects at rate 1 if the adversary makes one request per step. For path packing requests, our notion of injection rate is an intuitive measure of the fraction of network bandwidth demanded by the adversary.

We also consider *stochastic adversaries*. A stochastic adversary is also specified by a *request collection* and a *rate*; now, however, the adversary has a probability distribution over the power set of its request collection, and at each step draws a *set* of requests from this distribution. We say that a stochastic adversary injects at rate $1 - \epsilon$ if the expected number of requests injected at any step is at most $1 - \epsilon$. We also define the *variance* of the adversary to be the variance of the number of requests injected per step. Note that this model is very general; as a special case it clearly subsumes the case of a fixed input distribution at each network node, and it allows the adversary to adaptively modify the distribution at each time step. To differentiate stochastic adversaries from those in the previous paragraph, we will call the adversaries defined above *deterministic* adversaries.

A *queueing policy* specifies, for each edge e and each time step, which packet (amongst those waiting) is to be moved along edge e . A *greedy* policy 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 policies Q .

Our main goal will be to prove the *stability* of a given queueing policy on various networks and against various adversaries. We will say that a given queueing policy is *stable*, on a fixed network, against a fixed deterministic adversary, if there is a constant C (which may depend on the network size) such that the number of packets in the system at all times is bounded by C . We say that the policy is stable against a stochastic adversary if the probability of having more than k packets in the system goes to zero as k goes to infinity.

2 Results

It is tempting to conjecture that for every graph G , every greedy queueing discipline is stable for a deterministic adversary under injection rate $\rho < 1$ (or even $\rho = 1$). Our first result shows this does hold for a large class of graphs.

(1) *If G is a tree or a DAG, then every greedy queueing discipline is stable for every deterministic adversary with rate at most 1 on path packings.*

Unfortunately this result does not extend to the case of graphs with directed cycles, as our next result shows.

(2) *Let G be a directed cycle on $n \geq 3$ nodes. Then for the greedy queueing disciplines FIFO and longest-in-system (LIS), there exist deterministic adversaries with rate 1 on single paths for which these policies are unstable.*

Interestingly, the result is not universal over all greedy queueing disciplines, as indicated by the next result.

(3) *If G is a directed cycle, then for every adversary with injection rate at most 1 on path packings, the Furthest-to-Go (FTG) policy is stable.*

For the LIS discipline, which fails at rate 1, we identify the exact threshold at which it is stable by showing

(4) *If G is a directed cycle, then for every $\epsilon > 0$ and every adversary with injection rate at most $1 - \epsilon$ on path packings, the LIS policy is stable.*

We also show that any greedy discipline is stable on the directed cycle for a smaller injection rate of single paths.

(5) *Let \mathcal{P} be an arbitrary greedy queueing discipline, and G the directed cycle. Then \mathcal{P} is stable against any adversary with injection rate at most $\frac{1}{2} - \epsilon$ on single paths.*

Given this sensitivity to both the queueing discipline and the underlying network our next focus is whether there exists a queueing discipline which is stable for all networks. We show that LIS has this property, against an adversary that can inject single *shortest paths*.

(6) *Let G be an arbitrary directed graph and $\epsilon > 0$. Then LIS is stable against any deterministic adversary with rate $1 - \epsilon$ whose request set consists of single shortest paths.*

All our results for injection rates bounded away from 1 also hold for stochastic adversaries of bounded variance. In particular, results (4), (5), and (6) extend as they are, and results (1) and (3) extend for the case of injection rate at most $1 - \epsilon$.

The results above show that for stochastic adversaries of bounded variance, a number of general statements can be made based on rate alone. Indeed, it becomes natural to consider whether some form of the following question might hold in the affirmative: Is every greedy queueing discipline on every graph G stable against every stochastic adversary of bounded variance injecting path packings at rate $1 - \epsilon$? Awerbuch and Leighton [2] have recently reported a negative resolution to this question, at this level of generality; see Section 6 for more details.

In addition to providing a good technique for obtaining stability results, the adversarial model (at least in the deter-

ministic case) is able to distinguish between different queueing disciplines. This sensitivity to the queueing discipline is not unnatural, yet prior queueing theoretic results do not seem to highlight this. Perhaps this is a consequence of the “Poisson arrival and exponential service time” models, which intuitively tend to make packets within queues indistinguishable from each other. A different reason explaining the sensitivity of our model to the queueing discipline may be the fact that it is the “determinism” of the adversary which allows for it.

While we have restricted ourselves to constant service times throughout this investigation, it is clear that the adversarial injection model can be applied to the case of queueing networks with servers having general service time distribution. It seems plausible that many of the results presented here could be extended to such models.

The remainder of the paper is structured as follows. Section 3 discusses our results for the ring. Sections 4 and 5 discuss our results for trees, DAGs, meshes, and general networks. We conclude in Section 6. A brief overview of previous work can be found in Appendix A.

3 The Ring

Throughout this section, our underlying graph G will be the n -node cycle, with vertices numbered $0, \dots, n-1$. As noted above and in Appendix A, even this case is quite non-trivial, both from the point of view of classical queueing theory and within our setting. In the first part of this section, we will consider arbitrary greedy protocols, and prove result (5) and its stochastic counterpart. We next show that the ring is not stable under all greedy disciplines at rate 1 (result (2)). We then show the stability of two specific queueing disciplines, LIS and FTG, by proving results (3) and (4) at the end of the section.

3.1 Arbitrary Greedy Protocols

First, we consider a deterministic adversary.

Theorem 1 *Let G denote the n -node cycle, \mathcal{P} an arbitrary greedy protocol, and \mathcal{A} an arbitrary deterministic adversary that injects single paths at rate $\frac{1}{2} - \varepsilon$, for some $\varepsilon > 0$. Then \mathcal{P} is stable, and there are never more than $O(\varepsilon^{-1}(n + \varepsilon^{-1}))$ packets in the system.*

Proof: We proceed by defining an appropriate potential function and showing that it decreases once the number of packets in the system is large enough. Since the adversary injects at rate $\frac{1}{2} - \varepsilon$, we can fix an r so that $\frac{r}{2r+1} \geq \frac{1}{2} - \varepsilon$ and conclude that no more than r packets will be injected in each consecutive interval of $2r + 1$ time steps.

We define the length ℓ_v of a node v to be 1 if it is empty, and otherwise the number of packets it contains. To define our potential function, we need to impose an artificial “order” on the packets in each queue; let us do this using some arbitrary lexicographic order on the packets. Given this, we

define the weight w_p of a packet p to be the sum of the lengths of all nodes lying strictly between it and its destination, plus the number of packets ahead of it in the order at its current node. Finally, for technical reasons, we also add to each packet weight an initial “overhead” of $2r + 1$, which it “claims” when it is absorbed. Our potential Φ is the sum of all weights of all packets.

We can picture an interval of $2r + 1$ consecutive time steps composed of a sequence of single time steps, as follows.

- (1) \mathcal{A} may inject 0 or more packets.
- (2) Packets move for one step according to the protocol \mathcal{P} and are absorbed (if they reach their destination).

It will turn out to make the analysis easier if we break step (2) into the following two parts — it is clear that the step-by-step behavior of the system is left unchanged:

- (2a) \mathcal{A} re-arranges the order of each queue arbitrarily.
- (2b) Packets move according to the FIFO discipline (the packet at the head of each queue leaves and enters the back of the queue at which it arrives); packets reaching their destination are absorbed.

Let I be an interval of $2r + 1$ consecutive time steps. We now analyze the net change in Φ over steps of type (1), (2a), and (2b) in the interval I . Suppose there are currently k packets in the system.

During I , at most r packets are injected in steps of the form (1). The weight of each is at most $(k + r + n) + (2r + 1)$, the latter term coming from our definition of “overhead.” Also, the injection of p can increase the weight of every other packet by at most 1 (including others that were just injected). Thus Φ increases by at most

$$r(k + 3r + n + 1) + r(k + r) = 2rk + rn + 4r^2 + r. \quad (1)$$

during I due to steps of type (1).

Now consider steps of type (2a). We claim that Φ doesn’t change here. Namely, consider a given queue, and imagine that the adversary re-arranges it by a sequence of consecutive transpositions. Each transposition increase the weight of one packet by exactly one, and decreases the weight of another by exactly one; so there is no change.

In each step of type (2b), we claim that the weight of each unabsorbed packet decreases by at least 1. Consider an unabsorbed packet x . The weight of x is equal to the number of packets between x and its destination $dest(x)$, plus the number of empty queues between x and $dest(x)$. Let Q denote the queue just in front of $dest(x)$. Now the set of empty queues between x and $dest(x)$ can be partitioned into contiguous intervals, and on the movement phase, each interval shrinks at its back (as the full queue bordering it sends a packet forward) and may grow at its front (if the full queue bordering it in front has only one packet). Thus, the number of empty queues does not increase. Similarly, the number of packets between x and $dest(x)$ does not increase, so we just have to argue that one of these two terms decreases. Consider the following two cases.

(a) Q is non-empty. Then a packet in Q will pass across $dest(x)$, and decrease the number of packets between x and $dest(x)$.

(b) Q is empty. Then the interval of empty queues between x and $dest(x)$ that contains Q diminishes by 1, and so the number of empty queues between x and $dest(x)$ diminishes by 1.

Note that absorption can't increase the length of a node and so does not increase Φ .

Thus, we have shown that in steps in I of type (2b), any packet that moves the full $2r + 1$ times will decrease its weight by $2r + 1$. But any packet that does not move the full $2r + 1$ times must have been absorbed during this time step, so its weight goes down by $2r + 1$ as well. Thus Φ goes down by at least $k(2r + 1) = 2rk + k$. Adding this to equation (1), we see that the potential drops by at least

$$k - rn - 4r^2 - r. \quad (2)$$

This will be positive once k is $\Omega(\varepsilon^{-1}(n + \varepsilon^{-1}))$; and it is not hard to argue from Equation (2) that k will never exceed $O(n\varepsilon^{-1} + \varepsilon^{-2})$. \square

We note that this result (as well as our other results for deterministic adversaries of rate $\rho < 1$) hold in the following stronger model: for a parameter T , the adversary can make at most $\lfloor \rho T \rfloor$ requests over any interval of T steps. The proof above carries through directly, using T in place of the number $2r + 1$; the resulting bound on the number of packets in the system is $O(\varepsilon^{-1}(n + T))$.

It is now remarkably easy to obtain essentially the same bound for a stochastic adversary — recall the class of such adversaries includes, as very basic special cases, the standard probabilistic models from queueing theory.

Theorem 2 *Let G denote the n -node cycle, \mathcal{P} an arbitrary greedy protocol, and \mathcal{S} an arbitrary stochastic adversary that injects paths at rate $\frac{1}{2} - \varepsilon$ and variance bounded by θ . Then \mathcal{P} is stable against \mathcal{S} .*

Proof: As in the previous proof, we break a single time step into steps (1), (2a), and (2b). Now, since we are in the stochastic model, the adversary \mathcal{A} performs a single “injection” in step (1), and packets then move only once.

We use the same potential function Φ as before, except that we do not add an “overhead” term to the weight of each packet. Consider the expected change in Φ in a single time step; suppose there are currently k packets in the system.

In step (1), suppose that X packets are injected, where X is a random variable with expectation at most $\frac{1}{2} - \varepsilon$ and variance at most θ . By analogy with the previous proof, the expected increase in potential is at most

$$\begin{aligned} E[X(k + X) + X(k + n + X)] \\ &\leq EX(2k + n) + 2EX^2 \\ &\leq (1 - 2\varepsilon)k + \left(\frac{1}{2} - \varepsilon\right)n + 2\theta + 2. \end{aligned}$$

Now, again there is no change to Φ in step (2a), and it goes down by at least $k - n$ in step (2b). Thus, the expected net change to Φ is

$$-2\varepsilon k + \left(\frac{3}{2} - \varepsilon\right)n + 2\theta + 2. \quad (3)$$

Let Φ_t denote the value of the potential after t steps, and $\beta = 4\left(\frac{n}{\varepsilon} + \frac{\theta}{\varepsilon}\right)\left(n + \frac{n}{\varepsilon} + \frac{\theta}{\varepsilon}\right)$. Then if $\Phi_t \geq \beta$, we can conclude that there are at least $2(n + \theta)\varepsilon^{-1}$ packets in the system at time t , and hence (assuming $n \geq 3$) by (3) that

$$E[\Phi_{t+1} | \Phi_t] \leq \Phi_t - \frac{11}{2}. \quad (4)$$

Equation (4) now allows the use of Martingale inequalities (given a bound on the variance of the edge capacity of injections), to yield a result indicating that the probability that the number of packets in the system is equal to $2(n + \theta)\varepsilon^{-1} + j$ goes down exponentially with j . In particular, it implies the stability of \mathcal{P} against the adversary \mathcal{S} . \square

While it is true that a very general class of potential-function arguments for deterministic adversaries yield analogous results for stochastic adversaries, we do not attempt to formalize this principle at the present time.

3.2 Some Instability Thresholds

Clearly every protocol is unstable against a deterministic adversary that is allowed to inject single paths at rate $1 + \varepsilon$, for any $\varepsilon > 0$. Here we consider the following question: what can be said about the stability of greedy protocols against deterministic adversaries that inject paths at rate exactly 1?

The answer, it turns out, depends on the protocol and on the underlying network. We show in Section 3.4 that the Furthest to Go protocol (priority to the packet with the greatest distance to its destination) is in fact stable at injection rate 1 for path packings on the ring, and in Section 5 we show that any greedy protocol is stable at injection rate 1 for path packings on any directed acyclic graph. Here, on the other hand, we exhibit 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 3 *There is a deterministic adversary \mathcal{A} that injects single paths at rate 1, such that LIS on the ring is unstable against \mathcal{A} . There is a deterministic adversary \mathcal{A}' that injects single paths at rate 1, such that FIFO on the ring is unstable against \mathcal{A}' .*

Proof: We consider only the LIS case here; the construction for FIFO is very similar.

We describe the adversary \mathcal{A} . 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. (In particular, our self loops will move from i to

$i + 1$ to $i + 2$, not the other way.) It is not difficult to refine this argument so that the adversary injects shorter paths.

\mathcal{A} works as follows:

- For $k = 1, 2, 3, \dots$
 - 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 that at the end of iteration k of this process, there will be one packet at node 1 and $kn - 1$ packets queued at node 0. Thus the number of packets becomes unbounded. \square

3.3 The Longest-in-System Protocol

In the next two parts, we obtain tight bounds for two natural greedy protocols, namely LIS and FTG (defined in the previous section). Here we prove a result for LIS that strengthens the bound of Theorem 1: we allow an adversary of rate $1 - \varepsilon$, and we allow its request set to consist of path packings, as opposed to single paths. Given Theorem 3, this result is essentially tight.

Theorem 4 *Let G denote the n -node cycle and \mathcal{A} an arbitrary deterministic adversary that injects path packings at rate $1 - \varepsilon$, for some $\varepsilon > 0$. Then LIS is stable against \mathcal{A} , and there are never more than $O(\varepsilon^{-1}(n^2 + \varepsilon^{-1}))$ packets in the system.*

Proof: As usual, we choose r large enough so that $\frac{r}{r+1} \geq 1 - \varepsilon$; we focus on an interval I of $r + 1$ consecutive time steps; and a single time step will consist of the following two steps.

- (1) \mathcal{A} may inject 0 or more path packings.
- (2) Packets move according to LIS and are absorbed.

To prove stability, we first modify the potential function as follows. We first assign a color to each packet in the system, so that two packets receive the same color if and only if they were injected by \mathcal{A} as part of the same path packing. Thus, a single ‘‘color class’’ consists of all the packets in one path packing.

If p is a packet, then we define w_p to be the total time that would be required for p to be absorbed under LIS if there were no future injections; we add an overhead of $r + 1$ to w_p that is reclaimed on absorption.

If γ is a color class, we define

$$c_\gamma = \max_{p \in \gamma} w_p.$$

Now our potential function is defined as $\Phi = \sum_\gamma c_\gamma$. So our potential function is a sum of the maximum time till absorption (plus the overhead term) in each color class. One point to note is the following: if p is a packet that has already been absorbed, then $w_p = 0$; and if γ is a color class all of whose

packets have been absorbed then $c_\gamma = 0$. We say that a color class γ is *active* if $c_\gamma > 0$.

Suppose there are currently m active colors in the system. Then the decrease of Φ over steps in I of type (2) is $r + 1$ per active color, for a total decrease of at least

$$mr + m.$$

The difficulty is in showing that the increase in Φ in steps of type (1) can be bounded by a function of m , and not just by a function of the total number of packets in the system (which may be much larger).

For this, we need to prove the following strengthening of a result of Mansour and Patt–Shamir [10] in the case of the ring. The result of [10] states that under any greedy packet–routing discipline, in any network, the total time for a packet to reach its destination is bounded by the distance to its destination plus the total number of packets in the system. Here we show

Claim 1 *Let G , as above, denote the n -node ring. Let p be a packet at distance d from its destination. If there are no future absorptions, then the time until absorption of p is at most $d + m - 1$, where m denotes the number of active color classes.*

Proof: Our proof is based on a construction from [10]; we sketch it briefly in the next paragraph and refer the reader to this paper for more extensive details.

[10] defines a *time path* to be a function τ mapping time steps to packets, in such a way that if $\tau(t) \neq \tau(t + 1)$, then the two packets $\tau(t)$ and $\tau(t + 1)$ reside at the same node at time $t + 1$. We say that the function τ is ‘‘following’’ the packet $\tau(t)$ at time t . A time path is said to be *blocked* by a packet q at time step t if at time t , q traverses the edge for which $\tau(t)$ is waiting. Now suppose a time step t' satisfies the property that a packet q blocks τ at time t' , and q meets τ at some time $t'' > t'$. Moreover, suppose that t' is minimal with this property. Then we define $\text{Switch}(\tau)$ to be the time path obtained by following $\tau(t)$ unless $t' \leq t < t''$, in which case it follows the packet q ; that is, $\text{Switch}(\tau)(t) = q$ for $t' \leq t < t''$ and is equal to $\tau(t)$ otherwise. Consider taking an initial time path which follows a packet p from start to destination, and then repeatedly invoking the Switch operation as long as it is applicable. [10] shows that such a sequence of Switch es must terminate; when it does, the resulting time path is never blocked by the same packet twice, and the number of nodes it traverses remains the same.

For our purposes, we need to define an operation that keeps track of color classes as well. First we extend the definition of *time path* as follows. Let z denote a special reserved symbol, P the set of packets, and V the set of nodes of the cycle. A time path is now a function that maps a time step to a pair from the Cartesian product $(P \cup \{z\}) \times V$, subject to the following conditions.

- (i) If $p \in P$ and $\tau(t) = (p, v)$, then p resides at node v at time t .

- (ii) If $\tau(t) = (p, v)$ and $\tau(t + 1) = (q, w)$ then
 - (a) If $p \in P$ and $p \neq q$, then p resides at node w at time $t + 1$.
 - (b) If $p = z$, then (v, w) is an edge of G .

Essentially, we now explicitly keep track of the node at which a time path resides, as well as the packet it is following. Conditions (i) and (ii)(a) are straightforward; condition (ii)(b) in effect says that the time path cannot be delayed as long as it is following the special symbol z .

Now consider the packet p in the statement of the claim, and suppose we start with the time path that simply follows p and then repeatedly invoke the Switch operation until it is no longer applicable. Let τ be the resulting time path. Suppose that at time t' , τ is blocked by a packet q , and at time t'' it meets a packet s , and q and s belong to the same color class. Again, suppose that t' is minimal with this property. In this situation, we define the following new time path, which we denote by $S_2(\tau)$:

- (i) Follow τ until time $t' - 1$, just before τ is blocked by q .
- (ii) Then follow packet q until it is absorbed (which cannot come after the origin of s , since they belong to the same color class).
- (iii) Then follow the special symbol z forward one node per time step.
- (iv) Now, since Switch was not applicable, q does not meet τ again after time t' . Thus z begins strictly ahead of τ but not ahead of s . But τ eventually meets s ; since z moves forward one node per time step, it reaches s before τ does.
When z first reaches s , $S_2(\tau)$ begins following s .
- (v) Finally, when τ first meets s , $S_2(\tau)$ resumes following τ .

Our construction is thus to repeatedly perform Switch whenever it is applicable, and otherwise to perform the S_2 operation if it is applicable. We must show that this sequence of operations terminates; here, the argument is very similar to the one in [10]. For a time path τ , define $\rho_2(\tau)$ to be the minimal time step at which Switch is applicable to τ , or the minimal time step at which S_2 is applicable to τ if Switch is not applicable; and it is undefined otherwise. Suppose that our sequence of transformations produces time paths $\tau_0, \tau_1, \tau_2, \dots$. We now argue that if there are indices $i < j$ such that $\rho_2(\tau_i) = \rho_2(\tau_j)$, then there is a k between i and j such that $\rho_2(\tau_k) < \rho_2(\tau_i)$. To see this, note that τ_{i+1} is not delayed at time step $\rho_2(\tau_i)$, but that τ_j is; thus, a Switch or S_2 operation must have been invoked before time $\rho_2(\tau_i)$ on some τ_k between τ_{i+1} and τ_j . Finally, by an argument in [10], this implies that the sequence τ_0, τ_1, \dots is finite.

So when this sequence of transformations terminates, we have a time path τ which is only blocked at most by every color class but one (which doesn't block it at all). Since τ either crosses an edge or is blocked at every time step in the interval before p is absorbed, this implies the claimed bound. \square

From this claim, Theorem 4 follows easily, since the increase in Φ in steps of type (1) is now seen to be at most

$$r(m + n + r) + r(r + 1) = rm + rn + 2r^2 + r.$$

\square

Using the technique of Theorem 2, we obtain

Corollary 5 *Let G denote the n -node cycle and \mathcal{S} an arbitrary stochastic adversary that injects path packings with rate $1 - \varepsilon$, for some $\varepsilon > 0$, and variance bounded by θ . Then LIS is stable against \mathcal{S} .*

3.4 The Furthest-to-Go Protocol

In this section, we prove that the FTG protocol is stable for the ring, first for injection of path-packings by a deterministic adversary at injection rate 1, and then for injection of path-packings by a stochastic adversary at injection rate $1 - \varepsilon$. For simplicity, we assume that all packets are traveling in a clockwise direction.

To prove this, we define a quantity $\psi(i, j, t)$ for $0 \leq i, j \leq n - 1$. Throughout this section, all arithmetic on node names is mod n . The quantity $\psi(i, j, t)$ denotes the number of packets at time t in the queues in nodes $i, i + 1, \dots, j$ (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 ψ is determined). We first prove the following lemma.

Lemma 1 *Assume the system is empty at time 0. For any time $t \geq 0$, $\psi(i, j, t) \leq j - i$.*

Proof: We first argue that the only way $\psi(i, j, t)$ can increase is due to the insertion of packets and can only go up by one per insertion of a path-packing. 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 $\psi(i, j, t)$. Additionally, only one packet per path-packing may traverse $(i - 1, i)$, and so $\psi(i, j, t + 1) - \psi(i, j, t) \leq 1$.

We now prove the lemma by double induction, the outer induction on $j - i$, and the inner on t . By hypothesis $\psi(i, i, 0) = 0$. Assume $\psi(i, i, t - 1) = 0$. Then $\psi(i, i, t) = 0$, since by the FTG protocol any packet inserted at time t at node i that needs to cross the edge $(i - 1, i)$ must have the highest priority and is moved at time t : no other packet in the queue can have more than distance n to go, and none currently has distance n to go.

Now assume that $\psi(i, j, t) \leq j - i$ is true for any $j - i < k$ and any t . We claim that it is true for any i, j such that $j - i = k$ and any t . Certainly $\psi(i, j, 0) = 0$ is true by

hypothesis. Now assume that $\psi(i, j, t-1) \leq j-i$. We consider two cases. If $\psi(i, j, t-1) < j-i$ then the claim is proven since ψ can increase by at most one. In the other case, $\psi(i, j, t-1) = j-i$. Then $\psi(i, j, t) \leq j-i+1$. Suppose (by way of contradiction) that $\psi(i, j, t) = j-i+1$. By assumption we know that $\psi(i, j-1, t) \leq j-i-1$, so that there are at least $\psi(i, j, t) - (j-i-1) \geq 2$ packets at node j at the end of time t that must traverse the edge $(i-1, i)$. At most one of them could have moved to node j during time step t , so at least $\psi(i, j, t) - (j-i) \geq 1$ were present in the queue at node j when packets were moved. But if any such packets were present, one would have had the highest priority at node j , and it would have been moved to $j+1$ by the FTG protocol, and therefore $\psi(i, j, t) \leq \psi(i, j, t-1) \leq j-i$. \square

Notice that the proof only required $\psi(i, j, 0) \leq j-i$, rather than the stronger assumption of an empty system at time 0. The theorem below follows immediately from the lemma, since the total number of packets in the system at time t can be upper bounded by $\sum_{i=0}^{n-1} \psi(i, i-1, t)$, which will be no greater than $n(n-1)$.

Theorem 6 *Let G denote the n -node cycle and \mathcal{A} an arbitrary deterministic adversary that injects path packings at rate 1. Then FTG is stable against \mathcal{A} , and there are never more than $n(n-1)$ packets in the system.*

We now extend the above theorem to the case of a stochastic worst-case adversary with injection rate $1-\epsilon$. Our main theorem here is:

Theorem 7 *Let G denote the n -node cycle and \mathcal{S} an arbitrary stochastic adversary that injects path packings with rate $1-\epsilon$, for some $\epsilon > 0$, and variance bounded by θ . Then FTG is stable against \mathcal{S} .*

We prove this by introducing a potential function which attempts to capture the intuition of the proof of Lemma 1. The potential is defined in terms of the quantities $\psi(i, j, t)$ as follows: $\phi(i, j, t) = \max\{\psi(i, k, t) + (j-k) : i \leq k \leq j\}$.

Lemma 2 *For all i , and for all t , the quantity $\phi(i, i+n-1, t)$ satisfies the following Martingale property: If $\phi(i, i+n-1, t-1) \geq n$ then $E[\phi(i, i+n-1, t) | \phi(i, i+n-1, t-1)] \leq \phi(i, i+n-1, t-1) - \epsilon$.*

Proof: We divide the time step from the beginning of t (or the end of $t-1$) to the end of t into two sub-phases. In the first step we consider the injections. In the second we consider the routing progress. Let ϕ_{-1} , $\phi_{\frac{1}{2}}$, and ϕ refer to the potential $\phi(i, i+n-1, t)$ at the beginning, the middle and the end of the time step t respectively. Define $\psi_{-1}(i, k, t)$ and $\psi_{\frac{1}{2}}(i, k, t)$ similarly. We assume that $\phi_{-1} \geq n$ which implies $\phi_{\frac{1}{2}} \geq n$.

We argue as in the non-stochastic case, that as packets are inserted at the beginning of the time step t , the potential increases by at most the number of inserted packets at time step t which wish to cross the edge $i-1$ to i . Since the

expected number of insertions is $1-\epsilon$, we conclude that

$$E[\phi_{\frac{1}{2}}] - \phi_{-1} \leq 1 - \epsilon. \quad (5)$$

For the second sub-phase, let k be any index which achieves $\psi_{\frac{1}{2}}(i, k, t) + i + n - 1 - k = \phi_{\frac{1}{2}} \geq n$. We will show for any such k that $\psi_{\frac{1}{2}}(i, k, t) = \psi(i, k, t) + 1$ and in turn we get $\phi \leq \phi_{\frac{1}{2}} - 1$. We assert that for any such k , there is at least one packet queued at the node k at the middle of time step t which wishes to cross the edge $i-1$ to i . For $k \neq i$, this follows or else $\psi_{\frac{1}{2}}(i, k, t) + i + n - 1 - k < \psi_{\frac{1}{2}}(i, k-1, t) + i + n - 1 - (k-1)$. In the case $k = i$, we claim our assumption that $\phi_{\frac{1}{2}} \geq n$ guarantees there must be at least one packet at node i which wishes to cross the edge $i-1$ to i . Otherwise $\psi_{\frac{1}{2}}(i, i, t) + i + n - 1 - i = n - 1$. Thus for every such k , one packet needing to cross edge $(i-1, i)$ is moved from k to $k+1$ when packets are routed. This implies that $\psi_{\frac{1}{2}}(i, k, t) = \psi(i, k, t) + 1$. We conclude that

$$\phi \leq \phi_{\frac{1}{2}} - 1. \quad (6)$$

From (5) and (6) we get that $E[\phi] \leq \phi_{-1} - \epsilon$, provided $\phi_{\frac{1}{2}} \geq n$. \square

The lemma now allows us to use Martingale type tail inequalities as in the proof of Theorem 2. Theorem 7 now follows.¹

4 Results for General Networks

Our purpose here is to note the following partial generalization of Theorem 4 to the case in which G now denotes an arbitrary directed graph. Here, we have a deterministic adversary that injects single shortest paths at rate $1-\epsilon$. We use the same potential function as in the proof of Theorem 4. Note that now each packing consists of single packet, so we can apply the theorem of Mansour and Patt-Shamir [10] directly to conclude that the potential drops whenever $k = \Omega(\epsilon^{-1} \cdot \text{diam}(G) + \epsilon^{-2})$. Thus we can show

Theorem 8 *Let G denote an arbitrary directed graph of diameter d , and \mathcal{A} denote an arbitrary deterministic adversary that injects single shortest paths at rate $1-\epsilon$, for some $\epsilon > 0$. Then LIS is stable against \mathcal{A} , and there are never more than $O(\epsilon^{-1}(d + \epsilon^{-1}))$ packets in the system.*

Corollary 9 *Let G denote an arbitrary directed graph and \mathcal{S} an arbitrary stochastic adversary that injects shortest paths with rate $1-\epsilon$, for some $\epsilon > 0$, and variance bounded by θ . Then LIS is stable against \mathcal{S} .*

5 Meshes, Trees, and DAG's

An argument that builds on Lemmas 2 and 4 in Leighton [9] yields the following theorem.

¹The proof of Lemma 2 may appear to be independent of the variance of the stochastic adversary. Actually, this is not really the case. The variance does show up in the application of the Martingale tail inequality and it is still the case that our bound will not hold for an adversary with infinite variance.

Theorem 10 For one-bend routing on the mesh, the furthest-to-go discipline is stable for any adversary that injects path packings at rate $1 - \epsilon$.

An induction argument similar in spirit to the proof of Theorem 6 allows us to show that any greedy protocol is stable on a directed acyclic graph against the strongest type of deterministic adversary.

Theorem 11 Let G denote an arbitrary directed acyclic graph, \mathcal{P} an arbitrary greedy protocol, and \mathcal{A} an arbitrary deterministic adversary that injects path packings at rate 1. Then \mathcal{P} is stable against \mathcal{A} .

Proof: For e an edge of G , let $Q_t(e)$ denote the queue at the tail of edge e at time t , and let $D_t(e)$ denote the set of packets at time t that are eventually destined to cross edge e . Finally, $G(e)$ is a multiset of edges defined inductively as follows.

- If the tail of e is a sink, then $G(e) = \{e\}$.
- Otherwise the tail of e is entered by edges f_1, \dots, f_k ; we define $G(e)$ to be the multiset union of $\{e\}, G(f_1), \dots, G(f_k)$.

We claim that for all $t \geq 1$ and all $e \in G$, we have

$$|D_t(e)| < |G(e)|. \quad (7)$$

The theorem will then follow, since this gives an absolute upper bound on the number of packets in the system.

The proof of this claim proceeds by induction on t , and for t by induction on $|G(e)|$. The claim clearly holds when $t = 1$ and when $|G(e)| = 1$. Now let $t > 1$, and e emanate from a non-sink, so that it is entered by edges f_1, \dots, f_k . By the inductive definition of $G(e)$, we have

$$|G(f_1)| + \dots + |G(f_k)| + 1 = |G(e)|.$$

Also, $D_t(e)$ can only increase by one when a path-packing is inserted. So if $|D_{t-1}(e)| < |G(e)| - 1$, the induction will hold, and if $Q_{t-1}(e)$ is non-empty, then a packet will cross edge e and offset the increase due to injection.

Finally, if $|D_{t-1}(e)| = |G(e)| - 1$, and $Q_{t-1}(e)$ is empty, then $D_{t-1}(e)$ can be written as the union of sets D_1, \dots, D_k , where the packets in D_i must cross edge f_i . Thus we have

$$|D_1| + \dots + |D_k| \geq |D_{t-1}(e)|,$$

and also

$$|D_{t-1}(f_i)| \geq |D_i|.$$

So by the pigeonhole principle,

$$|D_{t-1}(f_i)| \geq |G(f_i)|$$

for some i , contradicting the induction hypothesis. \square

The upper bound on the number of packets in the system that follows from this proof is exponential in the number of edges of G ; we have a lower bound showing that it can be quadratic. For tree networks, the same proof gives a much better bound; here, the sets $G(e)$ all have linear size.

6 Conclusion

We have introduced a new approach for the study of queueing networks. We can view continuous packet routing as one non-trivial application area in which service time distributions are all identical and independent of the class of service (in this case, one class per path) as well as independent of the number of jobs (i.e., packets) waiting for service at any edge. Clearly extending the adversarial approach to more general queueing networks offers additional challenges.

As was stated in the introduction, prior work on packet routing that assumed constant time per edge traversal also usually assumed random destinations. Our goal (and the goal of Harchol-Balter and Black [3]) was to extend these results to specified destinations and paths. For stability, it is obviously a necessary condition that the arrival rate of packets is such that the induced arrival rate on any edge is less than one. Our primary open question, then, is to provide a general sufficient condition for stability (in our adversarial model, or in any of the more traditional models).

As mentioned in Section 2, recent work of Awerbuch and Leighton [2] shows that in this adversarial model, stability cannot be determined based on rate alone. In particular, they report the following results: (i) the *Newest-In-System* discipline is stable for every network and every adversary with rate $\rho < 1$ (albeit with an exponential upper bound on queue size); and (ii) there exists a network, with deterministic and stochastic adversaries of rate $\rho < 1$, for which some greedy queueing discipline is not stable. They also indicate that their packet-based multicommodity flow algorithm [1] may be of use in designing queueing disciplines for the adversarial model.

It remains an interesting open question to find some set of general conditions (depending, for example, on the rate, queueing 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.

Acknowledgements

We thank Nabil Kahale, Tom Leighton, and Ken Sevcik for valuable discussions on queueing theory and routing.

References

- [1] B. AWERBUCH AND F. T. LEIGHTON. Improved approximations for the multi-commodity flow problem and local competitive routing in networks. *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing*, 1994.
- [2] B. AWERBUCH AND F. T. LEIGHTON. personal communication.
- [3] M. HARCHOL-BALTER AND P. E. BLACK. Queueing analysis of oblivious packet-routing algorithms. *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1994.

- [4] M. HARCHOL-BALTER AND D. WOLFE. Bounding delays in packet-routing networks. *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing*, 1995.
- [5] K. MANI CHANDY, J. HOWARD AND D. TOWSLEY. Product form and local balance in queuing networks. *Journal of the ACM*, April 1977, pp 250-263.
- [6] L. KLEINROCK. **Queueing systems**. Wiley, New York, 1975.
- [7] F. P. KELLY. **Reversibility and Stochastic Networks**. Wiley, New York, 1979.
- [8] N. KAHALE AND T. LEIGHTON. Greedy dynamic routing on arrays. *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1995.
- [9] T. LEIGHTON. Average case analysis of greedy routing algorithms on arrays. *Proceeding of the Second Annual ACM Symposium on Parallel Algorithms and Architectures*, 1990.
- [10] Y. MANSOUR, B. PATT-SHAMIR. Greedy packet scheduling on shortest paths. *Proceedings of the Tenth Annual ACM Symposium on Principles of Distributed Computing*, 1991.
- [11] D. MITRA AND R. CIESLAK. Randomized parallel communications. *Proceedings of the International Conference on Parallel Processing*, 1986.
- [12] M. MITZENMACHER. Bounds on the Greedy Routing Algorithm for Array Networks. *Proceedings of the Sixth Annual ACM Symposium on Parallel Algorithms and Architectures*, 1994.
- [13] 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.

A Appendix: Brief overview of previous work

Queueing theory is a well developed subject of fundamental importance to computer science, providing the mathematical foundation and the main analytical tools for the performance evaluation of computer systems. As clearly articulated in a number of recent papers (see [13], [12] and [4]), queueing theory usually assumes an *immediate* queueing discipline — for example, *processor sharing* — or the assumption that the service times are exponentially distributed. For routing networks, on the other hand, an edge traversal (which plays the role of a server) takes a fixed constant amount of time. We will first discuss why replacing exponentially distributed servers by constant time servers has thus far been such a stumbling block.

A.1 Basic concepts

We begin with a review of some basic definitions and results from queueing theory (see, for example, the texts of Kelly [7] and Kleinrock [6]).

We can view a queueing network as a Markov process in which the “state” of the system changes whenever there is an injection or service completion. Here the state of the

network is a vector describing the state of each server; the state of a server describes the jobs currently waiting (i.e., queued) for service as well as the job(s) being serviced and the remaining time of such service. We say that a queueing network is stable if this Markov process has a stationary distribution and call this the equilibrium state. In particular, a stable network has the property that if N represents the total number of jobs in the system, then $Prob[N > \eta]$ limits to 0 as η increases. For our adversarial models, we have defined stability in terms of this boundedness property (either in an absolute or probabilistic sense).

When the external arrivals to a server are all generated by independent Poisson distributions and all of the service times are exponentially distributed, then the queueing network is an example of a *product form* network. In addition, when the queueing discipline is FIFO, the network is called a classed *Jackson queueing network*. (The term “classed” indicates that jobs are partitioned into classes with different requirements.) Another example of a product form network is obtained when the servers are continuously operating in a processor sharing mode; that is, if at any instant of time, there are k jobs requiring service from a server v having service rate r , then v is able to serve each of these k jobs at a rate of r/k . In a product form network, *total job arrivals* at a server (e.g., new injections plus jobs from elsewhere in the network) constitute a time independent (but not necessarily Poisson) distribution. The salient property of a product form queueing network is that the equilibrium state (if it exists) is the product of the equilibrium states of each server; that is, in a product form network we can consider each server as if it were an isolated single server with an independent input distribution (and rate corresponding to the total arrival rate) and with its given service time distribution. Characterizations of which networks are product form can be found in Chandy,Howard and Towsley ([5]). In particular, for class independent service times, they show that a network using a FIFO queueing discipline must have exponentially distributed service times. Thus dynamic packet routing cannot generally be viewed as a case of product form networks.

Product form networks are relatively easy to analyze once one can analyze the total arrival rate at each server. For example, in the Jackson network model, suppose the total arrival rate ρ_v is less than the service rate λ_v at each server v . (Here “rate” denotes the mean of the corresponding distribution.) Then the system is stable and furthermore the expected number of jobs at each server is $\rho_v/(\lambda_v - \rho_v)$. Little’s Theorem states that $N = T * R$ where N is the expected number of jobs in the system, T is the expected time in the system and R is the total injection rate into the system. Thus a bound on the expected number of jobs at each server yields a bound on the expected time (for a random job) in the system.

A.2 Routing on arrays

Before discussing attempts (see [13], [12] and [4]) to apply queueing theory results to the case of constant time servers, we note that the first results on dynamic packet routing did not use queueing theory.

Leighton ([9]) 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 and packets are injected at each node according to a Bernoulli distribution with rate $\alpha < 4/n$. We note that all the dynamic packet routing results to date concern packets having 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.

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). Without appealing to queueing theory, 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 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 farthest-to-go scheduling. The precise nature of these packet delay results seems beyond what we can hope to derive from applying general queueing theory results. On the other hand, the Leighton and Kahale-Leighton results are specific to certain networks, utilize the random destination assumption (although perhaps not in any essential way), and seem to depend on restricted routing rules.

A.3 Markovian networks

A different approach to network stability and expected packet delays is taken in Stamoulis and Tsitsiklis [13]. They consider the case of layered, *Markovian networks*. In Markovian queueing networks, jobs are indistinguishable 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 routing, the Markovian 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. They consider such networks under the assumption of Poisson arrivals. They observe that for layered Markovian 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 which can be layered (e.g., dimension by dimension routing to random destinations in the hypercube).² 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 queueing 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.

Harchol-Balter and Black [3] consider the analysis of packet routing in arrays and toroidal networks with respect to the Jackson network model. Following similar experiments by Mitra and Cieslak [11] for the Omega network, Harchol-Balter and Black simulate packet routing on array networks and conjecture that the queue sizes that obtain under the exponentially distributed edge traversal time assumption are an upper bound for the queue sizes obtained with constant time edge traversal. Mitzenmacher [12] proves this conjecture for one bend routing on arrays as follows. He first observes that the equilibrium state for a network of exponentially distributed servers with mean service rate 1 is the same as the equilibrium state for a network of processor sharing servers with service rate 1 (recall that both are product form networks) and then applies the Stamoulis and Tsitsiklis results to the case of one bend routing on arrays for packets with random destinations (which can be viewed as a layered Markovian network). Combining these observations with the total arrival rates calculated by Harchol-Balter and Black, he derives very precise bounds on the queue sizes and expected time in system for random packet routing on arrays. It is tempting to believe that the experiments and conjecture of Harchol-Balter and Black apply to a much wider context.

However, Harchol-Balter and Wolfe [4] give evidence that it will not be a simple task to apply the approach in [13] to the general study of dynamic packet routing. First they show that the “layered” assumption in [13] is not necessary as they are able to derive the same results for any Markovian network. But they also show that without the Markovian network assumption, it is no longer necessarily true that the set of delays for FIFO with constant time servers is statistically dominated by processor sharing servers.

²Stamoulis 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.