SIAM J. COMPUT.
Vol. 33, No. 2, pp. 286–303

NEW STABILITY RESULTS FOR ADVERSARIAL QUEUING*

ZVI LOTKER†, BOAZ PATT-SHAMIR†, AND ADI ROSÉN‡

Abstract. We consider the model of "adversarial queuing theory" for packet networks introduced by Borodin et al. [*J. ACM*, 48 (2001), pp. 13–38]. We show that the scheduling protocol first-in-first-out (FIFO) can be unstable at any injection rate larger than $1/2$ and that it is always stable if the injection rate is less than $1/d$, where d is the length of the longest route used by any packet. We further show that *every* work-conserving (i.e., greedy) scheduling policy is stable if the injection rate is less than $1/(d+1)$.

Key words. adversarial queuing theory, network protocols, stability, lower bounds

AMS subject classifications. 68M12, 68M20, 68Q25, 68W15

DOI. 10.1137/S0097539702413306

1. Introduction.

Recent years have seen a growing amount of work being concentrated on analyzing packet networks under nonprobabilistic scenarios rather than under probabilistic assumptions (see, e.g., [7, 4, 1, 14, 12, 13, 3, 5]). Much of this work makes use of the model of "adversarial queuing theory" proposed by Borodin et al. [7]. The model can be briefly described as follows. Time proceeds in discrete steps. In each step, packets are injected into the network with their routes. Each packet traverses its respective route hop by hop in a store-and-forward fashion. In each time step, one packet may cross each link, and all other packets waiting for that link are stored in a buffer at the tail of that link. The behavior of the system is determined by the *queuing policy*. The queuing policy chooses, at each time step, for each link, which of the competing packets should be forwarded over that link. One of the main questions in the adversarial queuing model is the question of *stability*. That is, under what conditions is there a bound on the size of the link buffers, as opposed to them growing to infinity as time proceeds? The conditions involve the topology of the network, the queuing policy used, and the injection pattern of the packets. The latter is characterized in the framework of adversarial queuing theory by the *rate* at which packets are injected. Intuitively, the rate of injection is said to be r if, for every link e in the network, the average number of packets requiring e, injected by the adversary in any time step, is at most r (a formal definition of the model is given in section 2). Note that in this model one does not assume any probabilistic assumptions on the behavior of the traffic. Rather, answers are sought under the only assumption that the total bandwidth requested by the adversary is not more than the total bandwidth the network provides.

In the framework of adversarial queuing theory, it is known that some networks are stable for *every* greedy protocol as long as the rate of injection is less than 1, while other networks do not exhibit this phenomenon [7, 4]. The networks which

*Received by the editors August 21, 2002; accepted for publication (in revised form) June 17, 2003; published electronically January 22, 2004. A preliminary version of this paper appeared in *Proceedings of the* 14*th Annual ACM Symposium on Parallel Algorithms and Architectures*, Winnepeg, MB, Canada, 2002, pp. 192–199.
http://www.siam.org/journals/sicomp/33-2/41330.html
†Department of Electrical Engineering, Tel Aviv University, Tel Aviv, 69978, Israel (zvilo@eng.tau.ac.il, boaz@eng.tau.ac.il).
‡Department of Computer Science, Technion, Haifa, 32000, Israel (adiro@technion.ac.il). This author's research was supported in part by the Fund for the Promotion of Research at the Technion and by the Technion V.P.R. Fund.

are always stable have been named "universally stable" networks [7] and have been fully characterized [14, 2]. From the point of view of protocols, some protocols are known to be universally stable; i.e., they are stable on any network topology for any rate of injection $r < 1$. Such protocols are, for example, longest-in-system (LIS) and furthest-to-go (FTG). Other natural protocols, however, are known not to always be stable, e.g., first-in-first-out (FIFO), nearest-to-go (NTG), last-in-first-out (LIFO), and furthest-from-source (FFS) [4]. Furthermore, the protocol NTG (and FFS and LIFO) exhibit the phenomenon of being unstable on certain networks even at arbitrarily low injection rates [7]. Previous papers in this area mentioned as one of the main interesting open problems the question of determining the rate at which the (very commonly used) FIFO policy is guaranteed to be stable and if such rate exists at all. Prior to the present work, it was known that FIFO is not universally stable and that it can be unstable for $r > 0.85$ [4]. This bound was improved to 0.8357 by Díaz et al. [11] and further improved to 0.749 by Koukopoulos, Nikoletseas, and Spirakis [15]. Díaz et al. [11] also presented a formula to calculate, for any given network, a bound so that FIFO is stable on that network if the injection rate is below that bound. In particular, they consider as parameters the number of edges in the network, denoted m, the length of the longest route used by any packet, denoted d, and the maximum in-degree in the network, denoted α; their bound is at most $\frac{1}{2dm\alpha}$ for any network.

The contribution of this paper is twofold. First, we show that FIFO can be unstable for any rate greater than $1/2$ and that, on the other hand, FIFO is always stable if the rate is less than $1/d$. Second, we extend the stability proof for FIFO to show that *any* greedy policy is stable if the injection rate is less than $1/(d+1)$, while previously it was only known for general greedy protocols that the system is stable if the injection rate is bounded by $1/m$ [6].[1] We remark that our stability proofs do not only show that the buffers have bounded size if the rate is sufficiently low. They show, in addition, that the buffer size in this case has an upper bound independent of network parameters (depending only on the parameters of the adversary).

Our instability proof entails new techniques that greatly simplify the analysis of the FIFO policy. In particular, we develop a technique that enables us to construct adversaries for some *acyclic* parameterized networks that we call "gadgets" and then compose these gadgets and adversaries to form a cyclic network and a single adversary that together show instability. Furthermore, we simplify the specification of the adversary by defining conditions under which the adversary is allowed to "reroute" packets, thus allowing us to specify routes for the packets "on the fly."

Recently, subsequent to the initial publication of the present work [17], Bhattacharjee and Goel proved that FIFO can be unstable at arbitrarily low injection rates [8]. Some of the techniques used in their work (such as concatenation of parameterized gadgets and packet rerouting) are similar to the techniques we use in the present work.

Organization. The rest of this paper is organized as follows. In section 2 we define the model formally. In section 3 we prove that FIFO can be unstable for any rate greater than $1/2$. In section 4 we prove our stability results. We conclude with some remarks in section 5.

2. Formal model. We use the adversarial queuing model [7], defined as follows. The communication network is modeled by a directed graph $G = (V, E)$, and we

[1] Recently, we learned that Zhang, Duan, and Hou [18] proved that FIFO is stable for injection rates less than $1/d$. This result was obtained independently of ours.

denote $|V| = n$, $|E| = m$. Each node $v \in V$ represents a communication switch, and each edge $e \in E$ represents a link between two switches. In each node, there is a *buffer* associated with each outgoing link. Buffers store *packets*. Packets are *injected* into the network with a *route*, which is a simple directed path in G. When a packet is injected, it is placed in the buffer of the first link on its route. The system proceeds in global time steps numbered $0, 1, \ldots$. Each time step is divided into two substeps. In the first substep, one packet is sent from each nonempty buffer over its corresponding link. In the second substep, packets are received by the nodes at the other end of the links; they are absorbed (eliminated) if that node is their destination, and otherwise they are placed in the buffer of the next link on their respective routes. In addition, new packets are injected in the second substep.

The task of the *protocol* is to select which packet to send over a link if there is more than one packet in the buffer associated with that link. We remark that we are interested in *greedy* protocols (in fact, the definitions above allow only such protocols), in which a link cannot be idle in a time step if its buffer is nonempty in the first substep. The protocol FIFO selects the packets to be sent from a buffer in the same order as their arrival order at that buffer.

The injection of the packets into the network is modeled as being done by an *adversary*. Following [7], we use the following parameterized definition for the adversary.

DEFINITION 2.1. *Let $\mathcal{A}$ be an adversary. $\mathcal{A}$ is called a (w, r) adversary if, for some $r \leq 1$, called the* rate *of $\mathcal{A}$, and some integer $w \geq 1$, called the* window size *of $\mathcal{A}$, the following holds. For any time $t \in \mathcal{N}$, let $\mathcal{I}^t$ be the set of packets injected during the w time steps from t to $t + w - 1$, inclusive. Let Π^t be the set of paths that the packets in $\mathcal{I}^t$ have to follow. Then the maximum number of times any edge appears in Π^t is at most rw.*

For our instability results we use a weaker adversary, which is not allowed to inject bursty traffic. We call this adversary a *rate-r adversary* [4]: for every interval of time of length t and every edge e, a rate-r adversary may inject at most $\lceil rt \rceil$ packets whose routes require e.

3. Instability of FIFO. In this section we prove that FIFO can be unstable at rate $\frac{1}{2} + \epsilon$ for any $\epsilon > 0$. The high level view of the proof is as follows. First, we define a small acyclic graph called "gadget," which has special "ingress" and "egress" edges. Gadgets can be composed in series by identifying the egress edge of one gadget with the ingress edge of its successor, getting a "daisy chain." We show that a rate-r adversary (for $r > \frac{1}{2}$) can increase the size of a given queue in the ingress edge of the chain by any desired factor to get a large queue at the egress edge of the chain (using a sufficiently long chain). We then prove that a queue in the egress edge of the chain can be translated to a queue of fresh packets in the ingress edge of the chain by losing only a fraction of the size of the queue.

Since in our construction packets have long routes, we find it more convenient to specify the routes in an "on-line" fashion. That is, when we construct the adversary, we do not specify the complete routes of the packets when they are injected (even though we can, in principle). Rather, we prove below some conditions that allow us to reroute packets without violating the capacity constraints. Formally, this is done by altering the adversary. We find this technique useful in the sense that it makes the construction more "localized." We stress that this is just a matter of representation: the actual adversary used to prove the results is the same rate-r adversary used, e.g., in [4, 11, 15].

The proof is structured as follows. In section 3.1 we specify the conditions under which packets can be rerouted. In section 3.2 we specify and analyze a rate-r adversary for two daisy-chained gadgets. Some small adversaries used for "gluing" and the overall adversary are specified in section 3.3.

3.1. Packet rerouting. In this section we prove a technical lemma that allows us to construct adversaries "on the fly" for FIFO. Informally, it says that if there is a set of packets that have routes that already share a single edge, then these packets can be arbitrarily rerouted as long as they are routed to new edges. In fact, the rerouting technique can be applied to a large class of queuing policies defined below.

DEFINITION 3.1. *A queue policy is called* historic *if the scheduling decisions are independent of the remaining routes beyond the next edge of each packet.*

Note that policies that are based on the arrival time at the buffer (such as FIFO and LIFO), on the injection time (e.g., LIS and NIS), or on the route from the source (e.g., FFS) are examples of historic policies. Note that a historic queue policy must not even depend on the destinations of the packets. For example, FTG and NTG are not historic. (Historic policies are called *nonpredictive* in [16].)

First, we define formally the notion of new edges.

DEFINITION 3.2. *Let G be a graph, $\mathcal{Q}$ a queuing policy, and $\mathcal{A}$ a rate-r adversary. Let t be a time step in the execution of $\mathcal{Q}$ in G under $\mathcal{A}$. Let P be a subset of the packets that are in the network at time t. Let t^* be the minimum injection time of all packets in P. An edge e is* new *to P if e is not a member of any route of a packet (either in P or not) injected by $\mathcal{A}$ at times $\tau \geq t^* - \lceil \frac{1}{r} \rceil$.*

We remark that since in this paper we deal with rates larger than $\frac{1}{2}$, then $\lceil \frac{1}{r} \rceil \leq 2$.

We can now state and prove our rerouting claim.

LEMMA 3.3. *Let $\mathcal{Q}$ be a deterministic historic queue policy, G a graph, $\mathcal{A}$ a rate-r adversary, and t a time step. Let $P(t)$ be the set of packets in the network at time t. For each $p \in P(t)$, denote the next edge to be traversed by p at time t by e_p, and denote the complete path of p by $q_p e_p r_p$. Let $P_0 \subseteq P(t)$ be a set of packets whose routes have at least one edge common to all. Then for any set of paths $\{r'_p \mid p \in P_0\}$ that consist of edges that are new to $P(t)$, there exists a rate-r adversary $\mathcal{A}'$ such that the following holds true.*

1. *The execution of the system under $\mathcal{A}$ and $\mathcal{A}'$ is identical until time t.*
2. *For every packet p injected by $\mathcal{A}$ there is a packet p injected by $\mathcal{A}'$ at the same time.*
3. *If $p \in P_0$, then its route under $\mathcal{A}'$ is $q_p e_p r'_p$.*
4. *If $p \notin P_0$, then its route under $\mathcal{A}'$ is $q_p e_p r_p$.*

Proof. Define $\mathcal{A}'$ as follows. $\mathcal{A}'$ injects the same number of packets as $\mathcal{A}$ and at the same times. For $p \in P_0$, set the route of p to $q_p e_p r'_p$. For $p \notin P_0$ set the route of p to $q_p e_p r_p$. Clearly, claims (1), (2), (3), and (4) follow directly from the assumption that $\mathcal{Q}$ is historic and by the construction. We need only to verify that $\mathcal{A}'$ is a rate-r adversary. To see this, first note that the load on any non-new edge may have only been reduced. Now, consider any edge e in $\bigcup_{p \in P_0} r'_p$. Let $\hat{e}$ be the edge common to the routes of all $p \in P_0$. Let t^* be the minimum injection time over all packets in $P(t)$.

Consider any time interval $[t_1, t_2]$. If $t_2 < t^*$, then the number of packets injected in $[t_1, t_2]$ by $\mathcal{A}'$ and that require e is equal to the number of packets injected in $[t_1, t_2]$ by $\mathcal{A}$ and that require e. If $t_2 \geq t^*$ we consider time intervals $I = [t_1, t^*)$ and $I' = [t^*, t_2]$. For interval I the number of packets injected by $\mathcal{A}'$ and that require e is the same as for $\mathcal{A}'$, which is at most $\lceil ((t^* - \lceil \frac{1}{r} \rceil) - t_1) r \rceil$, since e is a new edge with

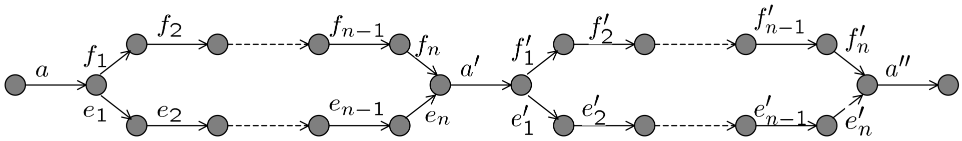

FIG. 3.1. *The graph F_n^2: two F_n gadgets glued together. The left gadget is called F, and the right gadget is called F'. Edge a' is the egress of F and the ingress of F'.*

respect to $P_0 \subseteq P(t)$ and time t^* is the minimum injection time of all packets in $P(t)$ (see Definition 3.2). For interval I' the number of packets injected in I' by $\mathcal{A}'$ and that require e is at most the number of packets injected in I' by $\mathcal{A}$ and that require $\hat{e}$. This is at most $\lceil (t_2 - t^* + 1)r \rceil$. The total number of packets injected by $\mathcal{A}'$ in $[t_1, t_2]$ and that require e is therefore at most $\lceil ((t^* - \lceil \frac{1}{r} \rceil) - t_1)r \rceil + \lceil (t_2 - t^* + 1)r \rceil$. This is at most $\lceil (t_2 - t_1 + 1)r \rceil$, as required. □

Remark 1. Lemma 3.3 allows us to use a "dynamic" adversary that changes the routes of packets on-line. However, this is only a matter of presentation: we do not change the power of the adversary; we only construct it in a succession of refinements. The main advantage of the lemma is that it allows us to modify the remainder of the routes arbitrarily, under the specified restrictions (shared edge in old routes, new edges in modified routes, and historic policy); we do not have to worry about capacity constraints of new edges.

Remark 2. Note that a packet may be rerouted several times, as long as the number of reroutings is finite.

3.2. Gadgets and their adversaries. We now define the gadgets that we use and their local adversaries.

DEFINITION 3.4. *A* gadget *is a directed acyclic graph with one edge called* ingress *emanating from a degree-*1 *source, and one edge called* egress*, leading to a degree-*1 *sink. Given two gadgets G, H, define $G \circ H$ to be the gadget that results from identifying the egress of G with the ingress of H. The ingress of $G \circ H$ is the ingress of G, and the egress of $G \circ H$ is the egress of H.*

For any gadget F, let F^0 denote the single edge graph which is both ingress and egress. For $i > 0$, we denote $F^i = F^{i-1} \circ F$. We call the "$\circ$" operation *daisy-chaining.*

We will use a parametric gadget denoted F_n, which consists of ingress edge a, egress edge a', and two parallel paths of length n from the ingress edge to the egress edge, whose edges are denoted $e_1, \ldots, e_n$ and $f_1, \ldots, f_n$. Figure 3.1 shows F_n^2.

We will construct an adversary that maintains the following *gadget invariant.*

DEFINITION 3.5. *$C(S, F_n)$ is said to be true at a given time if the following holds at that time on graph F_n.*

(1) *The total number of packets in the buffers of $e_1, \ldots, e_n$ is S.*
(2) *For each $i = 1, \ldots, n$, the buffer of e_i is nonempty, and the packets in e_i have remaining routes $e_i, e_{i+1}, \ldots, e_n, a'$.*
(3) *There are S packets in the buffer of edge a, all with the same remaining route $a, f_1, \ldots, f_n, a'$.*
(4) *There are no other packets in F_n.*

In our construction, we use a daisy chain of many gadgets. However, we start by considering two daisy-chained gadgets, namely the graph F_n^2 (Figure 3.1). We denote the first gadget of F_n^2 by F, and the second by F', and add a prime to the name of all edges in F'. The conditions of the following lemma are designed to allow repeated rerouting, but essentially the idea is to have the condition $C(S, F)$ carry over from one gadget to the next, with a larger value S.

LEMMA 3.6. *Let $r = \frac{1}{2} + \epsilon$ for some $\epsilon > 0$. There exist numbers n and S_0 that depend on ϵ, such that for any $S > S_0$, if in the graph F_n^2 we have that for some time τ, all packets present at time τ were injected after time τ_0, and*

- *$C(S, F)$ holds at time τ, and*
- *F' is empty at time τ, and*
- *no packets using edges in F' were injected in the time interval $[\tau_0 - \lceil 1/r \rceil, \tau]$,*

then there exists a rate-r adversary for F_n^2 such that at time $\tau + 2S + n$, $C(S', F')$ holds for some $S' \geq S(1 + \epsilon)$, and F is empty.

Proof. To define the adversary, we use the notation $R_i \stackrel{\text{def}}{=} \frac{1-r}{1-r^i}$ for $1 \leq i \leq n$. Note, for later reference, that for all i,

$$\frac{R_i}{r + R_i} = R_{i+1} \ . \tag{3.1}$$

We first choose parameters under the constraints below:

$$n > \max\left(\frac{\log \epsilon - 2}{\log r}, \quad 1 - \frac{1}{\log r}\right),$$
$$S_0 > \max\left(2n, \quad \frac{n}{2(R_n - R_{n+1})}\right).$$

We remark that for small ϵ values, we get $n = \Theta(\log \frac{1}{\epsilon})$ and $S_0 = \Theta(nr^{-n}) = \Theta(\frac{1}{\epsilon} \log \frac{1}{\epsilon})$ (see the appendix for a detailed derivation of the asymptotic bounds).

Let us assume, for simplicity of notation, that $\tau = 0$. We now specify the adversary that will create a situation where $C(S', F')$ holds for $S' = 2S(1 - R_n)$. In the adversary specification, as well as in the ensuing analysis, we ignore floors and ceilings for the sake of simplicity of presentation. We remark that carrying these throughout the computations would add only additive terms that can be compensated for by using a larger S_0 value (cf. [4, 11, 15]).

The adversary is as follows.

(1) Extend the routes of all packets stored in F at time 0 by adding the path $e_1', \ldots, e_n', a''$.

(2) For every edge e_i' in F' ($i = 1, \ldots, n$), packets are injected at rate r in the time steps $i, i+1, \ldots, i+t_i$, where $t_i \stackrel{\text{def}}{=} \frac{2S}{r+R_i}$. The route of each of these packets is the single edge e_i'.

(3) In the time interval $[1, S]$, rS packets are injected, at rate r, with route $a, f_1, \ldots, f_n, a', f_1', \ldots, f_n', a''$.

(4) Let $X = S' - rS + n$. X packets are injected in the first $X \cdot \frac{1}{r}$ time steps of the interval $[S + n + 1, S + n + S]$, with routes $a', f_1', \ldots, f_n', a''$. (We show later that $0 \leq X \leq rS$.)

First, note that this is a rate-r adversary: part (1) is justified by Lemma 3.3, since the routes of all packets stored in F share the edge a', and the extensions are for new edges as defined in Definition 3.2.

Edges $e_1', \ldots, e_n'$ are used only by part (2) at rate r. Edges $f_1', \ldots, f_n'$ and a'' are used at rate r in parts (3) and (4), which cover disjoint time intervals. It remains to show that $0 \leq X \leq rS$.

CLAIM 3.7. *For every $r < 1$, we have $0 < X \leq rS$.*

Proof. First we prove that $X > 0$. By definitions,

$$\begin{aligned} X \; > \; X - n = S' - rS \\ = 2S(1 - R_n) - rS \\ = S\left(2 - \frac{2-2r}{1-r^n} - r\right) \; . \end{aligned}$$

Now,

$$\begin{aligned} 2 - \frac{2-2r}{1-r^n} - r &= \frac{r - 2r^n + r^{n+1}}{1-r^n} \\ &> \frac{r-2r^n}{1-r^n} \\ &> 2r(1-2r^{n-1}) \\ &> 0 \; , \end{aligned}$$

since $r^n < r^{n-1} < 1/2$ by the choice of n, and hence $X > 0$. Next we prove that $X \le rS$. By the definitions,

$$\begin{aligned} rS - X &= rS - (2S(1-R_n) - rS + n) \\ &= 2S(r + R_n - 1) - n \; . \end{aligned}$$

Since $S \ge S_0 > \frac{n}{2(R_n - R_{n+1})} \ge \frac{n}{2(r+R_n-1)}$ by assumption, we get $rS - X > 0$. ∎

We now show that in fact at time $2S + n$, $C(S', F')$ holds and that F is empty. This will be sufficient, since by the definition of S' we have that

$$\begin{aligned} S' &= 2S(1-R_n) \\ &= 2S\left(\frac{r}{1-r^n} - \frac{r^n}{1-r^n}\right) \\ &\ge 2S(r - 2r^n) \\ &\ge 2S\left(\frac{1}{2} + \epsilon - \frac{\epsilon}{2}\right) \\ &= S(1+\epsilon) \; . \end{aligned}$$

(The inequalities follow from the fact that $1 - r^n \le 1$ and since $r^n \le 1/2$ and $4r^n < \epsilon$ by the choice of n.)

We now proceed to prove that $C(S', F')$ holds at time $2S + n$. Let us call the packets described in part (1) of the definition of the adversary *old* packets, the packets described in part (2) *new short* packets, and the packets described in parts (3) and (4) *new long* packets.

We start with the following straightforward property.

CLAIM 3.8. *In each step in the time interval* $[1, 2S]$, *one old packet crosses* a'.

Proof. Since there are S old packets in the buffers of edges e_i, there are no other packets in these buffers, and none of them is empty at time 0, then these S packets will arrive at the tail of a' one in each time step in time interval $[1, S]$. The S packets stored at the tail of a at time 0 will arrive at the tail of a', one in each time step, in the time interval $[n, S + n]$. Since $S > n$, the claim follows. ∎

The next claim shows that old packets cross edges e'_i at rates that decrease as i grows. This is due to the injection of the new short packets.

CLAIM 3.9. *The following holds for any edge* e'_i, $i \in [1, n]$.

(1) *At times* $[0, i]$, *no packet arrives at the tail of* e'_i.
(2) *At times* $[i + 1, 2S + i]$, *old packets arrive at the tail of* e'_i *at rate* R_i.
(3) *At time* $i + 2S + 1$, *there are no new short packets in the buffer of* e'_i.

Proof. Part (1) is straightforward by the fact that old packets must cross at least $i + 1$ edges before they arrive at the tail of e'_i and because no new packet is injected for e'_i before time i. Part (2) is proven by induction on i. For the basis $i = 1$ we have that packets arrive at the tail of e'_1 at rate $R_1 = 1$ by Claim 3.8. For the induction step, let $i > 1$. The induction hypothesis says that packets arrive at the tail of e'_{i-1} at rate R_{i-1}. By part (2) of the definition of the adversary, new packets are injected at the tail of e'_{i-1} at rate r. Note that $R_{i-1} + r > 1$. Since the queue policy is FIFO, it follows that old packets cross e'_{i-1}, and hence arrive at the tail of e'_i, at rate $\frac{R_{i-1}}{R_{i-1}+r}$. By (3.1) this is exactly R_i. This proves part (2). To see that part (3) is true, note that as a consequence of part (2), we have that short new packets cross e'_i at rate $\frac{r}{R_i+r}$. The last short new packet for e'_i is injected at time $i + t_i = i + \frac{2S}{r+R_i}$, in which time there are $t_i(r + R_i - 1)$ packets in the buffer of e'_i. Using the definition of t_i, it follows that all new short packets of e'_i will be absorbed by time

$$i + t_i + t_i(r + R_i - 1) = i + t_i(r + R_i) = i + 2S \; . \qquad \square$$

Using the above claims, we show that $C(S', F')$ holds at time $2S + n$. We start with part (1) of $C(S', F')$.

CLAIM 3.10. *At time* $2S+n$, *there is a total of* S' *old packets stored in the buffers of edges* e'_i.

Proof. By Claim 3.9, $2S \cdot R_n$ old packets cross a'' by time $2S + n$. On the other hand, by Claim 3.8, all the $2S$ old packets crossed a' by time $2S$. The claim follows. $\square$

Next, we prove that part (2) of $C(S', F')$ holds.

CLAIM 3.11. *If* $S > S_0$, *then none of the buffers of* e'_i *is empty at time* $2S + n$. *Moreover, the route of any packet stored in* e'_i *at that time is* $e'_i, \ldots, e'_n, a''$.

Proof. The claim on the remaining routes is obvious from the construction. We now prove that the buffer of e'_i is not empty. The last short packet for e'_i is injected in e'_i at time $i + t_i$, and, as argued in the proof of Claim 3.9, it crosses e'_i at time $2S + i$. Hence all packets that arrive at the buffer of e'_i in the time interval $[i + t_i, 2S + i]$ are still in the buffer of e'_i at time $2S + i$. All these packets are old packets that arrive from e_{i-1}. By Claim 3.9, there are $(2S - t_i)R_i$ such packets. Let $Q_i \stackrel{\text{def}}{=} (2S - t_i)R_i$ be the number of packets in the buffer of e'_i at time $2S + i$. Note that by definition, $t_i \leq t_{i+1}$ and $R_i \geq R_{i+1}$, and hence $Q_i \geq Q_{i+1}$ for $1 \leq i < n$. In addition, since only $n - i$ packets may leave the buffer of e'_i in the time interval $[2S + i, 2S + n]$, it is sufficient to prove that $Q_n \geq n$. Substituting the values we get

$$\begin{aligned} Q_n &= (2S - t_n)R_n \\ &= 2SR_n - t_nR_n \\ &= 2S\left(R_n - \frac{R_n}{r + R_n}\right) \\ &= 2S\left(R_n - R_{n+1}\right) \; . \end{aligned}$$

Since $S \geq S_0 > \frac{n}{2(R_n - R_{n+1})}$, we get that $Q_n \geq n$. $\square$

We now prove that part (3) of $C(S', F')$ holds.

CLAIM 3.12. *The number of packets at the tail of* a' *at time* $2S + n$ *is* S'.

Proof. First, observe that in time interval $[1, S+n]$ the number of packets that arrive at the tail of a' is exactly $2S$, and they start arriving at time 1. Therefore at time $S+n$ there are exactly $S-n$ packets in the buffer of a'. In addition, by part (3) of the definition of the adversary, rS new long packets are injected at the tail of a in the time interval $[1, S]$. These packets start crossing a at time $S+1$, since they are queued behind the S old packets stored in a at time 0. Hence the new long packets start arriving at a' at time $S+n+1$. In addition, part (4) of the definition of the adversary says that X new long packets are injected at the tail of a' during time interval $[S+n, 2S+n]$. In conclusion, there are $X+rS$ new long packets arriving at a' in the interval $[S+n, 2S+n]$. Together with the $S-n$ packets stored at the tail of a' at time $S+n$, we have that at time $2S+n$, the number of packets stored in the buffer of a' is exactly $rS+X-n=S'$, by definition of X. All these packets have the paths that are required by $C(S', F')$. ◻

To conclude the proof of Lemma 3.6, we argue that F is empty at time $2S+n$. This follows from the fact that there are no injections into edges of F during time interval $[0, 2S+n]$ and that all the $2S$ packets present in F at time 0 arrive at the tail of the ingress of F' by time $S+n$. ◻

3.3. Putting the gadgets together. In this section we describe how to construct the overall adversary, using the gadget adversary described in section 3.2, and a few other simple adversaries used to glue things together.

The idea of the proof is to use a sufficiently long daisy chain of gadgets that blows up the queue size by a sufficiently large factor (that depends on the length of the chain and r) and then "stitch together" the egress of the chain to its ingress, getting a queue of fresh packets. The stitching process loses a fraction (that depends on r) of the queue size, but this loss is more than compensated by the chain of gadgets.

Fix $r=\frac{1}{2}+\epsilon$ for $\epsilon>0$ and S_0 and n as in the proof of Lemma 3.6. Consider the graph F_n^M that consists of a daisy chain of M F_n gadgets, where M is a parameter. Let the kth gadget be denoted by $F(k)$ for $1 \le k \le M$. We now prove the following lemma.

LEMMA 3.13. *Let M be a positive integer, and consider the graph F_n^M. If for some time τ we have that all packets present in the network were injected after time τ_0, and*

- *$C(S, F(1))$ holds at time τ for $S \ge S_0$,*
- *there are no other packets in F_n^M at time τ, and*
- *the edges of $F_n^M \setminus F(1)$ were not used by any injection in the time interval $[\tau_0 - \lceil 1/r \rceil, \tau]$,*

then there is a rate-r adversary such that at some time $t>\tau$, there are S' packets at the egress of F_n^M, for $S' \ge S(1+\epsilon)^{M-1}/2$, and there are no other packets in F_n^M.

Proof. We first prove the following claim.

CLAIM 3.14. *Let $1 \le i \le n$. If at time τ we have that all packets present in the network were injected after time τ_0, and*

- *$C(S, F(1))$ holds for $S \ge S_0$,*
- *there are no other packets in $F(1), \ldots, F(i)$, and*
- *the edges of $F(2), \ldots, F(M)$ were not used by any injection in the time interval $[\tau_0 - \lceil 1/r \rceil, \tau]$,*

then there is a rate-r adversary and time $t_i \ge \tau$ such that

- *$C(S', F(i))$ holds for $S' \ge S(1+\epsilon)^{i-1}$ at time t_i,*
- *there are no other packets in F_n^M, and*

- *the edges of $F(i+1), \ldots, F(M)$ were not used by any injection in the time interval $[\tau_0 - \lceil 1/r \rceil, t_i]$.*

Proof. The proof is by induction on i. For $i = 1$ the claim is trivial with $t_1 = \tau$. For the induction step, assume that the lemma holds for $1 < i < M$, i.e., that there exists an adversary $\mathcal{A}_i$ and time t_i such that at time t_i, $C(S_i, F(M))$ holds for $S_i \geq S(1+\epsilon)^{i-1}$. Consider now the subgraph that consists of $F(i)$ and $F(i+1)$. By the induction hypothesis, we may apply Lemma 3.6 to know that there exists an adversary $\mathcal{A}$ such that at time $t_i + 2S_i + n$, $C(S', F(i+1))$ holds for $S' \geq S_i(1+\epsilon) \geq S(1+\epsilon)^i$. We note that the packets injected by $\mathcal{A}$ (as specified in Lemma 3.6) do not use any edge in $F(i+2), \ldots, F(M)$ and that the application of this adversary leaves $F(i)$ empty of packets. This proves the claim with $t_{i+1} = t_i + 2S_i + n$ and the adversary that results from concatenating the adversaries $\mathcal{A}_i$ and $\mathcal{A}$. ∎

To complete the proof of Lemma 3.13, we observe that if at time t we have that $C(S, F_n)$ holds for some gadget F_n and $S \geq S_0$, and if no injections are done in the interval $[t, t+S+n]$, then at time $t+S+n$ there are at least $S/2$ packets queued at the egress of F_n. This is true since during time interval $[t+1, t+S+n]$ exactly $2S$ packets arrive at the tail of the egress of F_n, and, therefore, at time $t+S+n$ there are $S - n \geq S_0 - n \geq S/2$ packets in the egress buffer.

Note that a packet may be rerouted in the whole construction at most $M-1$ times: once for each gadget $F(2), \ldots, F(M)$. This completes the proof of Lemma 3.13. ∎

We now specify two more constructions. The first shows how to establish $C(S, F_n)$ starting from a state in which the only packets in F_n are in the buffer of the ingress of F_n. The second construction shows how to replace a queue of packets with another queue of *fresh* packets. This is necessary so we can stitch the end of the daisy chain to its beginning.

We now claim the existence of an adversary that establishes $C(S, F_n)$ starting from a single buffer. The construction is a variant of the adversary presented in the proof of Lemma 3.6.

LEMMA 3.15. *For any $\epsilon > 0$, let n and S_0 be as in Lemma 3.6. Let $S > S_0$, and let τ be a time step. Suppose that at time τ all the packets in the network are $2S$ packets stored in the ingress edge of F_n, they all have remaining routes of length 1, and they were all injected after time τ_0 for some τ_0. If the other edges of F_n were not used by any injection in the time interval $[\tau_0 - \lceil 1/r \rceil, \tau]$, then there is a rate-$r$ adversary for $r = \frac{1}{2} + \epsilon$ such that at time $\tau + 2S + n$ condition $C(S', F_n)$ holds for $S' \geq S(1+\epsilon)$.*

Proof. Let us again assume for convenience of notation that $\tau = 0$. We use the notations and definitions of t_i and R_i from the proof of Lemma 3.6. We also define $S' = 2S(1 - R_n)$. The adversary is defined as follows.

(1) Extend the route of the packets stored in the ingress edge a to be $a, e_1, e_2, \ldots, e_n, a'$.
(2) For each $1 \leq i \leq n$, inject packets at rate r with the single edge route e_i in the time interval $[i, t_i]$.
(3) In the first $(S'+n)/r$ time steps of time interval $[1, 2S]$ inject $S'+n$ packets at rate r. The first n packets have path of length 1 (i.e., a only), and the rest have the path $a, f_1, \ldots, f_n, a'$. Observe that indeed $(S'+n)/r \leq 2S$, by the choice of n and S_0.

We first note that this is a rate-r adversary by Lemma 3.3. We now prove that $C(S', F_n)$ holds at time $2S+n$. First, observe that in each step of the interval $[1, 2S]$,

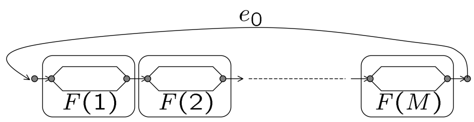

FIG. 3.2. *The graph used in the proof of Theorem* 3.17. *The edge between $F(i)$ and $F(i+1)$ is the egress of $F(i)$ and the ingress of $F(i+1)$, and it is part of both $F(i)$ and $F(i+1)$.*

a single packet crosses a. By the same arguments as those in the proofs of Claims 3.9, 3.10, and 3.11 (applied here to F_n, instead of F' there), we have that at time $2S+n$ there are S' packets in the buffers of edges $e_1, \ldots, e_n$, that none of these buffers is empty, and that the packets in e_i have remaining routes $e_i, e_{i+1}, \ldots, e_n, a'$. Next, consider a. After $2S$ time steps, all old packets leave a; after additional n time steps all packets with path of length 1 injected in step (3) disappear too, and therefore, at time $2S+n$, we have exactly S' packets in a, with remaining routes $a, f_1, \ldots, f_n, a'$, as required. ∎

We now show the existence of an adversary that replaces a queue of old packets with another (smaller) queue of fresh packets. To do this, we consider a graph of three edges in series, called a_0, a_1, and a_2. The routes that will be traversed by old packets will all end at a_0, and the fresh packets all start at the tail of a_2. (We use three edges instead of two so as to avoid cyclic routes in our final construction.)

LEMMA 3.16. *Suppose that at time τ there are S packets stored in the buffer of a_0 with remaining routes of length* 1. *Then for any $r > 0$ there exists a rate-r adversary such that at time $\tau + S + rS + r^2S$ there are r^3S packets stored in the buffer of a_2 and there are no other packets in the network. Moreover, all the packets stored in the buffer of a_2 were injected at the tail of a_2 after time $\tau + S$.*

Proof. Let us assume for convenience of notation that $\tau = 0$. Call the packets that exist in the network at time 0 *old packets*. The execution is as follows.

(1) In the time interval $[1, S]$, rS packets are injected at the tail of a_0. These packets have routes a_0, a_1, a_2. All these packets are queued behind the old packets, and they start to move only at time S.

(2) In the time interval $[S+1, S+rS]$, r^2S packets are injected at the tail of a_2. These packets mix with the packets that were injected in step (1). At time $S+rS$, there is a queue of r^2S packets waiting for a_2, and no other packets exist in the network.

(3) In the time interval $[S+rS, S+rS+r^2S]$, r^3S new packets are injected at the tail of a_2. These packets are queued behind the packets injected in steps (1) and (2).

Note that by time $S+rS+r^2S$, all packets from steps (1) and (2) are absorbed. ∎

We are now ready to prove our main result. Note that the assumption of a specific initial state does not restrict the generality of the statement (see, e.g., [4]).

THEOREM 3.17. *For every $\epsilon > 0$ there exists a graph G_ϵ, a rate-r adversary for $r = \frac{1}{2} + \epsilon$, and an initial configuration such that FIFO is unstable on G_ϵ under that adversary starting from that initial configuration.*

Proof. The graph is defined as follows. Let S_0 and n be as required by Lemma 3.6 for ϵ. Choose M such that $\frac{r^3(1+\epsilon)^M}{4} > 1$. The graph consists of F_n^M (i.e., M daisy-chained gadgets), with one additional edge called e_0 connecting the head of the egress edge of the last gadget in the chain ($F(M)$) to the ingress edge of the first gadget in the chain ($F(1)$). See Figure 3.2.

In the initial configuration, there are $S^* > 2S_0$ packets in the ingress edge of

$F(1)$, all with paths of length 1 (i.e., their paths are composed of the ingress of $F(1)$). The adversary is defined by an iterative construction that works as follows. Let $S_1 = S^*$.

(1) Apply the adversary of Lemma 3.15 to get a configuration where $C(S_2, F(1))$ holds for $S_2 \geq \frac{S_1}{2}(1+\epsilon)$.

(2) Apply the adversary of Lemma 3.13 to get a configuration where S_3 packets are stored in the egress of $F(M)$ for $S_3 \geq S_2 \frac{(1+\epsilon)^{M-1}}{2}$.

(3) Apply the adversary of Lemma 3.16 to the three-edge path that consists of the egress of $F(M)$, then e_0, and then the ingress of $F(1)$. This results in S_4 packets stored at the tail of the ingress of $F(1)$, all with paths of length 1, for $S_4 \geq r^3 S_3$. Let $S_1 \leftarrow S_4$, and go to step (1).

We first claim that the above construction is indeed a valid rate-r adversary. We claim inductively that the conditions that allow the construction of each adversary by Lemmas 3.15, 3.13, and 3.16 hold when these constructions are applied. We base the proof for each iteration of the adversary on the following condition, which we will prove by induction on time to hold at the start of each iteration. The condition is that when the iteration starts at time τ, then it holds that all the buffers in the network are empty except the buffer of the ingress of $F(1)$; that all packets in this buffer have routes of length 1; and that in time interval $[\tau_0 - \lceil 1/r \rceil, \tau]$ there were no injections of packets requiring any other edge in the network, where τ_0 is the earliest injection time of any packet residing in the buffer of the ingress of $F(1)$ at time τ. This condition clearly holds for time $\tau = 0$ when we start with the initial configuration as described above.

Now assume that the above condition holds at time τ when an iteration starts. Then the conditions of Lemma 3.15 hold. We therefore can apply the adversary of Lemma 3.15. The resulting situation is the situation as required by the conditions of Lemma 3.13. Observe that the adversary of Lemma 3.15 does not use any edge in the network beyond the edges of $F(1)$; therefore the conditions of that lemma on the nonuse of the edges of $F_n^M \setminus F(1)$ hold. We can therefore apply the adversary of Lemma 3.13. The resulting situation is the situation in the conditions of Lemma 3.16. Observe also that by Lemma 3.13 there are no other packets in the network at that time. We can now apply the adversary of Lemma 3.16. This results in a set of packets in the buffer of the ingress of $F(1)$. Note that now there are no other packets in the network. Further observe that all the packets at the ingress of $F(1)$ were injected at least S time steps into the activation of the adversary of Lemma 3.16 and that once they start being injected, no other packet is injected. Therefore it follows that no edge of the graph, except the ingress of $F(1)$, was used by the adversary after time $\tau_0 - \lceil 1/r \rceil$, where τ_0 is the earliest injection time of the packets at the ingress of $F(1)$. We therefore have that the condition for the start of the iteration holds again.

We note that no packet is rerouted more than M times: once in step (1) and at most $M-1$ times in step (2).

Finally, we show that S_1 grows unboundedly under this adversary. After step (1) is executed, we have that $S_2 \geq \frac{S_1}{2} \cdot (1+\epsilon)$. Hence, after step (2) we have that $S_3 \geq S_2 \frac{(1+\epsilon)^{M-1}}{2} \geq S_1 \frac{(1+\epsilon)^M}{4}$. Finally, after step (3), we have that the number of packets stored at the tail of the ingress edge of $F(1)$ is $S_4 \geq r^3 S_3 \geq S_1 \frac{r^3(1+\epsilon)^M}{4}$. By the choice of M we have that $S_4 > S_1$. ∎

4. Stability under low injection rates. In this section we show that FIFO, and in fact any greedy protocol, is stable if the injection rate is below some threshold.

We start with the case where the network is initiated with empty buffers. We later consider the case where the adversary starts the system with an arbitrary initial configuration of an arbitrary set of packets in the buffers.

We start with the case where the network starts with empty buffers. We prove that any network is stable with *any greedy* protocol in the face of a (w, r) adversary, if $r \leq 1/(d+1)$, where d denotes the length (in edges) of the longest path followed by any packet. In particular, we prove below that any packet stays in any one queue no more that $\lfloor wr \rfloor$ time steps. For a certain class of protocols, which includes the protocol FIFO, the bound can be improved to $1/d$.

THEOREM 4.1. *For any network, if the sequence of packets is injected by a (w, r) adversary, with $r \leq 1/(d+1)$, and the schedule is a greedy schedule, then no packet stays in the same buffer more than $\lfloor wr \rfloor$ time steps.*

Proof. We prove, by induction on t, that any packet that arrives at a buffer at time step t leaves this buffer by time $t + \lfloor wr \rfloor$.

The base of the induction is any $t \leq dwr+1$. Let p be a packet that arrives at the buffer at the tail of edge e at time $t \leq dwr+1$. Assume towards a contradiction that p is in the same buffer at the end of time step $t+\lfloor wr \rfloor$. This means that for each of the $\lfloor wr \rfloor$ time steps in $[t+1, t+\lfloor wr \rfloor]$ some other packet was sent over edge e (since we consider a *greedy* protocol). Therefore we can identify $\lfloor wr \rfloor + 1$ packets that require edge e and are injected into the network by the end of time step $t + \lfloor wr \rfloor - 1$ (these are the packet p itself and the $\lfloor wr \rfloor$ packets that were sent over e). Since $t \leq dwr+1$, we have $t + \lfloor wr \rfloor - 1 \leq (d+1)wr$. By the definition of the adversary the number of packets that require e and are injected by the end of any time step $t' \leq (d+1)wr$ is at most $\lceil (d+1)r \rceil \lfloor wr \rfloor$. Since we assume $r \leq 1/(d+1)$ this is at most $\lfloor wr \rfloor$. This is a contradiction to the fact that we identified $\lfloor wr \rfloor + 1$ packets.

We now prove the claim for any $t > dwr+1$. This is done based on the induction hypothesis that for any packet that arrives at some buffer at time $t' < t$, this packet leaves the buffer by time step $t' + \lfloor wr \rfloor$. Let p be a packet that arrives at the buffer at the tail of edge e at some time step t. Consider any packet that requires edge e and was injected by time step $t - d\lfloor wr \rfloor$. Using the induction hypothesis we know that such a packet left the buffer into which it was injected by time step $t - d\lfloor wr \rfloor + \lfloor wr \rfloor$, left the next buffer by time step $t - d\lfloor wr \rfloor + 2\lfloor wr \rfloor$, and left the ith buffer on its path by time step $t - d\lfloor wr \rfloor + i\lfloor wr \rfloor$. It therefore arrived at its destination by time step $t - d\lfloor wr \rfloor + d\lfloor wr \rfloor = t$ (since the length of its path is at most d, and all its "arrival times" are earlier than t, so the induction hypothesis holds). It follows that any packet that can delay packet p from going over edge e must be injected at time step $t - d\lfloor wr \rfloor + 1$ or later. Now assume towards a contradiction that packet p is still at the tail of edge e at the end of time step $t+\lfloor wr \rfloor$. That is, there are $\lfloor wr \rfloor$ other packets that crossed edge e in $[t+1, t+\lfloor wr \rfloor]$. As before, this identifies $\lfloor wr \rfloor + 1$ distinct packets that require edge e, are present in the network at the end of time step t or later, and are injected by time step $t+\lfloor wr \rfloor - 1$. However, we know that any packet injected by time step $t - d\lfloor wr \rfloor$ already left the network by the end of time step t. Therefore those $\lfloor wr \rfloor + 1$ packets must have been injected in $[t - d\lfloor wr \rfloor + 1, t + \lfloor wr \rfloor - 1]$. There are $\lfloor wr \rfloor (d+1) - 1$ time steps in this interval; therefore the number of packets that require e and can be injected during this interval is bounded by $\lceil (d+1)r \rceil \lfloor wr \rfloor$. Since $r \leq 1/(d+1)$ this is at most $\lfloor wr \rfloor$, a contradiction. ∎

For protocols where a packet arriving at a certain buffer at time t has priority over any packet injected after time t, we can relax the condition that $r \leq 1/(d+1)$ to be $r \leq 1/d$. Note that among such protocols are the protocols FIFO and LIS.

Specifically, we define the following concept.

Definition 4.2. *A* time priority *protocol is a greedy protocol under which a packet arriving at a buffer at time t has priority over any other packet that is injected after time t.*

For time priority protocols, we have the following.

Theorem 4.3. *For any network, if the sequence of packets is injected by a (w, r) adversary, with $r \leq 1/d$, and the protocol is a time priority protocol, then no packet stays in the same buffer more than $\lfloor wr \rfloor$ time steps.*

The proof of Theorem 4.3 is the same as the proof of Theorem 4.1 with one change applied at two places: in the present case, when assuming towards a contradiction that packet p is still in the same buffer at the end of time step $t+\lfloor wr \rfloor$ and identifying the packets that cause this delay, we know that those packets must have been injected no later than time step t (rather than time $t + \lfloor wr \rfloor - 1$). This is because packets injected after time step t will not delay packet p if the protocol is a time priority protocol. This allows us to prove the lemma with the relaxed condition that $r \leq 1/d$. For completeness we give below the full proof.

Proof. We prove that any packet that arrives at any buffer at time step t leaves this buffer by time step $t + \lfloor wr \rfloor$. The proof is by induction on t.

We prove the base of the induction for any $t \leq dwr$. Let p be a packet that arrives at the buffer at the tail of edge e at time $t \leq dwr$. Assume towards a contradiction that p is in the same buffer at the end of time step $t + \lfloor wr \rfloor$. This means that during the $\lfloor wr \rfloor$ time steps in $[t+1, t+\lfloor wr \rfloor]$ some other packet was sent over edge e (since we consider a *greedy* protocol). We can therefore identify $\lfloor wr \rfloor + 1$ packets that require edge e (these packets are the packet p itself and the $\lfloor wr \rfloor$ packets that were sent over e). These packets must have been injected into the system by the end of time step t; any packet injected after t will not delay p according to a time priority protocol. Now, by the definition of the adversary the number of packets that require e and are injected by the end of any time step $t \leq dwr$ is at most $\lceil dr \rceil \lfloor wr \rfloor$. Since we assume $r \leq 1/d$ this is at most $\lfloor wr \rfloor$. This is a contradiction to the fact that we identified $\lfloor wr \rfloor + 1$ packets.

We now prove the claim for any $t > dwr$. This is done based on the induction hypothesis that for any packet that arrives at some buffer at time $t' < t$, this packet leaves this buffer by time step $t' + \lfloor wr \rfloor$. Let p be a packet that arrives at the buffer at the tail of edge e at some time step t. Consider any packet that requires edge e and was injected by time step $t - d\lfloor wr \rfloor$. Using the induction hypothesis we know that such a packet left the buffer into which it was injected by time step $t - d\lfloor wr \rfloor + \lfloor wr \rfloor$, left the next buffer by time step $t - d\lfloor wr \rfloor + 2\lfloor wr \rfloor$, and left the ith buffer on its path by time step $t - d\lfloor wr \rfloor + i\lfloor wr \rfloor$. It therefore arrived at its destination by time step $t - d\lfloor wr \rfloor + d\lfloor wr \rfloor = t$ (since the length of its path is at most d, and all its "arrival times" are earlier than t, so the induction hypothesis holds). It follows that any packet that can delay packet p from going over edge e must be injected at time step $t - d\lfloor wr \rfloor + 1$ or later. Now assume towards a contradiction that packet p is still at the tail of edge e at the end of time step $t + \lfloor wr \rfloor$. That is, there are $\lfloor wr \rfloor$ other packets that crossed edge e in $[t + 1, t + \lfloor wr \rfloor]$. As before, this identifies $\lfloor wr \rfloor + 1$ distinct packets that require edge e and are injected by the end of time step t; any packet injected after t will not delay p since the protocol is a time priority protocol. However, we know that any packet injected by time step $t - d\lfloor wr \rfloor$ already left the network by the end of time step t. Therefore those $\lfloor wr \rfloor + 1$ packets must have been injected in $[t - d\lfloor wr \rfloor + 1, t]$. There are $\lfloor wr \rfloor d$ time steps in this interval; therefore the number of packets that require e and can be injected during this interval is bounded by $\lceil dr \rceil \lfloor wr \rfloor$. Since $r \leq 1/d$ this is at most $\lfloor wr \rfloor$, a contradiction. ∎

We now show that similar results hold when the adversary is allowed to initiate the system with an arbitrary set of packets in the network. In this case, the requirement for the rate is that it is less than (rather than at most) $1/(d+1)$ (or $1/d$ for time priority protocols). This follows from the fact that if an adversary starts the system with some initial set of packets and then injects packets as a (w, r) adversary, then the same sequence of packets can be given by an adversary that starts the system with an empty initial configuration and then injects packets as a (w^*, r^*) adversary for any $r^* > r$ and an appropriately chosen w^*. In the following we call an initial configuration an *S-initial-configuration*, for $S \geq 0$, if S is the maximum, over the edges $e \in E$, of the number of packets requiring e, in the initial configuration. We now give the following observation.

OBSERVATION 4.4. *Any sequence of packets given by a (w, r) adversary that starts with an S-initial-configuration can be given by a (w^*, r^*) adversary that starts with a 0-initial-configuration (i.e., with empty buffers) for any $r^* > r$ and $w^* = \lceil \frac{S+w+1}{r^*-r} \rceil$.*

Proof. We show that indeed a (w^*, r^*) adversary can inject the same sequence of packets without the need of an initial nonempty configuration. The new adversary will start with an empty configuration, will inject the packets of the initial state in time step 1, and will later inject in every time step t the same packets that the old adversary injected in time step $t-1$. By construction the new adversary starts with an empty configuration. It remains therefore to show that it is a valid (w^*, r^*) adversary.

To see that we show that in every consecutive w^* time steps the adversary injects at most $\lfloor w^* r^* \rfloor$ packets requiring any particular edge. We first note that

$$\lfloor w^* r^* \rfloor - w^* r > w^* r^* - 1 - w^* r = w^*(r - r^*) - 1 = \left\lceil \frac{S + w^* + 1}{r^* - r} \right\rceil (r - r^*) - 1 \geq S + w \ .$$

Therefore,

$$\lfloor w^* r^* \rfloor > S + w + w^* r \geq S + wr + w^* r \geq S + \left\lceil \frac{w^*}{w} \right\rceil \lfloor wr \rfloor \ .$$

We have two types of time intervals: one time interval that includes time step 1 (i.e., $[1, w^*]$) and any other time interval of w^* consecutive time steps. For the latter case, the new adversary injects in some time interval $[\tau, \tau + w^* - 1]$ the same packets as the old adversary injected in time interval $[\tau - 1, \tau - 1 + w^* - 1]$. For every edge e the maximum number of packets requiring e injected by the (w, r) adversary is at most $\lceil \frac{w^*}{w} \rceil \lfloor wr \rfloor$, so the new adversary does not violate its constraints. For time interval $[1, w^*]$, the number of packets requiring any particular edge given by the (w, r) adversary, either in the initial configuration or the first w^* time steps, is at most $S + \lceil \frac{w^*}{w} \rceil \lfloor wr \rfloor$. ∎

The following two corollaries now follow immediately from the above observation and from Theorems 4.1 and 4.3.

COROLLARY 4.5. *For any network, if the system is started with an S-initial-configuration, the sequence of packets is injected by a (w, r) adversary with $r < 1/(d+1)$ and the schedule is a greedy schedule, then no packet stays in the same buffer more than $\lfloor \lceil \frac{S+w+1}{\frac{1}{d+1} - r} \rceil \cdot \frac{1}{d+1} \rfloor$ time steps.*

COROLLARY 4.6. *For any network, if the system is started with an S-initial-configuration, the sequence of packets is injected by a (w, r) adversary with $r < 1/d$*

and the protocol is a time priority protocol, then no packet stays in the same buffer more than $\lfloor \lceil \frac{S+w+1}{\frac{1}{d}-r} \rceil \cdot \frac{1}{d} \rfloor$ *time steps.*

5. Conclusions. In this paper we show upper and lower bounds on the rates at which FIFO is stable. These results improve upon previous bounds [4, 11, 15]. We note that our lower bounds use shortest-paths (and hence noncircular) routes.

We also show that any greedy protocol is always stable against a (w, r) adversary for $r < 1/(d+1)$, where d is the length of the longest route used (or $r < 1/d$ for a certain class of protocols). Results in [7] show that the protocol FTG (and in fact also LIFO and NTS) can be unstable for arbitrarily low rates. The proofs there use a network and a set of paths such that in order to show that FTG is unstable for rate r, packets with paths of length $16/r$ are used. In view of these results, our bounds on r, in terms of d, are optimal up to a small constant factor. Furthermore, our stability results indicate that in order to show that FIFO can be unstable at arbitrarily low rates, one would need correspondingly long paths for the packets, as opposed to the (small) constant size networks (and hence constant size packet routes) used to prove previous results on the instability of FIFO.

The technique we use for the instability result, of constructing gadgets and chaining them, can be applied to various gadgets. For example, one can extract a gadget structure from the constructions of [4] or [11], compose them as in Theorem 3.17, and improve on the original bounds. Conceptually, our lower bound consists of two elements: the chain idea and a "good" gadget. We believe that this technique may lead to further improvements.

Appendix: Asymptotic bounds for Lemma 3.6. In this appendix we give asymptotic bounds for the parameters n and S_0 used in Lemma 3.6. Specifically, we show the following. Let $\epsilon > 0$ be given. We define the following quantities:

$$r = \frac{1}{2} + \epsilon \ , \tag{5.1}$$

$$R_i = \frac{1-r}{1-r^i} \quad \text{for all } i \geq 0 \ , \tag{5.2}$$

$$n = \max\left(\frac{\log\epsilon - 2}{\log r}, \quad 1 - \frac{1}{\log r}\right) \ , \tag{5.3}$$

$$S_0 = \max\left(2n, \quad \frac{n}{2(R_n - R_{n+1})}\right) \ . \tag{5.4}$$

We prove that $n = \Theta(\log \frac{1}{\epsilon})$ and $S_0 = \Theta(nr^{-n}) = \Theta(\frac{1}{\epsilon}\log\frac{1}{\epsilon})$ when $\epsilon \to 0^+$ (i.e., since all quantities are functions of ϵ, we may consider the case when $\epsilon \to 0^+$).

We remark that in what follows we do not attempt to get tight constant factors. We use only crude estimates that, however, allow us to prove tight asymptotic bounds.

We start with n. Note first that by (5.1), $\log\epsilon < \log r$, and hence $\log\epsilon - 2 < \log r - 1$. For $\epsilon < \frac{1}{2}$, we also have $\log r < 0$, and hence $\frac{\log\epsilon-2}{\log r} > 1 - \frac{1}{\log r}$. It follows from (5.3) that for $\epsilon < 1/2$, $n = \frac{\log\epsilon-2}{\log r}$. Moreover, for $0 < \epsilon < 1/\sqrt{2} - 1/2$, we have $1/2 < r < 1/\sqrt{2}$, and therefore

$$\frac{\log\epsilon - 2}{-1} \ < \ n \ < \ \frac{\log\epsilon - 2}{-1/2} \ .$$

The latter bounds are equivalent to

$$\log\frac{1}{\epsilon} + 2 < \ n \ < 2\log\frac{1}{\epsilon} + 4 \ , \tag{5.5}$$

and hence, for $\epsilon \to 0^+$, we have that

$$n = \Theta\left(\log \frac{1}{\epsilon}\right) . \tag{5.6}$$

We now consider S_0. To show the claim, we bound S_0 for $0 < \epsilon < 1/4$. We start by estimating the difference $R_n - R_{n+1}$. Using (5.2) we have

$$\begin{aligned} R_n - R_{n+1} &= \frac{1-r}{1-r^n} - \frac{1-r}{1-r^{n+1}} \\ &= \frac{(1-r)(1-r^{n+1}) - (1-r)(1-r^n)}{(1-r^n)(1-r^{n+1})} \\ &= \frac{1-r^{n+1}-r+r^{n+2}-1+r^n+r-r^{n+1}}{(1-r^n)(1-r^{n+1})} \\ &= r^n \cdot \frac{1+r^2-2r^{n+1}}{(1-r^n)(1-r^{n+1})} . \end{aligned} \tag{5.7}$$

Using (5.7), we bound $(R_n - R_{n+1})/r^n$ from both sides by constants as follows. For $0 < \epsilon < 1/4$ we have $\frac{1}{2} < r < \frac{3}{4}$, and by (5.5) we have $n > 4$. Hence

$$1 < 1 + r^2 - 2r^{n+1} \; < \; \frac{1+r^2-2r^{n+1}}{(1-r^n)(1-r^{n+1})} \; < \; \frac{1+r^2-2r^{n+1}}{\frac{1}{4}} < 8 . \tag{5.8}$$

Therefore, $R_n - R_{n+1} = \Theta(r^n)$ for $\epsilon \to 0^+$. Moreover, for $0 < \epsilon < 1/32$ we have by (5.7), (5.8), and (5.5) that $R_n - R_{n+1} < \frac{1}{4}$, and hence $2n < \frac{n}{2(R_n - R_{n+1})}$. It therefore follows from (5.4) that for $\epsilon \to 0^+$,

$$S_0 \; = \; \frac{n}{2(R_n - R_{n+1})} \; = \; \Theta(nr^{-n}) , \tag{5.9}$$

as desired. Finally, note that by (5.3) we have that for $\epsilon < 1/2$,

$$nr^{-n} = nr^{-\frac{\log \epsilon - 2}{\log r}} = 4n2^{-\log \epsilon} = \frac{4n}{\epsilon} . \tag{5.10}$$

Combining (5.6), (5.9), and (5.10), we conclude that $S_0 = \Theta\left(\frac{1}{\epsilon} \log \frac{1}{\epsilon}\right)$ for $\epsilon \to 0^+$.

REFERENCES

[1] W. Aiello, E. Kushilevitz, R. Ostrovsky, and A. Rosén, *Adaptive packet routing for bursty adversarial traffic*, J. Comput. System Sci., 60 (2000), pp. 482–509.

[2] C. Àlvarez, M. Blesa, and M. Serna, *Universal stability of undirected graphs in the adversarial queueing model*, in Proceedings of the 14th Annual ACM Symposium on Parallel Algorithms and Architectures, Winnepeg, MB, Canada, 2002, pp. 183–197.

[3] M. Andrews, *Instability of FIFO in session oriented networks*, in Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, CA, 2000, pp. 440–447.

[4] M. Andrews, B. Awerbuch, A. Fernández, J. Kleinberg, T. Leighton, and Z. Liu, *Universal stability results for greedy contention-resolution protocols*, J. ACM, 48 (2001), pp. 39–69.

[5] M. Andrews and L. Zhang, *The effects of temporary sessions on network performance*, in Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, CA, 2000, pp. 448–457.

[6] A. Borodin, *private communication*, 2001.

[7] A. Borodin, J. Kleinberg, P. Raghavan, M. Sudan, and D. Williamson, *Adversarial queuing theory*, J. ACM, 48 (2001), pp. 13–38.

[8] R. Bhattacharjee and A. Goel, *Instability of FIFO at arbitrarily low rates in the adversarial queueing model*, in Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, Cambridge, MA, 2003, pp. 160–167.

[9] R. Cruz, *A calculus for network delay, Part* I*: Network elements in isolation*, IEEE Trans. Inform. Theory, 37 (1991), pp. 114–131.

[10] R. Cruz, *A calculus for network delay, Part* II*: Network analysis*, IEEE Trans. Inform. Theory, 37 (1991), pp. 132–141.

[11] J. Díaz, D. Koukopoulos, S. Nikoletseas, M. Serna, P. Spirakis, and D. Thilikos, *Stability and non-stability of the FIFO protocol*, in Proceedings of the 13th ACM Symposium on Parallel Algorithms and Architectures, Crete, Greece, 2001, pp. 48–52.

[12] D. Gamarnik, *Stability of adversarial queues via fluid model*, in Proceedings of the 39th Annual Symposium on Foundations of Computer Science, Palo Alto, CA, 1998, pp. 60–70.

[13] D. Gamarnik, *Stability of adaptive and nonadaptive packet routing policies in adversarial queueing networks*, SIAM J. Comput., 32 (2003), pp. 371–385.

[14] A. Goel, *Stability of networks and protocols in the adversarial queuing model for packet routing*, Networks, 34 (2001), pp. 219–224.

[15] D. Koukopoulos, S. E. Nikoletseas, and P. G. Spirakis, *The Range of Stability for Heterogeneous and FIFO Queueing Networks*, Electronic Colloquium on Computational Complexity Report TR01-099, 2001. Available online from http://www.eccc.uni-trier.de/eccc.

[16] T. Leighton, B. Maggs, and S. Rao, *Universal packet routing algorithms*, in Proceedings of the 29th Annual Symposium on Foundations of Computer Science, White Plains, NY, 1988, pp. 256–269.

[17] Z. Lotker, B. Patt-Shamir, and A. Rosén, *New stability results for adversarial queuing*, in Proceedings of the 14th Annual ACM Symposium on Parallel Algorithms and Architectures, Winnipeg, MB, Canada, 2002, pp. 192–199.

[18] Z. Zhang, Z. Duan, and Y. T. Hou, *Fundamental trade-offs in aggregate packet scheduling*, in Proceedings of the 9th Annual IEEE International Conference on Network Protocols, Riverside, CA, 2001.