New Stability Results for Adversarial Queuing

Extended Abstract

Zvi Lotker
Dept. of Electrical Engineering
Tel Aviv University
Tel Aviv 69978, Israel
zvilo@eng.tau.ac.il

Boaz Patt-Shamir
Dept. of Electrical Engineering
Tel Aviv University
Tel Aviv 69978, Israel
boaz@eng.tau.ac.il

Adi Rosén*
Dept. of Computer Science
Technion
Haifa 32000, Israel
adiro@cs.technion.ac.il

ABSTRACT

We consider the model of "adversarial queuing theory" for packet networks introduced by Borodin et al. [6]. 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 no more 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 no more than $1/(d+1)$.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Packet-switching networks*; C.4 [**Computer Systems Organization**]: Performance of Systems

General Terms

Algorithms, Performance, Theory

Keywords

Adversarial queuing theory

1. INTRODUCTION

Recent years have seen a growing amount of work being concentrated on analyzing packet-switching networks under non-probabilistic scenarios rather than under probabilistic assumptions [6, 2, 4, 12, 10, 11, 1, 3]. Much of this work makes use of the model of "adversarial queuing theory" proposed by Borodin et al. [6]. The basic model can be briefly described as follows. Time proceeds in discrete steps. In each step, packets are injected into the system 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 link-buffer at the tail of that link. The behavior of the system in the basic model 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 there is 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 [6, 2]. The networks which are always stable have been named "universally stable" networks [6], and have been fully characterized [12]. 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 $r < 1$. For example, Longest-In-System (LIS) and Furthest-To-Go (FTG). Other natural protocols however are known not to be always stable, e.g., First-In-First-Out (FIFO), Nearest-To-Go (NTG), Last-In-First-Out (LIFO), and Furthest-From-Source (FFS) [2]. Furthermore, the protocol NTG (and FFS, and LIFO) exhibit the phenomenon of being unstable on certain networks even at arbitrary low rates [6]. One of the main interesting open problems is this area is to determine 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 $r > 0.85$ [2]. This bound was improved to 0.8357 by Díaz et al. [9], and more recently further improved to 0.749 by Koukopoulos et al. [13]. On the other hand, [9] showed an elaborate formula to calculate, for a 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, m, the length of the longest route used by any packet, d, and the maximum in-degree in the network α; their bound is at most $1/2dm\alpha$ for any network.

*Supported in part by The Center for Absorption in Science, State of Israel; the Fund for the Promotion of Research at the Technion; and the Technion V.P.R. Fund.

SPAA'02, August 10-13, 2002, Winnipeg, Manitoba, Canada.

The contribution of this paper is twofold. First, we show that FIFO can be unstable for any rate greater than 0.5, and that on the other hand, FIFO is always stable if the rate is not greater than $1/d$. Second, the stability proof shows in fact that *any* greedy policy is stable if the injection rate is at most $1/(d+1)$. This result refines the notion of "universal stability" of networks, and shows in fact that any network has a bound such that all greedy protocols are stable if the injection rate is below that bound.[1] We remark that our stability proofs 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 is never bigger than the maximal burst size.

Our instability proof entails new techniques that greatly simplify the analysis of the FIFO policy, and thus may lead to further improved results. In particular, we develop a technique that enables us to construct adversaries for some *acyclic* networks, and then compose these networks and adversaries to form a cyclic network and a single adversary, that together show instability.

Paper 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 0.5. 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 [6], defined as follows. The communication network is modeled by a directed graph $G = (V, E)$, and 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 system 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 sub-steps. In the first sub-step, one packet is sent from each non-empty buffer over its corresponding link. In the second sub-step, 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 sub-step.

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 only allow such protocols), in which a link cannot be idle in a time step if its buffer is non-empty in the first sub-step. The protocol *First-In-First-Out (FIFO)* selects the packets to be sent from a buffer in the same order as their arrival order to that buffer.

The injection of the packets into the network is modeled as being done by an *adversary*. Following [6], we use the following parameterized definition for the adversary.

Definition 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 maximal number of times any edge appears in Π^t is at most rw.

For our instability results we use a weaker adversary, that is not allowed to inject bursty traffic. We call this adversary a *rate r adversary* [2]: 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," that 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 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 can be translated to a queue of fresh packets in the ingress edge 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 [2, 9, 13].

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 re-routed so 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 2. 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 to the buffer (such as FIFO and LIFO), or on injection time (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 *non-predictive* in [14].)

First, we define formally the notion of new edges.

Definition 3. 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 system at time t.

[1]It was recently brought to our attention that previously it was known that if the rate of injection is bounded by $1/m$, then the system remains stable with any greedy protocol [5].

Let t^* be the minimal injection time of all packets in P. An edge e is *new to* P if e is not a member in any route 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 $1/2$, $\lceil 1/r \rceil \leq 2$ and hence this term is usually insignificant.

We can now state and prove our rerouting claim.

LEMMA 3.1. *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 system 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 $\mathcal{A}'$ are identical until time t.

(2) If $p \in P_0$, then its route under $\mathcal{A}'$ is $q_p e_p r'_p$.

(3) if $p \notin P_0$, then its route under $\mathcal{A}'$ is $q_p e_p r_p$.

PROOF. Define $\mathcal{A}'$ as follows. All packets have the same injection times they have in $\mathcal{A}$. 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) and (3) 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 minimal 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 require e is the equal to the number of packets injected in $[t_1, t_2]$ by $\mathcal{A}$ and 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 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 respect $P_0 \subseteq P(t)$ and time t^* is the minimum injection time of all packets in $P(t)$ (see Definition 3). For interval I' the number of packets injected in I' by $\mathcal{A}'$ and require e is at most the number of packets injected in I' by $\mathcal{A}$ and 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 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 3.1 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, so long as the number of reroutings is finite.

3.2 Gadgets and their adversaries

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

Definition 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 1 shows F_n^2.

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

Definition 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 non-empty, 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 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 so as to allow repeated rerouting, but essentially the idea is to have the condition $C(S, F)$ carry over from one gadget to the next.

LEMMA 3.2. *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 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 $R_i \stackrel{\text{def}}{=} \dfrac{1 - r}{1 - r^i}$, for $1 \leq i \leq n$. Note, for later reference, that for all i,

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

We first choose parameters under the constraints below:

$$n > \max\left(\frac{\log \epsilon - 2}{\log r}, 1 - \frac{1}{\log r}\right) \; ,$$

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

We remark that for small ϵ values, we get $n = \Theta(\log \frac{1}{\epsilon})$ and $S_0 = \Theta(r^{-n}) = \Theta(\frac{1}{\epsilon})$.

Let us assume, for simplicity of notation, that $\tau = 0$. We now specify the adversary that will create a situation where $C(S', F')$

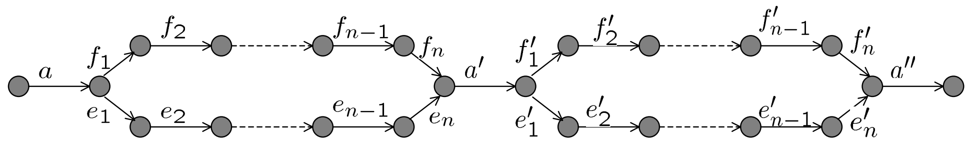

Figure 1: *The graph F_n^2: Two F_n gadgets glued together. The left gadget is called F and the right gadget is called F'. Edge a' is the egress of F and the ingress of F'.*

holds, for $S' = 2S(1 - R_n)$. In the adversary specification, as well as in the ensuing analysis, we ignore floors and ceilings for 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. [2, 9, 13]).

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 will show later that $0 \le X \le rS$).

We first note that this is a rate r adversary: Part 1 is justified by Lemma 3.1, 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.

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 Part 3 and 4 which cover disjoint time intervals. It remains to show that $0 \le X \le rS$.

CLAIM 3.3. *For every $r < 1$, we have $0 < X \le rS$.*

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 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 $rS - X > 0$. $\square$

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 3.4. *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. $\square$

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 3.5. *The following holds for any edge e'_i, $i \in \{1, \ldots, n\}$.*

(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 in 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 3.4. 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 Eq. 1 this is exactly R_{i+1}. 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

$$i + t_i + t_i(r + R_i - 1) = i + t_i(r + R_i) = i + 2S .$$

□

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

PROOF. By Claim 3.5, $2S \cdot R_n$ old packets cross a'' by time $2S + n$. On the other hand, by Claim 3.4, all the $2S$ old packets crossed a' by time $2S$. The claim follows. □

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

CLAIM 3.7. *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 $e'_i \dots e'_n a''$.*

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 3.5, 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 3.5, 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, $t_i \le t_{i+1}$ and $R_i \ge R_{i+1}$, and hence $Q_i \ge Q_{i+1}$ for $1 \le i < n$. 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 $Q_n \ge n$. 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 $S \ge S_0 > \frac{n}{2(R_n - R_{n+1})}$, we get that $Q_n \ge n$. □

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

CLAIM 3.8. *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 paths as required by $C(S', F')$. □

To conclude the proof of Lemma 3.2, 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 in 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 3.2. Consider the graph F_n^M that consists of a daisy chain of M F_n gadgets, where M is a parameter. Let the k'th gadget be denoted by $F(k)$, for $1 \le k \le M$. We now prove the following lemma.

LEMMA 3.9. *Let M be a positive integer, and consider the graph F_n^M. If for some time τ we have that all packets 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 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 $t > \tau$ we have that there are S' packets at the egress of F_n^M, for $S' \ge S(1+\epsilon)^{M-1}/2$.

PROOF. We first prove the following claim.

CLAIM 3.10. *Let $1 \le i \le n$. If at time τ we that all packets were injected after time τ_0, and*

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

PROOF. 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 $S_i \ge S(1 + \epsilon)^{i-1}$. Consider now the subgraph that consists of $F(i)$ and $F(i + 1)$. By the induction hypothesis, we may apply Lemma 3.2 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 $S' \ge S_i(1 + \epsilon) \ge S(1 + \epsilon)^i$. We note that the packets injected by $\mathcal{A}$ (as specified in Lemma 3.2) do not use any edge in $F(i + 2), \dots, F(M)$. 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 3.9, 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 3.9.

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 3.2.

LEMMA 3.11. *For any $\epsilon > 0$, let n and S_0 be as in Lemma 3.2. Let $S > S_0$ and let τ be a time step. Suppose that at time τ all the packets in the system are $2S$ packets stored in the ingress edge of F_n, and they were all injected after time τ_0 for some τ_0. If the 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 that $\tau = 0$. We use the notations and definitions of t_i and R_i from the proof of Lemma 3.2. We also define $S' = 2S(1 - R_n)$. The adversary is defined as follows.

(1) Change 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 3.1. We now prove that $C(S', F_n)$ holds at time $2S+n$. First, observe that in each step of the interval $[1, 2S]$, a single packet crosses a. By the same arguments as in the proofs of Claims 3.5, 3.6 and 3.7 (applied here to F_n, instead of F' there), we have that at time $2S+n$ there are S' packets in buffers of edges $e_1, \ldots, e_n$, 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 units 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.)

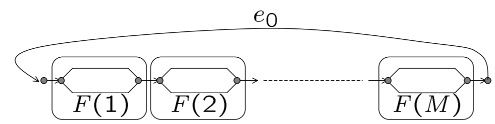

Figure 2: *The graph used in the proof of Theorem 3.13. The edge between $F(i)$ and $F(i+1)$ is the egress of $F(i)$ and the ingress of $F(i+1)$.*

LEMMA 3.12. *Suppose that at time 0 there are S packets stored in the buffer of a_0. Then for any $r > 0$ there exists a rate r adversary such that at time $S + rS + r^2S$ there are r^3S packets stored in the buffer of a_2 and there are no other packets in the system. Moreover, all the packets stored in the buffer of a_2 were injected at the tail of a_2 after time 0.*

PROOF. 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 $a_0a_1a_2$. 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 system.

(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., [2]).

THEOREM 3.13. *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 3.2 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 2.

In the initial configuration, there are $S^* > 2S_0$ packets in the ingress edge 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 3.11 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 3.9 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 3.12 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 $F(1)$, for $S_4 \geq r^3S_3$. Let $S_1 \leftarrow S_4$, and go to Step 1.

Observe that the adversary defined above is a rate r adversary. The main apparent difficulty in this is the packet rerouting done extensively throughout the construction. However, we argue that all these rerouting satisfy the condition of Lemma 3.1, i.e., packets are rerouted to edges that are new with respect to the set of all packets existing in the network in the time of rerouting. First, note that Step (2) is justified since the adversary of Step (1) does not inject any packet beyond the ingress of $F(1)$. Second, note that Step (3) is justified because there is no rerouting involved. Finally, note that starting Step (1) again is allowed since when Step (1) begins, all packets in the system were injected during the last invocation of Step (3), and they do not use any edge in $F(1), \ldots, F(M)$, and therefore the conditions of Lemma 3.11 are satisfied.

Furthermore, 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 $S_4 > S_1$, and we are done. □

4. STABILITY UNDER LOW INJECTION RATES

In this section 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 an (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 $t + \lfloor wr \rfloor$.

We prove the base of the induction for any $t \leq dwr + 1$. Let p be a packet that arrives to the buffer at the tail of edge e at time $t \leq dwr + 1$. Assume towards a contradiction that p is at 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). I.e., we identify $\lfloor wr \rfloor + 1$ packets that require edge e and are injected into the system 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 $\lfloor wr \rfloor$. 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 to 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 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 $t - d\lfloor wr \rfloor + 2\lfloor wr \rfloor$, etc. I.e., it arrived to 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 $t - wd + 1$ 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 in 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 6. 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 result.

THEOREM 4.2. *For any network, if the sequence of packets is injected by an (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 4.2 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 $r \leq 1/d$. Details can be found in the appendix.

We note that similar results can be proved for the case where the adversary is allowed to initiate the system with an arbitrary set of packets in the buffers. We omit the details from this abstract.

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 [2, 9, 13]. We note that our lower bounds use shortest-paths (and hence non-circular) routes.

The technique used in this paper, of constructing gadgets and chaining them, can be applied to various gadgets. For example, one can extract a gadget structure from the constructions of [2] or [9], compose them as in Theorem 3.13, 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.

We also show that any greedy protocol is always stable against a (w, r) adversary for $r \leq 1/(d+1)$, where d is the length of the longest route used (or $r \leq 1/d$ for a certain class of protocols). Results in [6] show that the protocol FTG (and in fact also LIFO and NTS) can be unstable for arbitrary low rates. The proofs there use a network and a set of paths such that 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 results indicate that in order to show that FIFO can be unstable for arbitrary low rates, one would need correspondingly large networks, as opposed to the (small) constant size networks used to prove previous results on the instability of FIFO.

The major question of whether the protocol FIFO can be unstable at arbitrary low rates remains open.

6. REFERENCES

[1] M. Andrews, Instability of FIFO in Session Oriented Networks. In *Proc. of the 11th SODA*, 2000.

[2] M. Andrews, B. Awerbuch, A. Fernández, J. Kleinberg, T. Leighton, and Z. Liu, "Universal Stability Results for Greedy Contention-Resolution Protocols", JACM 48(1), pp 39–69, 2001.

[3] M. Andrews, and L. Zhang, The Effect of Temporary Sessions on Network Performance. In *Proc. of the 11th SODA*, 2000.

[4] W. Aiello, E. Kushilevitz, R. Ostrovsky, A. Rosén. Adaptive packet Routing for Bursty Adversarial Traffic, *Journal of Computer and System Sciences*, Vol. 60, No. 3, pp. 482-509, 2000.

[5] A. Borodin, Personal Communication, 2001.

[6] A. Borodin, J. Kleinberg, P. Raghavan, M. Sudan, and D. Wiliamson, "Adversarial Queuing Theory", *JACM* 48(1), pp. 13-38, 2001.

[7] R. Cruz, "A Calculus for Network Delay, Part I: Network Elements in Isolation," *IEEE Transactions on Information Theory*, pp. 114–131, 1991.

[8] R. Cruz, "A Calculus for Network Delay, Part II: Network Analysis," *IEEE Transactions on Information Theory*, pp. 132–141, 1991.

[9] J. Díaz, D. Koukopoulos, S. Nokoleseas, M. Serna, P. Spirakis, and D. Thilikos, Stability and non-Stability of the FIFO Protocol. In *Proc. of Thirteenth ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, 2001.

[10] D. Gamarnik, Stability of Adversarial Queues via Fluid Model. In Proc. of the 39th FOCS, pp. 60–70, 1998.

[11] D. Gamarnik. Stability of Adaptive and Non-Adaptive Packet Routing Policies in Adversarial Queuing Networks. In *Proc. of the 31st STOC*, pp. 206–214, 1999.

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

[13] D. Koukopoulos, S. E. Nikoletseas, P. G. Spirakis, The Range of Stability for Heterogeneous and FIFO Queueing Networks. *Electronic Colloquium on Computational Complexity* (099). 2001.

[14] Tom Leighton, Bruce Maggs, and Satish Rao. Universal packet routing algorithms. In *29th Annual Symposium on Foundations of Computer Science*, pages 256–269, White Plains, NY, October 1988.

APPENDIX

In this appendix we prove Theorem 4.2.

We prove, by induction on t, that any packet that arrives at any buffer at time step t, leaves that buffer by time step $t + \lfloor wr \rfloor$. The base case is any $t \leq dwr + 1$. Let p be a packet that arrives at the buffer at the tail of edge e at time $t \leq dwr$. Assume towards a contradiction that p is at the same buffer at the end of time step $t + \lfloor wr \rfloor$. Since the protocol is greedy, this means that there are at least $\lfloor wr \rfloor + 1$ packets that use edge e in the time interval $[t, t + \lfloor wr \rfloor]$ (these are p, and the packets that were sent over e in the interval $[t, t + \lfloor wr \rfloor]$). However, since the protocol is time-priority, these packets must have been injected into the system by the end of time step t (otherwise they cannot delay p). By definition of the adversary, the number of packets that may use e and are injected by the end of time step t is at most $\lceil t/w \rceil \lfloor wr \rfloor \leq \lceil dr \rceil \lfloor wr \rfloor \leq \lfloor wr \rfloor$ since $r \leq 1/d$, a contradiction.

Now let $t > dwr$, and assume by induction that any packet that arrives at some buffer at time $t' < t$, leaves this buffer by time $t' + \lfloor wr \rfloor$. Let p be a packet that arrives at the buffer at the tail of edge e at some time t. Assume towards a contradiction that packet p is still at the tail of e at the end of time $t + \lfloor wr \rfloor$. Then there is a set of $\lfloor wr \rfloor + 1$ distinct packets that use edge e in the time interval $[t, t + \lfloor wr \rfloor]$. Since we have a time-priority protocol, all these packets were injected by the end of time step t. Moreover, we now prove that all these packets were injected at time $t - wd + 1$ or later. To see that, consider any packet q that was injected by time $t - d\lfloor wr \rfloor$. By induction, q left the first buffer on its path by time $t - d\lfloor wr \rfloor + \lfloor wr \rfloor$, left the next buffer by time $t - d\lfloor wr \rfloor + 2\lfloor wr \rfloor$, and so on. Hence q arrived at its destination by time $t - d\lfloor wr \rfloor + d\lfloor wr \rfloor = t$: since the length of its path is at most d, all its "arrival times" are earlier than t, so we may apply the induction hypothesis. Thus, all the $\lfloor wr \rfloor + 1$ packets that use e in the time interval $[t, t + \lfloor wr \rfloor]$ must have been injected in the interval $[t - d\lfloor wr \rfloor + 1, t]$. There are $\lfloor wr \rfloor d$ time steps in this interval, and therefore the number of packets that require e and can be injected during this interval in bounded by $\lceil dr \rceil \lfloor wr \rfloor$. Since $r \leq 1/d$ this is at most $\lfloor wr \rfloor$, a contradiction. This completes the proof of Theorem 4.2.