Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 16 (2003)

FIFO is Unstable at Arbitrarily Low Rates (Even in Planar Networks)*

Dimitrios Koukopoulos† *Marios Mavronicolas*‡ *Paul Spirakis*§

Abstract

We prove that the FIFO (*First-In-First-Out*) protocol is *unstable* in the standard model of *Adversarial Queueing Theory* [7] for arbitrarily low rates of packet injection. In order to prove this, we proceed as follows:

(1) We first consider the extension of the standard model to networks with *dynamic capacities,* which was introduced in [8]. We assume that each network link may arbitrarily take on a value in the two-valued integer set $\{1, C\}$ where $C > 4$ is an integer parameter (the *high* capacity). Here, for any $r > 0$, we construct a FIFO network (whose size is a small polynomial in $\frac{1}{r}$) which is unstable at any rate at least r in this setting.

(2) Then, we show how to *simulate* the construction in *(1)* in order to produce a FIFO network with all link capacities being now equal to C, which is also unstable at any rate at least r in this setting.

(3) Finally, we provide a simple simulation of the construction in *(2)* in order to produce a FIFO network (whose size is still a small polynomial in $\frac{1}{r}$) with all capacities being now equal to 1, which is similarly unstable. Since all capacities are equal to 1 in the standard model of Adversarial Queueing Theory [7], this implies our main result: FIFO is unstable in the standard model of Adversarial Queueing Theory model for arbitrarily low rates of packet injection.

We emphasize that all of our networks are *planar*; we allow though the paths of packets to have cycles of edges that can be repeated a *bounded* number of times.

Our result closes a major open problem, that of FIFO (in)stability, in the standard model of Adversarial Queueing Theory, which was already posed in the original pioneering work of Borodin *et al.* [7].

Note: Due to lack of space, many of our proofs are only sketched in this extended abstract; full proofs are included in a clearly marked Appendix that may be read at the discretion of the Program Committee.

*This work has been accepted for (electronic) publication in *Electronic Colloquium on Computational Complexity (ECCC),* ECCC Reports 2003, Technical Report TR03-016, accepted on March 24, 2003. This work has been partially supported by the IST Program of the European Union under contract numbers IST-1999-14186 (ALCOM-FT) and IST-2001-33116 (FLAGS), by funds from the Joint Program of Scientific and Technological Collaboration between Greece and Cyprus, and by funds for the promotion of research at University of Cyprus.

†**Contact Author.** Department of Computer Engineering and Informatics, University of Patras, Rion, 265 00 Patras, Greece, & Computer Technology Institute, 61 Riga Feraiou, P. O. Box 1122, 261 10 Patras, Achaia, Greece. Fax: +30-261-0-960442, Email: `Dimitrios.Koukopoulos@cti.gr`

‡Departemnt of Computer Science, University of Cyprus, 1678 Nicosia, Cyprus. Email: `mavronic@ucy.ac.cy`

§Department of Computer Engineering and Informatics, University of Patras, Rion, 265 00 Patras, Greece, & Computer Technology Institute, P. O. Box 1122, 261 10 Patras, Greece. Email: `spirakis@cti.gr`

ISSN 1433-8092

1 Introduction

Motivation, Framework and Statement of Contribution. We are interested in the behavior of *packet-switched* networks in which packets arrive dynamically at the *nodes,* and they are routed across the *links* at discrete time steps. Earlier research work on analyzing the performance of packet-switched networks has considered *probabilistic* assumptions and modeled packet injection as an oblivious randomized process; see, e.g., [12] for a wealth of results in this direction. Nevertheless, recent years have witnessed a vast amount of research work on the analysis of packet-switched networks under *non-probabilistic* assumptions. In particular, the model of *Adversarial Queueing Theory,* proposed in the pioneering work of Borodin *et al.* [7], has replaced probabilistic assumptions with *worst-case* ones; that work assumes an *adversary* $\mathcal{A}$ that controls packet generation and path determination in an adversarial way. In doing so, the adversary $\mathcal{A}$ may not exceed some specific *rate* r of packet injections.

The framework of our work is Adversarial Queueing Theory, henceforth abbreviated as AQT. More specifically, we are interested in issues of *stability* – will the number of packets in the network remain bounded at all times? The answer to this question may depend on the adversarial injection rate r, the network topology, and the *contention-resolution protocol* used when more than one packets need to cross a given link at the same time step. Taking these factors into account, say that a protocol P is *stable* on a network $\mathcal{N}$ [7] against an adversary $\mathcal{A}$ with injection rate r if there exists a (universal) constant B (that may depend on $\mathcal{N}$, $\mathcal{A}$ and r) such that the number of packets in the network (starting from an empty initial configuration) is bounded by B at all times. The major goal of our study is to establish stability properties of the common, FIFO (*First-In-First-Out*) protocol within the framework of AQT.

In their very early work introducing AQT, Borodin *et al.* [7, Section 8] already posed the open problem of whether FIFO may become unstable in the model of AQT when the injection rate of the adversary is arbitrarily low: "For packet routing, can FIFO be made unstable for arbitrarily small positive rates of injection in the adversarial model?" The principal contribution of our work is an *affirmative* answer to this fundamental open question from [7].

In our analysis, we consider a realistic extension due to Borodin *et al.* [8] to the standard model of AQT originally proposed in [7]; in this extension, the adversary is able to control (in addition) the *capacity* of each link, which is the rate at which the link forwards outgoing packets.[1] Henceforth, call this the *dynamic capacities* model of AQT (as opposed to the *standard* model of AQT [7]). Besides the inherent practical interest in this extended model, we chose to adopt it in our analysis as the host of an important immediate step in our proof that FIFO is unstable (at arbitrarily low rates of packet injection) in the standard model of AQT.

Summary of Contribution. We prove here that FIFO can be unstable at arbitrarily low rates of packet injections in the (standard) model of AQT. Our proof technique employs the dynamic capacities model of AQT [8] as the host of an important immediate step in our adversarial constructions. More specifically, our proof consists of the following steps:

1. We consider first a restriction of the dynamic capacities model where at each time step, each link capacity may take any one of two integer values 1 and $C > 4$, where C is a parameter called the *high capacity.* In this model, given any rate $r > 0$, we construct an adversary $\mathcal{A}$ and a FIFO network $\mathcal{N}_r$, (whose size is a small polynomial in $\frac{1}{r}$). We prove that $\mathcal{N}_r$ is unstable for the adversary $\mathcal{A}$ at all rates at least r.

2. We then modify the adversary $\mathcal{A}$ and the network $\mathcal{N}_r$ via appropriate changes to the network topology and to the paths of the injected packets. For any $r > 0$ this yields an adversary $\mathcal{A}'$ and a FIFO network $\mathcal{N}'_r$ in a way that all link capacities in the network $\mathcal{N}'_r$ are equal to C at all times.

[1]In the standard model of AQT [7], all link capacities are beyond the control of the adversary and are all equal to 1 at all times.

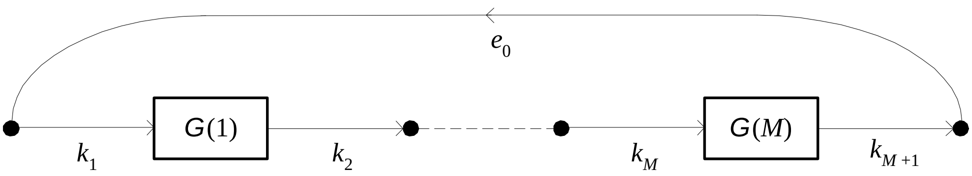

Figure 1: The network $\mathcal{N}_r$. The edge between $\mathcal{G}(i)$ and $\mathcal{G}(i+1)$ is the output edge of $\mathcal{G}(i)$ and the input edge of $\mathcal{G}(i+1)$.

We prove that $\mathcal{N}'_r$ is unstable when the adversary $\mathcal{A}'$ injects packets into the network at rate at least r.

3. Finally, we modify the adversary $\mathcal{A}'$ and the network $\mathcal{N}'_r$ via appropriate changes to the network topology and to the paths of the injected packets. This yields an adversary $\mathcal{A}''$ and a FIFO network $\mathcal{N}''_r$ in a way that all link capacities in the network $\mathcal{N}'_r$ are equal to 1 at all times. (Thus, this complies to the standard model of AQT [7].) We prove that $\mathcal{N}''_r$ is unstable when the adversary $\mathcal{A}''$ injects packets into the network at rate at least r. Hence, this implies that FIFO is unstable at arbitrarily low injection rates in the standard model of AQT [7], which is our main result.

We remark that our final network $\mathcal{N}''_r$ is *planar* (as also are the intermediate networks $\mathcal{N}_r$ and $\mathcal{N}'_r$). Note that existing large-scale communication platforms for computation and coordination, such as the *Internet,* are inherently planar; indeed, quantitative studies of graph-theoretic models for Internet topology, such as the one in [23], take this planarity feature into account by modelling the Internet using regular (planar) topologies (such as rings, trees and stars), other well-known planar topologies such as the ARPAnet or the NSFnet backbone, and other randomly generated planar topologies. Hence, we consider that our instability result is not only of mere mathematical interest, but it enjoys a more natural and intuitive appeal to the contemporary technology of communication networks.

We also note that the adversary $\mathcal{A}'$, and hence the adversary $\mathcal{A}''$ as well, are allowed to inject packets along *non-simple* paths with repeated edges. (This is allowed in the standard model of AQT [7, Section 3].) This feature is not far from practical reality where *control messages* and *daemons* are installed to periodically visit a particular sequence of network switches. Consider, for example, periodically sent daemon messages that collect performance information in the context of *implicit feedback* schemes for *closed-loop flow control* [21] from which changes in service rates are inferred. We emphasize that we take though special care in our adversarial constructions so that all employed non-simple paths are traversed a number of times that is *bounded* by a fixed function of C. (This is so as to comply to the definitions and the rate restriction of the standard model of AQT [7, Section 3].)

Finally, we believe that the introduction of *simulation* techniques in our proofs is a second major contribution to our work. They allow to prove instability in a model with a stronger adversary and inherit via simulation the instability to a model with a weaker adversary. Simulation techniques may be found useful in other instability proofs as well (both inside and outside AQT).

Details and Intuition for the Contribution.

The initial network $\mathcal{N}_r$ and adversary $\mathcal{A}$. The (planar) network $\mathcal{N}_r$ is a *chain* of M gadgets $\mathcal{G}_1, \ldots, \mathcal{G}_M$, for some integer $M > 0$ that will be appropriately chosen later (as a function of the rate r). Roughly speaking, each *gadget* has an *input edge* and an *output edge*. The output edge of each gadget is input to the next, except for the output edge of the last gadget which connects, via a single edge e_0, to the input of the first. (See Figure 1 for an illustration.) Each gadget is a (planar) subnetwork consisting of two consecutive long chains of edges and a few other edges shortcutting some segments of these chains.

The actions of the adversary are grouped into separate *phases* of suitable durations. We prove that the *network population* (i.e., number of packets in the network) increases from one phase to the next. In turn, each phase is split into a number of consecutive *subphases*. At the end of each subphase, packets are only queued in a single gadget; thus, each subphase corresponds to the movement of packets from one gadget to the next.

In each subphase, the adversary injects a first *flow* of packets into paths of edges of capacity C; simultaneously, the adversary delays a second packet flow (conflicting with the first) over paths of edges of capacity 1. Then, in the next consecutive time interval, the adversary does the opposite: it delays the first flow by changing all of its path edges to capacity 1, while it amplifies and expedites the second flow by both injecting more packets and changing all of its path edges to capacity C. This alternating throttling phenomenon, combined with suitably delaying older flows (via single edge injections) results to an increase of the number of packets in the network in each subphase, and this increase is inherited to the entire phase. This technique represents the essence and the main idea upon which our adversarial construction for the instability proof is built.

In some more detail, the instability proof (for the network $\mathcal{N}_r$ under the adversary $\mathcal{A}$) is inductive on the number of phases; thus, the induction step must assure that the initial conditions for one phase (induction hypothesis) must be reproduced for the next phase. In doing so, however, the proof allows for a small temporary decrease in the network population at the end of a phase. This decrease is counterbalanced by suitably selecting the number of subphases within each phase, which is taken equal to approximately the number of gadgets in the network.

The network $\mathcal{N}'_r$ and the adversary $\mathcal{A}'$. Overall, the instability proof for the network $\mathcal{N}'_r$ is a simulation of the instability proof of the network $\mathcal{N}_r$ under the adversary $\mathcal{A}$. The network $\mathcal{N}'_r$ is obtained from the network $\mathcal{N}_r$ by replacing all edges of the network $\mathcal{N}_r$ that underwent changes in their capacity (from C to 1) by appropriate (still planar) cyclic subnetworks. The role of these subnetworks is to simulate the drop of capacity from C to 1 via some sort of "busy-waiting" of some of the packets in the cycles. Thus, some packets are now assigned to non-simple paths. The proof of instability for the network $\mathcal{N}_r$ under the adversary $\mathcal{A}$ is a *simulation* proof; that is, we simulate the (already established) instability of the network $\mathcal{N}_r$ (under the adversary $\mathcal{A}$) over to the network $\mathcal{N}'_r$ (under the adversary $\mathcal{A}'$).

We prove that such cyclic paths only need to have a length bounded by a function of C; thus, packets assigned to non-simple paths only need to traverse these paths a number of times bounded by a function of C. However, in order to guarantee that the injection rate of the adversary is r, we must carefully account for multiple edge passings (due to the packets assigned to non-simple paths).

The network $\mathcal{N}''_r$ and the adversary $\mathcal{A}''$. Overall, the instability proof for the network $\mathcal{N}''_r$ under the adversary $\mathcal{A}''$ is a simulation of the instability proof for the network $\mathcal{N}'_r$ under the adversary $\mathcal{A}'$. Obtaining the network $\mathcal{N}''_r$ from the network $\mathcal{N}_r$ is intuitively very simple: we just replace each edge that ever took on capacity C by C parallel edges each of capacity 1. This replacement clearly simulates all instances where capacities were equal to C in a network where all capacities equal to 1 at all times. Some special care is needed though to handle groups of packets accumulated in an edge of capacity C (in the adversarial construction for the network $\mathcal{N}'_r$ under the adversary $\mathcal{A}'$) whose number is not a multiple of C. For the simulation to be "perfect" (so that none of them lack behind), we have to slightly modify the network $\mathcal{N}'_r$ by adding some small cycles of two edges each, and we also have to suitably adjust the packet paths assigned by the adversary $\mathcal{A}''$.

Related Work and Comparison.

Adversarial Queueing Theory and FIFO Instability. AQT was developed in the pioneering work of Borodin *et al.* [7] as a more robust model for packet generation and path determination in packet-switched networks. In recent years, AQT has received a lot of flourishing interest and attention; see, e.g., [1, 3, 5, 8, 15, 18, 19, 20, 22]. Extensions and variations to AQT have appeared in [2, 8]. Specifically, Aiello *et al.* [2] have considered an extension to adaptive path selection; Borodin *et al.* [8] introduced the dynamic capacities model of AQT considered in this work, and a related model of *slowdowns,* where links may temporarily "cease" without forwarding any packets. In work that predated AQT, Cruz [13, 14] designed a similarly adversarial "leaky-bucket" model of permanent sessions to capture the *burstiness* of inputs in communication networks. (Andrews [4] demonstrates instability of FIFO in the model of Cruz.)

The instability of FIFO in the standard model of AQT was first established by Andrews *et al.* [5, Theorem 2.10] for injection rates at least 0.85. Improved lower bounds of 0.8357 and 0.749 on threshold rates for FIFO instability were subsequently presented by Diaz *et al.* [15, Theorem 3] and by Koukopoulos *et al.* [18, Theorem 5.1]. The previous record of an injection rate for FIFO instability is due to Lotker *et al.* [20, Theorem 3.13]; in a breakthrough work, they presented a construction of a FIFO network which is unstable at any injection rate r larger than $\frac{1}{2}$. The construction of Lotker *et al.* [20, Section 3] uses gadgets and has inspired the use of gadgets in our construction as well; however, dropping the instability rate down to 0 has required the use of very different gadgets and more involved adversaries for them, which exploit their power of dynamically changing the capacities, than the ones used in [20]. This has resulted in a far more delicate analysis in our instability proof.

Independently of our work and at around the same time, Bhattacharjee and Goel [6] claimed a similar to (but different than) our result on the instability of FIFO in the standard model of AQT. Specifically, they present a different network construction and an adversary that leads to instability at any arbitrarily low injection rate. The construction of Bhattacharjee and Goel [6] applies to a slightly different version of the standard model of AQT [7] where packets are restricted to follow paths with no repeated edges. However, their network is highly *non-planar.* In comparison, our network $\mathcal{N}_r$ is *planar,* while its adversary generates paths with no repeated edges; however, this network is unstable (at arbitrarily low rate) in the dynamic capacities model of AQT. Moreover, our network $\mathcal{N}''_r$ is still planar, and it becomes unstable in the standard model of AQT; however, its adversary generates paths with repeated edges. Thus, our results are *incomparable* with the claimed result of Bhattacharjee and Goel [6], and none of them implies the other.

The dynamic capacities model of AQT employed in this work was introduced in the recent work of Borodin *et al.* [8]. That work introduced *stability-preserving transformations* to prove that some stability results carry through from the standard model of AQT to the dynamic capacities model of AQT. The simulation techniques we introduce in this work follow the opposite direction: they establish that the *instability* properties of FIFO are unfortunately inherited down as one goes from the dynamic capacities model of AQT to the standard model of AQT. Thus, our simulations may be viewed as *instability-preserving transformations.*

Relation to Bramson's Work [9, 10, 11]. Bramson studied FIFO stability (and instability) in two different probabilistic models. In [11], Bramson showed that FIFO is stable on *any* network and for all rates $r < 1$ if packets are injected by a Poisson process, and the time for a packet to traverse an edge is an i.i.d. exponential random variable (i.e., the network is *Kelly-type* [17]). Bramson [9, 10] also showed that FIFO can become unstable at arbitrarily low injection rates in a model of *job-shop scheduling.*

Superficially, it might seem that a minor modification to Bramson's techniques could imply our result (and the related claimed result of Bhattacharjee and Goel [6] as well). However, Bramson's constructions [9, 10, 11] as well as the open problem of the stability of FIFO in the standard model of Adversarial Queueing Theory (originally posed in [7]) have both been known for quite some time now, and no connection has so far been found. We will attempt to give some reasons why. In Bramson's constructions [9, 10], the same job can visit the same shop many times (actually, the number of times depends on execution time!), while it may receive a *different* mean processing time on each visit. If one tries to adapt Bramson's technique to a FIFO network, she immediately faces the technical problem of forcing different packets queued up at the same link to have different traversal times. Thus, the same link should appear to be of different speed to different packets. If one tries to implement this via additional injections, then the "extra" packets needed will violate the rate threshold (r) of the adversary in AQT.

Hence, a network like ours that can delay packets for arbitrary long durations seems inevitable. We also note that, unlike our construction, Bramson needs non-simple paths of *unbounded* length. We conclude that Bramson's results [9, 10, 11], although seminal for FIFO stability in Probabilistic Queueing Theory, do not lead to resolution of the problem of FIFO stability in AQT, which we solve here.

Road Map. The rest of this paper is organized as follows. Section 2 presents our model definitions. Section 3 shows the instability of FIFO in networks with dynamic capacities. Section 4 shows how to simulate the previous construction in order to prove FIFO instability in networks where all links have capacity C at all times; it also shows how to make all capacities equal to 1. We conclude, in Section 5, with a discussion of our results and an open problem.

2 The Model

Our model definitions are patterned after those in [7]; they are appropriately adjusted to allow edge capacities to vary arbitrarily, as put forward in the model of *dynamic capacities* [8, Section 2].

The *communication network* is modeled as a directed graph $\mathcal{N}$ with *nodes* and *edges*. Each node represents a *communication switch*; each edge 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 (possibly *non-simple*) directed path from *source* (first node on its route) to *destination* (last node on its route) in $\mathcal{N}$. At the time of injection, the packet is placed in the buffer of its source; the packet is absorbed when it reaches its destination.

The network proceeds in (global) discrete time steps. Edges can have different integer capacities, which may or may not vary over time. Denote $\mathsf{C}_e(t)$ the *capacity* of edge e at time step t. That is, we assume that edge e is capable of simultaneously transmitting up to $\mathsf{C}_e(t)$ packets at time t. The FIFO protocol gives priority to packets that arrived in the queue at the earliest time. Any packets that wish to travel along an edge e at a particular time step but are not sent wait in a queue for edge e. At each step, an *adversary* generates a set of requests. A *request* is a (possibly non-simple) *path* specifying the route followed by a packet. The system *configuration* at a given time step includes the packets (in order) in all queues of the network.

For any edge e of the network $\mathcal{N}$ and for any interval $\mathcal{T}$ of consecutive time steps, define $N(e, \mathcal{T})$ to be the number of paths injected by the adversary $\mathcal{A}$ during the time interval $\mathcal{T}$ that traverse edge e. Naturally, the contribution of each non-simple path traversing edge e to the number $N(e, \mathcal{T})$ is the number of times (or *multiplicity* in the terminology of Graph Theory) it traverses edge e.

For any constant r, where $0 < r \leq 1$, an *adversary $\mathcal{A}$ of rate r* is an adversary that injects packets subject to the following *load condition*: For every edge e and interval $\mathcal{T}$, $N(e, \mathcal{T}) \leq r \sum_{t \in \mathcal{T}} \mathsf{C}_\mathsf{e}(t)$. This load condition specifies the *dynamic capacities* model of AQT [8]. The special case where $\mathsf{C}_\mathsf{e}(t) = 1$ at all times t corresponds to the *standard* model of AQT [7].

In the adversarial constructions we present here for proving instability, we assume that there is a sufficiently large number of packets in the initial configuration. This will imply instability for networks with an *empty* initial configuration, as established by Andrews *et al.* [5, Lemma 2.9]. For simplicity, we will omit floors and ceilings from our analysis.

3 Unstable FIFO Network with Dynamic Capacities

Consider any integer parameter $C > 4$. For our purposes, it suffices to consider FIFO networks where for each edge e and time step t, $\mathsf{C}_e(t) \in \{1, C\}$. We prove:

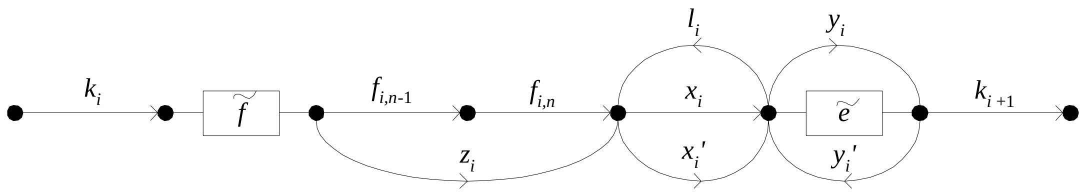

Figure 2: The gadget $\mathcal{G}(i)$

Theorem 3.1 *Given any $r > 0$, there exists a planar* FIFO *network $\mathcal{N}_r$ and an adversary $\mathcal{A}$ of rate r, that uses capacities 1 and $C > \frac{2}{r}$, such that the network $\mathcal{N}_r$ is unstable under the adversary $\mathcal{A}$.*

The Network $\mathcal{N}_r$.

The network $\mathcal{N}_r$ is a chain of M planar subnetworks (*gadgets*) $\mathcal{G}(1), \ldots, \mathcal{G}(M)$, with an extra edge e_0 connecting the *output edge* of $\mathcal{G}(M)$ to the *input edge* of $\mathcal{G}(1)$. (The parameter M will be determined later.) Thus, each *gadget* has a single input edge and a single output edge. The output edge of gadget $\mathcal{G}(i)$ is the input edge of gadget $\mathcal{G}(i+1)$. (See Figure 1 for an illustration.) The i^{th} gadget, $\mathcal{G}(i)$, $1 \leq i \leq M$, is a planar directed graph that consists of:

- An *input edge* k_i, and an *output edge* k_{i+1}.
- A chain of n edges $f_{i,j}$, where $1 \leq j \leq n$, that has as source the destination of the edge k_i and destination the source of the edge x_i, and an edge z_i that has common source with the edge $f_{i,n-1}$ and common destination with the edge $f_{i,n}$.
- Three parallel edges, two of which $x_i, x_i^{'}$ have common source and destination and one l_i with opposite source and destination to the other two edges.
- A chain of $n+1$ edges $e_{i,j}$, where $0 \leq j \leq n$, that has as source the destination of the edge x_i and destination the source of the edge k_{i+1} and two edges $y_i, y_i^{'}$, where the edge y_i has common source with the edge $e_{i,0}$ and common destination with the edge $e_{i,n}$, while the edge $y_i^{'}$ has opposite source and destination to the edge y_i.

Let $\tilde{\text{f}}$ be the path $f_{i,1}, \ldots, f_{i,n-2}$. Let $\tilde{\text{e}}$ be the path $e_{i,0}, \ldots, e_{i,n}$. Let f be the path $f_{i,1}, \ldots, f_{i,n}$. Let $e_i(j,k)$ be the path $e_{i,j}, \ldots, e_{i,k}$ $(k > j)$. Let $f_i(j,k)$ be the path $f_{i,j}, \ldots, f_{i,k}$ $(k > j)$. (See Figure 2 for an illustration.)

Our construction will define a sequence $\tilde{\tau}$ of time steps (τ, $\tau^{'}$, $\tau^{''}$,...) such that the packets of the system are all queued inside a single gadget at time step τ, and the number of these packets is $2s$. More specifically, let $\mathcal{G}(i)$ be the gadget associated with time τ. Then: *(i)* There are $2s$ packets, all queued in the queues $\tilde{\text{e}}$ and $x_i, x_i^{'}$, so that none of these queues is empty. *(ii)* No other queue in $\mathcal{G}(i)$ has any packets. *(iii)* The packets in queues $e_{i,j}$ $(0 \leq j \leq n)$ are required to traverse the path $e_i(j,n), k_{i+1}$ while the packets in queues $x_i, x_i^{'}$ are required to traverse the path y_i, k_{i+1}.

The Adversary.

We divide time into *phases*. We will demonstrate that the number of packets at the end of each phase is larger than at the beginning of the phase. This implies instability. Each phase consists of $M+2$ subphases. During each of the M subphases (*move* subphases), the packets move from gadget to gadget. The remaining two subphases (*connection* subphases) are used to reproduce the system configuration for the next phase (with packets that do not have previous history and populate the same subset of queues of the first gadget, as at the beginning of the previous phase).

The adversary, at the beginning of each subphase, assigns extensions of paths to the packets that are queued into the system. The path extension covers edges of the current gadget and some edges of the next gadget. (Call this *on-line path extension*[2].) We make sure that packets leaving a gadget are absorbed (finish) in the next subphase. So, the motion of packets is achieved by new injections. For the rest of the proof, assume that s_0, the number of packets in the initial configuration, satisfies $s_0 \geq 4nC^3$.

Lemma 3.2 *Consider any rate $r > 0$. If a packet set L of t packets is inserted into a chain of n edges (with unit capacities) in the first t steps of a time period of $t+n$ steps, wanting to traverse all the edges of the chain, then there is a sequence of adversarial injections of rate r such that: (i) The number of packets remaining in the system is $\leq rt$; (ii) all the edges have at least one packet; (iii) only the packets from L are queued into the chain queues at time step $t+n$.*

Proof: Consider the path of n queues e_i with $1 \leq i \leq n$ and the packet set L of t packets that require to traverse this path during a time period of $t+n$ steps. The adversary injects a set K_i of packets with $1 \leq i \leq n$ in queue e_i requiring to traverse only the queue e_i. K_i packets are injected in queue e_i with rate r at the time steps of the time interval $[i, i+t_i]$ where $t_i = \frac{t}{r+R_i}$ with $R_i = \frac{1-r}{1-r^i}$. Also, from the definition of R_i, we can estimate the quantity R_{i+1} recursively. Thus, $R_{i+1} = \frac{R_i}{R_i+r}$. Therefore, the number of injected K_i packets in queue e_i is $|K_i| = \frac{t}{r+R_i}$. Notice that the adversary does not inject the larger number of packets it can into each queue e_i but a smaller number. No packet arrives at queue e_i at times $[0, i]$. At times $[i+1, t+i]$ packets from the set L arrive in queue e_i with rate R_i where they are mixed with K_i packets. This has as a result, at the end of this period of $t+n$ time steps, the queues e_i not to contain any K_i packets, but only L packets.

In order to show that indeed packets from the set L arrive in queue e_i with rate R_i at time i we can use induction. For the basis of the induction, $i = 1$, packets arrive in e_i from set L with rate $R_1 = 1$. For the induction step let $i > 1$. The inductive hypothesis states that packets from the set L arrive in queue $e^{'}_{i-1}$ at rate R_{i-1} during $t+n$ time steps. However, the adversary injects into e_{i-1} a set K_{i-1} of $|K_{i-1}| = \frac{rt}{r+R_{i-1}}$ packets at the first $|t_1| = \frac{t}{r+R_{i-1}}$ time steps. Therefore, during the first t_1 time steps of t a number of $K_{i-1} + R_{i-1}t_1 = \frac{rt}{r+R_{i-1}} + R_{i-1}\frac{t}{r+R_{i-1}} = t$ packets in total mixed with each other, while all the other packets that belong to the set L are queued after them. Therefore, a number of L packets remain in e_i at the end of this time period, while all the K_{i-1} packets are absorbed. The number of L packets that leave e_{i-1} arriving in e_i has rate $\frac{R_{i-1}}{R_{i-1}+r}$ which is exactly R_i. Hence, the number of L packets that remain in the queues e_i is $t - \frac{R_n t}{R_n+r} = t - \frac{\frac{1-r}{1-r^n}t}{r+\frac{1-r}{1-r^n}} = rt\frac{1-r^n}{1-r^{n+1}} \leq rt$, as needed. ■

Let S be the sequence of adversarial injections of packets defined in the proof of Lemma 3.2. (These are only the injected K_i packets; see the proof.)

The population growth during a subphase. In the sequel, we assume that $n > \frac{\ln 2}{\ln \frac{1}{r}}$. We also denote by $2s_i$ the number of packets at time τ, i.e. at the beginning of a subphase. We let T_i be the time period of the i^{th} subphase. Let $|T_i| = \frac{2s_i}{C} + 2\frac{C-1}{C^2}s_i + n$. Let $\mathcal{N}_i$ be the network of the two chained gadgets $\mathcal{G}(i)$ and $\mathcal{G}(i+1)$. Define $\varepsilon = r - \frac{3C^2-1}{2C^3-2C}$. Since $r > \frac{2}{C}$ one can check that $\varepsilon = r - \frac{3C^2-1}{2C^3-2C} > 0$ whenever $C > 4$. We prove:

Lemma 3.3 *Let $r = \frac{3C^2-1}{2C^3-2C} + \varepsilon$ for any $\varepsilon > 0$. There is a suitable set of adversarial packet injections such that the packet population of $\mathcal{N}_i$ at the end of T_i is larger than at its beginning, and in fact $2s_{i+1} \geq 2s_i(1+\varepsilon)$.*

[2]We adopt a technique introduced by Lotket *et al.* [20, Lemma 3.1] that permits the adversary to specify paths in an on-line fashion. Thus, the adversary does not specify the complete path of the packets when they are injected, but constructs it in a succession of refinements.

Sketch of proof: Assume that *the initial system configuration* at time τ is as follows: (i) there are $2s_i$ packets (packet set S_0) in total that are queued in the queues $e_{i,0}, \ldots, e_{i,n}$ and $x_i, x_i^{'}$, none of which is empty. The packets in queues $e_{i,j}$ $(0 \leq j \leq n)$, have remaining routes $e_{i,j}, \ldots, e_{i,n}, k_{i+1}$ $f_{i+1,1}, \ldots, f_{i+1,n-2}, z_{i+1}, x_{i+1}^{'}, e_{i+1,0}, \ldots, e_{i+1,n}, k_{i+2}$, while the packets in queues $x_i, x_i^{'}$ require to traverse the edges y_i, k_{i+1} $f_{i+1,1}, \ldots, f_{i+1,n-2}, z_{i+1}, x_{i+1}^{'}, e_{i+1,0}, \ldots, e_{i+1,n}, k_{i+2}$, and (ii) no other queue in $\mathcal{G}(i)$ and no queue in $\mathcal{G}(i+1)$ has any packets.

For simplicity of notation we assume that $\tau = 0$. The adversary makes injections in a time period T_i with duration $|T_i| = \frac{2s_i}{C} + \frac{2(C-1)s_i}{C^2} + n$. During this time period all the edges of the network $\mathcal{N}_i$ have capacity C except some edges that have unit capacity in specific time intervals of T_i: *(i)* The edge $x_{i+1}^{'}$ has unit capacity in the time interval $[1, \frac{2s_i}{C} + n]$, *(ii)* the edge x_{i+1} has unit capacity in the time interval $[\frac{2s_i}{C} + n + 1, \frac{2s_i}{C} + \frac{2(C-1)s_i}{C^2} + n]$ and *(iii)* the edges $e_{i+1,0}, \ldots, e_{i+1,n}$ have unit capacity in the time interval $[1, \frac{2s_i}{C} + \frac{2(C-1)s_i}{C^2} + n]$.

Adversary's behavior. During this period the adversary makes the following injections: *(i)* During time interval $[1, \frac{2s_i}{C} + n]$ the adversary injects a set X of $|X| = \frac{2(C-1)rs_i}{C}$ packets in queue x_i requiring to traverse the edges $x_i, l_i, x_i^{'}, y_i, y_i^{'}, e_{i,0}, e_{i,1}, \ldots, e_{i,n},\ k_{i+1}, f_{i+1,1}, \ldots, f_{i+1,n},\ x_{i+1}, y_{i+1}, k_{i+2}$. *(ii)* During time interval $[n+1, \frac{2s_i}{C} + \frac{2(C-1)s_i}{C^2} + n]$ the adversary makes injections into the path $e_{i+1,1}, \ldots, e_{i+1,n}$, by using the set S of adversarial injections. *(iii)* During time interval $[\frac{2s_i}{C} + n + 1, \frac{2s_i}{C} + \frac{2(C-1)s_i}{C^2} + n]$ the adversary injects a set Y of $|Y| = \frac{2r(C-1)s_i}{C}$ packets in queue $x_{i+1}^{'}$ requiring to traverse the edges $x_{i+1}^{'}, y_{i+1}, k_{i+2}$ and a set Z of $|Z| = \frac{2r(C-1)s_i}{C^2}$ packets in queue x_{i+1} requiring to traverse the edge x_{i+1}.

Using a detailed calculation, we show that

$2s_{i+1} = 2s_i[\frac{(C-1)^2}{C^2} - \frac{C-1}{C^2+C} + 2r\frac{C-1}{C} + r\frac{2C-1}{C^2}\frac{1-r^n}{1-r^{n+1}}] + n - nr\frac{1-r^n}{1-r^{n+1}}$. Since $r \leq 1$, $\frac{1-r^n}{1-r^{n+1}} \leq 1$. So, $nr\frac{1-r^n}{1-r^{n+1}} \leq nr$. Therefore, $n - nr\frac{1-r^n}{1-r^{n+1}} \geq n - nr \geq 0$. Thus, $2s_{i+1} \geq 2s_i f(C,r)$, where

$$f(C,r) = \frac{(C-1)^2}{C^2} - \frac{C-1}{C^2+C} + 2r\frac{C-1}{C} + r\frac{2C-1}{C^2}\frac{1-r^n}{1-r^{n+1}}$$

Since $r\frac{2C-1}{C^2}\frac{1-r^n}{1-r^{n+1}} > 0$, it suffices to have $g(C,r) > 1$ where the function g is defined as $g(C,r) = \frac{(C-1)^2}{C^2} - \frac{C-1}{C^2+C} + 2r\frac{C-1}{C}$ because then $2s_{i+1} > 2s_i$. But, $g(C,r) = \frac{C^4-2C^3+C}{C^4+C^3} + r\frac{2C-2}{C}$. So, $g(C,r) > 1$, which implies that $r > \frac{3C^2-1}{2C^3-2C}$. Hence, there exists an $\varepsilon > 0$ such that $r = \frac{3C^2-1}{2C^3-2C} + \varepsilon$. Then we get, by substitution into $f(C,r)$, that $2s_{i+1} \geq 2s_i(1 + \frac{2(C-1)}{C} + \varepsilon) \geq 2s_i(1+\varepsilon)$, as needed. ∎

The population growth in a phase. Assume now that, at the beginning of a phase, queue k_1 contains $2s$ packets (configuration $\mathcal{C}_0$). Let $\mathcal{C}_1$ be the initial system configuration of the network $\mathcal{N}_r$. Let $\mathcal{C}_1$ have $2s^{'}$ packets. We prove:

Lemma 3.4 *There is a sequence of adversarial injections such that after a period of* $\frac{2s}{C} + \frac{2(C-1)}{C^2}s + n$ *steps the configuration* $\mathcal{C}_0$ *changes to* $\mathcal{C}_1$ *with* $2s^{'} \geq 2s(1+\varepsilon)$.

Sketch of proof: The proof is similar to the proof of Lemma 3.3, taking $i = 0$ and $s_i = s$. Thus, all the edges have capacity C except from the edges $x_1^{'}, x_1$ and $e_{1,0}, \ldots, e_{1,n}$ that change capacities from C to 1 in corresponding time intervals as the edges $x_{i+1}^{'}, x_{i+1}$ and $e_{i+1,0}, \ldots, e_{i+1,n}$ in Lemma 3.3. Also, the adversary makes packet injections with similar paths in corresponding time intervals as in Lemma 3.3 in some edges of the gadget $\mathcal{G}(1)$. However there is one exception concerning the first injection of packets (set X in Lemma 3.3). Here, the path assigned to these packets consists of the edges $k_1, f_{1,1}, \ldots, f_{1,n-2}$, z_1, y_1, k_2 as they are injected in queue k_1. The rest is the same. ∎

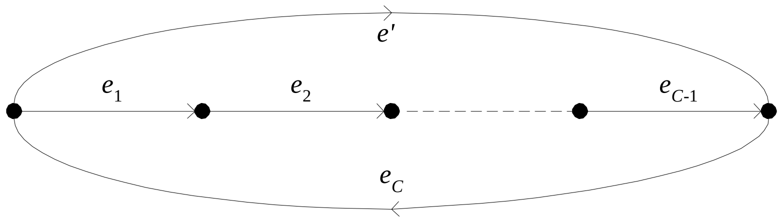

Figure 3: The network $\mathcal{B}$

Through an inductive application of Lemma 3.3, we prove:

Lemma 3.5 *Consider that there are $2s$ packets in gadget $\mathcal{G}(1)$ of $\mathcal{N}_r$ at time τ. Then, at the end of the M* move *subphases there are $2s^{'} > 2s(1+\varepsilon)^{M-1}$ packets in the system, all queued at the output edge k_{M+1} of $\mathcal{G}(M)$.*

Sketch of proof: The proof is split in two parts. The first part proves by induction on the number i of move subphases ($1 \leq i \leq M$) that if there are $2s$ packets in gadget $\mathcal{G}(1)$ of $\mathcal{N}_r$ at time τ, then at the end of the M move subphases there are $2s^{'} > 2s(1+\varepsilon)^{M-1}$ packets in the system, all queued in $\mathcal{G}(M)$. The second part proves that all $2s^{'}$ packets in $\mathcal{G}(M)$ at the end of the M move subphases are queued at the output edge k_{M+1} of $\mathcal{G}(M)$. ■

Lemma 3.6 *Assume that at time t, $2s$ packets are queued in queue k_{M+1}. There is a sequence of adversarial injections of rate r such that at time $t_1 = t + \frac{2s}{C} + r\frac{2s}{C} + r^2\frac{2s}{C}$ there are $r^3 2s$ packets in queue k_1, all being injected after time t.*

Sketch of proof: We take all edges equal to capacity C. The basic idea of the adversarial injections is to replace the packets arriving at the output edge k_{M+1} of the $\mathcal{G}(M)$ gadget with a number of packets in the edge k_1 that are injected in k_1 and they do not have previous history. This is done in three steps. In the first step, a set of packets X are injected requiring to traverse the edges k_{M+1}, e_0, k_1 that are blocked by the s packets in edge k_{M+1} at the beginning of this step. In the second step, a set Y of packets are injected requiring to traverse k_1 that mix with X. In the third step, a set of new packets are injected in k_1 that are blocked there by the previously injected packets that are absorbed. ■

We are now ready to prove Theorem 3.1. Put $2s$ packets in queue k_1 at the beginning of a phase. In the first subphase, the packets move to $\mathcal{G}(1)$ and there are $2s_1 \geq 2s(1+\varepsilon)$ remaining packets by Lemma 3.4. At the end of the M subphases, we will have in the queue k_{M+1} of $\mathcal{G}(M)$ a total of $2s_2 \geq 2s_1(1+\varepsilon)^{M-1}$ packets, by Lemma 3.5. Finally, at the beginning of the next phase we will have a total of $2s^{'} = 2s_3 \geq 2s_2 r^3$ packets, all new and in queue k_1 again, by Lemma 3.6.

Note that $s^{'} \geq r^3(1+\varepsilon)^M s$. For instability we need $s^{'} > s$. It must be then that $r^3(1+\varepsilon)^M > 1$, i.e. $M > \frac{3\ln(\frac{1}{r})}{\ln(1+\varepsilon)}$. This completes the proof of Theorem 3.1.

4 Unstable FIFO Network with Unit Capacities

Proposition 4.1 *Given any $r > 0$ there exists a planar* FIFO *network $\mathcal{N}_r^{'}$ and an adversary $\mathcal{A}$ of rate r, using only capacities $C > \frac{2}{r}$, so that the network $\mathcal{N}_r^{'}$ is unstable.*

Proof: For simplicity, we show how to simulate here a single packet flow passing via an edge e of network $\mathcal{N}_r$ which undergoes capacity changes in our previous adversarial construction. We replace every such edge e by the network $\mathcal{B}$ (See Figure 3). At intervals T for which $\mathsf{C}_e(t) = C \ \forall t \in T$, the packets paths are modified to use edge $e^{'}$. (Note that in $\mathcal{B}$ all edges are of capacity C). At intervals T for which $\mathsf{C}_e(t) = 1 \ \forall t \in T$, the packets paths are modified as follows: *(1)* The first C packets that traverse e, are now traversing $e_1, \ldots, e_{C-1}$ and exiting. The next $(C-1)C$ packets that traverse e (i.e. $C-1$ groups of C packets each) follow the path $e_1, \ldots, e_C, e_1$. *(2)* Apply (1) again to the packets queued at e_1 for the whole interval T in which $\mathsf{C}_e(t) = 1$ for all time steps $t \in T$.

Note that: *(i)* For any period $t > C^2$ only t packets exit $\mathcal{B}$ (since only C out of C^2 packets are exiting). *(ii)* We now show that the injection rate threshold is preserved when we count edge multiplicities in the paths of the packets. In the original dynamic network $\mathcal{N}_r$, for all time intervals T such that $\mathsf{C}_e(t) = 1$ for all time steps $t \in T$, we had $N(e,T) \leq r|T|$. Notice that each packet passes $e_1, \ldots, e_C$ at most C times. Hence in the modified network $\mathcal{N}_r'$ we have $N^{'}(e,T) \ \leq \ CN(e,T) \ \leq \ Cr|T| \ = r\sum_{t \in T} \mathsf{C}_e(t)$, since $\mathsf{C}_e(t) = C$ for all times $t \in T$ and all edges e of $\mathcal{N}_r'$. For multiple flows entering e, the idea is quite similar. ■

Combining our simulations, we now show, given any $r > 0$, how to construct a *planar* network $\mathcal{N}_r^{''}$ of all edge capacities equal to 1 and an adversary of rate r, so that $\mathcal{N}_r^{''}$ is unstable.

Theorem 4.2 *Given any $r > 0$, there exists a* planar FIFO *network $\mathcal{N}_r^{''}$ and an adversary $\mathcal{A}$ of rate r that uses unit capacities, such that the network $\mathcal{N}_r^{''}$ is unstable under the adversary $\mathcal{A}$.*

Sketch of proof: This is a simulation proof. Each edge with capacity C is replaced by C parallel, capacity 1, edges. The paths of every $x \leq C$ packets concurrently passing via such an edge e of the network $\mathcal{N}_r^{'}$ are modified so that each packet (in x) passes via a separate edge. The resulting network is $\mathcal{N}_r^{''}$, which clearly simulates $\mathcal{N}_r'$. Note that the size of $\mathcal{N}_r^{''}$ is polynomial in C and $\frac{1}{r}$.

■

5 Conclusions and Open Problem

A recent paper [16] reminded us of the following joke that was circulated in Italy around the 1920's:

> "Mussolini claims that the ideal citizen is intelligent, honest and fascist. Unfortunately, no one is perfect, which explains why everyone is either honest and fascist but not intelligent; intelligent and fascist but not honest; or honest and intelligent but not fascist."

Motivated by the original question of Borodin *et al.* [7] on the instability of FIFO at arbitrarily low injection rates, we faced the challenge of constructing, for any given $r > 0$, an unstable at rate r FIFO network which is planar, routes packets along simple paths, and uses only unit capacities. Theorem 3.1 finds such a network which is planar and routes packets along simple paths, but it uses though non-unit (equal to C) capacities. Theorem 4.2 finds another such network which is planar and uses only unit capacities, but this routes packets along non-simple paths. Finally, Bhattacharjee and Goel [6, Theorem 5.4] have found a third such network which routes packets along simple paths and it uses only unit capacities, but, unfortunately, it is not planar. The main question left open by our work is whether there exists, for any given $r > 0$, an *ideal,* unstable at r, FIFO network, which is planar, routes packets along simple paths, and uses only unit capacities.

References

[1] M. Adler and A. Rosén, "Tight Bounds for the Performance of Longest-in-System on DAGs," *Proceedings of the 19th Annual Symposium on Theoretical Aspects of Computer Science,* H. Alt and A. Ferreira eds., pp. 88–99, Vol. 2285, Lecture Notes in Computer Science, Springer-Verlag, Antibes–Juan Les Pins, France, March 2002.

[2] W. Aiello, E. Kushilevitz, R. Ostrovsky and A. Rosén, "Adaptive Packet Routing for Bursty Adversarial Traffic," *Proceedings of the 30th Annual ACM Symposium on Theory of Computing,* pp. 359–368, May 1998.

[3] C. Alvarez, M. Blesa, and M. Serna, "Universal Stability of Undirected Graphs in the Adversarial Queueing Model," *Proceedings of the 14th Annual ACM Symposium on Parallel Algorithms and Architectures,* pp. 183–197, August 2002.

[4] M. Andrews, "Instability of FIFO in Session-Oriented Networks," *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms,* pp. 440–447, January 2000.

[5] M. Andrews, B. Awerbuch, A. Fernandez, J. Kleinberg, T. Leighton and Z. Liu, "Universal Stability Results for Greedy Contention-Resolution Protocols," *Journal of the ACM,* Vol. 48, No. 1, pp. 39–69, January 2001.

[6] R. Bhattacharjiee and A. Goel, "Instability of FIFO at Arbitrarily Low Rates in the Adversarial Queueing Model," Technical Report no. 02-776, Department of Computer Science, University of Southern California, November 2002.

[7] A. Borodin, J. Kleinberg, P. Raghavan, M. Sudan and D. Williamson, "Adversarial Queueing Theory," *Journal of the ACM,* Vol. 48, No. 1, pp. 13–38, January 2001.

[8] A. Borodin, R. Ostrovsky and Y. Rabani, "Stability Preserving Transformations: Packet Routing Networks with Edge Capacities and Speeds," *Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms,* pp. 601–610, January 2001.

[9] M. Bramson, "Instability of FIFO Queueing Networks," *Annals of Applied Probability,* Vol. 4, No. 2, pp. 414–431, 1994.

[10] M. Bramson, "Instability of FIFO Queueing Networks with Quick Service Times," *Annals of Applied Probability,* Vol. 4, No. 3, pp. 693–718, 1994.

[11] M. Bramson, "Convergence to Equilibria for Fluid Models of FIFO Queueing Networks," *Queueing Systems: Theory and Applications,* Vol. 22, pp. 5–45, 1996.

[12] H. Chen and D. D. Yao, *Fundamentals of Queueing Networks,* Springer, 2000.

[13] R. L. Cruz, "A Calculus for Network Delay. Part I: Network Elements in Isolation," *IEEE Transactions on Information Theory,* Vol. 37, pp. 114–131, 1991.

[14] R. L. Cruz, "A Calculus for Network Delay. Part II: Network Analysis," *IEEE Transactions on Information Theory,* Vol. 37, pp. 132–141, 1991.

[15] J. Diaz, D. Koukopoulos, S. Nikoletseas, M. Serna, P. Spirakis and D. Thilikos, "Stability and Non-Stability of the FIFO Protocol," *Proceedings of the 13th Annual ACM Symposium on Parallel Algorithms and Architectures,* pp. 48–52, July 2001.

[16] M. Herlihy, N. Shavit and O. Waarts, "Linearizable Counting Networks," *Distributed Computing,* Vol. 9, No. 4, pp. 193–203, 1996.

[17] F. P. Kelly, *Reversibility and Stochastic Networks,* Wiley, 1979.

[18] D. Koukopoulos, M. Mavronicolas, S. Nikoletseas and P. Spirakis, "On the Stability of Compositions of Universally Stable, Greedy, Contention-Resolution Protocols," *Proceedings of the 16th International Symposium on DIStributed Computing,* D. Malkhi ed., pp. 88–102, Vol. 2508, Lecture Notes in Computer Science, Springer-Verlag, Toulouse, France, October 2002.

[19] D. Koukopoulos, M. Mavronicolas, S. Nikoletseas and P. Spirakis, "The Impact of Network Structure on the Stability of Greedy Protocols," *5th Italian Conference on Algorithms and Complexity,* Rome, Italy, May 2003. Accepted for publication.

[20] Z. Lotker, B. Patt-Shamir and A. Rosén, "New Stability Results for Adversarial Queueing," *Proceedings of the 15th Annual ACM Symposium on Parallel Algorithms and Architectures,* pp. 192–199, August 2002.

[21] K. K. Ramakrishnan and R. Jain, "A Binary Feedback Scheme for Congestion Avoidance in Computer Networks," *ACM Transactions on Computer Systems,* Vol. 8, No. 2, pp. 49–56, 1990.

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

[23] E. W. Zegura, K. L. Calvert and M. J. Donahoo, "A Quantitative Comparison of Graph-Based Models for Internet Topology," *IEEE/ACM Transactions on Networking,* Vol. 3, No. 6, pp. 770–783, December 1997.

Appendix

A Proof of Lemma 3.3

Consider the network $\mathcal{N}(i)$ of the two chained gadgets $\mathcal{G}(i)$, $\mathcal{G}(i+1)$. Assume that *the initial system configuration* at time τ is as follows: (i) there are $2s_i$ packets (packet set S_0) in total that are queued in the queues $e_{i,0}, \dots, e_{i,n}$ and $x_i, x_i^{'}$, none of which is empty. The packets in queues $e_{i,j}$ $(0 \le j \le n)$, have remaining routes $e_{i,j}, \dots, e_{i,n}, k_{i+1}$ $f_{i+1,1}, \dots, f_{i+1,n-2}, z_{i+1}, x_{i+1}^{'}, e_{i+1,0}, \dots, e_{i+1,n}, k_{i+2}$, while the packets in queues $x_i, x_i^{'}$ require to traverse the edges y_i, k_{i+1} $f_{i+1,1}, \dots, f_{i+1,n-2}, z_{i+1}, x_{i+1}^{'}, e_{i+1,0}, \dots, e_{i+1,n}, k_{i+2}$, and (ii) no other queue in $\mathcal{G}(i)$ and no queue in $\mathcal{G}(i+1)$ has any packets.

For simplicity of notation we assume that $\tau = 0$. The adversary makes injections in a time period T_i with duration $|T_i| = \frac{2s_i}{C} + \frac{2(C-1)s_i}{C^2} + n$. During this time period all the edges of the network $\mathcal{N}_i$ have capacity C except some edges that have unit capacity in specific time intervals of T_i: (i) The edge $x_{i+1}^{'}$ has unit capacity in time interval $[1, \frac{2s_i}{C} + n]$, (ii) the edge x_{i+1} has unit capacity in time interval $[\frac{2s_i}{C} + n + 1, \frac{2s_i}{C} + \frac{2(C-1)s_i}{C^2} + n]$ and (iii) the edges $e_{i+1,0}, \dots, e_{i+1,n}$ have unit capacity in time interval $[1, \frac{2s_i}{C} + \frac{2(C-1)s_i}{C^2} + n]$.

Adversary's behavior. During this period the adversary makes the following injections:

- During time interval $[1, \frac{2s_i}{C} + n]$ the adversary injects a set X of $|X| = \frac{2(C-1)rs_i}{C}$ packets in queue x_i requiring to traverse the edges $x_i, l_i, x_i^{'}, y_i, y_i^{'}, e_{i,0}, e_{i,1}, \dots, e_{i,n}$, $k_{i+1}, f_{i+1,1}, \dots, f_{i+1,n}$, $x_{i+1}, y_{i+1}, k_{i+2}$.
- During time interval $[n + 1, \frac{2s_i}{C} + \frac{2(C-1)s_i}{C^2} + n]$ the adversary makes injections into the path $e_{i+1,1}, \dots, e_{i+1,n}$, by using the set S of adversarial injections.
- During time interval $[\frac{2s_i}{C} + n + 1, \frac{2s_i}{C} + \frac{2(C-1)s_i}{C^2} + n]$ the adversary injects a set Y of $|Y| = \frac{2r(C-1)s_i}{C}$ packets in queue $x_{i+1}^{'}$ requiring to traverse the edges $x_{i+1}^{'}, y_{i+1}, k_{i+2}$ and a set Z of $|Z| = \frac{2r(C-1)s_i}{C^2}$ packets in queue x_{i+1} requiring to traverse the edge x_{i+1}.

Evolution of the system configuration. During time interval $[1, \frac{2s_i}{C} + \frac{2(C-1)s_i}{C^2} + n]$ the S_0 packets traverse their path. The first packets of set S_0 arrive in queue $x_{i+1}^{'}$ after the first n steps of this time interval as S_0 packets have to traverse the chain of edges $k_{i+1}, f_{i+1,1}, \dots, f_{i+1,n-2}, z_{i+1}$ that have capacity C. During time interval $[n + 1, \frac{2s_i}{C} + n]$ the S_0 packets are delayed in queue $x_{i+1}^{'}$ due to its unit capacity. Therefore, a number of $|S_1| = \frac{2(C-1)s_i}{C}$ packets remain in queue $x_{i+1}^{'}$ at time step $\frac{2s_i}{C} + n$, while $|S_2| = \frac{2s_i}{C}$ packets traverse the edge $x_{i+1}^{'}$ towards $e_{i+1,0}, \dots, e_{i+1,n}, k_{i+2}$. At the rest $\frac{2(C-1)s_i}{C^2}$ time steps of the time interval $[1, \frac{2s_i}{C} + \frac{2(C-1)s_i}{C^2} + n]$ the S_1 packets traversing their path arrive in the queue $e_{i+1,0}$. From S_1 packets, a set S_3 of $|S_3| = 2s_i\frac{(C-1)^2}{C^2}$ packets remain queued in queue $e_{i+1,0}$ at the end of T_i due to its unit capacity. Therefore, $|S_4| = \frac{2(C-1)s_i}{C^2}$ packets from S_1 packets can traverse the edge $e_{i+1,0}$ in $\frac{2(C-1)s_i}{C^2}$ time steps. Hence, during time interval $[n + 1, \frac{2s_i}{C} + \frac{2(C-1)s_i}{C^2} + n]$ the number of packets, which arrive in the path $e_{i+1,1}, \dots, e_{i+1,n}$ wanting to traverse it, is $|S_5| = |S_2| + |S_4|$ packets. From these packets a set H of $|H| = n$ packets arrive in queue $e_{i+1,1}$ at the last n steps of period T_i. By using the set S of adversarial injections (defined in the proof of Lemma 3.2), H packets remain queued in $e_{i+1,1}$ and a number of $|S_7| = [2s_i\frac{2C-1}{C^2} - n]\frac{r-r^{n+1}}{1-r^{n+1}}$ packets are preserved in queues $e_{i+1,1}, \dots, e_{i+1,n}$ such that all these queues are not empty at the end of T_i.

The X packets that are injected in queue x_i during time interval $[1, \frac{2s_i}{C} + n]$ traverse their path till queue k_{i+1} where they are delayed by S_0 packets till time step $\frac{2s_i}{C}$ because S_0 packets need $\frac{2s_i}{C}$ time steps to traverse the edge k_{i+1} due to its capacity C. During the rest n time steps, all the S_0 packets traverse the path $f_{i+1,1}, \ldots, f_{i+1,n-2}, z_{i+1}$ along with nC packets from set X, the first of which are queued in queue $f_{i+1,n}$ at time step $\frac{2s_i}{C} + n$. During the rest time steps of period T_i, all X packets traverse their path arriving in queue x_{i+1} due to the capacity C of the edges $k_1, f_{1,1}, \ldots, f_{1,n}$ in its path. In queue x_{i+1}, X packets are mixed with Z packets. This mixing along with the unit capacity of edge x_{i+1} results in the delay (in x_{i+1}) of a portion X' of $|X'| = 2s_i[r\frac{C-1}{C} - \frac{C-1}{C^2+C}]$ packets from the X packets. All the Y packets that are injected in queue x'_{i+1} during time interval $[\frac{2s_i}{C} + n + 1, \frac{2s_i}{C} + \frac{2(C-1)s_i}{C^2} + n]$ are delayed by the S_1 packets in x'_{i+1} due to the capacity C of x'_{i+1}.

At the end of period T_i, the number of packets in queues $e_{i+1,j}$ $(0 \leq j \leq n)$ that have remaining routes $e_{i+1,j}, \ldots, e_{i+1,n}, k_{i+2}$ and in queues x_{i+1}, x'_{i+1} requiring to traverse the edges y_{i+1}, k_{i+2} are $2s_{i+1} = |X'| + |Y| + |S_3| + |H| + |S_7|$. Substituting the corresponding estimated quantities of packets in this equation we take

$2s_{i+1} = 2s_i[\frac{(C-1)^2}{C^2} - \frac{C-1}{C^2+C} + 2r\frac{C-1}{C} + r\frac{2C-1}{C^2}\frac{1-r^n}{1-r^{n+1}}] + n - nr\frac{1-r^n}{1-r^{n+1}}$.

Since $r \leq 1$, we take $\frac{1-r^n}{1-r^{n+1}} \leq 1$. So, $nr\frac{1-r^n}{1-r^{n+1}} \leq nr$. Therefore, $n - nr\frac{1-r^n}{1-r^{n+1}} \geq n - nr \geq 0$. Thus, $2s_{i+1} \geq 2s_i f(C, r)$ where

$$f(C,r) = \frac{(C-1)^2}{C^2} - \frac{C-1}{C^2+C} + 2r\frac{C-1}{C} + r\frac{2C-1}{C^2}\frac{1-r^n}{1-r^{n+1}}$$

Since $r\frac{2C-1}{C^2}\frac{1-r^n}{1-r^{n+1}} > 0$ it suffices to prove $g(C,r) > 1$, where the function g is defined as

$$g(C,r) = \frac{(C-1)^2}{C^2} - \frac{C-1}{C^2+C} + 2r\frac{C-1}{C}$$

because then $2s_{i+1} > 2s_i$. But, $g(C,r) = \frac{C^4-2C^3+C}{C^4+C^3} + r\frac{2C-2}{C}$. So clearly $g(C,r) > 1$, which implies that $r > \frac{3C^2-1}{2C^3-2C}$. But $r = \frac{3C^2-1}{2C^3-2C} + \varepsilon$, where $\varepsilon > 0$. Then we get, by substitution into $f(C,r)$, that

$$\begin{aligned} 2s_{i+1} &\geq 2s_i(1 + \frac{2(C-1)}{C} + \varepsilon) \\ &\geq 2s_i(1+\varepsilon)\,. \end{aligned}$$

B Proof of Lemma 3.4

Consider the network $\mathcal{N}_r$ in Figure 1. In the initial system configuration $\mathcal{C}_0$ there is a set S_0 of $|S_0| = 2s$ packets queued in the queue k_1. We will show that there is a sequence of adversarial injections such that, after a period of $|T| = \frac{2s}{C} + \frac{2(C-1)}{C^2}s + n$ steps, the configuration $\mathcal{C}_0$ changes to $\mathcal{C}_1$ with $2s' \geq 2s(1+\varepsilon)$.

During time period T all the edges of the network $\mathcal{N}_r$ have capacity C except some edges that have unit capacity in specific time intervals of T. These edges and time intervals of T where they have unit capacity are similar to Lemma 3.3 taking $i = 0$, $s_i = s$ and $T_i = T$.

Adversary's behavior. During this period the adversary makes a suitable set of packet injections. These injections are similar to Lemma 3.3 taking $i = 0$ and $s_i = s$. The only difference is in the path of the packets of set X as they are injected in queue k_1 requiring to traverse the edges $k_1, f_{1,1}, \ldots, f_{1,n-2}, z_1, y_1, k_2$.

Evolution of the system configuration. The evolution of the system configuration from $\mathcal{C}_0$ to $\mathcal{C}_1$ is similar to Lemma 3.3 with $i = 0$, $s_i = s$ and $T_i = T$. Therefore, at the end of time period T, the number of packets in queues $e_{1,j}$ $(1 \leq j \leq n)$ that have remaining routes $e_{1,j}, \ldots, e_{1,n}, k_2$, and in queues $x_1, x_1^{'}$ requiring to traverse the edges y_1, k_2 are $2s^{'} = |X^{'}| + |Y| + |S_3| + |H| + |S_7|$. Similarly to Lemma 3.3, it is proved that this number of packets is larger than the number of S_0 packets for $r = \frac{3C^2-1}{2C^3-2C} + \varepsilon$. Therefore, after a period of $\frac{2s}{C} + 2\frac{(C-1)s}{C^2} + n$ steps the configuration $\mathcal{C}_0$ changes to $\mathcal{C}_1$ with $2s^{'} \geq 2s(1+\varepsilon)$.

C Proof of Lemma 3.5

We violate the definition of configuration to shorten the configuration of the network at time τ as $\langle s, \mathcal{G}(i) \rangle$ in case there are $2s$ packets at time τ, all queued in gadget $\mathcal{G}(i)$ and no packets in other gadgets of the network. This notation will be used for convenience in our proof.

The proof is split in two parts. First we prove that if $\langle s, \mathcal{G}(1) \rangle$ is the configuration of $\mathcal{N}_r$ at time τ, with $2s$ packets, then at the end of the M subphases there are $2s^{'} > 2s(1+\varepsilon)^{M-1}$ packets in the system, all queued in $\mathcal{G}(M)$. Then we prove that all $2s^{'}$ packets in $\mathcal{G}(M)$ at the end of the M subphases are queued at the output edge k_{M+1} of $\mathcal{G}(M)$. The first part of the proof is by induction on the number i of move subphases $(1 \leq i \leq M)$.

Basis case. For $i = 1$, the claim is trivial with $t_1 = \tau$.

Induction step. Consider that there is some time $t_i \geq \tau$ such that the system configuration is $\langle s_i, \mathcal{G}(i) \rangle$ for $2s_i > 2s(1+\varepsilon)^{i-1}$. Let now consider a subnetwork that consists of a chain of two gadgets $\mathcal{G}(i)$ and $\mathcal{G}(i+1)$. Applying Lemma 3.3, there is a suitable set of adversarial packet injections and a time period T_i such that at time $t_i + T_i$ the system configuration is $\langle s_{i+1}, \mathcal{G}(i+1) \rangle$ for $2s_{i+1} > 2s(1+\varepsilon)^i$ and all the packets in the system are only queued in $\mathcal{G}(i+1)$. Assigning $t_{i+1} = t_i + T_i = t_i + \frac{2s_i}{C} + 2\frac{(C-1)s_i}{C^2} + n$ and concatenating the set of adversarial packet injections. The proof of the first part is now complete.

From the first part we have that at time t_M the system configuration is $\langle s_1, \mathcal{G}(M) \rangle$ for $2s_1 \geq 2s(1+\varepsilon)^{M-1}$. If we do not make any injection in the time interval $[t_M, t_M + \frac{2s_1}{C} + 1]$ and consider that all the edges have capacity C except the output edge k_{M+1} of the gadget $\mathcal{G}(M)$ that has unit capacity, then the $2s_1$ packets that have been queued at the queues of $\mathcal{G}(M)$ at time t_M will arrive at the output edge k_{M+1} of $\mathcal{G}(M)$.

Furthermore, $\frac{2s_1}{C} + 1$ packets depart from the output edge k_{M+1} during the time interval $[t_M, t_M + \frac{2s_1}{C} + 1]$. Therefore, at time $t_M + \frac{2s_1}{C} + 1$, there are $2s^{'} = 2s_1 - \frac{2s_1}{C} - 1 \geq 2s_0 - \frac{2s_0}{C} - 1$ packets at the output edge k_{M+1}. If we consider $1 < n < \frac{s_0}{4C^3}$, then $2s^{'} \geq 2s_0 - \frac{2s_0}{C} - \frac{s_0}{4C^3} = \frac{(8C^3-8C^2-1)s_0}{4C^3}$. But, $2s \geq 2s_0$ and $2s^{'} \geq 2s(1+\varepsilon)^{M-1}$. So, $2s^{'} \geq \frac{(8C^3-8C^2-1)s}{4C^3}(1+\varepsilon)^{M-1}$ packets exist at the output edge k_{M+1} of the gadget $\mathcal{G}(M)$. For our specified C values, we have $2s^{'} \geq 2s(1+\varepsilon)^{M-1}$. This completes our proof.

D Proof of Lemma 3.6

Consider the network $\mathcal{N}_r$. At time t there is a set S_0 of $|S_0| = 2s$ packets queued in the queue k_{M+1} of the gadget $\mathcal{G}(M)$ requiring to traverse the edge k_{M+1}. We will show that there is a sequence of adversarial injections of rate r such that at time $t_1 = t + \frac{2s}{C} + r\frac{2s}{C} + r^2\frac{2s}{C}$ there are $r^3 2s$ packets in queue k_1, all being injected in k_1 after time t. We consider that all the edges have capacity C during time interval $(t, t_1]$. The sequence of adversarial injections happens in three rounds as follows:

- *Round 1:* This round lasts for $\frac{2s}{C}$ time steps. During this round the edges k_{M+1}, e_0, k_1 have capacity C. The adversary injects a set X of $|X| = r\frac{sC}{C} = 2rs$ packets in k_{M+1} requiring to traverse the edges k_{M+1}, e_0, k_1. The X packets are blocked in queue k_{M+1} because of the S_0 packets that are

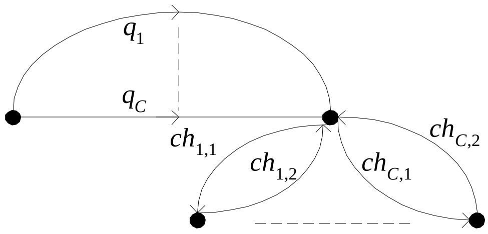

Figure 4: Subnetwork $\mathcal{D}$

queued in k_{M+1} at the beginning of this round. The S_0 packets have been absorbed at the end of this round.

- *Round 2:* This round lasts for $\frac{2rs}{C}$ time steps. During this round the edges k_{M+1}, e_0, k_1 have capacity C. The adversary injects a set Y of $|Y| = r\frac{2rsC}{C} = 2r^2s$ packets in k_1. The Y packets arrive simultaneously at k_1 with the X packets and they mix in proportion equal to their sizes. At the end of this round, there is a set Z of $|Z| = 2r^2s$ packets in the system that are queued in k_1, and no other packets exist in the system. Note that some of these packets have been injected in k_{M+1} and the rest in k_1.
- *Round 3:* This round lasts for $\frac{2r^2s}{C}$ time steps. During this round the edge k_1 has capacity C. The adversary injects a set L of $|L| = r\frac{2r^2sC}{C} = 2r^3s$ packets in k_1. The L packets blocked in k_1 by the Z packets. At the end of this round, all the Z packets have been absorbed. Therefore, at time $t + \frac{2s}{C} + r\frac{2s}{C} + r^2\frac{2s}{C}$ all the packets in the system are the $|L| = r^3 2s$ packets that have been injected in k_1 during this round and they are queued in k_1.

E Proof of Theorem 4.2

In order to simulate the behavior of packets flows passing over an edge on the network $\mathcal{N}_r'$ we replace each edge in $\mathcal{N}_r'$ with a subnetwork $\mathcal{D}$ whose edges have unit capacity. The network $\mathcal{D}$ (Figure 4) consists of C parallel edges q_l that have common source and destination ($1 \leq l \leq C$). For each edge q_l there is a small chain of two edges $ch(l,k)$ ($1 \leq k \leq 2$) that has as source and destination the destination of $\mathcal{D}$. Also, the packet paths are modified such that the adversary instead of injecting a packet flow of rCt packets into a queue of $\mathcal{N}_r'$ during t time steps, it injects rt packets into each queue q_l of $\mathcal{D}$ in $\mathcal{N}_r''$. The additional chains in $\mathcal{D}$ are used by the adversary when it wants to delay for one time step the first C packets of a packet flow that are inserted each one in each queue q_l of $\mathcal{D}$. This happens when the corresponding edge replaced by $\mathcal{D}$ in $\mathcal{N}_r'$ has a packet flow Y (which size is not a multiple of C) queued into it at some time and a new packet flow X is inserted into it which should leave this edge after all the the packets of flow Y leave. In order to handle this case the adversary forwards the first C packets of flow X that enter the queues q_l of the analyzer to traverse the chain $ch(l,k)$ after traversing q_l.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'