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

FIFO is Unstable at Arbitrarily Low Rates*

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

Abstract

In this work, we study the stability of the FIFO (First-In-First-Out) protocol in the context of Adversarial Queueing Theory. As an important intermediate step, we consider *dynamic capacities*, where each network link capacity may arbitrarily take on values in the two-valued set of integers $\{1, C\}$ for $C > 1$ being the *high capacity* (a parameter). In this context:

(1) We construct a FIFO network of only eight nodes which is already unstable at rate $r = 0.41$. This is the current record for instability of FIFO over networks of fixed-size (independent of r).

(2) For every $r > 0$ we then construct a FIFO network (whose size increases with $\frac{1}{r}$) which is unstable at rate r.

Subsequently, we show how to simulate the particular FIFO network in (2) above with dynamic capacities 1, C, in order to produce a FIFO network with all link capacities being equal, while preserving instability thresholds. Hence, we eventually show our main result: FIFO can become unstable in the usual model of *unit capacities* links, for arbitrarily low packet injection rates. This closes a major open problem (the question of FIFO stability) in the field of Adversarial Queueing Theory posed in the pioneering work of Borodin *et al.* [5].

Keywords: Routing, Adversarial Queueing Theory, FIFO, Stability.

<u>*Note 1:*</u> Many of our proofs are only sketched in this extended abstract; full proofs are included in a clearly marked Appendix.

*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 (research project "Efficiency and Performance of Distributed Systems: Capabilities and Limitations"), by the Greek General Secretariat for Research and Technology (project $\Pi ENE\Delta$ 99/ALKAD), 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. We are interested in the behavior of *packet-switched networks* in which packets arrive dynamically at the *nodes* and they are routed in discrete time steps across the *links*. Recent years have witnessed a vast amount of work on analyzing packet-switched networks under *non-probabilistic* assumptions (rather than stochastic ones); We work within a model of *worst-case* continuous packet arrivals, originally proposed by Borodin *et al.* [5] and termed *Adversarial Queueing Theory* to reflect the assumption of an *adversarial* way of packet generation and path determination.

A major issue that arises in such a setting is that of *stability*– will the number of packets in the network remain bounded at all times? The answer to this question may depend on the *rate* of injecting packets into the network, the *capacity* of the links, which is the rate at which a link forwards outgoing packets, and the *protocol* that is used when more than one packet wants to cross a given link in a single time step. The underlying goal of our study is to establish stability properties of FIFO protocol when packets are injected by an adversary (rather than by an oblivious randomized process).

In this work we consider that the adversary besides the packet injections in paths which it determines, it also can set the capacities of edges in each time step as an important intermediate step in our constructions. Note that we continue to assume uniform packet sizes. Furthermore, we consider greedy contention-resolution protocols- always advance a packet across a queue (but one packet at each discrete time step) whenever there resides at least one packet in the queue. Roughly speaking a protocol P is *stable* [5] on a network $\mathcal{G}$ against an adversary $\mathcal{A}$ of rate r if there is an integer B (which may depend on $\mathcal{G}$ and $\mathcal{A}$) such that, starting from an empty configuration, the number of packets in the system is bounded at all times by B.

One of the main interesting open problems in this area that first posed by "Borodin *et al.*" [5] is to determine if FIFO can be made unstable for arbitrarily small positive rates of injection in the adversarial model. This is an important problem taking into account that FIFO is the most popular protocol for contention resolution in the Internet and other networks due to its simplicity and it has received a lot of interest in the last years (see, e.g., [1, 2, 4, 9, 10, 11, 12]).

Contribution. In this paper we prove that FIFO can be unstable at arbitrarily low packet injection rates in the Adversarial Queueing Model. Our method of proving this main result uses the model of dynamic link capacities as an important intermediate step. More specifically, we initially consider networks whose links can have a dynamically changing capacity which may have any of the two integer values, 1 and C. Here $C > 1$ is an integer called the *high capacity*. In this framework, we first obtain the following results:

(a) We construct a FIFO network of only 8 links that is unstable for any rate $r \geq 0.41$ (for large enough C values). This is the current record for FIFO instability over fixed-size networks (networks where size is independent of r).

(b) We present an innovative parameterized adversarial construction $\mathcal{A}$ and a parameterized network family $\mathcal{G}_r$ which, for any $r > 0$, gives a system ($\mathcal{G}_r$,$\mathcal{A}$,FIFO) that is unstable at rates $\geq r$.

Our results above use the model of *simple paths* for packet paths (paths that do not contain overlapping edges). We then modify the construction (Interpreter A_1) via suitable changes in the network topology and packet paths (Figure 1), and thus we obtain, for any $r > 0$, a system ($\mathcal{G}_r^{'}$,$\mathcal{A}^{'}$,FIFO) where all links have capacity C all the times, which is unstable at rates $\geq r$. Finally, we demonstrate how to simulate the above family of FIFO networks by a family of FIFO networks where all links have capacity 1 all the times (Interpreter A_2) and thus, for any $r > 0$, we provide a system ($\mathcal{G}_r^{''}$,$\mathcal{A}^{''}$,FIFO) which is unstable at rates $\geq r$. These simulation steps require the use of *non-simple* packet paths (paths that

Figure 1: Simulations

contain overlapping edges). We take care here so that the non-simple paths have a length bounded by a fixed function of the network size. Thus, taking the simulation steps together we show that, for any injection rate $r > 0$, we can construct a FIFO network in the usual Adversarial Queueing Theory setting that is unstable at rates $\geq r$. The size of our final network is exponential in $\frac{1}{r}$. Our result closes one of the major remaining open questions in the field of Adversarial Queueing Theory.

Related Work. The problem of dynamic packet arrivals and routing has been a major topic of study within the field of *Queueing Theory* [8]. Typical assumption in this field are that packets are generated according to a Poisson process, and that the time to traverse an edge is an exponentially-distributed random variable, rather than a fixed constant. Instability in stochastic networks was first demonstrated in [13]. *Adversarial Queueing Theory* was developed by Borodin *et al.* [5] as a more realistic model that replaces traditional stochastic assumptions in Queueing Theory by more robust, worst-case ones. Adversarial queueing theory received a lot of interest in the study of stability and instability issues (see, e.g., [2, 9, 11, 12, 14]). Borodin *et al.* in [6] studied for the first time the impact on stability when the edges in a network can have capacities and slowdowns and proved that the universal stability of networks is preserved for dynamically changing capacities and speeds, while the universal stability of Longest-In-System protocol is not preserved. In this work we continue this study as an intermediate step in our constructions to the stability properties of the FIFO protocol.

The problem of FIFO stability has been investigated under various models leading to a number of contrasting results. Till now there is no model for which FIFO has been proved unstable for arbitrarily low injection rates. Kelly [10] showed that in stochastic networks where packet sizes and service times are exponentially distributed, if the service time distribution at a server is session-independent then FIFO is stable. Our result here is in contrast to Kelly's result showing that the Stochastic and Adversarial Queueing models are different. Andrews [1] showed that FIFO is unstable for network loads close to 1 in session-oriented networks [7]. Bramson [4] proved that FIFO is stable in the session-oriented model if time is scaled, such that the packet sizes and the burstiness tend to zero and the injection rate of the adversary is constant. The instability of FIFO for *small-size* networks in the adversarial queueing model where the network size is independent of r was first established by Andrews *et al.* [2, Theorem 2.10] for injection rates $r \geq 0.85$. Lower bounds of 0.8357 and 0.749 on FIFO instability were presented by Diaz *et al.* [9, Theorem 3] and Koukopoulos *et al.* [11, Theorem 5.1] respectively. Here we further improve the current record for FIFO instability over fixed-size networks presenting a FIFO network that is unstable for $r = 0.41$. Except adversarial constructions for small-size networks, there are parameterized constructions for networks with *unbounded size.* In the context of FIFO instability Lotker *et al.* [12] proved an injection rate lower bound of 0.5 through a very nice parameterized construction on rate r; the network size is a function of r that goes to infinity very fast as r goes down to 0.5. Here using a parameterized construction in the same spirit as Lotker *et al.* [12] we prove that FIFO is unstable at arbitrarily low injection rates.

Road Map. The rest of this paper is organized as follows. Section 2 presents our model definitions. the model of dynamic link capcities.Section 3 deals with FIFO instability in networks with dynamic capacities where packet injections follow the simple path model. Section 4 presents how to simulate the family of the unstable FIFO networks $\mathcal{G}_r$ that described in Section 3 by a family of FIFO networks where all links have capacity C all the times that are also unstable at rates $\geq r$. Then, it shows how the family of FIFO networks with uniform capacities C that are unstable for any $r > 0$, can be simulated by a family of FIFO networks with unit link capacities that are also unstable at rates $\geq r$. We conclude, in Section 5, with a discussion of our results and some open problems.

2 The Model

Our model definitions are patterned after those in [5, Section 3]. For our intermediate step, we extend this model to reflect the fact that edge capacities may vary arbitrarily as in [6, Section 2]. A *routing network* is a directed graph with *nodes* and *edges*. Time proceeds in discrete steps. A *packet* is an atomic entity that resides at a node at the end of any step. It must travel along paths in the network from its *source* to its *destination,* both of which are nodes in the network. When it reaches its destination, we say that it is *absorbed*. During each step, a packet may be sent from its current node along one of the outgoing edges from that node. 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.

Any packets that wish to travel along an edge e at a particular time step but are not sent wait in a queue for edge e. The *delay* of a packet is the number of steps spent by the packet while waiting in queues. At each step, an em adversary generates a set of requests. A *request* is a *path* specifying the route followed by a packet.[1] We say that the adversary generates a set of packets when it generates a set of requested paths. The adversary we use to prove instability in the case of the small-size FIFO network predetermines the paths of the injected packets such that the path traversed by each packet is fixed at the time of injection. However, in the other adversarial constructions the packets have long paths. In order to handle this difficulty, we adopt a technique introduced by Lotker *et al.* [12, Lemma 3.1] that permits the adversary to specify packet paths in an "on-line" fashion without changing the power of the adversary. That is, we can construct an adversary that does not specify the complete path of the packets when they are injected, but it constructs it in a succession of refinements. There are no computational restrictions on how the adversary chooses its requests in any given time step.

Fix any arbitrary positive integer $w \geq 1$. For any edge e of the network and any sequence of w consecutive time steps, define $N(w, e)$ to be the number of paths injected by the adversary during the time interval of w consecutive time steps that traverse edge e. For any constant r, $0 < r \leq 1$, a (w, r)-*adversary* is an adversary that injects packets subject to the following *load condition*: For every edge e and for every sequence τ of w consecutive time steps, $N(\tau, e) \leq r \sum_{t \in \tau} \mathsf{C}_\mathsf{e}(t)$.

We say that a (w, r)-adversary injects packets at rate r with *window size* w. The assumption that $r \leq 1$ ensures that it is not necessary a priori that some edge of the network is congested (which would surely happen when $r > 1$). We say that a (w, r)-adversary follows the model of uniform capacities if for every link e $C_e(t) = C$ at all times t. We remark that the model of unit capacities is the usual Adversarial Queueing Model setting. Roughly speaking, this model views the time evolution of a packet-switched communication network as a game between an *adversary* and a *contention-resolution* protocol. A *contention-resolution* protocol specifies, for each pair of an edge e and a time step, which packet among those waiting at the tail of edge e will be moved along edge e. In this work, we restrict attention to FIFO that gives priority to the packets that arrived in the queue at the earliest time.

In the adversarial constructions we study here for proving instability, we assume that there is a sufficiently large number of packets s_0 in the initial system configuration. This will imply instability results for networks with an *empty* initial configuration, as established by Andrews *et al.* [2, Lemma 2.9]. For simplicity, and in a way similar to that in [2] and in works following it, we omit floors and ceilings from our analysis, and we sometimes count time steps and packets only roughly. This may only result to loosing small additive constants, while it implies a gain in clarity.

[1] In Section 3, it is assumed, as it is common in packet routing, that all such paths are simple paths with no overlapping edges. In Section 4, we use non-simple paths whose size is bounded by the network size.

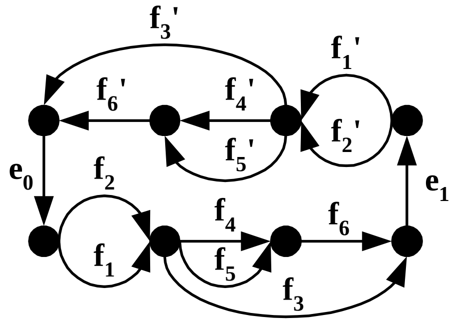

Figure 2: The network $\mathcal{G}$

3 FIFO Networks with Dynamic Capacities

3.1 A Small-Size FIFO Unstable Network

Theorem 3.1 *For any $r > 0.41$, there is a a network $\mathcal{G}$ whose size is independent of r and an adversary $\mathcal{A}$ of rate r, such that the system ($\mathcal{G}$, $\mathcal{A}$, FIFO) is unstable in the model of dynamic capacities.*

Sketch of Proof: Consider the network $\mathcal{G}$ in Figure 2. We break the construction of the adversary $\mathcal{A}$ into phases.

Inductive Hypothesis: At the beginning of phase j (suppose j is even), there are s_j packets that are queued in the queues $f_1^{'}, f_2^{'}, f_4^{'}, f_5^{'}, f_6^{'}$ (in total) requiring to traverse the edges e_0, f_2 and the number of packets in queues $f_1^{'}, f_4^{'}$ is larger than in queues $f_2^{'}, f_5^{'}$.

Induction Step: At the beginning of phase $j + 1$ there will be more than s_j packets (s_{j+1} packets) that are queued in the queues f_1, f_2, f_4, f_5, f_6 (in total) requiring to traverse the edges $e_1, f_2^{'}$ and the number of packets in queues f_1, f_4 is larger than in queues f_2, f_5.

We will construct an adversary $\mathcal{A}$ such that the induction step holds. The main ideas of the construction of $\mathcal{A}$ are (a) the careful tuning of the duration of each round of every phase j (as a function of the high capacity C, the injection rate r and the number of packets in the system at the beginning of phase j, s_j) to maximize the growth of the packet population in the system and, (b) the careful setting of the capacities of some edges to 1 for specified time intervals in order to accumulate packets. Proving that the induction step holds, we ensure that the inductive hypothesis will hold at the beginning of phase $j + 1$ for the symmetric edges with an increased value of s_j, $s_{j+1} > s_j$. From the inductive hypothesis, initially, there are s_j packets (called $S - flow$) in the queues $f_1^{'}, f_2^{'}, f_4^{'}, f_5^{'}, f_6^{'}$ requiring to traverse the edges e_0, f_2. In order to prove the induction step, it is assumed that there is a set S with a large enough number of $|S| = s_j$ packets in the initial system configuration. During phase j the adversary plays three rounds of injections. The sequence of injections is as follows:

Round 1: This round lasts $|T_1| = \frac{s_j}{C}$ time steps. During this round the edges $f_1^{'}, e_0, f_1, f_4, f_6, e_1, f_2^{'}$ have capacity C, while all the other edges have unit capacity.

Adversary's behavior. During this round the adversary injects a set X of $|X| = rC|T_1|$ packets in queue $f_1^{'}$ wanting to traverse the edges $f_1^{'}, e_0, f_1, f_4, f_6, e_1, f_2^{'}$. Also, the adversary injects a set S_1 of $|S_1| = r|T_1|$ packets in queue f_2 wanting to traverse the edge f_2.

Evolution of the system configuration. All the S packets will traverse their initial edges in s_j time steps blocking the packets of set X in queue e_0. At the same time, the packets of set S are delayed in queue f_2 due to the packets of set S_1 and the unit capacity of the edge f_2. The remaining packets of the set S in f_2 at the end of this round are $|S^{'}| = |S| - |T_1|\frac{|S|}{|S|+r|T_1|}$ packets, while the rest traverse the edge f_2 and they are absorbed. The remaining packets of the set S_1 in queue f_2 at the end of this round are $|S_1^{'}| = |S_1| - |T_1|\frac{|S_1|}{|S|+|S_1|}$. The rest packets of the set S_1 traverse the edge f_2 and they are absorbed. Therefore, the number of packets in queue f_2 at the end of this round requiring to traverse edge f_2 is $|S^{''}| = |S^{'}| + |S_1^{'}|$.

Round 2: It lasts $|T_2| = \frac{C-1+r}{C^2} s_j$ time steps. During this round the edges $f_2, f_5, f_6, e_1, f_2^{'}$ have capacity C, while all the other edges have unit capacity.

Adversary's behavior. The adversary injects a set Y of $|Y| = rC|T_2|$ packets in queue f_2 requiring to traverse the edges $f_2, f_5, f_6, e_1, f_2^{'}$. Furthermore, the adversary injects a set S_2 of $|S_2| = r|T_2|$ packets in queue f_1 requiring to traverse the edge f_1.

Evolution of the system configuration. The packets of set Y are blocked by the set $S^{''}$ in queue f_2. At the same time, X packets are delayed in queue f_1 by the S_2 packets with which they get mixed with proportion equal to their sizes. Also, X packets delay in f_1 because it uses unit capacity. Therefore, the remaining packets of X in queue f_1 is $|X^{'}| = |X| - |T_2|\frac{|X|}{|X|+|S_2|}$. The remaining packets of the set S_2 in f_1 at the end of this round are $|S_2^{'}| = |S_2| - |T_2|\frac{|S_2|}{|X|+|S_2|}$. The rest packets of the set S_2 traverse the edge f_1 and they are absorbed. Therefore, the total number of packets in queue f_1 at the end of this round requiring to traverse the edge f_2 is $|F| = |X^{'}| + |S_2^{'}|$.

Round 3: It lasts $|T_3| = \frac{r^3+r^2[2C^2+2C-3]+r[C^4+2C^3-2C^2-4C+3]-C^3+2C-1}{C^3[C^2+C-1+r]} s_j$ time steps. During this round the edges $f_1, f_3, e_1, f_2^{'}$ have capacity C, while all the other edges have unit capacity.

Adversary's behavior. During this round the adversary injects a set Z of packets in queue f_1 requiring to traverse the edges $f_1, f_3, e_1, f_2^{'}$ where $|Z| = rC|T_3|$.

Evolution of the system configuration. The F packets block the Z packets in queue f_1, while they traverse f_1. From the F packets only the $X^{'}$ packets have as destination another edge than f_1, while the rest are absorbed after traversing f_1. Because edge f_4 has unit capacity, $X^{'}$ packets are delayed in this queue. From these packets a portion $X^{''}$ of packets requiring to traverse the edges $f_4, f_6, e_1, f_2^{'}$ remain there where $|X^{''}| = |X^{'}| - |T_3|$. Because edge f_5 has unit capacity, Y packets are delayed in this queue. From these packets a portion $Y^{'}$ of packets requiring to traverse the edges $f_5, f_6, e_1, f_2^{'}$ remain there where $|Y^{'}| = |Y| - |T_3|$. Note that during this round $|K| = 2|T_3|$ packets arrive in queue f_6 from queues f_4, f_5. However, the edge f_6 has unit capacity and the duration of this round is $|T_3|$ steps. Consequently, half of these packets will remain in queue f_6 at the end of this round. Therefore, L packets will remain in queue f_6 at the end of this round requiring to traverse the edges $f_6, e_1, f_2^{'}$ where $|L| = |T_3|$. Therefore, the number of packets in queues f_1, f_2, f_4, f_5, f_6 requiring to traverse the edges $e_1, f_2^{'}$ at the end of this round is $s_{j+1} = |X^{''}| + |Y^{'}| + |Z| + |L|$ (1). In order to have instability, we must have $s_{j+1} > s_j$ (2). Replacing s_{j+1} in (2) from (1) we take $r^4 + r^3[2C^3 + 3C^2 - 3C - 1] + +r^2[C^5 + 3C^4 + C^3 - 8C^2 + C + 3] + r[2C^5 - C^4 - 7C^3 + 5C^2 + 3C - 3] > C^5 + 2C^4 - 3C^3 + 2C - 1$ (3).

This argument can be repeated for an infinite number of phases showing that the number of packets in the system increases forever. Furthermore in order to guarantee that the number of packets in queues f_1, f_4 is larger than in queues f_2, f_5, we must have $|Z| + |X^{''}| > |Y^{'}|$. Replacing the above quantities we take $r^4C + r^3[2C^3 + C^2 - 3C] + r^2[C^5 + C^4 - 3C^3 - 2C^2 + C] + r[C^2 - C] > C^4 - C^3$ (4). Any r that satisfies Inequality (3) satisfies Inequality (4), too. Therefore the instability threshold of rate r for the system ($\mathcal{G}$, $\mathcal{A}$, FIFO) is defined by the Inequality (3). When C tends to infinity, the instability threshold converges to 0.41. ■

3.2 An Infinite Family of FIFO Unstable Networks

Theorem 3.2 *For any $r > 0$, there is a network $\mathcal{G}_r$ and an adversary $\mathcal{A}$ of rate r, such that the system ($\mathcal{G}_r$, $\mathcal{A}$, FIFO) is unstable in the model of dynamic capacities.*

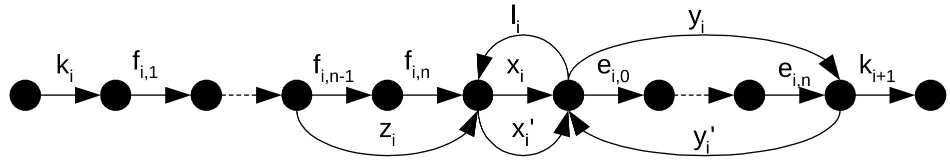

Figure 3: The gadget $\mathcal{F}(i)$

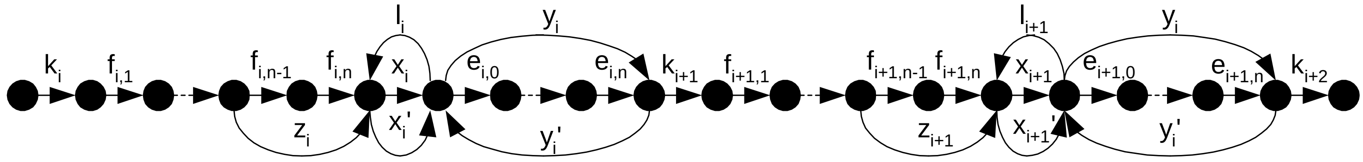

Figure 4: A chain of two gadgets $\mathcal{N}$

The proof of the theorem is structured into two parts. In Section 3.2.1, we demonstrate the basic component (gadget) of the infinite network family $\mathcal{G}_r$ and the construction of $\mathcal{G}_r$. In Section 3.2.2, we present the adversarial construction that leads the system ($\mathcal{G}_r$, $\mathcal{A}$, FIFO) to instability.

3.2.1 The Network $\mathcal{G}_r$

Gadget topology. The network $\mathcal{G}_r$ is a cascade of M similar subnetworks called *gadgets* (see Figure 3). The i^{th} gadget, $\mathcal{F}(i)$, of the network $\mathcal{G}_r$ where $1 \leq i \leq M$ is a directed acyclic graph that consists of:

- An input edge k_i, and an output edge k_{i+1}.
- 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 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 $f_{i,n}$.
- 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.

Gadgets concatenation. Given two gadgets $\mathcal{G}$, $\mathcal{H}$ define $\mathcal{G} \circ \mathcal{H}$ to be the gadget that results from identifying the sink of $\mathcal{G}$ with the source of $\mathcal{H}$, the source of $\mathcal{G} \circ \mathcal{H}$ with the source of $\mathcal{G}$, and the sink of $\mathcal{G} \circ \mathcal{H}$ with the sink of $\mathcal{H}$. We call the operation $\circ$ *chaining*.

Network topology. The network $\mathcal{G}_r$ is a concatenation of M gadgets $\mathcal{F}(1), \ldots, \mathcal{F}(M)$ with one additional edge e_0 bridging the output edge of $\mathcal{F}(M)$ to the source of the input edge of $\mathcal{F}(1)$ (Figure 5).

Network size. Each gadget's size is $Gs = 2n + 9$ where $C > n > \max\{\frac{\lg(\varepsilon)-\lg(2)}{\lg(r)}, 1 - \frac{1}{\lg(r)}\}$. Thus, $Gs > 2(\max\{\frac{\lg(\varepsilon)-\lg(2)}{\lg(r)}, 1 - \frac{1}{\lg(r)}\}) + 9$. The network $\mathcal{G}_r$ consists of M gadgets where $M > \frac{\lg(16C^6)-\lg[(8C^3-8C^2-1)^2r^3]}{\lg(1+\varepsilon)}$, for any $r > \varepsilon > 0$. Hence, the size of the network is polynomial in $\frac{1}{\lg(r)}$.

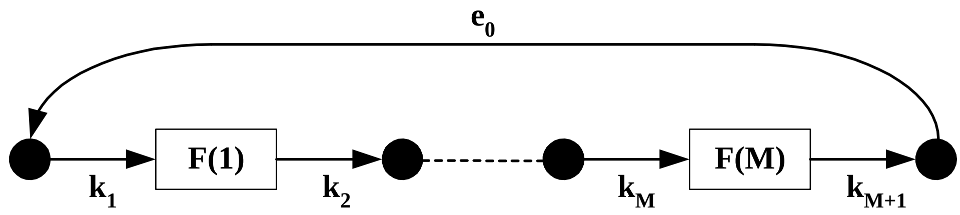

Figure 5: The network $\mathcal{G}_r$

System configuration. The configuration C^t of the system ($\mathcal{G}_r$, $\mathcal{A}$, FIFO) in every time step t is a collection of sets $\{S_e^t : e \in \mathcal{G}_r\}$, such that S_e^t is the set of packets waiting in the queue of the edge e at the end of step t. Because of the construction of our adversary the packets move between consecutive gadgets such that at specific time steps the packets are queued inside only one gadget. This permits us to define the system configuration at a time t as $C^t(s, \mathcal{F})$ where $\mathcal{F}$ is the only gadget in the system whose queues have packets at time t and the number of these packets is $2s$. Therefore, we can see the time evolution of the system as a sequence of such configurations. At time t the system ($\mathcal{G}_r$, $\mathcal{A}$, FIFO) has the following configuration ($C^t(s, \mathcal{F}(i))$): (i) there are $2s$ packets 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 $e_{i,j}$, where $0 \leq j \leq n$, have remaining routes $e_{i,j}, \ldots, e_{i,n}, k_{i+1}$, while the packets in queues $x_i, x_i^{'}$ require to traverse the edges y_i, k_{i+1} and, (ii) no other queue in $\mathcal{F}(i)$ has any packets.

3.2.2 The Adversary

Basic Idea. Time is divided into phases. We will show that the number of packets at the end of each phase increases comparing to the beginning of the phase. Each phase is divided into a number of $M + 2$ subphases. From this number, M subphases (*load* subphases) refer to the evolution of system configuration as the packets move from gadget to gadget. At the beginning of each of these subphases, there is a number of packets that are queued in the queues of only one gadget some of which will continue to the next gadget during the subphase. The other two subphases (*connection* subphases) are used for bridging consecutive phases. One of the *connection* subphases is used for avoiding path overlapping. It replaces the number of packets arriving at the output edge k_{M+1} of the $\mathcal{F}(M)$ gadget with a number of packets in the input edge k_1 of the $\mathcal{F}(1)$ gadget that do not have previous history. The other one is used along with a number of proper injections for the reproduction of the initial system configuration of the $\mathcal{F}(1)$ gadget (packets are queued in a specific subset of queues of the first gadget). At the end of each of these subphases we will show that the number of packets in the system increases except the *connection* subphase that is used for avoiding path overlapping. At the end of this subphase, the number of packets in the system decreases. However we manage to overcome this decrease and guarantee the increase of the number of packets in the system at the end of the phase comparing to its beginning choosing suitably the number M of gadgets.

Packet rerouting. Based on the technique introduced by Lotker *et al.* [12, Lemma 3.1] the adversary at the beginning of each subphase assigns to the packets that are queued into the system an extension to their path, which consists of edges that do not overlap with the path that has been already traversed by the packets. The new path covers edges of the gadget where the packets at the beginning of a subphase are queued and edges of the next gadget. For example the paths of the packets that are queued in the gadget $\mathcal{F}(i)$, where $1 \leq i \leq M$, are extending to the next gadget $\mathcal{F}(i + 1)$ during a subphase as follows: (i) the packets in queue $e_{i,j}$, where $0 \leq j \leq n$, have remaining routes $e_{i,j}, \ldots, e_{i,n}, k_{i+1}, f_{i+1,1}, \ldots, f_{i+1,n-2}, z_{i+1}, x_{i+1}^{'}, e_{i+1,0}, \ldots, e_{i+1,n}, k_{i+2}$ and, (ii) the packets in queues $x_i, x_i^{'}$ require to traverse the edges $y_i, k_{i+1}, f_{i+1,1}, \ldots, f_{i+1,n-2}, z_{i+1}, x_{i+1}^{'}, e_{i+1,0}, \ldots, e_{i+1,n}, k_{i+2}$.

Note that in our adversarial construction we assume that all the packets that leave the gadget, in which packets are queued at the end of a subphase, during the subphase are absorbed and they do not continue to the next gadget. This ensures that there are no packets in other gadgets.

Flow through a chain. The lemma below is based on Lotker *et al.* [12, Lemma 3.2 (Claims 3.5, 3.7)].

Lemma 3.3 *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 an adversary of rate r, such that the number of packets remaining into the system is $|L^{'}| \leq rt$, all the edges have at least one packet and only $L^{'}$ packets are queued into the chain queues at time step $t+n$.*

Sketch of Proof: The basic idea behind this adversarial construction is that the adversary injects packets that require to traverse only the edge where they are injected in order to block packets that want to traverse all the edges of the chain. The packets that want to traverse a single edge are injected in proper chosen time intervals such that at the end of the examined time period there are no such packets in the queues of the chain and all the queues of the chain are not empty. ■

Subphase Population Growth. The following Lemma proves that the number of packets in the system ($\mathcal{G}_r$, $\mathcal{A}$, FIFO) increases for a subphase with duration $|T| = \frac{2s_i}{C} + 2\frac{(C-1)s_i}{C^2} + n$. We assume that $r > \varepsilon > 0$, $C > n > \max\{\frac{\lg(\varepsilon)-\lg(2)}{\lg(r)}, 1 - \frac{1}{\lg(r)}\}$, and $s_0 > 4nC^3$.

Lemma 3.4 *Let $r = \frac{3C^2-1}{2C^3-2C} + \varepsilon$. Then, there is a network $\mathcal{N}$ that consists of two chained gadgets $\mathcal{F}(i), \mathcal{F}(i+1)$, an adversary $\mathcal{A}$ of rate r and a time period T (defined above), such that the number of packets in the system ($\mathcal{N}$, $\mathcal{A}$, FIFO) increases at the end of the time period T.*

Sketch of Proof: Consider the network $\mathcal{N}$ in Figure 4. Assume that *the initial system configuration* at time τ is as follows: (i) there are $2s_i$ packets 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 $e_{i,j}$, where $0 \leq j \leq n$, 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 $\mathcal{F}(i)$ and no queue in $\mathcal{F}(i+1)$ has any packets.

In order to prove that the final system configuration holds, it is assumed that there is a large enough number of $2s_i$ packets in the initial system configuration (packet set S). This permits us to ignore floors and ceilings in the adversary specification for simplicity of presentation as they add only additive terms that are compensated for. Furthermore, for simplicity of notation we assume that $\tau = 0$. The adversary makes injections in a time period T with duration $|T| = \frac{2s_i}{C} + \frac{2(C-1)s_i}{C^2} + n$. During this time period all the edges of the network $\mathcal{N}$ have capacity C except some edges that have unit capacity in specific time intervals of period T: (i) The edge $x_{i+1}^{'}$ and the edges $e_{i+1,0}, \ldots, e_{i+1,n}$ have unit capacity in time interval $[1, \frac{2s_i}{C} + n]$, while (ii) the edge x_{i+1} and the edges $e_{i+1,0}, e_{i+1,1}, \ldots, e_{i+1,n}$ have unit capacity in time interval $[\frac{2s_i}{C} + n + 1, \frac{2s_i}{C} + \frac{2(C-1)s_i}{C^2} + n]$.

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 $e_{i+1,1}, \ldots, e_{i+1,n}$, based on Lemma 3.3. (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 $x_{i+1}^{'}, y_{i+1}, k_{i+2}$. (iv) 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 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. At the end of this period, the number of packets in queue $e_{i+1,j}$ for $0 \leq j \leq n$ that have remaining routes $e_{i+1,j}, \ldots, e_{i+1,n}, k_{i+2}$ are a set S_3 of $|S_3| = 2s_i\frac{(C-1)^2}{C^2}$ packets in $e_{i+1,0}$, a set H of $|H| = n$ packets and a number of $|S_7| = [2s_i\frac{2C-1}{C^2} - n]\frac{r-r^{n+1}}{1-r^{n+1}}$ packets in

$e_{i+1,1}, \ldots, e_{i+1,n}$. Furthermore, at the end of this period, a portion X' of $|X'| = 2s_i[r\frac{C-1}{C} - \frac{C-1}{C^2+C}]$ packets from the X packets are queued in x_{i+1} and the set Y of packets are queued in x'_{i+1}. Thus, at the end of the time period T the number of packets in the system is $2s_{i+1} = X' + Y + S_3 + H + S_7$. Replacing the quantities in the right part of the previous equation we take $2s_{i+1} = 2s_i[\frac{(C-1)^2}{C^2} - \frac{C-1}{C^2+C} + 2r\frac{C-1}{C} + r\frac{2C-1}{C^2}\frac{1-r^n}{1-r^{n+1}}] + n - nr\frac{1-r^n}{1-r^{n+1}}$. However $\frac{1-r^n}{1-r^{n+1}} \leq 1$ as $r \leq 1$. Thus, $nr\frac{1-r^n}{1-r^{n+1}} \leq nr$. Therefore $n - nr\frac{1-r^n}{1-r^{n+1}} \geq n - nr \geq 0$. Thus, $2s_{i+1} \geq 2s_i[\frac{C^4-2C^3+C}{C^4+C^3} + \frac{1}{C^2}[r(2C-2)C + (2C-1)\frac{r-r^{n+1}}{1-r^{n+1}}]]$ (1). But, $1 - r^n \leq 1$ and $n > \max\{\frac{\lg(\varepsilon)-\lg(2)}{\lg(r)}, 1 - \frac{1}{\lg(r)}\}$. Therefore $r^n \leq \frac{1}{2}$ and $2r^n < \varepsilon$. Also, consider that $r > \varepsilon$. Then, from inequality (1) we take $2s_{i+1} \geq 2s_i[\frac{C^4-2C^3+C}{C^4+C^3} + \frac{1}{C^2}(r(2C-2)C + (2C-1)(r-\varepsilon))]$ (2). It suffices to show $\frac{C^4-2C^3+C}{C^4+C^3} + \frac{2C-2}{C^2}rC > 1 \Rightarrow r > \frac{3C^2-1}{2C^3-2C}$ (3). Replace $r = \frac{3C^2-1}{2C^3-2C} + \varepsilon$ for $r > \varepsilon > 0$ in (2). Then, $2s_{i+1} \geq 2s_i[\frac{C^4-2C^3+C}{C^4+C^3} + \frac{1}{C^2}[2(C-1)C\frac{3C^2-1}{2C^3-2C} + 2(C-1)C\varepsilon + (2C-1)\frac{3C^2-1}{2C^3-2C}]] \geq 2s_i[1 + \frac{2(C-1)}{C}\varepsilon] \geq 2s_i(1+\varepsilon)$. Therefore, $2s_{i+1} \geq 2s_i(1+\varepsilon)$ for r such that $r = \frac{3C^2-1}{2C^3-2C} + \varepsilon$ where $r > \varepsilon > 0$. ■

Phase Population Growth. Now we will prove that the number of packets in the system ($\mathcal{G}_r$, $\mathcal{A}$, FIFO) increases in a phase. First, we will show that if there is a number of packets $s \geq$ a suitable number of initial packets s_0 in queue k_1, then after a time period the configuration of $\mathcal{F}(1)$ gadget will have more than s packets positioned as it is described in Section 3.2.1. Then, we will prove that as the packets move in consecutive gadgets in network $\mathcal{G}_r$ their number increases. After that, we will show how we can avoid path overlapping in our adversarial construction replacing the number of packets that arrive at the output edge of the $\mathcal{F}(M)$ gadget with a number of packets in the edge k_1 that are injected in k_1 and they do not have previous history. We assume that $r > \varepsilon > 0$, $r > \frac{1}{C}$, $C > n > \max\{\frac{\lg(\varepsilon)-\lg(2)}{\lg(r)}, 1 - \frac{1}{\lg(r)}\}$ and $s_0 > 4nC^3$.

Lemma 3.5 *Let a number $2s$ of packets in queue k_1 at time t. Then, there is an adversary $\mathcal{A}$ of rate r with $r = \frac{3C^2-1}{2C^3-2C} + \varepsilon$ and some time t_1 where the system has more than $2s'$ packets which configuration will be $C^{t_1}(s', \mathcal{F}(1))$, for $s' \geq s(1+\varepsilon)$.*

Sketch of Proof: The proof of this Lemma is similar to the proof of Lemma 3.4. The differences concern the edges that change capacities and the paths that are assigned to the injected packets by the adversary. Thus, the edges x'_1, x_1 and $e_{1,0}, \ldots, e_{1,n}$ 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 3.4. Furthermore, the adversary makes packet injections with similar paths in corresponding time intervals as in Lemma 3.4 in some edges of the gadget F(1). However there is one exception concerning the first injection of packets (set X in Lemma 3.4). Here the path assigned to these packets consists of edges that belong to $\mathcal{F}(1)$, while in Lemma 3.4 it contains edges that belong to the gadget where the packets in the system at the beginning of a subphase are queued and edges of the next consecutive gadget. ■

Lemma 3.6 *Let at time τ the system ($\mathcal{G}_r$,$\mathcal{A}$,FIFO) has configuration $C^\tau(s, \mathcal{F}(1))$. Then, there is an adversary $\mathcal{A}$ of rate r such that at some time $t > \tau$, there are $s' > s\frac{(8C^3-8C^2-1)s}{4C^3}(1+\varepsilon)^{M-1}$ packets in the system that are queued at the output edge k_{M+1} of the gadget $\mathcal{F}(M)$.*

Sketch of Proof: In order to prove this lemma, we need the following claim.

Claim 3.7 *Let $1 \leq i \leq M$. If at time τ all the packets in the network $\mathcal{G}_r$ have been injected after time τ_0, the system configuration is $C^\tau(s, \mathcal{F}(1))$ for $s \geq s_0$ and there are no other packets in the gadgets $\mathcal{F}(i)$ of $\mathcal{G}_r$, then there is an adversary $\mathcal{A}$ of rate r and some time $t_i \geq \tau$, such that the system configuration is $C^{t_i}(s', \mathcal{F}(i))$ for $s' > s(1+\varepsilon)^{i-1}$ and there are no other packets in the gadgets $\mathcal{F}(i), \ldots, \mathcal{F}(M)$ of $\mathcal{G}_r$.*

From the Claim 3.7 we take that, at time t_M the system configuration is $C^{t_M}(s', \mathcal{F}(M))$ for $s' \geq s(1+\varepsilon)^{M-1}$. Also, we consider that there is not any injection in the time interval $[t_M, t_M + \frac{2s'}{C} + 1]$ and all the edges have capacity C except the output edge k_{M+1} of the gadget $\mathcal{F}(M)$ that has unit capacity. After that we estimate the number of packets remaining in k_{M+1} at the end of the above time interval concerning that $1 < n < \frac{s_0}{4C^3}$. ■

Lemma 3.8 *Let s packets that are queued in k_{M+1} at time t. Then, for any $r > 0$ there is an adversary $\mathcal{A}$ of rate r, such that at a time t_1 there are $r^3 s$ packets in queue k_1 that have been injected there after time t.*

Sketch of Proof: We consider that all the edges have capacity C that permit us to use standard techniques to prove this lemma. The basic idea of the construction of the adversary is to replace the packets arriving at the output edge k_{M+1} of the $\mathcal{F}(M)$ 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 the adversary injects a set of packets X 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 the adversary injects a set Y of packets requiring to traverse k_1 that mix with X. In the third step we inject a set of new packets in k_1 that are blocked there by the previously injected packets that are absorbed. ∎

Proof of Instability (Theorem 3.2). Consider the network $\mathcal{G}_r$ in Figure 5. Choose M such that $M > \frac{\lg(16C^6)-\lg[(8C^3-8C^2-1)^2)r^3]}{\lg(1+\varepsilon)}$, In the initial configuration, there are $s' > 2s_0$ packets in total that are queued in k_1. The adversarial construction is built iteratively as follows:

0. Let $s_1 = |s'|$.
1. Apply Lemma 3.5 to get a configuration $C(s_2, \mathcal{F}(1))$, where the gadget $\mathcal{F}(1)$ has $s_2 \geq \frac{s_1(8C^3-8C^2-1)(1+\varepsilon)}{4C^3}$ packets.
2. Apply Lemma 3.6 to get a configuration where the number of packets that are queued at the output edge k_{M+1} of gadget $\mathcal{F}(M)$ is $s_3 \geq s_2 \frac{8C^3-8C^2-1}{4C^3}(1+\varepsilon)^{M-1}$.
3. Apply Lemma 3.8. This results in s_4 newly injected packets at the tail of the edge k_1 that are queued in k_1 and $s_4 \geq r^3 s_3$.
4. Let $s_1 = s_4$, and go to Step 1.

We will show that following this adversary construction the number of packets (s_1) in the system increases unboundedly. After the execution of Step (1), we have $s_2 \geq \frac{s_1(8C^3-8C^2-1)(1+\varepsilon)}{4C^3}$. After Step (2) we have that $s_3 \geq s_2 \frac{8C^3-8C^2-1}{4C^3}(1+\varepsilon)^{M-1} \geq s_1 \frac{(8C^3-8C^2-1)^2(1+\varepsilon)^M}{16C^6}$. Finally, after Step (3), we have that the number of packets stored at the tail of the edge k_1 is $s_4 \geq r^3 s_3 \geq s_1 \frac{(8C^3-8C^2-1)^2 r^3 (1+\varepsilon)^M}{16C^6}$. Choosing M to be such that $\frac{(8C^3-8C^2-1)^2 r^3(1+\varepsilon)^M}{16C^6} > 1 \Rightarrow M > \frac{\lg(16C^6)-\lg[(8C^3-8C^2-1)^2 r^3]}{\lg(1+\varepsilon)}$, we have that $s_4 > s_1$. This suffices to guarantee that the number of packet in the system increases unboundedly for an infinite time period. The instability threshold of rate r for the system ($\mathcal{G}_r$, $\mathcal{A}$, FIFO) is defined by the inequality $r > \frac{3C^2-1}{2C^3-2C}$. For C tends to infinity, the threshold of rate r tends to zero.

4 Instability in FIFO Networks with Uniform Capacities

The first of our theorems below refers to a system where all capacities are uniform i.e. when all the capacities are equal to the same value C with $C > 1$, while the second one refers to a system where all links have capacity $C = 1$.

Theorem 4.1 *Let $\mathcal{G}_r$ be a* FIFO *network and $\mathcal{A}$ an adversary of rate r, and suppose that the system ($\mathcal{G}_r$, $\mathcal{A}$,* FIFO*) is unstable for any $r > 0$ in the model of dynamic capacities. Then, there exists a system ($\mathcal{G}_r'$, $\mathcal{A}'$,* FIFO*) that is unstable in the model of uniform capacities where $\mathcal{A}'$ is an adversary of rate r.*

Sketch of proof: Theorem 4.1 plays the role of a simulator. It takes as input the unstable system ($\mathcal{G}_r$, $\mathcal{A}$, FIFO) constructed in the previous section at rate $\geq r$ where r is the injection rate of the adversary $\mathcal{A}$ and $\mathcal{G}_r$ a network whose links face changes in their capacities according to the model of dynamic capacities and gives as output an unstable system ($\mathcal{G}_r^{'}$, $\mathcal{A}^{'}$, FIFO) at rate $\geq r$ with uniform link capacities. In order to achieve this the simulator suitably changes the topology of the network $\mathcal{G}_r$ with the replacement of its edges that face changes in their capacity, with a subnetwork whose edges have uniform capacity C. We call this subnetwork *simulation bridge*. There are two different topologies $\mathcal{B}1$, $\mathcal{B}2$ for a simulation bridge (see Figure 6). The topology of the simulation bridge that we use in $\mathcal{G}_r^{'}$ to replace an edge e that faces changes in its capacity on $\mathcal{G}_r$ depends on how many packet flows with different paths want to traverse e on overlapping time intervals. If one packet flow traverses e on $\mathcal{G}_r$ or if more than one packet flows want to traverse e during the same time intervals then we use topology $\mathcal{B}1$. Otherwise, if more than one packet flows want to traverse e in overlapping but not equal time intervals then we use topology $\mathcal{B}2$ that is used in $\mathcal{G}_r^{'}$ to replace e.

Furthermore, the simulator changes the adversary $\mathcal{A}$ to $\mathcal{A}^{'}$ permitting the injection of packets that follow the not simple-path model where paths can contain overlapping edges. This change in packet paths takes place when packets enter into the simulation bridges. The adversary modifications are based on the idea of packet rerouting in two levels. In the first level we make packet rerouting at the beginning of each subphase as in the case of dynamic capacities (Section 3.2.2). In the second level we make packet rerouting of the packets that try to traverse a simulation bridge during a subphase. Packet rerouting is based on the technique introduced by Lotker *et al.* [12, Lemma 3.1]. We should mention here that while packet rerouting in the first level concerns path extensions to not overlapping edges that do not overlap with edges already present in the path (simple-path model), packet rerouting in the second level concerns path extensions to edges that do not overlap with edges already present in a path (simple-path model) but there are overlaps among them (not simple-path model). However, we carefully do this so that the total packet length of the injected packets is of bounded size. Using the idea of packet rerouting in two levels we manage to simulate the behavior of edges that face changes in their capacity on the network $\mathcal{G}_r$ by the behavior of the simulation bridges that are used to replace them on the network $\mathcal{G}_r^{'}$. In order to achieve this, it suffices to show two properties. The first property simulates the behavior of an edge when there is transition to its capacity from C to 1, while the second one simulates the behavior of an edge when there is transition to its capacity from 1 to C.

Property 4.2 *If a packet set X of t packets where $t > \frac{C^3}{C-1}$ is inserted into a simulation bridge $\mathcal{B}1$ ($\mathcal{B}2$) during a time period of $\frac{t}{C}$ steps wanting to traverse the simulation bridge $\mathcal{B}1$ ($\mathcal{B}2$), then after $\frac{t}{C}$ steps the number of packets leaving the simulation bridge is $|X^{'}| = \frac{t}{C}$ packets and all the queues in $\mathcal{B}1$ except the first one have at least C packets (the total number of packets in $\mathcal{B}2$ is more than C^2).*

Property 4.3 *If a packet set Y of t packets with $t > \frac{C^3}{C-1}$ is queued into a simulation bridge $\mathcal{B}1$ ($\mathcal{B}2$) such that all the queues in $\mathcal{B}1$ except the first one have at least C packets (or the total number of packets in $\mathcal{B}2$ is more than C^2), then during a time period of $\frac{t}{C}$ steps all the Y packets will leave the simulation bridge $\mathcal{B}1$ ($\mathcal{B}2$).*

Combining both properties we guarantee that the used simulation bridges in each gadget of $\mathcal{G}_r^{'}$ behave like the corresponding edges in $\mathcal{G}_r$ they replace. Therefore, the system ($\mathcal{G}_r^{'}$, $\mathcal{A}^{'}$, FIFO) simulates the instability behavior of the system ($\mathcal{G}_r$, $\mathcal{A}$, FIFO). Remark that in order to satisfy the additional restriction in the number of packets that traverse a simulation bridge we should assume that the initial number of packets in Section 3.2.2 is $s_0 > 4nC^{n+5}$. This guarantees that $t > \frac{C^3}{C-1}$ as $s_0 > 4nC^{n+5} > t$. Therefore this restriction is satisfied. For simplicity during the simulation we slightly relax the FIFO property for the packets that enter into a simulation bridge that simulates the transition of the capacity of an edge in $\mathcal{G}_r^{'}$ from C to 1. In our construction this relaxation takes place only among packets of the same packet flow (packets that have the same route) and have a distance between them in the flow that is a function of C. Therefore, it does not affect the behavior of the system. However, this relaxation can be removed doing simple variations in the simulation bridge topology and the paths construction. ■

Theorem 4.4 *Let $\mathcal{G}_r^{'}$ be a FIFO network and $\mathcal{A}^{'}$ an adversary of rate r, and suppose that the system ($\mathcal{G}_r^{'}$,*

$\mathcal{A}'$, FIFO*) is unstable for any* $r > 0$ *when all edges have the same capacity* $C > 1$*. Then, there exists a system (*$\mathcal{G}''_r$, $\mathcal{A}''$, FIFO*) that is unstable when all edges have unit capacities where* $\mathcal{A}''$ *is an adversary of rate* r*.*

Sketch of Proof: Theorem 4.4 plays the role of a simulator. It takes as input the unstable system ($\mathcal{G}'_r$, $\mathcal{A}'$, FIFO) described in the previous theorem whose links have uniform capacity C at rate $\geq r$ and gives as output an unstable system ($\mathcal{G}''_r$, $\mathcal{A}''$, FIFO) at rate $\geq r$ with unit link capacities. In order to achieve this it changes the topology of the network $\mathcal{G}'_r$ with the replacement of all its edges with a subnetwork called *analyzer* whose edges have unit capacity (Figure 8) and it also modifies properly the adversary in order the paths assigned to the packets to be adjusted to the modified network topology. ■

5 Discussion and Directions for Further Research

We have shown a methodology that simulates a given unstable system ($\mathcal{G}_r$, $\mathcal{A}$, FIFO) at rate $r > 0$ with dynamic capacities with an unstable system ($\mathcal{G}''_r$, $\mathcal{A}''$, FIFO) at rate $r > 0$ with unit capacities using as an intermediate step an unstable system ($\mathcal{G}'_r$, $\mathcal{A}'$, FIFO) at rate $r > 0$ with uniform capacities. This closes a major open problem (the question of FIFO stability) in the field of Adversarial Queueing Theory. Furthermore, in the model of dynamic capacities we presented a small-size network that leads FIFO to instability at a rate $r \geq 0.41$ that represents the current record for the instability threshold of FIFO over networks of fixed size (network size is independent of r). An open question that arises in this context is given an arbitrarily low injection rate r if there is a fixed-size network and an adversary for which FIFO is unstable at r.

References

[1] M. Andrews, "Instability of FIFO in session-oriented networks," *In 11th. ACM-SIAM Symposium on Discrete Algorithms,* pp. 440–447, 2000.

[2] 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.

[3] M. Andrews and L. Zhang, "The Effect of Temporary Sessions on Network Performance," *Proceedings of the 11th Annual ACM-Symposium on Discrete Algorithms,* pp. 448–457, January 2000.

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

[5] 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.

[6] 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,* January 2001.

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

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

[9] 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.

[10] F. P. Kelly, *Reversability and stochastic networks,* Wiley, New York, 1979.

[11] D. Koukopoulos, M. Mavronicolas, S. Nikoletseas and P. Spirakis, "On the Stability of Compositions of Universally Stable, Greedy, Contention-Resolution Protocols," *16th International Symposium on DIStributed Computing (DISC'02),* LNCS 2508, pp. 88–102, October 2002.

[12] 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.

[13] A. N. Rybko, and A. L. Stolyar, "Ergodicity of Stochastic Processes Describing the Operation of Open Queueing Networks," *Problems of Information Transmission,* Vol. 28, pp. 199–220, 1992.

[14] P. Tsaparas, *Stability in Adversarial Queueing Theory,* M.Sc. Thesis, Computer Sc. Dept., Univ. of Toronto, 1997.

Appendix

A Proof of Lemma 3.3

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 do 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 $|Rem| = t - \frac{R_n t}{R_n+r} = t - \frac{\frac{1-r}{1-r^n}t}{r+\frac{1-r}{1-r^n}} = rt\frac{1-r^n}{1-r^{n+1}} \leq rt$.

B Proof of Lemma 3.4 (Evolution of the system configuration)

During time interval $[1, \frac{2s_i}{C} + \frac{2(C-1)s_i}{C^2} + n]$ the S packets that are queued in the system at the initial system configuration traverse their path. The first packets of set S arrive in queue x'_{i+1} after the first n steps of this time interval as S packets have to traverse the chain of edges $k_{i+1}, f_{i+1,1}, \ldots, f_{i+1,n-2}, z_{i+1}$ that have capacity C. During time interval $[n+1, \frac{2s_i}{C} + n]$ the S packets are delayed in queue x'_{i+1} because x'_{i+1} has 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} requiring to traverse the edges $e_{i+1,0}, \ldots, e_{i+1,n}, k_{i+2}$.

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 time step $\frac{2s_i}{C} + \frac{2(C-1)s_i}{C^2} + n$ because this edge has capacity 1. Therefore, $|S_4| = \frac{2(C-1)s_i}{C^2}$ packets from S_1 packets can traverse edge $e_{i+1,0}$ in $\frac{2(C-1)s_i}{C^2}$ time steps. Hence, during time interval $[n+1, \frac{2s_i}{C} + \frac{2(C-1)s_i}{C^2} + n]$, except S_4 packets, the number of packets which arrive (in total) in the path $e_{i+1,1}, \ldots, e_{i+1,n}$ and they can traverse it is $|S_5| = |S_2| + |S_4|$ packets.

From Lemma 3.3 if a packet set L of t packets is inserted into a path of length n during a time period of $t+n$ steps, then there is an adversary of rate r, such that the number of packets remaining into the system is $|L'| \leq rt$, all the path edges have at least one packet and only L' packets are queued into the queues of the path at the end of the time period. In our case $S_5 = \frac{2s_i}{C} + \frac{2(C-1)s_i}{C^2}$ packets want to traverse the path of n edges $e_{i+1,1}, \ldots, e_{i+1,n}$ in $\frac{2s_i}{C} + \frac{2(C-1)s_i}{C^2}$ time steps. From these packets $|H| = n$ packets arrive in queue $e_{i+1,1}$ at the last n steps of the subphase. Since the adversary operates according to Lemma 3.3 these packets remain queued there till the end of the subphase 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 $e_{i+1,1}, \ldots, e_{i+1,n}$, such that all these queues are not empty at time step $\frac{2s_i}{C} + \frac{2(C-1)s_i}{C^2} + n$.

The X packets that are injected in queue x_i during time interval $[1, \frac{2s_i}{C} + n]$ are blocked by the initial packets in queue k_{i+1} till time step $\frac{2s_i}{C}$. This happens because in $\frac{2s_i}{C}$ time steps all the S packets traverse the edge k_{i+1} in their path as all the edges they traverse till the edge k_{i+1} have capacity C. During the rest n time steps, all the S packets have traverse the path $f_{i+1,1}, \ldots, f_{i+1,n-2}, z_{i+1}$ along with nC packets from X packets the first of which are queued in queue $f_{i+1,n}$ at time step $\frac{2s_i}{C} + n$.

During time interval $[\frac{2s_i}{C} + n + 1, \frac{2s_i}{C} + \frac{2(C-1)s_i}{C^2} + n]$ all the X packets traverse their path arriving in queue x_{i+1} as the number of X packets is $|X| = \frac{2(C-1)rs_i}{C}$, this time interval has duration $\frac{2(C-1)s_i}{C^2}$ and all the edges in $\mathcal{F}_i$ and the edges $k_{i+1}, f_{i+1,1}, \ldots, f_{i+1,n}$ have capacity C. 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 packets from the X packets. The number of $X^{'}$ packets is $|X^{'}| = 2s_i[r\frac{C-1}{C} - \frac{C-1}{C^2+C}]$.

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 blocked by the S_1 packets in queue $x^{'}_{i+1}$. This happens because in this time interval $\frac{(2C-2)s_i}{C}$ packets can traverse this edge that has capacity C which is equal to the number of S_1 packets that are queued in $x^{'}_{i+1}$ at the beginning of this time interval.

C Proof of Lemma 3.5

Consider the network $\mathcal{G}_r$ in Figure 5. At time t there is a set S of $|S| = 2s$ packets queued in the queue k_1. We want to show that for any $r > \varepsilon > 0$, $r > \frac{1}{C}$, $C > n > \max\{\frac{\lg(\varepsilon)-\lg(2)}{\lg(r)}, 1 - \frac{1}{\lg(r)}\}$ and $s_0 > 4nC^3$, there is an adversary $\mathcal{A}$ of rate r with $r = \frac{3C^2-1}{2C^3-2C} + \varepsilon$, such that at time $t_1 = t + \frac{2s}{C} + \frac{(2C-2)s}{C^2} + n$ the final system configuration will be $C^{t_1}(s^{'}, \mathcal{F}(1))$, for $s^{'} \geq s(1+\varepsilon)$. Assume for convenience that $t = 0$.

The adversary makes injections in a time period T with duration $|T| = \frac{2s}{C} + \frac{2(C-1)s}{C^2} + n$. During this time period all the edges of the network $\mathcal{N}$ have capacity C except some edges that have unit capacity in specific time intervals of period T:

- The edge $x^{'}_1$ and the edges $e_{1,0}, \ldots, e_{1,n}$ have unit capacity in time interval $[1, \frac{2s}{C} + n]$.
- The edge x_1 and the edges $e_{1,0}, \ldots, e_{1,n}$ have unit capacity in time interval $[\frac{2s}{C} + n + 1, \frac{2s}{C} + \frac{2(C-1)s}{C^2} + n]$.

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

- During time interval $[1, \frac{2s}{C} + n]$ the adversary injects a set X of $|X| = \frac{2(C-1)rs}{C}$ packets in queue k_1 requiring to traverse the edges $k_1, f_{1,1}, \ldots, f_{1,n-2}, z_1, y_1, k_2$.
- During time interval $[n + 1, \frac{2s}{C} + \frac{2(C-1)s}{C^2} + n]$ the adversary makes injections into the path $e_{1,1}, \ldots, e_{1,n}$, based on Lemma 3.3.
- During time interval $[\frac{2s}{C} + n + 1, \frac{2s}{C} + \frac{2(C-1)s}{C^2} + n]$ the adversary injects a set Y of $|Y| = \frac{2r(C-1)s}{C}$ packets in queue $x^{'}_1$ requiring to traverse the edges $x^{'}_1, y_1, k_2$.

- During time interval $[\frac{2s}{C}+n+1, \frac{2s}{C}+\frac{2(C-1)s}{C^2}+n]$ the adversary injects a set Z of $|Z| = \frac{2r(C-1)}{C^2}$ packets in queue x_1 requiring to traverse the edge x_1.

Evolution of the system configuration. During time interval $[1, \frac{2s}{C}+\frac{2(C-1)s}{C^2}+n]$ the S packets that are queued in the system at the initial system configuration traverse their path. The first packet of set S arrive in queue $x_1^{'}$ after the first n steps of this time interval as S packets have to traverse the chain of edges $k_1, f_{1,1}, \ldots, f_{1,n-2}, z_1$ that have capacity C. During time interval $[n+1, \frac{2s}{C}+n]$ the S packets are delayed in queue $x_1^{'}$ because $x_1^{'}$ has unit capacity. Therefore, a number of $|S_1| = \frac{2(C-1)s}{C}$ packets remain in queue $x_1^{'}$ at time step $\frac{2s}{C}+n$ requiring to traverse the edges $x_1^{'}, e1, 0, e_{1,1}, \ldots, e_{1,n}, k_2$, while $|S_2| = \frac{2s}{C}$ packets traverse the edges $x_1^{'}$ requiring to traverse the edges $e_{1,0}, \ldots, e_{1,n}, k_2$.

At the rest $\frac{2(C-1)s}{C^2}$ time steps of the time interval $[1, \frac{2s}{C}+\frac{2(C-1)s}{C^2}+n]$ the S_1 packets traversing their path arrive in the queue $e_{1,0}$. From S_1 packets, a set S_3 of $|S_3| = 2s\frac{(C-1)^2}{C^2}$ packets remain queued in queue $e_{1,0}$ at time step $\frac{2s}{C}+\frac{2(C-1)s}{C^2}+n$ because this edge has capacity 1. Therefore, $|S_4| = \frac{2(C-1)s}{C^2}$ packets from S_1 packets traverse the edge $e_{1,0}$ in $\frac{2(C-1)s}{C^2}$ time steps arriving in queue $e_{1,1}$. Hence, during time interval $[n+1, \frac{2s}{C}+\frac{2(C-1)s}{C^2}+n]$, except S_4 packets, the number of packets which arrive (in total) in the path $e_{1,1}, \ldots, e_{1,n}$ and can traverse it is $|S_5| = |S_2| + |S_4|$ packets.

From Lemma 3.3 if a packet set L of t packets is inserted into a path of length n during a time period of $t+n$ steps, then there is an adversary of rate r, such that the number of packets remaining into the system is $|L^{'}| \leq rt$, all the path edges have at least one packet and only $L^{'}$ packets are queued into the queues of the path at the end of the time period. In our case $S_5 = \frac{2s}{C}+\frac{2(C-1)s}{C^2}$ packets want to traverse the path of n edges $e_{1,1}, \ldots, e_{1,n}$ in $\frac{2s}{C}+\frac{2(C-1)s}{C^2}$ time steps. From these packets $|H| = n$ packets arrive in queue $e_{i+1,1}$ at the last n steps of the subphase. Since the adversary operates according to Lemma 3.3 these packets remain queued there till the end of the subphase and a number of $|S_7| = [2s\frac{2C-1}{C^2}-n]\frac{r-r^{n+1}}{1-r^{n+1}}$ packets are preserved into the queues $e_{1,1}, \ldots, e_{1,n}$, such that all these queues are not empty at time step $\frac{2s}{C}+\frac{2(C-1)s}{C^2}+n$.

The X packets that are injected in queue k_1 during time interval $[1, \frac{2s}{C}+n]$ are blocked by the initial packets in queue k_1 till time step $\frac{2s}{C}$. This happens because in $\frac{2s}{C}$ time steps all the S packets traverse the edge k_1 as the edge k_1 has capacity C. During the rest n time steps, all the S packets have traverse the path $f_{1,1}, \ldots, f_{1,n-2}, z_1$ along with nC packets from X packets the first of which are queued in queue $f_{1,n}$ at time step $\frac{2s}{C}+n$.

During time interval $[\frac{2s}{C}+n+1, \frac{2s}{C}+\frac{2(C-1)s}{C^2}+n]$ all the X packets traverse their path arriving in queue x_1 as the number of X packets is $|X| = \frac{2(C-1)rs}{C}$, this time interval has duration $\frac{2(C-1)s}{C^2}$ time steps and the edges $k_1, f_{1,1}, \ldots, f_{1,n}$ have capacity C. In queue x_1 X packets are mixed with Z packets. This mixing along with the unit capacity of edge x_1 results in the delay (in x_1) of a portion $X^{'}$ of packets from the X packets. The number of $X^{'}$ packets is $|X^{'}| = 2s[r\frac{C-1}{C} - \frac{C-1}{C^2+C}]$.

The Y packets that are injected in queue $x_1^{'}$ during time interval $[\frac{2s}{C}+n+1, \frac{2s}{C}+\frac{2(C-1)s}{C^2}+n]$ are blocked by the S_1 packets in queue $x_1^{'}$. This happens because in this time interval $\frac{(2C-2)s}{C}$ packets can traverse this edge that has capacity C which is equal to the number of S_1 packets that are queued in $x_1^{'}$ at the beginning of this time interval.

At the end of this period, the number of packets in queue $e_{1,j}$ for $1 \leq j \leq n$ 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 $2s_{i+1} = X^{'} + Y + S_3 + H + S_7$.

From Lemma 3.4 this number of packets is larger than the number of initial packets. Thus, $2s^{'} \geq (1+\varepsilon)2s$ for $r = \frac{3C^2-1}{2C^3-2C}+\varepsilon$. Therefore, for any $r > \varepsilon > 0$, $r > \frac{1}{C}$, $C > n > \max\{\frac{\lg(\varepsilon)-\lg(2)}{\lg(r)}, 1-\frac{1}{\lg(r)}\}$

and $s_0 > 4nC^3$ at time $t_1 = t + \frac{2s}{C} + 2\frac{(C-1)s}{C^2} + n$ the final system configuration will be $C^{t_1}(s^{'}, \mathcal{F}(1))$, for $s^{'} \geq s(1+\varepsilon)$.

D Proof of Lemma 3.6

In order to complete the proof of Lemma 3.6, we consider Claim 3.7. Then, at time t_M the system configuration is $C^{t_M}(s^{'}, \mathcal{F}(M))$ for $s^{'} \geq s(1+\varepsilon)^{M-1}$. If we do not make any injection in the time interval $[t_M, t_M + \frac{2s^{'}}{C} + 1]$ and consider that all the edges have capacity C except the output edge k_{M+1} of the gadget $\mathcal{F}(M)$ that has unit capacity, then the $2s^{'}$ packets that have been queued at the queues of the gadget $\mathcal{F}(M)$ at time t_M will arrive at the output edge k_{M+1} of the gadget $\mathcal{F}(M)$. Furthermore, $\frac{2s^{'}}{C} + 1$ packets depart from the output edge k_{M+1} during the time interval $[t_M, t_M + \frac{2s^{'}}{C} + 1]$. Therefore, at time $t_M + \frac{2s^{'}}{C} + 1$, there are $s^{'} = 2s^{'} - \frac{2s^{'}}{C} - 1 \geq 2s_0 - \frac{2s_0}{C} - 1$ packets in the output edge k_{M+1}. If we consider $1 < n < \frac{s_0}{4C^3}$, then $s^{'} \geq 2s_0 - \frac{2s_0}{C} - \frac{s_0}{4C^3} = \frac{(8C^3-8C^2-1)s_0}{4C^3}$. However, because $s \geq s_0$ and $s^{'} \geq s(1+\varepsilon)^{M-1}$, then $s^{'} \geq \frac{(8C^3-8C^2-1)s}{4C^3}(1+\varepsilon)^{M-1}$ packets exist at the output edge k_{M+1} of the gadget $\mathcal{F}(M)$.

E Proof of Claim 3.7

Base case. For $i = 1$, the claim is trivial with $t_1 = \tau$. *Induction step.* Consider that there is an adversary $\mathcal{A}_\rangle$ of rate r and some time $t_i \geq \tau$, such that the system configuration is $C^{t_i}(s_i, \mathcal{F}(i))$ for $s_i > s(1+\varepsilon)^{i-1}$. Let now consider a subnetwork that consists of the chain of two gadgets $\mathcal{F}(i)$ and $\mathcal{F}(i+1)$. Applying Lemma 3.4, there is an adversary $\mathcal{A}_\rangle$ of rate r and a time period T_i, such that in the model of dynamic capacities the system configuration at time $t_i + T_i$ is $C^{t_i+T_i}(s_{i+1}, \mathcal{F}(i+1))$ for $s_{i+1} \geq s_i(1+\varepsilon) \geq s(1+\varepsilon)^i$. Note that at time $t_i + T_i$ the packets in the system are only queued in the gadget $\mathcal{F}(i+1)$ from Lemma 3.4. If we assign $t_{i+1} = t_i + T_i = t_i + \frac{2s_i}{C} + 2\frac{(C-1)s_i}{C^2} + n$ and concatenate the adversaries $\mathcal{A}$ and $\mathcal{A}_\rangle$ the claim has been proved.

F Proof of Lemma 3.8

Consider the network $\mathcal{G}_r$ in Figure 5. At time t there is a set S of $|S| = s$ packets queued in the queue k_{M+1} of the gadget $\mathcal{F}(M)$ requiring to traverse the edge k_{M+1}. We will show that for any $r > 0$ there is an adversary $\mathcal{A}$ of rate r, such that at time $t_1 = t + \frac{s}{C} + r\frac{s}{C} + r^2\frac{s}{C}$ all the packets in the system are the $r^3 s$ packets that are queued in k_1. These packets have been injected in k_1 after time t. We consider that all the edges have capacity C during time interval $(t, t_1]$. The adversary plays three rounds of injections as follows:

- *Round 1:* This round lasts for $\frac{s}{C}$ time steps. During this round the edges k_{M+1}, e_0, k_1 have capacity C. The adversary injects a set X of $|X| = r\frac{sC}{C} = rs$ 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 packets that are queued in k_{M+1} at the beginning of this round. The S packets have been absorbed at the end of this round.
- *Round 2:* This round lasts for $\frac{rs}{C}$ time steps. During this round the edges k_{M+1}, e_0, k_1 have capacity C. The adversary injects a set Y of $|Y| = r\frac{rsC}{C} = r^2 s$ 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.

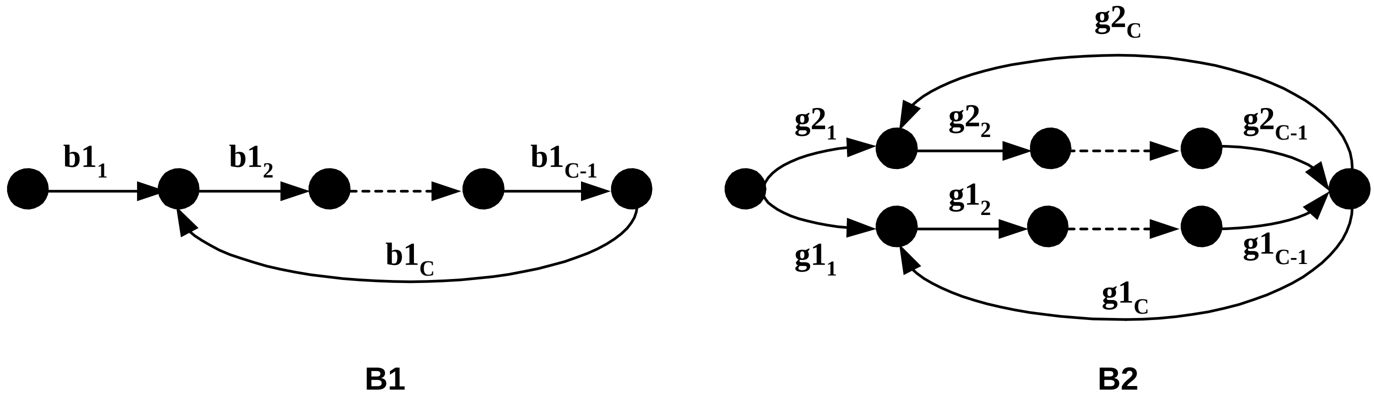

Figure 6: The simulation bridge $\mathcal{B}1$ and $\mathcal{B}2$

At the end of this round, there is a set Z of $|Z| = r^2 s$ 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 $\frac{r^2 s}{C}$ time steps. During this round the edge k_1 has capacity C. The adversary injects a set L of $|L| = r\frac{r^2 sC}{C} = r^3 s$ 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 $t + \frac{s}{C} + r\frac{s}{C} + r^2\frac{s}{C}$ all the packets in the system are the $|L| = r^3 s$ packets that have been injected in k_1 during this round and they are queued in k_1.

G Proof of Theorem 4.1

The Simulation Bridge topology. There are two different topologies $\mathcal{B}1$, $\mathcal{B}2$ for a simulation bridge (see Figure 6). The topology of the simulation bridge that we use in network $\mathcal{G}_r^{'}$ to replace an edge e that faces changes in its capacity on network $\mathcal{G}_r$ depends on how many packet flows with different paths want to traverse e on overlapping time intervals. If one packet flow traverses e on $\mathcal{G}_r$ or if more than one packet flows want to traverse e during the same time intervals then the topology of the simulation bridge $\mathcal{B}1$ we use in the network $\mathcal{G}_r^{'}$ to replace e consists of:

- a chain of $C-1$ edges $b1_j$ where $1 \leq j \leq C-1$,
- an edge $b1_C$, which source is common with the destination of the edge $b1_{C-1}$ and the destination is common with the source of the edge $b1_2$.

Otherwise, if more than one packet flows want to traverse e in overlapping but not equal time intervals then the topology of the simulation bridge $\mathcal{B}2$ we use in the network $\mathcal{G}_r^{'}$ to replace e consists of:

- two parallel chains of $C-1$ edges $g1_j$ and $g2_j$ with common source and destination where $1 \leq j \leq C-1$,
- one edge $g1_C$, which source is common with the destination of the edge $g1_{C-1}$ and destination is common with the source of the edge $g1_2$,

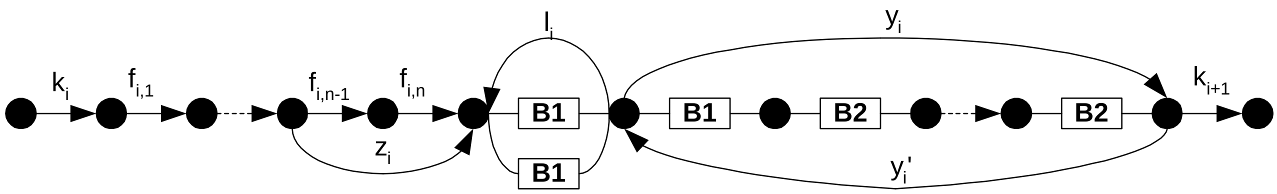

Figure 7: A gadget of $\mathcal{G}_r^{'}$

- one edge $g2_C$, which source is common with the destination of the edge $g2_{C-1}$ and destination is common with the source of the edge $g2_2$.

The Network topology. The network $\mathcal{G}_r^{'}$ has the same structure as the network $\mathcal{G}_r$ (see Figure 5). Hence, it is a cascade of $M^{'}$ similar subnetworks $\mathcal{F}^{'}(i)$ called *gadgets* (see Figure 3) where $1 \leq i \leq M^{'}$. The gadgets of the network $\mathcal{G}_r^{'}$ have similar topology with the ones of the network $\mathcal{G}_r$ with one difference. All the edges of each gadget in $\mathcal{G}_r$ that face changes in their capacity during an execution of the system ($\mathcal{G}_r$, $\mathcal{A}$, FIFO) are replaced by a simulation bridge. Therefore, the i^{th} gadget, $\mathcal{F}(i)^{'}$ (see Figure 7), of the network $\mathcal{G}_r^{'}$ where $1 \leq i \leq M^{'}$ is the same as the i^{th} gadget, $\mathcal{F}(i)$ (Section 3.2.1, gadget topology), of the network $\mathcal{G}_r$ where $1 \leq i \leq M$ with the following differences:

- Each of the edges $x_i, x_i^{'}, e_{i,0}$ are replaced by a simulation bridge $\mathcal{B}1$,
- Each edge of the chain of n edges $e_{i,j}$ where $1 \leq j \leq n$ are replaced by a simulation bridge $\mathcal{B}2$.

Network size. The size of each gadget is $Gs = 3C + n(2C+1) + 6$ where $C > n > \max\{\frac{\lg(\varepsilon)-\lg(2)}{\lg(r)}, 1 - \frac{1}{\lg(r)}\}$. Thus, $Gs > 3C + (2C+1)(\max\{\frac{\lg(\varepsilon)-\lg(2)}{\lg(r)}, 1 - \frac{1}{\lg(r)}\}) + 6$. Furthermore, the network $\mathcal{G}_r^{'}$ consists of a number $M^{'}$ of gadgets such that $M^{'} > \frac{\lg(16C^{2(n+5)})-\lg[(8C^{n+5}-8C^{n+4}-1)^2 r^3]}{\lg(1+\varepsilon)}$ for any $r > \varepsilon > 0$. Hence, the size of the network is polynomial in $\frac{1}{\lg(r)}$.

The Adversary

The adversary $\mathcal{A}^{'}$ of the system $\mathcal{G}_r^{'}, \mathcal{A}^{'}$, FIFO is similar to the adversary $\mathcal{A}$ of the system $\mathcal{G}_r, \mathcal{A}^{'}$, FIFO (see Section 3.2.2). The basic difference here is that the adversary can assign paths to the injected packets that contain overlapping edges (non-simple path model). Based on this property of the adversary, we define packet rerouting differently here.

Packet rerouting in two levels. Besides packet rerouting at the beginning of each subphase (see Section 3.2.2) there is also a second level of packet rerouting during each subphase when the packets arrive in the simulation bridges of the next gadget to which they are queued at the beginning of the subphase. As in Section 3.2.2 at the beginning of each subphase the adversary assigns to the packets that are queued into the system an extension to their path, which consists of non-overlapping edges that, also, do not overlap with the path that have been already traversed. The new path covers edges of the gadget where the packets at the beginning of a subphase are queued and edges of the next gadget. However, when the packets arrive in the simulation bridges of the next gadget to which they are queued at the beginning of each subphase, they are rerouted by the adversary with paths that consist of overlapping edges of the simulation bridges. Note that packet rerouting takes place not only when packets are inserted into a simulation bridge but also after packets complete a full cycle traversing the edges of the simulation bridge.

In order to explain better this second level of packet rerouting consider the simulation bridge $\mathcal{B}1(i)$ of the gadget $\mathcal{F}^{'}(i)$ on $\mathcal{G}_r^{'}$ that replaces the edge x_i in the $\mathcal{F}(i)$ gadget of network $\mathcal{G}_r$. The paths of the packets that arrive in the simulation bridge $\mathcal{B}1(i)$ of the gadget $\mathcal{F}^{'}(i)$ traversing the gadget $\mathcal{F}^{'}(i-1)$, where $1 \leq i \leq M$ during a subphase, are extended as follows:

- The first C packets that arrive in queue $b1(i,1)$ are assigned to traverse the edges $b1(i,1),\dots,b1(i,C-1)$.
- The rest $(C-1)C$ packets that arrive in queue $b1(i,1)$ are assigned to traverse the edges $b1(i,1),\dots,$ $b1(i,C-1), b1(i,C),\dots,b1(i,2)$.
- All the next packets are assigned to traverse the edges $b1(i,1), b1(i,2)$. These packets are mixed in $b1(i,2)$ with the packets that come from queue $b1(i,C)$.
- All the packets that arrive in $b1(i,2)$ from queues $b1(i,1), b1(i,C)$ are rerouted such that the first C packets are assigned to traverse the edges $b1(i,2),\dots,b1(i,C-1)$, while the rest $(C-1)C$ packets that arrive in queue $b1(i,2)$ are assigned to traverse the edges $b1(i,2),\dots,b1(i,C-1), b1(i,C),\dots,b1(i,2)$.

Notice that the simulation bridge $\mathcal{B}2$ is used in $\mathcal{G}_r^{'}$ to replace edges in $\mathcal{G}_r$ that not only face capacity changes but also they are traversed by different packet flows in different but overlapping time intervals. To explain this better consider the simulation bridge $\mathcal{B}2(i,j)$ in $\mathcal{G}_r^{'}$ that replaces the edge $e_{i,j}$ in the i^{th} gadget of network $\mathcal{G}_r$ where $1 \leq i \leq M$ and $1 \leq j \leq n$. Also consider that two packet flows X and Y arrive in $\mathcal{B}2(i,j)$ wanting to traverse it in the time intervals $[t_1,t_2]$ and $[t_1,t_3]$ with $t_2 < t_3$ respectively. Then the paths of the packets that arrive in the simulation bridge $\mathcal{B}2(i,j)$ are extending in two phases. During the first phase that covers the time interval $[t_1,t_2]$ the packet paths are extended as in the simulation bridge $\mathcal{B}1(i)$ above as only the edges $g1(i,j)$ are used. During the second phase that covers the interval $[t_2,t_3]$ only packets from set Y are inserted into the simulation bridge. Then, packet paths are extended as follows:

- The packet rerouting (similar to the first phase) for the packets that are queued in the edges $g1(i,1),\dots,g1(i,C)$ continues till time step t_3 or till all the packets leave the simulation bridge $\mathcal{B}2(i,j)$.
- The packets that inserted into the simulation bridge during the interval $[t_2,t_3]$ are forwarded into queue $g2(i,1)$. These packets traverse the edges $g2(i,1),\dots,g2(i,C), g2(i,2)\dots,g2(i,C)$ till time t_3 or till the last packets in queues $g1i,j$ leave $\mathcal{B}2(i,j)$. If the last packets in queues $g1i,j$ leave $\mathcal{B}2(i,j)$ before t_3 then the second C packets in queue $g2i,2$ are assigned to traverse the edges $g2(i,2),\dots,g1(i,C-1)$, while the rest packets that arrive in $\mathcal{B}2(i,j)$ are assigned to traverse the edges $g2(i,1), g2(i,2)$. These packets are mixed in $g2(i,2)$ with the packets that come from queue $g2(i,C)$.
- All the packets that arrive in $g2(i,2)$ from queues $g2(i,1), g2(i,C)$ are rerouted such that the first C packets are assigned to traverse the edges $g2(i,2),\dots,g2(i,C-1)$, while the rest $(C-1)C$ packets that arrive in queue $g2(i,2)$ are assigned to traverse the edges $g2(i,2),\dots,g2(i,C-1), g2(i,C),\dots,g2(i,2)$.

Note that in both simulation bridges after the end of the time period in which the corresponding edge in $\mathcal{G}_r$ has capacity 1 and returns to capacity C all the packets that are queued in the queues $b1(j)$ $(g1(j), g2(j))$ where $1 \leq j \leq C$ of the simulation bridge $\mathcal{B}1$ $(\mathcal{B}2)$ are rerouted to follow the path $b1(j),\dots,b1(C-1)$ $(g1(j),\dots,g1(C-1)$ and $g2(j),\dots,g2(C-1))$.

Simulating the Transition from Capacity C to 1 (Proof of Property 4.2). The adversary using the packet rerouting property in the second level as it has been described previously achieves C packets

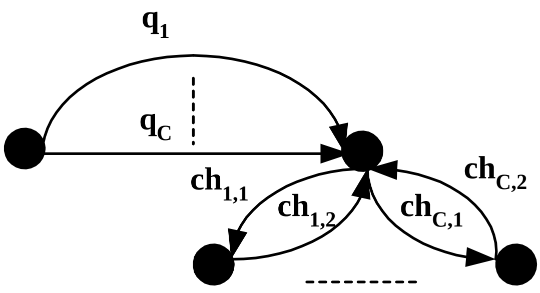

Figure 8: The analyzer

to leave the simulation bridge $\mathcal{B}1$ ($\mathcal{B}2$) every C time steps. Therefore, in a period of $\frac{t}{C}$ time steps $\frac{t}{C}$ packets will leave the simulation bridge $\mathcal{B}1$ ($\mathcal{B}2$). The number of packets that remain into the simulation bridge after $\frac{t}{C}$ time steps is $|X''| = \frac{(C-1)t}{C}$. In order all the queues in $\mathcal{B}1$ except the first one to have at least C packets after $\frac{t}{C}$ time steps (the total number of packets in $\mathcal{B}2$ is more than C^2) we should have $\frac{(C-1)t}{C} > C^2 \Rightarrow t > \frac{C^3}{C-1}$ because the simulation bridge $\mathcal{B}1$ has C edges with capacity C (the simulation bridge $\mathcal{B}2$ has two parallel chains of C edges each one with capacity C). This has as a result to need to remain at least C^2 packets in the simulation bridge after $\frac{t}{C}$ time steps.

Simulating the Transition from Capacity 1 **to** C **(Proof of Property 4.3).** All the edges have capacity C. Furthermore, the topology of each one of the simulation bridges is such that any packet have to traverse at most C edges in order to leave it. Also, every queue in $\mathcal{B}1$ except the first one has at least C packets (the total number of packets in $\mathcal{B}2$ is more than C^2). Therefore, in 1 time step C packets will leave the simulation bridge. Hence in $\frac{t}{C}$ steps all the Y packets will leave the simulation bridge.

H Proof of Theorem 4.4

Theorem 4.4 plays the role of a simulator. The simulator suitably changes the topology of the network $\mathcal{G}_r^{'}$ with the replacement of all its edges with a subnetwork called *analyzer* whose edges have unit capacity. The *analyzer* (Figure 8) consists of C parallel edges q_l with unit capacity that have common source and destination where $1 \leq l \leq C$. In the destination of the analyzer for each edge q_l there is a small chain of two edges $ch(l,k)$ where $1 \leq k \leq 2$ that has as source and destination the destination of the analyzer. The additional chains of two edges in the analyzer 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 in the analyzer one in each queue q_l. This happens when the corresponding edge replaced by the analyzer in the system ($\mathcal{G}_r^{'}$, $\mathcal{A}^{'}$, FIFO) has a packet flow queued into it at some time and a new packet flow is inserted into it which should leave this edge after all the already queued packets in it leave it. Furthermore, the simulator modifies the adversary $\mathcal{A}^{'}$ to $\mathcal{A}^{''}$ in order to deal with the new network structure. The main modification of the adversary $\mathcal{A}^{'}$ that suffices for the simulation of the system ($\mathcal{G}_r^{'}$, $\mathcal{A}^{'}$, FIFO) by the system ($\mathcal{G}_r^{''}$, $\mathcal{A}^{''}$, FIFO) is that the adversary now instead of injecting a packet flow of rCt packets into a queue of $\mathcal{G}_r^{'}$ with capacity C during t time steps, it injects rt packets into each queue q_l of the corresponding analyzer in $\mathcal{G}_r^{''}$ with unit capacity requiring to traverse it. Moreover, the adversary $\mathcal{A}^{''}$ simulates the behavior of an edge e in $\mathcal{G}_r^{'}$ which blocks a packet flow X into it till all the packets of a flow Y that are already queued in it leave. This simulation is achieved by forwarding the first C packets of the packet flow X that enter the queues q_l of the analyzer where the packets of Y are already queued 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'