Alcom-FT Technical Report Series
ALCOMFT-TR-02-86

The Impact of Structure on Stability in Adversarial Queueing Theory*

Dimitrios K. Koukopoulos[†] *Marios Mavronicolas*[‡] *Paul G. Spirakis*[§]

(March 6, 2002)

Abstract

In this paper, we continue the study of *stability* issues in the *Adversarial Queueing Theory* model for packet-switched routing in *communication networks.* We are interested in revealing structural properties of the networks that critically determine the stability or instability properties of common protocols, such as FIFO and NTG, running of them. We obtain the following structural results:

- We present an improved analysis of FIFO stability for *any* given network; this analysis relies on a careful consideration of the precise role played by several structural (graph-theoretic) parameters of the network, such as the maximum number of disjoint paths used by the adversary in any given execution, the maximum directed path length and the maximum in-degree of the underlying graph. The end result of our analysis is an improved upper bound on the adversary's *injection rate,* as a function of structural parameters of the network, that suffices to guarantee that FIFO is stable on the network; this result represents a substantial improvement over a corresponding result shown recently by Diaz *et al.* [8].

  To illustrate the strength and applicability of our analytical techniques, we first apply them to a particular, simple network with just three queues, originally studied in [10]. The application yields that this network is stable for FIFO against any adversary with injection rate no more than 0.2357; this represents an improvement over a corresponding result shown in [8].
- On the negative side, we are interested in determining structural properties of a network, stated in terms of *forbidden graph minors,* that suffice to imply that the network is not *universally stable* (i.e., it is not stable for *any* protocol in a given class). We introduce two new graph minors, and we show that they are forbidden (as graph minors) for any universally stable network; furthermore, we analyze their *instability thresholds:* these are lower bounds on the adversary's injection rate above which the graph minors are unstable. Interestingly, each of these graph minors is a subgraph of some graph minor in a family of three that were claimed in [10] to be forbidden graph minors for any universally stable network. Moreover, none of the three graph minors in [10] is a subgraph of the forbidden minors we identify, and this implies that those three graph minors do not really represent a structural characterization of universal stability, as it was originally claimed in [10].

*Submitted as a **Regular Presentation**. 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"), 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, P. O. Box 1122, 261 10 Patras, Greece. 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

1 Introduction

1.1 Motivation-Framework

In this paper, we continue thw study of *stability* issues in the *Adversarial Queueing* model, which was recently introduced in a pioneering paper by Borodin *et al.* [3], and further studied in, e.g., [2, 8, 10, 12, 13, 14]. In essence, the adversarial queueing model considers packet-switched *communication networks,* where packets arrive dynamically at the nodes and they are routed in discrete time steps across the edges; thus, it replaces traditional stochastic assumptions made in Queueing Theory (see, e.g., [5, 11] for textbook-style discussions) by the more robust, worst-case ones.

Roughly speaking, a (packet-switched) communication network is *stable* if the number of packets in it remains bounded, in the course of running the system arbitrarily long. Naturally, stability depends critically on the *rate* at which packets arrive into the system, and on the contention-resolution *protocol* employed when two or more packets compete to cross a given edge at the same time step.

The underlying goal is to determine whether it is feasible to prove stability even when packets are injected by an adversary, rather than by an oblivious randomized process. Adversarial Queueing Theory considers the time evolution of a packet-routing network as a game between an adversary and a protocol. The adversary, at each time step, may inject a set of packets in some nodes. For each packet, the adversary specifies a *simple* network path that the packet must traverse and, when the packet arrives to its final destination, it is absorbed by the system. If more than one packet wish to cross an edge e at the current time step, then a contention resolution protocol is used to resolve the conflict. We use the equivalent term policy or service discipline for such a protocol.

A crucial parameter of the adversary is its *injection rate.* The rate of an adversary in this model, is specified by a pair (r, b) where $b \geq 1$ is a natural number and $0 < r < 1$. The adversary must obey the following rule: "**Of the packets that the adversary injects in any interval I, at most $\lceil r|I| \rceil + b$ can have paths that contain any particular edge.**" Such a model allows for adversarial injection of packets that are "bursty". In addition, this model implies that in any interval I, the adversary cannot inject more than $\lceil r|I| \rceil + b$ packets belonging to one or more sets of packets with paths that have common edges even if packets from different sets do not want to traverse simultaneously any common edge.

In this paper, we consider only *greedy* (also known as *work-conserving*) protocols - those that advance a packet across an edge e whenever there is at least one packet waiting to use e. We, also, consider discrete time units (time steps) $t = 0, 1, 2, \ldots$. In each time unit a queue can serve exactly one packet. In particular, we study the stability properties of "First-In-First-Out" (FIFO) policy that gives priority at every queue to the packet that has arrived first in the queue. Also, in the second part of the paper considering the stability of networks based on structural conditions, we use the "Nearest-To-Go" (NTG) policy that gives priority at every queue to the packet that has to traverse the smallest number of queues, and a variation of this policy NEAREST-TO-GO-USING-LONGEST-IN-SYSTEM-FOR-TIES-RESOLUTION (NTG-U-LIS) that gives priority at every queue to the packet that has to traverse the smallest number of queues, while if there are two or more packets that have to traverse the same number of edges it gives priority to the oldest packet in the system.

The motivation for study the behavior of packet-switched networks and especially FIFO networks is to determine the conditions of *stability,* under which the number of packets in the system remains bounded as the system dynamically evolves in time. Especially, we examine FIFO protocol as FIFO is the most popular queueing discipline in todays networks because of its simplicity. Also, besides FIFO protocol stability another interesting area is network stability, which investigates the structural conditions, which when appear on graphs, characterize their stability. If these conditions are specified, they are important because they characterize the stability behavior of whole classes of graphs, where these structural conditions hold.

In [2, 3], the authors highlight a very basic algorithmic question: *When is a given contention-resolution protocol stable in a given network, against a given adversary?* More specifically, such kind of questions are based on the following definitions.

Definition 1.1 *We say that a protocol P is stable on a network G against an adversary A if there is a constant C (which may depend on G and A) such that, starting from an empty configuration, the number of packets in the system at all times is bounded by C.*

Definition 1.2 *We say that a graph G is universally stable if every greedy protocol is stable against every adversary of rate less than 1 on G.*

Definition 1.3 *We say that a protocol P is universally stable if it is stable against every adversary of rate less than 1, on every network.*

1.2 Contribution

In this work, we identify some structural conditions (either sufficient or necessary) for FIFO stability and universal stability in the adversarial queueing model. More specifically, we prove:

- We improve significantly the best ever known FIFO stability upper bound of a simple network with three queues (Figure 2) which was first presented in [10]. In particular, we show that for any adversary, this network is stable under FIFO for any adversary with injection rate $r \leq 0.2357$.
- Then, we extend the previous argument to work in full generality, for any network N proving that such a network has an injection rate upper bound for FIFO stability that depends only on the maximum number of used disjoint paths in the execution, the maximum in-degree, and the maximum directed path length. This result improves significantly the previous best known upper bound for FIFO stability in general graphs of [8].
- Finally, we specify structural conditions for network stability. Especially, we study the stability properties of specific simple graphs that are minors of the set of the three directed graphs presented in [10]. None of the graphs in [10] is a minor of the examined graphs here. For these graphs, we demonstrate adversary constructions that lead them to instability achieving significantly low instability lower bounds in short time for each execution.

A question raised at the end of [2], refers to the existence of a rate $r_0 > 0$ such that FIFO is stable against every adversary of rate $r \leq r_0$ and every network. This question was partially answered in the positive in [8] showing that for every network, FIFO is stable against any adversary with a small injection rate, that depends basically on the number of network queues. In fact injection rate is conversely proportional to the number of network queues. In this work we further improve the stability upper bound on injection rate for general graphs showing that the injection rate basically depends on the maximum number of the used disjoint paths in the execution. Actually, it is shown that the injection rate is conversely proportional to the *square root* of the maximum number of the used disjoint paths in the execution, where in the worst case the maximum number of the used disjoint paths in the execution is equal to the number of network queues.

1.3 Related Work and Comparison

In [2], Andrews *et al.* showed that all greedy protocols are stable on rings, trees, and DAGs (directed acyclic graphs). Also, in the same paper Andrews *et al.* proved the existence of a finite set of basic undirected graphs, such that a graph G is stable for any r if and only if none of these graphs is a minor of G. Goel [10] presented three simple graphs $H_1, H_{AAFKLL01}, H_3$ as the corresponding finite set of basic directed graphs that ensures the decidability of the question: *is G universally stable?* Each of the two simple graphs presented in this work is a minor of one of Goel's graphs. Moreover, none of Goel's graphs is a minor of these two graphs. These graphs are shown to be unstable here. The adversary constructions that achieve this are illustrated. Interestingly, these adversary constructions lead the examined graphs to instability for significantly low injection rate lower bounds and in short time for each execution.

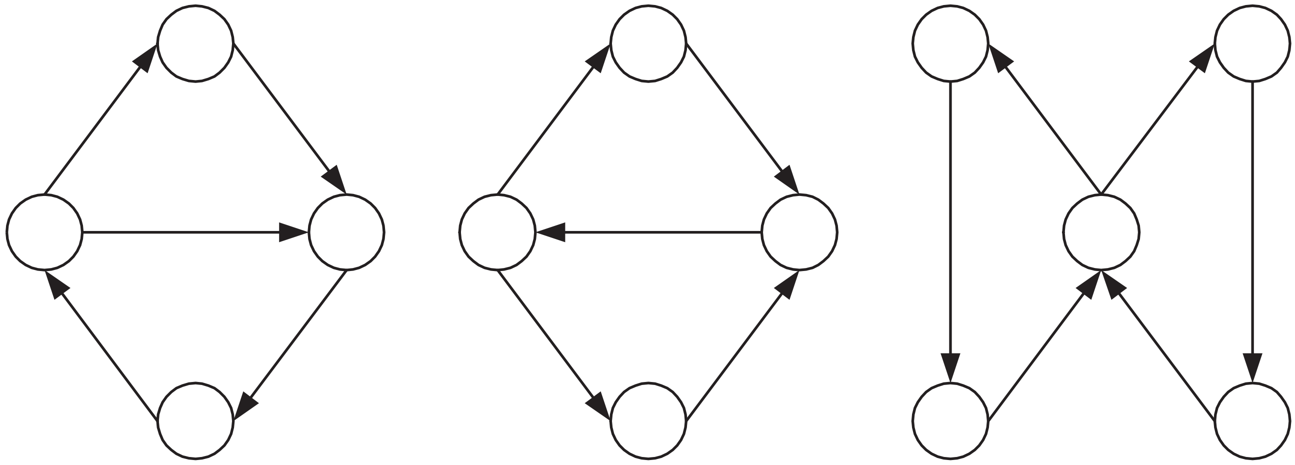

Figure 1: Goel's graph minors.

The current state-of-the-art mostly focuses on FIFO and also other greedy protocols on networks whose queues use a single service discipline. In particular, Andrews *et al.* [2] proved that the LIS, SIS, NTS, and FTG policies are universally stable, while FIFO is unstable on a particular network for injection rate $r \geq 0.85$. Later, Goel [10] presented an even simpler network of only three queues for which FIFO is unstable, as a corollary of his main task of investigating structural conditions for the decidability of the question if a given network is stable for FIFO. In [8], the FIFO instability bound is slightly improved by Diaz *et al.* by showing a network for which FIFO is unstable for injection rate $r \geq 0.8357$. Also, an open question raised by Andrews et al. in [2] is *partially* answered by Diaz *et al.* in [8] by showing for the first time upper bounds on the injection rate r for the stability of FIFO in networks with a finite number of queues. These bounds depend on the size of the network. Furthermore, Koukopoulos *et al.* [12, 13], open up the study of the stability properties of *combinations* of universally stable queueing disciplines presenting for the first time lower bounds for instability of certain combinations of universally stable queueing disciplines in the same network. Also in [12, 13], it is shown for the first

time universal stability of other combinations of universally stable service policies in the same network. Finally, in [13] the FIFO instability bound is significantly improved by Koukopoulos et al. by showing a network for which FIFO is unstable for $r \geq 0.749$.

The stability problem has been investigated extemsively, under various models of packet routing and in a number of overlapping areas; see, for example, [1, 4, 6, 7, 9, 11], and references therein, for works outside the framework of Adversarial Queueing Theory. We consider an adversary that is as weak as possible. Thus, for these purposes, we say that an adversary A has rate r if for every $t \geq 1$, every interval I of t steps, and every edge e, it injects no more than $\lceil rt \rceil$ packets during I that require e at the time of their injection.

We will present our lower bounds for systems that start from a non-empty initial configuration. This implies instability results for systems with an empty initial configuration, by the following lemma presented in Andrews et al. [2].

Lemma 1.1 (Andrews *et al.* **[2])** *Let G be a graph, P be a greedy protocol, and A an adversary of rate r, and suppose the system (G, A, P) is unstable starting with some non-empty initial configuration. Then, there exists a system $(G^{'}, A^{'}, P)$ that is unstable starting with an empty initial configuration, where $A^{'}$ is an adversary of rate r.*

1.4 Road Map

The rest of this paper is organized as follows. In Section 2, we present our sufficient conditions for FIFO stability on any network. Necessary conditions for universal stability are shown in Section 3. We conclude, in Section 4, with a discussion of our results and some open problems.

For simplicity, and in a way similar to that in [2], 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.

for any adversary with rate $r \leq 0.2357$. Our method exploits a basic property of FIFO that we prove here and which says that if $L(t_0)$ is the set of packets in the network at time t_0, then there is a time $t_1 = t_1(L, r)$ at which all these packets have been arrived to their end of path and have been removed. In subsection 2.2, the general case is presented showing an upper bound for FIFO stability in general graphs, which is independent of the initial configuration and it is a function of the maximum in-degree, the maximum directed path length and the maximum number of used disjoint paths in the execution. In section 3, we specify some structural conditions whose when they appear on graphs ensure the existence of an adversary that can lead them to instability. Especially, two simple graphs are presented which have not as minors any of the graphs of the finite set presented in [10]. For these graphs we demonstrate adversary constructions that lead them, interestingly, to significantly low instability lower bounds in short time for each execution.

2 Sufficient Structural Conditions for FIFO Stability

In this section, we present our sufficient conditions for FIFO stability on any network. Section 2.1 focuses on a specific, simple network, while the general case is treated in Section 2.2.

In [8], Diaz *et al.* gave for the first time a partial answer to an open question raised by Andrews et al. in [2] that refers to the existence of an injection rate $r_0 > 0$ such that FIFO is stable against every adversary of injection rate $r \leq r_0$ and every network, by showing for the first time upper bounds on r for the stability of FIFO in networks with a finite number of queues. Here, we further investigate this question providing an improved upper bound for FIFO stability.

In order to prove FIFO stability, we get an infinite time sequence $t_1, t_{AAFKLL01}, t_3, \ldots$. In the period between time 0 and t_1, we prove that the number of packets in the system remains bounded. Then, if the same argument is repeated at each new period starting at t_{i+1} and ending at t_{i+2}, we have proved that the number of packets in the system is a non-increasing function that guarantees the system's stability .

We next give some basic definitions that will be useful in the following analysis.

- A packet is *old* if it were injected in previous time periods than the current one. Denote M the earliest time step at which all the old packets in the system have been served. Denote s_p the number of packets in path p at time step M.

Definition 2.1 *Let's consider that j is the maximum number of used disjoint paths in the execution.*

2.1 A Specific Simple Network

Let us consider the network N_1 in Figure 2. As it has been already said, this network was considered in [10], to prove non-stability under any greedy protocol and in particular for the FIFO protocol. Also, it was used in [8] for the estimation of an upper bound for FIFO stability for the first time where it was proved that this FIFO network is stable for any adversary with injection rate $r \leq 0.1428$. In this subsection, we prove that given any initial packet configuration for N_1 , and for any adversary A with injection rate $r \leq r_0$, for a given r_0 to be computed, the number of packets in the system remains bounded. So we can conclude that FIFO is stable for N_1, under any adversary with rate $r \leq r_0$.

Theorem 2.1 *For the network N_1, given any initial configuration,* FIFO *is stable against any adversary with injection rate $0 < r \leq 0.2357$.*

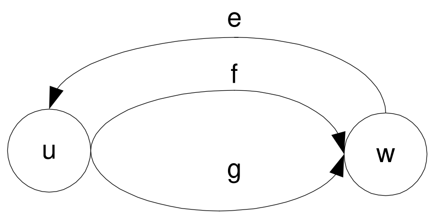

Figure 2: Network N_1.

Proof: Let's assume that time is splitted into phases. In the first part of each phase, we analyze the evolution of the system from the initial time step to the time step M in which all the initial packets have been served. The second part allows enough time to guarantee that the system configuration reproduces the initial number of packets.

Notice that the network has three queues. Therefore, the maximum number of used disjoint paths in the execution can vary from one to three. Assume that at time 0, the system has a configuration of $f(0)$ packets in queue f, $g(0)$ packets in queue g and $e(0)$ packets in queue e. Let $P(t)$ denote the number of packets in the system at time t. Then, $P(0) = f(0) + g(0) + e(0)$. Let's consider the path with the maximum length and estimate how much time in the worst case the "old" packet being last in a queue at time 0 and targeted with the largest simple path in the network needs to traverse its last edge in its path. The maximum path length is three and there are two paths with this length, f, e, g and g, e, f. We will examine f, e, g path and assume that the last packet in this path at time 0 wants to traverse all this path. Make $M_1 = f(0) \leq P(0)$. Let us run the system for M_1 steps.

As the protocol is FIFO, at time step M_1, all the $f(0)$ packets from queue f traverse edge f and a portion of them that want to traverse queue e or queue g pass to queue e where they get mixed with $(1+r)M_1$ packets at most in the worst case. This happens because the in-degree of queue e is two and consequently, in M_1 time steps, M_1 "old" packets can get into queue e from g, besides the "old" M_1 packets that get into queue e from queue f, and rM_1 newly injected packets can enter queue e. Also, $(j-1)rM_1$ more newly injected packets can get into the other $j-1$ used disjoint paths.

Next let's take $M_2 = (2+r)M_1$. Let us run the system for M_2 steps. As the protocol is FIFO, at time step $M_1 + M_2$, all the packets from $f(0)$ that have pass to the queue e at the previous period traverse queue e. A portion of them, that want to traverse queue g, passes to queue g, where they get mixed with rM_2 packets at most in the worst case. This happens because the in-degree of queue g is one and consequently in M_2 time steps only packets from queue e can get into g and rM_2 newly injected packets can enter queue g, while $(j-1)rM_2$ more newly injected packets can get into the other $j-1$ used disjoint paths.

Let's now consider $M_3 = (1+r)M_2$. Let us run the system for M_3 steps. As the protocol is FIFO, at time step $M_1 + M_2 + M_3$, all the packets from $f(0)$ that have pass to the queue g at the previous period traverse queue g. During M_3 time steps jrM_3 newly injected packets can get into the j used disjoint paths. Then at time $M = M_1 + M_2 + M_3$, the old packet that was last at queue f at time step 0 and was targeted with the largest simple path in the network will have traverse its last edge, which here is e. Therefore in this case $M = M_1 + (2+r)M_1 + (1+r)(2+r)M_1 = (5 + 4r + r^2)M_1$. However $M_1 \leq P(0)$. Thus,

$$M \leq (5 + 4r + r^2)P(0) \tag{1}$$

At time step M in the worst case, all the newly injected packets in the network at the time period $[0 \dots M]$ will remain in it. This number is less than or equal to jrM. Therefore, in the worst case at time step M the number of packets in the system, $P(M)$, is

$$P(M) \leq jrM \leq jr(5 + 4r + r^{AAFKLL01})P(0). \tag{2}$$

At time step M, s_p packets exist in each path p from the definition 6. When a path p has s_p packets, then in each time step of a time period of s_p time steps, r packets arrive into the path and one packet leaves it. Assume now $s = \max\{s_p\}$, the maximum s_p in the network at time step M. Then, the change of the configuration in path p (the change of the number of packets remain in path p), Δ_p, in $M + s$ time step will be $\Delta_p = \Sigma_0^{\max\{s_p\}}(r-1) = (r-1)\max\{s_p\} \leq (r-1)s_p$. Therefore, the total change of the system configuration in the network will be $\Sigma_p \Delta_p \leq \Sigma_p (r-1)s_p = (r-1)\Sigma_p s_p$. But, $P(M) = \Sigma_p s_p$. Thus,

$$\Sigma_p \Delta_p \leq (r-1)P(M). \tag{3}$$

We have shown that for a time period with $s = max(s_p)$ time steps the change of the system configuration is at most $(r-1)P(M)$. Consider now time intervals with duration: $s, rs, r^2 s, \ldots, r^k s$, where k is such that $r^k s \geq 1$ and $r^{k+1} s < 1$. The same argument as in the case of s time steps can be used for $r^i s$ time steps. For each of these time intervals the change of the system configuration will be at most $r^i(r-1)P(M)$. Let t_1 be the time at which $r^k s$ finishes. The packets in network N_1 at time t_1 are all new, therefore the number of packets in the network at time t_1 is at most:

$$P(t_1) \leq P(M) + (r-1)P(M) + r(r-1)P(M) + \ldots + r^{k-1}(r-1)P(M) = r^k P(M). \tag{4}$$

For the stability condition, we need $P(t_1)$ to be less than or equal to $P(0)$, which implies that we must choose an r such that, $r^k P(M) \leq P(0)$. But, $P(M) \leq jrM \leq jr(5+4r+r^2)P(0)$. Therefore substituting $P(M)$ in the above inequality we take $r^k jr(5+4r+r^2)P(0) \leq P(0)$. Therefore, $r^{k+1} \leq \frac{1}{j(5+4r+r^2)}$. Thus, $r \leq \frac{1}{\sqrt[k+1]{j(5+4r+r^2)}}$.

For $k = 1$ the above inequality takes its smallest value. Thus, $r \leq \frac{1}{\sqrt{j(5+4r+r^2)}}$. This inequality is sufficient to guarantee stability as $\frac{1}{\sqrt{j(5+4r+r^2)}} \leq \frac{1}{\sqrt[k+1]{j(5+4r+r^2)}}$. Therefore to guarantee the stability of the system, the injection rate r should be less than or equal to $\frac{1}{\sqrt{j(5+4r+r^2)}}$, which holds for $r \leq 0.2357$. Taking $r_0 \leq 0.2357$, we have shown that $P(t_1) \leq P(0)$ for any value of $r \leq r_0$. Repeat the argument, getting an infinite time sequence $t_1, t_2, t_3, \ldots$. In the period between time 0 and t_1, the number of packets in the system is bounded. Thus, we have proved that at the beginning of each new period starting at t_{i+1}, the number of packets in the system, is a non-increasing function, which guarantees stability. ■

2.2 General Networks

The previous argument, can be extended to work in full generality, for any network N with maximum number of used disjoint paths in the execution j, maximum in-degree α, and maximum directed path length β.

Theorem 2.2 *For any network N, given any initial configuration, there exist an $0 < r_N < 1$, such that* FIFO *is stable with respect any adversary with injection rate $r \leq r_N$, where r_N is independent of the initial configuration and it is a function of the maximum in-degree, the maximum directed path length and the maximum number of used disjoint paths in the network execution.*

Proof: Assume that N has maximum number of used disjoint paths in the execution j, maximum in-degree α, and maximum directed path length β and that the injection rate of A is r. Notice that if $\alpha = 1$ then we have a tree or a ring, for which it is known that such a network is universally stable [2], so we assume $\alpha > 1$. Let us denote the queues as $Q_1, Q_2, \ldots, Q_k$ and their loads at time $t \geq 0$ as $q_1(t), q_2(t), \ldots, q_k(t)$. Let $P(0) = \Sigma q_i(0)$ be the initial load. We will construct an infinite sequence of time periods, t_i, at which $P(t_i) \leq P(0)$ thus keeping the network stable. The fact that we are using a FIFO protocol implies that after a certain time all the old packets will leave the system. We will compute a bound to this time.

Let's now, consider the worst case of an old packet being last in a queue Q_j at time 0 and targeted with the largest simple path in the net. Rename the queues in this simple path as $Q_j \equiv Q_{j_0}, \ldots = Q_{j_{\beta-1}}$. Note that at time $M_1 = q_{j_0}$ all packets of this queue will have been served. Thus these packets have passed to the next queues in the path. Moreover, they can be delayed by at most rM_1 new injections. Furthermore, the size of any Q_{j_i} is bounded above by $(\alpha + r)M_1$. We repeat the same procedure, each time considering the last queue in the path that still contains old packets. After $\beta - 2$ additional steps $(M_2, M_3, \ldots, M_{\beta-1})$ all the old packets would disappear or being in Q_{j_β}.

Define $P(t) = \max_{i=0}^{k} \{q_i(t)\}$. Working in the previous way, an absolute bound for the delay of the last old packet in Q_j is $M = M_1 + \ldots + M_{\beta-1}$, where for every $0 < i < \beta$, we have $M_1 \leq q(\Sigma_{j<i} M_j)$, with $M_0 = 0$. Moreover, during a period of $q(t)$ steps starting at time t, we have $P(t + P(t)) \leq (\alpha + r)P(t)$. Solving the recurrence, we have that the total time is

$$M \leq \Sigma_{i=0}^{\beta-1} (\alpha + r)^i P(0) \tag{5}$$

At time step M all the old packets have been absorbed and only the newly injected packets in the time period $[0 \ldots M]$ will remain in the system. Because j is the maximum number of used disjoint paths in the execution then during this period at most jrM packets will be injected in the network. Therefore the total number of packets in the network at time step M is at most $P(M) \leq jrM$.

At time step M, s_p packets exist in each path p from the definition 6. When a path p has s_p packets, then in each time step of a time period of s_p time steps, r packets arrive into the path and one packet leaves it. Assume now $s = \max\{s_p\}$, the maximum s_p in the network at time step M. The change of the configuration (the change of the number of packets) in path p, Δ_p, in $M + s$ time step will be $\Delta_p = \Sigma_0^{\max\{s_p\}} (r-1) = (r-1)\max\{s_p\} \leq (r-1)s_p$. Therefore, the total change of the system configuration in the network will be $\Sigma_p \Delta_p \leq \Sigma_p (r-1)s_p = (r-1)\Sigma_p s_p$. However, $P(M) = \Sigma_p s_p$. Thus,

$$\Sigma_p \Delta_p \leq (r-1)P(M). \tag{6}$$

We have shown that for a time period with $s = max(s_p)$ time steps the change of the system configuration is at most $(r-1)P(M)$. Consider now, the time intervals with duration: $s, rs, r^2 s, \ldots, r^k s$, where k is such that $r^k s \geq 1$ and $r^{k+1} s < 1$. The same argument as in the case of s time steps can be used for $r^i s$ time steps. For each of these time intervals the change of the system configuration will be at most $r^i(r-1)P(M)$. Let t_1 be the time at which $r^k s$ finishes. The packets in network N_1 at time t_1 are all new, therefore the number of packets in the network at time t_1 is at most:

$$P(t_1) \leq P(M) + (r-1)P(M) + r(r-1)P(M) + \ldots + r^{k-1}(r-1)P(M) = r^k P(M). \tag{7}$$

For the stability condition, we need $P(t_1)$ to be less than $P(0)$, which implies that we must choose an injection rate r such that, $r^k P(M) \leq P(0)$. But, $P(M) \leq jrM \leq jr\Sigma_0^{\beta-1}(\alpha + r)^i P(0)$. Therefore substituting $P(M)$ in the above inequality we take $r^k jr \Sigma_0^{\beta-1} (\alpha + r)^i P(0) \leq P(0)$. Therefore, $r^{k+1} j \Sigma_0^{\beta-1} (\alpha + r)^i \leq 1$. Thus,

$$r \leq \frac{1}{\sqrt[k+1]{j\Sigma_0^{\beta-1}(\alpha + r)^i}}. \tag{8}$$

For $k = 1$ the above inequality takes its smallest value. Thus, $r \leq \frac{1}{\sqrt{j\Sigma_0^{\beta-1}(\alpha+r)^i}}$. This inequality is sufficient to guarantee stability as $\frac{1}{\sqrt{j\Sigma_0^{\beta-1}(\alpha+r)^i}} \leq \frac{1}{\sqrt[k+1]{j\Sigma_0^{\beta-1}(\alpha+r)^i}}$. The inequality $r \leq \frac{1}{\sqrt{j\Sigma_0^{\beta-1}(\alpha+r)^i}}$, is equivalent to finding in the real interval $(0, 1)$, the root r_N of

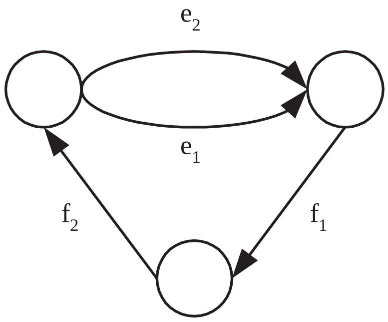

Figure 3: Network G_1.

$$-r^2j(\alpha+r)^\beta+r^2j+\alpha+r-1.$$

By the Bolzano's theorem, this polynomial has a root between 0 and 1, which is r_N. The root of this polynomial is bigger than the root giving in [8]. This holds because the injection rate upper bound for FIFO stability is conversely proportional to the number of network queues in [8], while here we show that the injection rate upper bound for FIFO stability is conversely proportional to the square root of the maximum number of used disjoint paths in the execution, where in the worst case the maximum number of used disjoint paths in the execution is equal to the number of network queues. ■

3 Necessary Structural Conditions for Universal Stability

In this section two simple graphs are presented that are minors of the finite set of graphs presented in [10]. None of the graphs of the finite set given in [10] is a minor of the graphs examined here. Furthermore, adversary constructions are demonstrated here that lead these two simple graphs, interestingly, to significantly low instability lower bounds in short time (small number of time rounds for each execution). Therefore, these graphs if they are minors or subgraphs of general graphs can be used as an indication that there is an adversary that can lead these graphs to instability. For the first adversary construction it is assumed that the used protocol in the network is NTG, while for the second adversary construction the used protocol in the network is $NTG-U-LIS$ that is a variation of NTG protocol that uses LIS for ties resolution.

We split the time into phases. In each phase we study the evolution of the system configuration by considering corresponding time rounds. For each phase, to show instability we inductively show that the number of packets in the system increases. This inductive argument can be applied repeatedly, thus showing instability. In addition, our constructions use symmetric networks of two parts. In addition, *in order for the induction to work it is assumed that there is a large enough number s of packets in the initial system configuration.* This implies instability for systems with an empty initial configuration, by lemma 1.1 above.

Theorem 3.1 *Let $r\geq 0.794$. There is an adversary A of rate r, such that the system (G_1,A,NTG) is unstable, starting from a non-empty initial configuration.*

Proof: Let's consider the network G_1 in Figure 3 that uses NTG policy for contention resolution.

Inductive Hypothesis: At the beginning of phase j, there are s packets queued in the queues e_1,e_2 requiring to traverse edge f_1.

Induction Step: At the beginning of phase $j+1$ there will be more than s packets queued in the queues e_1,e_2 requiring to traverse edge f_1.

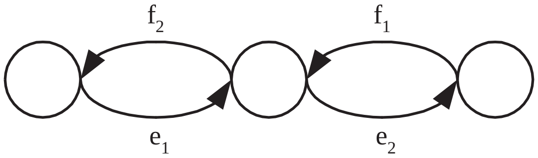

Figure 4: Network G_2.

We will construct an adversary A such that the induction step will hold. Proving that the induction step holds, we ensure that the inductive hypothesis will hold at the beginning of phase $j+1$ with an increased value of s, $s^{'} > s$. From the inductive hypothesis, initially, there are s packets (called $S-flow$) in the queues e_1, e_2 requiring to traverse edge f_1.In order to prove that the induction step works, we should consider that there is a large enough number of packets s in the initial system configuration. During phase j the adversary plays three rounds of injections. The sequence of injections is as follows:

Round 1: For s steps, the adversary injects in queue f_1 a set Z_1 of $|Z_1| = rs$ packets wanting to traverse edges f_1, f_2, e_1. These packets are blocked by the $S-flow$ packets in queue f_1 because f_1 uses NTG as queueing discipline and $S-flow$ packets coming from queues e_1, e_2 are nearest to their destination (queue f_1) than Z_1-flow packets (queue e_1). Therefore $S-flow$ packets have priority over Z_1-flow packets.

Round 2: For the next rs steps, the adversary injects a set Z_2 of $|Z_2| = r^2 s$ packets in queue f_2 requiring to traverse edges f_2, e_2, f_1 and a set Z_3 of $|Z_3| = r^2 s$ packets in queue e_1 requiring to traverse edge e_1. The Z_1-flow packets block Z_2-flow packets in queue f_2 because Z_1-flow packets have nearest to go (queue e_1) than Z_2-flow packets (queue f_1). Therefore, all Z_1-flow packets arrive at queue e_1 where they get mixed with Z_3-flow packets. The total number of packets arriving in queue e_1 during this round is $|Z_1| + |Z_3| = rs + r^2 s$ packets. However the duration of this round is rs steps. Therefore, rs packets will traverse edge e_1. Thus, at the end of this round, there will be a set X of $|X| = r^2 s$ remaining packets in queue e_1 wanting to traverse edge e_1. Also, at the end of this round there are $|Z_2| = r_2 s$ packets in queue f_2 wanting to traverse edges f_2, e_2, f_1.

Round 3: For the next $r^2 s$ steps, the adversary injects a set Z_4 of $|Z_4| = r^3 s$ packets in queue e_2 requiring to traverse edge e_2 and a set Z_5 of $|Z_5| = r^3 s$ packets in queue e_1 requiring to traverse edges e_1, f_1. The packets of the set X block Z_5-flow packets in queue e_1 because X packets have nearest to go (queue e_1) than Z_5-flow packets (queue f_1). Thus at the end of this round, there are $|Z_5| = r^3 s$ packets in queue e_2 wanting to traverse edges e_1, f_1. Furthermore, the Z_4-flow packets have priority over Z_2-flow packets in queue e_2 because Z_4-flow packets have nearest to go (queue e_2) than Z_2-flow packets (queue f_1). But, the number of Z_4-flow packets is $|Z_4| = r^3 s$ while this round has $r^2 s$ time steps. Therefore, along with the Z_4-flow packets and $r^2 s - r^3 s$ packets from Z_2-flow will traverse edge e_2. Thus at the end of this round, the remaining portion of Z_2-flow packets in queue e_2 that want to traverse edges e_2, f_1 is $|Y| = r^3 s$ packets.

At the end of this round the number of packets in queues e_1, e_2 requiring to traverse edge f_1 is $s^{'} = |Y| + |Z_5| = r^3 s + r^3 s = 2r^3 s$. In order to have instability, we must have $s^{'} > s$. Therefore, we should have $2r^3 s > s$, i.e. $r \geq 0.794$. This argument can be repeated for an infinite number of phases ensuring that the number of packets queued in the queues e_1, e_2 requiring to traverse edge f_1 at the end of a phase will be bigger than the number of packets queued in the queues e_1, e_2 requiring to traverse edge f_1 at the beginning of the phase.

■ ■

Theorem 3.2 *Let $r \geq 0.794$. There is an adversary A of rate r, such that the system (G_2,A,$NTG-U-LIS$) is unstable, starting from a non-empty initial configuration.*

Proof: Let's consider the network G_2 in Figure 4 that uses $NTG-U-LIS$ policy for contention resolution.

Inductive Hypothesis: At the beginning of phase j, there are s packets queued in the queues e_1, e_2 requiring to traverse edges e_1, f_2 and e_2, f_1, f_2 correspondingly.

Induction Step: At the beginning of phase $j+1$ there will be more than s packets queued in the queues e_1, e_2 requiring to traverse edges e_1, f_2 and e_2, f_1, f_2 correspondingly.

We will construct an adversary A such that the induction step will hold. Proving that the induction step holds, we ensure that the inductive hypothesis will hold at the beginning of phase $j+1$ with an increased value of s, $s^{'} > s$. From the inductive hypothesis, initially, there are s packets (called $S-flow$) in the queues e_1, e_2 requiring to traverse edges e_1, f_2 and e_2, f_1, f_2 correspondingly. In order to prove that the induction step works, we should consider that there is a large enough number of packets s in the initial system configuration. During phase j the adversary plays three rounds of injections. The sequence of injections is as follows:

Round 1: For s steps, the adversary injects in queue f_2 a set Z_1 of $|Z_1| = rs$ packets wanting to traverse edges f_2, e_1, e_2. These packets are blocked by the $S-flow$ packets in queue f_2 because f_2 uses $NTG-U-LIS$ as queueing discipline and $S-flow$ packets coming from queues e_1, e_2 are nearest to their destination (queue f_2) than Z_1-flow packets (queue e_2). Therefore $S-flow$ packets have priority over Z_1-flow packets.

Round 2: For the next rs steps, the adversary injects a set Z_2 of $|Z_2| = r^2 s$ packets in queue e_1 requiring to traverse edges e_1, f_2 and a set Z_3 of $|Z_3| = r^2 s$ packets in queue e_2 requiring to traverse edge e_2. The Z_1-flow packets block Z_2-flow packets in queue e_1, although they have the same number of edges to traverse in order to reach their destination because the system policy is $NTG-U-LIS$ that uses LIS policy to solve such kind of contentions and Z_1-flow packets are longer time in the system than Z_2-flow packets. Therefore, all Z_1-flow packets arrive at queue e_2 where they get mixed with Z_3-flow packets. The total number of packets arriving in queue e_2 during this round is $|Z_1| + |Z_3| = rs + r^2 s$ packets. However the duration of this round is rs steps. Therefore, rs packets traverse edge e_2 during this round, after which they are absorbed. Thus, at the end of this round, there will be a set X of $|X| = r^2 s$ remaining packets in queue e_2 wanting to traverse edge e_2. Also, at the end of this round there are $|Z_2| = r_2 s$ packets in queue e_1 wanting to traverse edges e_1, f_2.

Round 3: For the next $r^2 s$ steps, the adversary injects a set Z_4 of $|Z_4| = r^3 s$ packets in queue e_1 requiring to traverse edge e_1 and a set Z_5 of $|Z_5| = r^3 s$ packets in queue e_2 requiring to traverse edges e_2, f_1, f_2. The packets of the set X block Z_5-flow packets in queue e_2 because X packets have nearest to go (queue e_2) than Z_5-flow packets (queue f_2). Thus at the end of this round, there are $|Z_5| = r^3 s$ packets in queue e_2 wanting to traverse edges e_2, f_1, f_2. Furthermore, the Z_4-flow packets have priority over Z_2-flow packets in queue e_1 because Z_4-flow packets have nearest to go (queue e_1) than Z_2-flow packets (queue f_2). But, the number of Z_4-flow packets is $|Z_4| = r^3 s$ while this round has $r^2 s$ time steps. Therefore, along with the Z_4-flow packets and $r^2 s - r^3 s$ packets from Z_2-flow will traverse edge e_1. Thus at the end of this round, the remaining portion of Z_2-flow packets in queue e_1 that want to traverse edges e_1, f_2 is $|Y| = r^3 s$ packets.

At the end of this round the number of packets in queues e_1, e_2 requiring to traverse edges e_1, f_2 and e_2, f_1, f_2 correspondingly is $s^{'} = |Y| + |Z_5| = r^3 s + r^3 s = 2r^3 s$. In order to have instability, we must have $s^{'} > s$. Therefore, we should have $2r^3 s > s$, i.e. $r \geq 0.794$. This argument can be repeated for an infinite number of phases ensuring that the number of packets that are queued in the queues e_1, e_2 requiring to traverse edges e_1, f_2 and e_2, f_1, f_2 correspondingly at the end of a phase will be bigger than the number of packets queued in the queues e_1, e_2 requiring to traverse edges e_1, f_2 and e_2, f_1, f_2 correspondingly at the beginning of the phase. ∎

The previous two theorems are sufficient to guarantee that the following corollary holds.

Corollary 3.3 *If a graph G has as minors or subgraphs the graphs G_1 or G_2, then there is an adversary A of rate r that leads the graph G to instability for $r \geq 0.794$.*

4 Discussion and Directions for Further Research

In this paper, we investigate the problem of FIFO stability showing upper bounds for stability in every graph and the general problem of network stability trying to specify structural conditions on networks, which can be exploited by an adversary to lead such networks to instability. The upper bound for FIFO stability answers partially an open question raised in [2], which was first investigated in [8] improving significantly the previous best known injection rate upper bound for FIFO stability. The second part of the paper deals with structural conditions for network stability presenting simple graphs that can lead to instability general graphs that have the presented here simple graphs as minors or subgraphs.

Many interesting questions remain open in the context of stability. An open issue is the specification of a systematic adversary construction that can lead FIFO networks to instability as until now all the related works present ad-hoc adversary constructions. Another open issue is the examination of various classes of networks as concerns the issue of universal stability for any adversary. Finally, a challenging open question is whether there is any threshold that marks the transition from stability to instability in FIFO networks.

Acknowledgements:

We wish to thank Prof. Josep Diaz for helpful discussions (during the kickoff meeting of FLAGS) on structural conditions of networks that are responsible for instability.

References

[1] 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.

[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] 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.

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

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

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

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

[8] 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.

[9] D. Gamarnik, "Stability of Adversarial Queues via Fluid Models," *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science,* pp. 60-76, October 1998.

[10] A. Goel, "Stability of Networks and Protocols in the Adversarial Queueing Model for Packet Routing," *Networks,* Vol. 37, No. 4, pp. 219-224, 2001.

[11] F. P. Kelly, *Reversability and Stochastic Networks,* Wiley, New York, 1979.

[12] D. Koukopoulos, S. Nikoletseas and P. Spirakis, "The Range of Stability for Heterogeneous and FIFO Queueing Networks," *Electronic Colloquium on Computational Complexity (ECCC),* Reports 2001, Report TR01-099, ISSN 1433-8092, December 2001.

[13] D. Koukopoulos, S. Nikoletseas and P. Spirakis, "Stability Issues in Heterogeneous and FIFO Networks Under the Adversarial Queueing Model," Invited Keynote Address, *High Performance Computing 2001 (HiPC2001),* Hyderabad, India, December 2001.

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