The Impact of Network Structure on the Stability of Greedy Protocols*

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

[2]**Contact Author.** Department of Computer Engineering and Informatics, University of Patras, Rion, 265 00 Patras, Greece, & Research and Academic Computer Technology Institute, P. O. Box 1122, 261 10 Patras, Greece. E-mail: Dimitrios.Koukopoulos@cti.gr

[3]Department of Computer Science, University of Cyprus, 1678 Nicosia, Cyprus. E-mail: mavronic@ucy.ac.cy

[4]Department of Computer Engineering and Informatics, University of Patras, Rion, 265 00 Patras, Greece, & Research and Academic Computer Technology Institute, P. O. Box 1122, 261 10 Patras, Greece. E-mail: {nikole,spirakis}@cti.gr

*A preliminary version of this work has been appeared in the *Proceedings of the 5th Italian Conference on Algorithms and Complexity,* R. Petreschi, G. Persiano and R. Silvestri eds., pp. 251–263, Vol. 2653, Lecture Notes in Computer Science, Springer-Verlag, Rome, Italy, May 2003. This work has been partially supported by the IST Program of the European Union under contract numbers IST-1999-14186 (ALCOM-FT) and IST-2001-33116 (FLAGS), by funds from the Joint Program of Scientific and Technological Collaboration between Greece and Cyprus (research project "Efficiency and Performance of Distributed Systems: Capabilities and Limitations"), and by funds for the promotion of research at University of Cyprus.

Abstract

A *packet-switching* network is *stable* if the number of packets in the network remains bounded at all times. A very natural question that arises in the context of stability properties of such networks is how network structure precisely affects these properties.

In this work, we embark on a systematic study of this question in the context of *Adversarial Queueing Theory,* which assumes that packets are adversarially injected into the network. We consider *size, diameter,* maximum *vertex degree,* minimum number of *disjoint paths* that cover all edges of the network, and *network subgraphs* as crucial structural parameters of the network, and we present a comprehensive collection of structural results, in the form of stability and instability bounds on injection rate of the adversary for various greedy protocols:

- Increasing the size of a network may result in dropping its instability bound. This is shown through a novel, yet simple and natural, combinatorial construction of a size-parameterized network on which certain compositions of greedy protocols are running. The convergence of the drop to 0.5 is found to be fast with and proportional to the increase in size.
- Maintaining the size of a network small may already suffice to drop its instability bound to a substantially low value. This is shown through a construction of a FIFO network with size 22, which becomes unstable at rate 0.704. This represents the current state-of-the-art trade-off between network size and instability bound.
- The diameter, maximum vertex degree and minimum number of edge-disjoint paths that cover a network may be used as control parameters for the stability bound of the network. This is shown through an improved analysis of the stability bound of any arbitrary FIFO network, which takes these parameters into account.
- How much can network subgraphs that are forbidden for stability affect the instability bound? Through improved combinatorial constructions of networks and executions, we improve the state-of-the-art instability bound induced by certain known forbidden subgraphs on networks running a certain greedy protocol.

Our results shed more light and contribute significantly to a finer understanding of the impact of structural parameters on stability and instability properties of networks.

1 Introduction

1.1 Motivation and Framework

Objectives. A lot of research has been done in the field of packet-switched communication networks for the specification of their behavior. In such networks, packets arrive dynamically at the nodes and they are routed in discrete time steps across the edges. In this work, we embark on a study of the impact structural network properties have on the correctness and performance properties of networks. We study here *greedy* protocols as our test-bed. In some cases, we consider networks in which different switches can use different greedy protocols. This is motivated by the *heterogeneity* of modern large-scale networks such as the Internet.

Framework of Adversarial Queueing Theory. We focus on a basic adversarial model for packet arrival and path determination that has been recently introduced in a pioneering work by Borodin *et al.* [4]. It was developed as a robust counterpart to classical Queueing theory [6] that replaces stochastic by worst case assumptions. The underlying goal is to determine whether it is feasible to prove stability results even when packets are injected by an *adversary.* At each time step, the adversary may inject a set of packets into some nodes. For each packet, the adversary specifies a simple path that the packet must traverse; when the packet arrives to its destination, it is absorbed by the system. When more than one packets wish to cross a queue at a given time step, a *contention-resolution* protocol is employed to resolve the conflict. A crucial parameter of the adversary is its *injection rate* r, where $0 < r < 1$. Among the packets that the adversary injects in any time interval I, at most $\lceil r|I| \rceil$ can have paths that require any particular edge. We say that a packet p *requires* an edge e at time t if the edge e lies on the path from its position to its destination at time t.

Stability. Stability requires that the number of packets in the system remains bounded at all times. We say that a protocol P is *stable* [4] on a network $\mathcal{G}$ against an adversary $\mathcal{A}$ of rate r if there is a constant C (which may depend on $\mathcal{G}$ and $\mathcal{A}$) such that the number of packets in the system is bounded at all times by C. We say that *a protocol* P *is universally stable* [4] if it is stable against every adversary of rate less than 1 and on every network. We also say that *a network* $\mathcal{G}$ *is universally stable* [4] if every greedy protocol is stable against every adversary of rate less than 1 on $\mathcal{G}$. We say *forbidden subgraphs* for network stability when the packets follow non-simple paths (paths do not contain repeated edges, but they contain repeated vertices) [2, 8] any graph obtained by replacing any edge of the graphs $\mathcal{U}_1$ and $\mathcal{U}_2$ (see Figures 1 and 2) by disjoint directed paths.

Greedy Protocols. We consider six *greedy* contention-resolution protocols– ones that always advance a packet across a queue (but one packet at each discrete time step) whenever there

Protocol name	Which packet it advances:	**US**
Shortest-In-System (SIS)	The most recently injected packet	√ [1, Theorem 2.3]
Longest-In-System (LIS)	The least recently injected packet	√ [1, Theorem 2.5]
Furthest-To-Go (FTG)	The furthest packet from its destination	√ [1, Theorem 2.7]
Nearest-To-Source (NTS)	The nearest packet to its origin	√ [1, Theorem 2.8]
First-In-First-Out (FIFO)	The earliest arrived packet at the queue	**X** [1, Theorem 2.10]
Nearest-To-Go-Using-LIS (NTG-U-LIS)	The nearest packet to its destination or the same as LIS for tie-breaking	**X** [2, Lemma 7]

Table 1: Greedy protocols considered in this paper. (**US** stands for universally stable)

resides at least one packet in the queue (see Table 1).

Network Structure. Important parameters of network structure are:

- size– the number of queues in the network,
- diameter– the maximum directed path length in the network,
- maximum vertex degree– the maximum number of ingoing edges in a vertex in the network,
- minimum number of edge-disjoint paths that cover all edges of the network, and
- forbidden subgraphs for stability.

1.2 Contribution

Our work interestingly shows how network structure precisely affects stability. In particular, we present a comprehensive collection of structural results in the form of stability and instability bounds on injection rate of the adversary.

- We prove that increasing the network size can drop the lower bound on injection rate that guarantees instability for heterogeneous networks. This is shown through a novel, yet simple and natural, construction of a *size-parameterized* network on which compositions of LIS protocol with any of SIS, NTS and FTG protocol are running. In particular, we apply our construction in instances of a parameterized network family and we prove that when network size tends to infinity then the instability bound for the compositions of LIS protocol with any of SIS, NTS and FTG protocol converges to 0.5. The convergence of the drop to 0.5 is found to be fast with and proportional to the increase in size.

- We show how specific network graph parameters such as the maximum directed network path length, the maximum vertex degree and the minimum number of edge-disjoint paths that cover a network can be used as a control mechanism for proving a stability bound on any arbitrary network that uses FIFO as contention-resolution protocol. Our analysis obtains an upper bound on FIFO stability based on a fundamental FIFO property, namely that in any FIFO network, packets *exit the network* after some bounded time (by their size and the network structure). This result improves the previous known upper bound for FIFO stability of [7] for all networks. Furthermore, for several networks our stability bound is better than the one estimated in [12] such as the network $\mathcal{U}_1$ in Figure 1.
- We prove that applying a specific adversarial construction to a small size network suffices to drop its instability bound to a substantially small value. This is shown for a network with only 22 queues on which queues FIFO is running. This network is proven to be unstable for any $r \geq 0.704$. The same instability bound can be obtained applying the technique of [12] on a size-parameterized network with at least 361 queues. The technique we use exploits the existence of multiple parallel paths between a common origin and destination in the network topology as a delaying mechanism.
- We prove that certain forbidden subgraphs for universal stability can drop to lower values the instability bound of the networks where they are subgraphs. In particular, we study two simple graphs $\mathcal{U}_2$ and $\mathcal{U}_3$ (see Figures 2 and 7) that have been shown in [2, 8] to be forbidden subgraphs for universal stability in the model of *non-simple paths* (paths do not contain repeated edges, but they contain repeated vertices). Note that $\mathcal{U}_3$ is an extension of $\mathcal{U}_1$ (Figure 1) for $n = 0$, $m = 1$ and $d = 2$. For these graphs we show instability for lower rates than those in [2] via a different construction. Note that we assume that NTG-U-LIS protocol runs on the network queues as in [2]. The adversarial construction we apply is based on the exploitation of injected packet sets in specific time periods as delaying mechanisms for packet sets injected in following time periods.

1.3 Related Work

Adversarial Queueing Model. Adversarial Queueing Theory and corresponding stability and instability issues, received a lot of interest and attention (see, e.g., [1, 2, 7, 9, 11, 13]). The universal stability of SIS, LIS, NTS and FTG protocols was established by Andrews *et al.* [1].

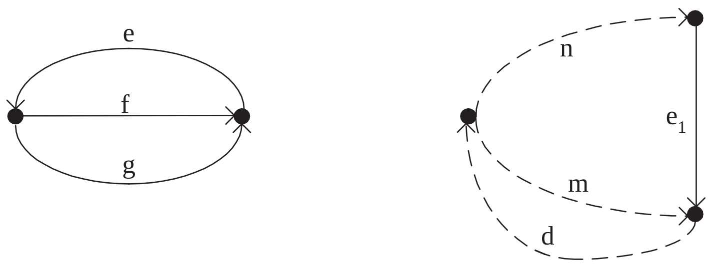

Figure 1: Network $\mathcal{U}_1$ and its extension $\Gamma(\mathcal{U}_1)$ [2, Lemma 7]

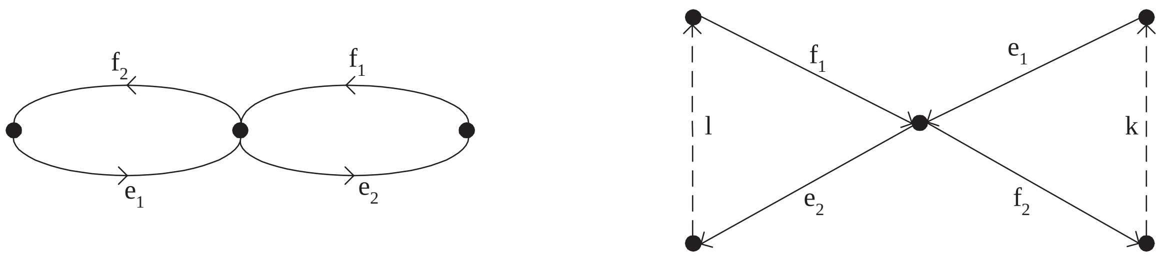

Figure 2: Network $\mathcal{U}_2$ and its extension $\Gamma(\mathcal{U}_2)$ [2, Lemma 7]

Stability in Heterogeneous Networks. The subfield of study of the stability properties of compositions of universally stable protocols has been opened recently by *Koukopoulos et al.* [9, 11] where lower bounds of 0.683 and 0.519 on the injection rate that guarantee instability for the composition pairs LIS-SIS, LIS-NTS and LIS-FTG were respectively presented.

Stability of FIFO *Networks.* The subfield of proving stability bounds for greedy protocols on every network was first initiated by Diaz *et al.* [7] showing an upper bound on injection rate that guarantees the stability of FIFO in networks with a finite number of queues which is based on network parameters. In an alternative work, Lotker *et al.* [12] proved that any greedy protocol can be stable in any network if the injection rate of the adversary is upper bounded by $1/(d+1)$, where d is the maximum path length that can be followed by any packet. Also, they proved that for a specific class of greedy protocols, time-priority protocols, the stability bound becomes $1/d$.

Instability of FIFO *Networks.* The instability of FIFO for *small-size* networks (in the model of adversarial queueing theory) was first established by Andrews *et al.* [1, Theorem 2.10] for injection rate $r \geq 0.85$. Lower bounds of 0.8357 and 0.749 on FIFO instability were presented by Diaz *et al.* [7, Theorem 3] and Koukopoulos *et al.* [9, Theorem 5.1]. Recently, it has been proved by Koukopoulos *et al.* [10] a lower bound of 0.41 for FIFO instability on a network with eight nodes. But, this bound holds for a model of dynamic capacities [5] that is an extension of the classical model of adversarial queueing theory. In this model of dynamic capacities [5], the capacities of network links can be changed dynamically by the adversary to any integer value in the interval $[1, C]$ with $C > 1$, while in the classical model of adversarial queueing theory all network link capacities are unit all the time.

An alternative approach for studying FIFO instability in the context of adversarial queueing theory is based on parameterized constructions for networks with *unbounded size*. Using this approach, Lotker *et al.* [12] proved an instability bound of $\frac{1}{2} + \epsilon$ for FIFO; the network size is a function of r that goes to infinity very fast as r goes down to 0.5. Recently, this result was improved by Bhattacharjee and Goel [3] showing that FIFO can become unstable for arbitrarily small injection rates on parameterized network constructions. Furthermore, Koukopoulos *et al.* [10] achieved partial progress towards the same goal showing that FIFO can become unstable for arbitrarily small injection rates on parameterized network constructions under the model of dynamic capacities that has been initiated in [5]. However, it has been recently found that there is a gap in the proof of the claim of [10] that this result holds, also, for the classical model of adversarial queueing theory.

Instability of Forbidden Subgraphs. In [2, Lemma 7], a characterization for directed network graphs (digraphs) universal stability is given when the packets follow non-simple paths (paths do not contain repeated edges). According to this characterization a digraph is universally stable if and only if it does not contain as subgraph any of the extensions of $\mathcal{U}_1$ ($\Gamma(\mathcal{U}_1)$) or $\mathcal{U}_2$ ($\Gamma(\mathcal{U}_2)$) where the parameters n, m, d, l, k represent numbers of consecutive edges with $l, k, n \geq 0$ and $m, d > 0$ (see Figures 1 and 2). These graphs have been shown [2, Lemma 7] to have instability bounds of 0.84089 for NTG-U-LIS protocol.

1.4 Road Map

The rest of this paper is organized as follows. Section 2 presents model definitions. Section 3 demonstrates our lower bounds on injection rate that guarantee instability for compositions of protocols. Section 4 shows upper bounds on injection rate that guarantee stability for FIFO. Section 5 presents a lower bound on injection rate that guarantees instability for a FIFO network. Section 6 shows our lower bounds on injection rate that guarantee instability for forbidden subgraphs. We conclude, in Section 7, with a discussion of our results and some open problems.

2 Definitions and Preliminaries

The adversarial queueing model considers a communication network that is modelled by a directed graph $\mathcal{G} = (V, E)$, where $|V| = n$, and $|E| = m(\mathcal{G})$. Each node $u \in V$ represents a communication switch, and each edge $e \in E$ represents a link between two switches. In each node, there is a buffer (queue) associated with each outgoing link. Buffers store packets that are injected into the network with a route, which is a simple directed path in $\mathcal{G}$. When a packet is injected, it is placed in the buffer of the first link on its route.

Important parameters of the structure of a network is the size $m(\mathcal{G})$ (the number of network queues), and the minimum number of edge-disjoint paths $j(\mathcal{G})$ that cover the graph $\mathcal{G}$. It holds that $\frac{1}{j(\mathcal{G})} \geq \frac{1}{m(\mathcal{G})}$. Other important parameters are the maximum vertex degree $\alpha(\mathcal{G})$ (the maximum number of ingoing edges in a vertex in the network), the diameter $d(\mathcal{G})$ (maximum directed path length in the network) and the existence of forbidden subgraphs for universal stability.

The definition of a *bounded adversary* $\mathcal{A}$ of rate (r, b) (where $b \geq 1$ is a natural number and $0 < r < 1$) in the adversarial queueing theory model [4] requires that for any edge e and any time interval I, the adversary injects no more than $r|I| + b$ packets during $|I|$ time steps that require edge e at their time of injection. Such a model allows for adversarial injection of packets that are "bursty" using the integer $b > 0$.

We say that a packet p *requires* an edge e at time t if the edge e lies on the path from its position to its destination at time t. For proving lower bounds on injection rate that guarantee instability, it is advantageous to have an adversary that is as weak as possible. Thus, we assume that $b = 0$. Given a network $\mathcal{G}$ and an edge $e \in \mathcal{G}$, we denote by $Q(e)$ the queue at e and we denote by $e(t)$ the size of $Q(e)$ at time t.

In order to formalize the behavior of a network under the adversarial queueing model, we use the notions of *system* and *system configuration.* A triple of the form $\langle \mathcal{G}, \mathcal{A}, \mathsf{P} \rangle$ where $\mathcal{G}$ is a network, $\mathcal{A}$ is an adversary and P is the used protocol on the network queues is called a system. The execution of the system proceeds in global time steps numbered $0, 1, \ldots$. Each time-step is divided in two sub-steps. In the first sub-step, one packet is sent from each non-empty buffer over its corresponding link. In the second sub-step, packets are received by the nodes at the other end of the links; they are absorbed (eliminated) if that node is their destination, and otherwise they are placed in the buffer of the next link on their respective routes. New packets are injected in the second sub-step.

In every time step t, the current configuration C^t of a system $\langle \mathcal{G}, \mathcal{A}, \mathsf{P} \rangle$ is a collection of sets $\{S_e^t : e \epsilon \mathcal{G}\}$, such that S_e^t is the set of packets waiting in the queue of the edge e at the end of step t. If the current system configuration is C^t, we obtain the system configuration C^{t+1} for the next time step as follows: (i) Addition of new packets to some of the sets S_e^t, each of which has an assigned path in $\mathcal{G}$, and (ii) for each non-empty set S_e^t deletion of a single packet $p \epsilon S_e^t$ and its insertion into the set S_f^{t+1} where f is the edge following e on its assigned path (if e is the last edge on the path of p, then p is not inserted into any set.) A time evolution of the system for an adversary of rate (r, b) is a sequence of such configurations $C^1, C^2, \ldots$, such that for all edges e and all intervals I, no more than $r|I| + b$ packets are introduced during I with an assigned path containing e.

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. A *greedy* contention-resolution protocol always specifies some packet to move along edge e if there are packets waiting to use edge e. In this work, we restrict attention to deterministic, greedy contention-resolution protocols. In particular, we consider:

- SIS (*Shortest-in-System*) gives priority to the most recently injected packet into the network;
- LIS (*Longest-in-System*) gives priority to the least recently injected packet into the network;
- FTG (*Furthest-to-Go*) gives priority to the packet that has to traverse the larger number of edges to its destination;
- NTS (*Nearest-to-Source*) gives priority to the packet that has traversed the smallest number of edges from its origin;
- FIFO (*First-In-First-Out*) gives priority to the earliest arrived packet at a queue;
- NTG-U-LIS (*Nearest-To-Go-Using-LIS*) gives priority to the nearest packet to its destination or the least recently injected packet for tie-breaking.

All these contention-resolution protocols require some tie-breaking rule in order to be unambiguously defined. In this work, whenever we are proving a positive result, we assume that the adversary can break the tie arbitrarily; for proving a negative result, we can assume any well-determined tie breaking rule for the adversary.

In our adversarial constructions for FIFO protocol we exploit the *fair mixing* property of FIFO according to which if two packet sets arrive at the same queue simultaneously will mix according to the initial proportions of their sizes. Also, we consider $\alpha(\mathcal{G}) > 1$ because if

$\alpha(\mathcal{G}) = 1$ then we have a tree or a ring that is known to be universally stable [1]. An important preliminary result on FIFO stability is given below.

Proposition 2.1 (Diaz *et al.* **[7])** *Let $r'_{\mathcal{G}}$ be a real number in $(0, 1)$ satisfying the equation $\frac{2-r'_{\mathcal{G}}}{1-r'_{\mathcal{G}}} r'_{\mathcal{G}} \Sigma_{i=0}^{d(\mathcal{G})-1} (\alpha(\mathcal{G}) + r'_{\mathcal{G}})^i = \frac{1}{m(\mathcal{G})}$. Then for any network $\mathcal{G}$, and any adversary with $r \leq r'_{\mathcal{G}}$ the system $\langle \mathcal{G}, \mathcal{A}, \mathsf{FIFO} \rangle$ is stable.*

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.* [1, Lemma 2.9]. Also, for simplicity, and in a way similar to that in [1], we omit floors and ceilings and sometimes count time steps and packets roughly. This only results to losing small additive constants while we gain in clarity.

3 Stability in Heterogeneous Networks

In this section we prove lower bounds on injection rate that guarantee instability for heterogeneous networks. In our proof we distinguish two types of packet injections:

- We denote by X_i the set of packets that are injected into the system in the i^{th} round of a phase. These packet sets are characterized as *investing flows* because they will remain in the system till the beginning of the next phase.
- We denote by $S_{i,j}$ the j^{th} set of packets the adversary injects into the system in the i^{th} round of a phase. These packet sets are characterized as *short intermediate flows* because they are injected on judiciously chosen paths of the network for blocking investing flows.

3.1 A Parameterized Network Family

We provide here a parameterized family of heterogeneous networks $\mathcal{N}_k$. The motivation that led us to such a parameterization in the network topology is *two-fold*:

- The existence of many parallel queues in the network allows the adversary to simultaneously inject several short intermediate flows that block the investing flows in the system, without violating the rule of the restricted adversarial model.

- Such a parameterized network topology construction, enables a parameterized analysis of the system configuration evolution into distinguished rounds whose number depends on the parameterized network topology. In LIS-FTG composition, the parameterization, besides the parallel edges, includes additional chains of queues for the exploitation of FTG in blocking investing flows.

3.2 A Parameterized Adversarial Construction

In order for our adversarial construction to work, we split the time into phases. In each phase we study the evolution of the system configuration by considering distinguished time rounds. For each phase, we inductively show that the number of packets in the system increases. Applying repeatedly this inductive argument we show instability.

Theorem 3.1 *Let $r > 0.5$. There is a network $\mathcal{N}_k$ where k is a parameter linear to the number of network queues and an adversary $\mathcal{A}$ of rate r such that the system $\langle \mathcal{N}_k, \mathcal{A}, \mathsf{Pr} \rangle$ is unstable if* Pr *is a composition of* LIS *protocol with any protocol of a)* SIS*, b)* NTS *and c)* FTG*.*

Proof: **Part a)** This proof is based on the preservation of all the investing flows injected during a phase into the system. We consider an instance of the parameterized network family (network $\mathcal{N}_k$ in Figure 3). All the queues use the LIS protocol except the queues $f_1, f_1^{'}, h_1, \ldots, h_{k-1}$, $h_1^{'}, \ldots, h_{k-1}^{'}$ that use the SIS protocol. Moreover, the queues $h_k, h_k^{'}$ can use either LIS or SIS protocol because there is no packet conflict in them.

Inductive Hypothesis: At the beginning of phase j, there are s_j packets that are queued in $g_1^{'}, h_1^{'}, \ldots, h_{k-1}^{'}$ requiring to traverse the edges e_0, f_1, f_2, g_1, h_1.

Induction Step: At the beginning of phase $j+1$ there will be more than s_j packets (s_{j+1} packets) that will be queued in $g_1, h_1, \ldots, h_{k-1}$ requiring to traverse the edges $e_1, f_1^{'}, f_2^{'}, g_1^{'}, h_1^{'}$.

We construct an adversary $\mathcal{A}$ such that the induction step holds. 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 $g_1^{'}, h_1^{'}, \ldots, h_{k-1}^{'}$ requiring to traverse the edges e_0, f_1, f_2, g_1, h_1. In order to prove that the induction step works it is assumed that there is a large enough number s_j of packets in the initial system configuration.

Phase j consists of $l = k+1$ rounds with $l \geq 3$, that is $k \geq 2$. The sequence of injections is as follows:

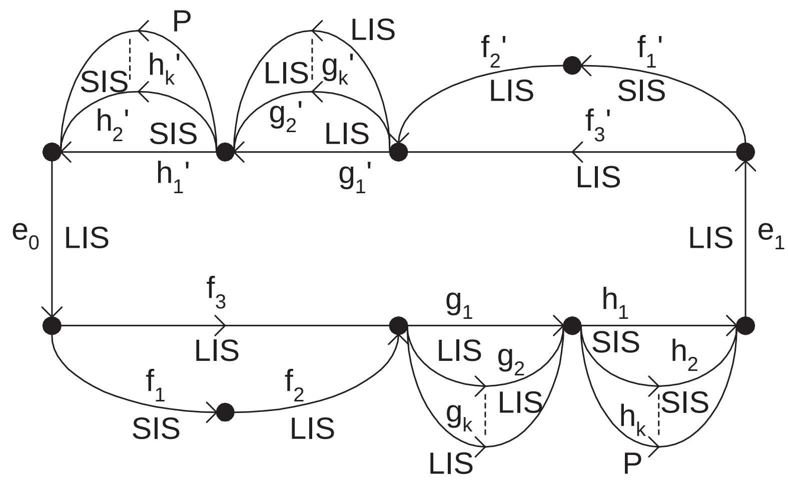

Figure 3: A network $\mathcal{N}_k$ that uses LIS-SIS protocols. (P can be either LIS or SIS.)

- **Round 1:** It lasts $|T_1| = s_j$ time steps.

  Adversary's behavior. During this round, the adversary injects a set X_1 of $|X_1| = r|T_1|$ packets in queue e_0 wanting to traverse the edges $e_0, f_3, g_1, h_1, e_1, f_1^{'}, f_2^{'}, g_1^{'}, h_1^{'}$ and a set $S_{1,1}$ of $|S_{1,1}| = r|T_1|$ packets in queue f_1 that require to traverse only the edge f_1.

  Evolution of the system configuration. $S - flow$ packets have priority over X_1 packets in queue e_0 because it uses LIS protocol. Therefore, X_1 packets remain in queue e_0 at the end of this round. After traversing the edge e_0, $S - flow$ packets are delayed by $S_{1,1}$ packets in queue f_1 because it uses SIS protocol. Thus, at the end of this round a portion Y of $|Y| = r|T_1|$ packets from $S - flow$ packets remain in queue f_1, while $S_{1,1}$ packets traverse the edge f_1 and they are absorbed.

- **Round 2:** It lasts $|T_2| = r|T_1|$ time steps.

  Adversary's behavior. During this round, the adversary injects a set X_2 of $|X_2| = r|T_2|$ packets in queue g_1 requiring to traverse the edges $g_1, h_2, e_1, f_1^{'}, f_2^{'}, g_1^{'}, h_1^{'}$. Also, it injects a set $S_{2,1}$ of $|S_{2,1}| = r|T_2|$ packets in queue f_2 wanting to traverse the edges f_2, g_2, h_1.

  Evolution of the system configuration. Y packets have priority over X_1 and $S_{2,1}$ packets in queues g_1 and f_2 correspondingly because these queues use LIS protocol. Furthermore, X_1 packets have priority over X_2 packets in queue g_1 because it uses the LIS protocol.

Since the number of rounds depends on the network topology (i.e. $l = k + 1$), we next analyze an intermediate round t, $3 \leq t < l$.

- **Round t (intermediate round):** It lasts $|T_t| = r|T_{t-1}|$ time steps. For readability reasons, we structure the description of this round in three distinguished parts. The first part deals with the adversary's behavior during round t, the second part discusses how the system configuration evolves, and the third part proves why this happens.

  Adversary's behavior. During this round, the adversary injects $t-1$ short intermediate flows $S_{t,1}, \ldots, S_{t,t-1}$ of $|S_{t,1}| = \ldots = |S_{t,t-1}| = r|T_t|$ packets. Each packet flow $S_{t,j}$ $(2 \leq j \leq t-1)$ is injected in queue g_j wanting to traverse the edges g_j, h_j. On the other hand, the packet flow $S_{t,1}$ is injected in queue f_2 wanting to traverse the edges f_2, g_t, h_1. In addition, the adversary injects the investing flow X_t of $|X_t| = r|T_t|$ packets in queue g_1 wanting to traverse the edges $g_1, h_t, e_1, f_1^{'}, f_2^{'}, g_1^{'}, h_1^{'}$.

  Evolution of the system configuration. $S_{t-1,j}$ packets $(2 \leq j \leq t-2)$ have priority over $S_{t,j}$ packets in queue g_j due to the LIS protocol. Also, $S_{t-1,1}$ packets have priority over $S_{t,1}$ and $S_{t,t-1}$ packets in queues f_2 and g_{t-1} correspondingly because these queues use the LIS protocol and $S_{t-1,1}$ packets are longer time in the system than $S_{t,1}$ and $S_{t,2}$ packets.

  At the end of round t, there is a number of $t-3$ different cases for the queues where $X_1, \ldots, X_{t-1}$ packets (that have been injected at rounds $1, \ldots, t-1$ correspondingly) are queued depending on their position at the beginning of round t and the injection rate r. Note that although these cases represent different system configurations with respect to the position of investing flows, the evolution in each case follows the same rules, as it is expressed in the claim below. We next illustrate case i $(1 \leq i \leq t-3)$:

  Case i: At the beginning of round t, a portion or all X_i packets along with $X_{i+1}, \ldots, X_{t-1}$ are queued in g_1, while the rest X_i packets are queued in h_i and all the $X_1, \ldots, X_{i-1}$ packets are queued in $h_1, \ldots, h_{i-1}$ correspondingly.

  At the end of round t, two subcases are possible:

  - $X_1, \ldots, X_{i-1}$ packets are queued in $h_1, \ldots, h_{i-1}$ correspondingly. A portion or all X_i packets in queue g_1 are queued with the rest X_i packets in h_i and $X_{i+1}, \ldots, X_{t-1}$ packets remain in g_1.
  - $X_1, \ldots, X_{i-1}$ packets are queued in $h_1, \ldots, h_{i-1}$ correspondingly. All the X_i packets in g_1 are queued with the rest X_i packets in h_i, a portion of X_{i+1} packets is queued in h_{i+1}, while the rest X_{i+1} packets along with $X_{i+2}, \ldots, X_{t-1}$ packets remain in queue g_1.

  X_{t-1} packets have priority over X_t packets in queue g_1 that uses LIS protocol because X_{t-1} packets are longer time in the system. Therefore, X_t packets remain in queue g_1 at the end of this round. Note that, in all possible system configurations at the end of round t, *the investing flows* $X_1, \ldots, X_t$ *remain into the system.* However, we have not proved yet why the system configuration evolves like that in each case. This is ensured by the following technical claim.

Claim 3.2 *If at the beginning of round* t *a portion or all* X_i $(1 \leq i \leq t-3)$ *packets are queued at the head of* g_1*, while the rest are queued in* h_i*,* $X_{i+1}, \ldots, X_{t-1}$ *packets are queued in* g_1*, and all* $X_1, \ldots, X_{i-1}$ *packets are queued in* $h_1, \ldots, h_{i-1}$ *correspondingly, then at the end of round* t *(i)* X_i *packets are delayed in* h_i*, (ii)* X_{i+1} *packets are delayed in* h_{i+1} *if* X_i *packets traverse* g_1 *before the end of* t *otherwise* X_{i+1} *packets remain in* g_1*, (iii) the investing flows* $X_{i+2}, \ldots, X_t$ *remain in* g_1*, and (iv) the investing flows* $X_1, \ldots, X_{i-1}$ *continue to remain in* $h_1, \ldots, h_{i-1}$ *correspondingly.*

Proof: Consider that *at the beginning of round* t, a portion or all X_i packets along with $X_{i+1}, \ldots, X_{t-1}$ are queued in g_1, while the rest X_i packets are queued in h_i and all the $X_1, \ldots, X_{i-1}$ packets are queued in $h_1, \ldots, h_{i-1}$ correspondingly.

At the end of round t, two subcases are possible:

- The short intermediate flows $S_{t-1,1}, \ldots, S_{t-1,i-1}$ block $X_1, \ldots, X_{i-1}$ packets in queues $h_1, \ldots, h_{i-1}$ correspondingly that use SIS protocol because $S_{t-1,1}, \ldots, S_{t-1,i-1}$ flows are shortest time in the system. A portion or all X_i packets in queue g_1 traverse it (there are no remaining time steps till the end of the round) and they are queued with the rest X_i packets in h_i that uses SIS protocol, where X_i packets continue to be blocked by the $S_{t-1,i} - flow$ packets that are shortest time in the system than X_i packets. Furthermore, $X_{i+1}, \ldots, X_t$ packets are queued in g_1 that uses LIS due to X_i packets that are in the system for a longer time.
- The short intermediate flows $S_{t-1,1}, \ldots, S_{t-1,i-1}$ block $X_1, \ldots, X_{i-1}$ packets in queues $h_1, \ldots, h_{i-1}$ correspondingly that use SIS protocol because $S_{t-1,1}, \ldots, S_{t-1,i-1}$ flows are shortest time in the system. All the X_i packets traverse the edge g_1 during this round before the end of this round. These packets are queued with the rest X_i packets in h_i, where they continue to be blocked by $S_{t-1,i}$ packets that are nearest to their source. Also, a portion of X_{i+1} packets is blocked in queue h_{i+1} that uses SIS protocol by $S_{t-1,i+1}$ packets that are shortest time in the system. The rest X_{i+1} packets along with $X_{i+2}, \ldots, X_t$ packets are queued in queue g_1. ■

Lemma 3.3 *The short intermediate flows* $S_{t-1,1}, \ldots, S_{t-1,t-2}$ *delay all the packets of the investing flows* $X_1, \ldots, X_t$ *in the system* $\langle \mathcal{N}_k, \mathcal{A}, \mathsf{LIS} - \mathsf{SIS} \rangle$ *till the end of round* t*.*

Proof: Due to Claim 3.2 the short intermediate flows $S_{t-1,1}, \ldots, S_{t-1,t-2}$ (that have been injected in the system during round $t-1$) delay the investing flows $X_1, \ldots, X_{t-2}$ in the system. In addition, the investing flow X_{t-1} is blocked in g_1 that uses LIS protocol by X_{t-2} flow. Also, note that the investing flow X_t that is injected by the adversary during round t is simultaneously blocked in g_1 by X_{t-1} flow because g_1 uses LIS protocol and X_{t-1} is in the system for a longer time compared to X_t since it has been injected in the system during round $t-1$. Therefore, the short intermediate flows $S_{t-1,1}, \ldots, S_{t-1,t-2}$ are enough to keep all the packets of the investing flows $X_1, \ldots, X_t$ in the system till the end of round t. ■

- **Round l:** It lasts $|T_l| = r|T_{l-1}|$ time steps.

  Adversary's behavior. During this round, the adversary injects an investing flow X_l of $|X_l| = r|T_l|$ packets in queue g_1 wanting to traverse the edges $g_1, h_l, e_1, f_1^{'}, f_2^{'}, g_1^{'}, h_1^{'}$.

  Evolution of the system configuration. X_{l-1} packets (that have been injected in the system at round $l-1$) have priority over X_l packets in queue g_1. If we follow a similar analysis as in the intermediate round t, we can prove that the short intermediate flows $S_{l-1,1}, \ldots, S_{l-1,l-2}$ have priority over $X_1, \ldots, X_{l-1}$ packets in the system. Therefore, at the end of round l, the number of packets that are queued in $g_1, h_1, \ldots, h_{l-2} = h_{k-1}$ requiring to traverse the edges $e_1, f_1^{'}, f_2^{'}, g_1^{'}, h_1^{'}$ is $s_{j+1} = |X_1| + \ldots + |X_l|$.

  In order to have instability, we must have $s_{j+1} > s_j$. This holds for $r^{k+2} - 2r + 1 < 0$. This argument can be repeated for an infinite number of phases ensuring the instability of the system $\langle \mathcal{N}_k, \mathcal{A}, \mathsf{LIS} - \mathsf{SIS} \rangle$. Also, $k \to \infty \Longrightarrow r^{k+2} \to 0$, because $0 < r < 1$. Thus, for instability it suffices $-2r + 1 < 0$, i.e. $r > 0.5$.

Part b) This part of the proof is similar to the first one. In particular, the topology of the used network in this part is similar to the first one. One difference in the network of *Part b* is the use of NTS protocol where SIS is used in the network of *Part a*. As in *Part a* the basic argument behind this long technical proof is that we have found a way to keep all the investing flows injected during a time period structured in rounds, into the system. More specifically, the injection of short intermediate flows with the same paths as in *Part a* is enough to guarantee their priority over investing flows when they conflict in queues that use NTS. This is shown in an inductive way, by demonstrating that all investing flows, injected till the current round, are still into the system.

Consider the network $\mathcal{N}_k$ in Figure 4. All the queues of the network use the LIS protocol except the queues that correspond to the edges $f_1, f_1^{'}, h_1, \ldots, h_{k-1}, h_1^{'}, \ldots, h_{k-1}^{'}$ that use the

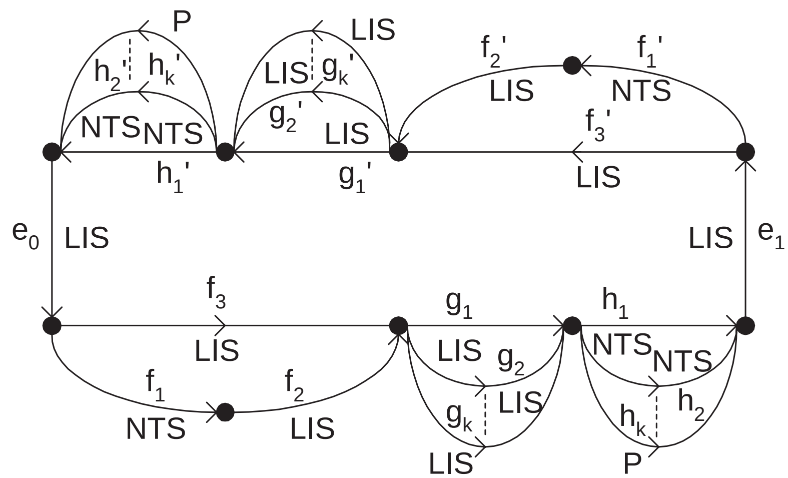

Figure 4: A network $\mathcal{N}_k$ that uses LIS-NTS protocols. (P can be either LIS or NTS.)

NTS protocol. Moreover, the queues $h_k, h_k^{'}$ can use either LIS or NTS protocol because there is no packet conflict in them.

Inductive Hypothesis: At the beginning of phase j, there are s_j packets that are queued in the queues $g_1^{'}, h_1^{'}, \ldots, h_{k-1}^{'}$ requiring to traverse the edges e_0, f_1, f_2, g_1, h_1.

Induction Step: At the beginning of phase $j+1$ there will be more than s_j packets (s_{j+1} packets) that will be queued in the queues $g_1, h_1, \ldots, h_{k-1}$ requiring to traverse the edges $e_1, f_1^{'}, f_2^{'}, g_1^{'}, h_1^{'}$.

We will construct an adversary $\mathcal{A}$ 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 $j+1$ for the symmetric the 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 $g_1^{'}, h_1^{'}, \ldots, h_{k-1}^{'}$ requiring to traverse the edges e_0, f_1, f_2, g_1, h_1. In order to prove that the induction step works it is assumed that there is a large enough number s_j of packets in the initial system configuration.

Phase j consists of $l = k+1$ rounds with $l \geq 3$, that is $k \geq 2$. The sequence of injections is as follows:

- **Round 1:** It lasts $|T_1| = s_j$ time steps.

  Adversary's behavior. During this round, the adversary injects a set X_1 of $|X_1| = r|T_1|$ packets in queue e_0 wanting to traverse the edges $e_0, f_3, g_1, h_1, e_1, f_1^{'}, f_2^{'}, g_1^{'}, h_1^{'}$ and a set $S_{1,1}$ of $|S_{1,1}| = r|T_1|$ packets in queue f_1 that require to traverse only the edge f_1.

Evolution of the system configuration. $S - flow$ packets have priority over X_1 packets in queue e_0 because it uses LIS protocol. So, for every arrival of an injected X_1 packet in e_0 there is at least one $S - flow$ packet there that is in the system for a longer time, so it has priority. Therefore, X_1 packets remain in queue e_0 at the end of this round. After traversing the edge e_0, $S - flow$ packets are delayed by $S_{1,1}$ packets in queue f_1 because it uses NTS protocol and $S_{1,1}$ packets are nearest to their source than $S - flow$ packets. So, all the $S_{1,1}$ packets traverse f_1 and they are absorbed along with some packets from $S - flow$. Thus, at the end of this round a portion Y of $|Y| = r|T_1|$ packets from $S - flow$ packets remain in queue f_1.

- **Round 2:** It lasts $|T_2| = r|T_1|$ time steps.

  Adversary's behavior. During this round, the adversary injects a set X_2 of $|X_2| = r|T_2|$ packets in queue e_0 requiring to traverse the edges $e_0, f_3, g_1, h_2, e_1, f_1^{'}, f_2^{'}, g_1^{'}, h_1^{'}$. Also, it injects a set $S_{2,1}$ of $|S_{2,1}| = r|T_2|$ packets in queue f_2 wanting to traverse the edges f_2, g_2, h_1.

  Evolution of the system configuration. Y packets have priority over X_1 and $S_{2,1}$ packets in queues g_1 and f_2 correspondingly because these queues use LIS protocol and Y packets are longer time in the system than X_1 and $S_{2,1}$ packets. Furthermore, X_1 packets have priority over X_2 packets in queue e_0 because it uses the LIS protocol.

Since the number of rounds depends on the network topology (i.e. $l = k + 1$), we next analyze an intermediate round t, $3 \leq t < l$.

- **Round t (intermediate round):** It lasts $|T_t| = r|T_{t-1}|$ time steps. For readability reasons, we structure the description of this round in three distinguished parts. The first part deals with the adversary's behavior during round t, the second part discusses how the system configuration evolves, and the third part proves why this happens.

  Adversary's behavior. During this round, the adversary injects $t - 1$ short intermediate flows $S_{t,1}, \ldots, S_{t,t-1}$ of $|S_{t,1}| = \ldots = |S_{t,t-1}| = r|T_t|$ packets. Each packet flow $S_{t,j}$ $(2 \leq j \leq t - 1)$ is injected in queue g_j wanting to traverse the edges g_j, h_j. On the other hand, the packet flow $S_{t,1}$ is injected in queue f_2 wanting to traverse the edges f_2, g_t, h_1. In addition, the adversary injects the investing flow X_t of $|X_t| = r|T_t|$ packets in queue f_3 wanting to traverse the edges $f_3, g_1, h_t, e_1, f_1^{'}, f_2^{'}, g_1^{'}, h_1^{'}$.

  Evolution of the system configuration. $S_{t-1,j}$ packets $(2 \leq j \leq t - 2)$ have priority over $S_{t,j}$ packets in queue g_j due to the LIS protocol. Also, $S_{t-1,1}$ packets have priority over $S_{t,1}$ and $S_{t,t-1}$ packets in queues f_2 and g_{t-1} correspondingly because these queues use the LIS protocol and $S_{t-1,1}$ packets are longer time in the system than $S_{t,1}$ and $S_{t,t-1}$ packets.

At the end of round t, there is a number of $t-3$ different cases for the queues where $X_1, \ldots, X_{t-1}$ packets (that have been injected at rounds $1, \ldots, t-1$ correspondingly) are queued depending on their position at the beginning of round t and the injection rate r. Note that although these cases represent different system configurations with respect to the position of investing flows, the evolution in each case follows the same rules, as it is expressed in the claim below. We next illustrate case i ($1 \leq i \leq t-3$):

Case i: At the beginning of round t, X_{t-1} packets are queued in f_3, a portion or all X_i packets along with $X_{i+1}, \ldots, X_{t-2}$ are queued in g_1, while the rest X_i packets are queued in h_i and all the $X_1, \ldots, X_{i-1}$ packets are queued in $h_1, \ldots, h_{i-1}$ correspondingly.

At the end of round t, two subcases are possible:

- $X_1, \ldots, X_{i-1}$ packets are queued in $h_1, \ldots, h_{i-1}$ correspondingly. A portion or all X_i packets in queue g_1 are queued with the rest X_i packets in h_i and $X_{i+1}, \ldots, X_{t-1}$ packets are queued in g_1.
- $X_1, \ldots, X_{i-1}$ packets are queued in $h_1, \ldots, h_{i-1}$ correspondingly. All the X_i packets in g_1 are queued with the rest X_i packets in h_i, a portion of X_{i+1} packets is queued in h_{i+1}, while the rest X_{i+1} packets along with $X_{i+2}, \ldots, X_{t-1}$ packets are queued in queue g_1.

X_{t-1} packets have priority over X_t packets in queue f_3 that uses LIS protocol because X_{t-1} packets are longer time in the system. Therefore, X_t packets remain in queue f_3 at the end of this round. Note that, in all possible system configurations at the end of round t, *the investing flows $X_1, \ldots, X_t$ remain into the system.* However, we have not proved yet why the system configuration evolves like that in each case. This is ensured by the following technical claim.

Claim 3.4 *If at the beginning of round t, a portion or all X_i ($1 \leq i \leq t-3$) packets are queued at the head of g_1, while the rest are queued in h_i, $X_{i+1}, \ldots, X_{t-2}$ packets are queued in h_i, X_{t-1} packets are queued in f_3 and all $X_1, \ldots, X_{i-1}$ packets are queued in $h_1, \ldots, h_{i-1}$ correspondingly, then at the end of round t (i) X_i packets are delayed in h_i, (ii) X_{i+1} packets are delayed in h_{i+1} if all X_i packets traverse g_1 before the end of t otherwise X_{i+1} packets remain in g_1, (iii) the investing flows $X_{i+2}, \ldots, X_{t-1}$ remain in g_1, (iv) the investing flow X_t remains in f_3, and (v) the investing flows $X_1, \ldots, X_{i-1}$ continue to remain in $h_1, \ldots, h_{i-1}$ correspondingly.*

Proof: Consider that *at the beginning of round* t, the investing flow X_{t-1} remains in f_3, a portion or all X_i packets along with $X_{i+1}, \ldots, X_{t-2}$ are queued in g_1, while the rest X_i packets are queued in h_i and all the $X_1, \ldots, X_{i-1}$ packets are queued in $h_1, \ldots, h_{i-1}$ correspondingly.

At the end of round t, two subcases are possible:

- The short intermediate flows $S_{t-1,1}, \ldots, S_{t-1,i-1}$ block $X_1, \ldots, X_{i-1}$ packets in queues $h_1, \ldots, h_{i-1}$ correspondingly that use NTS protocol because $S_{t-1,1}, \ldots, S_{t-1,i-1}$ flows are nearest to their source. A portion or all X_i packets in queue g_1 traverse it (there are no remaining time steps till the end of the round) and they are queued with the rest X_i packets in h_i that uses NTS protocol, where X_i packets continue to be blocked by the $S_{t-1,i} - flow$ packets that are nearest to their source than X_i packets. Furthermore, $X_{i+1}, \ldots, X_{t-1}$ packets are queued in g_1 that uses LIS due to X_i packets that are in the system for a longer time. Also, X_{t-1} packets delay X_t packets in f_3 that uses LIS because X_{t-1} packets are in the system for a longer time.
- The short intermediate flows $S_{t-1,1}, \ldots, S_{t-1,i-1}$ block $X_1, \ldots, X_{i-1}$ packets in queues $h_1, \ldots, h_{i-1}$ correspondingly that use NTS protocol because $S_{t-1,1}, \ldots, S_{t-1,i-1}$ flows are nearest to their source. All the X_i packets traverse the edge g_1 during this round before the end of this round. These packets are queued with the rest X_i packets in h_i, where they continue to be blocked by $S_{t-1,i}$ packets that are nearest to their source. Also, a portion of X_{i+1} packets is blocked in queue h_{i+1} that uses NTS protocol by $S_{t-1,i+1}$ packets that are nearest to their source. The rest X_{i+1} packets along with $X_{i+2}, \ldots, X_{t-1}$ packets are queued in queue g_1. Also, X_{t-1} packets delay X_t packets in f_3 that uses LIS because X_{t-1} packets are in the system for a longer time. ■

Lemma 3.5 *The short intermediate flows $S_{t-1,1}, \ldots, S_{t-1,t-2}$ delay all the packets of the investing flows $X_1, \ldots, X_t$ in the system $\langle \mathcal{N}_k, \mathcal{A}, \mathsf{LIS} - \mathsf{NTS} \rangle$ till the end of round t.*

Proof: Due to Claim 3.4 the short intermediate flows $S_{t-1,1}, \ldots, S_{t-1,t-2}$ (that have been injected in the system during round $t-1$) block the investing flows $X_1, \ldots, X_{t-2}$ in the system. In addition, the investing flow X_{t-1} is blocked in g_1 that uses LIS protocol by X_{t-2} flow. Also, note that the investing flow X_t that is injected by the adversary during round t is simultaneously blocked in f_3 by X_{t-1} flow because f_3 uses LIS protocol and X_{t-1} is in the system for a longer time compared to X_t since it has been injected in the system during round $t-1$. Therefore, the short intermediate flows $S_{t-1,1}, \ldots, S_{t-1,t-2}$ are enough to keep all the packets of the investing flows $X_1, \ldots, X_t$ in the system till the end of round t. ■

- **Round l:** It lasts $|T_l| = r|T_{l-1}|$ time steps.

  Adversary's behavior. During this round, the adversary injects an investing flow X_l of $|X_l| = r|T_l|$ packets in queue f_3 wanting to traverse the edges $g_1, h_l, e_1, f_1^{'}, f_2^{'}, g_1^{'}, h_1^{'}$.

  Evolution of the system configuration. X_{l-1} packets (that have been injected in the system at round $l-1$) have priority over X_l packets in queue g_1. If we follow a similar analysis as in the intermediate round t, we can prove that the short intermediate flows $S_{l-1,1}, \ldots, S_{l-1,l-2}$ have priority over $X_1, \ldots, X_{l-1}$ packets in the system. Therefore, at the end of round l, the number of packets that are queued in $g_1, h_1, \ldots, h_{l-2} = h_{k-1}$ requiring to traverse the edges $e_1, f_1^{'}, f_2^{'}, g_1^{'}, h_1^{'}$ is $s_{j+1} = |X_1| + \ldots + |X_l|$.

  In order to have instability, we must have $s_{j+1} > s_j$. This holds for $r^{k+2} - 2r + 1 < 0$. This argument can be repeated for an infinite number of phases ensuring the instability of the system $\langle \mathcal{N}_k, \mathcal{A}, \mathsf{LIS} - \mathsf{NTS} \rangle$. Also, $k \to \infty \Longrightarrow r^{k+2} \to 0$, because $0 < r < 1$. Thus, for instability it suffices $-2r + 1 < 0$, i.e. $r > 0.5$.

Part c) This part of the proof is similar to its spirit with the other two parts. However, the topology of the used network in this part has two significant differences with the networks that are used in the other parts. One difference in the network of *Part c* is the use of FTG protocol where SIS is used in the network of *Part a*. Another difference is that the network contains additional paths, comparing to the other two cases, that start at queues that use FTG. These paths have sufficient lengths, such that the injected short intermediate packet flows have the same blocking effects over the injected investing packet flows when they conflict in queues that use FTG, as happens in LIS-SIS and LIS-NTS cases. This is shown in an inductive way, by demonstrating that all investing flows, injected till the current round, are still into the system.

Consider the network $\mathcal{N}_k$ in Figure 5. All the queues of the network use the LIS protocol except the queues that correspond to the edges $f_1, f_1^{'}, h_1, \ldots, h_{k-1}, h_1^{'}, \ldots, h_{k-1}^{'}$ that use the FTG protocol. Moreover, the edges $h_k, h_k^{'}$ and the edges l_0, l_1, l_2, $l_0^{'}, l_1^{'}, l_2^{'}$, $g_{1,1}, g_{1,2}, g_{1,3}, g_{1,4}, g_{1,5}, \ldots$, $g_{k-1,1}, g_{k-1,2}, g_{k-1,3}, g_{k-1,4}, g_{k-1,5}$ and $g_{1,1}^{'}, g_{1,2}^{'}, g_{1,3}^{'}, g_{1,4}^{'}, g_{1,5}^{'}, \ldots, g_{k-1,1}^{'}, g_{k-1,2}^{'}, g_{k-1,3}^{'}, g_{k-1,4}^{'}$, $g_{k-1,5}^{'}$ can use either LIS or FTG protocol because there is no packet conflict in them.

Inductive Hypothesis: At the beginning of phase j, there are s_j packets that are queued in the queues $g_1^{'}, h_1^{'}, \ldots, h_{k-1}^{'}$ requiring to traverse the edges e_0, f_1, f_2, g_1.

Induction Step: At the beginning of phase $j+1$ there will be more than s_j packets (s_{j+1} packets) that will be queued in the queues $g_1, h_1, \ldots, h_{k-1}$ requiring to traverse the edges $e_1, f_1^{'}, f_2^{'}, g_1^{'}$.

We will construct an adversary $\mathcal{A}$ such that the induction step will hold. Proving that the

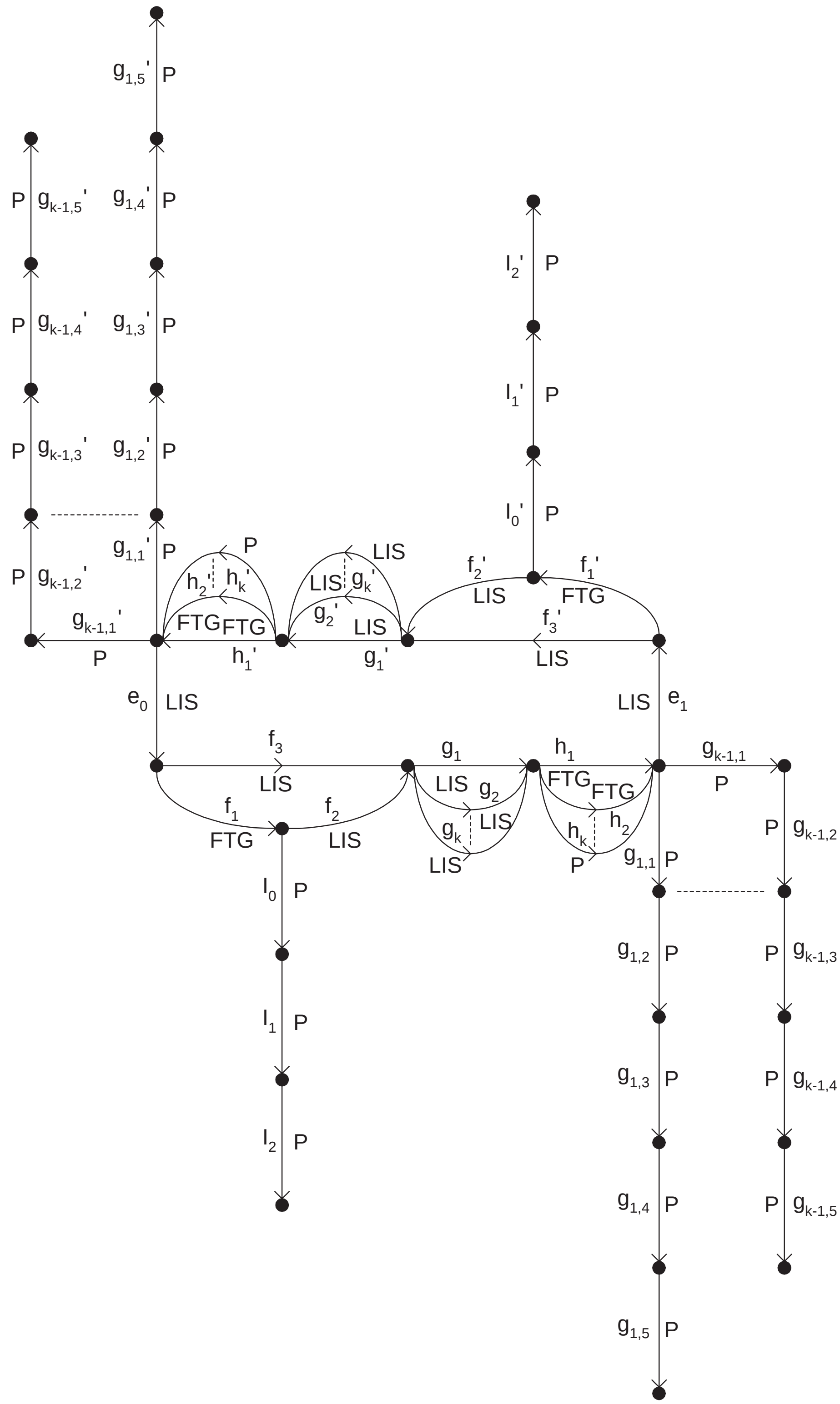

Figure 5: A network $\mathcal{N}_k$ that uses LIS-FTG protocols. (P can be either LIS or FTG.)

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 $g_1^{'}, h_1^{'}, \ldots, h_{k-1}^{'}$ requiring to traverse the edges e_0, f_1, f_2, g_1. In order to prove that the induction step works it is assumed that there is a large enough number s_j of packets in the initial system configuration.

Phase j consists of $l = k+1$ rounds with $l \geq 3$, that is $k \geq 2$. The sequence of injections is as follows:

- **Round 1:** It lasts $|T_1| = s_j$ time steps.

  Adversary's behavior. During this round, the adversary injects a set X_1 of $|X_1| = r|T_1|$ packets in queue e_0 wanting to traverse the edges $e_0, f_3, g_1, h_1, e_1, f_1^{'}, f_2^{'}, g_1^{'}$ and a set $S_{1,1}$ of $|S_{1,1}| = r|T_1|$ packets in queue f_1 that require to traverse the edges f_1, l_0, l_1, l_2.

  Evolution of the system configuration. $S-flow$ packets have priority over X_1 packets in queue e_0 because it uses LIS protocol. So, for every arrival of an injected X_1 packet in e_0 there is at least one $S-flow$ packet there that is in the system for a longer time, so it has priority. Therefore, X_1 packets remain in queue e_0 at the end of this round. After traversing the edge e_0, $S-flow$ packets are delayed by $S_{1,1}$ packets in queue f_1 because it uses FTG protocol and $S_{1,1}$ packets have furthest to go than $S-flow$ packets. So, all the $S_{1,1}$ packets traverse f_1 along with some packets from $S-flow$. Thus, at the end of this round a portion Y of $|Y| = r|T_1|$ packets from $S-flow$ packets remain in queue f_1 wanting to traverse the edges f_1, f_2, g_1.

- **Round 2:** It lasts $|T_2| = r|T_1|$ time steps.

  Adversary's behavior. During this round, the adversary injects a set X_2 of $|X_2| = r|T_2|$ packets in queue e_0 requiring to traverse the edges $e_0, f_3, g_1, h_2, e_1, f_1^{'}, f_2^{'}, g_1^{'}$. Also, it injects a set $S_{2,1}$ of $|S_{2,1}| = r|T_2|$ packets in queue f_2 wanting to traverse the edges $f_2, g_2, h_1, g_{1,1}, g_{1,2}, g_{1,3}, g_{1,4}, g_{1,5}$.

  Evolution of the system configuration. Y packets have priority over X_1 and $S_{2,1}$ packets in queues g_1 and f_2 correspondingly because these queues use LIS protocol and Y packets are longer time in the system than X_1 and $S_{2,1}$ packets. Furthermore, X_1 packets have priority over X_2 packets in queue e_0 because it uses the LIS protocol.

  Since the number of rounds depends on the network topology (i.e. $l = k+1$), we next analyze an intermediate round t, $3 \leq t < l$.

- **Round t (intermediate round):** It lasts $|T_t| = r|T_{t-1}|$ time steps. For readability reasons, we structure the description of this round in three distinguished parts. The first part deals with the adversary's behavior during round t, the second part discusses how the system configuration evolves, and the third part proves why this happens.

Adversary's behavior. During this round, the adversary injects $t-1$ short intermediate flows $S_{t,1}, \ldots, S_{t,t-1}$ of $|S_{t,1}| = \ldots = |S_{t,t-1}| = r|T_t|$ packets. Each packet flow $S_{t,j}$ ($2 \leq j \leq t-1$ is injected in queue g_j wanting to traverse the edges $g_j, h_j, g_{j,1}, g_{j,2}, g_{j,3}, g_{j,4}, g_{j,5}$. On the other hand, the packet flow $S_{t,1}$ is injected in queue f_2 wanting to traverse the edges $f_2, g_t, h_1, g_{1,1}, g_{1,2}, g_{1,3}, g_{1,4}, g_{1,5}$. In addition, the adversary injects the investing flow X_t of $|X_t| = r|T_t|$ packets in queue g_1 wanting to traverse the edges $g_1, h_t, e_1, f_1^{'}, f_2^{'}, g_1^{'}$.

Evolution of the system configuration. $S_{t-1,j}$ packets ($2 \leq j \leq t-2$) have priority over $S_{t,j}$ packets in queue g_j due to the LIS protocol. Also, $S_{t-1,1}$ packets have priority over $S_{t,1}$ and $S_{t,t-1}$ packets in queues f_2 and g_{t-1} correspondingly because these queues use the LIS protocol and $S_{t-1,1}$ packets are longer time in the system than $S_{t,1}$ and $S_{t,t-1}$ packets.

At the end of round t, there is a number of $t-3$ different cases for the queues where $X_1, \ldots, X_{t-1}$ packets (that have been injected at rounds $1, \ldots, t-1$ correspondingly) are queued depending on their position at the beginning of round t and the injection rate r. Note that although these cases represent different system configurations with respect to the position of investing flows, the evolution in each case follows the same rules, as it is expressed in the claim below. We next illustrate case i ($1 \leq i \leq t-3$):

Case i: At the beginning of round t, a portion or all X_i packets along with $X_{i+1}, \ldots, X_{t-1}$ are queued in g_1, while the rest X_i packets are queued in h_i and all the $X_1, \ldots, X_{i-1}$ packets are queued in $h_1, \ldots, h_{i-1}$ correspondingly.

At the end of round t, two subcases are possible:

- $X_1, \ldots, X_{i-1}$ packets are queued in $h_1, \ldots, h_{i-1}$ correspondingly. A portion or all X_i packets in queue g_1 are queued with the rest X_i packets in h_i and $X_{i+1}, \ldots, X_{t-1}$ packets remain in g_1.
- $X_1, \ldots, X_{i-1}$ packets are queued in $h_1, \ldots, h_{i-1}$ correspondingly. All the X_i packets in g_1 are queued with the rest X_i packets in h_i, a portion of X_{i+1} packets is queued in h_{i+1}, while the rest X_{i+1} packets along with $X_{i+2}, \ldots, X_{t-1}$ packets remain in queue g_1.

X_{t-1} packets have priority over X_t packets in queue g_1 that uses LIS protocol because X_{t-1} packets are longer time in the system. Therefore, X_t packets remain in queue g_1 at the end of this round. Note that, in all possible system configurations at the end of round

t, *the investing flows* $X_1, \ldots, X_t$ *remain into the system.* However, we have not proved yet why the system configuration evolves like that in each case. This is ensured by the following technical claim.

Claim 3.6 *If at the beginning of round* t *a portion or all* X_i $(1 \leq i \leq t-3)$ *packets are queued at the head of* g_1*, while the rest are queued in* h_i*, and all* $X_1, \ldots, X_{i-1}$ *packets are queued in* $h_1, \ldots, h_{i-1}$ *correspondingly, then at the end of round* t *(i)* X_i *packets are delayed in* h_i*, (ii)* X_{i+1} *packets are delayed in* h_{i+1} *if all* X_i *packets traverse* g_1 *before the end of* t *otherwise* X_{i+1} *packets remain in* g_1*, (iii) the investing flows* $X_{i+2}, \ldots, X_t$ *remain in* g_1*, and (iv) the investing flows* $X_1, \ldots, X_{i-1}$ *continue to remain in* $h_1, \ldots, h_{i-1}$ *correspondingly.*

Proof: Consider that *at the beginning of round* t, a portion or all X_i packets along with $X_{i+1}, \ldots, X_{t-1}$ are queued in g_1, while the rest X_i packets are queued in h_i and all the $X_1, \ldots, X_{i-1}$ packets are queued in $h_1, \ldots, h_{i-1}$ correspondingly.

At the end of round t, two subcases are possible:

- The short intermediate flows $S_{t-1,1}, \ldots, S_{t-1,i-1}$ block $X_1, \ldots, X_{i-1}$ packets in queues $h_1, \ldots, h_{i-1}$ correspondingly that use FTG protocol because $S_{t-1,1}, \ldots, S_{t-1,i-1}$ flows have farthest to go. A portion or all X_i packets in queue g_1 traverse it (there are no remaining time steps till the end of the round) and they are queued with the rest X_i packets in h_i that uses FTG protocol, where they continue to be blocked by the $S_{t-1,i} - flow$ packets that have farthest to go than X_i packets. Furthermore, $X_{i+1}, \ldots, X_t$ packets are queued in g_1 that uses LIS due to X_i packets that are in the system for a longer time.
- The short intermediate flows $S_{t-1,1}, \ldots, S_{t-1,i-1}$ block $X_1, \ldots, X_{i-1}$ packets in queues $h_1, \ldots, h_{i-1}$ correspondingly that use FTG protocol because $S_{t-1,1}, \ldots, S_{t-1,i-1}$ flows have farthest to go. All the X_i packets traverse the edge g_1 during this round before the end of this round. These packets are queued with the rest X_i packets in h_i, where they continue to be blocked by $S_{t-1,i}$ packets that are nearest to their source. Also, a portion of X_{i+1} packets is blocked in queue h_{i+1} that uses FTG protocol by $S_{t-1,i+1}$ packets that have farthest to go. The rest X_{i+1} packets along with $X_{i+2}, \ldots, X_t$ packets are queued in queue g_1. ■

Lemma 3.7 *The short intermediate flows* $S_{t-1,1}, \ldots, S_{t-1,t-2}$ *delay all the packets of the investing flows* $X_1, \ldots, X_t$ *in the system* $\langle \mathcal{N}_k, \mathcal{A}, \mathsf{LIS} - \mathsf{FTG} \rangle$ *till the end of round* t.

Proof: Due to Claim 3.6 the short intermediate flows $S_{t-1,1}, \ldots, S_{t-1,t-2}$ (that have been injected in the system during round $t-1$) block the investing flows $X_1, \ldots, X_{t-2}$ in the system. In addition, the investing flow X_{t-1} is blocked in g_1 that uses LIS protocol by X_{t-2} flow. Also, note that the investing flow X_t that is injected by the adversary during round t is simultaneously blocked in g_1 by X_{t-1} flow because g_1 uses LIS protocol and X_{t-1} is in the system for a longer time compared to X_t since it has been injected in the system during round $t-1$. Therefore, the short intermediate flows $S_{t-1,1}, \ldots, S_{t-1,t-2}$ are enough to keep all the packets of the investing flows $X_1, \ldots, X_t$ in the system till the end of round t. ■

- **Round l:** It lasts $|T_l| = r|T_{l-1}|$ time steps.

  Adversary's behavior. During this round, the adversary injects an investing flow X_l of $|X_l| = r|T_l|$ packets in queue f_3 wanting to traverse the edges $g_1, h_l, e_1, f_1^{'}, f_2^{'}, g_1^{'}$.

  Evolution of the system configuration. X_{l-1} packets (that have been injected in the system at round $l-1$) have priority over X_l packets in queue g_1. So, X_l packets remain in queue g_1 at the end of this round. If we follow a similar analysis as in the intermediate round t, we can prove that the short intermediate flows $S_{l-1,1}, \ldots, S_{l-1,l-2}$ have priority over $X_1, \ldots, X_{l-1}$ packets in the system. Therefore, at the end of round l, the number of packets that are queued in $g_1, h_1, \ldots,\ h_{l-2} = h_{k-1}$ requiring to traverse the edges $e_1, f_1^{'}, f_2^{'}, g_1^{'}$ is $s_{j+1} = |X_1| + \ldots + |X_l|$.

  In order to have instability, we must have $s_{j+1} > s_j$. This holds for $r^{k+2} - 2r + 1 < 0$. This argument can be repeated for an infinite number of phases ensuring the instability of the system $\langle \mathcal{N}_k, \mathcal{A}, \mathsf{LIS} - \mathsf{FTG} \rangle$. Also, $k \rightarrow \infty \Longrightarrow r^{k+2} \rightarrow 0$, because $0 < r < 1$. Thus, for instability it suffices $-2r + 1 < 0$, i.e. $r > 0.5$. ■

Notice that our method converges very fast to 0.5 for small values of the parameter k that depends on the network size. This can be shown easily if in the inequality $r^{k+2} - 2r + 1 < 0$ the parameters r, k are replaced by appropriate values. Therefore, for $k = 7$ the instability bound on injection rate is 0.501 and the number of network queues is 36 in the case of LIS-SIS and LIS-NTS (given by $8 + 4k$), while it is 102 in the case of LIS-FTG (given by $4 + 14k$).

4 Structural Conditions for FIFO Stability

In this section we show upper bounds on injection rate that guarantee stability for FIFO. We denote by old packet, any packet that was injected in previous time periods than the current one. The earliest time step in a time period, at which all the old packets in the system have been served is denoted by M. Also, we consider the disjoint paths $\Pi_1, \Pi_2, \ldots, \Pi_{j(\mathcal{G})}$. The number of packets in the path Π_j $(1 \leq j \leq j(\mathcal{G}))$ at time step M will be denoted by $s(\Pi_j)$. Furthermore, we denote the queues of $\mathcal{G}$ by $Q_1, Q_2, \ldots, Q_m$ and their loads at time $t \geq 0$ by $q_1(t), q_2(t), \ldots, q_m(t)$. We show:

Theorem 4.1 *Let $r_{\mathcal{G}}$ be a real number in $(0,1)$ satisfying the equation $r_{\mathcal{G}}^2 \Sigma_{i=0}^{d(\mathcal{G})-1}(\alpha(\mathcal{G})+r_{\mathcal{G}})^i = \frac{1}{j(\mathcal{G})}$. Then for any network $\mathcal{G}$, and any adversary with $r \leq r_{\mathcal{G}}$ the system $\langle \mathcal{G}, \mathcal{A}, \mathsf{FIFO} \rangle$ is stable.*

Proof: Let $P(0) = \Sigma q_i(0)$ be the initial load. We will construct an infinite sequence of consecutive distinguished 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.

Consider now 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 network. Rename the queues in this simple path as $Q_j \equiv Q_{j_0}, \ldots = Q_{j_{d(\mathcal{G})-1}}$. 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 $(\alpha(\mathcal{G})+r)M_1$. We repeat the same procedure, each time considering the last queue in the path that still contains old packets. After $d(\mathcal{G})-2$ additional steps $(M_2, M_3, \ldots, M_{d(\mathcal{G})-1})$ all the old packets would disappear or being in $Q_{j_{d(\mathcal{G})}}$. Define $P(t) = \max_{i=0}^{m}\{q_i(t)\}$. Working in the previous way, an absolute bound for the delay of the last old packet in Q_j is $M = M_1 + \ldots + M_{d(\mathcal{G})-1}$, where for every $0 < i < d(\mathcal{G})$, 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 $P(t+P(t)) \leq (\alpha(\mathcal{G})+r)P(t)$. Solving the recurrence, the total time is

$$M \leq \Sigma_{i=0}^{d(\mathcal{G})-1}(\alpha(\mathcal{G})+r)^i P(0)$$

At time step M all the old packets have been absorbed and only the injected packets in the time period $[0 \ldots M]$ will remain in the system. Because $j(\mathcal{G})$ is the minimum number of edge-disjoint paths in the network, during this period in the worst case at most $j(\mathcal{G})rM$ packets will be injected in the network. Therefore the total number of packets in the network at time step M is at most $P(M) \leq j(\mathcal{G})rM$. At time step M, $s(\Pi_j)$ packets exist in each disjoint path Π_j from the definitions. Note that the minimum number of packets in a disjoint path Π_j at time step M $(min\{s(\Pi_j)\})$ is significantly bigger comparing to the number of network edges. This allows us to assume that when a disjoint path Π_j has $s(\Pi_j)$ packets, then in each time step of a time period of $s(\Pi_j)$ time steps, r packets arrive into the path and one packet leaves it.

Assume now $s = \min\{s(\Pi_j)\}$. The change of the number of packets in the disjoint path Π_j in absolute values, Δ_{Π_j}, at $M+s$ time step will be

$$\Delta_{\Pi_j} = \Sigma_0^{\min\{s(\Pi_j)\}}|r-1| = |r-1|\min\{s(\Pi_j)\} \le |r-1|s(\Pi_j)$$

Thus, the total change of the system configuration will be

$$\Sigma_{\Pi_j}\Delta_{\Pi_j} \le \Sigma_{\Pi_j}|r-1|s(\Pi_j) = |r-1|\Sigma_{\Pi_j}s(\Pi_j)$$

But, $P(M) = \Sigma_{\Pi_j}s(\Pi_j)$. Thus,

$$\Sigma_{\Pi_j}\Delta_{\Pi_j} \le |r-1|P(M)$$

is at most the change of the system configuration for a time period with $s = min\{s(\Pi_j)\}$ steps.

Consider now, the consecutive time intervals with duration: $s, rs, r^2s, \ldots, r^ks$, where k is such that $r^ks \ge 1$ and $r^{k+1}s < 1$. The same argument as in the case of s time steps can be used for r^is 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^ks finishes. The packets in network $\mathcal{G}$ at time t_1 are all new. Thus, the number of packets in the system at time t_1 is at most

$$P(t_1) \le P(M) + (r-1)P(M) + r(r-1)P(M) + \ldots + r^{k-1}(r-1)P(M) = r^kP(M)$$

For stability, we need $P(t_1) \le P(0)$. Thus, we must choose an r such that $r^kP(M) \le P(0)$. But,

$$P(M) \le j(\mathcal{G})rM \le j(\mathcal{G})r\Sigma_{i=0}^{d(\mathcal{G})-1}(\alpha(\mathcal{G})+r)^iP(0)$$

Thus,

$$r^kj(\mathcal{G})r\Sigma_{i=0}^{d(\mathcal{G})-1}(\alpha(\mathcal{G})+r)^iP(0) \le P(0)$$

For $k=1$ this equation takes its smallest value

$$r^2\Sigma_{i=0}^{d(\mathcal{G})-1}(\alpha(\mathcal{G})+r)^i \le \frac{1}{j(\mathcal{G})}$$

This is equivalent to find in the real interval $(0,1)$, the root $r_{\mathcal{G}}$ of the polynomial

$$-r^2j(\mathcal{G})(\alpha(\mathcal{G})+r)^{d(\mathcal{G})} + r^2j(\mathcal{G}) + \alpha(\mathcal{G}) + r - 1$$

By the Bolzano Theorem, this polynomial has a root $r_{\mathcal{G}}$ in $(0,1)$. ∎

From Theorem 4.1 and the upper bound on injection rate that guarantees stability for FIFO, which has been estimated in [12] we have:

Corollary 4.2 *Let* $r^* = \max\{r_{\mathcal{G}}, \frac{1}{d(\mathcal{G})}\}$. *Then for every* $\mathcal{G}$*, and any adversary with* $r \leq r^*$ *the system* $\langle \mathcal{G}, \mathcal{A}, \mathsf{FIFO} \rangle$ *is stable.*

To illustrate the strength and applicability of our analytical techniques towards the upper bound on injection rate that guarantees FIFO stability which is given in [12], we apply them to network $\mathcal{U}_1$ (Figure 1). The upper bound for FIFO stability is 1/3 in [12], while in our case is 0.3371.

Now we show that our stability bound $r_{\mathcal{G}}$ is larger than the stability bound $r'_{\mathcal{G}}$ of Proposition 2.1:

Let

$$f(r) = \Sigma_{i=0}^{d(\mathcal{G})-1} (\alpha(\mathcal{G}) + r)^i$$

Note that $f(r)$ is monotone and increasing for $r > 0$. But,

$$\frac{2 - r'_{\mathcal{G}}}{1 - r'_{\mathcal{G}}} r'_{\mathcal{G}} m(\mathcal{G}) f(r'_{\mathcal{G}}) = 1$$

Also, $r'_{\mathcal{G}} f(r'_{\mathcal{G}}) \leq \frac{1}{2m(\mathcal{G})}$ because for all $r \in (0, 1)$ it holds that $\frac{1-r}{2-r} \leq \frac{1}{2}$.

Since $r_{\mathcal{G}} < 1$, it holds

$$r_{\mathcal{G}} f(r_{\mathcal{G}}) > r_{\mathcal{G}}^2 f(r_{\mathcal{G}}) = \frac{1}{j(\mathcal{G})}$$

From this equation along with $\frac{1}{j(\mathcal{G})} \geq \frac{1}{m(\mathcal{G})}$ and $r'_{\mathcal{G}} f(r'_{\mathcal{G}}) \leq \frac{1}{2m(\mathcal{G})}$, we take

$$r_{\mathcal{G}}^2 f(r_{\mathcal{G}}) > r'_{\mathcal{G}} f(r'_{\mathcal{G}})$$

This holds for $r_{\mathcal{G}}^2 > r'_{\mathcal{G}}$. Therefore, we have shown that in all networks $\mathcal{G}$, we have $\sqrt{r'_{\mathcal{G}}} < r_{\mathcal{G}}$ which implies $r_{\mathcal{G}} > r'_{\mathcal{G}}$ since $r'_{\mathcal{G}} \in (0, 1)$.

In order to illustrate with an example, that the technique we present in Theorem 4.1 gives better upper bounds for FIFO stability comparing to the technique proposed by Diaz *et al.* in [7] (Proposition 2.1), we apply them to network $\mathcal{U}_1$ (Figure 1). Then, the upper bound for FIFO stability is 0.0231 for [7], while in our case is 0.3371.

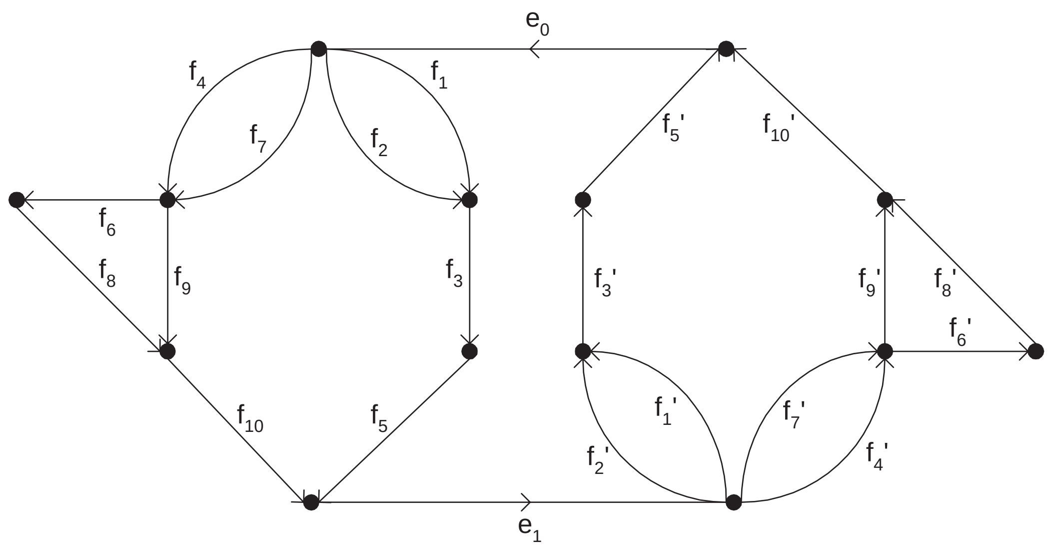

Figure 6: The Network $\mathcal{N}$

5 Instability of Small-Size FIFO Networks

In this section we present a lower bound on injection rate that guarantees instability for a FIFO network with only 22 queues. We show:

Theorem 5.1 *Let $r \geq 0.704$. There is a network $\mathcal{N}$ of 22 queues and an adversary $\mathcal{A}$ of rate r such that the system $\langle \mathcal{N}, \mathcal{A}, \mathsf{FIFO} \rangle$ is unstable.*

Proof: We consider the network $\mathcal{N}$ in Figure 6. We start with an informal description of our proof.

(i) We split the time into phases. In each phase we consider the evolution of the system configuration as a sequence of consecutive distinguished time rounds. Then, we inductively show that the number of packets in the system increases at the end of a phase comparing to the beginning. This inductive argument can be applied repeatedly, thus showing instability for an infinite time interval.

(ii) We use an inductive hypothesis with two parts. The first part specifies the position of the initial packets at the beginning of a phase and that their number is smaller than the number of packets in the corresponding subset of queues that will serve as initial packets at the beginning of the next phase. The second part guarantees that the initial packets in each phase will traverse their path as a continuous flow.

(iii) We achieve further delay of packets initially residing in the system by exploiting multiple parallel paths of the network topology between a common origin and destination.

(iv) We heavily exploit the *fair mixing* property of FIFO.

We now continue with the detailed proof.

Inductive Hypothesis: At the beginning of phase j, there are s_j packets that are queued in the queues $e_1, f_3', f_4', f_5', f_6', f_8'$ requiring to traverse the edges e_0, f_1, f_3, f_5, all these packets are able to depart from their initial edges to the symmetric part of the network (f_1, f_3, f_5) as a continuous flow in s_j time steps, the number of packets that are queued in queues f_4', f_6', f_8', e_0 is larger than the number of packets that are queued in queues f_3', f_5' and the number of packets that are queued in queue e_1 is less than the number of packets that are queued in queues f_4', f_6', f_8'.

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 $f_3, f_5, f_4, f_6, f_8, e_0$ requiring to traverse the edges e_1, f_1', f_3', f_5', all of which will be able to depart from their initial edges to the symmetric part of the network (f_1', f_3', f_5') in s_{j+1} time steps as a continuous flow, the number of packets that will be queued in queues f_4, f_6, f_8, e_0 will be larger than the number of packets that will be queued in queues f_3, f_5 and the number of packets that will be queued in queue e_0 will be less than the number of packets that will be queued in queues f_4, f_6, f_8.

Notice that our inductive argument claims that if at the beginning of phase j all s_j packets that are queued in queues $e_1, f_3', f_4', f_5', f_6', f_8'$ requiring to traverse the edges e_0, f_1, f_3, f_5, manage to traverse their initial edges in s_j time steps as a continuous flow, then at the beginning of phase $j+1$ all s_{j+1} packets, that will be queued in queues $e_0, f_3, f_5, f_4, f_6, f_8$ requiring to traverse the edges e_1, f_1', f_3', f_5', will be able to traverse their initial edges in s_{j+1} time steps as a continuous flow. This argument guarantees the reproduction of the inductive hypothesis in queues e_0, f_3, f_5, f_4, f_6 even if there are flows (in particular in queues f_3, f_4, f_5, f_6, f_8) that do not want to traverse the edges e_1, f_1', f_3', f_5' the packets of which are regularly spread among the packets that want to traverse these edges.

Furthermore, this argument implies the third part of the inductive argument, which claims that if at the beginning of phase j, the number of packets that are queued in queues e_1, f_4', f_6', f_8' is larger than the number of packets that are queued in queues f_3', f_5', then at the beginning of phase $j+1$ the number of packets that will be queued in queues e_0, f_4, f_6, f_8 will be larger than the number of packets that will be queued in queues f_3, f_5. This happens because in the first round of the adversary's construction we inject packets in queue $f_4^{'}$ and if the third part of the inductive hypothesis doesn't hold then we cannot guarantee that all the initial s_j packets will depart their initial edges to the edges f_1, f_3, f_5 in s_j time steps as a continuous flow. However, we include it into the inductive hypothesis for readability reasons.

We will construct an adversary $\mathcal{A}$ 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 $j+1$ 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 $e_1, f_3^{'}, f_4^{'}, f_5^{'}, f_6^{'}, f_8^{'}$ requiring to traverse the edges e_0, f_1, f_3, f_5. 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 $|T_1| = s_j$ time steps.

  Adversary's behavior. During this round, the adversary injects a set X of $|X| = r|T_1|$ packets in queue $f_4^{'}$ wanting to traverse the edges $f_4^{'}, f_6^{'}, f_8^{'}, f_{10}^{'}, e_0, f_2, f_3, f_5, e_1, f_1^{'}, f_3^{'}, f_5^{'}$. Also, it injects a set S_1 of $|S_1| = r|T_1|$ packets in queue f_1 wanting to traverse the edge f_1.

  Evolution of the system configuration. The packets of set S delay X packets in queue e_0 because the S packets in queues $e_1, f_4^{'}, f_6^{'}, f_8^{'}$ are more than the S packets in queues $f_3^{'}, f_5^{'}$ and the packets in e_1 are less than the packets in queues $f_4^{'}, f_6^{'}, f_8^{'}$ at the beginning of this round. Thus, all the S packets will traverse their initial edges in s_j time steps stopping the packets of set X in queue e_0.

  At the same time, the packets of set S are delayed in queue f_1 where they get mixed with S_1 packets. Notice that due to FIFO, the packets of sets S, S_1 mix in consecutive blocks according to their initial proportion of their sizes (fair mixing property). Since $|S| = |T_1|$ and $|S_1| = r|T_1|$, these proportions are $\frac{|S|}{|S_1|+|S|}$ and $\frac{|S_1|}{|S_1|+|S|}$, respectively. Thus, during the s_j steps of this round, the packets of sets S, S_1, which cross f_1 are, respectively, $\frac{|S|}{|S_1|+|S|}|T_1|$ and $\frac{|S_1|}{|S_1|+|S|}|T_1|$. Therefore, the remaining packets in queue f_1 are:

  - for packet set S: a set S_{rem} of $|S_{rem}| = |T_1| - \frac{|S|}{|S_1|+|S|}|T_1|$ packets,
  - for packet set S_1: a set $S_{1,rem}$ of $|S_{1,rem}| = |S_1| - \frac{|S_1|}{|S_1|+|S|}|T_1|$ packets.

- **Round 2:** It lasts $|T_2| = r|T_1|$ time steps.

  Adversary's behavior. The adversary injects a set Y of $|Y| = r|T_2|$ packets in queue f_4' requiring to traverse the edges $f_4', f_6', f_8', f_{10}', e_0, f_4, f_6, f_8, f_{10}, e_1, f_1', f_3', f_5'$. At the same time, the adversary injects a set S_2 of $|S_2| = r|T_2|$ packets in queue f_2 wanting to traverse the edge f_2, a set S_3 of $|S_3| = r|T_2|$ packets in queue f_3 wanting to traverse the edge f_3, and a set S_4 of $|S_4| = r|T_2|$ packets in queue f_5 wanting to traverse the edge f_5.

  Evolution of the system configuration. The packets of set Y are blocked by the set X in queue e_0. At the same time, the packet sets X, S_2 mix in consecutive blocks according to their initial proportion of their sizes due to FIFO. Since $|X| = r|T_1|$ and $|S_2| = r|T_2|$, these proportions are $\frac{|X|}{|X|+|S_2|}$ and $\frac{|S_2|}{|X|+|S_2|}$, respectively. Thus, during the $|T_2|$ steps of this round, the packets of sets X, S_2, that traverse the edge f_2 are:

  - for set X: a set X_{pass,f_2} of $|X_{pass,f_2}| = \frac{|X|}{|X|+|S_2|}|T_2|$ packets,
  - for set S_2: a set $S_{2,pass,f_2}$ of $|S_{2,pass,f_2}| = \frac{|S_2|}{|X|+|S_2|}|T_2|$ packets. These packets are absorbed.

  On the other hand, the remaining packets in queue f_2 are

  - for set X: a set X_{rem,f_2} of $|X_{rem,f_2}| = |X| - |X_{pass,f_2}|$ packets,
  - for set S_2: a set S_{2,rem,f_2} of $|S_{2,rem,f_2}| = |S_2| - |S_{2,pass,f_2}|$ packets.

  Notice that in queue f_1 at the beginning of this round, there are the $S_{rem}, S_{1,rem}$ packets that have remained there from the previous round. Since their total number is $|T_2|$, which is equal to the duration of this round, the packets of $S_{1,rem}$ do not delay the packets of S_{rem}. In addition, the $S_{1,rem}$ packets are absorbed after they traverse the edge f_1, therefore only the S_{rem} packets require the edge f_3. As a result the packet stream arriving from f_1 to f_3 does not contain packets at the positions of the $S_{1,rem}$ packets. Thus, we can consider it as a stream with empty spaces at the positions of the $S_{1,rem}$ packets. However, these empty spaces are uniformly spread for the duration of the time period. Thus, during this round, three different packet sets arrive at queue f_3:

  - X_{pass,f_2} where $|X_{pass,f_2}| = \frac{|X|}{|X|+|S_2|}|T_2|$ packets. These packets mixed with packets of set $S_{2,pass,f_2}$. But, since their total number is $|T_2|$, $S_{2,pass,f_2}$ packets do not delay X_{pass,f_2} packets. Also the $S_{2,pass,f_2}$ packets are absorbed after they traverse the edge f_2. Thus, only the X_{pass,f_2} packets require to traverse the edge f_3. As a result the stream arriving from f_2 to f_3 does not contain packets at the positions of the $S_{2,pass,f_2}$ packets.

- S_{rem}, where $|S_{rem}| = |S| - \frac{|S|}{|S_1|+|S|}|T_1|$,
- S_3, where $|S_3| = r|T_2|$.

Since the total number of packets of the three flows arriving to queue f_3 is

$$|T| = |X_{pass,f_2}| + |S_3| + |S_{rem}|$$

the corresponding proportions are:

- for X_{pass,f_2}: $\frac{|X_{pass,f_2}|}{T}$,
- for S_{rem}: $\frac{|S_{rem}|}{T}$,
- for S_3: $\frac{|S_3|}{T}$.

Thus, the remaining packets in queue f_3 from each flow at the end of this round are:

- for X_{pass,f_2}: a set X_{rem,f_3} of $|X_{rem,f_3}| = |X_{pass,f_2}| - \frac{|X_{pass,f_2}|}{|T|}|T_2|$ packets,
- for S_{rem}: a set S_{rem,f_3} of $|S_{rem,f_3}| = |S_{rem}| - \frac{|S_{rem}|}{|T|}|T_2|$ packets,
- for S_3: a set $S_{3,rem}$ of $|S_{3,rem}| = |S_3| - \frac{|S_3|}{|T|}|T_2|$ packets.

The technique of proportions can still be used even if some stream of packets has empty spaces, since the empty spaces are uniformly spread. Notice that during this round the stream arriving to the edge f_5 contains three different packet sets:

- the set S_4, where $|S_4| = r|T_2|$,
- the set S_{pass,f_3} of S_{rem} packets that traverse the edge f_3, where $|S_{pass,f_3}| = \frac{|S_{rem}|}{|T|}|T_2|$,
- the set X_{pass,f_3} of X_{pass,f_2} packets that traverse the edge f_3, where $|X_{pass,f_3}| = \frac{|X_{pass,f_2}|}{|T|}|T_2|$.

Notice also that the S_3 packets that traverse the edge f_3 are absorbed after they traverse edge f_3. Since the total number of packets in the three flows is $|T^{'}| = |S_4| + |S_{pass,f_3}| + |X_{pass,f_3}|$ the corresponding proportions are:

- for X_{pass,f_3}: $\frac{|X_{pass,f_3}|}{|T^{'}|}$,
- for S_{pass,f_3}: $\frac{|S_{pass,f_3}|}{|T^{'}|}$,
- for S_4: $\frac{|S_4|}{|T^{'}|}$.

Thus, the remaining packets from each flow in queue f_5 at the end of this round are:

- for X_{pass,f_3}: $|X_{pass,f_3}| - \frac{|X_{pass,f_3}|}{|T'|}|T_2|$,
- for S_{pass,f_3}: $|S_{pass,f_3}| - \frac{|S_{pass,f_3}|}{|T'|}|T_2|$
- for S_4: $|S_4| - \frac{|S_4|}{|T'|}|T_2|$.

- **Round 3:** It lasts $|T_3| = r|T_2|$ time steps.

  Adversary's behavior. During this round the adversary injects a set S_5 of $|S_5| = r|T_3|$ packets in queue f_4 requiring to traverse only f_4, a set S_6 of $|S_6| = r|T_3|$ packets in queue f_6 requiring to traverse only f_6, a set S_7 of $|S_7| = r|T_3|$ packets in queue f_8 requiring to traverse only f_8 and a set Z of $|Z| = r|T_3|$ packets in queue e_0 requiring to traverse the edges $e_0, f_7, f_9, f_{10}, e_1, f_1', f_3', f_5'$.

  Evolution of the system configuration. Y packets have priority over Z packets in queue e_0. At the same time the packet sets S_5, Y get mixed in queue f_4 in consecutive blocks according to their initial proportion of their sizes. These proportions are

  - for set Y: $\frac{|Y|}{|Y|+|S_5|}$,
  - for set S_5: $\frac{|S_5|}{|Y|+|S_5|}$.

  Thus, during the $|T_3|$ steps of this round, the packets of packet sets Y, S_5 that remain in queue f_4 are respectively:

  - for Y: a set Y_{rem} of $|Y_{rem}| = |Y| - \frac{|Y|}{|Y|+|S_5|}|T_3|$ packets,
  - for S_5: a set $S_{5,rem}$ of $|S_{5,rem}| = |S_5| - \frac{|S_5|}{|Y|+|S_5|}|T_3|$ packets.

  On the other hand the packets of packet sets Y, S_5 that traverse the edge f_4 are respectively:

  - for Y: a set Y_{pass} of $|Y_{pass}| = \frac{|Y|}{|Y|+|S_5|}|T_3|$ packets,
  - for S_5: a set $S_{5,pass}$ of $|S_{5,pass}| = \frac{|S_5|}{|Y|+|S_5|}|T_3|$ packets. The $S_{5,pass}$ packets are absorbed after they traverse the edge f_4 because they are single-edge injections.

  Furthermore, during this round the set of packets S_6 get mixed with the set Y_{pass} in consecutive blocks according to their initial proportion of their sizes. These proportions are:

  - for Y_{pass}: $\frac{|Y_{pass}|}{|S_6|+|Y_{pass}|}$
  - for S_6: $\frac{|S_6|}{|S_6|+|Y_{pass}|}$.

Thus, during the $|T_3|$ steps of this round, the packets that traverse the edge f_6 are respectively:

- for Y_{pass}: a set Y_{pass,f_6} of $|Y_{pass,f_6}| = \frac{|Y_{pass}|}{|S_6|+|Y_{pass}|}|T_3|$ packets,
- for S_6: a set $S_{6,pass}$ of $|S_{6,pass}| = \frac{|S_6|}{|S_6|+|Y_{pass}|}|T_3|$ packets. The $S_{6,pass}$ packets are absorbed after they traverse the edge f_6 because they are single-edge injections.

Therefore, the remaining packets in queue f_6 are:

- for Y_{pass}: a set Y_{rem,f_6} of $|Y_{rem,f_6}| = |Y_{pass}| - |Y_{pass,f_6}|$ packets,
- for S_6: a set $S_{6,rem}$ of $|S_{6,rem}| = |S_6| - |S_{6,pass}|$ packets.

Additionally, during this round the set of packets S_7 get mixed in queue f_8 with the set Y_{pass,f_6} in consecutive blocks according to their initial proportion of their sizes. These proportions are:

- for Y_{pass,f_6}: $\frac{|Y_{pass,f_6}|}{|S_7|+|Y_{pass,f_6}|}$
- for S_7: $\frac{|S_7|}{|S_7|+|Y_{pass,f_6}|}$.

Thus, during the $|T_3|$ steps of this round, the packets that traverse the edge f_8 are respectively:

- for Y_{pass}: a set Y_{pass,f_8} of $|Y_{pass,f_8}| = \frac{|Y_{pass,f_6}|}{|S_7|+|Y_{pass,f_6}|}|T_3|$ packets,
- for S_7: a set $S_{7,pass}$ of $|S_{7,pass}| = \frac{|S_7|}{|S_7|+|Y_{pass,f_6}|}|T_3|$ packets. The $S_{7,pass}$ packets are absorbed after they traverse the edge f_8 because they are single-edge injections.

Therefore, the remaining packets in queue f_8 are:

- for Y_{pass}: a set Y_{rem,f_8} of $|Y_{rem,f_8}| = |Y_{pass,f_6}| - |Y_{pass,f_8}|$ packets,
- for S_6: a set $S_{7,rem}$ of $|S_{7,rem}| = |S_7| - |S_{7,pass}|$ packets.

Notice that the total number of packets that are queued in queue f_5 at the end of round 2 is equal to

$$|P_1| = |X_{pass,f_3}| + |S_{pass,f_3}| + |S_4|$$

However, $|P_1| < |T_3|$, for all $r > 0$. Thus, all the packets in queue f_5 at the end of the previous round will traverse the edge f_5. Therefore, the remaining time during which packets arriving to edge f_5 from edge f_3 can traverse the edge f_5 is $t_{rem} = |T_3| - |P_1|$.

Notice also that the total number of packets that are queued in queue f_3 at the beginning of this round is

$$|P_2| = |X_{rem,f_3}| + |S_{rem,f_3}| + |S_{3,rem}|$$

However, $|P_2| \geq |T_3|$, for all r. Thus, a number of $X_{rem,f_3}, S_{rem,f_3}, S_{3,rem}$ packets can remain in queue f_3 at the end of this round. This number is $|P_3| = |P_2| - |T_3|$. From this number, the number of packets that belong to set $S_{3,rem}$ is

$$|S_{3,rem,f_3}| = |P_3| \frac{|S_{3,rem}|}{|S_{3,rem}| + |X_{rem,f_3}| + |S_{rem,f_3}|}$$

Except the remaining packets in queue f_3, there is a number of packets of sets X_{rem,f_3}, S_{rem,f_3} and $S_{3,rem}$ that traverse the edge f_3 during this round. This number of packets is $|T_3|$. From these packets, the packets that belong to $S_{3,rem}$ are absorbed because they are single-edge injections. The number of these packets is $\frac{|S_{3,rem}|}{|S_{3,rem}|+|X_{rem,f_3}|+|S_{rem,f_3}|}|T_3|$.

In $|T_3|$ steps, the same portion of X_{rem,f_3} and S_{rem,f_3} packets traverse the edge f_3, $X_{rem,f_3,pass}$ and $S_{rem,f_3,pass}$ correspondingly, because they get mixed in queue f_3 with the same proportion. The size of the portion of X_{rem,f_3} and S_{rem,f_3} packets that traverse edge f_3 is

$$|X_{pass,f_3,pass}| = |S_{rem,f_3,pass}| = \frac{|X_{rem,f_3}|}{|S_{3,rem}| + |X_{rem,f_3}| + |S_{rem,f_3}|}|T_3|$$

Consequently, the sum of these packets is $|P_4| = |X_{rem,f_3,pass}| + |S_{rem,f_3,pass}|$. However, $|P_4| \geq t_{rem}$ for all r. Therefore, during the remaining t_{rem} time steps of this round a number of $X_{pass,f_3,pass}$ and $S_{rem,f_3,pass}$ packets, which arrive to the edge f_5 from the edge f_3, will traverse the edge f_5. $X_{pass,f_3,pass}$ and $S_{rem,f_3,pass}$ packets arrive to the edge f_5 from the edge f_3 with the same proportion and their sum P_4 is greater or equal to the remaining time steps t_{rem}. Thus, the same number of $X_{pass,f_3,pass}$ and $S_{rem,f_3,pass}$ packets will traverse the edge f_5 (X_{pass,f_3,f_5} and S_{absorb,f_3,f_5} correspondingly). This number is $|X_{pass,f_3,f_5}| = |S_{absorb,f_3,f_5}| = \frac{t_{rem}}{2}$. Therefore, the number of packets from the packet set X_{rem,f_3} that will not arrive at queue e_1 is $|X_{f_3,f_5}| = |X_{rem,f_3}| - |X_{pass,f_3,f_5}|$.

In addition, the total number of packets that are in queue f_2 at the end of round 2 is $|P_5| = |T_3|$. Thus, all the X packets traverse the edge f_2 and arrive to the edge f_3 where they are blocked due to the packets that are already queued in f_3 at the end of the previous round. This happens because the number of packets that are queued in f_3 at the end of the previous round is greater than or equal to the time duration of the current round as we have shown previously.

The total number of packets that arrive at queue e_1 during this round is:

$$|P_{e_1}| = |X_{pass,f_3,f_5}| + |Y_{pass,f_8}|$$

For $r \geq 0.276$, $|P_{e_1}| \leq |T_3|$. Therefore for $r \geq 0.276$ the number of packets that arrive at queue e_1 is less or equal to the number of time steps of this round.

In order to have instability the number of packets that are queued in queues $f_3, f_4, f_5, f_6, f_8, e_0, e_1$ requiring to traverse the edges e_1, f_1', f_3', f_5' at the end of round 3, s_{j+1}, should be more than the initial s_j packets that were queued in the system at corresponding queues at the beginning of round 1. This holds for

$$|Z| + |Y_{rem}| + |Y_{rem,f_6}| + |Y_{rem,f_8}| + |X_{rem,f_2}| + |X_{f_3,f_5}| > s_j$$

This inequality implies that $r \geq 0.704$. The first part of the proof is now complete.

In order to conclude the proof we should also consider the following technical lemma:

Lemma 5.2 *For $r \geq 0.609$, the number of time steps that are needed for the arrival at queue e_1 of all the X packets that remain in queues f_3, f_5 at the end of round 3 is less than or equal to the number of time steps that are needed for the arrival at queue e_1 of all the Y, Z packets that remain in queues f_4, f_6, f_8, e_0 at the end of round 3.*

Proof: The required number of time steps for the arrival at queue e_1 of all the X packets that remain in queues f_3, f_5 at the end of round 3 is

$P_{X,e_1} = |X_{rem,f_3}| - |X_{pass,f_3,f_5}| + |S_{rem,f_3}| - |S_{absorb,f_3,f_5}| + |X_{rem,f_2}| + |S_{3,rem,f_3}|$.

where the quantities $|X_{rem,f_3}|, |X_{pass,f_3,f_5}|, |S_{rem,f_3}|, |S_{absorb,f_3,f_5}|, |X_{rem,f_2}|, |S_{3,rem,f_3}|$ have been estimated at round 3 of the adversarial construction.

On the other hand the number of packets remaining in queue f_4 at the end of round 3 is $Q(f_4) = |Y_{rem}| + |S_{5,rem}|$. The number of packets remaining in queue f_6 at the end of round 3 is $Q(f_6) = |Y_{rem,f_6}| + |S_{6,rem}|$ and the number of packets remaining in queue f_8 at the end of round 3 is $Q(f_8) = |Y_{rem,f_8}| + |S_{7,rem}|$. Also, the number of Z packets that remain in queue e_0 is $Q(e_0) = r|T_3|$.

In $r|T_3|$ time steps all the Z packets will arrive at queue f_{10}. Also in $r|T_3|$ time steps, all the Y_{rem,f_6}, Y_{rem,f_8} packets will arrive at queue f_{10} as $Q(f_6) + Q(f_8) < r|T_3|$ and all the Y_{rem} packets will depart queue f_4 towards queue f_{10}. Therefore, the required number of time steps for the arrival at queue e_1 of all the Y, Z packets that remain in queues

f_4, f_6, f_8, e_0 at the end of round 3 is $P_{Y,Z,e_1} = |Z| + |Y_{rem}| + |Y_{rem,f_6}| + |Y_{rem,f_8}|$ where the quantities $|Z|, |Y_{rem}|, |Y_{rem,f_6}|, |Y_{rem,f_8}|$ have been estimated at round 3 of the adversarial construction. For $r \geq 0.609$, the inequality $P_{Y,Z,e_1} \geq P_{X,e_1}$ holds. If this constraint holds, the third part of the inductive hypothesis of the theorem holds too. ∎

Notice that we have, till now, managed to reproduce the inductive hypothesis in queues $f_3, f_5, f_4, f_6, f_8, e_0, e_1$ but with some packet sets (in particular in queues f_4, f_6, f_8, f_3, f_5) that contain packets that do not want to traverse the edges e_1, f_1', f_3', f_5'. In order for the induction step to work we must show that all the X, Y, Z packets that remain in these queues at the end of round 3 will manage to depart to the symmetric part of the network (f_1', f_3', f_5') in s_{j+1} time steps as a continuous flow. This is shown by the following lemma.

Lemma 5.3 *All the X, Y, Z packets that remain in queues $f_3, f_5, f_4, f_6, f_8, e_0, e_1$ at the end of round 3 will manage to depart to the symmetric part of the network (f_1', f_3', f_5') in s_{j+1} time steps as a continuous flow.*

Proof: All the packets that are queued in queue e_1 at the end of round 3 want to traverse the edges e_1, f_1', f_3', f_5'. Also, from Lemma 5.2 for $r \geq 0.609$, the number of time steps that are needed for the arrival at queue e_1 of all the X packets that remain in queues f_3, f_5 at the end of round 3 is less than or equal to the number of time steps that are needed for the arrival at queue e_1 of all the Y, Z packets that remain in queues f_4, f_6, f_8, e_0 at the end of round 3.

Therefore, in order to prove this lemma, it is sufficient to show that all the Y, Z packets that remain in queues f_4, f_6, f_8, e_0 at the end of round 3 ($P_{Y,Z,e_1} = |Z| + |Y_{rem}| + |Y_{rem,f_6}| + |Y_{rem,f_8}|$) arrive at queue e_1 in P_{Y,Z,e_1} time steps as a continuous flow. But, this holds from Lemma 5.2. ∎

We have so far established two (non-trivial) sufficient constraints on r for instability, namely that $r \geq 0.704$ and $r \geq 0.609$. Clearly, taking $r \geq \max\{0.704, 0.609\} = 0.704$ suffices for instability of the network $\mathcal{N}$ in the constructed execution. This concludes our proof. ∎

6 Unstable Subgraphs

In this section we show lower bounds on injection rate that guarantee instability for forbidden subgraphs. Consider the networks $\mathcal{U}_2$ and $\mathcal{U}_3$ (see Figures 2, 7) that use NTG-U-LIS protocol.

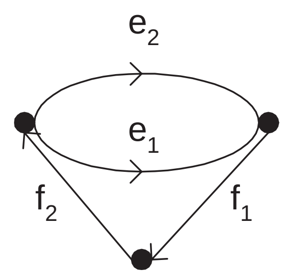

Figure 7: Network $\mathcal{U}_3$

Theorem 6.1 *Let $r \geq 0.794$. There is a network $\mathcal{U}_i$ and an adversary $\mathcal{A}$ of rate r such that the system $\langle \mathcal{U}_i, \mathcal{A}, \mathsf{NTG-U-LIS} \rangle$ is unstable where $\mathcal{U}_i$ is the network a) $\mathcal{U}_2$, b) $\mathcal{U}_3$.*

Proof: **Part a)** Consider the network $\mathcal{U}_2$ in Figure 2.

Inductive hypothesis: At the beginning of phase j, there are s_j packets (called S set of packets) in the queues e_1, e_2 requiring to traverse the 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_j packets, $s_{j+1} > s_j$, in the queues e_1, e_2 requiring to traverse the edges e_1, f_2 and e_2, f_1, f_2 correspondingly.

We will construct an adversary $\mathcal{A}$ such that the induction step will hold. Proving that the induction step holds, we ensure that the induction hypothesis will hold at the beginning of phase $j+1$ with an increased value of s_j, $s_{j+1} > s_j$. From the induction hypothesis, initially, there are s_j packets (called S set of packets) in the queues e_1, e_2 requiring to traverse the edges e_1, f_2 and e_2, f_1, f_2 correspondingly. In order to prove that the inductive argument works, we consider that there is a large enough number of packets s_j in the initial system configuration.

During phase j the adversary plays three rounds of injections. The sequence of injections is as follows:

- **Round 1:** It lasts $|T_1| = s_j$ time steps.

  Adversary's behavior. During this round, the adversary injects in queue f_2 a set Z_1 of $|Z_1| = r|T_1|$ packets wanting to traverse the edges f_2, e_1, e_2.

  Evolution of the system configuration. S packets have priority over Z_1 packets in queue f_2 because S packets are nearest to their destination (queue f_2) than Z_1 packets (queue e_2) and f_2 uses NTG-U-LIS. Thus, S packets reach their destination where they are absorbed, while Z_1 packets are queued in queue f_2.

- **Round 2:** It lasts $|T_2| = r|T_1|$ time steps.

  Adversary's behavior. During this round, the adversary injects a set Z_2 of $|Z_2| = r|T_2|$ packets in queue e_1 requiring to traverse the edges e_1, f_2 and a set Z_3 of $|Z_3| = r|T_2|$ packets in queue e_2 requiring to traverse the edge e_2.

  Evolution of the system configuration. Z_1 packets have priority over Z_2 packets in e_1. These flows have the same number of edges to traverse to reach their destination, but Z_1 packets are longer time in the system than Z_2 packets. Therefore, due to NTG-U-LIS that uses LIS to solve such kind of contentions, all Z_1 packets arrive at queue e_2 where they get mixed with Z_3 packets. The total number of packets arriving at queue e_2 during this round is $|Z_1| + |Z_3|$ packets. However, the duration of this round is $|T_2|$ time steps. Therefore, $|T_2|$ packets traverse the 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 $|X| = r|T_2|$ remaining packets in queue e_2 wanting to traverse the edge e_2 and $|Z_2| = r|T_2|$ packets in queue e_1 wanting to traverse the edges e_1, f_2.

- **Round 3:** It lasts $|T_3| = r|T_2|$ time steps.

  Adversary's behavior. During this round, the adversary injects a set Z_4 of $|Z_4| = r|T_3|$ packets in queue e_1 requiring to traverse the edge e_1 and a set Z_5 of $|Z_5| = r|T_3|$ packets in queue e_2 requiring to traverse the edges e_2, f_1, f_2.

  Evolution of the system configuration. X packets have priority over Z_5 packets in queue e_2 because X packets have nearest to go (queue e_2) than Z_5 packets (queue f_2). Thus at the end of this round, there are $|Z_5| = r|T_3|$ packets in queue e_2 wanting to traverse the edges e_2, f_1, f_2. Also, the Z_4 packets have priority over Z_2 packets in queue e_1 because Z_4 packets have nearest to go (queue e_1) than Z_2 packets (queue f_2). But, the number of Z_4 packets is $|Z_4| = r|T_3|$, while this round has $|T_3|$ time steps. Therefore, along with the Z_4 packets and $|T_3| - r|T_3|$ packets from Z_2 will traverse the edge e_1. Thus, at the end of this round, the remaining portion of Z_2 packets in queue e_1 that want to traverse the edges e_1, f_2 is $|Y| = r|T_3|$ packets. Totally at the end of this round, the number of packets in queues e_1, e_2 requiring to traverse the edges e_1, f_2 and e_2, f_1, f_2 correspondingly is $s_{j+1} = |Y| + |Z_5| = 2r|T_3|$.

  In order to have instability, we must have $s_{j+1} > s_j$. This holds for $2r|T_3| > |T_1|$, i.e. $r \geq 0.794$. This argument can be repeated for an infinite and unbounded number of phases ensuring that the number of packets in the queues e_1, e_2 requiring to traverse the edges e_1, f_2 and e_2, f_1, f_2 at the end of a phase is larger than at the beginning of the phase forever.

Part b) Consider the network $\mathcal{U}_3$ in Figure 7.

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

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

We will construct an adversary $\mathcal{A}$ 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_j, $s_{j+1} > s_j$. From the inductive hypothesis, initially, there are s_j packets (called S set of packets) in the queues e_1, e_2 requiring to traverse the edge f_1. In order to prove that the inductive argument works, we consider that there is a large enough number of packets s_j in the initial system configuration.

During phase j the adversary plays three rounds of injections as follows:

- **Round 1:** It lasts $|T_1| = s_j$ time steps.

  Adversary's behavior. During this round, the adversary injects in queue f_1 a set Z_1 of $|Z_1| = r|T_1|$ packets wanting to traverse the edges f_1, f_2, e_1.

  Evolution of the system configuration. S packets have priority over Z_1 packets in queue f_1 because S packets are nearest to their destination (queue f_1) than Z_1 packets (queue e_1). Thus, S packets reach their destination where they are absorbed, while Z_1 packets are queued in queue f_1.

- **Round 2:** It lasts $|T_2| = r|T_1|$ time steps.

  Adversary's behavior. During this round, the adversary injects a set Z_2 of $|Z_2| = r|T_2|$ packets in queue f_2 requiring to traverse the edges f_2, e_2, f_1 and a set Z_3 of $|Z_3| = r|T_2|$ packets in queue e_1 requiring to traverse the edge e_1.

  Evolution of the system configuration. Z_1 packets have priority over Z_2 packets in queue f_2 because Z_1 packets have nearest to go (queue e_1) than Z_2 packets (queue f_1). Therefore, all Z_1 packets arrive at queue e_1 where they get mixed with Z_3 packets, while Z_2 packets are queued in queue f_2. The total number of packets arriving in queue e_1 during this round is $|Z_1| + |Z_3|$ packets. However, the duration of this round is $|T_2|$ steps. Therefore, $|T_2|$ packets will traverse the edge e_1. Thus, at the end of this round, there will be a set X of $|X| = r|T_2|$ remaining packets in queue e_1 wanting to traverse the edge e_1 and $|Z_2| = r|T_2|$ packets in queue f_2 wanting to traverse the edges f_2, e_2, f_1.

- **Round 3:** It lasts $|T_3| = r|T_2|$ time steps.

  Adversary's behavior. During this round, the adversary injects a set Z_4 of $|Z_4| = r|T_3|$ packets in queue e_2 requiring to traverse the edge e_2 and a set Z_5 of $|Z_5| = r|T_3|$ packets in queue e_1 requiring to traverse the edges e_1, f_1.

  Evolution of the system configuration. The packets of the set X have priority over Z_5 packets in e_1 because X packets have nearest to go (queue e_1) than Z_5 packets (queue f_1). Thus, at the end of this round, there are $|Z_5| = r|T_3|$ packets in queue e_2 wanting to traverse the edges e_1, f_1, while X packets reach their destination where they are absorbed. Furthermore, the Z_4 packets have priority over Z_2 packets in queue e_2 because Z_4 packets have nearest to go (queue e_2) than Z_2 packets (queue f_1). But, the number of Z_4 packets is $|Z_4| = r|T_3|$, while this round has duration $|T_3|$ time steps. Therefore, along with the Z_4 packets and $|T_3| - r|T_3|$ packets from Z_2 will traverse the edge e_2. Thus at the end of this round, the remaining portion of the Z_2 packets in queue e_2 that want to traverse the edges e_2, f_1 is $|Y| = r|T_3|$ packets. Totally at the end of this round, the number of packets in queues e_1, e_2 requiring to traverse the edge f_1 is $s_{j+1} = |Y| + |Z_5| = 2r|T_3|$.

  In order to have instability, we must have $s_{j+1} > s_j$. This holds for $2r|T_3| > |T_1|$, i.e. $r \geq 0.794$. This argument can be repeated for an infinite and unbounded number of phases ensuring that the number of packets that are queued in e_1, e_2 requiring to traverse the edge f_1 at the end of a phase is larger than at the beginning of the phase forever. ■

7 Discussion and Directions for Further Research

In this work we have studied how network structure affects the stability properties of greedy contention-resolution protocols in the framework of Adversarial Queueing Theory [4]. We have shown that the lower bound on injection rate that guarantees instability for specific compositions of universally stable protocols drops when the network size increases. In particular we demonstrate size-parameterized adversarial constructions that lead to instability certain compositions of protocols for adversary's injection rate $r \in (0.5, 1]$. We also presented a FIFO network whose lower bound on injection rate that guarantees instability can be dropped to a low value without increasing its size. This result represents the current state-of-the-art trade-off between the network size and the lower bound on injection rate that guarantees instability. Furthermore, we presented an enhanced analysis for estimating upper bounds on injection rate that guarantee stability for FIFO on arbitrary networks that is based on the correct calibration of certain graph parameters such as the maximum directed network path length, the maximum vertex degree and the minimum number of edge-disjoint paths that cover the graph. Finally, we studied the instability behavior induced by certain known forbidden subgraphs on networks running NTG-U-LIS protocol presenting adversarial constructions that improve the state-of-the-art lower bound on injection rate that guarantees instability.

In this work we have presented some examples of the impact network structure has on the stability behavior of greedy protocols and networks. However, a lot of problems remain open. An important problem is to study the impact of network structure parameters on other greedy protocols. Another problem is whether the lower bound on injection rate that guarantees instability for compositions of protocols can be dropped further increasing the network size or whether it is affected by other network parameters that have not be determined yet. Finally, an interesting problem is whether there are upper bounds on injection rate that guarantee stability for forbidden subgraphs for universal stability.

References

[1] 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.

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

[3] R. Bhattacharjiee, and A. Goel, "Instability of FIFO at Arbitrarily Low Rates in the Adversarial Queueing Model," *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science,* pp. 160–167, October 2003.

[4] 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.

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

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

[7] 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.

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

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

[10] D. Koukopoulos, M. Mavronicolas, and P. Spirakis, "FIFO is Unstable at Arbitrarily Low Rates," *Electronic Colloquium on Computational Complexity (ECCC),* ECCC Reports 2003, Technical Report TR03-016, accepted on March 24, 2003.

[11] D. Koukopoulos, S. Nikoletseas, and P. Spirakis, "Stability Issues in Heterogeneous and FIFO Networks under the Adversarial Queueing Model," Invited Keynote Address, *Proceedings of the 8th International Conference on High Performance Computing,* B. Monien, V. K. Prasanna and S. Vajapeyam eds., pp. 3–14, Vol. 2228, Lecture Notes in Computer Science, Springer-Verlag, Hyderabad, India, December 2001.

[12] Z. Lotker, B. Patt-Shamir and A. Rosen, "New Stability Results for Adversarial Queuing," *Proceedings of the 14th Annual ACM Symposium on Parallel Algorithms and Architectures,* pp. 192–199, August 2002.

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