Detecting Stability in Heterogeneous Networks with Protocol Compositions*

Dimitrios Koukopoulos[1] Stavros D. Nikolopoulos[2] Leonidas Palios [2]

[1] *Department of Cultural Heritage Management and New Technologies, University of Ioannina, GR-30100 Agrinio, Greece.*
koukopou@ceid.upatras.gr

[2] *Department of Computer Science, University of Ioannina, GR-45110 Ioannina, Greece.*
{stavros, palios}@cs.uoi.gr

Abstract: A distinguishing feature of today's large-scale platforms for distributed computation and communication, such as the Internet, is their *heterogeneity,* manifested in particular by the fact that a wide variety of *communication protocols* are simultaneously running over different distributed hosts. A fundamental question that naturally arises in such heterogeneous distributed systems pertains to the stability of a large network in which a *composition* of protocols is employed.
A *packet-switched* network is *stable* under a greedy protocol (or a composition of protocols) if, for any adversary of injection rate less than 1, the number of packets in the network remains bounded at all times. We focus on a basic adversarial model for packet arrival and path determination in which the time-averaged arrival rate of packets requiring a single edge is no more than 1. Within this framework, we study the property of stability under various compositions of contention-resolution protocols (such as LIS (*Longest-in-System*), FIFO (*First-In-First-Out*), FFS (*Furthest-from-Source*), and NTG (*Nearest-to-Go*)) and different packet trajectories (simple and non-simple paths); we provide appropriate adversarial traffic patterns and we obtain instability results for families of network topologies under these compositions of protocols. Additionally, we describe optimal algorithms for detecting these families of topologies; they run in time and space linear on the number of network nodes and links. As these families of topologies characterize the universal stability (stability against any adversary and any protocol), our algorithms can also be used to decide universal stability.

Keywords: Packet-Switched Communication Networks, Network Stability, Linear Algorithms, Adversarial Queueing Theory.

*This research was co-funded by the European Union in the framework of the program "Pythagoras II" of the "Operational Program for Education and Initial Vocational Training" of the 3rd Community Support Framework of the Hellenic Ministry of Education, funded by national sources and the European Social Fund (ESF).

1 Introduction

Motivation-Objective. 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 links (edges). A major issue that arises in such a setting is that of network stability — will the number of packets in the network remain bounded at all times against any adversary under a single contention-resolution protocol (or a composition of protocols)? The stability of a network depends on the network structure, the traffic pattern defined by the adversary and the composition of protocols employed to resolve packet conflicts; the traffic pattern controls where and how packets are injected into the network, and defines their path (trajectory). Deciding the stability of a network may seem at first glance intractable as it is quantified over all adversaries; yet, Alvarez *et al.* [2] showed that the universal stability of networks can be decided in polynomial time and Blesa [5] shows that stability of FIFO networks can also be decided in polynomial time. Deciding the stability under a protocol is usually based on a characterization in terms of network topologies; such a characterization provides us with the family of network topologies that can be made unstable by some adversarial traffic pattern. Such a family of network topologies is the set of *forbidden subgraphs* for network stability.

The underlying goal of our work is two-fold: First, to study the stability of networks when a composition of protocols is employed for contention-resolution on the network queues; by *composition* of contention-resolution protocols, we mean the simultaneous use of different such protocols at different queues of the network. Secondly, to describe efficient algorithms for the detection of forbidden subgraphs in a given network, which will give us information on its stability.

Adversarial Queueing Theory. We consider a packet-switched communication network in which packets arrive dynamically at the nodes with predetermined paths, and they are routed at discrete time steps across the edges. We focus on a basic adversarial model for packet arrival and path determination that has been introduced in a pioneering work by Borodin *et al.* [6] under the name "Adversarial Queueing Theory." Roughly speaking, this model views the time evolution of a packet-switched communication network as a game between an *adversary* and a *protocol*. At each time step, the adversary may inject a set of packets into some nodes. For each packet, the adversary specifies a 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 an edge 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 contain any particular edge. Such a model allows for adversarial injection of packets, rather than for injection according to a randomized, oblivious process (cf. [7]).

Stability. Roughly speaking, a protocol (or a composition of protocols) is *stable* [6] on a network G against an adversary $\mathcal{A}$ of rate r if there is a constant B (which may depend on G and $\mathcal{A}$) such that the number of packets in the system is bounded at all times by B. On the other hand, *a protocol (or a composition of protocols) is universally stable* [6] if it is stable against every adversary of rate less than 1 and on every network. We also say that *a network (graph) G is universally stable* [6] if *every* protocol is stable against every adversary of rate less than 1 on G. Moreover, the property of network stability can be viewed under two different approaches; we refer to *simple-path stability* when packets follow simple paths (paths do not contain repeated edges and vertices), while we refer to *stability* when packets follow non-simple paths (paths do not contain repeated edges, but they may contain repeated vertices) [2].

Greedy Contention-Resolution Protocols. We consider only *greedy* protocols, i.e., protocols that always advance a packet across a queue (one packet at each discrete time step) whenever there is at least one packet in the queue. The protocol specifies which packet will be chosen. We study four greedy protocols (all of which enjoy simple implementations); see Table 1. The protocol LIS is universally stable [3,

Protocol name	Which packet it advances:
Longest-in-System (LIS)	The one that was least recently injected into the network
Nearest-to-Go (NTG)	The one that is nearest to its destination
Furthest-from-Source (FFS)	The one that its furthest from its origin
First-In-First-Out (FIFO)	The one that arrived earliest at the queue

Table 1: Contention-resolution protocols considered in this paper.

Section 2.1]. In contrast, FIFO (one of the most popular queueing disciplines, because of its simplicity) is not universally stable [3, Theorem 2.10]; FFS and NTG are also not universally stable. All these contention-resolution protocols require some tie-breaking rule in order to be unambiguously defined; here, we assume FIFO as a tie-breaking rule.

Contribution. In this work, we study the property of stability under composition of contention-resolution protocols. Our results are summarized as follows:

1. We present adversarial constructions that lead the networks U_1, U_2 (Figure 1) and S_1, S_2, S_3, S_4 (Figure 3) to instability when the following combinations of contention-resolution protocols are employed: (NTG, LIS), (NTG, FFS), (NTG, FFS, LIS), and (NTG, FIFO).
2. We present algorithms that detect whether a network contains as a subgraph the extensions of the networks $U_1, U_2, S_1, S_2, S_3, S_4$ obtained by replacing any edge by a disjoint directed path; any such network is unstable under the investigated compositions of protocols.

Our instability results have the following important consequences: Protocols which are universally stable may lead to instability when combined with other protocols; for example, the LIS protocol which is universally stable leads to instability when combined with NTG. Additionally, networks that have been shown stable for a protocol may become unstable when this protocol is combined with other protocols; for example, the network U_1 which has been proved stable for FIFO [19] becomes unstable under the composition (FIFO, NTG). These results together suggest that composing two protocols may turn out to exhibit more unstable behavior than a single protocol that is already known not to be universally stable (such as FIFO). Finally, our algorithms provide optimal ways to decide universal stability and simple-path universal stability, thus improving over the results by Alvarez *et al.* [2].

Related Work. The issue of composing distributed protocols (resp., objects) to obtain other protocols (resp., objects), and the properties of the resulting composed protocols (resp., objects), has a rich record in Distributed Computing Theory (see, e.g., [18]). For example, Fernández *et al.* [10] study techniques for the composition of (identical) causal DSM systems from smaller modules each individually satisfying causality. Herlihy and Wing [11] establish that a composition of linearizable memory objects (possibly distinct), each managed by its own protocols, preserves linearizability.

Adversarial Queueing Theory [6] received a lot of interest in the study of stability and instability issues (see, e.g., [3, 2, 9, 14, 15, 16, 17]). The universal stability of various natural greedy protocols such as LIS was established by Andrews *et al.* [3]. Also, several greedy protocols such as NTG and FFS have been proved unstable [3]. The instability of FIFO at arbitrarily low rates of injection has been proved by Bhattacharjee *et al.* [4]. The subfield of study of the stability properties of compositions of protocols was introduced by Koukopoulos *et al.* in [14, 15, 16], where the compositions of LIS with any of SIS, NTS and FTG protocols have been proved unstable, while any composition of any pair among SIS, NTS and FTG protocols have been proved stable.

The subfield of charactering universal stability in terms of forbidden subgraphs was first initiated by Andrews *et al.* [3], where a finite set of forbidden subgraphs was provided. This result implies that stability is decidable in polynomial time (however a constructive proof was not presented); the result was significantly improved in [12, 13, 2]. Recently, Blesa [5] presented a polynomial-time algorithm for both FIFO stability and FIFO simple-path stability of directed multi-graphs.

2 Theoretical Framework

The model definitions follow those in [6, Section 3]. A network is modeled by a finite multi-digraph G on n vertices and m edges; the term *multi-digraph* is used when multiple edges are allowed in a digraph. Each vertex $x \in V(G)$ represents a communication switch, and each edge $e \in E(G)$ represents a link between two switches. In each vertex, there is a queue associated with each outgoing edge. Time proceeds in discrete time steps. A *packet* is an atomic entity that resides at a queue at the end of any step. It must travel along paths in the network from its *source* to its *destination,* both of which are nodes in the network. When a packet is injected, it is placed in the queue of the first link on its route. When a packet reaches its destination, it is *absorbed.* At each step, a packet may traverse the edge in whose queue it is waiting. Any packets that wish to travel along an edge e at a particular time step, but they are not sent, they wait in the queue of the edge e. We say that the adversary generates a set of packets when it generates a set of requested paths. The only restriction on how the adversary chooses its requests is that for each edge e and each interval I, no more than $r|I|$ packets are introduced during I with an assigned path containing e. We will restrict our study to the case of *non-adaptive* routing.

The behavior of a network G under the adversarial queueing theory model is fully determined by the strategy of the adversary $\mathcal{A}$ and the set P of protocols on the network queues; thus, we use the triple $\langle G, \mathcal{A}, \mathsf{P} \rangle$ which defines a *system.*

Let G be a graph with no loops that models a routing network. A directed (resp. undirected) edge from x to y is denoted xy. The *multiplicity* of an edge xy, denoted by $\lambda(xy)$, is the number of edges joining the vertex x to y in G. For a set $C \subseteq V(G)$, the subgraph of G *induced* by C is denoted $G[C]$; for a set $S \subseteq E(G)$ of edges, the subgraph of G *spanned* by S is denoted $G\langle S \rangle$.

A *connected component* (or component) of an undirected graph G is a maximal set of vertices, say, $C \subseteq V(G)$, such that for every pair of vertices $x, y \in C$, there exists an $x{-}y$ path in the subgraph $G[C]$. A *biconnected component* (or bicomponent) of an undirected graph G is a maximal set of edges such that any two edges in the set lie on a simple cycle of G [8]; G is called *biconnected* if it is connected and contains only one biconnected component. A *strongly connected component* (or scc) of a directed graph G is a maximal set of vertices $C \subseteq V(G)$ such that for every pair of vertices x and y in the set C, there exists both a directed $x{-}y$ path and a directed $y{-}x$ path in the subgraph of G induced by the vertices in C; the graph G is called *strongly connected* if it is connected and contains only one scc. The *underlined graph* $G_{u\ell}$ of the digraph G is an undirected graph which results after making all the edges of G undirected and consolidating any duplicate edges.

Based on the above, we define a *strongly biconnected component* (or bi-scc) of a directed graph G to be a maximal set of edges $S \subseteq E(G)$ such that the subgraph $G\langle S \rangle$ is strongly connected and the underlined graph $G\langle S \rangle_{u\ell}$ is biconnected; the graph G is called *strongly biconnected* if its edge set $E(G)$ forms a single bi-scc. The graph U_1 of Figure 1 is strongly biconnected, whereas the graph U_2 contains two bi-scc.

The *subdivision* operation on an edge xy of a digraph G consists of the addition of a new vertex w and the replacement of xy by the two edges xw and wy; hereafter, we shall call it *edge-subdivision* operation. Given a digraph G on n vertices and m edges, $\mathcal{E}(G)$ denotes the family of digraphs which contains the digraph G and all the digraphs obtained from G by successive edge-subdivisions.

Our stability study will involve the digraphs U_1 and U_2 depicted in Figure 1; these are of special interest as they are the minimum forbidden subgraphs characterizing universal stability. Moreover, the family of digraphs obtained from U_1 and U_2 by successive edge-subdivisions (i.e., the digraphs in $\mathcal{E}(U_1) \cup \mathcal{E}(U_2)$ [2] — see Figure 2) are also not universally stable. The following result holds:

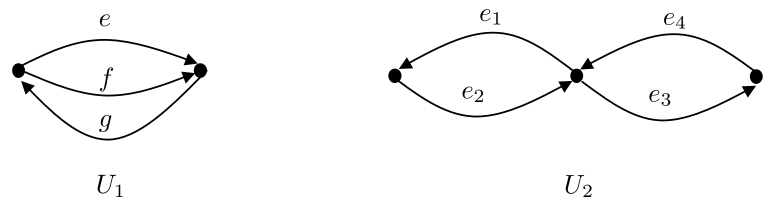

Figure 1: Not universally stable digraphs.

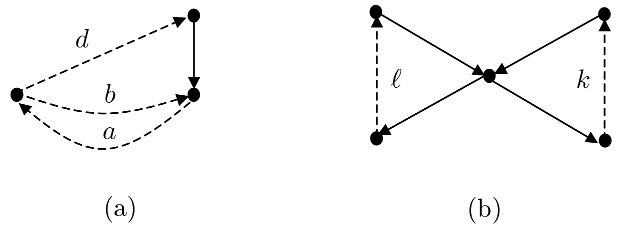

Figure 2: Family of digraphs formed by extensions of U_1 and U_2, where $a \geq 1$, $b \geq 1$, $d \geq 0$, $\ell \geq 0$, and $k \geq 0$. (a) a digraph in $\mathcal{E}(U_1)$; (b) a digraph in $\mathcal{E}(U_2)$.

Lemma 2.1. (Alvarez, Blesa, and Serna [2]): *A digraph G is universally stable if and only if G does not contain as a subgraph any of the digraphs in $\mathcal{E}(U_1) \cup \mathcal{E}(U_2)$.*

As our algorithms will need to detect whether a network contains a subgraph belonging to $\mathcal{E}(U_1) \cup \mathcal{E}(U_2)$, we give below an important property of the structure of these graphs.

Observation 2.1. Let G be a directed graph of the family $\mathcal{E}(U_1) \cup \mathcal{E}(U_2)$. Then, the graph G has the following structure: it contains

(a) a cycle $C = (x_0, x_1, x_2, \ldots, x_\ell, x_0)$, $\ell \geq 1$ and

(b) a path $P = (x_i, y_1, y_2, \ldots, y_k, x_j)$ such that $y_1, y_2, \ldots, y_k \notin C$, $x_i, x_j \in C$ and $k \geq 0$.

It is easy to see that, if P is an open path, i.e, $x_i \neq x_j$, then $G \in \mathcal{E}(U_1)$, whereas if P is a closed path, i.e, $x_i = x_j$, then $G \in \mathcal{E}(U_2)$.

The graphs in the family $\mathcal{E}(U_1) \cup \mathcal{E}(U_2)$ are multi-graphs, which makes it more difficult to handle. In order to avoid working with multi-graphs, we define the one-subdivided graph of a given graph. For a digraph G, the *one-subdivided* graph G^* of G is the element of $\mathcal{E}(G)$ which is obtained from G by applying one edge-subdivision operation on each edge of G. If G has n vertices and m edges, then clearly G^* has $n+m$ vertices and $2m$ edges. Moreover, G^* does not contain 2-cycles (cycle of length 2); in particular, every cycle in G^* has length greater than or equal to 4. Then, we can show:

Lemma 2.2. *Let G be a directed graph and let G^* be its one-subdivided graph. The graph G contains no subgraph in $\mathcal{E}(U_1) \cup \mathcal{E}(U_2)$ if and only if G^* contains no subgraph in $\mathcal{E}(U_1) \cup \mathcal{E}(U_2)$.*

Lemma 2.3. *Let G be a directed graph and let G^* be its one-subdivided graph. Let $C_1, C_2, \ldots, C_k$ be the strongly connected components of G^* and let n_i and m_i be the number of vertices and edges of the strong component C_i, respectively. Then, G contains no subgraph in $\mathcal{E}(U_1) \cup \mathcal{E}(U_2)$ if and only if G^* has a strong component C_i such that $m_i > n_i$, $1 \leq i \leq k$.*

Our stability study will also involve the digraphs S_1, S_2, S_3, S_4 depicted in Figure 3; these are of special interest as well, as they are the minimum forbidden subgraphs characterizing simple-path universal stability. It has been showed that all the digraphs in $\mathcal{E}(S_1) \cup \mathcal{E}(S_2) \cup \mathcal{E}(S_3) \cup \mathcal{E}(S_4)$ (see Figure 4) are not simple-path universally stable [2].

Lemma 2.4. (Alvarez, Blesa, and Serna [2]): *A digraph G is simple-path universally stable if and only if G does not contain as a subgraph any of the digraphs in $\mathcal{E}(S_1) \cup \mathcal{E}(S_2) \cup \mathcal{E}(S_3) \cup \mathcal{E}(S_4)$.*

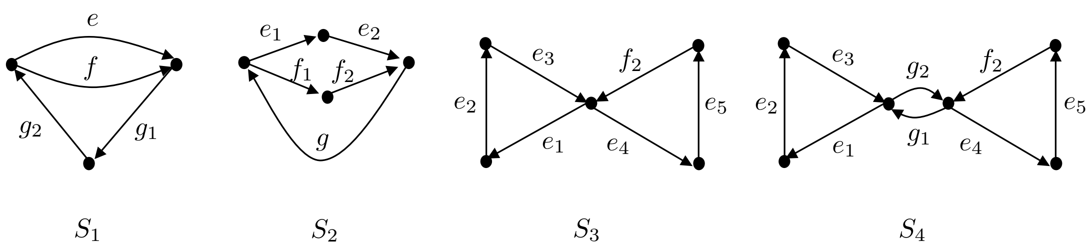

Figure 3: Not simple-path universally stable digraphs.

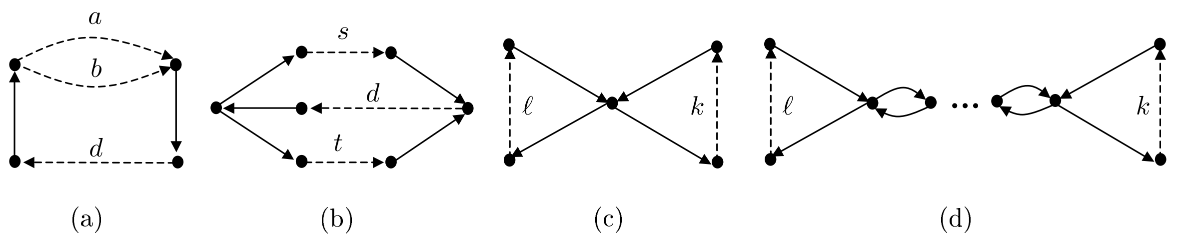

Figure 4: Family of digraphs formed by extensions of S_1, S_2, S_3, and S_4, where $a \geq 1$, $b \geq 1$, $d \geq 0$, $s \geq 0$, $t \geq 0$, $\ell \geq 1$, and $k \geq 1$. (a) a digraph in $\mathcal{E}(S_1)$; (b) a digraph in $\mathcal{E}(S_2)$; (c) a digraph in $\mathcal{E}(S_3)$; (d) a digraph in $\mathcal{E}(S_4)$.

In order to be able to detect whether a multi-digraph G contains a graph in $\mathcal{E}(S_1) \cup \mathcal{E}(S_2) \cup \mathcal{E}(S_3) \cup \mathcal{E}(S_4)$, we will consider the digraph $\widehat{G}$ obtained from G by setting the multiplicity of each edge of G to 1; we call $\widehat{G}$ the *reduced* graph of G. Obviously, $\widehat{G}$ is a simple digraph and for a multi-digraph G on n vertices and m edges, $\widehat{G}$ has n vertices and $m' \leq m$ edges. Below we present a result on which our algorithm for detecting subgraphs in $\mathcal{E}(S_1) \cup \mathcal{E}(S_2) \cup \mathcal{E}(S_3) \cup \mathcal{E}(S_4)$ relies.

Lemma 2.5. *Let G be a directed graph, $\widehat{G}$ be the reduced graph of G, and $C_1, C_2, \ldots, C_k$ be the scc of $\widehat{G}$. Let $C_{i,1}, C_{i,2}, \ldots, C_{i,k_i}$ be the bi-scc of the scc C_i and let $n_{i,j}$ and $m_{i,j}$ be the number of vertices and edges of the bi-scc $C_{i,j}$, respectively. Then, G contains no subgraph in $\mathcal{E}(S_1) \cup \mathcal{E}(S_2) \cup \mathcal{E}(S_3) \cup \mathcal{E}(S_4)$ if and only if $\widehat{G}$ has a strong component C_i which satisfies one of the following conditions:*

(i) *C_i contains a bi-scc $C_{i,j}$ such that: $n_{i,j} \geq 3$, $G\langle C_{i,j}\rangle_{u\ell}$ is a cycle, and there exists an edge xy in $G\langle C_{i,j}\rangle$ such that $\lambda(xy) \geq 2$;*

(ii) *C_i contains a bi-scc $C_{i,j}$ such that: $n_{i,j} \geq 3$ and $G\langle C_{i,j}\rangle_{u\ell}$ is not a cycle;*

(iii) *C_i contains two bi-scc $C_{i,p}$ and $C_{i,q}$ such that: $n_{i,p} \geq 3$, $n_{i,q} \geq 3$, and both graphs $G\langle C_{i,p}\rangle_{u\ell}$ and $G\langle C_{i,q}\rangle_{u\ell}$ are cycles;*

where $1 \leq i \leq k$ and $1 \leq j, p, q \leq k_i$.

3 Stability Under Compositions of Protocols

In this section, we show that the networks $U_1, U_2, S_1, S_2, S_3, S_4$ (Figures 1 and 3) are unstable under specific compositions of the NTG protocol with the LIS, FFS, and FIFO protocols. In order to establish the instability of a given network G under a composition of protocols, we first specify an initial configuration of G (i.e., the paths to be followed by the packets in G), and we construct a strategy for an adversary which results in a configuration identical to the initial configuration except with an increased number of packets. Then, if the strategy is repeatedly applied, the number of packets will exceed any bound, which will imply the lack of stability of G under the studied composition of protocols. Additionally, thanks to the work of Andrews *et al.* [3, Lemma 2.9], our results also imply lack of stability for networks with an *empty* initial configuration.

For simplicity, and in a way similar to that in [3] and in works following it, we omit floors and ceilings from our analysis, and we, sometimes, count time steps and packets only roughly. This may result in the loss of small additive constants, whereas it implies a gain in clarity.

3.1 Instability under Compositions of NTG with FFS and LIS

In this section we consider all combinations of the NTG protocol with the LIS and FFS protocols when packets are injected with non-simple and simple paths.

We start with the network U_1 for the composition of NTG with LIS protocol where packets are injected with non-simple paths. We have:

Theorem 3.1.1 *For the network U_1, there is an adversary $\mathcal{A}$ of rate $r \geq 0.841$ such that the system $\langle U_1, \mathcal{A}, (\mathsf{NTG}, \mathsf{LIS})\rangle$ is unstable.*

Proof. Consider that the edge f uses the LIS protocol, and that the edges e, g use the NTG protocol. We assume that at a given time, the network U_1 contains a set S of packets waiting in the queues of the edges e, f in order to traverse the edges e, g and f, g, respectively. We will construct a strategy for an adversary $\mathcal{A}$ such that after the application of this strategy, the network U_1 will be in a configuration similar to the starting configuration described above, for a set S' of packets, where $|S'| > |S|$. Then, if the strategy is repeatedly applied, the number of packets will exceed any bound, which will imply the lack of stability of U_1 under the stated composition of protocols.

The strategy consists of the following four rounds:

Round 1 ($|T_1| = |S|$ time steps): The adversary injects a set Z_1 of $|Z_1| = r|T_1|$ packets in g scheduled to traverse the edges g, f. The packets in S traverse e or f and reach g. If we assume that the first such packet reaches g before the first packet in Z_1 (if not, a constant offset of 1 needs to be included), then at the completion of the round, all the packets in S have been absorbed after having traversed g whereas the packets in Z_1 are queued in g; recall that $T_1| = |S|$ and note that in g the packets in S have priority over those in Z_1 because the former are closer to their destination.

Round 2 ($|T_2| = r|T_1|$ time steps): The adversary injects a set Z_2 of $|Z_2| = r|T_2|$ packets in g scheduled to traverse the edges g, e and a set Z_3 of $|Z_3| = r|T_2|$ packets in f scheduled to traverse f. In g, the packets in Z_1 have priority over those in Z_2; these flows have the same number of edges to traverse to reach their destination, but the packets in Z_1 have been waiting longer in g. Therefore, all the packets in Z_1 arrive at the queue of f along with the packets in Z_3. The total number of packets arriving at f during this round is $|Z_1| + |Z_3|$; as the duration of this round is $|T_2|$ time steps, $|T_2|$ of these packets traverse the edge f during this round. Thus, at the end of this round, the queue of f will contain a set X of $|X| = |Z_1| + |Z_3| - |T_2| = r|T_2|$ packets waiting to traverse the edge f and the queue of g will contain the packets in Z_2 waiting to traverse the edges g, e.

Round 3 ($|T_3| = r|T_2|$ time steps): The adversary injects a set Z_4 of $|Z_4| = r|T_3|$ packets in f scheduled to traverse f and a set Z_5 of $|Z_5| = r|T_3|$ packets in e scheduled to traverse e, g. In f, the packets in X have priority over those in Z_4 because the former have been longer in the system. Additionally, in e, the packets in Z_2 have priority over those in Z_5 because the former are closer to their destination. Thus, at the end of this round, the queue of f will contain the packets in Z_4 waiting to traverse the edge f and the queue of e will contain the packets in Z_5 waiting to traverse the edges e, g.

Round 4 ($|T_4| = r|T_3|$ time steps): The adversary injects a set Z_6 of $|Z_6| = r|T_4|$ packets in e scheduled to traverse e and a set Z_7 of $|Z_7| = r|T_4|$ packets in f scheduled to traverse f, g. In f, the packets in Z_4 have priority over those in Z_7 because the former have been longer in the system. Additionally, in e, the packets in Z_6 have priority over those in Z_5 because the former are closer to their destination; thus, at the end of this round, there are $|Z_5| + |Z_6| - |T_4| = r|T_4|$ packets in e waiting to traverse e, g.

In total, at the end of this round, the network contains a set S' of packets waiting in the queues of e and f to traverse the edges e, g and f, g, respectively, where $|S'| = |Z_7| + r|T_4| = 2r|T_4|$.

Observe that the situation at the end of the strategy is similar to the one in the beginning. Moreover, $|S'| > |S| \iff 2r|T_4| > |S| \iff r > \sqrt[4]{\frac{1}{2}} \approx 0.8409$. ∎

A similar theorem holds for the network U_1 under the compositions (NTG, FFS) and (NTG, LIS, FFS) when packets are injected with non-simple paths.

Theorem 3.1.2 *For the network U_1, there is an adversary $\mathcal{A}_i$ of rate $r \geq 0.841$ such that the system $\langle U_1, \mathcal{A}_i, \mathsf{P}_i \rangle$ is unstable where $i = \{1, 2\}$ and $\mathsf{P}_i \in \{(\mathsf{NTG}, \mathsf{FFS}), (\mathsf{NTG}, \mathsf{LIS}, \mathsf{FFS})\}$.*

As in Theorem 3.1.1, we assume that, in the beginning, the network contains packets waiting in the queues of the edges e, f to traverse the edges e, g and f, g, respectively. For the composition of protocols (NTG, FFS), the queue of f uses FFS and those of e, g use NTG; for (NTG, LIS, FFS), the queue of f uses LIS, that of g uses FFS, and that of e uses NTG. In all three cases, the adversaries' strategies consist of four rounds (for details, see the Appendix).

We can show similar results for the network U_2 where packets may follow non-simple paths, and the networks S_1, S_2, S_3 and S_4 where packets are injected with simple paths. In particular,

Theorem 3.1.3 *For the network U_2, there is an adversary $\mathcal{A}_i$ of rate $r \geq 0.794$ such that the system $\langle U_2, \mathcal{A}_i, \mathsf{P}_i \rangle$ is unstable where $i = \{1, 2, 3\}$ and $\mathsf{P}_i \in \{(\mathsf{NTG}, \mathsf{LIS}), (\mathsf{NTG}, \mathsf{FFS}), (\mathsf{NTG}, \mathsf{LIS}, \mathsf{FFS})\}$.*

In this case, we assume that, in the beginning, the network contains packets waiting in the queues of the edges e_2, e_3 to traverse the edges e_2, e_1 and e_3, e_4, e_1, respectively. For the composition of protocols (NTG, LIS), the queue of the edge e_4 uses LIS while all the other queues use NTG; for (NTG, FFS), the queue of e_4 uses FFS and all the other ones NTG; for (NTG, LIS, FFS), the queue of e_4 uses LIS, the queue of e_1 uses FFS, and those of e_2, e_3 use NTG. In all three cases, the adversaries' strategies consist of three rounds (for details, see the Appendix).

Theorem 3.1.4 *For the network S_i ($i = 1, 2, 3, 4$), there is an adversary $\mathcal{A}_i$ of rate $r \geq 0.841$ such that the systems $\langle S_i, \mathcal{A}_i, (\mathsf{NTG}, \mathsf{LIS}) \rangle$, $\langle S_i, \mathcal{A}_i, (\mathsf{NTG}, \mathsf{FFS}) \rangle$ and $\langle S_i, \mathcal{A}_i, (\mathsf{NTG}, \mathsf{LIS}, \mathsf{FFS}) \rangle$ are unstable where $i = \{1, 2, 3, 4\}$.*

For the network S_1, we assume that, in the beginning, the network contains packets waiting in the queues of the edges e, f to traverse the edges e, g_1 and f, g_1, respectively. For the composition (NTG, LIS), the queue of f uses LIS and all other queues use NTG; for (NTG, FFS), the queue of f uses FFS and all the others use NTG; for (NTG, LIS, FFS), f uses LIS, g_1 uses FFS, and e, g_2 use NTG. For the network S_2, we assume that, in the beginning, the network contains packets waiting in the queues of the edges e_2, f_2 to traverse the edges e_2, g and f_2, g, respectively. For (NTG, LIS), the queue of g uses LIS and all other queues use NTG; for (NTG, FFS), g uses FFS and all others use NTG; for (NTG, LIS, FFS), g uses FFS, f_2 uses LIS, and e_1, e_2, f_1 use NTG. For the network S_3, we assume that, in the beginning, the network contains packets waiting in the queues of the edges e_3, e_5 to traverse the edges e_3, e_1 and e_5, e_6, e_1, respectively. For (NTG, LIS), the queues of e_1, e_2 use LIS and all other queues use NTG; for (NTG, FFS), the queues of e_1, e_2 use FFS and all others use NTG; for (NTG, LIS, FFS), the queue of e_6 uses LIS, e_1, e_2 use FFS, and e_3, e_4, e_5 use NTG. For the network S_4, we assume that, in the beginning, the network contains packets waiting in the queues of the edges e_3, e_5 to traverse the edges e_3, e_1 and e_5, e_6, g_1, e_1, respectively. For (NTG, LIS), the queues of e_1, e_2 use LIS and all other queues use NTG; for (NTG, FFS), the queues of e_1, e_2 use FFS and all others use NTG; for (NTG, LIS, FFS), the queue of e_6 uses LIS, the queues of e_1, e_2 use FFS, and e_3, e_4, e_5, g_1, g_2 use NTG. In all cases, the adversaries' strategies consist of four rounds (for details, see the Appendix).

3.2 Instability under Compositions of NTG with FIFO

In this section we consider the composition of the NTG and FIFO protocols. Again, we start with the network U_1 where packets are injected with non-simple paths. We have:

Theorem 3.2.1 *For the network U_1, there is an adversary $\mathcal{A}$ of rate $r \geq 0.841$ such that the system $\langle U_1, \mathcal{A}, (\mathsf{NTG}, \mathsf{FIFO})\rangle$ is unstable.*

Proof. Consider that the edge e uses the FIFO protocol, and that the edges f, g use the NTG protocol. We work as in the proof of Theorem 3.1.1: we assume that initially the network U_1 contains a set S of packets waiting in the queues of the edges e, f in order to traverse the edges e, g and f, g, respectively; we will describe a strategy for an adversary $\mathcal{A}$ which results in a set S' of packets in the queues of e, f, where $|S'| > |S|$. The strategy consists of the following four rounds:
Rounds 1 and **2** of the strategy are identical to the corresponding rounds in the proof of Theorem 3.1.1; although the protocols on the edges e, f are different, they have the same effect. Thus, at the end of Round 2, the queue of f will contain a set X of $|X| = r|T_2|$ packets waiting to traverse the edge f and the queue of g will contain the set Z_2 of $|Z_2| = r|T_2|$ packets waiting to traverse the edges g, e.
Round 3 ($|T_3| = r|T_2|$ time steps): The adversary injects a set Z_4 of $|Z_4| = r|T_3|$ packets in the queue of e scheduled to traverse e and a set Z_5 of $|Z_5| = r|T_3|$ packets in f scheduled to traverse the edges f, g. In f, the packets in X have priority over those in Z_5 because the former are closer to their destination. Additionally, the packets in Z_4 arrive at e along with the packets in Z_2. The total number of packets arriving at e during this round is $|Z_4| + |Z_2|$ packets; as the duration of this round is $|T_3|$ time steps, $|T_3|$ packets traverse e during this round. Thus at the end of this round, there are $|Z_5| = r|T_3|$ packets in the queue of f waiting to traverse f, g and a set Y of $|Y| = r|T_3|$ packets in e waiting to traverse e.
Round 4 ($|T_4| = r|T_3|$ time steps): The adversary injects a set Z_6 of $|Z_6| = r|T_4|$ packets in the queue of e scheduled to traverse the edges e, g and a set Z_7 of $|Z_7| = r|T_4|$ packets in the queue of f scheduled to traverse the edges f, g. Among the packets in $Z_5 \cup Z_7$, $|T_4|$ traverse f and thus at the end of this round, there are $|W| = r|T_4|$ packets in f waiting to traverse f, g. Additionally, in e, the packets in Y have priority over those in Z_6 because the former reached the queue of e first. In total, at the end of this round, there are $|W| + |Z_6| = 2r|T_4|$ packets in the queues of e, f waiting to traverse the edges e, g and f, g, respectively.

Again, the situation at the end of the strategy is similar to the one in the beginning. Moreover, $|S'| > |S| \iff 2r|T_4| > |S| \iff r > \sqrt[4]{\frac{1}{2}} \approx 0.8409$. ∎

Next, consider the network U_2 where packets are injected with non-simple paths. Similarly to Theorem 3.2.1 we can show:

Theorem 3.2.2 *For the network U_2, there is an adversary $\mathcal{A}$ of rate $r \geq 0.867$ such that the system $\langle U_2, \mathcal{A}, (\mathsf{NTG}, \mathsf{FIFO})\rangle$ is unstable.*

The queues of the edges e_2, e_4 of U_2 use FIFO, and the queues of e_1, e_3 use NTG. For the adversary's strategy, we assume that, in the beginning, the network contains packets waiting in the queues of the edges e_2, e_3 to traverse the edges e_2, e_1 and e_3, e_4, e_1, respectively; the adversary's strategy consists of three rounds (for details, see the Appendix).

Finally, we consider the networks S_1, S_2, S_3, and S_4, where packets are injected with simple paths. Then, we can show:

Theorem 3.2.3 *For the network S_i ($i = 1, 2$), there is an adversary $\mathcal{A}_i$ of rate $r \geq 0.908$ such that the system $\langle S_i, \mathcal{A}_i, (\mathsf{NTG}, \mathsf{FIFO})\rangle$ is unstable.*

Theorem 3.2.4 *For the network S_i ($i = 3, 4$), there is an adversary $\mathcal{A}_i$ of rate $r \geq 0.9$ such that the system $\langle S_i, \mathcal{A}_i, (\mathsf{NTG}, \mathsf{FIFO})\rangle$ is unstable.*

In all four cases, the adversary's strategy consists of four rounds (see Appendix). For $\langle S_1, \mathcal{A}_1, (\mathsf{NTG}, \mathsf{FIFO})\rangle$, the queue of f uses FIFO and those of e, g_1, g_2 use NTG; we assume that initially we have packets waiting in the queues of e, f in order to traverse the edges e, g_1 and f, g_1, respectively. For $\langle S_2, \mathcal{A}_2, (\mathsf{NTG}, \mathsf{FIFO})\rangle$, the queue of f_2 uses FIFO and those of f_1, g, e_1, e_2 use NTG; in this case, initially the packets are waiting in the queues of e_2, f_2 in order to traverse the edges e_2, g and f_2, g, respectively. For $\langle S_3, \mathcal{A}_3, (\mathsf{NTG}, \mathsf{FIFO})\rangle$, the queues of e_3, e_6 use FIFO and the queues of e_1, e_2, e_4, e_5 use NTG; initially, the packets are in the queues of e_3, e_5 waiting to traverse the edges e_3, e_1 and e_5, e_6, e_1, respectively. For $\langle S_4, \mathcal{A}_4, (\mathsf{NTG}, \mathsf{FIFO})\rangle$, the queues of e_3, e_6 use FIFO and the queues of $e_1, e_2, e_4, e_5, g_1, g_2$ use NTG; the packets are in the queues of e_3, e_5 waiting to traverse the edges e_3, e_1 and e_5, e_6, g_1, e_1, respectively.

The application of subdivision operations to $U_1, U_2, S_1, S_2, S_3, S_4$ (as in [5]) in combination with Theorems 3.1.1-4 and 3.2.1-4 enables us to show that all the graphs in $\mathcal{E}(U_1) \cup \mathcal{E}(U_2) \cup \mathcal{E}(S_1) \cup \mathcal{E}(S_2) \cup \mathcal{E}(S_3) \cup \mathcal{E}(S_4)$ are unstable under the investigated compositions of protocols. Thus, we have:

Corollary 3.1 *If a network G contains any graph in $\mathcal{E}(U_1) \cup \mathcal{E}(U_2) \cup \mathcal{E}(S_1) \cup \mathcal{E}(S_2) \cup \mathcal{E}(S_3) \cup \mathcal{E}(S_4)$, than it is unstable under any of the compositions* (LIS, NTG), (FFS, NTG), (LIS, FFS, NTG), (FIFO, NTG) *of protocols.*

4 Detecting Unstable Subgraphs in a Given Network

In light of Corollary 3.1, a network is unstable under each of the investigated compositions of protocols if it contains as a subgraph an element of the set $\mathcal{E}(U_1) \cup \mathcal{E}(U_2) \cup \mathcal{E}(S_1) \cup \mathcal{E}(S_2) \cup \mathcal{E}(S_3) \cup \mathcal{E}(S_4)$. Therefore, it is important to be able to detect whether a given digraph contains a subgraph belonging to these families; in this section, we present optimal algorithms for doing so. Moreover, as the existence of subgraphs in a network G belonging to $\mathcal{E}(U_1) \cup \mathcal{E}(U_2)$ (resp., $\mathcal{E}(S_1) \cup \mathcal{E}(S_2) \cup \mathcal{E}(S_3) \cup \mathcal{E}(S_4)$), determines the universal stability (resp., simple-path universal stability) of G (Lemmas 2.1 and 2.4), our algorithms constitute optimal decision procedures for the (simple-path) universal stability of networks.

4.1 Detecting the Graphs in $\mathcal{E}(U_1) \cup \mathcal{E}(U_2)$

Our algorithm for detecting the existence of subgraphs belonging to $\mathcal{E}(U_1) \cup \mathcal{E}(U_2)$ relies on the result stated in Lemma 2.3; it works as follows:

Algorithm Detect-U-Family

1. Construct the one-subdivided graph G^* of the input digraph G;
2. Compute the strongly connected components $S_1, S_2, \ldots, S_k$ of the digraph G^*, and the number of vertices n_i and edges m_i of each strong component S_i, $1 \leq i \leq k$;
3. for $i = 1$ to k do
   if $m_i > n_i$ then return that G contains an element of $\mathcal{E}(U_1) \cup \mathcal{E}(U_2)$; exit;
4. return that G does not contain an element of $\mathcal{E}(U_1) \cup \mathcal{E}(U_2)$;

The correctness of the algorithm follows from Lemma 2.3. Regarding its time and space complexity, we have that, for a digraph G on n vertices and m edges, the one-subdivided graph G^* has $n + m$ vertices and $2m$ edges; G^* can be constructed in $O(n + m)$ time, and its strong components can also be computed in $O(n + m)$ time. Thus, the whole algorithm runs in $O(n + m)$ time; the space needed is $O(n + m)$. Hence, we have:

Theorem 4.1. *Using Algorithm Detect-U-Family, we can decide whether a digraph G on n vertices and m edges contains a graph in $\mathcal{E}(U_1) \cup \mathcal{E}(U_2)$ as a subgraph in $O(n + m)$ time and space.*

4.2 Detecting the Graphs in $\mathcal{E}(S_1) \cup \mathcal{E}(S_2) \cup \mathcal{E}(S_3) \cup \mathcal{E}(S_4)$

Our algorithm for detecting the existence of subgraphs belonging to $\mathcal{E}(S_1) \cup \mathcal{E}(S_2) \cup \mathcal{E}(S_3) \cup \mathcal{E}(S_4)$ relies on the result of Lemma 2.5; it works as follows:

Algorithm Detect-S-Family

1. Construct the reduced graph $\widehat{G}$ of the input digraph G;
2. Compute the strong components $C_1, C_2, \ldots, C_k$ of the graph $\widehat{G}$, $1 \le i \le k$;
3. Compute the bi-scc $C_{i,1}, C_{i,2}, \ldots, C_{i,k_i}$ of each strong component S_i, $1 \le i \le k$, and the number of vertices $n_{i,j}$ and edges $m_{i,j}$ of the bi-scc $C_{i,j}$, $1 \le j \le k_i$;
4. for $i = 1$ to k do
    for $j = 1$ to k_i do
        if $n_{i,j} \ge 3$ and $G\langle C_{i,j}\rangle_{u\ell}$ is not a cycle
        then return that G contains an element of $\mathcal{E}(S_2)$; exit;
        if $n_{i,j} \ge 3$ and $G\langle C_{i,j}\rangle_{u\ell}$ is a cycle
        then if there exists an edge xy in $G\langle C_{i,j}\rangle$ such that $\lambda(xy) \ge 2$
            then return that G contains an element of $\mathcal{E}(S_1)$; exit;
            else mark the bi-scc $C_{i,j}$;
    if C_i contains at least two marked bi-scc
    then return that G contains an element of $\mathcal{E}(S_3) \cup \mathcal{E}(S_4)$; exit;
5. return that G does not contain an element of $\mathcal{E}(S_1) \cup \mathcal{E}(S_2) \cup \mathcal{E}(S_3) \cup \mathcal{E}(S_4)$;

The correctness of the algorithm follows from Lemma 2.5. For an input graph G on n vertices and m edges, the construction of the reduced graph $\widehat{G}$ can be done in $O(n+m)$ time. The graph $\widehat{G}$ has n vertices and $m' \le m$ edges, and, thus, its strong components can be completed in $O(n+m)$ time. The bi-scc $C_{i,1}, C_{i,2}, \ldots, C_{i,k_i}$ of each strong component C_i, $1 \le i \le k$, can be computed in $O(n+m)$ time because $n_{i,j} \le m_{i,j}$ and $\sum_{j=1,k_i} m_{i,j} \le m_i$ since the bi-scc do not share edges. It is not difficult to see that all the operation of Step 4 are executed in linear time. Thus, the algorithm runs in $O(n+m)$ time; the space needed is $O(n+m)$. Therefore, we can state the following result:

Theorem 4.2. *Using Algorithm Detect-S-Family, we can decide whether a digraph G on n vertices and m edges contains a graph in $\mathcal{E}(S_1) \cup \mathcal{E}(S_2) \cup \mathcal{E}(S_3) \cup \mathcal{E}(S_4)$ as a subgraph in $O(n+m)$ time and space.*

5 Concluding Remarks

In this work, we proved instability results for families of network topologies under various compositions of contention-resolution protocols using the Adversarial Queueing Model and described optimal algorithms for detecting these families of network topologies; they run in time and space linear on the number of network nodes and links. Our algorithms can also be used to decide universal stability in linear time and space.

An interesting direction for further research would be to investigate whether other compositions of protocols are unstable on specific network topologies. Especially, it would be interesting to characterize the stability of the compositions of LIS with any of the SIS, NTS and FTG protocols that have been proved unstable in [14]. As far as it concerns single protocols, only the characterization of stability under FFS, FIFO and NTG-LIS are known [1, 2, 5, 19].

It would also be interesting to see whether the approach and algorithmic techniques used in this paper for detecting the $\mathcal{E}(U_1) \cup \mathcal{E}(U_2)$ and $\mathcal{E}(S_1) \cup \mathcal{E}(S_2) \cup \mathcal{E}(S_3) \cup \mathcal{E}(S_4)$ network topologies, can help develop optimal algorithms for other network topologies that are unstable under single or compositions of contention-resolution protocols.

References

[1] C. Alvarez, M. Blesa, J. Diaz, A. Fernandez, M. Serna, The Complexity of Deciding Stability under FFS in the Adversarial Model, *Information Processing Letters* **90**, 261–266, 2004.

[2] C. Alvarez, M. Blesa, M. Serna, A Characterization of Universal Stability in the Adversarial Queuing Model, *SIAM J. Computing* **34**, 41–66, 2004.

[3] M. Andrews, B. Awerbuch, A. Fernandez, J. Kleinberg, T. Leighton, Z. Liu, Universal Stability Results for Greedy Contention-Resolution Protocols, *J. ACM* **48**, 39–69, 2001.

[4] R. Bhattacharjee, A. Goel, Z. Lotker, Instability of FIFO at Arbitrarily Low Rates in the Adversarial Queueing Model, *SIAM J. Computing* **34**, 318–332, 2004.

[5] M. Blesa, Deciding Stability in Packet-Switched FIFO Networks Under the Adversarial Queuing Model in Polynomial Time, *Proc. 19th International Symposium on Distributed Computing*, LNCS 3724, pp. 429–441, 2005.

[6] A. Borodin, J. Kleinberg, P. Raghavan, M. Sudan, D. Williamson, Adversarial Queueing Theory, *J. ACM* **48**, 13–38, 2001.

[7] H. Chen, D. D. Yao, *Fundamentals of Queueing Networks*, Springer-Verlag, 2000.

[8] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms* (2nd edition), MIT Press, Inc., 2001.

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

[10] A. Férnandez, E. Jiménez and V. Cholvi, On the Interconnection of Causal Memory Systems, *Proc. 19th Annual ACM Symposium on Principles of Distributed Computing*, pp. 163–170, 2000.

[11] M. P. Herlihy and J. Wing, Linearizability: A Correctness Condition for Concurrent Objects, *ACM Transactions on Programming Languages and Systems* **12**, 463–492, 1990.

[12] D. Gamarnik, Stability of Adaptive and NonAdaptive Packet Routing Policies in Adversarial Queueing Networks, *SIAM J. Computing* **32**, 371–385, 2003.

[13] A. Goel, Stability of Networks and Protocols in the Adversarial Queueing Model for Packet Routing, *Networks* **37**, 219–224, 2001.

[14] D. Koukopoulos, M. Mavronicolas, S. Nikoletseas, P. Spirakis, On the Stability of Compositions of Universally Stable, Greedy, Contention-Resolution Protocols, *Proc. 16th Int'l Symposium on DIStributed Computing*, LNCS 2508, pp. 88–102, 2002.

[15] D. Koukopoulos, M. Mavronicolas, S. Nikoletseas, P. Spirakis, The Impact of Network Structure on the Stability of Greedy Protocols, *Theory of Computing Systems* **38**, 425–460, 2005.

[16] D. Koukopoulos, S. Nikoletseas, P. Spirakis, Stability Issues in Heterogeneous and FIFO Networks under the Adversarial Queueing Model, *Proc. 8th Int'l Conference on High Performance Computing*, LNCS 2228, pp. 3–14, 2001.

[17] Z. Lotker, B. Patt-Shamir, A. Rosén, New Stability Results for Adversarial Queuing, *SIAM J. Computing* **33**, 286–303, 2004.

[18] N. Lynch, *Distributed Algorithms,* Morgan Kaufmann, 1996.

[19] M. Weinard, Deciding the FIFO Stability of Networks in Polynomial Time, *Proc. 8th Int'l Conference on Algorithms and Complexity*, LNCS 3998, pp. 81–92, 2006.

Appendix

(Proofs of Lemmas)

A Proof of Lemma 2.5

By its definition, it follows that

(a) a bi-scc $C_{i,j}$ of the scc C_i of the digraph $\widehat{G}$ either is a cycle $O = (x_0, x_1, x_2, \ldots, x_r, x_0)$, $r \geq 2$, or contains a cycle $O = (x_0, x_1, x_2, \ldots, x_r, x_0)$, $r \geq 2$, and a path $P = (x_i, y_1, y_2, \ldots, y_{r'}, x_j)$ such that $y_1, y_2, \ldots, y_{r'} \notin O$, $x_i \neq x_j$ and $r' \geq 0$;

(b) two bi-scc $C_{i,p}$ and $C_{i,q}$ of the scc C_i have at most 1 vertex in common.

For each scc C_i, $1 \leq i \leq k$, let us consider the following undetected graph $\widetilde{G}_i$: it consists of k_i vertices $v_{i,1}, v_{i,2}, \ldots, v_{i,k_i}$, which correspond to the bi-scc $C_{i,1}, C_{i,2}, \ldots, C_{i,k_i}$, and two vertices v_p, v_q are connected by an edge in $\widetilde{G}_i$ if the corresponding bi-scc $C_{i,p}$ and $C_{i,q}$ have a common vertex. From the properties of the bi-scc of C_i, it is easy to see that the graph $\widetilde{G}_i$ is a tree.

($\Longleftarrow$) It is easy to see that if condition (i) holds, then the graph $G\langle C_i\rangle$ contains a subgraph $H \in \mathcal{E}(S_1)$. If condition (ii) holds, then the bi-scc $C_{i,j}$ contains a cycle $O = (x_0, x_1, x_2, \ldots, x_r, x_0)$, $r \geq 2$, and a path $P = (x_i, y_1, y_2, \ldots, y_{r'}, x_j)$ such that $x_i \neq x_j$ and $r' \geq 0$. Thus, the graph $G\langle C_i\rangle$ contains a subgraph $H \in \mathcal{E}(S_2)$. Suppose now that condition (iii) holds and there exists no bi-scc $C_{i,j}$ which satisfies conditions (i) or (ii). Then, all the bi-scc of the scc C_i are cycles. Let $v_{i,p}$ and $v_{i,q}$ be the vertices of $\widetilde{G}_i$ which correspond to the bi-scc $C_{i,p}$ and $C_{i,q}$. Since the graph $\widetilde{G}_i$ is a tree, there exists a unique $v_{i,p}$–$v_{i,q}$ path P in $\widetilde{G}_i$; let $(v_{i,p}, v_{i,1}, v_{i,2}, \ldots, v_{i,j}, v_{i,q})$ be the path P, $j \geq 0$. If the bi-scc $C_{i,1}$ has length $\ell \geq 3$, i.e., it consists of $n_{i,1} \geq 3$ vertices, then the graph $G\langle C_i\rangle$ contains a subgraph $H \in \mathcal{E}(S_3)$, otherwise (if $C_{i,1}$ is a 2-cycle) $G\langle C_i\rangle$ contains a subgraph $H \in \mathcal{E}(S_4)$.

($\Longrightarrow$) Suppose now that G is not simple-path universally stable. Then, G contains a subgraph $H \in \mathcal{E}(S_1) \cup \mathcal{E}(S_2) \cup \mathcal{E}(S_3) \cup \mathcal{E}(S_4)$, and, thus, H contains a cycle $O = (x_0, x_1, x_2, \ldots, x_r, x_0)$, $r \geq 2$. It follows that O belongs to a scc C_i of the graph $\widehat{G}$, $1 \leq i \leq k$. Let $C_{i,j}$ be the bi-scc of C_i which contains the cycle O. Since $r \geq 2$, the bi-scc $C_{i,j}$ has at least three vertices, i.e., $n_{i,j} \geq 3$. We distinguish two cases:
Case (a): $C_{i,j}$ contains the cycle O and a path $P = (x_i, y_1, y_2, \ldots, y_{r'}, x_j)$, $r' \geq 0$. Then, $H \in \mathcal{E}(S_2)$ and $G\langle C_{ij}\rangle_{u\ell}$ is not a cycle. Thus, condition (ii) holds.
Case (b): $C_{i,j}$ contains only the cycle $O = (x_0, x_1, x_2, \ldots, x_r, x_0)$, $r \geq 2$. If there exists an edge $x_i x_{i+1 \bmod r}$ in O such that $\lambda(x_i x_{i+1 \bmod r}) \geq 2$ in G, then $H \in \mathcal{E}(S_1)$ and, since $G\langle C_{i,j}\rangle_{u\ell}$ is a cycle, condition (i) holds. If there exists no such edge in O, then $H \in \mathcal{E}(S_3) \cup \mathcal{E}(S_4)$. Thus, H contains another cycle $O' = (x'_0, x'_1, x'_2, \ldots, x'_r, x'_0)$, $r \geq 2$, which belongs to a bi-scc, say, $C_{i,q}$, of C_i. If conditions (i) and (ii) do not hold for the bi-scc $C_{i,q}$, then the graph $G\langle C_{i,q}\rangle_{u\ell}$ is a cycle and $n_{i,q} \geq 3$. Thus, condition (iii) holds.

B Proof of Theorem 3.1.2

System $\langle U_1, \mathcal{A}_1, (\mathsf{NTG}, \mathsf{FFS})\rangle$.
Initially, there is a set S of packets waiting in the queues of e, f in order to traverse e, g and f, g, respectively. The adversary's strategy consists of four rounds of injections as follows:
Round 1 ($|T_1| = |S|$ time steps): The adversary injects in the queue of g a set Z_1 of $|Z_1| = r|T_1|$ packets scheduled to traverse the edges g, f. In g, the packets in S have priority over those in Z_1.

Round 2 ($|T_2| = r|T_1|$ time steps): The adversary injects a set Z_2 of $|Z_2| = r|T_2|$ packets in g scheduled to traverse the edges g, e, and a set Z_3 of $|Z_3| = r|T_2|$ packets in f scheduled to traverse f. The packets in Z_1 have priority over the packets of Z_2 in g and over the packets of Z_3 in f.
Round 3 ($|T_3| = r|T_2|$ time steps): The adversary injects a set Z_4 of $|Z_4| = r|T_3|$ packets in f scheduled to traverse f, and a set Z_5 of $|Z_5| = r|T_3|$ packets in e scheduled to traverse e, g. Thus at the end of this round, there is a set Y of $|Y| = r|T_3|$ packets in f waiting to traverse f. Additionally, in e, the packets in Z_2 have priority over those in Z_5.
Round 4 ($|T_4| = r|T_3|$ time steps): The adversary injects a set Z_6 of $|Z_6| = r|T_4|$ packets in e scheduled to traverse e, g, and a set Z_7 of $|Z_7| = r|T_4|$ packets in f scheduled to traverse f, g. The packets in Y delay those in Z_7, which remain waiting in f. Among the packets in $Z_5 \cup Z_6$ waiting in e, $|T_4|$ traverse e; thus, a set W of $|W| = |Z_5| + |Z_6| - |T_4|$ packets remain in e wanting to traverse e, g.
At the end of the four rounds, there are $|Z_7| + |W| = 2r|T_4|$ packets in the queues of e, f waiting to traverse e, g and f, g. The number of these packets exceeds $|S|$ if $2r^4 > 1$, i.e., $r \geq 0.841$.

System $\langle U_1, \mathcal{A}_2, (\mathsf{NTG}, \mathsf{FFS}, \mathsf{LIS})\rangle$.
Initially, there is a set S of packets waiting in the queues of e, f in order to traverse e, g and f, g, respectively. The adversary's strategy consists of four rounds of injections:
Round 1 ($|T_1| = |S|$ time steps): The adversary injects in g a set Z_1 of $|Z_1| = r|T_1|$ packets scheduled to traverse g, f. In g, the packets in S have priority over those in Z_1.
Round 2 ($|T_2| = r|T_1|$ time steps): The adversary injects a set Z_2 of $|Z_2| = r|T_2|$ packets in g scheduled to traverse g, e, and a set Z_3 of $|Z_3| = r|T_2|$ packets in f scheduled to traverse f. The packets of Z_1 have priority over those of Z_2 in g and over those of Z_3 in f; thus, the packets in Z_2 and Z_3 remain queued in e and f.
Round 3 ($|T_3| = r|T_2|$ time steps): The adversary injects a set Z_4 of $|Z_4| = r|T_3|$ packets in f scheduled to traverse f, and a set Z_5 of $|Z_5| = r|T_3|$ packets in e scheduled to traverse e, g. In f, the packets in Z_3 have priority over those in Z_4, while in e, the packets in Z_2 have priority over those in Z_5.
Round 4 ($|T_4| = r|T_3|$ time steps): The adversary injects a set Z_6 of $|Z_6| = r|T_4|$ packets in e scheduled to traverse e, and a set Z_7 of $|Z_7| = r|T_4|$ packets in f scheduled to traverse f, g. In f, the packets in Z_4 have priority over those in Z_7, while among the packets in $Z_5 \cup Z_6$ waiting in e, $|T_4|$ traverse it.
At the end of the four rounds, there are $|Z_5| + |Z_6| - |T_4|$ and $|Z_7|$ packets in the queues of e, f waiting to traverse e, g and f, g, respectively. The number of these packets exceeds $|S|$ if $2r^4 > 1$, i.e., $r \geq 0.841$.

C Proof of Theorem 3.1.3

System $\langle U_2, \mathcal{A}_1, (\mathsf{NTG}, \mathsf{LIS})\rangle$.
Initially, there is a set S of packets waiting in the queues of e_2, e_3 in order to traverse e_2, e_1 and e_3, e_4, e_1, respectively. The adversary's strategy consists of three rounds of injections:
Round 1 ($|T_1| = |S|$ time steps): The adversary injects in e_1 a set Z_1 of $|Z_1| = r|T_1|$ packets scheduled to traverse e_1, e_2, e_3. In e_1, the packets in S have priority over those in Z_1.
Round 2 ($|T_2| = r|T_1|$ time steps): The adversary injects a set Z_2 of $|Z_2| = r|T_2|$ packets in e_2 scheduled to traverse e_2, and a set Z_3 of $|Z_3| = r|T_2|$ packets in e_3 scheduled to traverse e_3, e_4, e_1. In e_2, the packets in Z_2 have priority over those in Z_1; a set X of $|Z_1| - (|T_2| - |Z_2|)$ packets remain in e_2 waiting to traverse e_2, e_3 and a set Y of $|T_2| - |Z_2|$ packets have priority over Z_3 packets in e_3. Thus, a set W of $|Z_3| + |Y| - |T_2|$ packets remain in e_3 scheduled to traverse e_3, e_4, e_1 at the end of this round.
Round 3 ($|T_3| = r|T_2|$ time steps): The adversary injects a set Z_4 of $|Z_4| = r|T_3|$ packets in e_2 scheduled to traverse e_2, e_1, and a set Z_5 of $|Z_5| = r|T_3|$ packets in e_3 scheduled to traverse e_3. X have priority over Z_4 packets in e_2. Thus, at the end of this round, there are $|X'| = |Z_4| + |X| - |T_3|$ packets in e_2 scheduled to traverse e_2, e_1. X and Z_5 packets have priority over W packets in e_3. Thus, $|Y'| = |X| + |W| + |Z_5| - |T_3|$ packets remain in e_3 requiring to traverse e_3, e_4, e_1.

At the end of the three rounds, there are $|X'| + |Y'|$ packets in the queues of e_2, e_3 waiting to traverse e_2, e_1 and e_3, e_4, e_1, respectively. The number of these packets exceeds $|S|$ if $2r^3 > 1$, i.e., $r \geq 0.794$.

System $\langle U_2, \mathcal{A}_2, (\mathsf{NTG}, \mathsf{FFS})\rangle$.
Initially, there is a set S of packets waiting in the queues of e_2, e_3 in order to traverse e_2, e_1 and e_3, e_4, e_1, respectively. The adversary's strategy consists of three rounds of injections as follows:
Round 1 ($|T_1| = |S|$ time steps): The adversary injects in e_1 a set Z_1 of $|Z_1| = r|T_1|$ packets scheduled to traverse e_1, e_2, e_3. In e_1, the packets in S have priority over those in Z_1.
Round 2 ($|T_2| = r|T_1|$ time steps): The adversary injects a set Z_2 of $|Z_2| = r|T_2|$ packets in e_2 scheduled to traverse e_2, and a set Z_3 of $|Z_3| = r|T_2|$ packets in e_3 scheduled to traverse e_3, e_4, e_1. In e_2, the packets in Z_2 have priority over those in Z_1; a set X of $|Z_1| - (|T_2| - |Z_2|)$ packets remain in e_2 waiting to traverse e_2, e_3 and a set Y of $|T_2| - |Z_2|$ packets have priority over Z_3 packets in e_3. Thus, a set W of $|Z_3| + |Y| - |T_2|$ packets remain in e_3 scheduled to traverse e_3, e_4, e_1 at the end of this round.
Round 3 ($|T_3| = r|T_2|$ time steps): The adversary injects a set Z_4 of $|Z_4| = r|T_3|$ packets in e_2 scheduled to traverse e_2, e_1, and a set Z_5 of $|Z_5| = r|T_3|$ packets in e_3 scheduled to traverse e_3. X have priority over Z_4 packets in e_2. Thus, at the end of this round, there are $|X'| = |Z_4| + |X| - |T_3|$ packets in e_2 scheduled to traverse e_2, e_1. X and Z_5 packets have priority over W packets in e_3. Thus, $|Y'| = |X| + |W| + |Z_5| - |T_3|$ packets remain in e_3 requiring to traverse e_3, e_4, e_1.
At the end of the three rounds, there are $|X'| + |Y'|$ packets in the queues of e_2, e_3 waiting to traverse e_2, e_1 and e_3, e_4, e_1, respectively. The number of these packets exceeds $|S|$ if $2r^3 > 1$, i.e., $r \geq 0.794$.

System $\langle U_2, \mathcal{A}_3, (\mathsf{NTG}, \mathsf{FFS}, \mathsf{LIS})\rangle$.
Initially, there is a set S of packets waiting in the queues of e_2, e_3 in order to traverse e_2, e_1 and e_3, e_4, e_1, respectively. The adversary's strategy consists of three rounds of injections as follows:
Round 1 ($|T_1| = |S|$ time steps): The adversary injects in e_1 a set Z_1 of $|Z_1| = r|T_1|$ packets scheduled to traverse e_1, e_2, e_3. In e_1, the packets in S have priority over those in Z_1.
Round 2 ($|T_2| = r|T_1|$ time steps): The adversary injects a set Z_2 of $|Z_2| = r|T_2|$ packets in e_2 scheduled to traverse e_2, and a set Z_3 of $|Z_3| = r|T_2|$ packets in e_3 scheduled to traverse e_3, e_4, e_1. In e_2, the packets in Z_2 have priority over those in Z_1; a set X of $|Z_1| - (|T_2| - |Z_2|)$ packets remain in e_2 waiting to traverse e_2, e_3 and a set Y of $|T_2| - |Z_2|$ packets have priority over Z_3 packets in e_3. Thus, a set W of $|Z_3| + |Y| - |T_2|$ packets remain in e_3 scheduled to traverse e_3, e_4, e_1 at the end of this round.
Round 3 ($|T_3| = r|T_2|$ time steps): The adversary injects a set Z_4 of $|Z_4| = r|T_3|$ packets in e_2 scheduled to traverse e_2, e_1, and a set Z_5 of $|Z_5| = r|T_3|$ packets in e_3 scheduled to traverse e_3. X have priority over Z_4 packets in e_2. Thus, at the end of this round, there are $|X'| = |Z_4| + |X| - |T_3|$ packets in e_2 scheduled to traverse e_2, e_1. X and Z_5 packets have priority over W packets in e_3. Thus, $|Y'| = |X| + |W| + |Z_5| - |T_3|$ packets remain in e_3 requiring to traverse e_3, e_4, e_1.
At the end of the three rounds, there are $|X'| + |Y'|$ packets in the queues of e_2, e_3 waiting to traverse e_2, e_1 and e_3, e_4, e_1, respectively. The number of these packets exceeds $|S|$ if $2r^3 > 1$, i.e., $r \geq 0.794$.

D Proof of Theorem 3.1.4

System $\langle S_1, \mathcal{A}_1, (\mathsf{NTG}, \mathsf{LIS})\rangle$.
Initially, there is a set S of packets waiting in the queues of e, f in order to traverse e, g_1 and f, g_1, respectively. The adversary's strategy consists of four rounds of injections as follows:
Round 1 ($|T_1| = |S|$ time steps): The adversary injects in g_1 a set Z_1 of $|Z_1| = r|T_1|$ packets scheduled to traverse g_1, g_2. In g_1, the packets in S have priority over those in Z_1.
Round 2 ($|T_2| = r|T_1|$ time steps): The adversary injects in g_2 a set Z_2 of $|Z_2| = r|T_2|$ packets scheduled to traverse g_2, e. In g_2, the packets in Z_1 have priority over those in Z_2.

Round 3 ($|T_3| = r|T_2|$ time steps): The adversary injects a set Z_3 of $|Z_3| = r|T_3|$ packets in g_2 scheduled to traverse g_2, f, and a set Z_4 of $|Z_4| = r|T_3|$ packets in e scheduled to traverse e, g_1. In g_2, the packets in Z_2 have priority over those in Z_3 and over the packets of Z_4 in e.
Round 4 ($|T_4| = r|T_3|$ time steps): The adversary injects a set Z_5 of $|Z_5| = r|T_4|$ packets in e scheduled to traverse e, and a set Z_6 of $|Z_6| = r|T_4|$ packets in f scheduled to traverse f, g_1. In f, the packets in Z_3 have priority over those in Z_6, while in e, the packets in Z_5 have priority over those in Z_4. Thus, at the end of this round, there are $|Y| = |Z_4| + |Z_5| - |T_4|$ packets in e scheduled to traverse e, g_1.
At the end of the four rounds, there are $|Z_6| + |Y|$ packets in the queues of e, f waiting to traverse e, g_1 and f, g_1, respectively. The number of these packets exceeds $|S|$ if $2r^4 > 1$, i.e., $r \geq 0.841$.

System $\langle S_2, \mathcal{A}_2, (\mathsf{NTG}, \mathsf{LIS})\rangle$.
Initially, there is a set S of packets waiting in the queues of e_2, f_2 in order to traverse e_2, g and f_2, g, respectively. The adversary's strategy consists of four rounds of injections as follows:
Round 1 ($|T_1| = |S|$ time steps): The adversary injects in g a set Z_1 of $|Z_1| = r|T_1|$ packets scheduled to traverse g, e_1. In g, the packets in S have priority over those in Z_1.
Round 2 ($|T_2| = r|T_1|$ time steps): The adversary injects a set Z_2 of $|Z_2| = r|T_2|$ packets in e_1 scheduled to traverse the edges e_1, e_2, and a set Z_3 of $|Z_3| = r|T_2|$ packets in g scheduled to traverse g, f_1. The packets in Z_1 have priority over the packets of Z_2 in e_1 and over the packets of Z_3 in g.
Round 3 ($|T_3| = r|T_2|$ time steps): The adversary injects a set Z_4 of $|Z_4| = r|T_3|$ packets in f_1 scheduled to traverse f_1, f_2, and a set Z_5 of $|Z_5| = r|T_3|$ packets in e_2 scheduled to traverse e_2, g. In e_2, the packets in Z_2 have priority over those in Z_5, while in f_1, the packets in Z_3 have priority over those in Z_4.
Round 4 ($|T_4| = r|T_3|$ time steps): The adversary injects a set Z_6 of $|Z_6| = r|T_4|$ packets in e_2 scheduled to traverse e_2, and a set Z_7 of $|Z_7| = r|T_4|$ packets in f_2 scheduled to traverse f_2, g. In f_2, the packets in Z_4 have priority over those in Z_7, while in e_2, the packets in Z_6 have priority over those in Z_5. Thus, at the end of this round, there are $|Y| = |Z_5| + |Z_6| - |T_4|$ packets in e_2 scheduled to traverse e_2, g.
At the end of the four rounds, there are $|Z_7| + |Y|$ packets in the queues of e_2, f_2 waiting to traverse e_2, g and f_2, g, respectively. The number of these packets exceeds $|S|$ if $2r^4 > 1$, i.e., $r \geq 0.841$.

System $\langle S_3, \mathcal{A}_3, (\mathsf{NTG}, \mathsf{LIS})\rangle$.
Initially, there is a set S of packets waiting in the queues of e_3, e_5 in order to traverse e_3, e_1 and e_5, e_6, e_1, respectively. The adversary's strategy consists of four rounds of injections as follows:
Round 1 ($|T_1| = |S|$ time steps): The adversary injects in e_1 a set Z_1 of $|Z_1| = r|T_1|$ packets scheduled to traverse e_1, e_2. In e_1, the packets in S have priority over those in Z_1.
Round 2 ($|T_2| = r|T_1|$ time steps): The adversary injects in e_2 a set Z_2 of $|Z_2| = r|T_2|$ packets scheduled to traverse e_2, e_3, e_4, e_5. In e_2, the packets in Z_1 have priority over those in Z_2.
Round 3 ($|T_3| = r|T_2|$ time steps): The adversary injects a set Z_3 of $|Z_3| = r|T_3|$ packets in e_2 scheduled to traverse e_2, e_3, and a set Z_4 of $|Z_4| = r|T_3|$ packets in e_5 scheduled to traverse e_5, e_6, e_1. In e_2, the packets in Z_2 have priority over those in Z_3 and over the packets of Z_4 in e_5.
Round 4 ($|T_4| = r|T_3|$ time steps): The adversary injects a set Z_5 of $|Z_5| = r|T_4|$ packets in e_3 scheduled to traverse e_3, e_1, and a set Z_6 of $|Z_6| = r|T_4|$ packets in e_5 scheduled to traverse e_5. In e_3, the packets in Z_3 have priority over those in Z_5, while in e_5, the packets in Z_6 have priority over those in Z_4. Thus, at the end of this round, there are $|Y| = |Z_4| + |Z_6| - |T_4|$ packets in e_5 scheduled to traverse e_5, e_6, e_1.
At the end of the four rounds, there are $|Z_5| + |Y|$ packets in the queues of e_3, e_5 waiting to traverse e_3, e_1 and e_5, e_6, e_1, respectively. The number of these packets exceeds $|S|$ if $2r^4 > 1$, i.e., $r \geq 0.841$.

System $\langle S_4, \mathcal{A}_4, (\mathsf{NTG}, \mathsf{LIS})\rangle$.
Initially, there is a set S of packets waiting in the queues of e_3, e_5 in order to traverse e_3, e_1 and e_5, e_6, g_1, e_1, respectively. The adversary's strategy consists of four rounds of injections as follows:
Round 1 ($|T_1| = |S|$ time steps): The adversary injects in e_1 a set Z_1 of $|Z_1| = r|T_1|$ packets scheduled to traverse e_1, e_2. In e_1, the packets in S have priority over those in Z_1.
Round 2 ($|T_2| = r|T_1|$ time steps): The adversary injects in e_2 a set Z_2 of $|Z_2| = r|T_2|$ packets scheduled to traverse e_2, e_3, g_2, e_4, e_5. In e_2, the packets in Z_1 have priority over those in Z_2.
Round 3 ($|T_3| = r|T_2|$ time steps): The adversary injects a set Z_3 of $|Z_3| = r|T_3|$ packets in e_2 scheduled to traverse e_2, e_3, and a set Z_4 of $|Z_4| = r|T_3|$ packets in e_5 scheduled to traverse e_5, e_6, g_1, e_1. In e_2, the packets in Z_2 have priority over those in Z_3 and over the packets of Z_4 in e_5.
Round 4 ($|T_4| = r|T_3|$ time steps): The adversary injects a set Z_5 of $|Z_5| = r|T_4|$ packets in e_3 scheduled to traverse e_3, e_1, and a set Z_6 of $|Z_6| = r|T_4|$ packets in e_5 scheduled to traverse e_5. In e_3, the packets in Z_3 have priority over those in Z_5, while in e_5, the packets in Z_6 have priority over those in Z_4. Thus, at the end of this round, there are $|Y| = |Z_4| + |Z_6| - |T_4|$ packets in e_5 scheduled to traverse e_5, e_6, g_1, e_1.
At the end of the four rounds, there are $|Z_5| + |Y|$ packets in the queues of e_3, e_5 waiting to traverse e_3, e_1 and e_5, e_6, g_1, e_1, respectively. The number of these packets exceeds $|S|$ if $2r^4 > 1$, i.e., $r \geq 0.841$.

System $\langle S_1, \mathcal{A}_1, (\mathsf{NTG}, \mathsf{FFS})\rangle$.
Initially, there is a set S of packets waiting in the queues of e, f in order to traverse e, g_1 and f, g_1, respectively. The adversary's strategy consists of four rounds of injections as follows:
Round 1 ($|T_1| = |S|$ time steps): The adversary injects in g_1 a set Z_1 of $|Z_1| = r|T_1|$ packets scheduled to traverse g_1, g_2. In g_1, the packets in S have priority over those in Z_1.
Round 2 ($|T_2| = r|T_1|$ time steps): The adversary injects in g_2 a set Z_2 of $|Z_2| = r|T_2|$ packets scheduled to traverse g_2, e. In g_2, the packets in Z_1 have priority over those in Z_2.
Round 3 ($|T_3| = r|T_2|$ time steps): The adversary injects a set Z_3 of $|Z_3| = r|T_3|$ packets in g_2 scheduled to traverse g_2, f, and a set Z_4 of $|Z_4| = r|T_3|$ packets in e scheduled to traverse e, g_1. In g_2, the packets in Z_2 have priority over those in Z_3 and over the packets of Z_4 in e.
Round 4 ($|T_4| = r|T_3|$ time steps): The adversary injects a set Z_5 of $|Z_5| = r|T_4|$ packets in e scheduled to traverse e, and a set Z_6 of $|Z_6| = r|T_4|$ packets in f scheduled to traverse f, g_1. In f, the packets in Z_3 have priority over those in Z_6, while in e, the packets in Z_5 have priority over those in Z_4. Thus, at the end of this round, there are $|Y| = |Z_4| + |Z_5| - |T_4|$ packets in e scheduled to traverse e, g_1.
At the end of the four rounds, there are $|Z_6| + |Y|$ packets in the queues of e, f waiting to traverse e, g_1 and f, g_1, respectively. The number of these packets exceeds $|S|$ if $2r^4 > 1$, i.e., $r \geq 0.841$.

System $\langle S_2, \mathcal{A}_2, (\mathsf{NTG}, \mathsf{FFS})\rangle$.
Initially, there is a set S of packets waiting in the queues of e_2, f_2 in order to traverse e_2, g and f_2, g, respectively. The adversary's strategy consists of four rounds of injections as follows:
Round 1 ($|T_1| = |S|$ time steps): The adversary injects in g a set Z_1 of $|Z_1| = r|T_1|$ packets scheduled to traverse g, e_1. In g, the packets in S have priority over those in Z_1.
Round 2 ($|T_2| = r|T_1|$ time steps): The adversary injects a set Z_2 of $|Z_2| = r|T_2|$ packets in e_1 scheduled to traverse the edges e_1, e_2, and a set Z_3 of $|Z_3| = r|T_2|$ packets in g scheduled to traverse g, f_1. The packets in Z_1 have priority over the packets of Z_2 in e_1 and over the packets of Z_3 in g.
Round 3 ($|T_3| = r|T_2|$ time steps): The adversary injects a set Z_4 of $|Z_4| = r|T_3|$ packets in f_1 scheduled to traverse f_1, f_2, and a set Z_5 of $|Z_5| = r|T_3|$ packets in e_2 scheduled to traverse e_2, g. In e_2, the packets in Z_2 have priority over those in Z_5, while in f_1, the packets in Z_3 have priority over those in Z_4.
Round 4 ($|T_4| = r|T_3|$ time steps): The adversary injects a set Z_6 of $|Z_6| = r|T_4|$ packets in e_2 scheduled to traverse e_2, and a set Z_7 of $|Z_7| = r|T_4|$ packets in f_2 scheduled to traverse f_2, g. In f_2, the packets in Z_4 have priority over those in Z_7, while in e_2, the packets in Z_6 have priority over those in

Z_5. Thus, at the end of this round, there are $|Y| = |Z_5| + |Z_6| - |T_4|$ packets in e_2 scheduled to traverse e_2, g.
At the end of the four rounds, there are $|Z_7| + |Y|$ packets in the queues of e_2, f_2 waiting to traverse e_2, g and f_2, g, respectively. The number of these packets exceeds $|S|$ if $2r^4 > 1$, i.e., $r \geq 0.841$.

System $\langle S_3, \mathcal{A}_3, (\mathsf{NTG}, \mathsf{FFS})\rangle$.
Initially, there is a set S of packets waiting in the queues of e_3, e_5 in order to traverse e_3, e_1 and e_5, e_6, e_1, respectively. The adversary's strategy consists of four rounds of injections as follows:
Round 1 ($|T_1| = |S|$ time steps): The adversary injects in e_1 a set Z_1 of $|Z_1| = r|T_1|$ packets scheduled to traverse e_1, e_2. In e_1, the packets in S have priority over those in Z_1.
Round 2 ($|T_2| = r|T_1|$ time steps): The adversary injects in e_2 a set Z_2 of $|Z_2| = r|T_2|$ packets scheduled to traverse e_2, e_3, e_4, e_5. In e_2, the packets in Z_1 have priority over those in Z_2.
Round 3 ($|T_3| = r|T_2|$ time steps): The adversary injects a set Z_3 of $|Z_3| = r|T_3|$ packets in e_2 scheduled to traverse e_2, e_3, and a set Z_4 of $|Z_4| = r|T_3|$ packets in e_5 scheduled to traverse e_5, e_6, e_1. In e_2, the packets in Z_2 have priority over those in Z_3 and over the packets of Z_4 in e_5.
Round 4 ($|T_4| = r|T_3|$ time steps): The adversary injects a set Z_5 of $|Z_5| = r|T_4|$ packets in e_3 scheduled to traverse e_3, e_1, and a set Z_6 of $|Z_6| = r|T_4|$ packets in e_5 scheduled to traverse e_5. In e_3, the packets in Z_3 have priority over those in Z_5, while in e_5, the packets in Z_6 have priority over those in Z_4. Thus, at the end of this round, there are $|Y| = |Z_4| + |Z_6| - |T_4|$ packets in e_5 scheduled to traverse e_5, e_6, e_1.
At the end of the four rounds, there are $|Z_5| + |Y|$ packets in the queues of e_3, e_5 waiting to traverse e_3, e_1 and e_5, e_6, e_1, respectively. The number of these packets exceeds $|S|$ if $2r^4 > 1$, i.e., $r \geq 0.841$.

System $\langle S_4, \mathcal{A}_4, (\mathsf{NTG}, \mathsf{FFS})\rangle$.
Initially, there is a set S of packets waiting in the queues of e_3, e_5 in order to traverse e_3, e_1 and e_5, e_6, g_1, e_1, respectively. The adversary's strategy consists of four rounds of injections as follows:
Round 1 ($|T_1| = |S|$ time steps): The adversary injects in e_1 a set Z_1 of $|Z_1| = r|T_1|$ packets scheduled to traverse e_1, e_2. In e_1, the packets in S have priority over those in Z_1.
Round 2 ($|T_2| = r|T_1|$ time steps): The adversary injects in e_2 a set Z_2 of $|Z_2| = r|T_2|$ packets scheduled to traverse e_2, e_3, g_2, e_4, e_5. In e_2, the packets in Z_1 have priority over those in Z_2.
Round 3 ($|T_3| = r|T_2|$ time steps): The adversary injects a set Z_3 of $|Z_3| = r|T_3|$ packets in e_2 scheduled to traverse e_2, e_3, and a set Z_4 of $|Z_4| = r|T_3|$ packets in e_5 scheduled to traverse e_5, e_6, g_1, e_1. In e_2, the packets in Z_2 have priority over those in Z_3 and over the packets of Z_4 in e_5.
Round 4 ($|T_4| = r|T_3|$ time steps): The adversary injects a set Z_5 of $|Z_5| = r|T_4|$ packets in e_3 scheduled to traverse e_3, e_1, and a set Z_6 of $|Z_6| = r|T_4|$ packets in e_5 scheduled to traverse e_5. In e_3, the packets in Z_3 have priority over those in Z_5, while in e_5, the packets in Z_6 have priority over those in Z_4. Thus, at the end of this round, there are $|Y| = |Z_4| + |Z_6| - |T_4|$ packets in e_5 scheduled to traverse e_5, e_6, g_1, e_1.
At the end of the four rounds, there are $|Z_5| + |Y|$ packets in the queues of e_3, e_5 waiting to traverse e_3, e_1 and e_5, e_6, g_1, e_1, respectively. The number of these packets exceeds $|S|$ if $2r^4 > 1$, i.e., $r \geq 0.841$.

System $\langle S_1, \mathcal{A}_1, (\mathsf{NTG}, \mathsf{FFS}, \mathsf{LIS})\rangle$.
Initially, there is a set S of packets waiting in the queues of e, f in order to traverse e, g_1 and f, g_1, respectively. The adversary's strategy consists of four rounds of injections as follows:
Round 1 ($|T_1| = |S|$ time steps): The adversary injects in g_1 a set Z_1 of $|Z_1| = r|T_1|$ packets scheduled to traverse g_1, g_2. In g_1, the packets in S have priority over those in Z_1.
Round 2 ($|T_2| = r|T_1|$ time steps): The adversary injects in g_2 a set Z_2 of $|Z_2| = r|T_2|$ packets scheduled to traverse g_2, e. In g_2, the packets in Z_1 have priority over those in Z_2.

Round 3 ($|T_3| = r|T_2|$ time steps): The adversary injects a set Z_3 of $|Z_3| = r|T_3|$ packets in g_2 scheduled to traverse g_2, f, and a set Z_4 of $|Z_4| = r|T_3|$ packets in e scheduled to traverse e, g_1. In g_2, the packets in Z_2 have priority over those in Z_3 and over the packets of Z_4 in e.
Round 4 ($|T_4| = r|T_3|$ time steps): The adversary injects a set Z_5 of $|Z_5| = r|T_4|$ packets in e scheduled to traverse e, and a set Z_6 of $|Z_6| = r|T_4|$ packets in f scheduled to traverse f, g_1. In f, the packets in Z_3 have priority over those in Z_6, while in e, the packets in Z_5 have priority over those in Z_4. Thus, at the end of this round, there are $|Y| = |Z_4| + |Z_5| - |T_4|$ packets in e scheduled to traverse e, g_1.
At the end of the four rounds, there are $|Z_6| + |Y|$ packets in the queues of e, f waiting to traverse e, g_1 and f, g_1, respectively. The number of these packets exceeds $|S|$ if $2r^4 > 1$, i.e., $r \geq 0.841$.

System $\langle S_2, \mathcal{A}_2, (\mathsf{NTG}, \mathsf{FFS}, \mathsf{LIS})\rangle$.
Initially, there is a set S of packets waiting in the queues of e_2, f_2 in order to traverse e_2, g and f_2, g, respectively. The adversary's strategy consists of four rounds of injections as follows:
Round 1 ($|T_1| = |S|$ time steps): The adversary injects in g a set Z_1 of $|Z_1| = r|T_1|$ packets scheduled to traverse g, e_1. In g, the packets in S have priority over those in Z_1.
Round 2 ($|T_2| = r|T_1|$ time steps): The adversary injects a set Z_2 of $|Z_2| = r|T_2|$ packets in e_1 scheduled to traverse the edges e_1, e_2, and a set Z_3 of $|Z_3| = r|T_2|$ packets in g scheduled to traverse g, f_1. The packets in Z_1 have priority over the packets of Z_2 in e_1 and over the packets of Z_3 in g.
Round 3 ($|T_3| = r|T_2|$ time steps): The adversary injects a set Z_4 of $|Z_4| = r|T_3|$ packets in f_1 scheduled to traverse f_1, f_2, and a set Z_5 of $|Z_5| = r|T_3|$ packets in e_2 scheduled to traverse e_2, g. In e_2, the packets in Z_2 have priority over those in Z_5, while in f_1, the packets in Z_3 have priority over those in Z_4.
Round 4 ($|T_4| = r|T_3|$ time steps): The adversary injects a set Z_6 of $|Z_6| = r|T_4|$ packets in e_2 scheduled to traverse e_2, and a set Z_7 of $|Z_7| = r|T_4|$ packets in f_2 scheduled to traverse f_2, g. In f_2, the packets in Z_4 have priority over those in Z_7, while in e_2, the packets in Z_6 have priority over those in Z_5. Thus, at the end of this round, there are $|Y| = |Z_5| + |Z_6| - |T_4|$ packets in e_2 scheduled to traverse e_2, g.
At the end of the four rounds, there are $|Z_7| + |Y|$ packets in the queues of e_2, f_2 waiting to traverse e_2, g and f_2, g, respectively. The number of these packets exceeds $|S|$ if $2r^4 > 1$, i.e., $r \geq 0.841$.

System $\langle S_3, \mathcal{A}_3, (\mathsf{NTG}, \mathsf{FFS}, \mathsf{LIS})\rangle$.
Initially, there is a set S of packets waiting in the queues of e_3, e_5 in order to traverse e_3, e_1 and e_5, e_6, e_1, respectively. The adversary's strategy consists of four rounds of injections as follows:
Round 1 ($|T_1| = |S|$ time steps): The adversary injects in e_1 a set Z_1 of $|Z_1| = r|T_1|$ packets scheduled to traverse e_1, e_2. In e_1, the packets in S have priority over those in Z_1.
Round 2 ($|T_2| = r|T_1|$ time steps): The adversary injects in e_2 a set Z_2 of $|Z_2| = r|T_2|$ packets scheduled to traverse e_2, e_3, e_4, e_5. In e_2, the packets in Z_1 have priority over those in Z_2.
Round 3 ($|T_3| = r|T_2|$ time steps): The adversary injects a set Z_3 of $|Z_3| = r|T_3|$ packets in e_2 scheduled to traverse e_2, e_3, and a set Z_4 of $|Z_4| = r|T_3|$ packets in e_5 scheduled to traverse e_5, e_6, e_1. In e_2, the packets in Z_2 have priority over those in Z_3 and over the packets of Z_4 in e_5.
Round 4 ($|T_4| = r|T_3|$ time steps): The adversary injects a set Z_5 of $|Z_5| = r|T_4|$ packets in e_3 scheduled to traverse e_3, e_1, and a set Z_6 of $|Z_6| = r|T_4|$ packets in e_5 scheduled to traverse e_5. In e_3, the packets in Z_3 have priority over those in Z_5, while in e_5, the packets in Z_6 have priority over those in Z_4. Thus, at the end of this round, there are $|Y| = |Z_4| + |Z_6| - |T_4|$ packets in e_5 scheduled to traverse e_5, e_6, e_1.
At the end of the four rounds, there are $|Z_5| + |Y|$ packets in the queues of e_3, e_5 waiting to traverse e_3, e_1 and e_5, e_6, e_1, respectively. The number of these packets exceeds $|S|$ if $2r^4 > 1$, i.e., $r \geq 0.841$.

System $\langle S_4, \mathcal{A}_4, (\mathsf{NTG}, \mathsf{FFS}, \mathsf{LIS})\rangle$.

Initially, there is a set S of packets waiting in the queues of e_3, e_5 in order to traverse e_3, e_1 and e_5, e_6, g_1, e_1, respectively. The adversary's strategy consists of four rounds of injections as follows:

Round 1 ($|T_1| = |S|$ time steps): The adversary injects in e_1 a set Z_1 of $|Z_1| = r|T_1|$ packets scheduled to traverse e_1, e_2. In e_1, the packets in S have priority over those in Z_1.

Round 2 ($|T_2| = r|T_1|$ time steps): The adversary injects in e_2 a set Z_2 of $|Z_2| = r|T_2|$ packets scheduled to traverse e_2, e_3, g_2, e_4, e_5. In e_2, the packets in Z_1 have priority over those in Z_2.

Round 3 ($|T_3| = r|T_2|$ time steps): The adversary injects a set Z_3 of $|Z_3| = r|T_3|$ packets in e_2 scheduled to traverse e_2, e_3, and a set Z_4 of $|Z_4| = r|T_3|$ packets in e_5 scheduled to traverse e_5, e_6, g_1, e_1. In e_2, the packets in Z_2 have priority over those in Z_3 and over the packets of Z_4 in e_5.

Round 4 ($|T_4| = r|T_3|$ time steps): The adversary injects a set Z_5 of $|Z_5| = r|T_4|$ packets in e_3 scheduled to traverse e_3, e_1, and a set Z_6 of $|Z_6| = r|T_4|$ packets in e_5 scheduled to traverse e_5. In e_3, the packets in Z_3 have priority over those in Z_5, while in e_5, the packets in Z_6 have priority over those in Z_4. Thus, at the end of this round, there are $|Y| = |Z_4| + |Z_6| - |T_4|$ packets in e_5 scheduled to traverse e_5, e_6, g_1, e_1.

At the end of the four rounds, there are $|Z_5| + |Y|$ packets in the queues of e_3, e_5 waiting to traverse e_3, e_1 and e_5, e_6, g_1, e_1, respectively. The number of these packets exceeds $|S|$ if $2r^4 > 1$, i.e., $r \geq 0.841$.

E Proof of Theorem 3.2.2

System $\langle U_2, \mathcal{A}_2, (\mathsf{NTG}, \mathsf{FIFO})\rangle$.

Initially, there is a set S of packets waiting in the queues of e_2, e_3 in order to traverse e_2, e_1 and e_3, e_4, e_1, respectively. The adversary's strategy consists of three rounds of injections as follows:

Round 1 ($|T_1| = |S|$ time steps): The adversary injects in e_1 a set Z_1 of $|Z_1| = r|T_1|$ packets scheduled to traverse e_1, e_2, e_3. In e_1, the packets in S have priority over those in Z_1.

Round 2 ($|T_2| = r|T_1|$ time steps): The adversary injects a set Z_2 of $|Z_2| = r|T_2|$ packets in e_2 scheduled to traverse e_2, and a set Z_3 of $|Z_3| = r|T_2|$ packets in e_3 scheduled to traverse e_3, e_4, e_1. The packets in Z_1 and the packets in Z_2 arrive at e_2 together; a set X_1 of $|Z_2| - \frac{|Z_2|}{|Z_1|+|Z_2|}|T_2|$ packets remain in e_2 waiting to traverse e_2, a set X_2 of $|Z_1| - \frac{|Z_1|}{|Z_1|+|Z_2|}|T_2|$ packets remain in e_2 waiting to traverse e_2, e_1 and a set Y of $\frac{|Z_1|}{|Z_1|+|Z_2|}|T_2|$ packets have priority over Z_3 packets in e_3. Thus, a set W of $|Z_3| + |Y| - |T_2|$ packets remain in e_3 scheduled to traverse e_3, e_4, e_1 at the end of this round.

Round 3 ($|T_3| = r|T_2|$ time steps): The adversary injects a set Z_4 of $|Z_4| = r|T_3|$ packets in e_2 scheduled to traverse e_2, e_1, and a set Z_5 of $|Z_5| = r|T_3|$ packets in e_3 scheduled to traverse e_3. X_1 and X_2 packets have priority over Z_4 packets in e_2. X_2 and Z_5 packets have priority over W packets in e_3.

At the end of the three rounds, there are $|W| + |Z_4|$ packets in the queues of e_2, e_3 waiting to traverse e_2, e_1 and e_3, e_4, e_1, respectively. The number of these packets exceeds $|S|$ if $r^3 \frac{2+r}{1+r} > 1$, i.e., $r \geq 0.867$.

F Proof of Theorem 3.2.3

System $\langle S_1, \mathcal{A}_1, (\mathsf{NTG}, \mathsf{FIFO})\rangle$.

Initially, there is a set S of packets waiting in the queues of e, f in order to traverse e, g_1 and f, g_1, respectively. The adversary's strategy consists of four rounds of injections as follows:

Round 1 ($|T_1| = |S|$ time steps): The adversary injects in g_1 a set Z_1 of $|Z_1| = r|T_1|$ packets scheduled to traverse g_1, g_2. In g_1, the packets in S have priority over those in Z_1.

Round 2 ($|T_2| = r|T_1|$ time steps): The adversary injects in g_2 a set Z_2 of $|Z_2| = r|T_2|$ packets scheduled to traverse g_2, e. In g_2, the packets in Z_1 have priority over those in Z_2.

Round 3 ($|T_3| = r|T_2|$ time steps): The adversary injects a set Z_3 of $|Z_3| = r|T_3|$ packets in e scheduled to traverse e, g_1, and a set Z_4 of $|Z_4| = r|T_3|$ packets in g_2 scheduled to traverse g_2, f. In e, the packets

in Z_2 have priority over those in Z_3 and over the packets of Z_4 in g_2.
Round 4 ($|T_4| = r|T_3|$ time steps): The adversary injects a set Z_5 of $|Z_5| = r|T_4|$ packets in e scheduled to traverse e, and a set Z_6 of $|Z_6| = r|T_4|$ packets in f scheduled to traverse f, g_1. In e, the packets in Z_5 have priority over those in Z_3. Thus, at the end of this round, a set X of $r|T_4|$ packets remain in e scheduled to traverse e, g_1 and a set Y of $|Z_6| - \frac{|Z_6|}{|Z_4|+|Z_6|}|T_4|$ packets remain in f scheduled to traverse f, g_1.
At the end of the four rounds, there are $|X| + |Y|$ packets in the queues of e, f waiting to traverse e, g_1 and f, g_1, respectively. The number of these packets exceeds $|S|$ if $r^4 \frac{1+2r}{1+r} > 1$, i.e., $r \geq 0.908$.

System $\langle S_2, \mathcal{A}_2, (\mathsf{NTG}, \mathsf{FIFO})\rangle$.
Initially, there is a set S of packets waiting in the queues of e_2, f_2 in order to traverse e_2, g and f_2, g, respectively. The adversary's strategy consists of four rounds of injections as follows:
Round 1 ($|T_1| = |S|$ time steps): The adversary injects in g a set Z_1 of $|Z_1| = r|T_1|$ packets scheduled to traverse g, e_1. In g, the packets in S have priority over those in Z_1.
Round 2 ($|T_2| = r|T_1|$ time steps): The adversary injects a set Z_2 of $|Z_2| = r|T_2|$ packets in e_1 scheduled to traverse the edges e_1, e_2, and a set Z_3 of $|Z_3| = r|T_2|$ packets in g scheduled to traverse g, f_1. The packets in Z_1 have priority over the packets of Z_2 in e_1 and over the packets of Z_3 in g.
Round 3 ($|T_3| = r|T_2|$ time steps): The adversary injects a set Z_4 of $|Z_4| = r|T_3|$ packets in e_2 scheduled to traverse e_2, g, and a set Z_5 of $|Z_5| = r|T_3|$ packets in f_1 scheduled to traverse f_1, f_2. In e_2, the packets in Z_2 have priority over those in Z_4, while in f_1, the packets in Z_3 have priority over those in Z_5.
Round 4 ($|T_4| = r|T_3|$ time steps): The adversary injects a set Z_6 of $|Z_6| = r|T_4|$ packets in e_2 scheduled to traverse e_2, and a set Z_7 of $|Z_7| = r|T_4|$ packets in f_2 scheduled to traverse f_2, g_1. In e_2, the packets in Z_6 have priority over those in Z_4. Thus, at the end of this round, a set X of $r|T_4|$ packets remain in e_2 scheduled to traverse e_2, g and a set Y of $|Z_7| - \frac{|Z_7|}{|Z_5|+|Z_7|}|T_4|$ packets remain in f_2 scheduled to traverse f_2, g_1.
At the end of the four rounds, there are $|X| + |Y|$ packets in the queues of e_2, f_2 waiting to traverse e_2, g and f_2, g, respectively. The number of these packets exceeds $|S|$ if $r^4 \frac{1+2r}{1+r} > 1$, i.e., $r \geq 0.908$.

G Proof of Theorem 3.2.4

System $\langle S_3, \mathcal{A}_3, (\mathsf{NTG}, \mathsf{FIFO})\rangle$.
Initially, there is a set S of packets waiting in the queues of e_3, e_5 in order to traverse e_3, e_1 and e_5, e_6, e_1, respectively. The adversary's strategy consists of four rounds of injections as follows:
Round 1 ($|T_1| = |S|$ time steps): The adversary injects in e_1 a set Z_1 of $|Z_1| = r|T_1|$ packets scheduled to traverse e_1, e_2. In e_1, the packets in S have priority over those in Z_1.
Round 2 ($|T_2| = r|T_1|$ time steps): The adversary injects in e_2 a set Z_2 of $|Z_2| = r|T_2|$ packets scheduled to traverse e_2, e_3, e_4, e_5. In e_2, the packets in Z_1 have priority over those in Z_2.
Round 3 ($|T_3| = r|T_2|$ time steps): The adversary injects a set Z_3 of $|Z_3| = r|T_3|$ packets in e_3 scheduled to traverse e_3, and a set Z_4 of $|Z_4| = r|T_3|$ packets in e_5 scheduled to traverse e_5, e_6, e_1. The packets in Z_2 and the packets in Z_3 arrive at e_3 together; a set X_1 of $|Z_3| - \frac{|Z_3|}{|Z_2|+|Z_3|}|T_3|$ packets remain in e_3 waiting to traverse e_3 and a set X_2 of $|Z_2| - \frac{|Z_2|}{|Z_2|+|Z_3|}|T_3|$ packets remain in e_3 waiting to traverse e_3, e_4, e_5. A set Y of $\frac{|Z_2|}{|Z_2|+|Z_3|}|T_3|$ packets have priority over Z_4 packets in e_5. Thus, a set W of $|Z_4| + |Y| - |T_3|$ packets remain in e_5 scheduled to traverse e_5, e_6, e_1 at the end of this round.
Round 4 ($|T_4| = r|T_3|$ time steps): The adversary injects a set Z_5 of $|Z_5| = r|T_4|$ packets in e_3 scheduled to traverse e_3, e_1, and a set Z_6 of $|Z_6| = r|T_4|$ packets in e_5 scheduled to traverse e_5. X_1 and X_2 packets have priority over Z_5 packets in e_3. X_2 and Z_6 packets have priority over W packets in e_5.
At the end of the four rounds, there are $|W| + |Z_5|$ packets in the queues of e_3, e_5 waiting to traverse e_3, e_1 and e_5, e_6, e_1, respectively. The number of these packets exceeds $|S|$ if $r^4 \frac{2+r}{1+r} > 1$, i.e., $r \geq 0.9$.

System $\langle S_4, \mathcal{A}_4, (\mathsf{NTG}, \mathsf{FIFO})\rangle$.
Initially, there is a set S of packets waiting in the queues of e_3, e_5 in order to traverse e_3, e_1 and e_5, e_6, g_1, e_1, respectively. The adversary's strategy consists of four rounds of injections as follows:
Round 1 ($|T_1| = |S|$ time steps): The adversary injects in e_1 a set Z_1 of $|Z_1| = r|T_1|$ packets scheduled to traverse e_1, e_2. In e_1, the packets in S have priority over those in Z_1.
Round 2 ($|T_2| = r|T_1|$ time steps): The adversary injects in e_2 a set Z_2 of $|Z_2| = r|T_2|$ packets scheduled to traverse e_2, e_3, g_2, e_4, e_5. In e_2, the packets in Z_1 have priority over those in Z_2.
Round 3 ($|T_3| = r|T_2|$ time steps): The adversary injects a set Z_3 of $|Z_3| = r|T_3|$ packets in e_3 scheduled to traverse e_3, and a set Z_4 of $|Z_4| = r|T_3|$ packets in e_5 scheduled to traverse e_5, e_6, g_1, e_1. The packets in Z_2 and the packets in Z_3 arrive at e_3 together; a set X_1 of $|Z_3| - \frac{|Z_3|}{|Z_2|+|Z_3|}|T_3|$ packets remain in e_3 waiting to traverse e_3 and a set X_2 of $|Z_2| - \frac{|Z_2|}{|Z_2|+|Z_3|}|T_3|$ packets remain in e_3 waiting to traverse e_3, g_2, e_4, e_5. A set Y of $\frac{|Z_2|}{|Z_2|+|Z_3|}|T_3|$ packets have priority over Z_4 packets in e_5. Thus, a set W of $|Z_4| + |Y| - |T_3|$ packets remain in e_5 scheduled to traverse e_5, e_6, g_1, e_1 at the end of this round.
Round 4 ($|T_4| = r|T_3|$ time steps): The adversary injects a set Z_5 of $|Z_5| = r|T_4|$ packets in e_3 scheduled to traverse e_3, e_1, and a set Z_6 of $|Z_6| = r|T_4|$ packets in e_5 scheduled to traverse e_5. X_1 and X_2 packets have priority over Z_5 packets in e_3. X_2 and Z_6 packets have priority over W packets in e_5. At the end of the four rounds, there are $|W| + |Z_5|$ packets in the queues of e_3, e_5 waiting to traverse e_3, e_1 and e_5, e_6, g_1, e_1, respectively. The number of these packets exceeds $|S|$ if $r^4 \frac{2+r}{1+r} > 1$, i.e., $r \geq 0.9$.