Instability of Networks with Quasi-Static Link Capacities*

DIMITRIOS KOUKOPOULOS
University of Patras and RACTI, Greece

MARIOS MAVRONICOLAS
University of Cyprus, Cyprus

PAUL SPIRAKIS
University of Patras and RACTI, Greece

Abstract

In this work, we continue the study of *stability* issues for packet-switched routing. More specifically, we adopt the *Adversarial Queueing Theory* framework, where an adversary controls rates of packet injections and determines packet paths. In addition, the power of the adversary is enhanced to include manipulation of *link capacities*. However, in doing so, the adversary may use only two possible (integer) values, namely 1 and $C > 1$; moreover, the capacity changes are not abrupt: once a link capacity is set to a value, it maintains this value for a period of time proportional to the number of packets in the system at the time of setting the link capacity to the value. We call this the *Adversarial, Quasi-Static Queueing Theory* model. Within this model, we obtain the following results:

- The protocol LIS (*Longest-in-System*) is unstable at rates $r > \sqrt{2} - 1$ for large enough values of C. The proof uses a small network of just ten nodes.

  This represents the current record for the instability threshold of LIS over models of Adversarial Queueing Theory with dynamic capacities.
- The composition of LIS with any of SIS (*Shortest-in-System*), NTS (*Nearest-to-Source*) and FTG (*Furthest-to-Go*) is unstable at rates $r > \sqrt{2} - 1$ for large enough values of C.

  These represent the first results on the instability thresholds of compositions of greedy protocols over models of Adversarial Queueing Theory with dynamic capacities.

---

*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) and 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").

- We present bounds for the instability thresholds of all *directed subgraphs* that are *forbidden* for stability.

  These bounds are lower than their counterparts for the classical Adversarial Queueing Theory model.

Keywords

adversarial queueing theory, stability, routing

1 Introduction

Motivation-Framework. We are interested in the behavior of *packet-switched networks* in which packets arrive dynamically at the *nodes* and they are routed in discrete time steps across the *links*. Recent years have witnessed a vast amount of work on analyzing packet-switched networks under *non-probabilistic* assumptions (rather than stochastic ones); We work within a model of *worst-case* continuous packet arrivals, originally proposed by Borodin *et al.* [4] and termed *Adversarial Queueing Theory* to reflect the assumption of an *adversarial* way of packet generation and path determination.

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

Most studies of packet-switched networks assume that one packet can cross a network link (an edge) in a single time step. This assumption is well motivated when we assume that all network links are identical. However, a packet-switched network can contain different types of links, which is common especially in large-scale networks like Internet. Then, it is well motivated to assign a capacity to each link. Furthermore, if each link capacity takes on values in the two-valued set of integers $\{1, C\}$ for $C > 1$, C takes on large values and each value stays for a long time, then we can consider approximately as a link failure the assigning of unit capacity to a link, while the assigning of capacity C to a link can be considered as the proper service rate. Therefore, the study of stability behavior of networks and protocols under our model of quasi-static capacities can be considered as an approximation of the fault-tolerance of a network where links can temporarily fail (zero capacity).

In this work we consider the impact on stability behavior of protocols and networks if the adversary besides the packet injections in paths which it determines, it also can set the capacities of network edges in each time step. This subfield of study was initiated by Borodin *et al.* in [5]. Note that we continue to assume uniform packet sizes. Furthermore, we consider greedy contention-resolution protocols- always advance a packet across a queue (but one packet at each discrete time step) whenever there resides at least one packet in the queue. The protocol specifies which packet will be chosen. We study five greedy protocols: LIS (*Longest-in-System*) gives priority to the packets that have been for the longest amount of time in the network; SIS (*Shortest-in-System*) gives priority to the packets that have been for the shortest amount of time in the network; FTG (*Furthest-to-Go*) gives priority to the packet that has the maximum number of edges in its prescribed path still to be traversed; NTS (*Nearest-to-Source*) gives priority to the packet that has traversed the minimum number of edges in its prescribed path; NTG-U-LIS (*Nearest-To-Go-Using-Longest-In-System*) gives priority to the packet that has the minimum number of edges in its prescribed path still to be traversed, while it gives priority to the packet that has been for the longest amount of time in the network for tie-breaking.

Roughly speaking a protocol P is *stable* [4] on a network G against an adversary $\mathcal{A}$ of rate r if there is an integer B (which may depend on $\mathcal{G}$ and $\mathcal{A}$) such that the number of packets in the system is bounded at all times by B. We say that *a protocol* P *is universally stable* [4] if it is stable against every adversary of rate less than 1 on every network. Also, *a network $\mathcal{G}$ is universally stable* [4] if every greedy protocol is stable against every adversary of rate less than 1 on $\mathcal{G}$.

Contribution. We define here the *weakest* possible adversary of dynamically changing *capacities* of network *links* in the context of Adversarial Queueing Theory where the adversary may set link capacities to any of two integer values 1 and C ($C > 1$ is a parameter called high capacity).* Moreover, once a link changes its capacity value, the model we use requires that the value stays fixed for at least a constant proportion of the number of packets in the system at the time when the capacity was last set. We call this the *Adversarial, Quasi-Static Queueing Theory* model. In this framework, we consider four protocols LIS, SIS, FTG, NTS; all four were shown universally stable in the standard model of Adversarial Queueing Theory.

- We construct a simple LIS network of only 10 nodes that is unstable at rates $r > \sqrt{2} - 1$ for large enough values of C (Theorem 3.1). This result is the first one that presents an instability threshold less than $\frac{1}{2}$ for a small-size network. Till now instability thresholds of $\frac{1}{2}$ or less have been proved only on parameterized networks. To show this, we use an adversarial construction that sets properly the capacities of various networks edges to 1 for specified time intervals in order to accumulate packets.
- We then consider networks where different protocols may run on their

*In the classical Adversarial Queueing Theory only one capacity value is available to the adversary.

nodes (heterogeneous networks, Internet). Thus, we prove that the composition of LIS with any of SIS, NTS and FTG is unstable at rates $r > \sqrt{2} - 1$ (for large enough values of C) (Theorem 4.1). To show this, we provide interesting combinatorial constructions of networks, for each queue of which we specify the contention-resolution protocol to be used.

- Finally, we examine the impact on network stability of dynamically changing network link capacities presenting bounds for the instability thresholds of all the *directed subgraphs* that are *forbidden* for stability. Through improved combinatorial constructions of networks and executions we present adversarial constructions that improve the state-of-the-art instability threshold induced by certain known forbidden subgraphs on networks running a certain greedy protocol (Theorems 5.1, 5.2). More specifically we improve the instability threshold of the six simple subgraphs in Figure 3 that have been proved in [2] to be forbidden subgraphs for the universal stability of networks.

Related Work. *Stability Issues under the Adversarial Queueing Model.* Adversarial Queueing Theory was developed by Borodin *et al.* [4] as a more realistic model that replaces traditional stochastic assumptions in Queueing Theory [6] by more robust, worst-case ones. It received a lot of interest in the study of stability and instability issues (see, e.g., [1, 7, 8, 9, 11, 12]). The universal stability of various natural greedy protocols (SIS, NTS, FTG, LIS) has been established by Andrews *et al.* [1]. Also, several greedy protocols such as Nearest-To-Go (NTG) have been proved unstable at arbitrary small rates of injection in [12]. The subfield of study of the stability properties of compositions of universally stable protocols was introduced by *Koukopoulos et al.* in [10, 8] where lower bounds of 0.683 and 0.519 on the instability threshold of the composition pairs LIS-SIS, LIS-NTS and LIS-FTG were presented.
Instability of Forbidden Subgraphs. In [2, Theorems 1, 2], a characterization for directed networks universal stability is given when the packets follow simple paths (paths do not contain repeated edges) that are pure (simple paths do not contain repeated vertices) and simple paths that are not pure (simple paths contain repeated vertices).[†] According to this characterization, a directed network graph is not pure simple path universally stable if and only if it does not contain as subgraphs any of the extensions of the subgraphs $\mathcal{U}_1$ or $\mathcal{U}_2$ [2, Theorem 1]; it is pure simple path universally stable if and only if it does not contain as subgraphs any of the extensions of the subgraphs $\mathcal{S}_1$ or $\mathcal{S}_2$ or $\mathcal{S}_3$ or $\mathcal{S}_4$ [2, Theorem 1] (see Figure 3). For purpose of completeness and comparison, we summarize, in Table 1, all results shown in this work and in [2] that provide instability bounds on the injection rate for the forbidden subgraphs ($\mathcal{S}_1$, $\mathcal{S}_2$, $\mathcal{S}_3$, $\mathcal{S}_4$, $\mathcal{U}_1$, $\mathcal{U}_2$).

[†]Corresponding characterization for the stability of undirected networks was shown in [1, Theorem 3.16].

	Apply to:	Instability (AQM)	Instability (AQSQM)
$\mathcal{S}_1$	pure s.p.	$r > 0.87055$ [2, Lemma 8]	$r > 0.8191$ (Thm. 5.1)
$\mathcal{S}_2$	pure s.p.	$r > 0.84089$ [2, Lemma 9]	$r > 0.8191$ (Thm. 5.1)
$\mathcal{S}_3$	pure s.p.	$r > 0.84089$ [2, Lemma 10]	$r > 0.8191$ (Thm. 5.1)
$\mathcal{S}_4$	pure s.p.	$r > 0.84089$ [2, Lemma 11]	$r > 0.8191$ (Thm. 5.1)
$\mathcal{U}_1$	not pure s.p.	$r > 0.84089$ [2, Lemma 7]	$r > 0.794$ (Thm. 5.2)
$\mathcal{U}_2$	not pure s.p.	$r > 0.84089$ [2, Lemma 7]	$r > 0.755$ (Thm. 5.2)

Table 1: Instability thresholds of forbidden subgraphs in AQM vs. AQSQM. We denote **AQM** the Adversarial Queueing Theory Model, **AQSQM** the Adversarial Quasi-Static Queueing Theory Model and *s.p.* the simple path.

Stability Issues in Dynamic Networks. Borodin *et al.* in [5] studied for the first time the impact on stability when the edges in a network can have capacities and slowdowns. They proved that the universal stability of networks is preserved under this varying context. Also, it was shown that many well-known universally stable protocols (SIS, NTS, FTG) do maintain their universal stability when either the link capacity or slowdown is changing dynamically, whereas the universal stability of LIS is not preserved. More specifically Borodin *et al.* in [5, Theorem 1] presented for the first time an instability threshold of $r > \frac{C}{2C-1} > 0.5$ for LIS protocol.

Road Map. The rest of this paper is organized as follows. Section 2 presents model definitions. Section 3 presents our instability bound for LIS. Section 4 demonstrates instability bounds for protocol compositions. Section 5 shows instability bounds for forbidden subgraphs. We conclude, in Section 6, with a discussion of our results and some open problems.

2 Model

The model definitions are patterned after those in [4, Section 3], adjusted to reflect the fact that edge capacities may vary arbitrarily as in [5, Section 2], but we address the weakest possible model of changing capacities. A *routing network* is a directed graph with *nodes* and *edges*. Time proceeds in discrete steps. A *packet* is an atomic entity that resides at a node at the end of any step. It must travel along paths in the network from its *source* to its *destination*, both of which are nodes in the network. When it reaches its destination, we say that it is *absorbed*. During each step, a packet may be sent from its current node along one of the outgoing edges from that node. Edges can have different integer capacities, which may or may not vary over time. Denote $C_e(t)$ the *capacity* of edge e at time step t. That is, we assume that edge e is capable of simultaneously transmitting up to $C_e(t)$

packets at time t.

Let $C > 1$ be an integer parameter. We demand that $\forall e$ and $\forall t$ $C_e(t) \in \{1, C\}$ (i.e. each edge capacity can get only two values, high and low). We also demand for each edge e that $C_e(t)$ stays at some value for a continuous period of time, at least proportional to the number of packets in the system at the time when the capacity was last set. We call this the *Adversarial, Quasi-Static Queueing Theory Model.* This model is the weakest possible of models implied by [5].

Any packets that wish to travel along an edge e at a particular time step but are not sent wait in a queue for edge e. The *delay* of a packet is the number of steps spent by the packet while waiting in queues. At each step, an em adversary generates a set of requests. A *request* is a *path* specifying the route followed by a packet.[‡] We say that the adversary generates a set of packets when it generates a set of requested paths. We restrict our study to the case of *non-adaptive* routing, where the path traversed by each packet is fixed at the time of injection, so that we are able to focus on queueing rather than routing aspects of the problem. (See [3] for an extension of the adversarial model to the case of *adaptive* routing.) There are no computational restrictions on how the adversary chooses its requests in any given time step.

Fix any arbitrary positive integer $w \geq 1$. For any edge e of the network and any sequence of w consecutive time steps, define $N(w,e)$ to be the number of paths injected by the adversary during the time interval of w consecutive time steps that traverse edge e. For any constant r, $0 < r \leq 1$, a (w,r)*-adversary* is an adversary that injects packets subject to the following *load condition*: For every edge e and for every sequence τ of w consecutive time steps, $N(\tau, e) \leq r \sum_{t \in \tau} C_e(t)$. We say that a (w,r)-adversary injects packets at rate r with *window size* w. The assumption that $r \leq 1$ ensures that it is not necessary a priori that some edge of the network is congested (that happens when $r > 1$).

In the adversarial constructions we study here for proving instability, we split time into *phases*. In each phase, we study the evolution of the *system configuration* by considering corresponding *time rounds*. For each phase, we inductively prove that the number of packets of a specific subset of queues in the system increases in order to guarantee instability. This inductive argument can be applied repeatedly, thus showing instability. Furthermore, 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]. For simplicity, and in a way similar to that in [1] and in works following it, we omit floors and ceilings from our analysis, and we sometimes count time steps and packets only roughly. This may only result to loosing small additive constants, while it implies a gain in clarity.

[‡]In this work, it is assumed, as it is common in packet routing, that all such paths are simple paths with no overlapping edges. However, in Section 5 we consider two different kinds of simple paths: simple paths that do not contain repeated vertices (pure simple paths) and simple paths that contain repeated vertices (not pure simple paths).

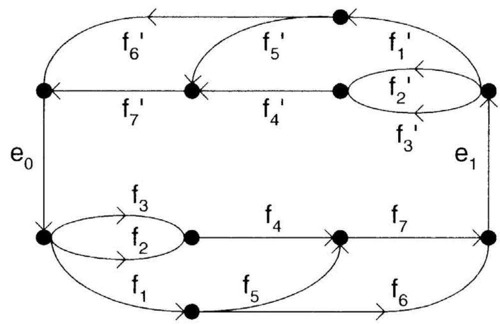

Figure 1: The network $\mathcal{N}$.

3 Instability Bound for LIS

Theorem 1 *Let $r > \sqrt{2} - 1$. There is a network $\mathcal{N}$ and an adversary $\mathcal{A}$ of rate r such that the system $\langle \mathcal{N}, \mathcal{A}, \mathsf{LIS} \rangle$ is unstable in the quasi-static model of capacities.*

Proof. Consider the network $\mathcal{N}$ in Figure 1. The construction of the adversary $\mathcal{A}$ is broken into phases.

Inductive Hypothesis: At the beginning of phase j (suppose j is even), there are s_j packets that are queued in the queues f_1', f_4', f_5', f_7' (in total) requiring to traverse the edges e_0, f_2, f_4.

Induction Step: At the beginning of phase $j+1$ there will be more than s_j packets (s_{j+1} packets) that are queued in the queues f_1, f_4, f_5, f_7 (in total) requiring to traverse the edges e_1, f_2', f_4'.

We will construct an adversary $\mathcal{A}$ such that the induction step holds. The main ideas of the construction of $\mathcal{A}$ are (a) the careful tuning of the duration of each round of every phase j (as a function of the high capacity C, the injection rate r and the number of packets in the system at the beginning of phase j, s_j) to maximize the growth of the packet population in the system and, (b) the careful setting of the capacities of some edges to 1 for specified time intervals in order to accumulate packets. Proving that the induction step holds, we ensure that the inductive hypothesis will hold at the beginning of phase $j+1$ for the symmetric edges with an increased value of s_j, $s_{j+1} > s_j$.

From the inductive hypothesis, initially, there are s_j packets (called $S-flow$) in the queues f_1', f_4', f_5', f_7' requiring to traverse the edges e_0, f_2, f_4. 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 as follows:

- **Round 1:** This round lasts $|T_1| = \frac{s_j}{C}$ time steps.

  Adversary's behavior. During this round the edges $e_0, f_1, f_5, f_7, e_1, f_2', f_4'$ have high capacity C, while all the other edges have unit capacity. The adversary injects a set X of $|X| = rC|T_1|$ packets in queue e_0 wanting to traverse the edges $e_0, f_1, f_5, f_7, e_1, f_2', f_4'$ and a set S_1 of $|S_1| = r|T_1|$ packets in queue f_2 wanting to traverse the edges f_2, f_4.

  Evolution of the system configuration. All the S packets will traverse their initial edges in s_j time steps blocking the packets of set X in queue e_0 because the network $\mathcal{N}$ uses LIS protocol and the packets of set S are longer time in the system than X packets. For the same reason, the S packets block all the packets of set S_1 in queue f_2. At the same time, the packets of set S are delayed in queue f_2 due to the unit capacity of the edge f_2. The remaining packets of the set S in f_2 at the end of this round are $|S'| = |S| - |T_1|$ packets. The packets of S that manage to traverse the edge f_2 continue traversing their remaining path and they are absorbed. Therefore, the number of packets in queue f_2 at the end of this round requiring to traverse the edges f_2, f_4 is a set S_2 of $|S_2| = |S'| + |S_1|$ packets.

- **Round 2:** It lasts $|T_2| = \frac{|S_2|}{C}$ time steps.

  Adversary's behavior. During this round the edges $f_4, f_7, e_1, f_2', f_4'$ have high capacity C, while all the other edges have unit capacity. The adversary injects a set Y of $|Y| = rC|T_2|$ packets in queue f_4 requiring to traverse the edges $f_4, f_7, e_1, f_2', f_4'$.

  Evolution of the system configuration. The packets of set Y are blocked by the packets of the set S_2 in queue f_4 because S_2 packets are longer time in the system than Y packets. The packets of set S_2 traverse the edge f_4 and they are absorbed. At the same time, X packets are delayed in queue f_1 due to its unit capacity. Therefore, the remaining packets of X in queue f_1 is a set $|X'|$ of $|X'| = |X| - |T_2|$ packets requiring to traverse the edges f_1, $f_5, f_7, e_1, f_2', f_4'$.

- **Round 3:** It lasts $|T_3| = \frac{|X'|}{C}$ time steps.

  Adversary's behavior. During this round the edges $f_1, f_6, e_1, f_2', f_4'$ have high capacity C, while all the other edges have unit capacity. The adversary injects a set Z of $|Z| = rC|T_3|$ packets in queue f_1 requiring to traverse the edges $f_1, f_6, e_1, f_2', f_4'$.

  Evolution of the system configuration. The X' packets block the Z packets in queue f_1 because they are longer time in the system. At the same time X' packets are delayed in queue f_5 due to the unit capacity of the edge f_5. Therefore, the remaining packets of X' in queue f_5 is a set $|X''|$ of $|X''| = |X'| - |T_3|$ packets requiring to traverse the edges $f_5, f_7, e_1, f_2', f_4'$. Moreover, Y packets are delayed in queue f_4 due to the unit capacity of the edge f_4 during this round. Therefore, the remaining packets of Y in queue f_4 is a set $|Y'|$ of $|Y'| = |Y| - |T_3|$ packets requiring to traverse the edges $f_4, f_7, e_1, f_2', f_4'$. Note that during this round $|K| = 2|T_3|$ packets arrive in queue f_7 from queues f_4, f_6. However, the edge f_7 has unit capacity and the duration of this round is $|T_3|$ time steps. Consequently, $|L| = |T_3|$ packets will remain in queue f_7 at the end of this round requiring to traverse the edges f_7, e_1, f_2', f_4'. Therefore, the number of packets in queues f_1, f_4, f_5, f_7 requiring to traverse the edges e_1, f_2', f_4' at the end of this round is $s_{j+1} = |X''| + |Y'| + |Z| + |L|$.

In order to have instability, we must have $s_{j+1} > s_j$. This holds for

$$r^2[C^3 + C^2 - C] + r[2C^3 - 3C^2 + 1] > C^3 + C^2 - 2C + 1$$

When C tends to infinity, the instability threshold converges to $\sqrt{2} - 1$. This argument can be repeated for an infinite number of phases showing that the number of packets in the system increases forever.

□

4 Instability Bounds for Protocol Compositions

Theorem 2 *Let $r > \sqrt{2} - 1$. There is a network $\mathcal{G}_i$ and an adversary $\mathcal{A}$ of rate r such that the system $\langle \mathcal{G}_i, \mathcal{A}, \mathsf{P} \rangle$ is unstable in the quasi-static model of capacities, if* P *is a composition of* LIS *protocol with any protocol of a)* SIS, *b)* NTS *and c)* FTG.

Proof Sketch. **Part a)** Consider the network $\mathcal{G}_1$ in Figure 2. The network edges $e_0, e_1, f_1, f_1', f_3, f_3'$ use LIS protocol, while the edges $f_2, f_2', f_4, f_4', f_5, f_5', f_6, f_6', f_7, f_7'$ use SIS protocol. The construction of the adversary $\mathcal{A}$ is broken into phases.

Inductive Hypothesis: At the beginning of phase j (suppose j is even), there are s_j packets that are queued in the queues f_1', f_4', f_5', f_6' (in total) requiring to traverse the edges e_0, f_2, f_3, f_4.

Induction Step: At the beginning of phase $j+1$ there will be more than s_j packets (s_{j+1} packets) that are queued in the queues f_1, f_4, f_5, f_6 (in total) requiring to traverse the edges e_1, f_2', f_3', f_4'.

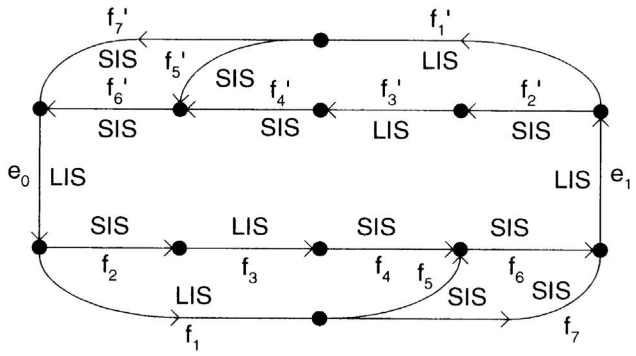

Figure 2: The network $\mathcal{G}_1$.

We will 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 f_1', f_4', f_5', f_6' requiring to traverse the edges e_0, f_2, f_3, f_4. 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 as follows:

- **Round 1:** This round lasts $|T_1| = \frac{s_j}{C}$ time steps.

  Adversary's behavior. During this round the edges $e_0, f_1, f_5, f_6, e_1, f_2', f_3', f_4'$ have high capacity C, while all the other edges have unit capacity. The adversary injects a set X of $|X| = rC|T_1|$ packets in queue e_0 wanting to traverse the edges $e_0, f_1, f_5, f_6, e_1, f_2', f_3', f_4'$ and a set S_1 of $|S_1| = r|T_1|$ packets in queue f_2 wanting to traverse the edge f_2.

  Evolution of the system configuration. All the S packets will traverse their initial edges in s_j time steps blocking the packets of set X in queue e_0 that uses LIS protocol because S packets are longer time in the system. At the same time, the packets of set S are delayed in queue f_2 that uses SIS protocol due to S_1 packets which are shorter time in the system and the unit capacity of the edge f_2. The remaining packets of the set S in f_2 at the end of this round requiring to traverse the edges f_2, f_3, f_4 are a set S_2 of $|S_2| = |S| - (|T_1| - |S_1|)$ packets. The S packets that manage to traverse the edge f_2 continue traversing their remaining path and they are absorbed.

- **Round 2:** It lasts $|T_2| = \frac{|S_2|}{C}$ time steps.

  Adversary's behavior. During this round the edges f_3, f_4, f_6, e_1, f_2', f_3', f_4'

have high capacity C, while all the other edges have unit capacity. The adversary injects a set Y of $|Y| = rC|T_2|$ packets in queue f_3 requiring to traverse the edges $f_3, f_4, f_6, e_1, f_2^{'}, f_3^{'}, f_4^{'}$.

Evolution of the system configuration. The packets of set Y are blocked by the set S_2 in queue f_3 that uses LIS protocol because S_2 packets are longer time in the system. The packets of set S_2 traverse the edge f_3 and they are absorbed. At the same time, the packet set X is delayed in queue f_1 due to the unit capacity of the edge f_1. Therefore, the remaining packets of X in queue f_1 is a set $|X^{'}|$ of $|X^{'}| = |X| - |T_2|$ packets requiring to traverse the edges $f_1, f_5, f_6, e_1, f_2^{'}, f_3^{'}, f_4^{'}$.

- **Round 3:** It lasts $|T_3| = \frac{|X^{'}|}{C}$ time steps.

Adversary's behavior. During this round the edges $f_1, f_7, e_1, f_2^{'}, f_3^{'}, f_4^{'}, f_3$ have high capacity C, while all the other edges have unit capacity. The adversary injects a set Z of $|Z| = rC|T_3|$ packets in queue f_1 requiring to traverse the edges $f_1, f_7, e_1, f_2^{'}, f_3^{'}, f_4^{'}$. Also, it injects a set S_3 of $|S_3| = r|T_3|$ packets in queue f_4 wanting to traverse the edge f_4, a set S_4 of $|S_4| = r|T_3|$ packets in queue f_5 wanting to traverse the edge f_5 and a set S_5 of $|S_5| = r|T_3|$ packets in queue f_6 wanting to traverse the edge f_6.

Evolution of the system configuration. The $X^{'}$ packets block the Z packets in queue f_1 that uses LIS protocol because they are longer time in the system. At the same time $X^{'}$ packets are delayed in queue f_5 that uses SIS protocol due to the unit capacity of the edge f_5 during this round and the S_4 packets that are shorter time in the system than $X^{'}$ packets. Therefore, the remaining packets of $X^{'}$ in queue f_5 is a set $|X^{''}|$ of $|X^{''}| = |X^{'}| - (|T_3| - S_4)$ packets requiring to traverse the edges $f_5, f_6, e_1, f_2^{'}, f_3^{'}, f_4^{'}$. Moreover Y packets are delayed in queue f_4 that uses SIS protocol due to the unit capacity of the edge f_4 during this round and the S_3 packets that are shorter time in the system than Y packets. Therefore, the remaining packets of Y in queue f_4 is a set $|Y^{'}|$ of $|Y^{'}| = |Y| - (|T_3| - S_3)$ packets requiring to traverse the edges $f_4, f_6, e_1, f_2^{'}, f_3^{'}, f_4^{'}$.

Note that during this round $|K| = 2|T_3| - |S_3| - |S_4|$ packets arrive in queue f_6 from queues f_4, f_5. However, the edge f_6 has unit capacity and uses SIS protocol that gives priority to the S_5 packets. Furthermore, the duration of this round is $|T_3|$ time steps. Consequently, the number of packets that remain in queue f_6 at the end of this round requiring to traverse the edges $f_6, e_1, f_2^{'}, f_3^{'}, f_4^{'}$ is $|L| = |K| + |S_5| - |T_3|$. Therefore, the number of packets in queues f_1, f_4, f_5, f_6 requiring to traverse the edges $e_1, f_2^{'}, f_3^{'}, f_4^{'}$ at the end of this round is $s_{j+1} = |X^{''}| + |Y^{'}| + |Z| + |L|$.

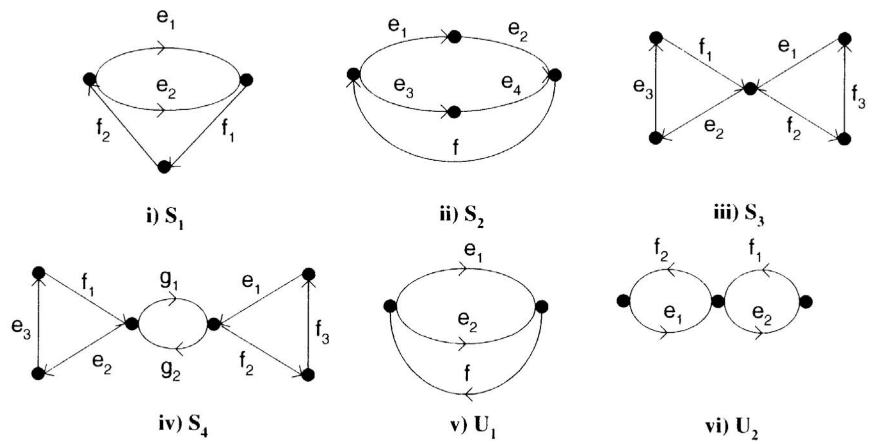

Figure 3: Pure simple-path networks $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4$, and not pure simple-path networks $\mathcal{U}_1, \mathcal{U}_2$.

In order to have instability, we must have $s_{j+1} > s_j$. This holds for

$$r^2[C^3 + 2C^2 - C - 1] + r[2C^3 - 3C^2 - C + 2] > C^3 + C^2 - 2C + 1$$

When C tends to infinity, the instability threshold converges to $\sqrt{2} - 1$. This argument can be repeated for an infinite number of phases showing that the number of packets in the system increases forever for $r > \sqrt{2} - 1$.

Parts b, c) *Parts a, b, c* are proved similarly. One difference is the replacement of SIS protocol in *Part a* by NTS in *Part b* and FTG in *Part c*. The topology of the network $\mathcal{G}_2$ and the adversary construction for proving instability in *Part b* are similar to *Part a*. But in *Part c* the topology of the network $\mathcal{G}_3$ contains additional paths that start at FTG queues and have sufficient lengths to guarantee the priority of specific packet sets against other packets whose preservation into the system is required for the inductive hypothesis reproduction. □

5 Instability Bounds for Forbidden Subgraphs

Theorem 3 *Let $r > 0.8191$. There is a network $\mathcal{S}_i$ and an adversary $\mathcal{A}$ of rate r such that the system $\langle \mathcal{S}_i, \mathcal{A}, \mathsf{NTG-U-LIS} \rangle$ is unstable in the quasi-static model of capacities when the adversary assigns to the packets simple-paths with no repeated vertices where $\mathcal{S}_i$ is the network a) $\mathcal{S}_1$, b) $\mathcal{S}_2$, c) $\mathcal{S}_3$ and d) $\mathcal{S}_4$.*

Proof Sketch. Part a) Consider the network $\mathcal{S}_1$ in Figure 3 that uses NTG-U-LIS protocol. We break the construction of the adversary $\mathcal{A}$ into phases.

Induction Hypothesis: At the beginning of phase j, there are s_j packets that are queued 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}, which will be queued 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 holds. 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 edge f_1. In order to prove that the induction step 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 $|T_1| = \frac{s_j}{C}$ time steps.

  Adversary's behavior. During this round all the network edges have high capacity C. The adversary injects in queue f_1 a set X of $|X| = rC|T_1|$ packets wanting to traverse the edges f_1, f_2.

  Evolution of the system configuration. The S packets block the X packets in queue f_1 because S packets are nearest to their destination than X packets. Therefore, S packets have priority over X packets, they traverse the edge f_1 and they are absorbed.

- **Round 2:** It lasts $|T_2| = \frac{|X|}{C}$ time steps.

  Adversary's behavior. During this round all the network edges have high capacity C. The adversary injects a set Y of $|Y| = rC|T_2|$ packets in queue f_2 requiring to traverse the edges f_2, e_1.

  Evolution of the system configuration. X packets block Y packets in queue f_2 because they have nearest to go than Y packets. X packets traverse the edge f_2 and they are absorbed.

- **Round 3:** It lasts $|T_3| = \frac{|Y|}{C}$ time steps.

  Adversary's behavior. During this round all the network edges have high capacity C. The adversary injects a set Z of $|Z| = rC|T_3|$ packets in queue f_2 requiring to traverse the edges f_2, e_2. Also, it injects a set Z_1 of $|Z_1| = rC|T_3|$ packets in queue e_1 requiring to traverse the edges e_1, f_1.

  Evolution of the system configuration. Y packets block Z packets in queue f_2 that uses NTG-U-LIS protocol because although Y packets are in the same distance from their destination with the Z packets they are longer time in the system than Z packets. Moreover Y packets block Z_1 packets in queue e_1 because they have nearest to go than Z_1 packets. Y packets traverse the edge e_1 and they are absorbed.

- **Round 4:** It lasts $|T_4| = \frac{|Z|}{C}$ time steps.

  Adversary's behavior. During this round the edge e_1 has unit capacity, while all the other edges have high capacity C. The adversary injects a set Z_2 of $|Z_2| = rC|T_4|$ packets in queue e_2 requiring to traverse the edges e_2, f_1.

  Evolution of the system configuration. Z packets block Z_2 packets in queue e_2 because they have nearest to go than Z_2 packets. Z packets traverse the edge e_2 and they are absorbed. Moreover Z_1 packets are delayed in queue e_1 due to the unit capacity of the edge e_1 during this round. Therefore, the remaining packets of Z_1 in queue e_1 at the end of this round is a set $|Z_1'|$ of $|Z_1'| = |Z_1| - |T_4|$ packets requiring to traverse the edges e_1, f_1, while the rest Z_1 packets traverse their remaining path and they are absorbed.

  Therefore, the number of packets in queues e_1, e_2 requiring to traverse the edge f_1 at the end of this round is $s_{j+1} = |Z_2| + |Z_1'|$. In order to have instability, we must have $s_{j+1} > s_j$. This holds for $\frac{r^4C + r^3(C-1)}{C} > 1$. When C tends to infinity, the instability threshold converges to 0.8191. This argument can be repeated for an infinite number of phases showing that the number of packets in the system increases forever for $r > 0.8191$.

Parts b, c, d) As in *Part a*, the adversary of *Parts b, c, d* assigns to the injected packets simple paths that cannot contain repeated edges and vertices. Again the adversarial construction is broken into phases that are further broken into four time rounds. The proof is based on induction on the number of phases. In *Part b* the *inductive argument* we prove is that if at the beginning of a phase j, there are s_j packets in the queues e_2, e_4 (f_1, f_3 in *Parts c, d*) requiring to traverse the edge f (the edges f_1, e_2 and f_3, e_1, e_2 in *Part c* and f_1, e_2 and f_3, e_1, g_2, e_2 in *Part d*), then at the beginning of phase $j+1$ there will be more than s_j packets in the queues e_2, e_4 (f_1, f_3 in *Parts c, d*) requiring to traverse the edge f (f_1, e_2 and f_3, e_1, e_2 in *Part c* and f_1, e_2 and f_3, e_1, g_2, e_2 in *Part d*). This inductive argument can be repeated for an infinite number of phases showing that the number of packets in the system increases forever. The basic idea behind the adversarial construction is that in the first three rounds of each phase all the edges have capacity C and the used mechanism for delaying packets that are injected in a round are packets injected in previous rounds, while in the last round the transition of a link's capacity from C to 1 is used as an additional blocking mechanism for delaying packets into the system. □

Theorem 4 *Let $r > r_i$. 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 in the quasi-static model of capacities when the adversary assigns to the packets simple-paths with repeated vertices where $\mathcal{U}_i$ is the network a) $\mathcal{U}_1$ with instability threshold $r_1 = 0.794$ and b) $\mathcal{U}_2$ with instability threshold $r_2 = 0.755$.*

Proof Sketch. The adversary can assign to the injected packets simple paths that can contain repeated vertices, which permits the better exploitation of the network topology for delaying packets into the system as longer paths can be assigned to the injected packets. The time is split into phases that are broken in four (*Part a*) and three (*Part b*) time rounds. The proof is based on induction on the number of phases. In *Part a* the *induction argument* we prove is that if at the beginning of a phase j, there are s_j packets in the queues e_1, e_2 (e_1, f_1 in *Part b*) requiring to traverse the edge f (f_2 in *Part b*), then at the beginning of phase $j+1$ there will be more than s_j packets in the queues e_1, e_2 (e_1, f_1 in *Part b*) requiring to traverse the edge f (f_2 in *Part b*). This inductive argument can be repeated for an infinite number of phases showing that the number of packets in the system increases forever. The basic idea behind the adversarial construction is that packet sets that are injected in a round are used as blocking mechanism for delaying packet sets that are injected in next rounds. Additionally, in some rounds it is used the transition of the capacity of a link from C to 1 as a blocking mechanism for delaying packets into the system. □

6 Discussion and Directions for Further Research

Note that the equation that defines the instability bound of LIS is $r^2[C^3+C^2-C]+r[2C^3-3C^2+1] > C^3+C^2-2C+1$, while the one for the compositions of protocols (LIS-SIS, LIS-NTS, LIS-FTG) is $r^2[C^3+2C^2-C-1]+r[2C^3-3C^2-C+2] > C^3+C^2-2C+1$. Although they differ for small C values when $C \to \infty$ they both reduce to $r^2+2r-1>0$, implying $r > \sqrt{2}-1$. Perhaps this is due to the structural similarity of the proposed networks; curiously the change of protocol does not affect the *limiting characteristic* equation for instability. Our results of Section 5 suggest that, for every unstable network, its instability threshold in the model of quasi-static capacities may be lower than for the classical adversarial queueing model. Proving (or disproving) this remains an open problem.

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

[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, 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] 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*, Dahlia Mahkli ed., pp. 88–102, LNCS 2508, October 2002.

[9] D. Koukopoulos, M. Mavronicolas, S. Nikoletseas, and P. Spirakis. The Impact of Network Structure on the Stability of Greedy Protocols. *Proceedings of the 5th Italian Conference on Algorithms and Complexity*, accepted.

[10] 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*, pp. 3–14, December 2001.

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

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

Dimitrios Koukopoulos is with the 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

Marios Mavronicolas is with the Department of Computer Science, University of Cyprus, 1678 Nicosia, Cyprus. E-mail: mavronic@ucy.ac.cy

Paul Spirakis is with the 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: spirakis@cti.gr