Adversarial Models for Priority-Based Networks

C. Àlvarez, M. Blesa, J. Díaz, and M. Serna
Department de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, Jordi Girona 1-3, Campus Nord, E-08034 Barcelona, Spain

A. Fernández
Laboratorio de Algoritmia Distribuida, Universidad Rey Juan Carlos, Tulipan s/n, Campus de Móstoles, E-28933 Madrid, Spain

In this article, we propose several variations of the adversarial queueing model and address stability issues of networks and protocols in those proposed models. The first such variation is the *priority model,* which is directed at static network topologies and takes into account the case in which packets can have different priorities. Those priorities are assigned by an adversary at injection time. A second variation, the *variable priority model,* is an extension of the priority model in which the adversary may dynamically change the priority of packets at each time step. Two more variations, namely the *failure model* and the *reliable model,* are proposed to cope with dynamic networks. In the failure and reliable models the adversary controls, under different constraints, the failures that the links of the topology might suffer. Concerning stability of networks in the proposed adversarial models, we show that the set of *universally stable* networks in the adversarial model remains the same in the priority, variable priority, failure, and reliable models. From the point of view of protocols (or queueing policies), we show that several protocols that are universally stable in the adversarial queueing model remain so in the priority, failure, and reliable models. However, we show that the *longest-in-system* (LIS) protocol, which is universally stable in the adversarial queueing model, is not universally stable in any of the other models we propose. Moreover, we show that no queueing policy is universally stable in the variable priority model. Finally, we analyze the problem of deciding stability of a given network under a fixed protocol. We provide a characterization of the networks that are stable under *first-in-first-out* (FIFO) and LIS in the failure model (and therefore in the reliable and priority models). This characterization allows us to show that the stability problem under FIFO and LIS in the failure model can be solved in polynomial time. © 2004 Wiley Periodicals, Inc. NETWORKS, Vol. 45(1), 23–35 2005

Keywords: stability; adversarial queueing theory; (contention-resolution) protocols; packet-switched networks

1. INTRODUCTION

The model of Adversarial Queueing Theory (AQT) proposed by Borodin et al. [8] considers the time evolution of a packet-routing network as a game between an adversary and a queueing policy. At each time step the adversary may inject a set of packets at some of the nodes. For each packet the adversary specifies the sequence of edges that it must traverse, after which the packet will be absorbed. If more than one packet tries to cross an edge e at the same time step, then the queueing policy chooses one of these packets to be sent across e. The remaining packets wait in the queue. The system evolves synchronously, and then this game advances to the next time step. The main goal of the model is to study *stability issues* of the network, under different *greedy* queueing policies. Stability is the property of deciding whether at any time the maximum number of packets present in the system is bounded by a constant that may depend on system parameters. Recall that a protocol is *greedy* if whenever there is at least one packet waiting to use an edge, the protocol advances a packet through the edge.

In the *adversarial* AQT *model* the adversary is restricted by a pair (r, b), where $b \geq 0$ is *the burstiness* and $0 < r < 1$ is the *injection rate.* The adversary must obey the following rule [4, 8]. For every time interval I,

Received April 2004; accepted September 2004
Correspondence to: M. Serna; e-mail: mjserna@lsi.upc.es
Contract grant sponsor: FET Programme of the EU; contract grant number: IST-2001-33116 (FLAGS)
Contract grant sponsor: Spanish CICYT; Contract grant numbers: TIC-2001-4917-E and TIC2002-04498-C05-03
Contract grant sponsor: Comunidad de Madrid; contract grant number: 07T/0022/2003
Contract grant sponsor: Universidad Rey Juan Carlos; contract grant number: PPR-2003-37
Contract grant sponsor: Catalan Research Council of the Generalitat de Catalunya (to M.B.); contract grant number: 2001FI-00659
Contract grant sponsor: *Distinció per a la recerca* (research distinction prize) of the Generalitat de Catalunya (to J.D.)
DOI 10.1002/net.20044
Published online in Wiley InterScience (www.interscience.wiley.com).

$$N_e(I) \leq r|I| + b, \quad (1)$$

where $N_e(I)$ denotes the number of packets injected by the adversary that have paths containing edge e during the time interval I.[1] In the following, we will refer to the adversarial AQT model simply as the ***adversarial model***.

In this article we consider generalizations of the adversarial AQT model that take into account the possibility that packets may have different (prefixed and dynamically changing) *priorities.* Inspired by the priority models and by the growing importance of wireless mobile networks, we also consider some variations of the adversarial model for *dynamic networks.*

1.1. Models of Traffic with Priorities

Considering priorities is a natural approach to model contemporary networks. Today's networked applications, such as data mining, e-commerce, and multimedia, are bandwidth hungry and time sensitive. These applications need networks that accommodate these requirements and guarantee some *Quality of Service* (QoS). Classifying and prioritizing network traffic is a basic technique to fulfill these goals. In most networks this is supported by mapping into the packets information about the type or the priority of their contents. For instance, the Internet Protocol (IP) supports QoS in the header of the packets, for example, by using the IP Precedence/ToS field. Although rarely used, the current Internet protocol IPv4 uses three bits of the Type of Service (ToS) byte to designate the IP Precedence. Eight priorities are possible, from 0 (default) to 7; the higher the number, the higher the priority. Then different queues are supported (priority queueing), which are serviced in strict order of queue priority. With these services, the routers and switches can police the traffic entering the network, assign priority, and ensure optimal paths through the IP network. In the new version of the protocol, IPv6, a longer traffic class field is included in the packet header, allowing a more detailed discrimination of priorities.

We are interested in analyzing the power of an adversary that can prioritize the packets. We will consider two settings: in the first one, a packet always has the same priority, while in the second one the adversary is allowed to modify the priority of a packet. Consequently, we define two new models for adversarial queueing theory, the ***priority model*** and the ***variable priority model***. When packets have priorities, each edge has a queue associated with every possible priority value. If at a certain time more than one packet tries to cross the same edge e, the queueing policy chooses the packet to be sent across e from the nonempty queue with highest priority. The limitations on the adversary are the same as in the adversarial model.

[1] Recall that in [8], the model is defined over windows of fixed size w and the equation $N_e(I) \leq r|I|$, for $|I| = w$. It is known that both models are equivalent [13].

As we said, in the priority model the priority of a packet is fixed at injection time and never changes, while in the variable priority model the adversary decides the priority of a packet at each time step. We will assume that the number of possible priorities that the adversary can use is prefixed.

1.2. Models for Dynamic Networks

Inspired by the priority models and by the growing importance of wireless mobile networks, we also consider some variations of the adversarial model for dynamic networks. In wireless mobile networks (ad hoc networks) some connections between nodes may fail or change quickly and unpredictably. Hence, in our dynamic network models edges can appear and disappear arbitrarily. Note that in the priority model, we can *simulate* the failure of an edge e by injecting a packet whose path only contains e with a priority higher than any other packet in the queue of e. Once this packet is in the queue, it will be sent first, and the remaining packets in the queue will have to wait until the next time step. It seems natural to introduce models for dynamic networks in which the adversary controls not only the packet arrivals, but also the edge failures. At any time step, the adversary can produce the failure of an edge. If an edge e fails, the packets in e's queue wait until the moment when the edge recovers. The constraints of an adversary are defined taking into consideration the number of time steps during any interval I that the edge e is down. We assume that a packet cannot cross a link when it fails, and that during an edge failure the packets that arrive wait at the queue of the edge.

Let $F_e(I)$ be the number of steps during a time interval I in which the edge e is down. We propose suitable restrictions on the adversary, obeying the rule: the number of packets introduced by the adversary during interval I, which have paths containing e, cannot be greater than the number of times that e is alive in the interval. Furthermore, as packets must follow a prespecified path, the adversary should not be able to fail an edge permanently. To guarantee that we keep a bound on the maximum number of failures of an edge e in any time interval, we propose a new model, the ***failure model***, in which the adversary is controlled by a common bound on both packet injections and edge failures, according to the restriction that, for all intervals I,

$$N_e(I) + F_e(I) \leq r|I| + b. \quad (2)$$

Observe than in this case, during a given interval, the injection rate limits the maximum number of failures and the maximum number of packet injections per edge.

With the aim of allowing a higher degree of edge failures, we define a new dynamic model. To do so, we introduce an additional parameter α. Then, in this model the adversary is characterized by the tuple (r, b, α), where r and b are defined as before and $r \leq \alpha \leq 1$. For any edge e and any interval I, the adversary must obey the constraint

$$N_e(I) + \alpha F_e(I) \leq r|I| + b. \quad (3)$$

In the model defined by Equation (3), we can consider two extreme cases: For $\alpha = 1$ we obtain the same constraint as Equation (2). However, for $\alpha = r$ we get a model in which an edge can be permanently down. Notice that, in the case when $r < \alpha \leq 1$, if the adversary produces an edge failure, then it is forced to recover the edge after at most $b/(\alpha - r)$ steps, because, otherwise, it will violate Equation (3). We are interested in this latter property, and will use the term ***reliable model*** to denote a model in which the adversary has parameters (r, b, α) with r and b as in the adversarial model, $r < \alpha \leq 1$, and which is constrained by Equation (3).

1.3. Greedy Protocols

As in [4, 8], we will only consider greedy protocols that apply their policies to the queues at the edges according to some local or global criteria. The main queueing protocols we consider are: FIFO, LIFO, SIS, LIS, NTG, FTG, NFS, and FFS.

The protocol LIFO (*last-in-first-out*) gives priority to the packet that arrived latest at the edge queue; in FIFO (*first-in-first-out*), precedence is given to the packet that has arrived first at the queue of the edge. The protocol that gives precedence to the packet last introduced into the system is SIS (*shortest-in-system*), while in LIS (*longest-in-system*) every queue gives precedence to the packet that has been in the system the longest time. The protocol NTG (*nearest-to-go*) assigns precedence to the packet that is closest to its destination and FTG (*farthest-to-go*) selects the packet that is farthest away from its destination. NFS (*nearest-from-source*) and FFS (*farthest-from-source*) consider the same policies but taking the distance to the source node of the packets as a reference point. NFS is sometimes called NTS (*nearest-to-source*). Aside from these protocols, we also consider the NTG–LIS protocol, which works as NTG, but resolving ties using the LIS protocol.

It is known that FTG, NTS, SIS, and LIS are universally stable in the adversarial model while FIFO, LIFO, NTG, and FFS are not [4].

1.4. Related Work

Two adversarial models for dynamic networks have been proposed in [6] and [5]. In both models the injected packets are defined by specifying only their source and destination, and thus are not forced to follow a prespecified path. In both cases the adversary is restricted to guarantee that a static multicommodity flow problem has a solution. Stability results are obtained using a load-balancing algorithm, for the case that the adversary injection rate is one, and the packets have a unique common destination. The main difference in both models is that in [6], the adversary has to provide a solution to the associated multicommodity flow problem, while in [5], the injection pattern must obey a condition that guarantees the existence of a solution.

The dynamic models proposed in this article consider the case in which the route to be followed by the injected packets is completely specified. Our models and the dynamic models proposed in [6] and in [5] have the common characteristic that, for every interval I, the adversary cannot inject at any edge e (or at any set S of nodes for the model in [5]) more packets than the number of packets that e can absorb (or the number of edges with only one extreme in S).

An interpretation of the heterogeneity and dynamicity of the network as *slowdowns* in the transmission of packets or variations in *link capacities* was studied in [9]. In both models (slowdown and capacity) packets are injected with a prespecified path, as in the original adversarial model (and ours). In their slowdown model, packets suffer delays while crossing a link. A packet suffers delay d while crossing edge e if it starts to traverse the link at time t and arrives to the tail of e at time $t + d$. During this transfer time the rest of packets that want to cross e wait in the queue of e. This article considers two cases: the *static* case, in which every link e has a fixed slowdown s_e (which gives the delay incurred by any packet that crosses e), and the *dynamic* case, in which the slowdown in link e depends of the time [i.e., if a packet starts to cross e at time t if suffers delay $s_e(t)$]. In the static slowdown, the adversary is controlled by a pair (r, b)[2] and satisfies the condition that for any edge e and any interval I,

$$N_e(I) \leq r \frac{|I|}{s_e} + b, \quad (4)$$

whereas in the dynamic slowdown model, the adversary is controlled by a pair (r, b) and is restricted by the condition that for any edge e and any interval I,

$$N_e(I) \leq r \sum_{t \in I} \frac{1}{s_e(t)} + b, \quad (5)$$

where for any edge e and any time step t, define $\overline{s_e}(t) = \max_{t'}\{s_e(t')|t' \leq t \leq t' + s_e(t')\}$.

The slowdown model has some similarities with the failure model. In the static case, the slowdown s_e of an edge e can be interpreted as "link e *fails* during $s_e - 1$ steps." However, there is a difference, because in the slowdown model a packet p is delayed after leaving e's queue, while in the failure model p waits in e's queue. This means that in the failure model when e is recovered, the next packet to be served might not be p.

In the capacity model every edge e in a network has an integral capacity c_e. Again, there are two different models: the static model, in which c_e does not change, and the

[2] Although in [9] the adversary is restricted in the window model, for simplicity, we express it here in the (r, b) model.

TABLE 1. Summary of concepts. For all the models $b \geq 0$ and $0 < r < 1$.

Model	Traffic pattern constraint	Additional features
Adversarial (AQT)	$N_e(I) \leq r\|I\| + b$	—
Priority	$N_e(I) \leq r\|I\| + b$	Every packet has a fixed priority
Variable priority	$N_e(I) \leq r\|I\| + b$	Every packet has a variable priority
Failure	$N_e(I) + F_e(I) \leq r\|I\| + b$	Every link (edge) e might fail
Reliable	$N_e(I) + \alpha F_e(I) \leq r\|I\| + b$, $r < \alpha \leq 1$	Every link (edge) e might fail

dynamic model, in which $c_e(t)$ can be different at every time step t. In general, at step t a link is able to transmit simultaneously up to $c_e(t)$ packets. The adversary is controlled by a pair (r, b) and the restriction that for every edge e and every interval I,

$$N_e(I) \leq r|I|c_e(t) + b. \tag{6}$$

Note that a zero capacity $c_e(t) = 0$ can be seen as a failure of link e at time t. This shows that the failure model and the dynamic capacity model are somewhat related.

There are several results in [9] under these models. First, they show that every universally stable network remains universally stable in the slowdown and the capacity models, even in the dynamic case. Then, they show that SIS, NTS, and FTG remain universally stable in all the models. However, the situation is different for LIS, because it is universally stable in the static slowdown model but it is not in the dynamic slowdown and capacity models. It has to be pointed out here that this result does not directly apply to the failure model, because the proof that LIS is not universally stable in the dynamic capacity model uses nonzero capacities (see Theorem 3.1 in [9]) and there is no trivial transformation.

1.5. Our Contributions

In this article we address stability issues in the proposed adversarial models, namely the *failure, reliable, priority,* and *variable priority* models (see Table 1 for a summary). Recall that a network is *stable under a protocol and an adversary* if the number of packets in the system at any time step remains bounded. Our first results are concerned with *universal stability* of networks. We show that the property that a network is stable under any adversary and queueing policy remains the same in the adversarial, priority, variable priority, failure, and reliable models.

From the point of view of *universal stability* of queueing policies, we show that NFS, SIS, and FTG (which are universally stable in the adversarial model [4]) remain so in the failure, reliable, and priority models. However, we show that LIS, a universally stable queueing policy in the adversarial model [4], is not universally stable in the failure, reliable, and priority models. Moreover, we show that no protocol is universally stable in the variable priority model.

Finally, we analyze the problem of deciding stability of a given network under a fixed protocol. We provide a characterization of the networks that are stable under FIFO and LIS in the failure model. Because this characterization has turned out to be the same as the one given in [2] for universal stability in the adversarial model, we conclude that the stability problem in the failure model under the FIFO and LIS protocols can be solved in polynomial time. Let us observe that finding the characterization of the stability under FIFO in the adversarial model remains still an open problem [2].

2. UNIVERSAL STABILITY OF NETWORKS

Let $\mathcal{M}$ denote a model in the set {*adversarial, reliable, failure, priority, variable priority*} as defined in the previous section. A communication system in a model $\mathcal{M}$ is formed by three main components: a network G, where G is a digraph, a greedy scheduling protocol $\mathcal{P}$ and a traffic pattern $\mathcal{A}$, which is represented by an adversary that follows the restrictions of $\mathcal{M}$. All the digraphs representing networks considered in this article may have multiple edges but no loops. The packets transmitted over those digraphs follow predefined *path* trajectories, which might repeat nodes but not edges. For all the proposed models, we can formally define different concepts around universal stability in the following way:

Definition 1. *The system $S = (G, \mathcal{A}, \mathcal{P})$ is stable in the model $\mathcal{M}$ if, at any time step, the maximum number of packets in the system is bounded by a fixed value, that may depend on system parameters.*

Definition 2. *The pair $(G, \mathcal{P})$ is stable in the model $\mathcal{M}$ if, for any adversary $\mathcal{A}$ following the restrictions of $\mathcal{M}$, the system $S = (G, \mathcal{A}, \mathcal{P})$ is stable in $\mathcal{M}$.*

The concept of *universal stability* applies both to networks and protocols.

Definition 3. *A network G is universally stable in the model $\mathcal{M}$ if, for any greedy queueing policy $\mathcal{P}$, the pair $(G, \mathcal{P})$ is stable in $\mathcal{M}$.*

Definition 4. *A greedy protocol $\mathcal{P}$ is universally stable in the model $\mathcal{M}$ if, for any digraph G, the pair $(G, \mathcal{P})$ is stable in $\mathcal{M}$.*

We want to compare the relative power of the adversaries in the different proposed models. To this aim, we need to specify when an adversary in any of the models can simulate another adversary in another different model.

Definition 5. *An adversary $\mathcal{A}$ in model $\mathcal{M}$ simulates adversary $\mathcal{A}'$ in model $\mathcal{M}'$ when for any network G and any protocol $\mathcal{P}$ if $(G, \mathcal{A}, \mathcal{P})$ is stable in $\mathcal{M}$, then $(G, \mathcal{A}', \mathcal{P})$ is stable in $\mathcal{M}'$.*

According to these definitions, now we can state the following relations:

Lemma 1. (1) *Any adversary with parameters (r, b) in the adversarial model can be simulated by an adversary with parameters (r, b) in the failure model.* (2) *Any adversary with parameters (r, b) in the failure model can be simulated by an adversary with parameters $(r, b, 1)$ in the reliable model.* (3) *Any adversary with parameters (r, b) in the failure model can be simulated by an adversary with parameters (r, b) and two different priorities in the priority model.* (4) *Any adversary with parameters (r, b, α) in the reliable model can be simulated by an adversary with parameters $(r + 1 - \alpha, b)$ in the failure model.*

Proof.

1. Any adversary that obeys Equation (1) can be seen as an adversary constrained by Equation (2) that never forces an edge failure.
2. By definition.
3. The priority model adversary $\mathcal{A}'$ will inject packets with two priorities, *low* and *high.* At each time step, $\mathcal{A}'$ injects with low priority the same packets that $\mathcal{A}$ injects. If an edge e fails, $\mathcal{A}'$ injects a packet (e) with *high* priority. Observe that the parameters (r, b) for $\mathcal{A}$ are also valid for $\mathcal{A}'$.
4. Given (r, b, α), with $r < \alpha \leq 1$, an adversary in the reliable model obeys the constraint (3). Because $\alpha \leq 1$, $N_e(I) + F_e(I) \leq r|I| + (1 - \alpha)F_e(I) + b$. Moreover, as $F_e(I) \leq |I|$, $N_e(I) + F_e(I) \leq r|I| + (1 - \alpha)|I| + b \leq (r + 1 - \alpha)|I| + b$. Thus, the adversary obeys Equation (2) of the failure model with $r' = 1 + r - \alpha < 1$. ∎

Observe that the failure and reliable models are equivalent. Then any stability and instability result for one model applies to the other as well. Then, for conciseness in the rest of the article we will only consider the failure and priority models (nonvariable and variable). Now we can state our main result in this section.

Theorem 2. *Given a digraph G, the following properties are equivalent*: (1) *G is universally stable in the adversarial model*, (2) *G is universally stable in the failure model*, (3) *G is universally stable in the priority model*, *and* (4) *G is universally stable in the variable priority model.*

Proof.

$1 \Rightarrow 4$. For any system $S = (G, \mathcal{A}, \mathcal{P})$ in the variable priority model, we define the system $S' = (G, \mathcal{A}', \mathcal{P}')$ in the adversarial model as follows. At each time step, $\mathcal{A}'$ injects the same packets that $\mathcal{A}$ injects (without priority). The queueing policy $\mathcal{P}'$ will determine the packet to be served, simulating the system S.

$4 \Rightarrow 3$. By definition.

$3 \Rightarrow 2$ and $2 \Rightarrow 1$. These follow from Lemma 1. ∎

3. UNIVERSAL STABILITY OF PROTOCOLS

In this section we address the universal stability property in the failure and priority models, from the point of view of the queueing policy. We will consider the basic protocols presented in the introduction. Recall that FTG, NTS, SIS, and LIS are universally stable in the adversarial model while FIFO, LIFO, NTG, and FFS are not [4]. Because any adversary in the adversarial model can be seen as an adversary in the other models, FIFO, LIFO, NTG, and FFS are not universally stable in the failure and priority models. Thus, we will not consider them any further in this section.

We will first show how any adversary with parameters (r, b) in the priority model can be simulated by an adversary with the same parameters in the adversarial model, for any protocol $\mathcal{P} \in \{\text{FTG, NTG, NFS, FFS}\}$. The simulation requires us to change the network slightly. Let G be a directed graph and $\mathcal{A}_\pi$ an adversary with parameters (r, b) in the priority model that uses at most π priorities. Every injected packet p has a priority π_p in the ordered interval $[1, \ldots, \pi]$, with 1 being the lowest priority.

Lemma 3. *For any system $S = (G, \mathcal{A}_\pi, \mathcal{P})$ in the priority model, for $\mathcal{P} \in \{FTG, NTG\}$, there is a system $S' = (G', \mathcal{A}', \mathcal{P})$ in the adversarial model such that*: *G is a subgraph of G'*; *$\mathcal{A}'$ has parameters (r, b)*; *if a packet p is injected in S at time t with path r, a packet p' is injected in S' at time t with a path r' obtained by concatenating r and a path of edges not in G*; *and if p crosses edge e at time t' in S, p' crosses e at time t' in S'.*

Proof. We first consider FTG and construct G' by attaching to every node of G as many outgoing disjoint paths as there are edges incident to the node. Each outgoing path will be associated with a different incoming edge and has length $(\pi - 1)d$, where d is the length of the longest directed path in G.

Then each packet p injected in S with priority π_p and path r is replaced in S' by a packet p' that first follows the same path r and then follows $(\pi_p - 1)d$ edges of the outgoing path associated to the last edge of r. Observe that this packet, while it remains in the queues of the original network G, still has at least $(\pi_p - 1)d + 1$ edges to cross. In these queues another packet q' that replaced a packet q with priority $\pi_q < \pi_p$ still has to cross at most $\pi_q d < (\pi_p$

$- 1)d + 1$ edges, and hence, will always be blocked by p'. Similarly, any packet that replaced one with higher priority than p will always block p'.

The proof for NTG is similar, but in this case p' has to cross $(\pi - \pi_p)d$ edges of the outgoing path. ∎

As a consequence of the previous lemma and the results in [4], we get

Theorem 4. *FTG is universally stable in the priority, failure, and reliable models.*

To get a similar result for NFS and FFS we have to relate two different networks.

Lemma 5. *For any system $S = (G, \mathcal{A}_\pi, \mathcal{P})$ in the priority model, for $\mathcal{P} \in \{NFS, FFS\}$, there is a system $S' = (G', \mathcal{A}', \mathcal{P})$ in the adversarial model such that: G is a subgraph of G'; $\mathcal{A}'$ has parameters (r, b'), where $b' = r(\pi - 1)d + b$; if a packet p is injected in S with path r, a packet p' is injected in S' with a path r' obtained by concatenating a path of edges not in G and r; and if p crosses edge e at time t in S, p' crosses e at time t in S'.*

Proof. We first consider NFS and construct G' by attaching to every node of G incoming disjoint paths as follows. For each node s, each edge e leaving s, and each priority $\pi_p \in \{1, \ldots, \pi - 1\}$, G' will have b disjoint incoming paths associated with the pair (e, π_p) of length $(\pi - \pi_p)d$ whose last node is s.

Then each packet p injected in S with priority $\pi_p < \pi$ at time t at the source node s and with edge e as the first edge in its path is replaced in S' by a packet p' injected at the first node of one of the b incoming paths associated with (e, π_p) at time $t - (\pi - \pi_p)d$. If several packets are injected in S at the same time with the same initial edge e and the same priority π_p, each uses a different disjoint incoming path from those associated with (e, π_p). Notice that p' follows the incoming path and then the same path as p. A packet p with priority $\pi_p = \pi$ is replaced in S' by a packet p' with the same injection time and path as in S.

Observe that any packet p' in S', while in the queues of the original network G, has already crossed at least $(\pi - \pi_p)d$ edges. Similarly, in those queues, any packet q' that replaced a packet q with priority $\pi_q > \pi_p$ has crossed at most $(\pi - \pi_q)d + d - 1 < (\pi - \pi_p)d$ edges, and hence, will always block p'. Similarly, any packet that replaced one with lower priority than p will always be blocked by p'.

Note that in S' a packet p' can be injected up to $(\pi - 1)d$ steps before the packet p it is replacing was injected in S. Hence, at any time interval I in S' the maximum number of injections for each edge is $r|I| + b + r(\pi - 1)d$. The first part $r|I| + b$ corresponds to the packets injected in I in S, and the second part $r(\pi - 1)d$, with the packets of low priority that have been injected early in S'.

The proof for FFS is similar, but in this case priority π_p is associated with incoming paths of length $(\pi_p - 1)d$. ∎

As a consequence of the previous theorem and the results in [4], we get

Theorem 6. *NFS is universally stable in the priority, failure, and reliable models.*

To show the universal stability of SIS in the priority model we follow similar arguments to those of Lemma 2.2 in [4] for showing the universal stability of SIS in the adversarial model.

Lemma 7. *For any network G and any adversary $\mathcal{A}_\pi$, the system $(G, \mathcal{A}_\pi, SIS)$ is stable under the priority model.*

Proof. The proof is by induction on the number of priorities. When $\pi = 1$ the SIS protocol was shown in [4] to be universally stable under the adversarial model.

Let us assume that the system $(G, \mathcal{A}_{\pi-1}, \text{SIS})$ is stable. Observe that in $(G, \mathcal{A}_\pi, \text{SIS})$ the system formed by the packets p with $\pi_p > 1$ is independent of the packets with priority 1, and therefore, they form a stable system. Let S be the maximum number of packets in the system requiring any particular edge, with priority larger than 1, at any time. Based on [4], we define the following recurrence:

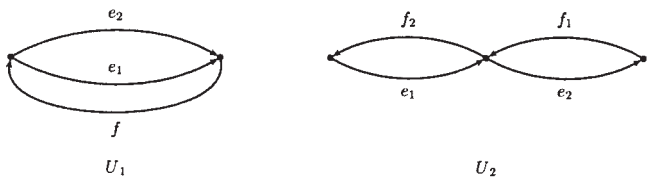
$$K_1 = b + S; \quad K_{j+1} = \frac{K_j + b}{1 - r}$$

Let d be the length of the longest simple path in G. By a similar argument to that of Lemma 2.2 in [4], but replacing b by $b + S$ in the base case of the induction, we can show that when a packet p, with priority one, arrives at the queue of the j-th edge e_j on its path there are at most $K_j - 1$ packets requiring any edge e in the path of p with priority over p.

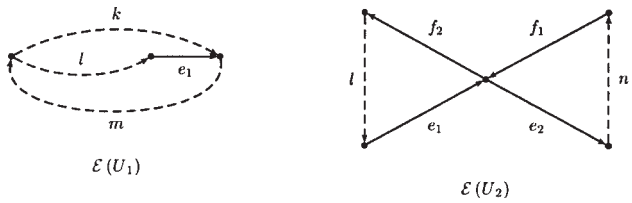
Then K_d denotes the maximum number of packets with priority one, in any queue. Therefore, the system with π priorities is stable. ∎

Theorem 8. *SIS is universally stable in the priority, failure, and reliable models.*

Even though the universal stability property of FTG, NFS, and SIS in the adversarial model is preserved in the priority model, we will see that this is not the case for LIS, which is universally stable in the adversarial model. The next result stages the nonuniversal stability of LIS in the failure model, and therefore, in all the other models. This is obtained by considering, for example, graph U_1 of Figure 1(a), and by showing that U_1 is not stable under LIS in the failure model. That proof is given in Section 4.2 as Lemma 19, where more detailed results about the stability under the



(a) Basic digraphs characterizing universal stability in the adversarial model [2].



(b) Extensions of the basic digraphs in Figure 1(a), where edges have been extended to paths. Let us denote by p_k, p_l, p_m and p_n those paths, where $k, m \geq 1$ and $l, n \geq 0$.

FIG. 1. Family of subgraphs characterizing universal stability in the adversarial model. (a) Basic digraphs characterizing universal stability in the adversarial model [2]. (b) Extensions of the basic digraphs in Figure 1(a), where edges have been extended to paths. Let us denote by p_k, p_l, p_m, and p_n those paths, where k, $m \geq 1$ and l, $n \geq 0$.

LIS protocol are provided. Putting this result together with Lemma 1 we get

Theorem 9. *LIS is not universally stable in the failure, reliable, and priority models.*

Finally, we consider universal stability of protocols in the variable priority model. In this case, the adversary is so powerful that no protocol is universally stable. In the following lemma, we show that any greedy protocol $\mathcal{P}$ is stable in the network U_1.[3]

Lemma 10. *For any greedy protocol $\mathcal{P}$, the pair $(U_1, \mathcal{P})$ is not stable in the variable priority model.*

Proof. We will construct an adversary that uses only two priorities, 0 and 1, with 1 being the highest. The adversary is described by rounds; during a round the priority of a packet will not change, but it might be reassigned at the beginning of the next round. At the beginning there are s packets that want to traverse edge f, and all of them get priority 1. The adversary $\mathcal{A}$ will play injections in five rounds:

ROUND 1. For s steps, the adversary injects rs packets of the form (fe_2) with priority 0. These injections get mixed with the initial packets at edge f, and are blocked because their priority is lower, independently of the greedy queueing policy.

ROUND 2. The adversary assigns priority 1 to the packets waiting at f's queue. For the next rs steps, the adversary injects a set of r^2s packets of the form (fe_1) and r^2s packets of the form (e_2), all of them with priority 0. Injections (fe_1) are blocked by the packets (fe_2) from the first round. The r^2s injections of the form (e_2) get mixed with packets (fe_2), so that $r^2s(e_2)$ packets are queued at e_2 at the end of the round.

ROUND 3. The adversary assigns priority 1 to the packets queued at f and 0 to the packets queued at e_2. For the next r^2s steps, the adversary injects r^3s packets of the form (e_2) with priority 0 and r^3s of the form (e_1f) with priority 0. The simple injections on e_2 get mixed with the packets at e_2 from the previous round. At the end of the round there will be r^3s such packets queued at e_2. The (e_1f) injections get mixed with the packets of the form (fe_1) at edge e_1. This

[3] Instability proofs are usually based on induction. The goal is to demonstrate that the number of packets in the system increases from phase to phase (and, by applying the inductive hypothesis, they can increase infinitely). The configuration of the system at the end of each phase must be the analogous to the configuration at the beginning (in terms of the type of packets and their location), but with an increased number of packets. For the sake of simplicity, in our proofs we only reproduce the inductive phase (which is composed of rounds), and sometimes we omit some additive constants in our analysis. These omissions, however, do not change the final result. The proofs of Lemmas 10, 13–16, 19–22 follow the same schema.

edge at the end of the round will have in total $r^3 s$ packets of the form $(e_1 f)$.

ROUND 4. The adversary assigns priority 0 to the packets queued at e_1 and priority 1 to the packets queued at e_2. For the next $r^3 s$ steps, the adversary injects $r^4 s$ packets of the form $(e_2 f)$ with priority 0 and $r^4 s$ packets of the form (e_1) with priority 1. The injections at e_1 block the last $r^4 s$ packets queued at e_1 at the beginning of the round. The $(e_2 f)$ injections are blocked by the (e_2) packets from the previous round. So at the end of the round there will be $r^4 s$ packets $(e_2 f)$ and $r^4 s$ packets $(e_1 f)$ queued at e_2 and e_1, respectively.

ROUND 5. The adversary assigns priority 1 to the packets queued at e_1 and e_2. For the next $r^4 s$ steps, the adversary injects $r^5 s$ packets of the form (f).

At the end there are $(r^5 + r^4)s$ packets (f) in the system. The adversary $\mathcal{A}$ described above makes the network U_1 unstable when $r^5 + r^4 > 1$, that is, $r \geq 0.857$. ∎

Hence, we can state,

Theorem 11. *There is no universally stable greedy protocol in the variable priority model.*

Now, by using Lemma 4.3 in [4], which implicitly states that if any protocol is stable for a system then there is also a stable greedy protocol for the system, we obtain

Corollary 12. *There is no universally stable protocol in the variable priority model.*

4. STABILITY UNDER A PROTOCOL

In this section we analyze the complexity of the problem of deciding whether a given network G is stable under a fixed protocol $\mathcal{P}$. There are few results for this problem. It has been shown that deciding stability under NTG–LIS is polynomially solvable [2]. The polynomial time decidability of stability under FFS in the case that the adversary can solve ties arbitrarily was shown in [1]. Further results for undirected graphs and other variations can be found in [4] and [2].

To characterize stability under a given protocol $\mathcal{P}$, first we start by identifying the families of digraphs that are stable under $\mathcal{P}$. Then, the simplest digraphs that are not stable should be identified. By iteratively applying subdivision operations over those simplest digraphs, we must "extend" them to define a family of digraphs. Stability under the protocol $\mathcal{P}$ will be characterized once it is shown that those extensions are not stable under $\mathcal{P}$ either. Our results show the same characterization as in [2], using the same basic graph family and the exclusion of a set of forbidden subgraphs. Before formally stating our results, we need to introduce some theoretical definitions over digraphs. We consider the following subdivision operations over digraphs:

Definition 6. *The subdivision of an edge* (u, v) *in a digraph* G *consists in the addition of a new node* w *and the replacement of* (u, v) *by the two directed edges* (u, w) *and* (w, v).

Definition 7. *The subdivision of a two-cycle* (u, v), (v, u) *in a digraph* G *consists in the addition of a new node* w *and the replacement of* (u, v), (v, u) *by the edges* (u, w), (w, u), (v, w), *and* (w, v).

Given a digraph G, $\varepsilon(G)$ denotes the family of digraphs including G and all the digraphs obtained from G by successive edge or two-cycle subdivisions. Given a family of digraphs $\mathcal{F}$, $\mathcal{S}(\mathcal{F})$ denotes the family of digraphs that contain a graph in $\mathcal{F}$ as a subgraph.

Figure 1(a) provides the two basic graphs needed to characterize universal stability, and Figure 1(b) gives the shape of the extensions of those graphs. This basic family provides the characterization of the stability properties. It is known that a digraph is universally stable in the adversarial model if and only if $G \notin \mathcal{S}(\varepsilon(U_1) \cup \varepsilon(U_2))$ [2]. The same property characterizes network stability under NTG-LIS [2] and FFS [1]. It is also known that, for a given digraph G, checking whether $G \notin \mathcal{S}(\varepsilon(U_1) \cup \varepsilon(U_2))$ can be done in polynomial time [2].

Nothing is known about the complexity of deciding stability in the adversarial model for any other queueing policy. In the following, we will provide a similar characterization of stability in the failure model under FIFO and LIS.

4.1. FIFO Stability Under the Failure Model

The FIFO protocol gives precedence to the packet that arrived first to the queue it schedules. One of the first results relating the adversarial model and this protocol was obtained in [4], where the FIFO protocol was shown not to be universally stable.

Much effort has recently been devoted to studying stability and instability properties of FIFO. A network-dependent absolute constant is provided in [10] such that FIFO is stable against any adversary with smaller injection rate. A lower bound of 0.749 for the instability is calculated in [11]. This bound was decreased to 0.5 [12]. In [14], it is shown that FIFO is stable if the injection rate is smaller than $1/(d - 1)$. Recently, it has been proved that FIFO can become unstable at arbitrarily low injections rates [7].

In the following, we show the instability of the FIFO protocol in the networks U_1, U_2 [see Fig. 1(a)] and their corresponding extensions $\varepsilon(U_1)$ and $\varepsilon(U_2)$ [see Fig. 1(b)] in the failure model. As a more general result, for network U_2 we show instability in the adversarial model.

The extended graphs are formed by applying the above introduced subdivision operations over U_1 and U_2 to paths.

Let us denote by p_k, p_l, p_m, and p_n those paths, where k, $m \geq 1$ and l, $n \geq 0$. The number of packets in the initial configuration is considered to be large enough to guarantee that the time given to every round is sufficient for the packets to cross the possible paths (p_k, p_l, p_m, or p_n) in their packet trajectory and arrive where they are supposed to arrive. For simplicity, we will omit floors and ceilings, and sometimes will count steps and packets roughly; by carrying these through the computations one loses some additive constants, which are offset by the fact that s was a large constant.

Lemma 13. *The pair (U_1, FIFO) is not stable in the failure model.*

Proof. At the beginning there are s packets that want to traverse edge f. The adversary $\mathcal{A}$ will play injections in five rounds:

ROUND 1. For s steps, the adversary injects rs packets of the form (fe_2). These injections get mixed with the initial packets at edge f, and are blocked there because the queueing protocol is FIFO.

ROUND 2. For the next rs steps, the adversary injects a set of r^2s packets of the form (fe_1) and r^2s packets of the form (e_2). Injections (fe_1) are blocked by the packets (fe_2) from the first round. The r^2s injections of the form (e_2) get mixed with packets (fe_2), so that $r^2s(e_2)$ packets are queued at e_2.

ROUND 3. For the next r^2s steps, the adversary injects r^3s packets of the form (e_2) and r^3s of the form (e_1f). The simple injections on e_2 will be blocked by the packets at e_2 from the previous round. The (e_1f) injections get blocked by the packets of the form (fe_1) at edge e_1. This edge at the end of the round will have in total r^3s packets of which $r^3s/(1 + r)$ are (e_1) and $r^4s/(1 + r)$ are (e_1f).

ROUND 4. For the next r^3s steps, the adversary injects r^4s packets of the form (e_2f) and makes edge e_1 fail for r^4s steps. The failures at e_1 block the last r^4s packets queued at e_1 at the beginning of the round. As the proportion is maintained, at the end of the round there are $r^4s/(1 + r)$ packets (e_1) and $r^5s/(1 + r)$ packets (e_1f). The (e_2f) injections are blocked by the (e_2) packets from the previous round.

ROUND 5. For the next r^4s steps, the adversary injects r^5s packets of the form (f). At the end, there are $(r^5 + (r^5/(1 + r)))s$ packets (f) in the system. The adversary $\mathcal{A}$ described above makes the network U_1 unstable when $r^5((2 + r)/(1 + r)) > 1$, that is, $r \geq 0.920$. ∎

Lemma 14. *The pair (U_2, FIFO) is not stable in the adversarial model.*

Proof. We will not use failures. At the beginning there are s packets that want to traverse edge f_2. The adversary $\mathcal{A}$ will play injections in five rounds:

ROUND 1. For s steps, the adversary injects rs packets of the form $(f_2e_1e_2)$. These injections get mixed with the initial packets at edge f_2, and are blocked there because the queueing protocol is FIFO.

ROUND 2. For the next rs steps, the adversary injects a set of r^2s packets of the form (f_2e_1) and r^2s packets of the form (e_2). All the injections (f_2e_1) are blocked by the older packets and stay at the end of the round. The injections (e_2) get mixed with the original packets, so at the end of the round there are r^2s packets that want to cross (e_2).

ROUND 3. For the next r^2s steps, the adversary injects r^3s packets of the form (e_1) and r^3s of the form $(e_2f_1f_2)$. The simple injections on e_1 get mixed with the old packets arriving at e_1; at the end there will remain r^3s packets that want to traverse e_1. The second set is delayed by the old packets.

ROUND 4. For the next r^3s steps, the adversary injects r^4s packets of the form (e_1f_2) and r^4s packets of the form (f_1). At the end of the round there are r^4s packets that want to traverse e_1f_2, $(r^4s)/(r + 1)$ packets that want to traverse f_1f_2 and $(r^5s)/(r + 1)$ packets that want to traverse only f_1.

ROUND 5. For the next r^4s steps, the adversary injects r^5s packets of the form (f_2). At the end there are $(r^5 + (r^4/(1 + r)))s$ packets (f_2) in the system. This implies instability for $r \geq 0.914$. ∎

Lemma 15. *Any graph in $\varepsilon(U_1)$ is not stable under FIFO in the failure model.*

Proof. Observe that, in the FIFO protocol, neither the length of the path nor the distances to the source or destination are important for the scheduling decisions. Only the order of arrival to the queues matters. From this fact, proofs for instability of U_1 and U_2 can be easily adapted just by using an adversary that replaces edges by paths. Notice, however, that the longer the paths the fewer packets in the network accumulates. For large enough initial configurations, instability can always be forced.

Let G_1 be the graph $\varepsilon(U_1)$ described in Figure 1(b). At the beginning, there are s packets that want to traverse edge p_m. The adversary $\mathcal{A}$ will play injections in five rounds:

ROUND 1. For s steps, the adversary injects rs packets of the form (p_mp_k). These injections get mixed with the initial packets at the first edge of p_m, and are blocked there because the queueing protocol is FIFO.

ROUND 2. For the next rs steps, the adversary injects a set of r^2s packets of the form $(p_mp_le_1)$ and r^2s packets of the

form (p_k). Injections $(p_m p_l e_1)$ are blocked by the packets $(p_m p_k)$ from the first round. A total of rm injections of the form (p_k) are not accumulated. The rest get mixed with packets $(p_m p_k)$, so that $r^2 s - rm$ packets of the form (p_k) are queued at the first edge of p_k.

ROUND 3. For the next $r^2 s$ steps, the adversary injects $r^3 s$ packets of the form (p_k) and $r^3 s$ of the form $(p_l e_1 p_m)$. Then, $r^3 s - rm$ injections on p_k will be blocked by the packets at the first edge of p_k remaining from the previous round. All the $(p_l e_1 p_m)$ injections get mixed with the packets of the form $(p_m p_l e_1)$ at the first edge of p_l, except rm of them. At the end of the round this edge will have in total $r^3 s - rm$ packets of which $r^3 s - rm/(1 + r)$ are $(p_l e_1)$ and $r^4 s - r^2 m/(1 + r)$ are $(p_l e_1 p_m)$.

ROUND 4. For the next $r^3 s - rm$ steps, the adversary injects $r^4 s - r^2 m$ packets of the form $(p_k p_m)$ and makes edge e_1 fail for $r^4 s - r^2 m$ steps. The failures at e_1 block the last $r^4 s - r^2 m - rl$ packets queued at the first edge of p_l at the beginning of the round. As the proportion is maintained, at the end of the round there are $r^4 s - r^2 m - rl/(1 + r)$ packets $(p_l e_1)$ and $r^5 s - r^3 m - r^2 l/(1 + r)$ packets $(p_l e_1 p_m)$. The $(p_k p_m)$ injections are blocked by the (p_k) packets from the previous round.

ROUND 5. For the next $r^4 s - r^2 m$ steps, the adversary injects $r^5 s - r^3 m$ packets of the form (p_m). At the end of this round there are $r^5 s - r^3 m + ((r^5 s - r^3 m - r^2 l)/(1 + r))$ packets (p_m) in the system. The adversary $\mathcal{A}$ described above makes the network $\varepsilon(U_1)$ unstable when $r^5 s - r^3 m + ((r^5 s - r^3 m - r^2 l)/(1 + r)) > s$. Note that $(r^4 m + r^2 l)/(1 + r) < (m + l)/(1 + r) = C$, so for large enough s, an injection rate r can be found such that $(r^5 + (r^5/(1 + r)))s - C > s$ holds, and thus $G_1 \in \varepsilon(U_1) \cup \varepsilon(U_2)$ is not stable under FIFO in the failure model. ∎

Lemma 16. *Any graph in $\varepsilon(U_2)$ is not stable under FIFO in the failure model.*

Proof. Now let G_2 be a graph in $\varepsilon(U_2)$ as described in Figure 1(b). At the beginning there are s packets that want to traverse edge f_2. The adversary $\mathcal{A}$ will play injections in five rounds:

ROUND 1. For s steps, the adversary injects rs packets of the form $(f_2 p_l e_1 e_2)$. These injections get mixed with the initial packets at edge f_2, and are blocked there because the queueing protocol is FIFO.

ROUND 2. For the next rs steps, the adversary injects a set of $r^2 s$ packets of the form $(f_2 p_l e_1)$ and $r^2 s$ packets of the form (e_2). All the injections $(f_2 p_l e_1)$ are blocked by the older packets and stay at the end of the round. The injections (e_2) get mixed with the original packets, so at the end of the round there are $r^2 s - rl$ packets that want to cross e_2.

ROUND 3. For the next $r^2 s$ steps, the adversary injects $r^3 s$ packets of the form $(p_l e_1)$ and $r^3 s$ of the form $(e_2 p_n f_1 f_2)$. The former get mixed with the old packets arriving at the first edge of p_l, and at the end there will remain $r^3 s$ packets. From the second set, $r^3 s - rl$ packets are delayed by the old packets.

ROUND 4. For the next $r^3 s$ steps, the adversary injects $r^4 s$ packets of the form $(p_l e_1 f_2)$ and provokes $r^4 s$ failures in e_2. At the end of the round there are $r^4 s$ packets that want to traverse $p_l e_1 f_2$ and $r^4 s - r^2 l$ packets from the previous round blocked at e_2 due to the failures.

ROUND 5. For the next $r^4 s$ steps, the adversary injects $r^5 s$ packets of the form (f_2). Because of the possibly different lengths of the paths p_l and p_n, there are $r|l - n|$ injections that will not be accumulated.

At the end there are $(r^4 + r^5)s - r^2 l - r|l - n|$ packets in the system waiting to traverse f_2. The adversary $\mathcal{A}$ makes the network $\varepsilon(U_2)$ unstable when $(r^4 + r^5)s - r^2 l - r|l - n| > s$. Notice that $r^2 l + r|l - n| < 2l + n = C$, so for large enough s, an injection rate r can be found such that $(r^4 + r^5)s - C > s$ holds, and thus $G_2 \in \varepsilon(U_1) \cup \varepsilon(U_2)$ is not stable under FIFO in the failure model. ∎

Then, putting all these results together we have:

Lemma 17. *Any graph in $\varepsilon(U_1) \cup \varepsilon(U_2)$ is not stable under FIFO in the failure model.*

As we have pointed out before, all networks $G \notin \mathcal{S}(\varepsilon(U_1) \cup \varepsilon(U_2))$ are universally stable in the adversarial model. From Theorem 2 the set of universally stable networks is the same for all the models considered in this article (adversarial, failure, priority, and variable priority). Hence, all networks $G \notin \mathcal{S}(\varepsilon(U_1) \cup \varepsilon(U_2))$ are stable under FIFO in all these models. Taking into account that if a network has an unstable subnetwork it is also unstable we get the following result.

Theorem 18. *For digraph G, the pair $(G, FIFO)$ is stable in the failure model if and only if G is universally stable in the adversarial model.*

A corollary of this result is the equivalence between FIFO stability in the failure model and universal stability in the adversarial model. Furthermore, as instability in the failure model implies instability in the priority and reliable models, the characterization of FIFO stability remains the same in the priority and reliable models. Observe also that stability under FIFO can be checked in polynomial time for the failure, priority, and reliable models.

4.2. LIS Stability under the Failure Model

The LIS protocol gives priority to the packet that was longer in the system, that is, that joined the network earlier. In [4], the LIS protocol was shown to be universally stable in the adversarial model, with $O(b/(1-r)^d)$ queue size per edge and delay of the packets of the order $O(b/(1-r)^d)$. However, as we have shown the protocol is not universally stable in the failure model. Now we will show that deciding stability under LIS in the failure model can be performed in polynomial time.

We proceed as in the case of FIFO by showing, respectively, the instability of the basic graphs given in Figure 1(a) and their extensions in Figure 1(b).

Lemma 19. *The pair* (U_1, LIS) *is not stable in the failure model.*

Proof. At the beginning there are s packets that want to traverse edge f. An adversary $\mathcal{A}$ playing injections exactly as does the adversary in the analogous Lemma 13 under FIFO would accumulate $r^4s + r^5s$ packets in the queue of edge f at the end of the fifth round. This would make the system unstable for $r \geq 0.857$. ∎

Lemma 20. *The pair* (U_2, LIS) *is not stable in the failure model.*

Proof. Starting with s packets that want to traverse edge f_2, let $\mathcal{A}$ be an adversary playing injections exactly as the adversary in the analogous Lemma 14 under FIFO. But the adversary we used in Lemma 14 was not producing failures, so substitute there the single injections of the form (f_1) at round 4 by failures of edge f_1. Thus, at the end of the fifth round, this would allow $r^4s + r^5s$ packets to accumulate in the queue of edge f_2 and prove that the system can become unstable for $r \geq 0.857$. ∎

Lemma 21. *Any graph in* $\varepsilon(U_1)$ *is not stable under LIS in the failure model.*

Proof. Let G_1 be the graph $\varepsilon(U_1)$ described in Figure 1(b). At the beginning there are s packets that want to traverse edge p_m. The adversary $\mathcal{A}$ will play injections in five rounds exactly as the adversary for FIFO (see Lemma 13) but modifying the packet trajectories by extending them to paths: where e_1 appears put $p_l e_1$, and replace e_2 and f in paths by p_k and p_m, respectively.

Also, instead of making edge e_1 fail at the fourth round, this adversary makes the first edge of p_l fail. At round five, because of the possibly different lengths of the paths p_l and p_k, there are $r|l-k|$ injections that will not be accumulated. The number of packets in the initial configuration is considered to be large enough to guarantee that packets arrive at the first edge of p_m. This adversary makes the system unstable under LIS when $r^4s - r^2m + r^5s - r^3m - r|k-l| > s$. Notice that $r^2m + r^3m + r|k-l| < 2m + l + k = C$ and, for large enough s, an injection rate r can be found such tat $(r^4 + r^5)s - C > s$ holds, thus showing instability of G_1 under FIFO in the failure model. ∎

Lemma 22. *Any graph in* $\varepsilon(U_2)$ *is not stable under LIS in the failure model.*

Proof. Now let G_2 be the graph $\varepsilon(U_2)$ described in Figure 1(b). At the beginning there are s packets that want to traverse edge f_2. The adversary $\mathcal{A}$ will play injections in five rounds. For the first three rounds it injects the same kind of packets as the adversary for FIFO (see Lemma 14) but modifying the packet trajectories by extending them to paths: where e_1 appears put $p_l e_1$, and replace e_2 in paths by $e_2 p_n$. We show the last two rounds:

ROUND 4. At the beginning there are r^3s packets of the form $(p_l e_1)$ and $r^3s - rl$ packets of the form $(e_2 p_n f_1 f_2)$. For the next r^3s steps, the adversary injects r^4s packets of the form $(e_1 f_2)$ and causes r^4s failures in edge f_1. From the injections in e_1, only $r^4s - rl$ will remain at the end of the round. The failures will accumulate $r^4s - rn - r^2l$ packets that want to traverse $f_1 f_2$.

ROUND 5. During $r^4s - rl$ steps, the adversary injects $r^5s - r^2l$ packets (f_2).

When the fifth round finishes, there are $(r^4 + r^5)s - 2r^2l - rn$ packets at f_2. Thus, instability would hold under LIS when $(r^4 + r^5)s - 2r^2l - rn > s$. Notice that $2r^2l + rn < 2l + n = C$ and, for large enough s, an injection rate r can be found such that $(r^4 + r^5)s - C > s$ holds, thus showing instability of G_2 under LIS in the failure model. ∎

Lemma 23. *Any graph in* $\varepsilon(U_1) \cup \varepsilon(U_2)$ *is not stable under LIS in the failure model.*

Therefore, as in the case of FIFO, we have

Theorem 24. *A digraph G is stable under LIS in the failure model if and only if G is universally stable in the adversarial model.*

Again, stability under LIS in the failure, reliable, and priority models coincides with universal stability in the adversarial model, and therefore, it can be checked in polynomial time for the failure, priority, and reliable models.

4.3. The Variable Priority Model

For the variable priority model the situation is simpler. It is easy to adapt the proof of Lemma 14 to get instability results for the graph U_2. This together with Lemma 10 gives the following result.

Lemma 25. *Let $\mathcal{P}$ be any greedy protocol. Any graph in $\varepsilon(U_1) \cup \varepsilon(U_2)$ is not stable under $\mathcal{P}$ in the variable priority model.*

Therefore, we have

Theorem 26. *Let $\mathcal{P}$ be any greedy protocol. A digraph G is stable under $\mathcal{P}$ in the variable priority model if and only if G is universally stable in the adversarial model.*

5. CONCLUSIONS AND OPEN PROBLEMS

We have proposed several variations on the adversarial model to cope with packet priorities and link failures. We have studied universal stability from the point of view of both the network and the queueing policy. We have also addressed the complexity of deciding stability under a fixed protocol.

We have shown that in the adversarial, failure, reliable, priority, and variable priority models, the set of networks that are universally stable remains the same. The models present a different behavior with respect to the universal stability of protocols, because LIS is universally stable in the adversarial model, but it is not universally stable in the other models. In contrast, we have shown that there are no universally stable protocols for the variable priority model.

We have proposed a new and natural way to model the behavior of queueing systems in dynamic networks. Compared to the slowdown models introduced in [9], our results show that the power of an adversary in the failure and in the dynamic slowdown model is quite similar. In both cases the LIS protocol is not universally stable. However, the static slowdown model is less powerful than the failure model as LIS remains universally stable [9].

The argument used in the proof of Theorem 4.1 in [9] can be used to show how to construct an adversary in the variable priority model that simulates an adversary in the dynamic slowdown model. It would be of interest to find constructions, similar to those given in Lemmas 3 and 5, to relate the power of the slowdown and failure models without changing the protocol.

Regarding the dynamic capacity model, the authors frequently use the trick of injecting $c - c_e(t)$ dummy packets, which only need to traverse link e. This can be done without violating the load condition for a network with static capacity c provided that $c_e(t) > 0$ (see Theorems 3.3 and 3.4 in [9]). It will be of interest to analyze the case with zero capacities.

It remains as an open problem to show the existence of a protocol that is universally stable in the failure model but not in the priority model.

All the already known characterizations of stability under a protocol are equivalent to universal stability in the adversarial model, even in the variable priority model. It is an interesting open question to know whether there is any protocol $\mathcal{P}$, not universally stable, for which there are networks that are not universally stable but that are stable under $\mathcal{P}$. Additional open questions concerning this property are given in Figure 2.

	Adversarial AQT	Failure	Priority	Reliable
LIS	trivial in [4]	P	P	P
NTG	P in [2]	P	P	P
FFS	P in [1]	P	P	P
LIFO	open	open	open	open
FIFO	open	P	P	P

FIG. 2. Complexity of deciding stability under a protocol in the different adversarial models. The property is decidable in polynomial time in those cases labeled with **P** (the uncited results are provided in this article). Observe that for LIFO, decidability remains an open question in all the models. Let us observe that the characterization of FIFO stability in the adversarial model remains an open problem [2].

The presented characterization of stability under given protocols, as is common in the literature, allows the adversary to inject packets with nonsimple paths. It would be interesting to derive stability characterizations when the adversary can inject simple paths only.

The most general adversarial model considering failures would be one in which the adversarial injections (requiring any edge e during the time interval I) would be constrained by

$$N_e(I) \leq r(|I| - F_e(I)) + b. \tag{7}$$

In such a case the number of consecutive time steps in which an edge can be down is unlimited. Observe that having an adversary that makes an edge e fail forever after some step is equivalent, from the point of view of stability, to considering the network that results from removing e from the set of edges. Then, we would deal with a system with a simpler topology with an additional finite set of packets requiring e that would be kept in the system forever. Therefore, the interesting cases are those in which the adversary follows restriction (7) but cannot make an edge fail forever. A restricted version modeling only short-lived failures was considered in [3]. The $\mathcal{D}_1$ model in [3] is obtained when considering restriction (7) and the existence of a w bounding the number of consecutive steps that any edge can be failed. In [3], it is shown that the $\mathcal{D}_1$ model is equivalent to the failure model. The remaining natural model to study would consider that the duration of a failure is not infinite but is not bounded either. To the best of our knowledge, every question concerning stability in such a model is currently an open question.

Acknowledgments

Part of this work was presented in *28th International Symposium on Mathematical Foundations of Computer Sci-*

ence, Bratislava, Slovakia, 2003. LNCS 2747:142–151, Springer. We wish to thank the anonymous referees for their constructive comments on this article.

REFERENCES

[1] C. Àlvarez, M. Blesa, J. Díaz, A. Fernández, and M. Serna, The complexity of deciding stability under FFS in the adversarial model, Informat Process Lett 90 (2004), 261–266.

[2] C. Àlvarez, M. Blesa, and M. Serna, A characterization of universal stability in the adversarial queuing model, SIAM J Comput 34 (2004), 41–46.

[3] C. Àlvarez, M. Blesa, and M. Serna, The impact of failure management on the stability of communication networks, 10th International Conference on Parallel and Distributed Systems (ICPADS'04), IEEE, Newport Beach, CA, 2004, pp. 153–160.

[4] M. Andrews, B. Awerbuch, A. Fernández, J. Kleinberg, T. Leighton, and Z. Liu, Universal stability results and performance bounds for greedy contention–resolution protocols, J ACM 48 (2001), 39–69.

[5] E. Anshelevich, D. Kempe, and J. Kleinberg, Stability of load balancing algorithms in dynamic adversarial systems, 34th Annual ACM Symposium on Theory of Computing (STOC'02), ACM, Montréal, Canada, 2002, pp. 399–406.

[6] B. Awerbuch, P. Berenbrink, A. Brinkmann, and C. Scheideler, Simple routing strategies for adversarial systems, 42th IEEE Symposium on Foundations of Computer Science (FOCS'01), IEEE, Las Vegas, NV, 2001, pp. 158–167.

[7] R. Bhattacharjee and A. Goel, Instability of FIFO at arbitrarily low rates in the adversarial queueing model, 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS'03), IEEE, Cambridge, MA, 2003, pp. 160–167.

[8] A. Borodin, J. Kleinberg, P. Raghavan, M. Sudan, and D. Williamson, Adversarial queueing theory, J ACM 48 (2001), 13–38.

[9] A. Borodin, R. Ostrovsky, and Y. Rabani, Stability preserving transformations: Packet routing networks with edge capacities and speeds, 12th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'01), SIAM, Washington, DC, 2001, pp. 601–610.

[10] J. Díaz, D. Koukopoulos, S. Nikoletseas, M. Serna, P. Spirakis, and D. Thilikós, Stability and non-stability of the FIFO protocol, 13th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA'01), ACM, Crete Island, Greece, 2001, pp. 48–52.

[11] D. Koukopoulos, M. Mavronicolas, S. Nikoletseas, and P. Spirakis, "On the stability of compositions of universally stable, greedy contention–resolution protocols," 16th International Symposium on Distributed Computing (DISC'02), D. Malkin (Editor), Toulouse, France, Lecture Notes in Computer Science 2508 (2002), 88–102.

[12] Z. Lotker, B. Patt-Shamir, and A. Rosén, New stability results for adversarial queuing, SIAM J Comput 33 (2004), 286–303.

[13] A. Rosén, A note on models for non-probabilistic analysis of packet switching networks, Informat Process Lett 84 (2002), 237–240.

[14] Z.-L. Zhang, Z. Duan, and Y. Hou, Fundamental trade-offs in aggregate packet scheduling. 9th IEEE International Conference on Network Protocols (ICNP'01), IEEE, Riverside, CA, 2001, pp. 129–137.