Maria Serna's Contributions to Adversarial Queuing Theory

Maria J. Blesa[a,1,*], Antonio Fernández Anta[b,2]

[a] *Universitat Politècnica de Catalunya, Barcelona*
[b] *IMDEA Networks Institute, Madrid*

Abstract

Adversarial Queuing Theory (AQT) is one of the areas to which Maria Serna has deeply contributed in her scientific career. Most of her research in this area took place during the lustrum $2001-2005$, while advising the PhD of Maria J. Blesa [19], and also in conjunction with other researchers. AQT is an adversarial model for the study of packet-switching communication networks under worst case conditions. In this model a fundamental concept is that of *stability* of a combination of network and packet scheduling protocol in front of an adversary that controls the arrival of packets. There is stability if the adversary cannot cause the number of packets in the network to grow unbounded. Maria's contributions, summarized in this document, include results in the stability of networks and protocols, and the definition and study of AQT models with important new characteristics, like priorities, failures, or networks with different bandwidths and packet lengths. These results keep impacting the new research currently done in the AQT model.

1. Introduction

Before summarizing the main results in this research area, let us introduce the basic concepts and present the state-of-the-art by the time that Maria started working in it.

1.1. *Stability*

The concept of stability is central and recurring in this article. In the context of packet-switching communication networks, *stability* refers to the property that the amount of packets in a network remains always bounded under some given operational conditions. The bound is a constant that might depend on system parameters, e.g., the diameter of the network, but not on the time. Indirectly, stability also implies that the size of the buffers required to store in-route packets is bounded. One of the main goals in the study of the behavior of packet-switched communication networks is to determine the conditions for stability. The problem of deciding stability and detecting sufficient conditions for it to hold has been investigated under various models of communication networks and traffic arrival assumptions [27, 28, 23, 32, 9]. Some models make probabilistic assumptions on the traffic pattern, while others replace these traditional stochastic arrival assumptions by worst-case inputs in the traffic pattern, in order to perform a worst-case analysis. The latter are closer to the traditional analysis of algorithms, and to the approach considered here.

In this paper, stability is studied in relation to the three main components of a synchronous communication system: a network $\mathcal{G}$, a traffic (arrival) pattern defined by $\mathcal{A}$, and a protocol $\mathcal{P}$. These three components define the communication system $(\mathcal{G}, \mathcal{A}, \mathcal{P})$. The topology of the *network*

[*] Corresponding author

[1] This research has been partially supported by the Agency for Management of University and Research Grants (AGAUR) of the Government of Catalonia under project 2017 SGR 786 (ALBCOM) and by MINECO and FEDER funds under grant GRAMM (ref. TIN2017-86727-C2-1-R).

[2] This research has been partially supported by the Spanish Ministry of Science, Innovation and Universities grant DiscoEdge (TIN2017-88749-R), the Comunidad de Madrid grant EdgeData-CM (P2018/TCS-4499), and the NSF of China grant 61520106005.

$\mathcal{G}$ is modelled by a (directed or undirected) graph, in which the nodes represent the hosts/routers, and the edges represent the links between them. The *traffic pattern* $\mathcal{A}$ controls where and how do packets join the system and, in the case of static source routing, additionally defines their trajectory. The *protocol* $\mathcal{P}$ is used to schedule packets in the hosts when more than one packet wants to cross the same edge (i.e., wants to leave the host with the same next destination) at the same time step. Packets waiting to traverse an edge are kept in a queue at the tail of the edge, and the protocol determines the order in which the waiting packets cross the edge.

Then, given a network $\mathcal{G}$, a protocol $\mathcal{P}$, and an adversary $\mathcal{A}$, we say that the system $(\mathcal{G},\mathcal{A},\mathcal{P})$ is *stable* if the number of packets in the system is always bounded. If a network $\mathcal{G}$ and a protocol $\mathcal{P}$ satisfy that the system $(\mathcal{G}, \mathcal{A}, \mathcal{P})$ is stable independently of the traffic pattern $\mathcal{A}$, we say that $\mathcal{G}$ is *stable under protocol* $\mathcal{P}$. A strongest notion of stability is that of *universal stability.* Universal stability can be addressed from the network or from the protocol point of view. A network $\mathcal{G}$ is *universally stable* if the system $(\mathcal{G}, \mathcal{A}, \mathcal{P})$ is stable, whatever the traffic pattern $\mathcal{A}$ is and whatever the protocol $\mathcal{P}$ is. Similarly, a protocol $\mathcal{P}$ is said to be *universally stable* if the system $(\mathcal{G}, \mathcal{A}, \mathcal{P})$ is stable for every network $\mathcal{G}$ and traffic pattern $\mathcal{A}$.

1.2. Adversarial Models

Adversarial models have provided good theoretical frameworks for traffic pattern in modern communication networks. These models can reflect the worst case behavior of connection-oriented networks with transient connections (such as ATM networks) as well as connectionless networks (such as the Internet). Different factors are used in those adversarial models to describe the adversary and quantify its power. Three of those factors refer to the *injection rate* (i.e., the frequency at which the adversary introduces packets into the network), the *burstiness* (i.e., the maximum number of packets that can be injected in an edge in one step), and the *initial configuration* (i.e., the initial quantity and placement of packets distributed over the network at time zero).

The *Adversarial Queuing Theory* model (AQT for short) proposed by Borodin et al. in [21] is a robust model of queuing and scheduling of network traffic that can be considered as a pioneering work in studying stability via worst-case analysis. The AQT model considers the time evolution of a packet-scheduling network as a game between an adversary and a queuing protocol. The system is considered to be *synchronous.* At each time step the adversary may inject a set of packets to some of the nodes. For each packet, the adversary specifies the route that it must traverse (static source routing), after which the packet will be consumed. If more than one packet wishes to cross an edge e at the same time step, then the queuing protocol chooses one of these packets to be sent across e. The remaining packets wait in the queue. This game then advances to the next time step. The goal of the adversary is to prevent the protocol from guaranteeing load and delay bounds. The first work on AQT of Borodin et al. [21] explored mainly conditions for the stability of certain networks under different protocols. This paper was quickly followed by a work by Andrews et al. [10], which started putting a stronger focus on the scheduling protocols. Due to the important initial contributions of both works, they interfered in each other, leading to extended versions that appeared published together (consecutively, in fact) in 2001 in the *Journal of the ACM* [22, 11]. These four works established the basis of a new research topic: the study of communication systems under the AQT adversarial model.

Before the journal versions in [22, 11] appeared in 2001, other authors started to study similar issues in types of networks other than packet-switched networks. It is worth highlighting the research from Gamarnik [32, 34, 33] in studying stability of adversarial queues via fluid models. Initial results on stability for input-output blocking networks [13, 15] and session-oriented networks [9, 14] also appeared. The first attempts to study adaptive packet routing policies under adversarial models were also done at that time [1, 12], in the 4-years period between [21, 10] and [22, 11]. The survey in [45] was a useful tool for knowing the state-of-the-art at that moment.

1.3. What Lies Ahead

Maria Serna has made fundamental contributions to the study of stability of communication systems under adversarial models. In particular, she has a number of contributions to deepen into the study of the AQT model. In the following sections we will formally define this model and present Maria's contributions to the study of universal stability of networks and protocols. Then, we will revise other derived models that she proposed and the main results she obtained in them. Finally, we will revise some recent research that was influenced by her work, in order to show how alive and in how good shape this research area still is.

2. The Adversarial Queuing Theory Model

In this section, we introduce the main concepts and definitions from the adversarial queuing theory that we use in the rest of the paper. Namely, the adversarial traffic pattern for packet injection, the initial configuration of the system and the different types of trajectories that the packets describe while traversing the network towards their destination.

2.1. Adversarial Traffic Pattern

As we stated before, the AQT model assumes that the arrival of packets is controlled by an adversary. Hence, in order not to trivially overload the system, and to be able to guarantee any form of stability, it is necessary to restrict the traffic injected into the network by the adversary, i.e, the traffic pattern $\mathcal{A}$. The constraints on the traffic pattern must ensure that, over long periods of time, the maximum traffic injected in a link is no more than the amount of traffic that the link can send across. This restriction is defined with a parameter $r \in (0, 1]$, that is the *injection rate*: the maximum frequency at which the adversary can inject packets into the network. The traffic pattern is restricted by the injection rate but allows a certain level of *burstiness*. This implies that, during each interval of time, the number of packets injected into the system during that time interval requiring any edge in their trajectory cannot exceed a certain bound proportional to the length of the time interval. The way the burstiness is defined has led to two models: the windowed adversarial model and the leaky-bucket adversarial model. The windowed adversarial model was introduced by Borodin et al. [21], while the leaky-bucket adversarial model was introduced by Andrews et al. [11]. Let $N_e(I)$ be the number of packets injected by the adversary in a time interval I, whose path require to traverse a particular edge e. The two adversarial models establish the following restrictions on the number of packets injected in the system:

Definition **2.1 (Windowed adversarial model).** *An adversary in the* windowed adversarial model *is restricted by two parameters* (w, r), *where* $w \geq 1$ *is the window size and* $0 < r \leq 1$ *is the injection rate. The adversary is allowed to inject sequences of packets under the restriction that, in any interval* I *of* w *consecutive time steps, the total number of packets* $N_e(I)$ *requiring any concrete link e is, at most,* $N_e(I) \leq rw$.

Definition **2.2 (Leaky-bucket adversarial model).** *An adversary in the* leaky-bucket adversarial model *is also restricted by two parameters* (b, r), *where* $b \geq 0$ *is the burstiness and* $0 < r \leq 1$ *is the injection rate. The adversary is allowed to inject sequences of packets under the restriction that, in any interval* I, *the total number of packets* $N_e(I)$ *requiring any concrete link e is, at most,* $N_e(I) \leq r|I| + b$.

It was shown in [36] that stability of networks with fluid traffic implies stability of networks with deterministically constrained traffic, and that large bursts do not cause instability. Thus, when exploring stability properties, the burstiness can be discarded as the element that could lead the system to instability or produce any difference between the two models. It has also been shown that adversaries in the windowed and in the leaky-bucket models have the same power provided that the injection rate $r < 1$ [46]. Knowing that, most of the subsequent research has assumed the leaky-bucket definition of adversary with injection rate $r < 1$.

2.2. Initial Configuration

The systems may start with an *initial configuration*, i.e. with a set of packets that are already in the system initially, at time zero, before the dynamics of the system starts. Those packets are queued somewhere in the network and have, either a predefined path to follow (in the case of static routing), or a target node to reach (in case of dynamic routing). The size of the initial configuration is arbitrarily big, but bounded. The packets forming the initial configuration have no history, and the queue in which they are counts as their injection point, and $t = 0$ counts as their injection time. Maria et al. [4] have shown that systems with empty and systems with non-empty initial configuration are equivalent with respect to stability, since any adversary in the latter can be transformed into an adversary in the former that behaves similarly.

2.3. Packet Trajectories

The transmission of information through a network is driven by the trajectory of each individual packet that traverses it. That trajectory is a *walk* over the network graph and its nature depends on whether the network is represented by a directed or an undirected graph. Additionally, in the latter case, on whether it is allowed only unilateral or also bilateral communication between the links. Thus, different types of walks can be considered when describing the trajectory of the network packets. Such a simple observation was not taken into account before Maria started working on this area. At that time some results considering networks as directed and as undirected graphs existed, but only a single type of walk was assumed.

When representing networks by directed graphs it is very clear which is the direction of the communication: the direction defined by the orientation of the arc. Thus, the trajectory of a packet traversing a digraph from a source vertex to a destination vertex can have one of the following forms:

- the trajectory is a directed path[3]
- the trajectory is a directed simple path[4]

Observe that, in the first case, the paths can visit any vertex more than once. We assume that it does not use more than once the same outgoing edge. In the second case, no graph node can be visited more than once (and thus no edge neither) by the same path. According to this distinction, two types of adversaries can also be distinguished, which will also describe different notions of stability and universal stability:

- An *adversary* that uses packets defining paths in $\mathcal{G}$, and
- A *simple-path adversary*, which uses packets describing simple paths in $\mathcal{G}$.

The general definition of system stability can be extended in order to capture these different types of adversaries. Thus, given a digraph $\mathcal{G}$ and a protocol $\mathcal{P}$ we can say that $\mathcal{G}$ is *stable* under protocol $\mathcal{P}$ (or, alternatively, the pair $(\mathcal{G}, \mathcal{P})$ is stable) if for any adversary $\mathcal{A}$, the system $(\mathcal{G}, \mathcal{A}, \mathcal{P})$ is stable; but we can also say that $\mathcal{G}$ is *simple-path stable* under protocol $\mathcal{P}$ (alternatively, the pair $(\mathcal{G}, \mathcal{P})$ is simple-path stable) if for any simple-path adversary $\mathcal{A}$, the system $(\mathcal{G}, \mathcal{A}, \mathcal{P})$ is stable. Observe that, since simple-paths are a special case of paths, any digraph stable under a certain protocol is also simple-path stable under the same protocol. However the opposite is not true.

This distinction applies also to the stronger notion of universal stability. Thus, given a digraph $\mathcal{G}$, we say that $\mathcal{G}$ is *universally stable* if, for any protocol $\mathcal{P}$, the pair $(\mathcal{G}, \mathcal{P})$ is *stable*. Additionally, $\mathcal{G}$ is said to be *simple-path universally stable* if, for any protocol $\mathcal{P}$, the pair $(\mathcal{G}, \mathcal{P})$ is *simple-path*

[3] A *path* in a graph $G = (V, E)$ is a walk over G in which, for every appearance of the same consecutive pair of vertices v_i, v_{i+1} of V, there must exist a different edge $\{v_i, v_{i+1}\}$ or $\{v_{i+1}, v_i\}$ in the edges E of the graph (such that each appearance can be connected via a different edge). A *directed path* over a directed graph $G = (V, E)$ is defined in the same way but, for every appearance of the same consecutive pair of vertices v_i, v_{i+1} of V, a different arc $(v_i, v_{i+1}) \in E$ must exist (such that each appearance can be connected via a different arc).

[4] A *simple path* in a graph is a walk in which all the vertices are different. If the graph is directed, the walk is a directed walk.

stable. As before, observe that any universally stable digraph is also simple-path universally stable, but the opposite is not true.

When representing networks by undirected graphs it has to be made explicit, for every edge, in which direction the communication is established. We denote as $\mathcal{G}^d$ the directed version of an undirected graph $\mathcal{G}$ in which the orientation of the edges is interpreted as bilateral. We will consider three different trajectories to be defined over a given network $\mathcal{G}$ represented as an undirected graph:

- the trajectory is a path in $\mathcal{G}$
- the trajectory is a directed path in $\mathcal{G}^d$
- the trajectory is a directed simple path in $\mathcal{G}^d$

According to this distinction, three types of adversaries can also be distinguished, which will also describe different notions of stability and universal stability:

- An *adversary* that uses any path in $\mathcal{G}^d$ as packet trajectory,
- An *undirected-path adversary* uses as packet trajectory any path in $\mathcal{G}$, and
- A *simple-path adversary*, where packets follow simple paths in $\mathcal{G}^d$.

Observe that, in the first case, an edge can be used twice by the same packet provided it is traversed in opposite directions, but both directions have different queues. In the second case the same edge can be traversed only once. This corresponds with the model used by Andrews et al. in [11]. The third case does not allow a packet to pass twice through the same vertex. Notice that, in this latter model, a multi-edge can only be traversed once and only in one direction. Observe that a simple path in $\mathcal{G}$ is equivalent to a simple path in $\mathcal{G}^d$, so that fourth possible case is equivalent.

The general definition of system stability can be extended again in order to capture these three types of adversaries. Thus, given a graph $\mathcal{G}$ and a protocol $\mathcal{P}$, the pair $(\mathcal{G}, \mathcal{P})$ is stable (respectively, undirected-path stable, simple-path stable) if for any adversary (respectively, undirected-path adversary, simple-path adversary) $\mathcal{A}$, the system $(\mathcal{G}, \mathcal{A}, \mathcal{P})$ is stable (respectively, undirected-path stable, simple-path stable). If a pair $(\mathcal{G}, \mathcal{P})$ is stable then $(\mathcal{G}, \mathcal{P})$ is undirected-path stable, and if $(\mathcal{G}, \mathcal{P})$ is undirected-path stable then $(\mathcal{G}, \mathcal{P})$ is simple-path stable (these inclusions are strict). In the same way, the definition of universal stability of networks, when represented as undirected graphs, will also have these different acceptations.

3. Universal Stability of Networks

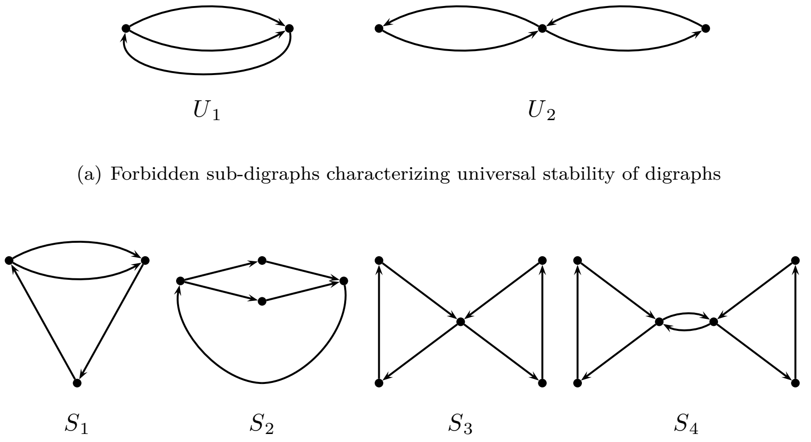
One of the first questions that arose in the community was whether it would be possible to detect stability in a networking system just from the knowledge of the topological structure of the network, and independently of the scheduling protocols used. This important issue was deeply tackled in Maria's research and she contributed to what is probably the most important result in this area: the characterization of universal stability of networks for several different types of packet trajectories and graph representations [4].

What was before?

From the network point of view, the initial work of Borodin et al. [21] contained the following basic results for the case of adversaries with injection rate $r = 1$:

- Every directed non-cyclic network is universally stable.
- Unidirectional rings with $n \geq 3$ nodes are not universally stable.
- The systems formed by unidirectional rings and rate 1 adversaries are always stable when the protocol used to schedule the packets is FTG.

The other pioneering contributions by Andrews et al. [10] dealt with adversaries with injection rate $r < 1$, where two main results on stability of networks were posted:



(a) Forbidden sub-digraphs characterizing universal stability of digraphs

(b) Forbidden sub-digraphs characterizing simple-path universal stability of digraphs

Figura 1: Characterization of the universal stability of networks (as directed graphs)

- The n-node ring is universally stable, with maximum queue-size and delay in $O(n)$.
- There exist commonly used graphs (e.g., arrays, hypercubes or the baseball graph) that are not universally stable.

What we identify as *undirected-path universal stability* of graphs was already addressed (but not with that name) in the pioneering work of Andrews et al. [11], where it was proved to be a property closed under minors, and therefore decidable in polynomial time. However no constructive algorithm was known by then. For the other modalities taken here into account, this question was unresolved.

What did Maria contribute to?

For sure, the most important result concerning the stability of networks is the full characterization of the stability property in terms of forbidden subgraphs. The characterization covers all the different types of stability discussed above, both when networks are modelled as digraphs, and when modelled as undirected graphs. This issue was an important open question in the area by the time of the development of the research, and only the work in [35] attempted to tackle it for an specific packet trajectory (i.e., the so-called simple paths). However, that attempt was disproved by the research of Maria and colleagues.

The characterizations were the base for the development of polynomial-time algorithms for deciding the property of universal stability of networks, for all its different modalities. Those algorithms were the first constructive methods existing for that aim.

3.1. Directed Graphs

As stated above, two types of universal stability can be defined when representing networks as directed graphs; namely, universal stability and simple-path universal stability. The former can be characterized with the forbidden directed graphs depicted in Figure 1(a), while the latter are characterized from the forbidden directed graphs depicted in Figure 1(b). In the following, we summarize the key theorems that conform them both.

Theorem **3.1 (Universal stability of digraphs).** *A digraph is universally stable if and only if it does not contain as sub-digraphs any of the digraphs in $\mathcal{E}(U_1) \cup \mathcal{E}(U_2)$.*[5]

[5] Given a digraph G, $\mathcal{E}(G)$ denotes the family of digraphs formed by G and all the digraphs obtained from

Instead of detecting the existence of a sub-digraph in $\mathcal{E}(U_1) \cup \mathcal{E}(U_2)$ in a given network, it is easier to think about checking rather the non-existence of those sub-digraphs.

Corollary **3.1.** *A strongly connected digraph $\mathcal{G}$ with n vertices is universally stable if, and only if, $\mathcal{G}$ is the directed cycle on n vertices.*

The universal stability of a given digraph can be decided in polynomial time since, according to Corollary 3.1, first the strongly connected components of the digraph need to be computed and second, to check whether all of them are just one directed cycle.

Theorem **3.2 (Simple-path universal stability of digraphs).** *A digraph $\mathcal{G}$ is simple-path universally stable if, and only if, $\mathcal{G}$ does not contain as a subgraph any of the graphs in $\mathcal{E}(S_1) \cup \mathcal{E}(S_2) \cup \mathcal{E}(S_3) \cup \mathcal{E}(S_4)$.*

This result has also a counterpart in terms of topological properties.

Corollary **3.2.** *A strongly connected digraph $\mathcal{G}$ is simple-path universally stable if, and only if, $\mathcal{G}$ is a subgraph of a decorated directed cycle graph.*[6]

Algorithm 1 checks the simple-path universal stability of a given strongly connected digraph $\mathcal{G}$ according to Corollary 3.2. The total execution time is polynomial.

3.2. Undirected Graphs

As introduced previously, three types of universal stability can be defined when representing networks as undirected graphs; namely, universal stability, undirected-path universal stability and simple-path universal stability. The first type can be characterized with the forbidden directed graphs depicted in Figure 2(a), the second type can be characterized with those depicted in Figure 2(b), while the third type is characterized from the forbidden directed graphs depicted in Figure 2(c). In the following, we summarize the key theorems that conform them.

Theorem **3.3 (Universal stability of graphs).** *A graph is universally stable if and only if it does not contain as subgraphs any of the graphs in $\mathcal{E}(H_1) \cup \mathcal{E}(H_2)$.*[7]

There is the corresponding result in terms of graph properties, which states the following:

Corollary **3.3.** *A graph $\mathcal{G}$ is universally stable if, and only if, all the vertices in $\mathcal{G}$ have degree at most one.*

Thus, the universal stability of a given graph can be decided in polynomial time since, according to Corollary 3.3, there is only the need to check whether the graph has a vertex with two incident edges.

Theorem **3.4 (Undirected-path universal stability of graphs).** *A graph $\mathcal{G}$ is undirected-path universally stable if, and only if, $\mathcal{G}$ does not contain as a subgraph any of the graphs in $\mathcal{E}(F_1) \cup \mathcal{E}(F_2) \cup \mathcal{E}(F_3)$*

G by successive arc or 2-cycle subdivisions. The subdivision of an arc (u, v) consists in the addition of a new vertex w and the replacement of (u, v) by the two arcs (u, w) and (w, v). The subdivision of a directed 2-cycle $\{(u, v),\ (v, u)\}$ consists in the addition of a new vertex w and the replacement of $\{(u, v),\ (v, u)\}$ by the arcs $\{(u, w), (w, u), (v, w), (w, v)\}$.

[6] A decorated directed cycle is obtained from a bilaterally oriented k-cycle with $k \geq 3$, and some oriented multi-trees after identifying one vertex from each oriented multi-tree with a vertex from the bilaterally oriented k-cycle. An oriented multi-tree is the unilateral oriented version of a multi-tree

[7] Given a graph G, $\mathcal{E}(G)$ denotes the family of graphs formed by G and all the graphs obtained from G by successive edge or 2-cycle subdivisions. The subdivision of an edge $\{u, v\}$ consists in the addition of a new vertex w and the replacement of $\{u, v\}$ by the two edges $\{u, w\}$ and $\{w, v\}$. The subdivision of an undirected 2-cycle $\{\{u, v\}, \{v, u\}\}$ consists in the addition of a new vertex w and the replacement of $\{\{u, v\}, \{v, u\}\}$ by the edges $\{\{u, w\}, \{w, u\}, \{v, w\}, \{w, v\}\}$.

Algorithm 1 : Simple-path universal stability of digraphs

```
INPUT: A strongly connected digraph $\mathcal{G}$
if $\mathcal{G}$ does not have a directed $k$-cycle with $k \geq 3$ then
  return YES
else
  Compute a directed cycle $C = (v_k, e_1, v_1, \ldots, v_{k-1} e_k, v_k)$ in $\mathcal{G}$ $(k \geq 3)$
  if any of the cycle arcs has multiplicity bigger than one then
    return NO
  end if
  Let $e'_i$ be the arc opposite to $e_i$, $1 \leq i \leq k$
  if all the arcs $e'_i$ are present in $\mathcal{G}$ and some of them have multiplicity bigger than one then
    return NO
  end if
  Let $G'$ be the digraph obtained by setting the multiplicity of all arcs in $\mathcal{G}$ to one
  and by removing the arcs in $C$ and all the opposite arcs $e'_i$ (if any).
  if there are two cycle vertices connected by a directed path in $G'$ then
    return NO
  else
    Compute the strongly connected components of $G'$
    if a strongly connected component of $G'$ contains a directed $k$-cycle with $k \geq 3$ then
      return NO
    else
      return YES
    end if
  end if
end if
```

Algorithm 2 : Sub-graph of an uni-cyclic graph

```
INPUT: A connected graph $\mathcal{G}$
Compute a spanning tree of $\mathcal{G}$
if there are more than one edge left then
  return NO
else
  return YES
end if
```

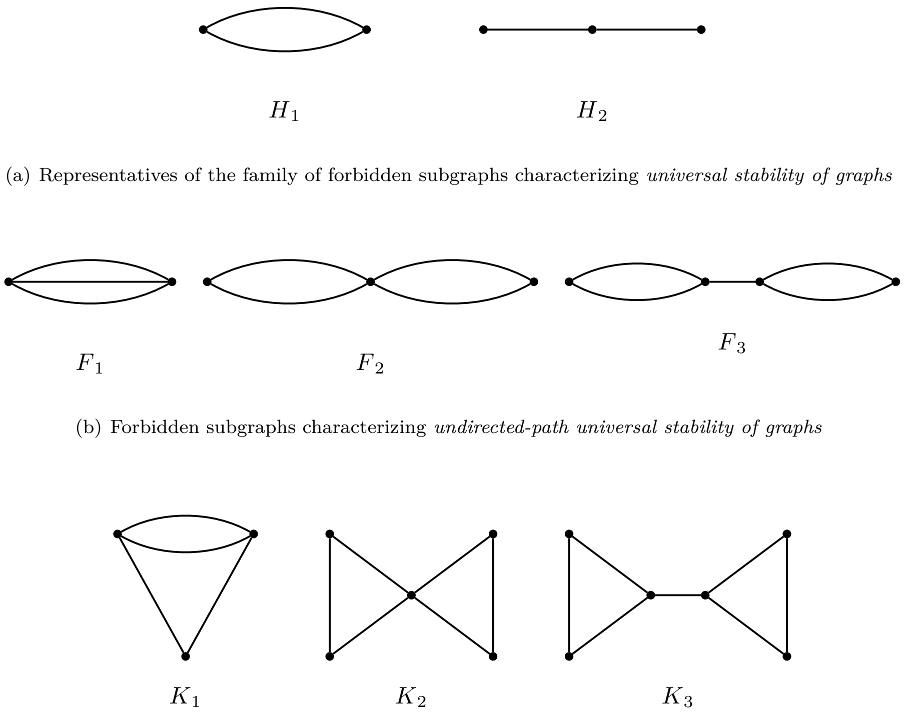
For the case of undirected-path universal stability of graphs, the corresponding graph property is the following:

Corollary **3.4.** *A connected graph $\mathcal{G}$ is undirected-path universally stable if and only $\mathcal{G}$ is a subgraph of a uni-cyclic graph.*[8]

Algorithm 2 checks whether a graph is subgraph of an uni-cyclic graph in polynomial time. Based on it, Algorithm 3 checks the undirected-path universal stability of a given connected graph $\mathcal{G}$ according to Corollary 3.4 in polynomial time.

Theorem **3.5 (Simple-path universal stability of graphs).** *A graph $\mathcal{G}$ is simple-path universally stable if, and only if, $\mathcal{G}$ does not contain as a subgraph any of the graphs in $\mathcal{E}(K_1) \cup \mathcal{E}(K_2) \cup \mathcal{E}(K_3)$.*

[8] A connected graph $\mathcal{G}$ with m edges and without multiple edges is a subgraph of an uni-cyclic graph if and only if any spanning tree of $\mathcal{G}$ has m or $m-1$ edges.



(a) Representatives of the family of forbidden subgraphs characterizing *universal stability of graphs*

(b) Forbidden subgraphs characterizing *undirected-path universal stability of graphs*

(c) Forbidden subgraphs characterizing *simple-path universal stability of graphs*

Figura 2: Characterization of the universal stability of networks (as undirected graphs)

Algorithm 3 : Undirected-path universal stability of graphs

```
INPUT: A connected graph $\mathcal{G}$
if some edge has multiplicity 3 then
  return NO
else
  Compute the connected components of $\mathcal{G}$
  if a connected component has two edges with multiplicity 2 then
    return NO
  end if
  || All the connected components have at most one edge with multiplicity 2
  if there is a connected component of $\mathcal{G}$ that is not a subgraph of an uni-cyclic graph then
    return NO
  else
    return YES
  end if
end if
```

This can be also stated from the point of view of the topological properties of the graph.

Corollary **3.5.** *A connected graph $\mathcal{G}$ is simple-path universally stable if and only if $\mathcal{G}$ is a subgraph of a decorated cycle graph.*[9]

Algorithm 4 checks the simple-path universal stability of a given connected graph $\mathcal{G}$ according

[9] A decorated cycle is obtained from a k-cycle with $k \geq 3$, and some multi-trees after identifying one vertex from each tree with a vertex from the cycle. A multi-tree is an undirected tree with multiple edges.

Algorithm 4 : Simple-path universal stability of graphs

```
INPUT: A connected graph $\mathcal{G}$
Let $G'$ be the graph obtained from $\mathcal{G}$ by setting all edge multiplicities to one
Compute the connected components of $G'$
if there is a connected component $H$ of $G'$ s.t. $H$ is not a subgraph of an uni-cyclic graph or $H$ is an uni-cyclic graph having $k \geq 3$ vertices in the cycle and with some cycle edge having multiplicity bigger than 1 in $\mathcal{G}$ then
    return NO
else
    return YES
end if
```

to Corollary 3.5 in polynomial time.

All the polynomial time algorithms described here are constructive ways of deciding the property of universal stability property in all its modalities. Note that, in case of disconnected graphs (respectively, digraphs), those algorithms must be applied to all the connected components of the graph (respectively, all the strongly connected components of the digraph). The most expensive operations are the computation of the connected components of a graph, and the computation of a spanning tree of a connected graph, but the combination of them all is still polynomial.

4. Stability under Protocols

In a network, we say that there is *contention* for a link whenever there are two or more packets at the queue associated to the link trying to cross it simultaneously. Since only one packet will get to cross the link in the next step, the rest of the packets will have to wait in a queue associated with the congested link. When congestion appears, priorities have to be given to packets in order to decide about the order in which they traverse the link. That is resolved with a *protocol* (or scheduling policy)

A protocol is said to be *greedy* if, as long as there are packets queued, at least one packet is processed at every unit of time. However, any non-greedy protocol can be simulated by a centralized greedy protocol. This greedy property is known as *work-conservation* in the queuing theory. The goal of applying a scheduling policy is mainly to provide some *quality of service* (QOS), usually by minimizing the size of the queues and/or the end-to-end delay for every packet.

Some natural greedy protocols which have been studied in the context of stability under adversarial models are:

FIFO (*first-in-first-out*), in which the packet entering first at the queue is the packet that will leave it first.[10]

LIS (*longest-in-system*), in which every queue gives priority to the packet that has been in the system the longest time. [11]

NTG (*nearest-to-go*), in which highest priority is assigned to the packet that still has to cross the smallest number of edges to reach its destination.[12]

FFS (*farthest-from-source*), in which the edge will be crossed by the packet that is farthest from its source node.[13]

[10] Counterpart, LIFO (*last-in-first-out*).

[11] Counterpart, SIS (*shortest-in-system*).

[12] Counterpart, FTG (*farthest-to-go*).

[13] Counterpart, NFS (*nearest-from-source*).

In general, and when the opposite is not explicitly said, the adversary decides how to break the ties. However, sometimes we want to impose the use of a secondary protocol that, in case of tie, decides which packet to schedule (e.g., NTG-LIS, which works as the NTG protocol, and solves ties using LIS).

What was before?

Concerning protocols, basically only the results by Andrews et al. [10] were available by the time that Maria started working on this subject. Those results can be summarized in the following four achievements:

- There exist simple greedy protocols that are universally stable (FTG, NFS, LIS, SIS). Upper bounds on the queue size and end-to-end delay are also shown for these protocols.
- There exists a simple distributed randomized greedy protocol that is universally stable with bounds on queue size and delay that are polynomial in $d \log m$, where d is the maximum path length and m is the number of edges in the network.
- There exist commonly used protocols that are not universally stable (e.g., FIFO, LIFO, FFS, NTG).

The posterior journal version [22] strengthened the results in [21] and [10] showing that, for every injection rate $0 < r < 1$ there is a queuing network for which NTG is unstable at rate r.

What did Maria contribute to?

For those queuing policies which are not universally stable, an interesting question is that of deciding the stability of a concrete network under a fixed protocol, or that of fixing bounds on the injection rate r to determine the inflexion point. Concerning bounds, Maria made important contributions on reducing the bounds for the the stability under FIFO and LIFO [29]. Even of higher interest is to be able to characterize the property of stability under a fixed protocol, i.e., to determine which networks are/are not stable when the protocol is fixed. In this sense, the main contribution of Maria is the characterization of stability under NTG-LIS and FFS [2], and the proof of its polynomial time decidability.

In the following we summarize in four items the main results concerning these achievements:

1. Stability under NTG-LIS was fully characterized, both when the networks are represented as directed graphs and as undirected graphs:

   THEOREM **4.1.** *A digraph $\mathcal{G}$ is stable (respectively, simple-path stable) under* NTG-LIS *if, and only if, $\mathcal{G}$ is universally stable (respectively, simple-path universally stable).*

   Since universal stability and simple-path stability for directed graphs can be decided in polynomial time, stability and simple-path stability under NTG-LIS can also be decided in polynomial time.

   THEOREM **4.2.** *A graph $\mathcal{G}$ is stable (respectively, undirected-path stable, simple-path stable) under* NTG-LIS *if, and only if, $\mathcal{G}$ is universally stable (respectively, undirected-path universally stable, simple-path universally stable).*

   Since universal stability, undirected-path universal stability and simple-path universal stability for undirected graphs can be decided in polynomial time, stability, undirected-path stability and simple-path stability under NTG-LIS can also be decided in polynomial time.

2. Stability under FFS was also fully characterized, both when the networks are represented as directed graphs and as undirected graphs:

   THEOREM **4.3.** *A digraph $\mathcal{G}$ is stable under* FFS *if and only if $\mathcal{G}$ is universally stable.*

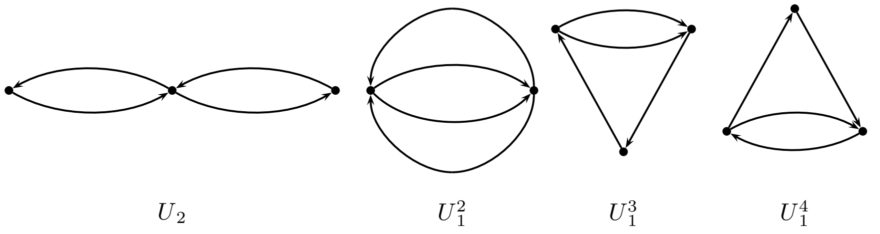


Figura 3: Smallest digraphs that are unstable under FIFO.

Based on the topological conditions determining the universal stability of directed graphs, that means that all the graphs containing, as subgraph, a graph in $\mathcal{E}(H_1) \cup \mathcal{E}(H_2)$ are not stable under FFS. As a consequence, the stability of a directed graph under FFS can be decided in polynomial time.

3. The characterisation of the set of networks that are stable under NTG-LIS and FFS turns out to be the same as the characterisation of the digraphs that are universally stable. This have some nice implications. One of them is that a digraph is universally stable if and only if it is stable under NTG-LIS or FFS. Furthermore, a digraph is stable under FFS if and only if it is stable under NTG-LIS.

4. Deciding whether a given network is stable under FIFO remained an open question for several years. Maria and her colleagues studied the stability and instability under FIFO of some networks, thus making a step further in the solving the open question. Her results allowed to fully characterise the property of stability under FIFO, and show that it is polynomial solvable by detecting the proposed forbidden subgraphs in terms of the topological properties stated here:

   THEOREM **4.4.** *Any digraph $\mathcal{G}$ is stable under* FIFO *if and only if it does not contain as subgraph a digraph from $\mathcal{E}(U_2) \cup \mathcal{E}\left(U_1^2\right) \cup \mathcal{E}\left(U_1^3\right) \cup \mathcal{E}\left(U_1^4\right)$.*

   COROLLARY **4.1.** *A strongly connected graph with 2 vertices is stable under* FIFO*, if and only if, at least one of the vertices has out-degree 1. A strongly connected digraph with more than 2 vertices is stable under* FIFO *if, and only if, it is a directed cycle.*

5. Adversarial Models Dealing with Priorities

In the previous sections we presented Maria's participation to the characterization of network and protocol stability, two research lines that were already active. In the next sections we present new research lines that were opened by Maria's work, mostly based on new models derived from AQT but never considered previously.

In this section, we revise results on generalizations of the AQT model in which packets may have different priorities [3]. This means that, whatever scheduling protocol is applied in a link, it must schedule packets of higher priority over those of smaller priority. Two different versions of the model are considered. When the priority assigned to a packet does not change over time we have the *priority model*. However, if the adversary can change the priority over time, we have the *variable priority model*. Independent on which priority model is considered, the adversary is restricted in the number of packets injected in the system by the Leaky-bucket Adversarial Model (Definition 2.2).

Observe that the AQT model considered so far is a special case of the models with priorities, in which all packets have the same priority. Hence, all the negative results (i.e., instability) that apply to AQT also apply to the priority models. The interesting question is whether positive results still hold when priorities are introduced. The first answer to that question concerns the characterization of universally stable networks and is positive. It was shown in [3] that the set of networks that are universally stable is exactly the same in the AQT model, the priority model, and the variable priority model.

THEOREM **5.1.** *Given a digraph $\mathcal{G}$, the following properties are equivalent: (1) $\mathcal{G}$ is universally stable in the* AQT *model, (2) $\mathcal{G}$ is universally stable in the priority model, (3) $\mathcal{G}$ is universally stable in the variable priority model.*

Moving to the universal stability of protocols, the first result is negative for the variable priority model. This result is not surprising, since this model gives the adversary a lot of power to control the scheduling of packets.

THEOREM **5.2.** *There is no universally stable protocol in the variable priority model.*

However, if we stick ourselves to the (non-variable) priority model, some of the popular universally stable scheduling protocols in AQT remain so.

THEOREM **5.3.** *The* FTG, NFS, *and* SIS *protocols are universally stable in the priority model.*

On the other side, LIS which is universally stable in the AQT model, is not so anymore. In fact, the following results are provided for the stability of FIFO and LIS.

THEOREM **5.4.** *Given a digraph $\mathcal{G}$ and a protocol $\mathcal{P} \in \{\text{FIFO}, \text{LIS}\}$, the pair $(\mathcal{G}, \mathcal{P})$ is stable in the priority model if, and only if, $\mathcal{G}$ is universally stable in the* AQT *model.*

Observe that this theorem implies that LIS is not universally stable, as mentioned. Additionally, it implies that stability under FIFO or LIS can be decided in polynomial time in the priority model.

6. Adversarial Models Dealing with Failures

Maria and her coauthors were the first to explore versions of the AQT model in which links and nodes may fail. (see [3, 5]). The first models in which links can fail were introduced in [3], as a natural generalization of the AQT model with priorities. In fact, two new models were proposed there which allow links to be unavailable during certain time steps. These models can be used to study dynamic networks in which the topology changes over time (e.g., wireless networks), where connections may appear and disappear as times evolves.

6.1. The Failure Model

Since it is shown that the same packet arrival and link failure patterns can be produced in the two dynamic models of [3], we present here only the so-called *failure model.* In this model, the adversary controls the link failures in addition to the injection of packets, and it is restricted by the same two parameters (b, r) as before. Let $N_e(I)$ the number of packets injected in interval I that have link e in their paths and $F_e(I)$ the number of steps in the interval I in which link e has failed (and hence no packet could cross it), it must hold that,

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

Observe that these models with link failures are related to the priority models because a step in which the link is not available is somewhat similar to a step in which a high priority packet is using it, blocking the transmission of other lower-priority packets. Not surprisingly, given this relationship, results from one model applies almost directly to the other. Hence, the main results for the failure model are the following (and are similar to the results for the priority model).

THEOREM **6.1.** *In the failure model,*

- *A digraph $\mathcal{G}$ is universally stable if, and only if, it is universally stable in the* AQT *model.*
- *The* FTG*,* NFS*, and* SIS *protocols are universally stable.*
- *Given a digraph $\mathcal{G}$ and a protocol $\mathcal{P} \in \{\text{FIFO}, \text{LIS}\}$, the pair $(\mathcal{G}, \mathcal{P})$ is stable if, and only if, $\mathcal{G}$ is universally stable in the* AQT *model.*

In the follow up work [5], this initial study of link failures is extended into a systematic exploration of the properties of systems with links and nodes failures. Moreover, multiple variants are considered, depending on whether a faulty element can still receive and transmit packets and, if so, depending on where those packets are kept. The models proposed in [5] are denoted as *edge_X_Y* and *node_X_Y*, where,

- $X \in \{R, nR\}$ indicates whether the queue of an element (link or node) can receive packets or not in a time step in which it is faulty.
- $Y \in \{B, nB\}$ indicates what happens with packets that were supposed to be received and placed in the queue of a faulty element whose queue cannot receive packets (i.e., models *edge_nR* and *node_nR*). The possibilities are that either the packets wait in the previous link of their path (option $Y = nB$), or they are received at the node and kept in an additional buffer (option $Y = B$). In this latter case, the buffer is shared by all the links of the node.

Another important assumption is that no element can be faulty more than w consecutive steps, for a given constant $w \geq 0$. Let us denote by $D_e(I)$ the number of steps in the interval I in which link $e = (u, v)$ is not available, either because the link is faulty or because node u is faulty. Then, the adversary decides the arrival and failures in the system, as long as its guarantees that, for every interval I,

$$N_e(I) \leq \lceil r(|I| - D_e(I)) \rceil + b,$$

and

$$|I| = D_e(I) \Rightarrow |I| \leq w.$$

Adversaries that satisfy this restriction are called (r, b, w)-adversaries. Many of the results presented in [5] are obtained by *simulation between models:* reproducing the arrival and failure patterns under one model in another model. Simulation, the technique introduced in [5], became a useful tool for the proof of later results.

6.2. The edge_R Model

In the failure model it was implicitly assumed that the queue of a faulty link could receive packets that would be stored there until the recovery of the link. Hence, the failure model was very similar to the *edge_R* model. In fact, (1) any valid (r, b)-adversary for the failure model is a (r, b, w)-adversary for the *edge_R* model, with $w = \lceil b/(1 - r) \rceil$, and (2) any (r, b, w)-adversary for the *edge_R* model is a valid (r, b)-adversary for the failure model, with $r = (r + w)/(w + 1)$. Hence, not surprisingly, the same results presented above for the failure model apply to the *edge_R* model.

Theorem **6.2.** *In the edge_R model,*

- *A digraph $\mathcal{G}$ is universally stable if, and only if, it is universally stable in the* AQT *model.*
- *The* FTG*,* NFS*, and* SIS *protocols are universally stable.*
- FIFO*,* LIFO*,* NTG*,* FFS*,* LIS*, and* LIQ[14] *are not universally stable.*

[14] LIQ gives priority to the packet that has waited the longest in queues [31].

6.3. The edge_nR Models

In the *edge_nR_nB* model, since the queue of a faulty link $e = (u, v)$ is not able to accept packets, then a packet p that is ready to cross a link $e' = (x, u)$ before e cannot do so. This fact has an important impact on the stability properties of networks and protocols.

THEOREM **6.3.** *In the edge_nR_nB model,*

- *Directed acyclic graphs and rings are not universally stable. More precisely, for any* $r < 1$ *there are directed acyclic graphs which are not stable under protocols* SIS*,* LIFO*,* NTG*,* NFS*,* FFS*, and* FTG*.*
- LIS *and* LIQ *are not universally stable.*

In the *edge_nR_B* model, the above packet p is able to cross link $e' = (x, u)$ although e's queue is not accesible, and it is held in a special buffer at node u until e is available. Until then, p is considered to be *blocked*. Once e is available, all the packets in the buffer are unblocked and can move to e's queue, one per time step. Not surprisingly, this model falls somehow between the previous two: rings and lines are universally stable, but there are DAGs that are not stable under a number of different protocols.

6.4. Failures of Nodes

By simulation between models, it can be shown that the *node_R_nB* model is similar to the *edge_R* model [5], leading to similar results: (1) the same set of networks is universally stable, (2) FTG, NFS, and SIS are also universally stable, and (3) FIFO, LIFO, NTG, FFS, LIS, and LIQ are not. Now, as soon as the model becomes more restrictive, the results are drastically more negative. If buffering is not available, i.e., in model *node_nR_nB*, any network with at least one link is unstable under any protocol. Buffering has an impact, because, although there are still DAGs that are not universally stable in the model *node_nR_B*, lines and rings are.

6.5. Failures of Links and Nodes

When combining failures in links and nodes, a instability result in one of the failure models involved implies instability in the combined model. However, it is not immediate that a universal stability result that holds in both (edge and node) models will hold in the combined model. The final contribution of [5] is to show that this is in fact the case for the models and results explored in the paper.

7. The Continuous Adversarial Queuing Theory Model

In the previous sections multiple aspects of the AQT model have been explored, and several variants of the basic model have been proposed an studied. However, all these models assume a synchronous evolution of the system state. They make time evolve in steps, so that in each step exactly one packet can cross each network link. This synchronous behavior implicitly means that all packets have the same length and all packets sent across all links incur the same delay. Since that is not the case in real networks, Maria and her coauthors proposed in [20] a generalization of the AQT model, that they called *Continuous* AQT (cAQT), in which packets can have different lengths, and links can have different bandwidths and latencies.

7.1. The cAQT *Model*

Formally, in the cAQT model each link e has a finite bandwidth B_e (in bits per second) and a propagation delay P_e (in seconds). [15] Similarly, each packet p has a finite length $L_p \leq L_{\text{máx}}$ (in bits). Hence, when packet p is sent across link e the total time since the transmission starts

[15] We use $X_{\text{máx}} = \text{máx}_e\{X_e\}$ and $X_{\text{mín}} = \text{mín}_e\{X_e\}$, for $X \in \{B, P\}$.

in the sending end of e until the packet is fully received in the receiving end of e is $P_e + L_e/B_e$ seconds. With the new elements of the cAQT model, the packet arrival restriction of AQT has to be adapted. Hence, the leaky-bucket adversarial model changes as follows:

Definition **7.1 (Leaky-bucket adversarial model in cAQT).** *An (r, b)-adversary in the cAQT model is restricted as follows. Let $S_e(|I|)$ be the total size (in bits) of the packets that the adversary injects in an interval I whose paths contain edge e. Then,*

$$S_e(|I|) \leq r|I|B_e + b,$$

for all I and all e.

As in AQT, the adversary is restricted by the burstiness $b \geq 0$ and the injection rate $0 < r \leq 1$. However, in cAQT the adversary is allowed to inject packets under a load condition that is defined at the bit level, instead of the packet level. Observe that Definition 7.1 applied to very small intervals imposes an upper limit on the length of any packet of $L_{\text{mín}} = b$ bits.

Interestingly, the definitions of universal stability of protocols and networks of AQT apply verbatim to cAQT. Also, since an AQT system can be seen as a special case of cAQT, all negative results for AQT apply to cAQT. However, the positive results do not apply directly, and have to be proven again.

7.2. Universal Stability of Networks

The following are some results on universal stability of networks that apply to both AQT and cAQT.

Theorem **7.1.** *Let $\mathcal{G}$ be a directed acyclic graph, $\mathcal{A}$ any (r, b)-adversary with $r \leq 1$ and $b \geq 1$, and $\mathcal{P}$ any greedy scheduling policy. The system $(\mathcal{G}, \mathcal{A}, \mathcal{P})$ is stable.*

Theorem **7.2.** *Let $\mathcal{G}$ be the n-node directed ring, $\mathcal{A}$ any (r, b)-adversary with $r = 1 - \varepsilon < 1$ and $b \geq 1$, and $\mathcal{P}$ any greedy scheduling policy. The system $(\mathcal{G}, \mathcal{A}, \mathcal{P})$ is stable, there are never more than*

$$\frac{b + 2P_{max}B_{max} + L_{max}}{\varepsilon B_{min}}(1 - \varepsilon)nB_{max} + b$$

bits in the system that require any given edge, and no packet spends more than

$$\frac{b + 2P_{max}B_{max} + L_{max}}{\varepsilon^2 B_{min}^2}nB_{max} + \frac{b}{\varepsilon B_{min}} + P_{max}$$

time in the system.

And it was also proven that the same set of networks is universally stable in AQT and in cAQT.

Theorem **7.3.** *A network $\mathcal{G}$ is universally stable in the cAQT model if and only if it is universally stable in AQT. Hence, the property of universal stability of networks in the cAQT model can be decided in polynomial time.*

7.3. Universal Stability of Protocols

The following are results of universal stability of protocols. The protocols presented, namely LIS, SIS, FTG, and NTG, were stable under the AQT model and remain stable under cAQT.

Theorem **7.4.** *Let $\mathcal{G}$ be the a network, $\mathcal{A}$ any (r, b)-adversary with $r = 1 - \varepsilon < 1$ and $b \geq 1$, and d the length of the longest simple directed path in $\mathcal{G}$. The system $(\mathcal{G}, \mathcal{A},$ LIS$)$ is stable, all packets spend less than*

$$\frac{b/B_{min} + P_{max}}{r\varepsilon^d}$$

time in the system, and there are always less than

$$\frac{b/B_{min} + P_{max}}{\varepsilon^d}B_{max} + b$$

bits in the system trying to cross any edge e.

Theorem **7.5.** *Let $\mathcal{G}$ be the a network, $\mathcal{A}$ any (r, b)-adversary with $r = 1 - \varepsilon < 1$ and $b \geq 1$, and d the length of the longest simple directed path in $\mathcal{G}$. Then, the systems $(\mathcal{G}, \mathcal{A}, \text{SIS})$, $(\mathcal{G}, \mathcal{A}, \text{FTG})$, and $(\mathcal{G}, \mathcal{A}, \text{NFS})$ are all stable,*

7.4. Instability of Protocols

Now, new protocols are presented that use to schedule packets taking into account the three new parameters introduced in cAQT versus AQT: packet length, link bandwidth, and link propagation delay. The protocols are the following,

- LPL-LIS (longest-packet-length) scheduling policy, which gives priority to the packet with longest length, and breaks ties according to LIS.
- SPL-NFS (slowest-previous-link) scheduling policy, which gives priority to the packet whose last crossed link was the slowest (i.e., had the smallest bandwidth), and breaks ties according to NFS.
- SPP-NFS (smallest-previous-propagation) scheduling policy, which gives priority to the packet whose previously traversed edge had the smallest propagation delay, and breaks ties according to NFS.

It was proven by Maria and her coauthors in [20] that these policies are not universally stable in cAQT while they are in AQT. To see that these protocols are stable in AQT it is enough to observe that in this model all packets have the same length, and all links the same bandwidth and propagation delay. Hence they become LIS and NFS, which are universally stable.

All the instability proofs use the same network topology, the baseball network introduced in [11], augmented with 4 incoming edges. This network $\mathcal{G}$ is the graph with nodes

$$V(\mathcal{G}) = \{v_0, v_1, w_0, w_1, v'_0, v'_1, w'_0, w'_1\},$$

and edges

$$E(\mathcal{G}) = \{(v_0, w_0), (v_1, w_1), (w_1, v_0), (w_1, v_0), (w_0, v_1),$$
$$(w_0, v_1), (v'_0, v_0), (v'_1, v_1), (w'_0, w_0), (w'_1, w_1)\}.$$

Theorem **7.6.** *Assume all the edges in $\mathcal{G}$ have bandwidth 1 and zero propagation delay. For any $r > 1/\sqrt{2}$ there is an (r, b)-adversary $\mathcal{A}$ that makes the system $(\mathcal{G}, \mathcal{A}, \text{LPL-LIS})$ unstable only with packets of length 1 and 2.*

Theorem **7.7.** *Assume that in $\mathcal{G}$ all the edges incoming to v_0 and v_1 have bandwidth 2, while the rest have bandwidth 1. All the edges have zero propagation delays. For any $r > 1/\sqrt{2}$ there is an (r, b)-adversary $\mathcal{A}$ that makes the system $(\mathcal{G}, \mathcal{A}, \text{SPL-NFS})$ unstable.*

Theorem **7.8.** *Assume that in $\mathcal{G}$ all the edges incoming to v_0 and v_1 have propagation delay 1, while the rest have zero propagation delay. All the edges have bandwidth 1. For any $r > 1/\sqrt{2}$ there is an (r, b)-adversary $\mathcal{A}$ that makes the system $(\mathcal{G}, \mathcal{A}, \text{SPP-NFS})$ unstable.*

8. Research Derived from Maria's Work in the Last Decade 2009 − 2019

After the publication of the first papers on the AQT model [21, 10], it came a lot of activity in the study of this model, peaking around the middle of the first decade of the 2000s. The collection of results above were part of that activity in adversarial models, and its impact measured in publications citing them also started to decline with the end of the first decade of the 21st century. However, even after the burst of excitement on AQT ended, research based on AQT and the above papers has continued at a lower but steady pace. This shows that these models and results are fundamental and they are not a simple fashionable area of research. Hence, we have explored the work published in the latest decade (2009 − 2019) that has been influenced by the papers and results presented in the previous sections. We have chosen a few lines of research, aggregated into collections of papers with a similar theme, and a few interesting isolated papers of especial relevance.

The first interesting new line of research has been the definition and study of *adversarial models for radio networks* with multiple access channels, prone to collision when packets are transmitted. In this line, Chlebus et al. defined a model, deeply inspired in AQT and the results presented above, for a single multiple access channel [26]. In this model, a restricted adversary injects packets to a set of n synchronous stations, and the stations send these packets over the channel, one packet at a time. The challenge is to propose distributed protocols that allow the stations to coordinate, so that stability is achieved (i.e., the total number of packets in the stations queues is bounded). They consider variants of this basic model depending on, e.g., whether the stations can detect collisions or they can attach control data to the packets they transmit. Beyond stability of a protocol at a given rate, they measure performance considering bounds on the queue sizes and on the time packets spend in the system (that they call latency).

For this model they study deterministic protocols and show in [25] that there are protocols that are stable for injection rate 1, even without collision detection, but with unbounded latency. They also show that achieving simultaneously stability and bounded latency is not possible for rate 1 in the leaky-bucket injection model, even with collision detection and using control data. On the other hand, in the windowed adversarial model, they show in [8] the existence of protocols that are stable at rate 1 and have bounded latency. The protocols are not adaptive, and they do neither require collision detection nor control data. Returning to the leaky-bucket adversary, in [6] a collection of deterministic protocols are studied and their latency is characterized as a function of the adversary rate and burstiness. The paper also compares by simulation these protocols with back-off randomized protocols. One extension of this line of work is a multiple access channel model in which the leaky-bucket adversary can jam the channel with a rate λ [7]. In this model, it is shown that stability and bounded latency can be achieved as long as $\rho + \lambda < 1$.

The natural generalization of the single multiple channel was explored in [24], in which multiple adversarial access channels are combined into a multihop radio network model. In this model, the scheduling policy at every node is combined with a transmission policy for the radio channel. This affects, for instance, the stability properties of LIS and SIS, which are universally stable in AQT.

Strongly inspired in the continuous model described in Section 7, another model for radio networks has been proposed in [30]. In this model, mobile clients arrive and leave, so that each has to be assigned to one of a subset of base stations, with a certain bandwidth. The adversary controls the arrival and departure of clients subject to admissibility restrictions. The problem, termed Station Assignment problem, is to decide the assignment of each client to a base station and to schedule its transmissions so there are no collisions. Impossibility conditions and algorithms are presented for this problem under a number of conditions.

Returning to wired networks, another line of research has explored the relation between the theoretical results in AQT and real networks. For instance, Berger et al. [17] introduced AQT elements in the OMNeT++ simulator, and were able to reproduce by simulation some of the analytical instability results presented in Section 3. They found interesting behaviors not present in the analysis, like the interplay among the initial system state, the injection rate, and some added randomness. In similar works [16, 18], the impact of bounded buffer sizes, packet losses, and synchronization of the adversary at different nodes is studied via simulations and analysis. The lack of synchronization among the networks nodes is also studied in [31] via analysis and simulations.

Modern large-scale multimedia networks, such as the Internet, are characterized by heterogeneity due to the versatile nature of their communication subsystems. That heterogeneity comes partially from the simultaneous running composition of different contention-resolution protocols over different network hosts and the existence of various types of network links. The works by Koukopoulos in the latest decade focused strongly in that matter. The study of stability on networks with fixed bi-modal link capacities [43, 44] started the way towards the consideration of networks with more dynamic properties and behaviors. In the latest decade, that research line was exhaustively studied (see, e.g., [39, 40, 41, 42]). Probably, the most significant conclusions of those works are, that the stability properties of FIFO networks are not preserved when FIFO is composed with NTG, and that some packet-trajectory variations of the property of stability can be fully characterized when compositions of NTG with other specific protocols are considered. In general, having compositions of protocols leads networks to worst instability behaviors than using a single protocol, and also to worst instability behaviors than using a composition of protocols but fixed network link capacities or slowdowns. Dynamic adversarial attacks influencing the capacities and slowdowns of the network links turn out to be very harmful (attacks changing link slowdowns are even more than attacks changing link capacities) for specific protocol compositions.

Other interesting applications of AQT and the results presented are the evaluation of the use of SIS, LIS, NFS, and FTG to schedule autonomous vehicles at road intersections [48]. Somewhat related is the generalization of AQT proposed in [37], since it can also be used to model much more than packets networks [38]. In this model, packets have types that can change as they move along their paths, and a link needs a setup time to switch from serving a type to another. Then, a link has to decide both its switching policy and its scheduling policy. This model can be used to model packet priorities and much more, since it gives the adversary strictly more power than the basic AQT model.

One more dimension that has been combined with AQT is energy efficiency. In [47] a new energy-aware model that generalizes the continuous AQT (cAQT) model is presented. In this model, links are allowed to change dynamically their operational speed (bandwidth) and are assumed to consume a power that depends superlinearly with the bandwidth used. The authors show that there are universally stable policies that are also energy-efficient.

In summary, the AQT model and Maria's contributions to its study keep inspiring new research lines and adversarial models. We believe this trend will continue for the many years to come.

References

[1] W. Aiello, E. Kushilevitz, R. Ostrovsky, and A. Rosén. Adaptive Packet Routing for Bursty Adversarial Traffic. *Journal of Computer System Sciences*, 60(3):482–509, 2000.

[2] C. Àlvarez, M. Blesa, J. Díaz, A. Fernández, and M. Serna. The complexity of deciding stability under FFS in the adversarial model. *Information Processing Letters*, 90(5):261–266, 2004.

[3] C. Àlvarez, M. Blesa, J. Díaz, A. Fernández, and M. Serna. Adversarial models for priority-based networks. *Networks*, 45(1):23–35, 2005.

[4] C. Àlvarez, M. Blesa, and M. Serna. A characterization of universal stability in the adversarial queueing model. *SIAM Journal on Computing*, 34(1):41–66, 2004.

[5] C. Àlvarez, M. Blesa, and M. Serna. The robustness of stability under link and node failures. *Theor. Comput. Sci.*, 412(50):6855–6878, 2011.

[6] L. Anantharamu, B.S. Chlebus, D.R. Kowalski, and M.A. Rokicki. Deterministic broadcast on multiple access channels. In *2010 Proceedings IEEE INFOCOM*, pages 1–5. IEEE, 2010.

[7] L. Anantharamu, B.S. Chlebus, D.R. Kowalski, and M.A. Rokicki. Packet latency of deterministic broadcasting in adversarial multiple access channels. *Journal of Computer and System Sciences*, 99:27–52, 2019.

[8] L. Anantharamu, B.S. Chlebus, and M.A. Rokicki. Adversarial multiple access channels with individual injection rates. *Theory of Computing Systems*, 61(3):820–850, 2017.

[9] M. Andrews. Instability of FIFO in session–oriented networks. *Journal of Algorithms*, 50(2):232–245, 2004.

[10] M. Andrews, B. Awerbuch, A. Fernández, J. Kleinberg, T. Leighton, and Z. Liu. Universal stability results for greedy contention–resolution protocols. In *37th. IEEE Symposium on Foundations of Computer Science* (FOCS'96), pages 380–389. IEEE Computer Society Press, 1996. Preliminary version of [11].

[11] M. Andrews, B. Awerbuch, A. Fernández, J. Kleinberg, T. Leighton, and Z. Liu. Universal stability results for greedy contention–resolution protocols. *Journal of the ACM*, 48(1):39–69, 2001.

[12] M. Andrews, A. Fernández, A. Goel, and L. Zhang. Source routing and scheduling in packet networks. *Journal of the ACM*, 52(4):582–601, 2005.

[13] M. Andrews and L. Zhang. Stability results for networks with input and output blocking. In *30th. Annual ACM Symposium on Theory of Computing* (STOC'98), pages 369–377. ACM Press New York, 1998.

[14] M. Andrews and L. Zhang. The effects of temporary sessions on network performance. In *11th. Annual ACM-SIAM Symposium on Discrete Algorithms* (SODA'00), pages 448–457. ACM Press New York & Society for Industrial and Applied Mathematics, 2000.

[15] M. Andrews and L. Zhang. Achieving stability in networks of input-queued switches. *IEEE/ACM Transactions on Networking*, 11(5):848–857, 2003.

[16] D. Berger. Effects and factors of network instability. Bachelor's thesis, Department of Computer Science, Technical University of Kaiserslautern, Kaiserslautern, Germany, 1 2012.

[17] D. Berger, M. Karsten, and J. Schmitt. Simulation of adversarial scenarios in OMNeT++: Putting adversarial queueing theory from its head to feet. In *Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques*, pages 291–298, 2013.

[18] D. Berger, M. Karsten, and J. Schmitt. On the relevance of adversarial queueing theory in practice. *ACM SIGMETRICS Performance Evaluation Review*, 42(1):343–354, 2014.

[19] M. Blesa. *Stability in Communication Networks under Adversarial Models.* PhD thesis, Universitat Politècnica de Catalunya – BarcelonaTech, 2006.

[20] M. Blesa, D. Calzada, A. Fernández, L. López, A.L. Martínez, A. Santos, M. Serna, and C. Thraves. Adversarial queueing model for continuous network dynamics. *Theory of Computing Systems*, 44(3):304–331, 2009.

[21] A. Borodin, J. Kleinberg, P. Raghavan, M. Sudan, and D. Williamson. Adversarial queueing theory. In *28th. Annual ACM Symposium on Theory of Computing* (STOC'96), pages 376–385. ACM Press New York, 1996. Preliminary version of [22].

[22] A. Borodin, J. Kleinberg, P. Raghavan, M. Sudan, and D. Williamson. Adversarial queueing theory. *Journal of the ACM*, 48(1):13–38, 2001.

[23] M. Bramson. Instability of FIFO queuing networks. *Annals of Applied Probability*, 4(2):414–431, 1994.

[24] B.S. Chlebus, V. Cholvi, and D.R. Kowalski. Universal routing in multi-hop radio networks. *arXiv preprint arXiv:1606.03571*, 2016.

[25] B.S. Chlebus, D.R. Kowalski, and M.A. Rokicki. Maximum throughput of multiple access channels in adversarial environments. *Distributed Computing*, 22(2):93–116, 2009.

[26] B.S. Chlebus, D.R. Kowalski, and M.A. Rokicki. Adversarial queuing on the multiple access channel. *ACM Transactions on Algorithms (TALG)*, 8(1):5, 2012.

[27] R.L. Cruz. A calculus for network delay. Part I: Network elements in isolation. *IEEE Transactions on Information Theory*, 37:114–131, 1991.

[28] R.L. Cruz. A calculus for network delay. Part II: Network analysis. *IEEE Transactions on Information Theory*, 37:132–141, 1991.

[29] J. Díaz, D. Koukopoulos, S. Nikoletseas, M. Serna, P. Spirakis, and D. Thilikós. Stability and non-Stability of the FIFO Protocol. In *13th annual ACM Symposium on Parallel Algorithms and Architectures* (SPAA'01), pages 48–52, Crete Island, Greece, 2001. ACM Press New York.

[30] A. Fernández Anta, D.R. Kowalski, M.A. Mosteiro, and P.WH. Wong. Scheduling dynamic parallel workload of mobile devices with access guarantees. *ACM Transactions on Parallel Computing*, 5(2), 2019.

[31] A. Fernández Anta, J.L. López-Presa, M.A. Lorenzo, P. Manzano, J. Martinez-Romo, A. Mozo, and C. Thraves. Performance of scheduling policies in adversarial networks with non-synchronized clocks. *Theory of Computing Systems*, 48(1):1–22, 2011.

[32] D. Gamarnik. Stability of adversarial queues via fluid models. In *39th. Annual IEEE Symposium on Foundations of Computer Science* (FOCS'98), pages 60–70, Palo Alto, USA, 1998. IEEE Computer Society Press.

[33] D. Gamarnik. Using fluid models to prove stability of adversarial queueing networks. *IEEE Transactions on Automatic Control*, 4:741–747, 2000.

[34] D. Gamarnik. On deciding stability of some scheduling policies in queueing systems. *Mathematics of Operations Research*, 27(2):272–294, 2002.

[35] A. Goel. Stability of networks and protocols in the adversarial queueing model for packet routing. *Networks*, 37(4):219–224, 2001.

[36] B. Hajek. Large bursts don't cause instability. *IEEE Transactions on Automatic Control*, 45:116–118, 2000.

[37] M. Kiwi, M. Soto, and C. Thraves. Adversarial queuing theory with setups. *Theoretical Computer Science*, 410(8-10):670–687, 2009.

[38] M.A. Kiwi and A. Russell. The chilean highway problem. *Theor. Comput. Sci.*, 326(1-3):329–342, 2004.

[39] D. Koukopoulos. Stability in heterogeneous multimedia networks under adversarial attacks. *J. UCS*, 15(2):444–464, 2009.

[40] D. Koukopoulos. The impact of dynamic adversarial attacks on the stability of heterogeneous multimedia networks. *Computer Communications*, 33(14):1695–1706, 2010.

[41] D. Koukopoulos. Instability behaviour of heterogeneous multimedia networks under dynamic adversarial attacks. *Mathematical and Computer Modelling*, 57(11-12):2671–2684, 2013.

[42] D. Koukopoulos. Stability in heterogeneous dynamic multimedia networks. In *Algorithms, Probability, Networks, and Games*, pages 261–280. Springer-Verlag GmbH, 2015.

[43] D. Koukopoulos, M. Mavronicolas, and P. Spirakis. Instability of networks with quasi-static link capacities. In *10th Internaltional Colloquium on Structural Information Complexity* (SIROCCO'03), volume 17 of *Proceedings in Informatics*, pages 179–194. Carleton Scientific, 2003.

[44] D. Koukopoulos, M. Mavronicolas, and P. Spirakis. Performance and stability bounds for dynamic networks. In *7th International Conference on Parallel Architectures, Algorithms and Networks* (I-SPAN'04), pages 239–246. IEEE Computer Society Press, 2004.

[45] M. Mavronicolas. Stability in routing: Networks and protocols. *Bulletin of the European Association on Theoretical Computer Science (EATCS)*, 74:119–133, 2001.

[46] A. Rosén. A note on models for non-probabilistic analysis of packet switching networks. *Information Processing Letters*, 84(5):237–240, 2002.

[47] Y. Shi, F. Zhang, and Z. Liu. A universally stable and energy-efficient scheduling protocol for packet switching network. In *2013 IEEE 12th International Symposium on Network Computing and Applications*, pages 118–126. IEEE, 2013.

[48] M. Vasirani and S. Ossowski. Evaluating policies for reservation-based intersection control. In *Proceedings of the 14th Portuguese Conference on Artificial Intelligence*, volume 14, pages 39–50, 2009.