Available online at www.sciencedirect.com
SCIENCE DIRECT®

Information Processing Letters 90 (2004) 261–266

Information Processing Letters

www.elsevier.com/locate/ipl

The complexity of deciding stability under FFS in the Adversarial Queueing model ☆

C. Àlvarez [a], M. Blesa [a], J. Díaz [a,*], A. Fernández [b], M. Serna [a]

[a] *Dept. Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, Campus Nord, E-08034 Barcelona, Spain*
[b] *Grupo de Sistemas y Comunicaciones, Universidad Rey Juan Carlos, Campus de Mostoles, E-28933 Madrid, Spain*

Received 24 February 2003; received in revised form 27 October 2003

Communicated by K. Iwama

Abstract

We address the problem of deciding whether a given network is stable in the Adversarial Queueing Model when considering *farthest-from-source* (FFS) as the queueing policy to schedule the packets through its links. We show a characterisation of the networks which are stable under FFS in terms of a family of forbidden subgraphs. We show that the set of networks stable under FFS coincide with the set of universally stable networks. Since universal stability of networks can be checked in polynomial time, we obtain that stability under FFS can also be decided in polynomial time.

MSC: 68Q25; 68R10; 90B12; 90B15; 90B22

Keywords: Graph algorithms; Packet-switched interconnection networks; Interconnection networks; Adversarial Queueing Theory; Greedy scheduling protocols; Network stability

☆ Work partially supported by the FET Program of the EU under contract number IST-2001-33116 (FLAGS), and by the Spanish CICYT project TIC-2001-4917-E. The first, second and fifth author were also supported by the Spanish CICYT project TIC2002-04498-C05-03 (TRACER). The second author was also supported by the Catalan Research Council of the Generalitat de Catalunya under grant 2001FI-00659. The third author was also supported by the *Distinció per a la recerca* (research distintion prize) of the Generalitat de Catalunya. The fourth author was partially supported by the Spanish MCyT under grant TIC2001-1586-C03-01, the Comunidad de Madrid under grant 07T/0022/2003, and the Universidad Rey Juan Carlos under grant PPR-2003-37.

* Corresponding author.

E-mail addresses: alvarez@lsi.upc.es (C. Àlvarez), mjblesa@lsi.upc.es (M. Blesa), diaz@lsi.upc.es (J. Díaz), anto@gsyc.escet.urjc.es (A. Fernández), mjserna@lsi.upc.es (M. Serna).

0020-0190/$ – see front matter
doi:10.1016/j.ipl.2004.02.016

1. Introduction

In recent times, an important model to study stability and load balancing issues in non-adaptive routing has been the Adversarial Queueing Theory (AQT) model proposed by Borodin et al. [4]. *Stability* refers to the fact that the number of packets in the system remains bounded as the system dynamically evolves in time. This bound, which can be a function of system parameters, is not dependent on time. Stability is studied considering that a synchronous communication system $(G, \mathcal{A}, \mathcal{P})$ is formed by three main components: the network G, the traffic pattern defined by $\mathcal{A}$, and the scheduling protocol $\mathcal{P}$. *Networks* are modelled with (directed or undirected) graphs in which nodes represent the hosts and edges represent the links between these hosts. The *traffic pattern* controls where and how packets join the system and defines their trajectory. The *protocol* determines the order in which the packets requiring to cross a link are scheduled to be forwarded.

Adversarial models have been shown to be good theoretical frameworks for traffic pattern in modern communication networks, since they can reflect the behaviour of connection-oriented networks with transient connections, such as ATM networks, as well as connection-less networks, such as the Internet. Adversarial models allow to analyse the system in a worst-case scenario, since they have replaced traditional stochastic arrival assumptions in the traffic pattern by worst-case inputs. Recent research on stability has mainly considered adversarial models.

The AQT model considers the time evolution of a packet-switched network as a game between an adversary and a queueing policy or protocol. The system is synchronous, i.e., there is a global notion of (discrete) *time step*. The adversary controls the traffic pattern by injecting at each time step a set of packets into the system. In this work, we consider *static packet routing*; in this setting, the adversary also specifies for each packet the complete path that the packet must traverse. The protocol schedules one step in the advance of the packets and then, the game goes on to the next time step. Since it is mainly interesting to study stability conditions in under-loaded packet networks, the power of the adversary is constrained in order not to trivially collapse the system by exceeding the capacity of the links. Moreover, packets do not remain in the system once they arrive to their destination. In general, at any interval of time I, the number of packets that the adversary can inject into the network which require to traverse any link e in their trajectory, cannot exceed a certain bound proportional to the length of the time interval. This bound is set to be $\lceil r|I|\rceil + b$, where $0 < r < 1$ is the *injection rate* (i.e., the frequency at which the adversary introduces packets into the network), and $b \geqslant 0$ is the *burstiness* (i.e., the maximum excess of packets that can be injected in one step requiring any particular link).

(*Store and forward*) *greedy protocols* are those that forward a packet across an edge e whenever there is at least one packet waiting to traverse the edge e. Three types of packets may wait to traverse an edge in a particular instant of time: the incoming packets arriving from adjacent edges, the packets injected directly into the edge, and the packets that could not be forwarded in previous steps. Since we consider unit-capacity edges, at each step of time only one packet from those ones is forwarded through the edge; the rest are kept in a queue at the head of the edge. In this work, we consider greedy queueing policies.

Some natural greedy protocols are *first-in-first-out* (FIFO), in which highest priority is given to the packet that has arrived first in the queue, and *farthest-from-source* (FFS), in which highest priority is assigned to the packet that is farthest from its source node. Other protocols are *last-in-first-out* (LIFO), *nearest-from-source* (NFS), *nearest-to-go* (NTG), *farthest-to-go* (FTG), *shortest-in-system* (SIS), and *longest-in-system* (LIS).

Universal stability and stability under a protocol. 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 G is universally stable if, for any protocol and any adversary, the resulting system is stable. A protocol $\mathcal{P}$ is universally stable if, for any network and any adversary the resulting system is stable.

Concerning the network point of view, it is known that the property is completely characterised and that deciding universal stability of networks can be solved in polynomial time, even under different network representation and packet trajectories [1,2]. Concerning the protocol point of view, it is known that FTG, NFS, SIS and LIS are universally stable, while FIFO, LIFO, NTG and FFS are not [3].

For those queueing polices which are not universally stable, a weaker notion of stability (the *stability under a protocol*) is addressed. Here the problem is to decide whether a given network is stable or not under a fixed queueing policy. Only one result is known: Deciding stability under NTG-LIS[1] is polynomially solvable and is equivalent to decide universal stability of networks [2].

In this paper we address the stability problem when the selected queueing policy is FFS. We show the polynomial time decidability of the stability under FFS in the general case in which the adversary can solve ties arbitrarily. Our main result shows a characterisation of the set of networks that are stable under FFS by identifying a family of forbidden subgraphs. Interestingly enough, the characterisation we obtain is 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 FFS.

An interesting open problem is to analyse the complexity of deciding stability under other protocols, in particular under the popular FIFO and LIFO protocols.

2. Preliminaries

The FFS protocol gives priority to the packet in the queue which is farthest (in terms of distance) from its source node. When ties among packets with the same priority happen, we assume that they are broken *arbitrarily* by the adversary.

We study the complexity of deciding stability under FFS in the adversarial queueing model. To characterise the property of stability under FFS, first we need to identify the families of digraphs which are stable under this protocol. Then, the simplest digraphs which are not stable should be identified. Moreover, by iteratively applying subdivision operations to those simplest digraphs, we must "extend" them to define a family of digraphs. Stability under FFS will be characterised once it is shown that those extensions are not stable either.

Before formally stating our results, we need to introduce some theoretical definitions over digraphs. All the digraphs considered in this paper may have multiple edges (arcs) but no loops.[2] The packets transmitted over those digraphs follow predefined *path* trajectories, which might repeat vertices but not edges. We consider the following subdivision operations over digraphs:

- The *subdivision of an arc* (u, v) in a digraph G 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* 2-*cycle* (u, v), (v, u) in a digraph G 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) and (w, v).

Given a digraph G, $\mathcal{E}(G)$ denotes the family of digraphs formed by G and all the digraphs obtained from G by successive arc or 2-cycle subdivisions.[3] Given a family of digraphs $\mathcal{F}$, let us denote as $\mathcal{S}(\mathcal{F})$ the family of digraphs that contain a graph in $\mathcal{F}$ as a subgraph.

Known results. Concerning the universal stability of protocols, it was already shown in [3] that the FFS protocol is not universally stable. Concerning the universal stability of networks, the property was characterised in [2] in terms of the forbidden subdigraphs for different packet trajectories. Fig. 1(a) provides the two basic forbidden digraphs needed to characterise that property when the packets follow a path trajectory, and Fig. 1(b) gives the shape of the extensions by (arc and 2-cycle) subdivisions of those graphs. When packets follow a path trajectory, this basic family provides the characterisation of the universal stability of networks. This result is stated in the following theorem.

Theorem 1 [2]. *A digraph G is universally stable if and only if $G \notin \mathcal{S}(\mathcal{E}(U_1) \cup \mathcal{E}(U_2))$.*

In the same work, it was also shown that this property can be checked in polynomial time.

[1] The protocol NTG-LIS works as NTG, but solves ties using the LIS protocol.

[2] Multiple edges share the same pair of different endpoints. The endpoints of a loop are a unique same vertex.

[3] Observe that, for a graph G, $\mathcal{E}(G)^d \subseteq \mathcal{E}(G^d)$, but it might be the case that $\mathcal{E}(G)^d \neq \mathcal{E}(G^d)$.

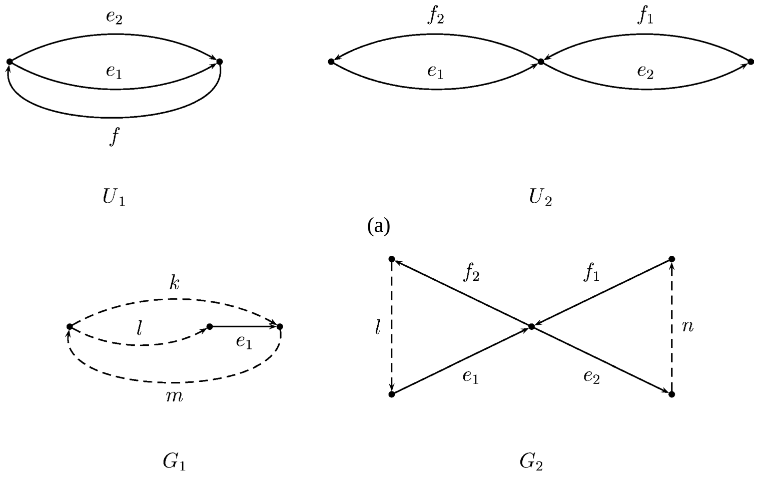

Fig. 1. Representatives of the family of digraphs characterising universal stability. (a) The two basic forbidden sub-digraphs. (b) Extensions by arc subdivision of U_1 and U_2.

3. Stability under FFS

We analyse the complexity of deciding whether a given network G is stable under FFS. We express the stability of a system in the same way as in [3]: given a digraph G, a queueing policy $\mathcal{P}$, and a given adversary $\mathcal{A}$, we say that the system $(G, \mathcal{A}, \mathcal{P})$ is *stable* if, at any time step, the maximum number of packets in the system is bounded. Instability can be expressed just with a pair: given a digraph G and queueing policy $\mathcal{P}$, the pair $(G, \mathcal{P})$ is *not stable* if there exists an adversary $\mathcal{A}$ such that the system $(G, \mathcal{A}, \mathcal{P})$ is not stable. We say that a digraph G is *stable under* FFS if, for any adversary $\mathcal{A}$, the system $(G, \mathcal{A}, \text{FFS})$ is stable.

All acyclic digraphs and directed cycles on any number of vertices are known to be universally stable [4,3]. Digraphs formed by connecting acyclically two universally stable sub-digraphs, are also universally stable [5]. Thus, those networks are also stable under the FFS protocol. The next networks to consider are then those depicted in Fig. 1. By Theorem 1 we know that they are not universally stable. However in [2] it is shown that they are not stable under NTG-LIS, but nothing is known about its stability when the protocol is FFS. We prove that they are not stable under FFS.

We show first the instability of the two basic digraph in Fig. 1(a), and then the instability of their extensions in Fig. 1(b). All our instability proofs are based on induction. A set of rounds compose a step of the induction reasoning. The goal is to demonstrate that the number of packets in the system can increase from one step to the next (and, by applying the inductive hypothesis, they can increase infinitely). The configuration of the system at the end of every step must be the same as at the beginning (in terms of the type of packets and their location). For the sake of simplicity, we only reproduce the inductive step and sometimes we omit some additive constants in our analysis, however, those omissions will not change the final result.

Lemma 1. *The pair* (U_1, FFS) *is not stable.*

Proof. Initially we have a set S of s packets trying to cross the path fe_1.

Round 1: for s steps, $\mathcal{A}$ injects a set X of rs packets that try to cross e_1 and a set Y of rs packets that try to cross fe_2. The packets in S block both sets.

Round 2: for rs steps, $\mathcal{A}$ injects a set X' of r^2s packets that try to cross e_1f and a set Y' of r^2s packets that try to cross e_2. The set X blocks X' and the set Y blocks Y'.

Round 3: for r^2s steps, $\mathcal{A}$ injects a set X'' of r^3s packets that try to cross e_1 and a set Y'' of r^3s packets that try to cross e_2f. The set X'' blocks r^3s packets of X' and Y' blocks Y''.

Round 4: for $2r^3s$ steps, $\mathcal{A}$ injects a set S' of $2r^4s$ packets that try to cross fe_1. They are blocked by the remaining packets in X' and Y''.

At the end of the fourth round, there are $2r^4s$ packets of the form fe_1. The adversary described above achieves that FFS is not stable in U_1 when $2r^4 > 1$. □

Lemma 2. *The pair* (U_2, FFS) *is not stable.*

Proof. Initially we have a set S of $s/2$ packets that try to cross the path $e_1 f_2$ and $s/2$ packets that try to cross the path $e_2 f_1 f_2$.

Round 1: for s steps $\mathcal{A}$ injects a set X of rs packets that try to cross $f_2 e_1 e_2$. The packets in S block the new set at edge f_2.

Round 2: for rs steps $\mathcal{A}$ injects a set X' of r^2s packets that try to cross e_2 and a set Y of r^2s packets that try to cross $e_1 f_2$. The set X blocks the new packets at their initial queues.

Round 3: for r^2s steps $\mathcal{A}$ injects a set X'' of r^3s packets that try to cross $e_2 f_1 f_2$ and a set Y' of r^3s packets that try to cross e_1. The packets in X'' are blocked by the packets in X' and r^3s of the packets in Y are blocked by the packets in Y'.

At the end of the third round, there are $2r^3s$ packets of the form $e_1 f_2$ and of the form $e_2 f_1 f_2$. The adversary described above achieves that FFS is not stable in U_2 when $2r^3 > 1$. □

In the following, we show the instability of any graph in the family of graphs that are extensions of the two basic graphs.

Lemma 3. *Any graph in* $\mathcal{E}(U_1)$ *is not stable under* FFS.

Proof. Let G_1 be a graph in $\mathcal{E}(U_1)$. If G_1 is obtained after some 2-cycle subdivision, then G_1 contains a subgraph in $\mathcal{E}(U_2)$ and therefore it is not stable under FFS. Let us assume that G_1 is obtained by successive arc subdivision as described in Fig. 1(b). This graph is formed by extending the edges of U_1 to paths with lengths k, l and m. Let us denote by p_k, p_l and p_m those paths, where $k, m \geqslant 1$ and $l \geqslant 0$. We also assume that $k > l$, otherwise the adversary will play a symmetric pattern.

The strategy followed by the adversary $\mathcal{A}$ is the following. At the beginning there are s packets that want to traverse paths $p_m p_k$. The adversary $\mathcal{A}$ plays injections in four rounds.

Round 1: for s steps, $\mathcal{A}$ injects a set X of rs packets that try to cross p_k and a set Y of rs packets that try to cross $p_m p_l e_1$. The packets in S block rs packets from set Y and $rs - rm$ packets of set X.

Round 2: for rs steps, $\mathcal{A}$ injects a set X' of r^2s packets that try to cross $p_k p_m$ and a set Y' of r^2s packets that try to cross $p_l e_1$. The set X blocks $r^2s - rm$ packets of the set X', and the set Y blocks $r^2s - rm$ packets of the set Y'.

Round 3: for $r^2s - rm$ steps, $\mathcal{A}$ injects a set X'' of $r^3s - r^2m$ packets that try to cross p_k and a set Y'' of $r^3s - r^2m$ packets that try to cross $p_l e_1 p_m$. The set X'' blocks $r^3s - r^2m$ packets of X' and the set Y' blocks $r^3s - r^2m$ packets of Y''.

Round 4: for $2(r^3s - r^2m)$ steps, $\mathcal{A}$ injects a set X''' of $2(r^4s - r^3m)$ packets to cross $p_m p_k$. The set X''' is blocked, except for the initial rl injections, provided that $r^3s - r^2m + l + 1 > k$. The last condition guarantees a continuous flow of old packets through p_m after l steps.

When the fourth round finishes, there are at least $2(r^4s - r^3m) - rl$, packets waiting to traverse path $p_m p_k$. Notice that as m, l and k are fixed, for big enough s, an injection rate r can be found such that $2(r^4s - r^3m) - rl > s$ and $r^3s - r^2m + l + 1 > k$, and thus G_1 is not stable under FFS. □

Lemma 4. *Any graph in* $\mathcal{E}(U_2)$ *is not stable under* FFS.

Proof. Let G_2 be a graph in $\mathcal{E}(U_2)$ obtained by successive arc subdivisions as described in Fig. 1(b).[4] This graph is formed by extending the edges of U_2 to two paths of lengths l and n respectively. Let us denote by p_l and p_n those paths, and let us assume that $l, n \geqslant 0$ and that $n \leqslant l$. Otherwise, consider an analogous strategy for the adversary played over the symmetric network in which e_2 plays the role of f_2.

In the initial step, we have a set of s packets distributed in the following way: α packets that try to

[4] Observe that, all the graphs obtained by applying, at least once, the 2-cycle subdivision operation, must contain a subgraph that is obtained from U_2 applying only arc subdivisions.

cross $p_l e_1 f_2$ and β packets that try to cross $e_2 p_n f_1 f_2$, where $\alpha + \beta = s$ and $\beta > l + 2$.

Round 1: for s steps, $\mathcal{A}$ injects a set X of rs packets that try to cross $f_2 p_l e_1 e_2$. The condition $\beta > l + 2$ guarantees a continuous flow of packets from S through f_2. The packets in S block the new set, but at most $\min\{r(l+1), r(n+2)\}$ injections cannot be accumulated, so at least $r(s-n)$ packets from X are blocked.

Round 2: for $r(s-n)$ steps, $\mathcal{A}$ injects a set X' of $r^2(s-n)$ packets that try to cross $e_2 p_n$ and a set Y of $r^2(s-n)$ packets that try to cross $p_l e_1 f_2$. The set X blocks all the new packets, except the first rl packets from X'.

Round 3: for $r^2(s-n)$ steps, $\mathcal{A}$ injects a set X'' of $r^3(s-n)$ packets that try to cross $p_l e_1$ and a set Y' of $r^3(s-n)$ packets that try to cross $e_2 p_n f_1 f_2$. The set X'' blocks $r^3(s-n)$ packets of Y and the remaining packets from X' block at least $r^3(s-n) - rl$ injections from Y'.

At the end of the 3rd round, there are at least $r^3(s-n)$ packets that want to traverse $p_l e_1 f_2$ and at least $r^3(s-n) - rl$ packets that want to traverse $e_2 p_n f_1 f_2$. Thus there are a total of at least $2r^3(s-n) - rl$ packets in the system. For s large enough, an injection rate r can be found such that the total number of packets is increased, i.e., $2r^3(s-n) - lr > s$ and the $\beta > l + 2$ initial condition is also satisfied, $r^3(s-n) - rl > l + 2$. Thus, G_2 is not stable under FFS. □

It is known that, if a graph G has a subgraph which is not stable, then G is not stable. Taking this into account, and using all the instability results shown in Lemmas 1 to 3, we can state the following theorem.

Theorem 2. *A digraph G is stable under* FFS *if and only if $G \notin \mathcal{S}(\mathcal{E}(U_1) \cup \mathcal{E}(U_2))$.*

Together the previous theorem and the results in [2] (Theorem 1 and more) provide the following results.

Theorem 3. *A digraph G is stable under* FFS *if and only if G is universally stable.*

Corollary 1. *Stability under* FFS *of a given digraph can be decided in polynomial time.*

References

[1] C. Àlvarez, M. Blesa, M. Serna, Universal stability of undirected graphs in the adversarial queueing model, in: 14th ACM Symp. on Parallel Algorithms and Architectures (SPAA'02), Winnipeg, Manitoba, ACM Press, New York, 2002, pp. 183–197.

[2] C. Àlvarez, M. Blesa, M. Serna, A characterization of universal stability in the adversarial queueing model, SIAM J. Comput., in press.

[3] M. Andrews, B. Awerbuch, A. Fernández, J. Kleinberg, T. Leighton, Z. Liu, Universal stability results for greedy contention-resolution protocols, J. ACM 48 (1) (2001) 39–69.

[4] A. Borodin, J. Kleinberg, P. Raghavan, M. Sudan, D. Williamson, Adversarial queueing theory, J. ACM 48 (1) (2001) 13–38.

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