The complexity of deciding stability under FFS in the adversarial model*

C. Àlvarez[1] M. Blesa[1] J. Díaz[1] A. Fernández[2] M. Serna[1]

[1] Dept. Llenguatges i Sistemes Informàtics
Universitat Politècnica Catalunya
Jordi Girona 1–3 Campus Nord, 08034 Barcelona, Spain
{alvarez,mjblesa,diaz,mjserna}@lsi.upc.es

[2] GSyC
Universidad Rey Juan Carlos
Tulipan s/n, Campus de Mostoles, 28933 Madrid, Spain
anto@gsyc.escet.urjc.es

Abstract

In this paper we show the polynomial time decidability of the stability property under the Farthest From Source (FFS) scheduling policy in the case that the adversary can solve ties arbitrarily. This result follows from a characterization of the stability property in terms of a family of forbidden subgraphs.

1 Introduction

In recent times, an important model to study stability and load balancing issues in non-adaptive routing has been the Adversarial Queuing Theory (AQT) model proposed by Borodin et al. [4]. This model considers the time evolution of a packet-routing network as a game between an adversary and a queueing policy or protocol. The adversary controls the traffic pattern by injecting at each time step a set of packets into the system. For each packet the adversary specifies a sequence of edges that it must traverse, before the packet is absorbed. If more than one packet wishes to cross an edge e, at the same time step, then

*Work partially supported by the FET Programme of the EU under contract number IST-2001-33116 (FLAGS), and by the Spanish CICYT projects TIC-2001-4917-E and TIC2001-1586-C03-01. The second author was also supported by the Catalan Research Council with the pre-doctoral grant 2001FI-00659.

the queueing policy chooses one of these packets to send across e. The remaining packets wait in the queue. The game advances to the next time step. The main purpose of the model is the study of the stability of the system, i.e. whether at any time the maximum number of packets present in the system is bounded by a constant that may depend on network parameters. We are interested in analyzing the network behavior under *greedy* queueing policies. A protocol is *greedy* if whenever there is at least one packet waiting to use an edge, the protocol advances a packet through the edge.

The constraints of an adversary in the AQT model are specified by two parameters, the *injection rate* r, $0 < r < 1$, and an integer $b \geq 0$ which denote the *burstiness*. The power of the adversary is controlled by r and b in the following way: if $N_e(I)$ denotes the number of packets which have paths containing a particular edge e injected by the adversary during a time interval I, then $N_e(I) \leq r|I| + b$.

The main greedy protocols considered in the literature (see [4] and [3]) are: LIFO (Last In First Out), which gives priority to the packet that joined later the queue of the edge; FIFO (First In First Out), in which highest priority is given to the packet that has arrived first in the queue of the edge; SIS (Shortest In System), which gives priority to the packet introduced into the system later; LIS (Longest In System) in which every queue gives priority to the packet that has been in the system the longest; NTG (Nearest To Go), in which highest priority is assigned to the packet that is closest to its destination; FTG (Furthest To Go) in which highest priority is assigned to the packet that is farthest away from its destination; NFS (Nearest From Source)[1] and FFS (Farthest From Source) consider the same policies but taking the distance to the source node of the packets as reference point.

A network is *universally stable* if for any queueing policy and any adversary the resulting system is stable. Deciding universal stability of networks represented by digraphs can be solved in polynomial time [2]. The same complexity results are known for undirected graphs and other types of packet trajectory restrictions (see [3] and [1]).

Fixing the queueing policy a natural question to ask is whether it is *universally stable*. This means that every digraph is stable against any adversary under the selected queuing policy. 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 that are not universally stable other interesting problem is to analyze the complexity of deciding the stability of a given network under the fixed protocol. Only one result for the case of digraphs is known: Deciding stability under NTG-LIS is polynomially solvable [2]. Recall that NTG-LIS works as NTG, but solving ties using the LIS protocol. Further results for undirected graphs and other types of packet trajectory restrictions can be found in [3, 1].

In this paper we show the polynomial time decidability of the stability property under FFS in the case that the adversary can solve ties arbitrarily. Our main result shows that a digraph is stable under FFS if and only if it is universally stable.

[1] Also in the literature as NTS (Nearest To Source).

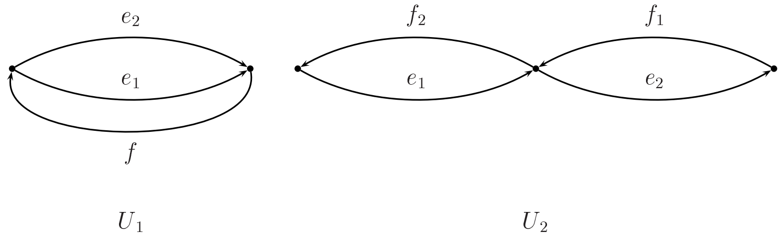

Figure 1: The two basic digraphs characterizing universal stability.

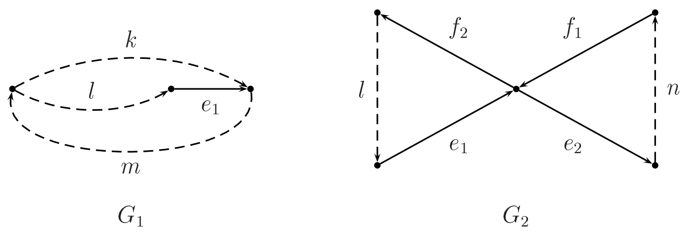

Figure 2: Extensions by arc subdivision of the two basic digraphs characterizing universal stability (see Figure 1).

2 Definitions and previous results

Through the paper, all digraphs G are loop-less, but may have multiple edges. In the same way as in [3], given a digraph G, and a queuing policy $\mathcal{P}$, we say that for a given adversary $\mathcal{A}$, the system $S = (G, \mathcal{A}, \mathcal{P})$ is *stable* if, at any time step, the maximum number of packets in the system is bounded.

In the FFS protocol, priority is given to the packet in the queue which is farthest from its source. In the following, we assume that the adversary may break ties arbitrarily. Before formally stating our results, we need to introduce some graph theoretical definitions. 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.

Given a family of digraphs $\mathcal{G}$, $\mathcal{S}(\mathcal{G})$ denotes the family of digraphs that contain a graph in $\mathcal{G}$ as a subgraph.

Figure 1 provides the two basic graphs needed to characterize universal stability, and Figure 2 gives the shape of the arc subdivision extensions of those graphs. This basic family provides the characterization of the universal stability of networks. The following result was proved in [2].

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 paper it is also shown that the above property can be checked in polynomial time.

3 Stability under FFS

In this section we analyze the complexity of deciding whether a given network G is stable under FFS. We show first the instability of the two basic graphs given in Figure 1. Our instability proofs show one cycle of injections, at the end of the cycle the system has the same type of load, but will have increased the number of packets.

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 the set 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''.
Then, for $2r^4 > 1$, FFS is not stable in U_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_1f_2 and $s/2$ packets that try to cross the path $e_2f_1f_2$ (see Figure 1).

Round 1: for s steps $\mathcal{A}$ injects a set X of rs packets that try to cross $f_2e_1e_2$. 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_1f_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 $e_2f_1f_2$ 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'.

Then, for $2r^3 > 1$, FFS is not stable in U_2. ■

Next, we show the non-stability of any graph in the family of graphs that are an extension of the two basic graphs.

Lemma 3 *Any graph in $\mathcal{E}(U_2)$ is not stable under* FFS.

Proof: Let now G_2 be a graph in $\mathcal{E}(U_2)$ obtained by successive arc subdivisions as described in Figure 2. 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 $l, n \geq 0$ and that $n \leq l$. Otherwise, consider an analogous strategy for the adversary played over the symmetric network in which e_2 plays the role of f_2.

For sake of simplicity we will omit some additive constants in our analysis, these omissions will not change the final result.

In the initial step, we have a set of s packets distributed in the following way: α packets that try to 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 can not 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 third 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.

Finally 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. ■

Now we prove that all the extensions of the digraph U_1 are not stable.

Lemma 4 *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 Figure 2. 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 $k, m \geq 1$ and $l \geq 0$. 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 $\mathcal{A}$ will play 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 $r^2 s$ packets that try to cross $p_k p_m$ and a set Y' of $r^2 s$ packets that try to cross $p_l e_1$. The set X blocks $r^2 s - rm$ packets of the set X', and the set Y blocks $r^2 s - rm$ packets of the set Y'.

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

Round 4: for $2(r^3 s - r^2 m)$ steps $\mathcal{A}$ injects a set X''' of $2(r^4 s - r^3 m)$ packets that try to cross $p_m p_k$. The set X''' is blocked, except for the initial rl injections, provided that $r^3 s - r^2 m + l + 1 > k$. 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 $2(r^4 s - r^3 m) - rl$, 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 $2(r^4 s - r^3 m) - rl > s$ and $r^3 s - r^2 m + l + 1 > k$, and thus G_1 is not stable under FFS. ■

Therefore, from Theorem 1 and taking into account that if a graph G has an non-stable subgraph then G is not stable, we have

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

Putting together the previous theorem and Theorem 1 we have

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

References

[1] C. Àlvarez, M. Blesa, and M. Serna. Universal stability of undirected graphs in the adversarial queueing model. In *14th ACM Symposium on Parallel Algorithms and Architectures*, pages 183–191, Winnipeg, Canada, August 2002. ACM Press New York, NY, USA.

[2] C. Àlvarez, M. Blesa, and M. Serna. Universal stability of directed and undirected graphs in the adversarial queueing model. Technical Report, Software department, Universitat Politècnica de Catalunya, 2003.

[3] 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. Previously at FOCS'96, pages 380–389.

[4] A. Borodin, J. Kleinberg, P. Raghavan, M. Sudan, and D. Williamson. Adversarial queueing theory. *Journal of the ACM*, 48(1):13–38, 2001. Previously at STOC'96, pages 376–385.

[5] Z. Lotker, B. Patt-Shamir and A. Rosén New Stability Results for Adversarial Queuing. In *14th ACM Symposium on Parallel Algorithms and Architectures*, pages 192–199, Winnipeg, Canada, August 2002. ACM Press New York, NY, USA.