Universal Stability Results for Low Rate Adversaries in Packet Switched Networks

Juan Echagüe, Vicent Cholvi, and Antonio Fernández

***Abstract*—In this work, we consider a generalized version of the adversarial model and study the universal stability (stability in any network) of scheduling policies under low load rates. We show that any work-conserving policy is universally stable at any load rate $r < (1/d)$, where d is the largest number of links crossed by any packet. We also show that system-wide time priority policies are universally stable at any load rate $r < (1/(d-1))$.**

***Index Terms*—Adversarial model, packet scheduling policies, packet-switched networks, stability.**

I. Introduction

IN THE LAST few years, much of the analysis of worst case behavior of connectionless networks and scheduling policies has been done by using an "adversarial" approach [1]–[6]. Under this model, time is seen as discrete, and the time evolution of a packet-switching network is seen as a game between a bounded adversary and a queue policy. At each time step the adversary injects a set of packets in some of the nodes of the network. The adversary is free to choose both the source and the destination node of any injected packet. Furthermore, it also specifies the sequence of links (the route) that each individual packet must traverse. Its only restriction (hence the term "bounded") is that it can not fully load any link. Packets are transmitted between adjacent nodes so that the scheduling policy decides at every step which packets have to cross each link. A packet will be absorbed after traversing its route.

Under this adversarial model, one crucial issue when characterizing the performance of networks and policies is that of *stability*. Being consistent with the standard use of the term (see for instance [1]–[6]), we say that a network is *stable* under a given scheduling policy if for any bounded adversary the backlog at any node (i.e., the number of bits "in transit") is bounded (by a value that does not depend on the time). It is known [2] that when no link is fully loaded, network stability (bounded backlogs) also means deterministically bounded delays, sustainable with finite buffers without packet loss, the main objective of deterministic analysis. We also say that a scheduling policy is *universally stable* if all networks are stable under it.

We emphasize that this is a connectionless model; however, all our stability results are also applicable to connection oriented networks [2] (the converse direction does not hold). We have extended the classical adversarial models from Borodin *et al.* [1] and Andrews *et al.* [2] to introduce different link capacities and packet lengths. Consequently, our model subsumes the classical connection oriented models [7], [8].

We model our network as a set of nodes interconnected by directed point-to-point links. Each node contains a server for each outgoing link and we allow different service rates (link bandwidths). Hence $C_e \geq 1$ will denote the service rate of server e, measured in bits per unit of time. We also associate a propagation delay P_e to each link e. Each server schedules the packets that must cross the link using a nonpreemptive scheduling policy (maybe different at each server). To make our results stronger, we do not allow cut-through at the nodes, i.e., a packet must be received completely in one node before it can be sent out through any outgoing link of the node.

We use a generalized version of the model of adversary of Andrews *et al.* [2], which is commonly used in the literature. This adversary is defined (as theirs) by a pair of parameters (b, r), where $b \geq 1$ is a natural number and r is $0 < r < 1$. In our model we do not restrict the number of packets injected by the adversary, but the number of bits (bits are still grouped into packets that appear at their ingress node instantaneously). The parameter b (usually called *burstiness*) models the short bursts of bits we can inject into the network. The parameter r (called the *load rate*) is the sustainable proportion of the link bandwidth C_e at which bits that require to cross link e can be injected, for all e. If we denote by $A_e(x)$ the total number of bits that the adversary injects during any time interval of length x that traverses the link associated with server e, the adversary must satisfy that $A_e(x) \leq b + rxC_e$ (for all $x \geq 0$).

A. Previous Work

The adversarial approach was initially proposed by Borodin *et al.* [1]. In [2], Andrews *et al.* provide a list of universally stable packet scheduling policies under a slightly more general adversarial model. They showed that policies like *Farthest-to-Go* (FTG), *Nearest-to-Source* (NTS), *Shortest-in-System* (SIS) and *Longest-in-System* (LIS) are universally stable. In contrast, they showed that packet scheduling policies like *First-in-First-Out* (FIFO), *Last-in-First-Out* (LIFO), *Nearest-to-Go* (NTG) and *Farthest-from-Source* (FFS) are not universally stable. Furthermore, LIFO, NTG and FFS can be made unstable at arbitrarily

Manuscript received May 16, 2003. The associate editor coordinating the review of this letter and approving it for publication was Dr. D. Petr. This work was supported in part by the Spanish Ministerio de Ciencia y Tecnología (MCyT) under Grant TIC2001-1586-C03-01, the Comunidad de Madrid under Grant 07T/0022/2003, and by the Universidad Rey Juan Carlos under Grant PPR-2003-37.

J. Echagüe is with the Departamento de Ingenier a y Ciencia de los Computadores, Univeritat Jaume I, 12071 Castelló de la Plana, Spain.

V. Cholvi is with the Departamento de Lenguajes y Sistemas Informáticos, Universitat Jaume I, 12071 Castelló de la Plana, Spain (e-mail: vcholvi@uji.es).

A. Fernández is with the Grupo de Sistemas y Comunicaciones, Universidad Rey Juan Carlos, E-28933 Móstoles, Madrid, Spain.

Digital Object Identifier 10.1109/LCOMM.2003.821324

low load rates [1]. However, the situation has been far more complex for FIFO. Andrews *et al.* [2] proved that FIFO is unstable when the load rate is at least 0.85. This bound was reduced to 0.8357 by Díaz *et al.* [3], to 0.749 by Koukopoulos *et al.* [4] and to 0.5 by Lokter *et al.* [6]. Finally, independently, Bhattacharjee and Goel [5] and Koukopoulos *et al.* [9] seem to have different but similar results that show that FIFO can be unstable at arbitrarily low constant load rates (see [9] for a comparison of both results).

Taking a different approach, Lokter *et al.* [6] also proved that any *work conserving* (also known as greedy: those that always schedule packets if there is anyone in the queue) scheduling policy is stable if the load rate is no bigger than $1/(d+1)$ (where d is the largest number of links crossed by any packet). They also reduced that bound to $1/d$ for *time priority* scheduling policies (i.e., policies under which, a packet arriving at a buffer at time t has priority over any other packet that is injected after time t). In this work we use the term *system-wide time priority* to denote these policies. FIFO and LIS are examples of system-wide time priority scheduling policies.

On the other hand and by considering session oriented networks (i.e., networks where all packets belong to some session and packets from the same session follow the same route), Charny and Le Boudec [10] proved that FIFO is stable if $r < (1/(d-1))$. This result was also obtained by Zhang and Duan [11] using a different proof technique. Charny and Le Boudec also showed that any improvement of this bound (beyond the equality case) has to consider some network parameter or the number of sessions, since for any d large enough, any $r > (1/(d-1))$, and any constant D they found a (large) network and adversarial injections such that no packet crosses more than d links but the maximum delay is larger than D. The stability result in [10] was extended in [12] to (possibly non FIFO) schedulers of the Guaranteed Rate type. Finally, Tassiulas and Georgiadis [13] have shown that an unidirectional ring is stable for $r < 1$.

B. Our Results

In this paper we generalize the technique used by Zhang and Duan [11] to apply it to both work conserving and system-wide time priority scheduling policies. We show that work conserving scheduling policies are stable if the load rate is lower than $1/d$. Furthermore, we also show that any system-wide time priority scheduling policy is stable if the load rate is lower than $1/(d-1)$. Given the results of Charny and Le Boudec [10] for FIFO (which is system-wide time priority) mentioned above, to improve this latter bound (if possible) we need to consider some additional parameters.

Observe that these results improve those of Lokter *et al.* [6] and that they are valid in heterogeneous networks, i.e. networks where hosts run simultaneously different scheduling policies, as long as those scheduling policies fulfill the above mentioned requirements. Note also that, whereas Lokter *et al.* use the adversarial model of Borodin *et al.* [1], here we use that of Andrews *et al.* [2], which, as it has been shown by Rosén [14], is a little more general.

The rest of the letter is organized as follows. In Sections II and III we provide, respectively, the stability conditions for work conserving and system-wide time priority packet scheduling policies. The conclusions are provided in Section IV.

II. Stability Conditions for Work Conserving Policy

In this section we obtain an expression of the longest time a packet can wait at the queue of any server (link) with any work conserving policy. This expression is then used to derive an upper bound on the load rate r to guarantee stability.

In what follows, we denote as d_p the number of links that a packet p has to cross, and define $d = max_p\{d_p\}$. We denote C_i^p and P_i^p as the service rate and propagation delay, respectively, of the ith server for packet p. We also denote by a_i^p and f_i^p the time instants that packet p arrives and departs,[1] respectively, at its ith server, where $1 \leq i \leq d_p$. Hence, packet p crosses its ith link in time step f_i^p and arrives at its $i+1$st server queue at time step $a_{i+1}^p = f_i^p + P_i^p$. Finally, we denote by Q_i^p the time interval packet p takes to cross its ith link, i.e. $Q_i^p = f_i^p - a_i^p + P_i^p$. Let $Q = \max_{p, 1 \leq i \leq d_p} Q_i^p$.

Observation 1: Any packet arrives at the queue of its ith server in at most $(i-1)Q$ time after being injected, i.e. $a_i^p - a_1^p \leq (i-1)Q$.

The next theorem provides a bound on the load rate that guarantees network stability under any work conserving scheduling policy.

Theorem 2.1.: Any network in which all servers use a work conserving packet scheduling policy and packets are injected by a (b, r) adversary is stable if $r < (1/d)$. Furthermore, the worst-case end-to-end delay D is bounded above by $D \leq d(b/(1-rd))$.

Proof: The proof has two parts. First, we prove that if $r < (1/d)$ then $Q < \infty$ (which implies stability). Second, we prove that, if the first part is true, then D is also bounded above by $d(b/(1-rd))$.

Part (1): We base our proof in finding the conditions that make $Q < \infty$. Let p be a packet that attains at its ith server the maximum Q. Let t_B be the last time no later than a_i^p that no packet was scheduled by the server. (That happened because the queue was empty, since the policy is work conserving.) Hence, we have that the interval $(t_B, f_i^p]$ is a busy period for the ith server (i.e., during that interval the ith server buffer is nonempty).

Define ϕ_i^p as the set formed by all packets served by the ith server during the interval $(t_B, f_i^p]$ and let p^* be the oldest packet in ϕ_i^p (i.e., $\forall p' \in \phi_i^p (a_1^{p'} \geq a_1^{p^*})$). Hence, by definition of p^*, all packets in ϕ_i^p must have been injected during the interval $[a_1^{p^*}, f_i^p]$. (Remember that packets are injected instantaneously at their ingress nodes.)

Based on the above mentioned scenario and on the definition of the adversarial model, $Q_i^p C_i^p$ will be bounded by the maximum number of bits injected during the interval $[a_1^{p^*}, f_i^p]$ minus the bits served (by the ith server) during the interval (t_B, a_i^p).

[1]A packet is considered to have arrived at a scheduler only when its last bit has been received, and it to have departed when its last bit has been serviced.

Let us assume that the ith link for p is the jth link for p^*. Then

$$
\begin{aligned}
Q_i^p C_i^p &= Q C_i^p \\
&\leq r C_i^p \left(f_i^p - a_1^{p^*} \right) + b - C_i^p \left(a_i^p - t_B \right) \\
&\leq r C_i^p \left(f_i^p - a_1^{p^*} \right) + b - r C_i^p \left(a_i^p - t_B \right) \\
&= r C_i^p \left(f_i^p - a_i^p + t_B - a_1^{p^*} \right) + b \\
&\leq r C_i^p Q + r C_i^p \left(t_B - a_1^{p^*} \right) + b \\
&= r C_i^p Q + r C_i^p \left(t_B - a_j^{p^*} + a_j^{p^*} - a_1^{p^*} \right) + b \\
&\leq r C_i^p Q + r C_i^p (d-1) Q + r C_i^p \left(t_B - a_j^{p^*} \right) + b \\
&\leq r d C_i^p Q + b.
\end{aligned}
$$

The fifth step follows from the fact that $f_i^p - a_i^p \leq Q$, the sixth step follows from Observation 1 and the fact that $j \leq d$, and the seventh step follows since $a_j^{p^*} \geq t_B$.

Then, if $r < (1/d)$ we obtain that $Q \leq (b/(C_i^p(1 - rd)))$, with $C_i^p(1 - rd) > 0$. This implies that $Q < \infty$.

Part (2): The proof is very similar to that in Theorem 1 in [10] and uses the time-stopping method. Consider a packet p that traverses a path with d_p hops, where $d_p \leq d$. For any time $t > 0$, consider the virtual system made of the original network, where all sources are stopped at time t. This network satisfies the assumption of Part (1), since there is only a finite number of bits at the input. Call $D'(t)$ the worst case end-to-end delay of packet p for the virtual network indexed by t. From the above derivation we see that $D'(t) \leq d_p Q \leq d(b/1 - rd)$ for all t. Letting t tend to $+\infty$ shows that the worst case delay remains bounded above by $d(b/(1 - rd))$. ∎

III. Stability Conditions for System-Wide Time Priority Packet Schedulers

The next theorem provides a bound on the load rate that guarantees network stability under any system-wide time priority scheduling policy.

Theorem 3.1.: Any network in which all servers use a system-wide time priority packet scheduling policy and packets are injected by a (b, r) adversary is stable if $r < (1/(d-1))$. Furthermore, the worst-case end-to-end delay D is bounded above by $D \leq d(b/(1 - r(d-1)))$.

Proof: The proof here is similar to the proof of Theorem 2.1.

Part (1): The main difference is that now, $Q_i^p C_i^p$ will be bounded by the maximum number of packets injected during $[a_1^{p^*}, a_i^p]$ (instead of $[a_1^{p^*}, f_i^p]$, since the policy is system-wide time priority) minus the packets served during the busy period interval (t_B, a_i^p).

Making some algebra we found that if $r < (1/d - 1)$ we obtain that $Q \leq (b/C_i^p(1 - r(d-1)))$, with $C_i^p(1 - r(d-1)) > 0$. This implies that $Q < \infty$.

Part (2): By using the same reasoning as in Part (2) in Theorem Theorem 2.1. ∎

IV. Conclusions

In this work, we have given a bound of $r < (1/d)$ on the load rate below which all work-conserving scheduling policies are stable in any network. We obtain a slightly better bound of $r < (1/d - 1)$ for system-wide time priority policies. Both bounds depend on the length of the longest packet route d. Two main questions remain open. The first one is whether our bound in the case of work conserving policies is tight (note that, by result of Charny and Le Boudec in [10], it seems clear that our bound is tight in the case of system-wide time priority policies). The second one is to improve our bounds by considering some network parameters other than d.

Acknowledgment

The authors thank the anonymous reviewers for their constructing suggestions and comments.

References

[1] A. Borodin, J. Kleinberg, P. Raghavan, M. Sudan, and D. Williamson, "Adversarial queueing theory," *J. Assoc. Comput. Mach.* , vol. 48, no. 1, pp. 13–38, 2001.

[2] M. Andrews, B. Awerbuch, A. Fernandez, J. Kleinberg, T. Leighton, and Z. Liu, "Universal-stability results and performance bounds for greedy contention-resolution protocols," *J. Assoc. Comput. Mach.*, pp. 39–69, Jan. 2001.

[3] J. Díaz, D. Koukopoulos, S. Nikoletseas, M. Serna, P. Spirakis, and D. Thilikos, "Stability and nonstability of the fifo protocol," in *Proc. 13th Annu. ACM Symp. on Parallel Algorithms and Architectures*, 2001, pp. 48–52.

[4] D. Koukopoulos, S. Nikoletseas, and P. Spirakis, "The range of stability for heterogeneous and fifo queueing networks," in *Electronic Colloq. on Computational Complexity*, 2001.

[5] R. Bhattacharjee and A. Goel, "Instability of Fifo Arbitrarily Low Rates in the Adversarial Queueing Model," Dep. Computer Sci., Univ. Southern California, Tech. Rep. 02-776, Nov. 2002.

[6] Z. Lotker, B. Patt-Shamir, and A. Rosén, "New stability results for adversarial queueing," in *ACM Symp. on Parallel Algorithms and Architectures*, Aug. 2002.

[7] R. L. Cruz, "A calculus for network delay. Parts I and II," *IEEE Trans. Inform. Theory*, vol. 37, pp. 114–131, Jan. 1991.

[8] R. L. Cruz, "A calculus for network delay. Parts I and II," *IEEE Trans. Inform. Theory*, vol. 37, pp. 132–141, Jan. 1991.

[9] D. Koukopoulos, M. Mavronicolas, and P. Spirakis, "Fifo is unstable at arbitrarily low rates (even in planar networks)," in *Electronic Colloq. on Computational Complexity*, Apr. 2003.

[10] A. Charny and J.-Y. Le Boudec, "Delay bounds in a network with aggregate scheduling," in *First Int. Workshop on Quality of Future Internet Services*, Berlin, Germany, Sept. 2000.

[11] Z.-L.Zhi-Li Zhang, Z.Zhenhai Duan, and Y. T.Yiwei Thomas Hou, "Fundamental tradeoff in aggregate packet scheduling," in *Proc. 9th Int. Conf. on Network Protocols, ICNP 2001*, Riverside, CA, Nov. 2001.

[12] V. Firoiu, J.-Y. Le Boudec, D. Towsley, and Z.-L. Zhang, "Theories and models for internet quality of service," *Proc. IEEE*, vol. 90, no. 9, pp. 1565–1591, Sept. 2002.

[13] L. Tassiulas and L. Georgiadis, "Any work conserving policy stabilizes the ring with spatial reuse," *IEEE/ACM Trans. Networking*, pp. 205–208, Apr. 1996.

[14] A. Rosén, "A note on models for nonprobabilistic analysis of packet-switching networks," *Inform. Processing Lett.*, vol. 73, no. 5–6, pp. 213–220, 2002.