A Note on Models for Non-Probabilistic Analysis of Packet-Switching Networks

Adi Rosén*

Abstract

We consider two models commonly used in the literature to model adversarial injection of packets into a packet switching network. We establish the relation between these two types of models, and between them and the set of sequences of packets that allow stability. We also consider the adaptive setting in which packets are injected with only their source and destination but without a prescribed path to follow.

1 Introduction

The study of the behavior of packet switching networks against adversarial injection of packets has received considerable attention in recent years. See, for example, [5, 2, 1, 7, 6, 3]. In such networks, as modeled in the literature, packets are injected into the network in a continuous manner, and are transmitted between adjacent switches (nodes) in discrete time steps. Each link, connecting two adjacent nodes, can transmit a single packet in each time step. The packets travel to their destinations in a "store-and-forward" manner, being stored in buffers at intermediate switches. Since the bandwidth of the links is limited, two natural questions arise in this setting: what are the sizes of the buffers in the network and what are the delays incurred by the packets. In particular, the question of stability is of importance: is there a (finite) bound on the size of the buffers which is independent of the length of time the network is active ? Or does the number of packets that are in the network grow to infinity as time advances ? Naturally, the answers to these questions depend on the network topology, on the sequence of packets injected, and on the protocol used to route and schedule the packets.

Most of the recent work analyzing scenarios along the above lines with no probabilistic assumptions makes use of one of two frameworks in order to define the set of sequences of packets to be considered. Both frameworks model the sequences of packets as being given by an adversary, but use different parameters, and a different definition, in order to define the set of sequences that a particular adversary can inject into the network. One was introduced by Borodin et al. [5], and is referred here as the "Adversarial Queuing Theory" model, and

*Dept. of Computer Science, Technion, Haifa 32000, Israel. e-mail: adiro@cs.technion.ac.il.

the other was introduced in this context by Andrews et al. [2], and is referred here as the "Leaky Bucket" model. Some confusion exists in the literature concerning the relative power of these adversaries, and concerning necessary and sufficient conditions to have stability. This short note is intended to formally establish the relationship between the two models, and necessary and sufficient conditions for stability.

2 Definitions

In what follows we assume a given network, modeled as a directed graph $G = (V, E)$, $|V| = n$, $|E| = m$. All of the definitions and claims in the sequel are given for a specific network.

A packet is identified by (s, d, π), where $s \in V$ is the source node, $d \in V$ is the destination node, and π is a simple path leading from s to d. Any packet $p = (s, d, \pi)$ is injected at some time step t into its source node s, and has to follow π in order to reach d.

We first define the two models of adversaries common in the literature. The first one is used, e.g., in [5, 7, 6, 1], and the second one in, e.g., [2, 3].

Definition 1: An $AQT(w, r)$ adversary, for $w \geq 1$, $r \leq 1$, is an adversary that is allowed to inject any sequence of packet into the network, as long as the following condition holds. Consider any w consecutive time steps, and all the packets injected in these times steps. Then, the total load created by the paths associated with these packets, on any edge $e \in E$, is at most wr.

Definition 2: An $LB(b, r)$ adversary, for $b \geq 0$, $r \leq 1$, is an adversary that is allowed to inject any sequence of packet into the network, as long as the following condition holds. For any $T \geq 1$, consider any T consecutive time steps, and all the packets injected in these times steps. Then, the total load created by the paths associated with these packets, on any edge $e \in E$, is at most $Tr + b$.

We now define sets of sequences. By σ we denote a sequence of packets.

Definition 3:

- $\mathcal{AQT}(r < 1) = \{\sigma : \sigma \text{ can be given by } AQT(w, r), \text{ for some } w \geq 1, r < 1\}$.
- $\mathcal{AQT}(r \leq 1) = \{\sigma : \sigma \text{ can be given by } AQT(w, r), \text{ for some } w \geq 1, r \leq 1\}$.
- $\mathcal{LB}(r < 1) = \{\sigma : \sigma \text{ can be given by } LB(b, r), \text{ for some } b \geq 0, r < 1\}$.
- $\mathcal{LB}(r \leq 1) = \{\sigma : \sigma \text{ can be given by } LB(b, r), \text{ for some } b \geq 0, r \leq 1\}$.

We now define the set of sequences of packets for which stability is possible.

Definition 4: The set $\mathcal{ST}$ is the set of all (finite and infinite) sequences of packets for which there exist a feasible schedule for the packets and a constant $B < \infty$, such that at any time step $t \geq 1$, the number of packets present in the network at time t, is at most B.

3 Relation between the models

In this section we establish the following:

$$\mathcal{LB}(r<1)=\mathcal{AQT}(r<1)\subsetneqq\mathcal{AQT}(r\leq 1)\subsetneqq\mathcal{LB}(r\leq 1)=\mathcal{ST}\ .$$

Fact 1: $\mathcal{LB}(r<1)=\mathcal{AQT}(r<1)$.

Proof:
$\underline{\mathcal{AQT}(r<1)\subseteq\mathcal{LB}(r<1)}$: Let σ be a sequence of packets injected by an $AQT(w,r)$ adversary, for some $w\geq 1$, and some $r<1$. Let $b'=wr$, and $r'=r$. Then a $LB(b',r')$ adversary can also inject σ. To see that let $T\geq 1$, and consider time interval I of T consecutive time steps. Let k and $t'\leq w$ be integers such that $T=kw+t'$. Then, since σ is given by $AQT(w,r)$, for any $e\in E$ the number of packets that require e and are injected in I is at most $(k+1)wr$. Therefore σ can also be injected by $LB(b',r')$, as $Tr'+b'=(kw+t')r'+wr\geq(k+1)wr$.
$\underline{\mathcal{LB}(r<1)\subseteq\mathcal{AQT}(r<1)}$: Let σ be a sequence of packets injected by an $LB(b,r)$ adversary, for some $b\geq 0$, and some $r<1$. For any $0<\delta<1$ let $w'=\frac{b}{(1-r)\delta}$, and $r'=r+(1-r)\delta$. Then an $AQT(w',r')$ adversary can also inject σ. To see that consider any interval I of consecutive w' time steps. Since σ is given by $LB(b,r)$, for any $e\in E$, the number of packets that require e and are injected in I is at most $w'r+b=\frac{b}{(1-r)\delta}\cdot r+b$. Therefore σ can also be injected by $AQT(w',r')$, as $w'r'=\frac{b}{(1-r)\delta}\cdot(r+(1-r)\delta)=\frac{b}{(1-r)\delta}\cdot r+b$. □

Fact 2: $\mathcal{LB}(r\leq 1)=\mathcal{ST}$.

Proof:
$\underline{\mathcal{LB}(r\leq 1)\subseteq\mathcal{ST}}$: The protocols Furthest-To-Go and Nearest-To-Origin are stable for any adversary $LB(b,r)$, $b\geq 0$, $r\leq 1$, and any network [2, 7]. The claim then follows.
$\underline{\mathcal{ST}\subseteq\mathcal{LB}(r\leq 1)}$: Let σ be a (infinite) sequence of packets for which there is a feasible schedule and a constant $B<\infty$ such that at any time at most B packets are present in the network. Now observe that for any time interval I of $T\geq 1$ consecutive time steps, and for any $e\in E$, the sequence σ can have at most $T+B$ packets that are injected in I and request e. Otherwise, since e can transmit at most T packets during I, there would be at the end of I more than B packets in the network. Therefore σ can be given by a $LB(B,r=1)$ adversary. □

Fact 3: $\mathcal{AQT}(r<1)\neq\mathcal{AQT}(r\leq 1)$.

Proof:
A sequence of packets that is in $\mathcal{AQT}(r=1)$ but not in $\mathcal{AQT}(r<1)$ is the sequence of packets all requiring a single edge, being injected one packet per time step. □

Fact 4: $\mathcal{AQT}(r \leq 1) \subsetneqq \mathcal{LB}(r \leq 1)$

Proof:

The fact that $\mathcal{AQT}(r \leq 1) \subseteq \mathcal{LB}(r \leq 1)$ follows from the same arguments used to prove $\mathcal{AQT}(r < 1) \subseteq \mathcal{LB}(r < 1)$ in the proof of Fact 3 (These arguments did not use $r < 1$).

To see that $\mathcal{AQT}(r \leq 1) \neq \mathcal{LB}(r \leq 1)$ observe that any of the following two types of sequences are in $\mathcal{LB}(r \leq 1)$ but not in $\mathcal{AQT}(r \leq 1)$. In both cases all the packets have to cross the same single edge. (a) In time step 1 two packet are injected; in any later time step, one packet is injected. (b) The sequence is composed of phases $i \geq 0$. Phase i is composed of a first time step where two packets are injected, then i time steps with one packet per time step, and then one time step with no packet being injected. $\square$

4 The adaptive case

A variation of the setting considered in the previous section is the setting where the packets are injected with no prescribed path to follow. Only the source and the destination are given, and the protocol is free to chose the path for every packet. These need not even be simple paths. The question of stability in this setting is considered e.g. in [1, 7].

A schedule for a sequence of packets in this setting specifies for each packet when to move (as in the non-adaptive case), and in addition, over which edge.

The necessary and sufficient conditions for a sequence of the non-adaptive case to allow stability can be translated to the adaptive setting. That is, a sequence of packets in the adaptive case allows stability if and only if one can associate with each packet a path such that the resulting instance of the non-adaptive case allows stability. More formally,

Fact 5: Consider a sequence of packets for the adaptive case, (s_i, d_i), $i \geq 1$. Then, there is a feasible schedule for that sequence and a constant $B < \infty$ such that at any time there are at most B packets in the network, if and only if there is a sequence of paths π_i, $i \geq 1$, one path for each packet, such that the instance (s_i, d_i, π_i), $i \geq 1$, is in $\mathcal{ST}$.

Proof:

One direction is trivial: if the created instance is in $\mathcal{ST}$ then the paths, schedule and bound B guaranteed by it being in $\mathcal{ST}$ can be applied in the adaptive case to guarantee that at most B packets are present in the network at any time.

For the second direction observe that if at any time there are at most B packets present in the network, then all but at most B packets reach their destination according to the schedule. We now define paths for all the packets in the sequence. For each packet that reaches its destination, p_i, define its path π_i to be the path it followed according to the schedule of the adaptive case, eliminating any cycles. For any other packet (i.e., a packet that never reaches its destination) define its path to be an arbitrary path from its source to its destination. The non-adaptive instance created by associating these paths to the packets is in $\mathcal{ST}$ since it has a feasible schedule and a bound $B < \infty$ on the number of packets in the network: the schedule is the schedule of the adaptive case for all packets that reach their destination (eliminating from the schedule any cycles in the original path), and an empty schedule for packets that do not reach their destination (i.e., these packets never leave their source nodes). The same bound B on the number of packets present in the network at any given time applies. $\square$

We note that for the non-adaptive case it is known that for any sequence in $\mathcal{ST}$, a schedule that maintains stability can be computed by an on-line and distributed protocol [2, 7]. For the adaptive case, if a sequence of packets is such that an adversary can associate paths with the packets such that the resulting sequence is in $\mathcal{AQT}(r < 1)$ (and hence in $\mathcal{LB}(r < 1)$), then a schedule that maintains stability can be computed by an on-line and distributed protocol [1]. Gamarnik [7] shows that an on-line, but centralized, algorithm can compute a schedule that maintains stability for any sequence of packets that satisfy the following condition: an adversary can associate *fractional* paths with the packets, such that the created sequence is in $\mathcal{AQT}(r < 1)$. Gamarnik also shows [8] that the distributed protocol of [1] can be applied to such sequences albeit with worse performance (in terms of the parameters of the adversary). [1] For the special case where all packets have the same node as the destination node, Awerbuch et al. [4] recently showed an algorithm that maintains stability for any sequence such that the adversary can associate paths with the packets so that the resulting sequence is in $\mathcal{AQT}(r \leq 1)$. For the general case (i.e., arbitrary destinations) the question of whether there is an on-line, distributed or not, algorithm that computes a schedule that maintains stability given *any* instance of the adaptive problem for which stability is possible [2], remains open.

Acknowledgments Some of the the formal statements in this note were formulated following many useful discussions on Adversarial Queuing Theory with Bill Aiello, Allan Borodin, David Gamarnik, Rafi Ostrovsky and Eyal Kushilevitz. Thanks are also due to Ronny Lempel for useful discussions.

References

[1] W. Aiello, E. Kushilevitz, R. Ostrovsky, A. Rosén, Adaptive Packet Routing for Bursty Adversarial Traffic. *Journal of Computer and System Sciences*, Vol. 60, No. 3, pp. 482–509, 2000.

[2] M. Andrews, B. Awerbuch, A. Fernández, J. Kleinberg, T. Leighton, and Z. Liu, "Universal Stability Results for Greedy Contention-Resolution Protocols", JACM 48(1), pp. 39–69, 2001.

[1] In fact it is shown in [8] that any sequence of packets for which fractional paths create an instance in $\mathcal{AQT}(r < 1)$, has also integral paths that create an instance in $\mathcal{AQT}(r < 1)$.

[2] i.e., including instances for which the adversary can only associate paths for the packets such that the resulting instance is in $\mathcal{ST} \setminus \mathcal{AQT}(r < 1)$.

[3] M. Andrews, and L. Zhang, The Effect of Temporary Sessions on Network Performance. In *Proc. of the 11th SODA*, pp. 448-457, 2000.

[4] B. Awerbuch, P. Berenbrink, A. Brinkmann, and C. Scheideler, Simple Routing Strategies for Adversarial Systems. In *Proc. of the 42nd IEEE Symposium on Foundations of Computer Science*, pp. 158-167, 2001.

[5] A. Borodin, J. Kleinberg, P. Raghavan, M. Sudan, and D. Wiliamson, "Adversarial Queuing Theory", JACM 48(1), pp. 13–38, 2001.

[6] D. Gamarnik, Stability of Adversarial Queues via Fluid Model. In *Proc. of the 39th FOCS*, pp. 60–70, 1998.

[7] D. Gamarnik, Stability of Adaptive and non-Adaptive Packet Routing Policies in Adversarial Queuing Networks. In *Proc. of the 31st STOC*, pp. 206–214, 1999.

[8] D. Gamarnik, Stability of Adaptive and non-Adaptive Packet Routing Policies in Adversarial Queuing Networks. Manuscript. 2000.