Stability of Networks and Protocols in the Adversarial Queueing Model for Packet Routing

Ashish Goel[*]

Stanford University
June 15, 1997

Abstract

The adversarial queueing theory model for packet routing was suggested by Borodin et al. [2]. We give a complete and simple characterization of all networks that are universally stable in this model. We also show that a specific greedy protocol, SIS (Shortest In System), is stable against stochastic adversaries with exponentially vanishing tails.

1 The Adversarial Model for Packet Injection

In traditional queueing theory, the source which generates network traffic is typically assumed to be stochastic. Adversarial Queueing Theory developed out of a recent need for more robust models for these sources. The growing complexity of network traffic makes it increasingly unrealistic to model traffic as, say, a Poisson stream. It is therefore desirable to have a general robust framework which makes as few assumptions about the network traffic as possible. Such a framework was developed by Borodin et al. [2] in the context of packet routing and several very interesting results in this model were proved by Andrews et al. [1]. We refer the reader to [2, 1] for a more thorough motivation of the adversarial model. In this model, packets are injected into the network by an adversary. To keep things simple, it is assumed that the route of each packet is given along with the packet itself. The adversary is limited in the following way: over any w consecutive steps, the adversary can inject at most $\lceil w(1-\epsilon)\rceil$ path packings into the network, where a path packing is an edge disjoint collection of simple paths in a network. Each path can be looked upon as a route that a packet needs to follow. A single path packing may, therefore, correspond to many packets. The parameter w is called the burst size, and $1-\epsilon$ is the injection rate. Such adversaries are called (w,ϵ)-adversaries.

Once injected, packets follow their routes one edge at a time till they reach their destination. Only one packet may cross an edge during one time step, so a protocol is needed to decide which packet should cross an edge if more than one packets are waiting. A protocol is said to be greedy if it always forwards a packet on an edge if there are packets waiting to cross that edge. Any non-greedy protocol can be simulated by a (centralized) greedy protocol. Distributed greedy protocols are the simplest to implement and reason about, and we limit ourselves to such protocols.

[*] Department of Computer Science, Stanford CA 94305. Research supported by ARO Grant DAAH04-95-1-0121 and NSF Grant CCR9304971.

Definition 1.1 *A protocol P is stable on network G (or equivalently, the pair (P, G) is stable) if for all (w, ϵ)-adversaries, the maximum number of packets in the system, as well as the maximum delay a packet experiences, is bounded.*

Definition 1.2 *A protocol P is universally stable if (P, G) is stable for all networks G. Similarly, a network G is universally stable if (P, G) is stable for all greedy protocols P.*

Some common greedy protocols are defined below. To define a greedy protocol, it is sufficient to specify how contention is resolved when more than one packets want to traverse the same edge.

FIFO: First In First Out.

SIS: Shortest In System – the packet which was injected last wins.

LIS: Longest In System – the packet which was injected first wins.

NTG: Nearest To Go – the packet nearest its final destination wins.

FTG: Furthest to Go – the packet furthest away from its final destination wins.

One of the most interesting results of Andrews et al. is that several natural greedy protocols for packet routing are not universally stable. Specifically, FIFO is not stable! But fortunately, several natural greedy protocols are stable on all networks – for example, LIS and SIS, which give priority to the oldest and latest packet, respectively. Andrews et al. show a similar dichotomy for networks. They show that all greedy protocols are stable on rings, trees, and DAGs. But they do not characterize all universally stable networks. We give a complete characterization of such networks in Section 2. Specifically, we show that a connected undirected graph is universally stable if and only if it has at most one cycle. For directed graphs, we give a simple decision procedure to determine universal stability.

Borodin et al. [2] introduced the notion of stochastic adversaries. The number of packet packings injected by a stochastic adversary during time step i is X_i, where X_is are i.i.d. random variables with mean less than 1. However, the path packings chosen by the adversary need not come from a probability distribution. Hence stochastic adversaries are more powerful than traditional source models, where packet injection is probabilistic for each source-destination pair. Borodin et al. give some results for stability of protocols against stochastic adversaries on directed cycles. In Section 3, SIS is shown to be stable on all networks against all adversaries with exponentially vanishing tails (A stochastic adversary is said to have an exponentially vanishing tail if the random variables X_i have an exponentially vanishing tail.).

2 Universal Stability of Networks

Andrews et al. [1] prove that not all graphs are universally stable, by giving a counter-example. They also attempt to characterize graphs that are universally stable. They claim that for undirected graphs, the property of universal stability of a graph is closed under minor inclusion. Using the results of Robertson and Seymour [3], it follows that universal stability can be decided in polynomial time. While this approach does offer a lot of insight, it is unsatisfactory for several reasons. First, the results of Robertson and Seymour are non-constructive. We are guaranteed that there are only finitely many minor-minimal graphs which are not universally stable, but there is no general technique for finding these forbidden minors. Moreover, the results of Andrews et al. hold only for undirected graphs. Andrews et al. do observe that large classes of commonly used networks are unstable.

A simple and constructive characterization of universally stable graphs is given in this section. An undirected connected graph G with n vertices is shown to be universally stable iff it has at most n edges (Theorem 2.7). For directed graphs, the set of forbidden minors is given (Theorem 2.5) along with a simple decision procedure (Theorem 2.6).

Definition 2.1 *For any directed graph G, the line graph $L(G)$ of G is defined in the natural way: the edge set of G forms the vertex set of $L(G)$ and there is an edge in $L(G)$ from e_1 to e_2 if, in the graph G, the head of e_1 coincides with the tail of e_2.*

Lemma 2.1 *If digraphs G_1 and G_2 are universally stable, then so is any graph G formed by joining them with edges that go only from G_1 to G_2.*

Proof: Assume that the adversary we are working against has rate $1 - \epsilon$ and burst size w. Since G_1 is stable, any packets originating in G_1 get out of G_1 within T_1 time steps, where T_1 is some function of w and ϵ. Some of these packets may then enter G_2. Now, during a time window T_2 units long, any new packets that enter G_2 must have been introduced during $T_1 + T_2$ contiguous units. The number of path packings introduced during this interval can be at most $(T_1 + T_2 + w)(1 - \epsilon)$. Choose any ϵ' such that $0 < \epsilon' < \epsilon$. Now, set $T_2 = (T_1 + w)(1 - \epsilon)/(\epsilon - \epsilon')$. During any window of size $t \geq T_2$ at most $t(1-\epsilon') - (T_2(\epsilon - \epsilon') - (T_1 + w)(1 - \epsilon)) = t(1 - \epsilon')$ path packings are introduced into G_2. This implies that these packets could have been introduced by an adversary of burst size T_2 and rate ϵ'. But by definition of universal stability, G_2 is stable against such an adversary. Therefore, the traversal time for a packet as well as the queue lengths are bounded in G. ■

Lemma 2.1 implies that all acyclic digraphs are universally stable. 2.1.

Corollary 2.1 *A digraph is universally stable iff all of its strongly connected components are universally stable.*

We now focus on graphs that are strongly connected. The following result by Andrews et al. [1] will be useful:

Lemma 2.2 *(Andrews et al.[1]) The directed cycle on any number of vertices is universally stable.*

The same result holds for the undirected cycle, as the undirected cycle can be modeled as two disconnected directed cycles. The next logical question to ask is the following: Does there exist a strongly connected digraph that is more complicated than a cycle and is still universally stable? The answer is no. Consider the following two simple digraphs, H_1 and H_2, also drawn in Figure 1.

$\underline{H_1:}$ There are just two vertices u and v with two parallel edges e_1 and e_2 from u to v and an edge f from v to u.

$\underline{H_2:}$ There are three vertices u, v and w with the following edges: $e_1 = (uv)$, $e_2 = (vu)$, $f_1 = (uw)$, $f_2 = (wu)$.

Neither of these two simple graphs is universally stable. For now, we ignore the fact that the paths specified have to be simple. We will come back to it later in the section.

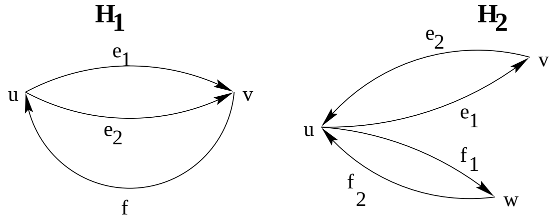

Figure 1: The graphs H_1 and H_2.

Lemma 2.3 *H_1 is not universally stable.*

Proof: To prove instability, the adversary and the greedy protocol are allowed to work in collusion. Suppose there are s packets waiting to cross edge f, where s is larger than some large enough constant. The adversary introduces packets at a rate $r < 1$, and operates in four rounds. At the end of the fourth round, there will be more than s packets waiting to cross edge f. Our proof goes along the lines of a similar proof in [1] for a larger graph.

The first round lasts for s steps: the adversary introduces sr packets of the form (fe_2) and delays all of these. At the beginning of the second round, there are sr packets of type (fe_2) waiting at edge f. During the second round (duration sr), sr^2 packets of type (e_2) and another sr^2 packets of type (fe_1) are introduced. Further, all these packets are delayed. At the beginning of round 3, there are sr^2 packets each of type (e_2) and (fe_1). The adversary now introduces (in sr^2 steps) sr^3 packets each of types (e_2) and (e_1f), delaying each of these. At the beginning of the fourth round, there are sr^3 packets each of type (e_2) and (e_1f). During the fourth round (duration sr^3) the adversary introduces sr^4 packets each of types (e_2f) and (e_1). The packets of type (e_2f) are delayed, whereas packets of type (e_1) are allowed to pass through, delaying packets of type (e_1f) from the previous round. Notice that this is the only step when the greedy protocol differs from LIS. At the end of the fourth round, there are sr^4 packets each of types (e_1f) and (e_2f) waiting. All these packets need to cross edge f.

During the above procedure, the number of packets waiting at edge f went from s to $2sr^4$. If $r > 0.5^{1/4} \approx 0.84$, then the number of packets goes up, and the adversary can use this procedure over and again to make the number of packets in the system unbounded.

In the above proof, the number of packets at the end of round 4 will actually be $2sr^4$ minus some constant. But since s was chosen to be larger than a large enough constant, this is not a problem. To initially generate a constant number of packets at edge f, we can attach a large acyclic graph at u, where the newly added acyclic portion is used to generate the initial packets. The new graph (ie. H_1 with an acyclic graph attached) is not universally stable. Corollary 2.1 now implies that H_1 is not universally stable. ■

Simple modifications to the above proof result in the following corollaries.

Corollary 2.2 *The pair (FIFO,H_1) is not stable.*

Corollary 2.3 *Any graph obtained by replacing the edges e_1, e_2 and f in H_1 by disjoint directed paths is not universally stable. In particular, such a graph is not stable for FIFO.*

Lemma 2.4 *H_2 is not universally stable.*

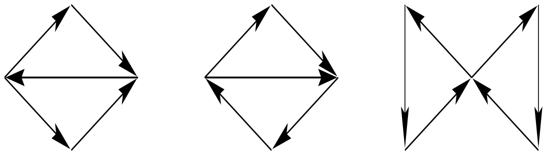

Figure 2: The forbidden minors for a universally stable digraph.

Proof: The proof is similar to that for Lemma 2.3. The adversary again operates in four rounds. Suppose there are s packets waiting to cross edge e_1. In the first round (duration s), the adversary introduces sr packets of type $(e_1e_2f_1f_2)$ and delays them. In the second round (duration sr), the avdersary injects sr^2 packets each of types (e_2) and (f_2) and delays them. In the third round (duration sr^2), the adversary injects sr^3 packets each of types (e_2e_1) and (f_2) and again delays the newly introduced packets. In the fourth round (duration sr^3), the adversary introduces sr^4 packets each of types (e_2) and (f_2e_1). During this round, the adversary delays packets of type (e_2e_1) from the previous round and (f_2e_1) from the current round. Therefore, at the end of the fourth round, there are $2sr^4$ packets that need to cross edge e_1. ■

Corollary 2.4 *The pair (FIFO,H_2) is not stable. Any graph obtained by replacing the edges e_1, e_2, f_1 and f_2 in H_2 by disjoint directed paths is not universally stable. In particular, such a graph is not stable for FIFO.*

Any strongly connected digraph must either be a cycle, or it must consist of at least two cycles which either share an edge or a vertex. If these cycles share an edge, this digraph would be unstable by Lemma 2.3. If they share a vertex, this digraph would be unstable by Lemma 2.4. Imposing the restriction that each packet must follow a simple path, the forbidden minors for universally stable digraphs are given in Figure 2.

Theorem 2.5 *A digraph G is universally stable iff it does not contain any of the forbidden minors drawn in Figure 2.*

There is an even simpler procedure to check universal stability of a digraph G. Define a feasible line graph $L_F(G)$ of a digraph G as follows: the edge set of G forms the vertex set of $L(G)$ and there is an edge in $L(G)$ from e_1 to e_2 if, in the graph G, the head of e_1 coincides with the tail of e_2, and e_1 and e_2 do not form a directed cycle. The last condition captures the restriction that a packet cannot traverse the same edge in two directions consecutively.

Theorem 2.6 *A digraph G is universally stable iff in its feasible line graph $L_F(G)$, all strongly connected components are simple cycles.*

The above theorems completely characterize universally stable graphs, and give an efficient ($O(mn)$) procedure for deciding universal stability.

For undirected graphs, the characterization is even simpler.

Theorem 2.7 *A connected undirected graph with n vertices is universally stable iff it has no more than n edges.*

3 Against Stochastic Adversaries

Borodin et al. [2] introduce the notion of stochastic adversaries. They show that several commonly used protocols such as LIS and FTG are stable on a directed cycle against stochastic adversaries with exponentially vanishing tails[1]. We show that SIS is stable on all networks against all exponential adversaries (ie. stochastic adversaries with exponentially vanishing tails).

Theorem 3.1 *SIS is stable against all exponential adversaries on all graphs. Further, the random variables corresponding to queue lengths and packet delays have exponentially vanishing tails.*

The proof of Theorem 3.1 is omitted from this version. Instead, we prove the same result against 0-1 adversaries, defined below. The proof of Theorem 3.1 is similar to that of Theorem 3.2 and Corollary 3.3 but we need to use some Chernoff bounds to make the proof go through.

Definition 3.1 *A 0-1 stochastic adversary generates 0 path packings with probability ϵ and 1 path packing with probability $1-\epsilon$ during any single time step, for some constant $\epsilon > 0$.*

We first define a non-greedy protocol, SIS-NG, which takes at least as much time to route any packet as SIS. SIS-NG is then shown to be stable against a 0-1 stochastic adversary.

Definition 3.2 *The non greedy protocol SIS-NG does the following: a packet p is blocked at each edge in its path till the number of path packings introduced after p becomes less than the duration for which p has been waiting at this edge.*

Informally, SIS-NG assumes that each packet gets blocked once at each edge along its path by each of the path packings introduced after it.

Lemma 3.1 *Given the same sequence of requests, SIS takes at most as long to route any packet as SIS-NG.*

Theorem 3.2 *SIS-NG is stable against all 0-1 stochastic adversaries on arbitrary graphs.*

Proof: In this proof, the protocol will implicitly be assumed to be SIS-NG, and the mean number of path packings injected by the adversary during a single step will be $1-\epsilon, 0 < \epsilon < 1$.

Let $P_i(\alpha)$ represent the probability of α path packings being generated by the 0-1 adversary between the time when packet p was introduced and when it crosses the ith edge on its path. Clearly, $P_0(0) = 1$, and $P_0(\alpha) = 0$ for all $\alpha > 0$. If there are β packets in the system when p crosses its ith edge, and there were α packets when p crossed its $(i-1)$th edge, then during the previous β steps, exactly $\beta - \alpha$ path packings must have been introduced, and no packet must have been introduced during the current step. This argument can be formalized to obtain the following recursive equation:

$$P_i(\beta) = \sum_{\alpha \le \beta} \binom{\beta}{\alpha} P_{i-1}(\alpha)\epsilon^{\alpha+1}(1-\epsilon)^{\beta-\alpha}$$

[1] These results were originally reported for adversaries with bounded variance, but the restriction that the input process have an exponential tail is essential.

The solution to the above equation is $P_i(\beta) = \epsilon^i(1-\epsilon^i)^\beta$. This can be verified inductively. Clearly, this solution is valid for $i = 1$. Suppose the solution is valid for all $i \leq j$. Now,

$$\begin{aligned} P_{j+1}(\beta) &= \sum_{\alpha \leq \beta} \binom{\beta}{\alpha} \epsilon^j (1-\epsilon^j)^\alpha \epsilon^{\alpha+1}(1-\epsilon)^{\beta-\alpha} \\ &= \epsilon^{j+1} \sum_{\alpha \leq \beta} \binom{\beta}{\alpha} \left(\epsilon(1-\epsilon^j)\right)^\alpha (1-\epsilon)^{\beta-\alpha} \\ &= \epsilon^{j+1}\left(1-\epsilon+\epsilon(1-\epsilon^j)\right)^\beta \\ &= \epsilon^{j+1}(1-\epsilon^{j+1})^\beta \end{aligned}$$

, which completes the inductive proof. Let d be the maximum distance that a packet p needs to cover (clearly, $d \leq m$). Let β_d represent the number of path packings introduced after p but before p completes its journey. Expected delay for packet p is at most

$$\begin{aligned} \mathbf{E}(d + d\beta_d) &= d\left(1 + \sum_{\beta \geq 0} \beta \epsilon^d (1-\epsilon^d)^\beta\right) \\ &= d\left(1 + \frac{1-\epsilon^d}{\epsilon^d}\right) \\ &= \frac{d}{\epsilon^d} \end{aligned}$$

This completes the proof of stability of SIS-NG against 0-1 adversaries. ■

Corollary 3.3 *If SIS-NG is used against 0-1 adversaries, the distributions of queue lengths and packet delays have exponentially vanishing tails.*

Proof: Prob(delay $> d + kd$) $\leq$ Prob($\beta_d > k$) $\leq \epsilon^d \sum_{\beta > k}(1-\epsilon^d)^\beta = (1-\epsilon^d)^{k+1}$, which proves that packet delays have exponentially vanishing tails.

Prob(number of packets $> dkm + dm$) $<$ Prob(oldest packet is at least $dk + d$ units old). Now, there can be at most m packets per time step, and incrementing k by one corresponds to d time steps. Using the bound on delay obtained earlier in this proof, the above probability is at most $md \sum_{\beta > k}(1-\epsilon^d)^\beta = (dm/\epsilon^d)(1-\epsilon^d)^k$, which has an exponentially vanishing tail. ■

The expected delay is exponential in d. But for a large class of networks, d can be logarithmic in the network size, which would result in polynomial delays and queue sizes. Also, by Lemma 3.1, all the above results hold for SIS as well.

References

[1] M. Andrews, B. Awerbuch, A. Fernandez, J. Kleinberg, T. Leighton, and Z. Liu. Universal stability results for greedy contention-resolution protocols. *37th IEEE symposium on Foundations of Computer Science*, pages 380–389, 1996.

[2] A. Borodin, J. Kleinberg, P. Raghavan, A. Sudan, and D. Williamson. Adversarial queueing theory. *28th ACM Symposium on Theory of Computing*, pages 376–385, 1996.

[3] N. Robertson and P.D. Seymour. Recent results on graph minors. *Surveys in Combinatorics*, Cambridge University Press, 1985.