

Adversarial Queueing Model for Continuous Network Dynamics

Maria J. Blesa, Daniel Calzada, Antonio Fernández, Luis López
Andrés L. Martínez, Agustín Santos, María J. Serna

Report LSI-05-10-R

first version 29th March 2005

revised version 26th May 2005



Departament de Llenguatges i Sistemes Informàtics
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Adversarial Queueing Model for Continuous Network Dynamics [★] ^{★★}

María J. Blesa¹, Daniel Calzada², Antonio Fernández³, Luis López³,
Andrés L. Martínez³, Agustín Santos³, and María J. Serna¹

¹ ALBCOM, LSI, Universitat Politècnica de Catalunya, E-08034 Barcelona, Spain
{mjblesa,mjserna}@lsi.upc.edu

² ATC, EUI, Universidad Politécnica de Madrid, E-28031 Madrid, Spain
dcalzada@eui.upm.es

³ LADyR, GSyC, ESCET, Universidad Rey Juan Carlos, E-28933 Madrid, Spain
{anto,llopez,aleonar,asantos}@gsyc.escet.urjc.es

Abstract. In this paper we start the study of generalizing the Adversarial Queueing Theory (AQT) model towards a continuous scenario in which the usually assumed synchronicity of the evolution is not required anymore. We consider a model, named *continuous AQT* (CAQT), in which packets can have arbitrary lengths, and the network links may have different speeds (or bandwidths) and propagation delays. We show that, in such a general model, having bounded queues implies bounded end-to-end packet delays and vice versa. From the network point of view, we show that networks with directed acyclic topologies are universally stable, i.e., stable independently of the protocols and the traffic patterns used in it, and that this even holds for traffic patterns that make links to be fully loaded. Concerning packet scheduling protocols, we show that the well-known LIS, SIS, FTG and NFS protocols remain universally stable in our model. We also show that the CAQT model is strictly stronger than the AQT model by presenting scheduling policies that are unstable under the former while they are universally stable under the latter.

1 Introduction

The Adversarial Queueing Theory (AQT) model [1,2] has been used in the latest years to study the stability and performance of packet-switched networks. The AQT model, (like other adversarial models) allows to analyze the system in a worst-case scenario, since it replaces traditional stochastic arrival assumptions in the traffic pattern by worst-case inputs. In this model, the arrival of packets to

[★] Partially supported by EU IST-2001-33116 (FLAGS), IST-2004-15964 (AEOLUS), COST-295 (DYNAMO), and by Spanish MCyT TIC2002-04498-C05-03 (TRACER), by the Comunidad de Madrid 07T/0022/2003, and by the Universidad Rey Juan Carlos project PPR-2004-42.

^{★★} We address the reader to the appendix for details on the proofs of the theorems.

the network (i.e., the traffic pattern) is controlled by an adversary that defines, for each packet, the place and time in which it joins the system and, additionally it might decide the path it has to follow. In order to study non-trivial overloaded situations, the adversary is restricted so that it can not overload any link (in an amortized sense). Under these assumptions, we study the *stability* of network systems $(\mathcal{G}, \mathcal{P}, \mathcal{A})$, which are represented by three elements: the network topology \mathcal{G} , the protocol \mathcal{P} used for scheduling the packets at every link, and the adversary \mathcal{A} , which defines the traffic pattern. Stability is the property that at any time the maximum number of packets present in the system is bounded by a constant that may depend on system parameters.

The original AQT model assumes a synchronous behavior of the network, that evolves in steps. In each step at most one packet crosses each link. Implicitly, this assumption means that all the packets have the same size and all the links induce the same delay in each packet transmission. There have been generalizations of the AQT model to dynamic networks, like networks with failures [3,4,5,6] and networks with links with different and possibly variable capacities or delays [7,8,9]. These works still assume a synchronous network evolution, to the point that, for instance in [7] all capacities and slow-downs must have an integral value. To the best of our knowledge, the work included in [10] is the only generalization of the AQT model considering packets of arbitrary lengths (up to a maximum) or links of arbitrary (not integral) speeds and propagation delays. In that model the adversary is more powerful than in the AQT model, and a sufficient condition on the adversary injection rate for assuring network stability is presented.

In this paper we propose a generalization of the AQT model allowing arbitrary packet lengths, link speeds (bandwidths), and link propagation delays. The network traffic flow is considered to be continuous in time. Since we do not restrict a synchronous system evolution anymore, we call this model *continuous* AQT (CAQT). Note that all the results for the AQT model which are concerned with instability, also hold for our CAQT model, e.g., the instability of the FIFO protocol at any constant rate [11]. The CAQT model is inspired in the traffic conditions of the session oriented model proposed by Cruz [12], which is widely studied in the communication networks literature. The synchronous assumptions of the AQT model limit the capacity of the adversary as well. In the CAQT model the adversary is more powerful, and any instability result shown in the AQT model can be reproduced in ours.

We show that several results from the AQT model still hold in the CAQT model. First, we show that having bounded queue size implies having bounded packet end-to-end delays and vice versa. Then, we show that networks with a directed acyclic graph (DAG) topology are always stable even if the links are fully loaded. Concerning packet scheduling protocols, we show that the well-known LIS, SIS, FTG and NFS protocols remain universally stable in our model. Finally, we show that some protocols whose policies are based on criteria concerning the length of the packets, the bandwidth of the links or their propagation delay, can configure unstable systems.

2 System Model

Like AQT, the CAQT model represents a network as a finite directed graph \mathcal{G} in which the set of nodes $V(\mathcal{G})$ represent the hosts, and the set of edges $E(\mathcal{G})$ represent the links between those hosts. Each link $e \in E(\mathcal{G})$ in this graph has associated a positive but not infinite transmission speed (a bandwidth), denoted as B_e . The bandwidth of a link establishes how many bits can be transmitted in the link per second. Instead of considering the bandwidth as a synonym for parallel transmission, we relate the bandwidth to the transmission velocity. We consider that only one bit can be put in a link $e \in E(\mathcal{G})$ at each time, and that conceptually the sender puts the associated signal level to the corresponding bit for $1/B_e$ seconds for each bit. This means that a bit can be partially transmitted or partially received at a given time. Let us denote as $B_{\min} = \min_{e \in E(\mathcal{G})} B_e$ and as $B_{\max} = \max_{e \in E(\mathcal{G})} B_e$ the minimum and maximum bandwidth, respectively, of the edges in \mathcal{G} .

Each link $e \in E(\mathcal{G})$ has also associated a propagation delay, denoted here as P_e , being $P_e \geq 0$. This delay, measured in seconds, establishes how long it takes for a signal (the start of a bit, for instance) to traverse the link. This parameter has to do with the propagation speed of the changes in the signal that carry the bits along the physical medium used for the transmission. We will denote as $P_{\min} = \min_{e \in E(\mathcal{G})} P_e$ and $P_{\max} = \max_{e \in E(\mathcal{G})} P_e$ the minimum and maximum propagation delay, respectively, of the edges in \mathcal{G} .

Like in the AQT model, we assume the existence of an adversary that defines the traffic pattern of the system by choosing when and where to inject packets into the system, and the path to be followed by each of them. We assume that a packet path is edge-simple, in the sense that it does not contain the same edge more than once (it can visit the same vertex several times, though). Again, we restrict the adversary so that it can not trivially overload any link. To do so, we also define two system-wide parameters: the *injection rate* r (with $0 < r \leq 1$), and the *burstiness* b (with $b \geq 1$). For every link $e \in E(\mathcal{G})$, if we denote by $N_e(I)$ the total size (in bits) of the packets injected by the adversary in the interval I whose path contains link e , it must be satisfied that

$$N_e(I) \leq r|I|B_e + b.$$

We call an adversary \mathcal{A} that satisfies this restriction an (r, b) -adversary. The injection rate r is sometimes expressed alternatively as $(1 - \varepsilon)$, with $\varepsilon \geq 0$.

Regarding packet injections, we assume that the adversary injects packets instantaneously. From the above restriction, this implies that packets have a maximum size of b bits. In general, we will use L_p to denote the length (in bits) of a packet p , and $L_{\max} = \max_p L_p \leq b$ to denote the maximum packet length. Once a packet p starts being transmitted through a link $e \in E(\mathcal{G})$, it will only take $P_e + L_p/B_e$ units of time more until it crosses it completely.

Let us now look at the packet switching process. We assume that each link has associated an *output queue*, where the packets that have to be sent across the link are held. The still unsent portion of a packet that is being transmitted

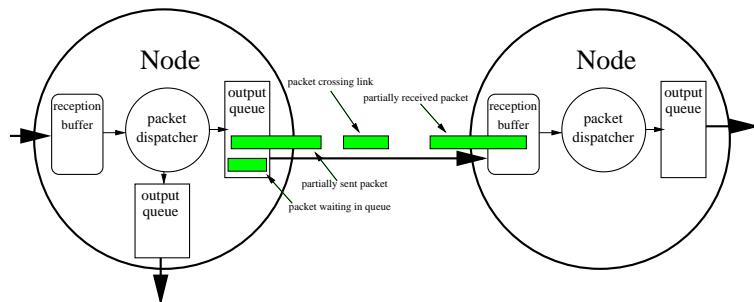


Fig. 1. Elements involved in the nodes and links of the network in the CAQT model.

is also held in this queue. In fact, if a bit has only been partially sent, we assume that the still unsent portion of the bit still resides in this queue. A packet can arrive to a node either by direct injection of the adversary or by traversing some incoming link. In the latter case we assume that only full packets are dispatched (moved to an output queue). Hence, we assume that each link has a *reception buffer* in the receiving node where the portion of a partially received packet is held. As soon as the very last bit of a packet is completely received, the packet is dispatched instantaneously (by a packet dispatcher) to the corresponding output queue (or removed, if this is the final node of the packet). Figure 1 shows these network elements.

The definition of *stability* in the CAQT model is analogous to the definitions stated under other adversarial models.

Definition 1. Let \mathcal{G} be a network with a bandwidth and a propagation delay associated to each link, \mathcal{P} be a scheduling policy, and \mathcal{A} an (r, b) -adversary, with $0 < r \leq 1$ and $b \geq 1$. The system $(\mathcal{G}, \mathcal{P}, \mathcal{A})$ is stable if, at every moment, the total number of packets (or, equivalently, the total number of bits) in the system is bounded by a value C , that can depend on the system parameters.

We also use common definitions of *universal stability*. We say that a scheduling policy \mathcal{P} is universally stable if the system $(\mathcal{G}, \mathcal{P}, \mathcal{A})$ is stable for each network \mathcal{G} and each (r, b) -adversary \mathcal{A} , with $0 < r < 1$ and $b \geq 1$. Similarly, we say that a network \mathcal{G} is universally stable if the system $(\mathcal{G}, \mathcal{P}, \mathcal{A})$ is stable for each *greedy* scheduling policy⁴ \mathcal{P} and each (r, b) -adversary \mathcal{A} , with $0 < r < 1$ and $b \geq 1$.

Some additional notation is needed to describe the state of the queues and the packets at a specific time step. We will use $Q_t(e)$ to denote the queue size

⁴ Greedy (or work-conserving) protocols are those forwarding a packet across a link e whenever there is at least one packet waiting to traverse e . Three types of packets may wait to traverse a link in a particular instant of time: the incoming packets arriving from adjacent links, the packets injected directly into the link, and the packets that could not be forwarded in previous steps. At each time step, only one packet from those waiting is forwarded through the link; the rest are kept in a queue.

(in bits) of edge $e \in E(\mathcal{G})$ at time t , and define $Q_{\max}(e) = \max_t Q_t(e)$. Similarly, we will use $R_t(e)$ to denote the number of bits at time t that are crossing link e , or already crossed it but are still in its reception buffer at the target node of e . Then, we define $R_{\max}(e) = \max_t R_t(e)$. Observe that $R_{\max}(e) < P_e B_e + L_{\max}$ and is hence bounded. $A_t(e)$ will denote the number of bits in the system that require to cross e and still have to be transmitted across link e at time t . The bits in $Q_t(e)$ are included in $A_t(e)$, but those in $R_t(e)$ are not.

3 General Results

We point out some general results that apply to every system $(\mathcal{G}, \mathcal{P}, \mathcal{A})$ in the CAQT model, independently of which is the network topology, the protocol used and the traffic pattern.

3.1 Relation between maximum queue size and maximum delay

We show that for injection rate $r < 1$, having bounded queues is equivalent to having bounded end-to-end packet delay. This generalizes a result from the AQT model to the stronger CAQT model.

Theorem 1. *Let \mathcal{G} be a network, \mathcal{P} a protocol, and \mathcal{A} an (r, b) -adversary with $r \leq 1$ and $b \geq 1$. If the maximum end-to-end delay is bounded by D in the system $(\mathcal{G}, \mathcal{P}, \mathcal{A})$, then the maximum queue size of an edge e is bounded by $(D - P_e)B_e$.*

Proof: We prove the claim by contradiction. Suppose there is some time t at which e has $Q_t(e) > (D - P_e)B_e$ bits in its queue. Then, the last packet p to completely cross e (out of those with bits in the queue at time t) will do so at a time

$$t' \geq t + \frac{Q_t(e)}{B_e} + P_e > t + \frac{(D - P_e)B_e}{B_e} + P_e = t + D.$$

Therefore, the end-to-end delay of p cannot be bounded by D . ■

Theorem 2. *Let \mathcal{G} be a network with $m = |E(\mathcal{G})|$ links, \mathcal{P} a greedy protocol, and \mathcal{A} an (r, b) -adversary, with $r = 1 - \varepsilon < 1$ and $b \geq 1$. If the maximum queue size is bounded by Q in the system $(\mathcal{G}, \mathcal{P}, \mathcal{A})$, then the end-to-end delay of a packet p with path e_1, \dots, e_d is bounded by*

$$\sum_{i=1}^d \frac{mQ + \sum_{e \in E(\mathcal{G})} R_{\max}(e) + b}{\varepsilon B_{e_i}} + P_{e_i}.$$

Proof: We bound the time p takes to cross every edge e_i . Let us assume that p arrives at the queue of e_i at time t . Note that all the bits in the system are either in the output queues, crossing links, or in the reception buffers. Then, there are

at most $mQ + \sum_e R_{\max}(e)$ bits in the whole system at time t . Hence, the queue of e_i will be empty after at most an interval of time of length Δ , such that

$$\Delta B_e = mQ + \sum_e R_{\max}(e) + r\Delta B_e + b.$$

Hence, p will completely cross e by time

$$t' \leq t + \frac{mQ + \sum_e R_{\max}(e) + b}{(1-r)B_e} + P_e.$$

■

Then, the following corollary follows from the above two lemmas.

Corollary 1. *Let \mathcal{G} be a network, \mathcal{P} a greedy protocol, and \mathcal{A} an (r, b) -adversary, with $r < 1$ and $b \geq 1$. In the system $(\mathcal{G}, \mathcal{P}, \mathcal{A})$ the maximum end-to-end delay experienced by any packet is bounded if and only if the maximum queue size is bounded.*

3.2 Initial Configurations

The moment in which a system $(\mathcal{G}, \mathcal{P}, \mathcal{A})$ starts its dynamics is usually denoted as t_0 , and usually $t_0 = 0$. The system can start either with no packet placed at any element of the network or with some kind of initial configuration. Usually, an initial configuration C_0 consists of a set S of packets located in the output queues of the network links. Trivially, any such initial configuration for a system $(\mathcal{G}, \mathcal{P}, \mathcal{A})$ can be built from an empty initial configuration at time 0 if we allow a large enough burstiness. Thus, any system $(\mathcal{G}, \mathcal{P}, \mathcal{A})$ that starts with a non-empty initial configuration as described can be simulated by another system $(\mathcal{G}, \mathcal{P}, \mathcal{A}')$ that starts with an empty one.

Theorem 3. *Let $A_S = \max_e A_0(e)$ be the maximum number of bits that have to be transmitted across any given edge in the paths of the set S of packets. A system $(\mathcal{G}, \mathcal{P}, \mathcal{A})$, where \mathcal{G} is a network, \mathcal{P} a greedy protocol, and \mathcal{A} an (r, b) -adversary with $r \leq 1$ and $b \geq 1$, that starts with an initial configuration C_0 consisting of a set S of packets in the network output queues can be simulated by a system $(\mathcal{G}, \mathcal{P}, \mathcal{A}')$ starting from an empty configuration, where \mathcal{A}' is an $(r, A_S + b)$ -adversary.*

Corollary 2. *A policy or network that is universally stable for systems with empty initial configurations is also universally stable for initial configurations in which there are initially packets in the network output queues.*

4 Stability of networks

We focus first on the study of stability of networks. We show that networks with a directed acyclic graph topology are universally stable, even when the traffic

pattern can fully load the links, i.e., even for the injection rate $r = 1$. Note that this proof is not a direct adaptation of the one in [1] for the corresponding analogous result in the AQT model.

Theorem 4. *Let \mathcal{G} be a directed acyclic graph, \mathcal{P} any greedy protocol, and \mathcal{A} any (r, b) -adversary with $r \leq 1$ and $b \geq 1$. The system $(\mathcal{G}, \mathcal{P}, \mathcal{A})$ is stable.*

Proof: Let us first denote with T_e the node at the tail of link e (i.e., the node that contains the output queue of e), for every edge $e \in E(\mathcal{G})$. Let us also denote with $in(v)$ the set of incoming links to node v , for all $v \in V(\mathcal{G})$. Let us define the function Ψ on the edges of \mathcal{G} as

$$\Psi(e) = Q_0(e) + b + R_{\max}(e) + \sum_{e' \in in(T_e)} \Psi(e').$$

If we call nodes without incoming links *sources*, we will show that $A_t(e) + R_t(e)$ is bounded by $\Psi(e)$, for all e and all $t \geq 0$, by induction on the maximum distance of T_e to a source (i.e., the length of the longest directed path from any source to T_e). Then, stability follows.

The base case of the induction is when T_e is a source. In this case, $A_t(e) = Q_t(e)$ and $\Psi(e) = Q_0(e) + b + R_{\max}(e)$. Let us fix a time t and consider two cases, depending on whether in the interval $[0, t]$ the output queue of e was empty at any time. If it was never empty, then by the restriction on the adversary and the fact that \mathcal{P} is greedy we have that

$$Q_t(e) \leq Q_0(e) + rtB_e + b - tB_e \leq Q_0(e) + b.$$

Otherwise, if time t' was the last time in interval $[0, t]$ that the queue of e was empty (i.e., $Q_{t'}(e) = 0$), by the same facts,

$$Q_t(e) \leq Q_{t'}(e) + r(t - t')B_e + b - (t - t')B_e \leq b.$$

Clearly, in either case,

$$A_t(e) + R_t(e) \leq Q_t(e) + R_{\max}(e) \leq Q_0(e) + b + R_{\max}(e) = \Psi(e).$$

Now, let us assume that the maximum distance of T_e to any source is $k > 0$. Note that for any edge $e' \in in(T_e)$, the maximum distance of $T_{e'}$ to a source is at most $k - 1$. Then, by induction hypothesis, we assume that $(A_t(e') + R_t(e')) \leq \Psi(e')$ for all $t \geq 0$ and all $e' \in in(T_e)$. Note that $A_t(e) \leq Q_t(e) + \sum_{e' \in in(T_e)} (A_t(e') + R_t(e'))$. Again, we fix t and consider separately the case when the output queue of e was never empty in the interval $[0, t]$ and the case when it was. In the first case we have that

$$\begin{aligned} A_t(e) &\leq Q_0(e) + rtB_e + b - tB_e + \sum_{e' \in in(T_e)} (A_0(e') + R_0(e')) \\ &\leq Q_0(e) + b + \sum_{e' \in in(T_e)} \Psi(e'). \end{aligned}$$

In the second case, if time t' was the last time in interval $[0, t]$ that the queue of e was empty (i.e., $Q_{t'}(e) = 0$), we have that

$$\begin{aligned} A_t(e) &\leq Q_{t'}(e) + r(t - t')B_e + b - (t - t')B_e + \sum_{e' \in \text{in}(T_e)} (A_{t'}(e') + R_{t'}(e')) \\ &\leq b + \sum_{e' \in \text{in}(T_e)} \Psi(e'). \end{aligned}$$

In either case, we have that

$$A_t(e) + R_t(e) \leq Q_0(e) + b + R_{\max}(e) + \sum_{e' \in \text{in}(T_e)} \Psi(e') = \Psi(e).$$

■

5 Stability of queueing policies

Stability can also be studied from the point of view of the protocols. Unstable protocols in the AQT model are also unstable in the CAQT model. In the following, we show that the so-called LIS, SIS, FTG and NFS protocols are universally stable in the CAQT model, as they were in the AQT model [2].

5.1 Universal stability of LIS

The LIS (*longest-in-system*) protocol gives priority to the packet which was earliest injected in the system. Independently of the network topology and the (r, b) -adversary, any system $(\mathcal{G}, \text{LIS}, \mathcal{A})$ is stable.

We first show a bound on the time that a packet needs to cross his path. Consider some packet p , injected at time T_0 , and whose path crosses edges e_1, e_2, \dots, e_d , in this order. We use T_i to denote the time instant in which p finishes crossing edge e_i , for $i = 1, \dots, d$. Let t denote some time in $[T_0, T_d]$. Let us denote by g_t the injection time of the oldest packet in the system at time t . We define $c = \max_{t \in [T_0, T_d]} t - g_t$.

Lemma 1. $T_d - T_0 \leq (1 - \varepsilon^d)c + \frac{(1 - \varepsilon^d)(\frac{b}{B_{\min}} + P_{\max})}{1 - \varepsilon}$.

Proof: Packet p reaches the tail of edge e_i at time T_{i-1} . Thus, from the definition of c , only packets injected in the interval $[T_{i-1} - c, T_0]$ can block p in the queue of e_i . The packets injected in that interval include p and have at most $r(T_0 - T_{i-1} + c)B_{e_i} + b$ bits. Hence,

$$\begin{aligned} T_i &\leq T_{i-1} + \frac{r(T_0 - T_{i-1} + c)B_{e_i} + b - L_p}{B_{e_i}} + \frac{L_p}{B_{e_i}} + P_{e_i} \\ &= \varepsilon T_{i-1} + (1 - \varepsilon)(c + T_0) + \frac{b}{B_{e_i}} + P_{e_i} \\ &\leq \varepsilon T_{i-1} + (1 - \varepsilon)(c + T_0) + \frac{b}{B_{\min}} + P_{\max} \end{aligned}$$

Thus, solving the recurrence, we obtain

$$\begin{aligned}
T_d &\leq ((1 - \varepsilon)(c + T_0) + \frac{b}{B_{\min}} + P_{\max}) \sum_{i=0}^{d-1} \varepsilon^i + \varepsilon^d T_0 \\
&= ((1 - \varepsilon)(c + T_0) + \frac{b}{B_{\min}} + P_{\max}) \frac{1 - \varepsilon^d}{1 - \varepsilon} + \varepsilon^d T_0 \\
&= (1 - \varepsilon^d) c + \frac{(1 - \varepsilon^d)(\frac{b}{B_{\min}} + P_{\max})}{1 - \varepsilon} + T_0
\end{aligned}$$

and the claim follows. \blacksquare

Theorem 5. *Let \mathcal{G} be a network, \mathcal{A} an (r, b) -adversary with $r = 1 - \varepsilon < 1$, and d the length of the longest simple directed path in \mathcal{G} . Then all packets spend less than $(\frac{b}{B_{\min}} + P_{\max})/(r\varepsilon^d)$ time in the system $(\mathcal{G}, \text{LIS}, \mathcal{A})$.*

Proof: Let $c' = (\frac{b}{B_{\min}} + P_{\max})/((1 - \varepsilon)\varepsilon^d)$ and assume that at time t is the first time that a packet p satisfies $t - g_t = c'$, i.e., the first time that a packet p has been in the system for c' time. We apply the previous Lemma 1 to this packet p . From that lemma, p should have been absorbed in at most

$$\begin{aligned}
&(1 - \varepsilon^d) c' + \frac{(1 - \varepsilon^d)(\frac{b}{B_{\min}} + P_{\max})}{1 - \varepsilon} \\
&= c' - \frac{(\frac{b}{B_{\min}} + P_{\max}) - (1 - \varepsilon^d)(\frac{b}{B_{\min}} + P_{\max})}{1 - \varepsilon} < c'
\end{aligned}$$

time, and a contradiction is reached. \blacksquare

Corollary 3. *Let \mathcal{G} be a network, \mathcal{A} an (r, b) -adversary with $r = 1 - \varepsilon < 1$, and d the length of the longest edge-simple directed path in \mathcal{G} . Then, the system $(\mathcal{G}, \text{LIS}, \mathcal{A})$ is stable, and there are always less than $(\frac{b}{B_{\min}} + P_{\max}) \varepsilon^{-d} B_{\max} + b$ bits trying to cross any edge e .*

5.2 Universal stability of SIS

The SIS (*shortest-in-system*) protocol gives priority to the packet which was injected the latest in the system. In the case of the SIS protocol, bounding the size of the packets recently injected is related to bounding the time that a packet p requires to cross the edge e . The following lemma provides us with such a bound:

Lemma 2. *Let p be a packet that, at time t , is waiting in the queue of edge $e \in E(\mathcal{G})$. At that instant, let $k - 1$ be the total size in bits of the packets in the system that also require e and that may have priority over p (i.e., that were injected later in the system). Then p will start crossing e in at most $(k + b)/(\varepsilon B_e)$ units of time.*

Proof: The lemma can be proved by contradiction. Suppose that the packet p is not transmitted across the edge e in the claimed units of time. Then, other packets different from p must have crossed the edge meanwhile. Those packets, must have also had priority over p ; however, during that time, the only packets in the system that have priority over p are, either those comprising the $k - 1$ bits existing at time t , or those that have been injected meanwhile. During that time, a total size of at most

$$(k - 1) + \left((1 - \varepsilon) \frac{k + b}{\varepsilon B_e} B_e + b \right)$$

bits belong to packets that have priority over p . The packet p would be transmitted right after these bits are transmitted; since these bits would require at most

$$\frac{(k - 1) + \left((1 - \varepsilon) \frac{k + b}{\varepsilon B_e} B_e + b \right)}{B_e} < \frac{k + b}{\varepsilon B_e}$$

units of time to be sent across e , this leads to a contradiction. \blacksquare

Observe that, once the packet p starts being transmitted through the link e , it will only take $P_e + L_p/B_e$ units of time more until it crosses it completely. Using the bound obtained in Lemma 2 in a recursive way, we can derive more general bounds, thus proving the universal stability of the SIS protocol.

Theorem 6. *Let \mathcal{G} be a network, \mathcal{A} an (r, b) -adversary with $r = 1 - \varepsilon < 1$ and $b \geq 1$, and d the length of the longest edge-simple directed path in \mathcal{G} . The system $(\mathcal{G}, \text{SIS}, \mathcal{A})$ is stable and, moreover:*

- no queue ever contains $k_d + L_{\max}$ bits, and
- no packet spends more than $(d(b + \varepsilon L_{\max}) + \sum_{i=1}^d k_i)/(\varepsilon B_{\min}) + dP_{\max}$ time in the system.

where k_i is defined according to the following recurrence:

$$k_i = \begin{cases} b & \text{for } i = 1 \\ k_{i-1} + (1 - \varepsilon) \left(\frac{k_{i-1} + b}{\varepsilon B_{\min}} + \frac{L_{\max}}{B_{\min}} + P_{\max} \right) B_{\max} + b & \text{for } 1 < i \leq d \end{cases}$$

Proof: We first show that, when a given packet p , with path $\Pi_p = \{e_1, \dots, e_{|\Pi_p|}\}$, $|\Pi_p| \leq d$, arrives at the queue of e_i , the total size of the packets with priority over p that also require some edge $e_j \in \Pi_p$ is at most $k_i - 1$.

This can be proved by induction on the position i of the edge e_i in the path Π_p . The claim holds for $i = 1$, since the only packets requiring any $e_j \in \Pi_p$ that at the time of p 's injection could have priority over p are those injected at the same time; whose total size is at most $b - 1$ bits (since $L_p \geq 1$). Let us now assume that the claim holds for some $i \geq 1$ (inductive hypothesis). Then, using the bound in Lemma 2, p will completely arrive at the queue of e_{i+1} in at most

$$\frac{k_i + b}{\varepsilon B_{e_i}} + \frac{L_p}{B_{e_i}} + P_{e_i}$$

units of time more. During that time, at most another

$$\begin{aligned} & (1 - \varepsilon) \left(\frac{k_i + b}{\varepsilon B_{e_i}} + \frac{L_p}{B_{e_i}} + P_{e_i} \right) B_{e_j} + b \\ & \leq (1 - \varepsilon) \left(\frac{k_i + b}{\varepsilon B_{\min}} + \frac{L_{\max}}{B_{\min}} + P_{\max} \right) B_{\max} + b \end{aligned}$$

bits of packets requiring an edge e_j could be injected. According to the SIS policy, those packets have priority over p . Thus, when p has completely arrived to e_{i+1} , there is a total amount of at most

$$(k_i - 1) + (1 - \varepsilon) \left(\frac{k_i + b}{\varepsilon B_{\min}} + \frac{L_{\max}}{B_{\min}} + P_{\max} \right) B_{\max} + b = k_{i+1} - 1$$

bits (belonging to packets in the system) which require an edge e and have priority over p . This validates the induction and proves the claim.

Suppose now that, at some point, there are $k_d + L_{\max}$ bits in the queue of some edge e . Then, the latest packet arrived into the queue found at least k_d bits there, which contradicts the above inductive proof. This proves the first claim of the theorem.

Also, by combining the initial claim with the bound in Lemma 2, we get that a packet p takes at most $\frac{k_i + b}{\varepsilon B_{e_i}} + \frac{L_p}{B_{e_i}} + P_{e_i}$ units of time to cross the i th edge in its path Π_p . Since every path defined over the graph \mathcal{G} has length at most d , then no packet spends more than

$$\sum_{i=1}^d \left(\frac{k_i + b}{\varepsilon B_{e_i}} + \frac{L_p}{B_{e_i}} + P_{e_i} \right) \leq \frac{d(b + \varepsilon L_{\max}) + \sum_{i=1}^d k_i}{\varepsilon B_{\min}} + dP_{\max}$$

units of time in the system. This proves the second claim of the theorem. \blacksquare

5.3 Universal stability of FTG

The FTG (*farthest-to-go*) protocol gives priority to the packet which still has to traverse the longest path until reaching its destination. We show that FTG is universally stable by using the fact that all the packets have to traverse at least one edge, and that all the packet go at most d edges further.

Theorem 7. *Let \mathcal{G} be a network with $m = |E(\mathcal{G})|$ links, \mathcal{A} an (r, b) -adversary with $r < 1$ and $b \geq 1$, and d the length of the longest edge-simple directed path in \mathcal{G} . The system $(\mathcal{G}, \text{FTG}, \mathcal{A})$ is stable and:*

- *there are never more than k_1 bits in the system,*
- *no queue ever contains more than $k_2 + b$ bits, and*
- *no packet spends more than $dP_{\max} + (d(b + \varepsilon L_{\max}) + \sum_{i=2}^d k_i)/(\varepsilon B_{\min})$ time in the system.*

where k_i is defined according to the following recurrence:

$$k_i = \begin{cases} 0 & \text{for } i > d \\ mk_{i+1} + mb + \sum_{e \in E(\mathcal{G})} R_{\max}(e) & \text{for } 1 \leq i \leq d \end{cases}$$

Proof: We claim that, for all i , the size of packets in the system that still have to cross at least i edges (they could be crossing one of these edges) is at most k_i . The proof is done by a backward induction on i .

The claim is trivial for $i > d$, since each packet has to cross at most d edges. By induction hypothesis we consider the claim true for $i + 1$. Consider now a particular edge e . We use $Q_t^i(e)$ to denote the bits in the queue of edge e that belong to packets that still have to cross *at least* i edges at time t . Let t' be the latest time no later than t such that $Q_{t'}^i(e) = 0$.⁵ Then, any packet p whose bits are accounted in $Q_t^i(e)$ either was in some other edge at time t' , and hence had at least $i + 1$ edges to cross, or else it has been injected after t' . Since we consider a greedy protocol, packets have been continuously sent across edge e in the interval $(t', t]$. Hence,

$$\begin{aligned} Q_t^i(e) &\leq k_{i+1} + ((1 - \varepsilon)(t - t')B_e + b) - (t - t')B_e \\ &\stackrel{(ind.hyp.)}{=} k_{i+1} + b - \varepsilon(t - t')B_e. \end{aligned}$$

With this result, we can calculate the following:

- (i) An upper bound on the total size of the packets *in the system* that still have to cross at least i edges:
Knowing that for a concrete edge e , $Q_t^i(e) \leq k_{i+1} + b - \varepsilon B_e(t - t')$, and that $R_t(e)$ bits are either crossing e or in its reception buffer at time t , we can conclude that the size of the packets in the system that still have to cross i or more edges at any time t is at most

$$\begin{aligned} \sum_{e \in E(\mathcal{G})} (Q_t^i(e) + R_t(e)) &\leq mk_{i+1} + mb + \sum_{e \in E(\mathcal{G})} (R_t(e) - \varepsilon(t - t')B_e) \\ &\leq mk_{i+1} + mb + \sum_{e \in E(\mathcal{G})} R_{\max}(e) \\ &= k_i, \end{aligned}$$

and hence the inductive step holds.

This trivially shows the first claim of the theorem, since no more than k_1 bits belong to packets that need to traverse at least one edge, which are all the packets in the system.

- (ii) An upper bound on the size of the queue of any edge e :
The maximum queue size of any edge e can be calculated when considering $i = 1$, since all packets in the queue need to cross at least one edge (e itself). Then, independently of the time instant, $Q_{\max}(e) = \max_t Q_t^1(e) \leq k_2 + b$.

⁵ Observe that, w.l.o.g. we can assume that the system had empty initial configuration (see Corollary 2), and thus such a time t' always exists.

(iii) An upper bound on $t - t'$:

Since for any concrete edge e , $Q_t^i(e) \geq 0$, we obtain that

$$t - t' \leq \frac{k_{i+1} + b}{\varepsilon B_e} \leq \frac{k_{i+1} + b}{\varepsilon B_{\min}},$$

and so the maximum amount of time that a packet p with path e_1, e_2, \dots, e_d spends in the system is bounded by

$$\sum_{i=1}^d \left(\frac{k_{i+1} + b}{\varepsilon B_{e_i}} + \frac{L_p}{B_{e_i}} + P_{e_i} \right) \leq \frac{d(b + \varepsilon L_{\max}) + \sum_{i=2}^d k_i}{\varepsilon B_{\min}} + dP_{\max}.$$

■

5.4 Universal stability of NFS

The NFS (*nearest-from-source*) protocol gives priority to the packet which is closest to its origin, i.e., which has traversed the less portion of its whole path. We show that NFS is universally stable by using a similar argument as the one used for FTG; however the bounds will be provided now taking the length of the longest path as a reference point.

Theorem 8. *Let \mathcal{G} be a network with $m = |E(\mathcal{G})|$ links, \mathcal{A} an (r, b) -adversary with $r < 1$ and $b \geq 1$, and d the length of the longest edge-simple directed path in \mathcal{G} . The system $(\mathcal{G}, \text{NFS}, \mathcal{A})$ is stable and:*

- *there are never more than k_d bits in the system,*
- *no queue ever contains more than $k_{d-1} + b$ bits, and*
- *no packet spends more than $dP_{\max} + (d(b + \varepsilon L_{\max}) + \sum_{i=1}^{d-1} k_i)/(\varepsilon B_{\min})$ time in the system.*

where k_i is defined according to the following recurrence:

$$k_i = \begin{cases} 0 & \text{for } i = 0 \\ mk_{i-1} + mb + \sum_{e \in E(\mathcal{G})} R_{\max}(e) & \text{for } 1 \leq i \leq d \end{cases}$$

Proof: We claim that, for all i , the total size of the packets in the system that have (completely) crossed less than i edges is at most k_i . The proof is done by induction on i .

The claim is trivial for $i = 0$, since it is not possible to cross less than 0 edges. By induction hypothesis we consider the claim true for $i - 1$. Consider now a particular edge e . We use $Q_t^i(e)$ to denote the bits in the queue of edge e that belong to packets that have crossed *less than* i edges at time t . Let t' be the latest time no later than t such that $Q_{t'}^i(e) = 0$.⁶ Then, any packet p whose

⁶ Observe that, w.l.o.g. we can assume that the system had empty initial configuration (see Corollary 2), and thus such a time t' always exists.

bits are accounted in $Q_t^i(e)$ either was at some other edge at time t' , and hence had crossed less than $i - 1$ edges, or else it has been injected after t' . Since we consider a greedy protocol, packets have been continuously sent across edge e in the interval $(t', t]$. Hence,

$$\begin{aligned} Q_t^i(e) &\leq k_{i-1} + ((1 - \varepsilon)(t - t')B_e + b) - (t - t')B_e \\ &\stackrel{(ind.hyp.)}{=} k_{i-1} + b - \varepsilon(t - t')B_e. \end{aligned}$$

With this result, we can calculate the following:

- (i) An upper bound on the total size of the packets *in the system* that have crossed less than i edges:

Knowing that for a concrete edge e , $Q_t^i(e) \leq k_{i-1} + b - \varepsilon B_e(t - t')$, and that $R_t(e)$ bits are either crossing e or in its reception buffer at time t , we can conclude that the size of the packets in the system that have crossed less than i edges at any time t is at most

$$\begin{aligned} \sum_{e \in E(\mathcal{G})} (Q_t^i(e) + R_t(e)) &\leq mk_{i-1} + mb + \sum_{e \in E(\mathcal{G})} (R_t(e) - \varepsilon(t - t')B_e) \\ &\leq mk_{i-1} + mb + \sum_{e \in E(\mathcal{G})} R_{\max}(e) \\ &= k_i, \end{aligned}$$

and hence the inductive step holds.

This trivially shows the first claim of the theorem, since no more than k_d bits belong to packets that have traversed less than d edges, which are all the packets in the system.

- (ii) An upper bound on the size of the queue of any edge e :

The maximum queue size of any edge e can be calculated when considering $i = d$, since no packet in the queue has crossed d edges. Then, independently of the time instant, $Q_{\max}(e) = \max_t Q_t^d(e) \leq k_{d-1} + b$.

- (iii) An upper bound on $t - t'$:

Since for any concrete edge e , $Q_t^i(e) \geq 0$, we obtain that

$$t - t' \leq \frac{k_{i-1} + b}{\varepsilon B_e} \leq \frac{k_{i-1} + b}{\varepsilon B_{\min}},$$

and so the maximum amount of time that a packet p with path e_1, e_2, \dots, e_d spends in the system is bounded by

$$\sum_{i=1}^d \left(\frac{k_{i-1} + b}{\varepsilon B_{e_i}} + \frac{L_p}{B_{e_i}} + P_{e_i} \right) \leq \frac{d(b + \varepsilon L_{\max}) + \sum_{i=1}^{d-1} k_i}{\varepsilon B_{\min}} + dP_{\max}.$$

■

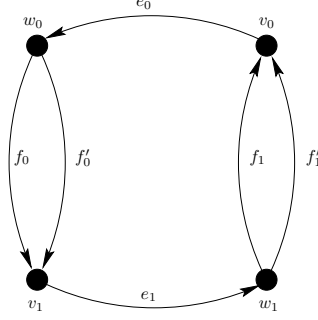


Fig. 2. Baseball network \mathcal{G}_B presented in [2].

6 Instability of queueing policies

In this section we introduce some new protocols that base their policies in the main features of the CAQT model, namely, the length of the packets, the edge bandwidths and the edge propagation delays. We show that the CAQT model is strictly stronger than the AQT model by presenting scheduling policies that are unstable under the former while they are universally stable under the latter.

6.1 Instability by difference in packet length

Consider the LPL (*longest-packet-length*) protocol which gives priority to the packet with longest length. Let us denote as LPL-LIS the same protocol when ties are broken according to the LIS policy. Note that LPL-LIS is universally stable under the AQT model, since in this model all packets have the same length and hence the policy simply becomes LIS [2]. However, we show here that LPL-LIS is unstable in an extension of AQT with multiple packet lengths just by considering two different packet lengths (1 and 2). For simplicity we will assume that time advances in synchronous steps (as in AQT). Packets of length 2 take 2 steps to cross each link. In the LPL-LIS protocol, these double packets will have priority over the single packets. Note that this model is trivially included in CAQT. To show the instability of the LPL-LIS protocol, we use the baseball network presented in [2] (see Figure 2).

Theorem 9. *Let \mathcal{G}_B be the graph with nodes $V(\mathcal{G}_B) = \{v_0, v_1, w_0, w_1\}$, and edges $E(\mathcal{G}_B) = \{(v_0, w_0), (v_1, w_1), (w_1, v_0), (w_1, v_1), (w_0, v_1), (w_0, v_0)\}$. All the edges in $E(\mathcal{G}_B)$ have bandwidth 1 and null propagation delay. For $r > 1/\sqrt{2}$ there is an (r, b) -adversary \mathcal{A} that makes the system $(\mathcal{G}_B, \text{LPL-LIS}, \mathcal{A})$ to be unstable only with packets of length 1 and 2.*

Proof: We assume an initial configuration in which there are s_0 packets of length 1, distributed roughly evenly among w_1 and v_1 , and all of them trying to cross e_0 , for a large enough s_0 . This is the base case of the induction. Then, we show

that at the end of phase j there are at least $s_0 + j$ packets of length 1 distributed evenly among w_{1-i} and v_{1-i} that want to cross edge e_i , where $i = j \bmod 2$. This clearly holds for phase 0.

Then, we consider phase j . Without loss of generality we assume j to be odd. By induction hypothesis at the beginning of phase j there is a set S of s packets of length 1 that want to cross e_0 in w_1 and v_1 . The sequence of injections is divided into subphases as follows.

- (i) We let one step go to allow the first packet in S to reach the queue of e_0 (note that after then all the packets will follow without interruption). Then, for the next s steps, we inject a set S_1 of rs packets of length 1 that want to traverse edges $e_0 f'_0 e_1$. These are blocked by the packets in S , since they all have the same length, but those in S are older.
- (ii) Then, for the next rs steps we inject a set of $r^2 s$ packets of length 1 that want to traverse edges $e_0 f'_0 e_1$. These are blocked by the packets in S_1 . We also delay the flow of packets in S_1 through f'_0 using single-edge injections of packets of length 2. The new packets block the packets in S_1 , since they are longer. Roughly $r^2 s/2$ packets of length 2 can be injected, and hence roughly $rs - 2r^2 s/2$ packets of S_1 get to cross f'_0 . Then, at the end of this sub-phase there are still $r^2 s$ packets of S_1 in w_0 .

This completes phase j . Clearly, at the end of it there are $r^2 s$ packets in w_0 and $r^2 s$ packets in v_0 , all of length 1 and trying to traverse e_1 . Since $2r^2 s > s$ for large enough s , the induction hypothesis holds. ■

6.2 Instability by difference in bandwidth

Consider the SPL (*slowest-previous-link*) protocol which gives priority to the packet whose last crossed link was the slowest, i.e., had the smallest bandwidth. This policy aims to equilibrate the lost in transmission velocity suffered in previous links. Let us denote as SPL-NFS this protocol, when ties are broken according to the NFS protocol. Observe that the SPL-NFS protocol is equivalent to NFS in the AQT model and thus universally stable [2]. However, we show that in a similar way as shown for the LPL-LIS protocol, the SPL-NFS protocol can be made unstable in the CAQT model.

Theorem 10. *Let \mathcal{G}_B be the graph with nodes $V(\mathcal{G}_B) = \{v_0, v_1, w_0, w_1\}$ and edges $E(\mathcal{G}_B) = \{(v_0, w_0), (v_1, w_1), (w_1, v_0), (w_0, v_1), (w_0, v_1), (w_0, v_1)\}$. Let \mathcal{G} be the graph obtained from \mathcal{G}_B whose set of nodes is $V(\mathcal{G}) = V(\mathcal{G}_B) \cup \{v'_0, v'_1, w'_0, w'_1\}$, and whose set of edges is $E(\mathcal{G}) = E(\mathcal{G}_B) \cup \{(v'_0, v_0), (v'_1, v_1), (w'_0, w_0), (w'_1, w_1)\}$. Those edges incident to v_0 and v_1 have bandwidth 2, while the rest have bandwidth 1. All the edges have null propagation delays. For $r > 1/\sqrt{2}$ there is an (r, b) -adversary \mathcal{A} that makes the system $(\mathcal{G}, \text{SPL-NFS}, \mathcal{A})$ to be unstable.*

Proof: The proof is similar to the proof of Theorem 9 above but the injections are done only at the queues of the new 4 edges. Using the same induction

hypothesis and sequence of injections we have that in the first sub-phase the packets in S still block those in S_1 because the latest edge crossed by the latter is faster, while in the second sub-phase the single injections (now injections of path length 2) block the packets in S_1 by NFS. ■

6.3 Instability by difference in propagation delays

Consider the SPP (*smallest-previous-propagation*) protocol which gives priority to the packet whose previously traversed edge had smallest propagation delay, and combine it with NFS to break ties. Let us denote this protocol as SPP-NFS. Observe that the SPP-NFS protocol is equivalent to NFS in the AQT model and thus universally stable [2]. However, we show with the SPP-NFS protocol as example, that just the fact of considering propagation delays can make a policy unstable in CAQT.

Theorem 11. *Let \mathcal{G}_B be the graph with nodes $V(\mathcal{G}_B) = \{v_0, v_1, w_0, w_1\}$ and edges $E(\mathcal{G}_B) = \{(v_0, w_0), (v_1, w_1), (w_1, v_0), (w_0, v_1), (w_0, v_1), (w_0, v_1)\}$. Let \mathcal{G} be the graph obtained from \mathcal{G}_B whose set of nodes is $V(\mathcal{G}) = V(\mathcal{G}_B) \cup \{v'_0, v'_1, w'_0, w'_1\}$, and whose set of edges is $E(\mathcal{G}) = E(\mathcal{G}_B) \cup \{(v'_0, v_0), (v'_1, v_1), (w'_0, w_0), (w'_1, w_1)\}$. Those edges incident to v_0 and v_1 have propagation delay 1, while the rest have null propagation delay. All the edges have unary bandwidth. For $r > 1/\sqrt{2}$ there is an (r, b) -adversary \mathcal{A} that makes the system $(\mathcal{G}, \text{SPP-NFS}, \mathcal{A})$ to be unstable.*

Proof: The proof is similar to the proof of Theorem 9 above but the injections are done only at the queues of the new 4 edges. Packets in S block those in S_1 because the latter crossed an edge with larger propagation delay. Then, the single (two-edge in this case) injections block the packets in S_1 by NFS. ■

7 Conclusions and open questions

We consider a networking scenario in which packets can have arbitrary lengths, and the network links may have different speeds and propagation delays. Taking into account these features, we have presented a generalization of the well-known Adversarial Queueing Theory (AQT) model which does not assume anymore synchronicity in the evolution of the system, and makes it more appropriate for more realistic continuous scenarios. We called it the CAQT model.

We have shown that, in the CAQT model having bounded queues is equivalent to having bounded packet end-to-end delays. From the network point of view, we show that networks with a directed acyclic topologies are universally stable even when the traffic pattern fully loads the links. From the protocol point of view, we have also shown that the well-known LIS, SIS, FTG and NFS protocols remain universally stable in the CAQT model. New protocols have also been proposed which are universally stable in the AQT model but unstable in the CAQT model.

Many interesting questions remain still open in the CAQT model. More results are needed concerning the stability of networks, starting from simple topologies like the ring, to finally tackle the universal stability of networks. It would be of interest to know the queue sizes to be expected (as it was studied in [2,13] for the AQT model), as well as which conditions guarantee that all the packets are actually delivered to destination (as it was studied in [14] for AQT).

8 Acknowledgments

The authors would like to thank the unknown referees from the 30th International Symposium on Mathematical Foundations of Computer Science (MFCS'05) for their comments and suggestions.

References

1. Borodin, A., Kleinberg, J., Raghavan, P., Sudan, M., Williamson, D.: Adversarial queueing theory. *Journal of the ACM* **48** (2001) 13–38
2. Andrews, M., Awerbuch, B., Fernández, A., Kleinberg, J., Leighton, T., Liu, Z.: Universal stability results for greedy contention-resolution protocols. *Journal of the ACM* **48** (2001) 39–69
3. Awerbuch, B., Berenbrink, P., Brinkmann, A., Scheideler, C.: Simple routing strategies for adversarial systems. In: 42th. IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press (2001) 158–167
4. Anshelevich, E., Kempe, D., Kleinberg, J.: Stability of load balancing algorithms in dynamic adversarial systems. In: 34th. Annual ACM Symposium on Theory of Computing, ACM Press (2002) 399–406
5. Álvarez, C., Blesa, M., Díaz, J., Fernández, A., Serna, M.: Adversarial models for priority-based networks. *Networks* **45** (2005) 23–35
6. Álvarez, C., Blesa, M., Serna, M.: The impact of failure management on the stability of communication networks. In: 10th International Conference on Parallel and Distributed Systems, IEEE Computer Society Press (2004) 153–160
7. Borodin, A., Ostrovsky, R., Rabani, Y.: Stability preserving transformations: Packet routing networks with edge capacities and speeds. *Journal of Interconnection Networks* **5** (2004) 1–12
8. Koukopoulos, D., Mavronicolas, M., Spirakis, P.: Instability of networks with quasi-static link capacities. In: 10th International Colloquium on Structural Information Complexity. Volume 17 of Proceedings in Informatics., Carleton Scientific (2003) 179–194
9. Koukopoulos, D., Mavronicolas, M., Spirakis, P.: Performance and stability bounds for dynamic networks. In: 7th International Conference on Parallel Architectures, Algorithms and Networks, IEEE Computer Society Press (2004) 239–246
10. Echagüe, J., Cholvi, V., Fernández, A.: Universal stability results for low rate adversaries in packet switched networks. *IEEE Communication Letters* **7** (2003) 578–580
11. Bhattacharjee, R., Goel, A., Lotker, Z.: Instability of FIFO at arbitrarily low rates in the adversarial queueing model. *SIAM Journal on Computing* **34** (2004) 318–332

12. Cruz, R.: A calculus for network delay. Part I (network elements in isolation) and II (network analysis). *IEEE Transactions on Information Theory* **37** (1991) 114–141
13. Weinard, M.: The necessity of timekeeping in adversarial queueing. In: 4th International Workshop on Efficient and Experimental Algorithms. Volume 3503 of *Lecture Notes in Computer Science.*, Springer-Verlag (2005) 440–451
14. Rosén, A., Tsirkin, M.: On delivery times in packet networks under adversarial traffic. In: 16th ACM Symposium on Parallel Algorithms and Architectures, ACM Press (2004) 1–10