Instability Phenomena in Underloaded Packet Networks with QoS Schedulers

M.Ajmone Marsan, M.Franceschinis, E.Leonardi, F.Neri, A.Tarello

Abstract— **Instability in packet-switching networks is normally associated with overload conditions, since queueing network models show that, in simple configurations, only overload generates instability. However, some results showing that instability can happen also in underloaded queueing networks appeared in the recent literature. Underload instabilities can be produced by complex scheduling algorithms, that bear significant resemblance to the Quality of Service (QoS) schedulers considered today for packet networks. In this paper, we study with fluid models and with adversarial queueing theory possible underload instabilities due to strict-priority schedulers and to Generalized Processor Sharing (GPS) schedulers.**

I. Introduction

Simplicity has been for a long time a characteristic of packet schedulers within IP routers; indeed, packets arriving at a router, after being checked for correctness and routed, were normally inserted into simple FIFO queues, awaiting their turn for transmission. The desire for high performance router architectures and Quality of Service (QoS) guarantees is now changing this picture, specially for what concerns packet scheduling algorithms. Priority queues and/or Generalized Processor Sharing (GPS) queues [1], [2] are now widely accepted as useful tools for the differentiation of the treatment of packet flows with different QoS requirements [3], [4]. Input Queued (IQ) switches with Virtual Output Queueing (VOQ) are based on complex heuristic schedulers, that try to approximate Maximum Weight Matching (MWM) schedulers with acceptable complexity.

In spite of these significant increases in the complexity of scheduling algorithms, when it comes to performance evaluation issues, and to considerations about possible instabilities due to the level of traffic at the router queues, the common belief is that only overload generates instability, while underloaded queues may induce delays longer than desired, but always remain stable. This general wisdom goes back to the models of packet-switching networks originally developed by Kleinrock [5], and based on Jackson queueing networks [6]. Stability results for more general classes of queueing networks, such as BCMP networks [7] and Kelly networks [8], also confirmed the general result that only overload generates instability, and the same effect was due to the insensitivity in GI/GI/1 queues of average queue sizes to the scheduling discipline for all work-conserving schedulers that treat customers independently of their service time.

The first results showing that instability can happen also in underloaded queueing networks[1] started to appear less than a decade ago [9], [10], when some classes of queueing networks were identified, for which underload does not automatically guarantee stability, i.e., for which the backlog at some queue in the network can indefinitely grow also when such queue is not overloaded. It is important to observe that these underload instabilities are often produced either by customer routes that visit several times the same queues, or by variations of the customer service times at the different queues, or by complex scheduling algorithms.

The first hints to the possible connections between underload instabilities in queueing networks and unstable behaviors in packet-switching networks have appeared very recently in [11], [12], where it was shown that underload instability phenomena can arise in networks of IQ switches with MWM schedulers even in the case of acyclic packet routes, and service times that vary only according to channel capacities.

In [13] it was shown that unstable behaviors can be observed at underload for a class of open queueing networks called *re-entrant lines*, where customers visit several times the same queue, and all stations serve incoming customers according to a strict-priority scheduling.

In this paper we discuss the possible underload instabilities due to QoS schedulers in packet-switching networks, focusing both on strict priority schedulers and on GPS schedulers. The considered scenarios always refer to the case of acyclic packet routes, and consider customer service times that vary only according to channel capacities. With these assumptions, the considered scenarios bear a significant resemblance to the approaches being currently considered for QoS provisioning in the Internet, specially DiffServ [3], [4].

In particular, when considering GPS schedulers, we examine both the case of exact matching of the actual packet rates to the GPS rates, and the more realistic case of inaccurate estimation of the actual packet rates. While the stability of any queueing network with GPS schedulers was proved for the case in which the actual packet rates do not exceed the GPS rates [2], [13], in this paper we prove that a queueing network with GPS schedulers may be unstable when some of the actual packets rates exceed the GPS rates, thus confirming

This work was supported by the MIUR PLANET-IP Project. The authors are with: Dipartimento di Elettronica, Politecnico di Torino, Italy; {surname}@mail.tlc.polito.it

[1] We say that a queueing network is underloaded when the traffic loading each queue is less than the service capacity at the same queue. Traffic at a queue is the product of the average customer arrival rate times the average customer service time.

the conjecture formulated in [2][2].

Instead, when considering strict priority schedulers, we prove that all queueing networks are stable, provided that the priority ordering of packet flows does not change from a router (queue) to another (that is, if flow 1 has higher priority than flow 2 at a router, the same happens also at all other routers). Note that, on the contrary, when the priority ordering of packet flows can change from a router to another, instability may arise, as proved in [10].

Finally, we prove that the combination of priority and FIFO schedulers (for example due to packet flows that have identical priority at some routers) may lead to a weak form of instability for loads larger than 0.8. The fact that the combination of priority and FIFO may lead to instability is not surprising, due to the known possible instability of FIFO for loads above 0.85 [14]. The fact that the instability region expands to lower load values is instead quite remarkable.

II. Definitions and Notations

We consider an acyclic network of J discrete-time queues represented by row vector Q, whose j-th component, $q^{(j)}, 1 \leq j \leq J$, is a descriptor associated with the j-th queue. However, all results in this paper apply also to networks of continuous-time queues.

The network of queues handles F customer flows. Customers belonging to flow f, $1 \leq f \leq F$, arrive at the network from outside, receive service at a number of queues, and leave the network. In most applications, the routing of customers belonging to flow f is deterministic; however, more general acyclic routings are allowed by our model.

Customers belonging to the same flow, and stored at the same physical queue, form a *virtual queue*. The number of virtual queues in a network is denoted by N. A physical queue is also called a station. Of course, $N \leq FJ$, since routing is acyclic, and, at most, all customer flows visit all stations. In addition, $N \geq \max(F, J)$, since each flow must visit at least one station, and any station must be visited by at least one flow (otherwise the station can be removed from the network). Let $L(k) = j$ be the system location function that associates virtual queue k with the physical queue j at which customers are enqueued. $L^{-1}(j)$ is the counter-image of j through function $L(k)$. In general, $L^{-1}(j)$ returns a set of virtual queues of cardinality $|L^{-1}(j)|$; however, when $N = J$, each virtual queue is in one-to-one correspondence with a physical queue.

Let $X_n = (x_n^{(1)}, x_n^{(2)}, \ldots, x_n^{(N)})$ be the row vector whose k-th component $x_n^{(k)}$, $1 \leq k \leq N$, represents the number of customers at virtual queue k in the system at time n. We assume in the paper that X_0 has null components. We suppose that the service time required by customer i of virtual queue k is a random variable $S_i^{(k)}$ distributed according to a discrete-time general distribution with average $s^{(k)}$; service times distributions are assumed to be bounded. We consider only non-preemptive atomic service policies, i.e., policies that serve customers in an atomic fashion, with no service interruption. However, all our results also hold for work-conserving preemptive service policies.

The evolution of the number of customers at virtual queue k is described by $x_{n+1}^{(k)} = x_n^{(k)} + e_n^{(k)} - d_n^{(k)}$, where $e_n^{(k)}$ represents the number of customers that entered virtual queue k (and thus physical queue $L(k)$) in time interval $(n, n{+}1]$, and $d_n^{(k)}$ represents the number of customers departed from virtual queue k in time interval $(n, n{+}1]$. $E_n = (e_n^{(1)}, e_n^{(2)}, \ldots, e_n^{(N)})$ is the vector of entrances in virtual queues, and $D_n = (d_n^{(1)}, d_n^{(2)}, \ldots, d_n^{(N)})$ is the vector of departures from virtual queues. With this notation, the system evolution equation can be written as:

$$X_{n+1} = X_n + E_n - D_n \tag{1}$$

The entrance vector E_n is sum of two terms: vector $A_n = (a_n^{(1)}, a_n^{(2)}, \ldots, a_n^{(N)})$, representing the customers arrived at the system from outside, and vector $R_n = (r_n^{(1)}, r_n^{(2)}, \ldots, r_n^{(N)})$ of recirculating customers; $r_n^{(k)}$ is the number of customers departed from some virtual queue and entered into virtual queue k in time interval $(n, n+1]$. Note that when customers do not traverse more than one queue (as it is the case for a switch in isolation), vector R_n is null for all n, and $E_n = A_n$.

The $N \times N$ matrix $P_n = [p_n^{(k,l)}]$ is the *routing matrix*, whose element $p_n^{(k,l)}$ represents the fraction of customers departing from virtual queue k in time interval $(n, n+1]$ that enter virtual queue l. Note that, since routing is acyclic, P_n can be put in (upper or lower) triangular form by reordering rows and columns. Since in most applications the routing of each flow is deterministic, P_n belongs to the class of deterministic doubly sub-stochastic matrices, i.e., matrices whose elements take values in the set $\{0, 1\}$ and whose lines (both rows and columns) comprise elements whose sums do not exceed 1. Thus, throughout this paper we suppose that $P_n = P$ is deterministic, even if our models and most of the results can be easily extended to more general contexts[3].

By noting that $R_n = D_n P$, the evolution of virtual queues can be rewritten as:

$$X_{n+1} = X_n + A_n - D_n(I - P) \tag{2}$$

where I denotes the identity matrix.

Finally, let us introduce the following norm definition:

Definition 1: Given a vector $Z \in \mathbb{R}^N$, $Z = (z^{(k)},\ 1 \leq k \leq N)$, and a location function $L(k) = j$, from $1 \leq k \leq N$ to $1 \leq j \leq J$, with $J \leq N$, norm $||Z||_L$ is defined as:

$$||Z||_L = \max_{j=1,\ldots,J} \left\{ \sum_{k \in L^{-1}(j)} |z^{(k)}| \right\} \tag{3}$$

[2] Only a particular sub-class of networks of queues implementing GPS schedulers was proved in [2] to be always stable in underload conditions, even when some actual packet rates exceed the GPS rates.

[3] It is possible to apply the fluid model methodology to all systems of queues forming an *open network*. In such a case, if $E[X]$ denotes the expectation of random quantity X and I denotes the identity matrix, $\Delta = I + E[P_n] + E[P_n]^2 + E[P_n]^3 + \ldots = (I - E[P_n])^{-1}$ exists and is finite, i.e., $I - E[P_n]$ is invertible for all n. It is, however, necessary to further assume that the routing matrix P_n satisfies the strong law of large numbers: $\lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} P_i = P$ with probability 1.

A. Traffic and System Stability Definitions

Definition 2: The external arrival processes $A_n = (a_n^{(1)}, a_n^{(2)}, \dots, a_n^{(N)})$ are stationary if $E[A_n] = \Lambda = (\lambda^{(1)}, \lambda^{(2)}, \dots, \lambda^{(N)})$, i.e., if $E[A_n]$ does not depend on time interval $[n, n+1)$.

Let $\Pi = \Lambda(I-P)^{-1}$ be a vector whose k-th component $\pi^{(k)}$ represents the average effective arrival rate at virtual queue k. Let $W_n = (w_n^{(1)}, w_n^{(2)}, \dots, w_n^{(N)})$ be the workload provided at each virtual queue by customers that entered the network of queues in time interval $[n, n+1)$, i.e., the global amount of work required at each virtual queue by customers entering the network in time unit $[n, n+1)$. The average workload provided at each virtual queue by customers that entered the system of queues in time interval $[n, n+1)$ is given by: $\Omega = E[W_n] = (\omega^{(1)}, \omega^{(2)}, \dots, \omega^{(N)})$, with $\omega^{(k)} = \pi^{(k)} s^{(k)}$, being $S = (s^{(1)}, s^{(2)}, \dots, s^{(N)})$ the vector of the average service times at virtual queues.

Definition 3: A stationary traffic pattern is admissible if $||\Omega||_L < 1$

Definition 4: A stationary arrival process A_n at virtual queues satisfies the strong law of large numbers if[4]: $\lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} A_i = \Lambda \qquad \text{w.p. } 1$

Definition 5: A system of queues is *rate-stable* if:

$$\lim_{n\to\infty} \frac{X_n}{n} = \lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n} (E_i - D_i) = 0 \qquad \text{w.p. } 1$$

Definition 6: A system of queues *achieves 100% throughput* if it is rate-stable under any admissible traffic pattern satisfying the strong law of large numbers.

Definition 7: A system of queues is *stable* if, for every $\epsilon > 0$, there exists an $M > 0$, such that for every n:

$$\mathrm{P}(X_n > M) < \epsilon$$

Note that system stability implies the tightness of the family of probability measures describing the system evolution.

Definition 8: A traffic pattern is $[\rho, \phi]$ regulated if, for every window $\mathcal{W}$ comprising ϕ consecutive time units, the total workload provided by customers entering the network in $\mathcal{W}$ satisfies:

$$|| \sum_{n\in\mathcal{W}} W_n ||_L \leq \rho\phi$$

Note that any $[\rho, \phi]$ regulated traffic pattern with $\rho < 1$ and finite ϕ falls in the class of (ρ_0, σ) leaky-bucket constrained traffic patterns[5], with $\rho_0 = \rho$ and $\sigma = \rho\phi$. In addition, any (ρ_0, σ) leaky-bucket constrained traffic pattern, with $\rho_0 < 1$ and finite σ, falls within the class of $[\rho, \phi]$ regulated traffic patterns for $\phi \geq \phi_{\min}$. To see this, it is enough to choose ρ such that $\rho_0 < \rho < 1$ and then to consider any $\phi \geq \phi_{\min} = \frac{\sigma}{\rho - \rho_0}$. Thus, the class of all $[\rho, \phi]$ regulated traffic patterns with $\rho < 1$ and finite ϕ, and the class of all (ρ_0, σ) leaky-bucket constrained traffic patterns with $\rho_0 < 1$ and finite σ are coincident.

Definition 9: A system of queues is said to be *universally stable* if, under any $[\rho, \phi]$ regulated traffic pattern, with $\rho < 1$ and any $\phi < \infty$, the number of customers in the system remains finite.

Note that universal stability implies that delays in the queueing network are bounded under any leaky-bucket constrained traffic with finite burstiness.

The above stability definitions have different strengths, and different implications in the application domain of packet-switching networks. The weaker stability Definitions are 5 and 6; they cannot guarantee the boundedness of queue lengths, and are thus of lesser interest. Definition 7 is the standard definition for stability in queueing systems. Definition 9 is instead the stability definition deriving from adversarial queueing theory, and is the strongest of our stability conditions. Stability under Definition 9 implies all other forms of stability, as shown in [15]; instability under Definition 7 implies instability under Definition 9.

III. Analytical Techniques

A. Previous Work

Two existing analytical tools were mainly applied in stability studies: the stochastic Lyapunov function methodology [16], and the fluid limit theory [13]. Both tools, and the latter in particular, can be applied to general networks of queues, under some mild regularity conditions on the stochastic processes describing the system. Using these tools, general necessary and sufficient conditions for the stability of networks of queues were identified, in the case of networks with two stations that operate under a strict priority service discipline. In addition, it was also possible to prove that a large class of underloaded networks of FIFO queues[6] [17], as well as networks of GPS-queues [13], where GPS rates are exactly matched to packet flow rates, are stable (according to Definitions 5 and 7).

As an alternative to the two cited stochastic techniques, a new analytical framework, called Adversarial Queueing Theory (AQT) [14], [18], was developed.

Applying AQT, several interesting results were obtained. Common service disciplines, such as FIFO, LIFO, and Nearest-To-Go (NTG) (where priority is given to packets nearest to their destination), were proved not to be universally stable in all networks of queues. Instead, other service disciplines, such as Longest-In-System (LIS) (where priority is given to packets first injected in the system), Shortest-In-System (SIS) and Furthest-To-Go (FTG), were proved to be universally stable in all networks of queues [14].

[4] The notation "w.p. 1" means " with probability 1".

[5] A (ρ_0, σ) leaky-bucket constrained traffic pattern is a traffic pattern such that in any window of t consecutive time units, the amount of workload arriving at any network server does not exceed $\rho_0 t + \sigma$.

[6] The result applies only to a restricted class of FIFO queueing networks: the class of generalized Kelly-type networks. In more general FIFO networks it was shown that some underload instability may arise. However, since FIFO queueing networks modeling communication networks always fall in the class of generalized Kelly-type networks, we only consider such networks in this paper.

It is important to observe that, as proved in [15], universal stability implies rate-stability and stability of the queueing network (the opposite is, of course, not true).

Finally, we notice that an important stability result for underloaded GPS networks was obtained in [2] by applying Network Calculus [19], [20] concepts. In [2], any underloaded network of queues implementing GPS schedulers was proved to be stable when fed with leaky-bucket constrained traffic flows, whenever the GPS rates assignment to flows satisfies the Consistent Relative Session Treatment (CRST) constraint. The CRST constraint implies that, for any pair of flows f_i and f_j whose paths are not disjoint, the ratios between GPS weights and packet actual rates are in the same order relation at every shared node.

B. Adversarial Queueing Theory

Adversarial queueing theory [14], [18] follows a deterministic approach to define stability criteria for queueing networks, in the sense that packet generation at the network inputs, packet service times, and packet routing, are supposed to be deterministic.

In AQT, a queueing network is represented by a directed graph, where edges represent queues, and nodes represent routing points. Packets originate at a node, follow some path consisting of a set of consecutive edges and nodes, and then leave the network at another node. Different paths may share some edges.

Time is assumed to proceed by discrete steps, named time units. When two or more packets following different paths need to traverse the same edge, only one can be chosen in a time unit, while the others are forced to wait in the node. The decision about which packet has to be served in case of contention defines the service discipline at the nodes. Only work-conserving service disciplines are considered, i.e., service disciplines according to which a packet must be served at a node whenever some packets are waiting for service.

The basic idea of AQT is the following. At each time unit, an adversary injects in the network a set of packets, each one characterized by a particular path, which is defined when the packet is generated, and cannot be changed. Of course, the network can be flooded with packets, if no restriction is set on packet generation processes. Therefore, for any $0 \leq \rho < 1$, a (ρ, ϕ) *adversary* injects a $[\rho, \phi]$ regulated traffic pattern into the network. The aim of a (ρ, ϕ) adversary is trying to find a deterministic traffic pattern under which the length of some queue grows toward infinity, and the network becomes unstable. More precisely, let G identify a network (a graph), S be a service discipline, and A an adversary: system (G, S) is said to be *universally stable* if, for every initial condition on G, and for every adversary A, a constant integer M exists, such that any queue length is upper bounded by M. In this case the queueing network G is universally stable under service policy S.

A queueing network G is said to be *universally stable* if it is universally stable under any service discipline S and, similarly, a scheduling policy S is said to be *universally stable* if it is stable for all networks G.

It is worth noting that the queueing network stability criterion provided by AQT is very tight, because it derives from a worst-case analysis under a wide class of deterministic arrival patterns. In this sense, AQT is somewhat similar to Network Calculus [19], [20].

C. Fluid Models

The evolution of a queueing network is traditionally described by vector equation (2), where D_n is a function of both X_n and the scheduling policy. An alternative expression of the evolution of the queue lengths vector can be provided in terms of the cumulative processes. Let

- $\mathcal{A}(n)$ be the cumulative number of arrivals in time interval $[0, n)$, i.e., $\mathcal{A}(n) = \sum_{i=0}^{n} A_i$;
- $\mathcal{D}(n)$ be the cumulative number of departures in time interval $[0, n)$, i.e., $\mathcal{D}(n) = \sum_{i=0}^{n} D_i$;
- $\mathcal{T}(n)$ be the cumulative amount of work provided to virtual queues in time interval $[0, n)$ whose k-th component represents the amount of work provided to virtual queue k.

With these definitions, the equation describing the queue lengths vector evolution becomes:

$$X_{n+1} = X_0 + \mathcal{A}(n) - \mathcal{D}(n)[I - P] \tag{4}$$

where $\mathcal{D}(n)$ can be expressed as a stochastic function of $\mathcal{T}(n)$.

By linearly interpolating X_n, $\mathcal{A}(n)$, $\mathcal{D}(n)$ and $\mathcal{T}(n)$ we can define their extensions to continuous time: $\hat{X}(t)$, $\hat{\mathcal{A}}(t)$, $\hat{\mathcal{D}}(t)$ and $\hat{\mathcal{T}}(t)$. We first focus our attention on the behavior of:

$$\lim_{m \to \infty} \frac{1}{m} \hat{X}(mt) = X(t)$$

which is called the *fluid limit* of the queue length vector (this limit exists under weak regularity conditions). Note that $X(0) = \lim_{m \to \infty} \frac{1}{m} \hat{X}(0) = 0$ whenever a finite number of customers is in the system at time 0 (as we assume throughout the paper).

It was shown in [13] that the fluid limit of the queue length vector exists with probability 1 if $\mathcal{A}(n)$ and $\mathcal{P}(n)$ satisfy the strong law of large numbers, i.e., $\lim_{m \to \infty} \frac{\mathcal{A}(mn)}{m} = \Lambda n$, and $\lim_{m \to \infty} \frac{\sum_{i=1}^{m} S_i^{(k)}}{m} = s^{(k)}$.

In addition, the fluid limit is a continuous function which is derivable almost everywhere (i.e., it is derivable in all points of $t \in \mathbb{R}^+$, except for at most a set of null Lebesgue measure), with probability 1.

In order to study the behavior of $X(t)$, the following theorem [13] is fundamental.

Theorem 1: The fluid limit $X(t)$ is a solution of the fluid model differential equations, obtained by averaging the stochastic equations of the system evolution:

$$X(t) = X(0) + \Lambda t - D(t)(I - P) \tag{5}$$

and:

$$D(t) = T(t)\Gamma \tag{6}$$

where Γ is a diagonal matrix whose non-null elements $\gamma^{(k)}$ are equal to the inverse of the average service times at the different virtual queues, and $T(t) = (t^{(1)}(t), t^{(2)}(t), \dots, t^{(N)}(t))$ is a non-decreasing function describing the time spent serving the different virtual queues in time interval $[0, t)$, which must satisfy additional constraints that identify the service discipline at queues.

For example, if the service discipline at queue j is work-conserving, then $T(t)$ satisfies the following equation:

$$\int_0^\infty \left[\sum_{k \in L^{-1}(j)} x^{(k)}(t) \right] d \left[t - \sum_{k \in L^{-1}(j)} t^{(k)}(t) \right] = 0 \quad (7)$$

i.e., the server is working whenever there is some unit of work to be performed. Indeed, inside the integral we have the sum of virtual queue lengths at station j, and the integral is performed with respect to the difference between time and the work performed at station j; if queues are empty, the integral is zero; if queues are nonempty, the time increments must equal the performed work, so that again the integral becomes zero.

The following results relate the behavior of the solution of the fluid model equations (5) to the behavior of the network of queues.

Theorem 2: If $X(t) = 0 \ \ \forall t > 0$ is the *only* solution of the fluid model equations, under the initial condition $X(0) = 0$, then the system of queues is rate-stable [13]. In this case the fluid model is said to be *weakly stable.*

Theorem 3: Consider a fluid Markov model (i.e., a process with continuous state space, that satisfies the Markov property). If t^* exists, such that for all fluid solutions, $X(t) = 0 \ \ \forall t > t^*$ under any initial condition with $||X(0)||_L = 1$, then the system of queues is Harris recurrent [13] (which is an extension of the concept of recurrence for Markov chains). In this case the fluid model is said to be *stable.*

Theorem 4: If there exists $t > 0$ such that $X(t) > 0$ for *every* solution of the fluid model equations (5), under the initial condition $X(0) = 0$, then the system is unstable (i.e., some queue lengths grow to infinity) with probability 1. In this case the fluid model is said to be *weakly unstable.*

Theorem 5: Consider a fluid Markov model. If an initial condition exists with $||X(0)||_L = 1$, such that, for every solution of the fluid model equations, $\limsup_{t\to\infty} ||X(t)||_L = \infty$, then the queueing system is not Harris recurrent, i.e., it is not stable with probability 1. In this case the fluid model is said to be *unstable.*

Note that for fluid models the definitions of weak instability and instability are not the converse definitions of weak stability and stability, respectively. Nothing can be said about the behavior of the system of queues when the fluid model is neither (weakly) stable nor (weakly) unstable. This is due to the fact that, in general, fluid limits are only a subset of the solutions of the fluid model differential equations. Thus, it is difficult to isolate fluid limits, discarding fluid solutions that do not correspond to fluid limits, and then infer information on the queue system behavior when fluid solutions exhibit different behaviors (i.e., some solutions are null and some are not null).

IV. New Results

In this section we present our stability results for networks of queues. We start by considering in the first subsection networks of queues implementing GPS schedulers, and then we consider priority service disciplines.

By using the fluid model methodology it was previously shown that a network of queues with GPS schedulers achieves 100% throughput under any admissible traffic pattern when GPS weights are equal to or greater than the corresponding effective average rates [13]. This result was further strengthened by showing that networks of queues implementing GPS schedulers are universally stable under the same condition on flow rates. For these reasons GPS schedulers are often considered the optimal architectural solution for communication networks supporting traffic streams with QoS requirements. However, since neither any form of Call Admission Control (CAC), nor any type of ingress traffic shaping is currently implemented in packet-switching networks adopting the Internet protocol suite, it is impossible to obtain a precise knowledge about the average effective flow rates at network routers (and at their schedulers). The calibration of GPS weights at routers (and at GPS schedulers within routers) often relies on local traffic measurements or rough estimates. As a consequence, a mismatch between GPS weights and effective average rates is not only possible, but likely. In the next subsection we discuss an example of a network of queues implementing GPS schedulers in which instability arises (according to Definition 7) under admissible traffic when the nominal flow rates are not exactly matched to the effective average flow rates.

We then focus our attention on networks of queues with strict priority schedulers. We first prove that such networks are universally stable, provided that the priority ordering of packet flows does not change from a router (queue) to another (that is, if at a router flow 1 has higher priority than flow 2, the same happens also at all other routers). Note that, on the contrary, when the priority ordering of packet flows can change from a router to another, rate-instability may arise, as proved in [9], [10], [13]. Finally, we prove that networks of queues implementing a combination of priority and FIFO schedulers (for example due to packet flows that have identical priority at some routers) are rate-stable, but not universally stable for loads larger than 0.8.

A. Networks of GPS queues

Consider the queueing network shown in Figure 1, which comprises four physical queues traversed by four different flows. Each flow enters the network at a physical queue, and follows a simple route[7] that traverses three physical queues. Three virtual queues are co-located at each physical queue, the first storing packets that have just entered the network, the second storing in-transit packets, the last storing packets which are about to leave the network. The $N = 12$ virtual queues are numbered as shown in the figure. Each station of

[7] Routes are said to be ‘simple’ when they are acyclic.

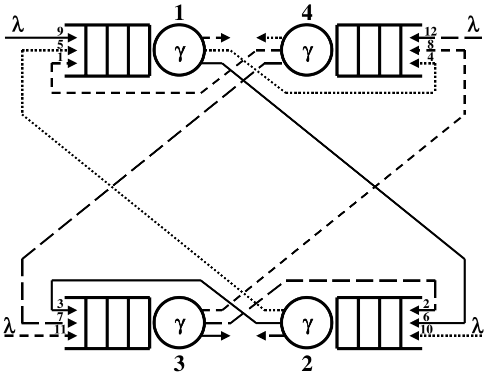

Fig. 1. Queueing network exhibiting instability

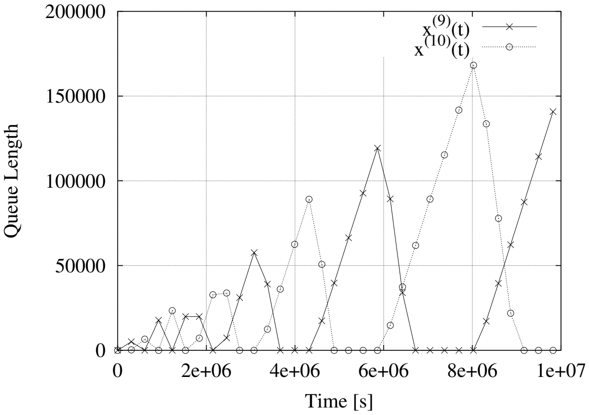

Fig. 2. Evolution of $x^{(9)}$ and $x^{(10)}$ by simulation with GPS weights not matched to flow rates

the queueing network adopts a GPS service discipline. In order to reduce the number of parameters of our model, we assume a perfect symmetry among the nominal flow rates; thus, rate α is assigned at all stations to virtual queues storing packets that are about to leave the system, rate β is assigned to in-transit flows, and rate δ is assigned to flows entering the network.

We assume the server capacity at each station to be 1, so that relation $\alpha + \beta + \delta \leq 1$ must always hold. Figure 2 reports the simulated evolution of the lengths of some of the virtual queues in the network of Figure 1 as a function of time, assuming that all exogenous packet arrival processes (at virtual queues $i = 9, 10, 11, 12$) are Poisson with rate $\lambda^{(i)} = \lambda = \frac{1}{3.25}$ $[s^{-1}]$, that service times are exponentially distributed random variables with average parameter $s^{(i)} = s = \frac{1}{\gamma} = 1$ $[s]$ $\forall i$, and that the GPS weights α, β, δ are respectively equal to 0.6, 0.3 and 0.1. The plots clearly show that the queueing network is unstable, in spite of the fact that no server in the network is overloaded (at each physical queue j, $\sum_{k \in L^{-1}(j)} \omega^{(k)} = \frac{3}{3.25} \approx 0.923$).

This behavior is explained through a fluid model.

Theorem 6: Under admissible input flow rates, the fluid model of the queueing network in Figure 1 can be unstable when the GPS weights α, β and δ are such that $\delta < \alpha$ and $\delta < \beta$. In particular, instability arises when:

$$\begin{cases} \dfrac{\lambda}{\gamma} \geq \dfrac{1-\alpha}{2(1-\alpha+\beta)} & \text{if } \alpha > \beta \\ \dfrac{\lambda}{\gamma} \geq \dfrac{1-\beta}{2(1-\beta+\alpha)} & \text{if } \alpha < \beta \end{cases} \tag{8}$$

Proof: In order to write the system of deterministic differential equations that drive the fluid model associated with the queueing network, we need to introduce some notation. Given any function $f(t)$, let $\mathbf{U}_f$ be the Heaviside step function of $f(t)$, i.e., a function taking value 0 when $f(t) \leq 0$, and equal to 1 when $f(t) > 0$. For each virtual queue k, let k^* be the index of the virtual queue that comes before k in the packet route (if k is the first queue of the route, $k^* = 0$); let $c^{(k)}$ be the GPS weight associated with virtual queue k. We can then write, $\forall k = 1, \ldots, 12$, two equations; the first one says that the change in the length of the size of virtual queue k is obtained as the difference between the service rates at queues k^* and k; the second equation says that the service rate at virtual queue k is either (if the virtual queue is not empty) its GPS rate $c^{(k)}$ scaled by the largest possible coefficient $r_{L(k)}(t)$ accounting for empty virtual queues at the same station, or (if virtual queue k is empty) the minimum between the service rate at virtual queue k^* and the product $c^{(k)} r_{L(k)}(t)$:

$$\begin{cases} \dot{x}^{(k)}(t) = \gamma \left[\dot{t}^{(k^*)}(t) - \dot{t}^{(k)}(t) \right] \\ \dot{t}^{(k)}(t) = \mathbf{U}_{x^{(k)}} c^{(k)} r_{L(k)}(t) + \\ \qquad + (1 - \mathbf{U}_{x^{(k)}}) \min \left[\dot{t}^{(k^*)}(t), c^{(k)} r_{L(k)}(t) \right] \end{cases} \tag{9}$$

Note that $\dot{t}^{(0)}(t)$ must be considered constant with respect to time t, and equal to $\frac{\lambda}{\gamma}$. Note, finally, that $\dot{t}(h)$ depends on the scaling factor $r_j(t)$, $\forall j = 1, 2, 3, 4$, which is defined according to the following expresssion:

$$r_j(t) = \arg \max_{r_j(t) \geq 1} \left\{ \sum_{h \in L^{-1}(j)} \dot{t}^{(h)}(t) \leq 1 \right\} \tag{10}$$

i.e., $r_j(t)$ is the maximum parameter that makes the instantaneous work provided by the server at station j less or equal to 1.

In spite of the apparent complexity of Equation System (9), its solution is rather simple: indeed, the components $x^{(k)}(t)$ of the solution $X(t)$ are piecewise linear, while the components $\dot{t}^{(k)}(t)$ of the solution $\dot{T}(t)$ are piecewise constant; moreover, the discontinuity points correspond to time instants where some queue becomes empty.

We can for example solve (9) in the case $\alpha + \beta + \delta = 1$, $\gamma = 1$ $[s^{-1}]$, $\alpha > \beta$, assuming that at time $t = 0$ the system is in the following initial condition:

$$x^{(k)}(0) = \begin{cases} q > 0 & k = 9, 12 \\ 0 & k \neq 9, 12 \end{cases}$$

The following values of $\dot{t}^{(k)}(t)$ can be computed:

$$\dot{t}^{(k)}(t) = \begin{cases} \dfrac{1-\alpha-\beta}{1-\alpha+\beta} & k = 1, 4, 5, 8, 10, 11 \\ \dfrac{\beta}{1-\alpha+\beta} & k = 2, 3, 6, 7 \\ \dfrac{\alpha+3\beta-1}{1-\alpha+\beta} & k = 9, 12 \end{cases} \tag{11}$$

Indeed, due to symmetry, the values of $\dot{t}^{(k)}(t)$ at stations 1 and 4 must be the same, as well as at stations 2 and 3. We can thus write:

$$\dot{t}^{(11)}(t) = \dot{t}^{(10)}(t) = x \le \lambda$$
$$\dot{t}^{(8)}(t) = \dot{t}^{(5)}(t) = y$$
$$\dot{t}^{(6)}(t) = \dot{t}^{(7)}(t) = z$$

Note that it must be $y \le x$, because virtual queues 5 and 8 are initially empty, and they receive x as input.

We can also write that:

$$\dot{t}^{(1)}(t) = \dot{t}^{(4)}(t) = y$$

because

$$c^{(1)} = c^{(4)} = \alpha > \frac{1}{3} > \lambda$$

Then we can deduce:

$$\dot{t}^{(9)}(t) = \dot{t}^{(12)}(t) = 1 - 2y$$

because the whole service capacity is always used at a station if at least one virtual queue is not empty.

However, since:

$$y \le x \le \lambda \le \frac{1}{3}$$

we get $1 - 2y \ge y$. This, together with $\beta > \delta$, forces $x = y$: indeed, according to (9), the solution $y < x$ is inammissible because it would cause virtual queue 5 and 8 receiving less (or equal) service than virtual queues 9 and 12 while growing up, although their own GPS weight β is strictly higher than δ, characterizing virtual queues 9 and 12. Now, considering stations 2 and 3, we can write:

$$\dot{t}^{(2)}(t) = \dot{t}^{(3)}(t) = z$$

Indeed, it cannot be $\dot{t}^{(2)}(t) > z$, because virtual queue 2 receives z as input, and $x^{(2)} = 0$; and it cannot be $\dot{t}^{(2)}(t) < z$, because $\alpha > \beta$.

To avoid that the service capacity at physical queues 2 and 3 exceeds 1, it must be $z < 1 - 2x$ (otherwise, the service capacity would be at least equal to $2-3x$). As a result, virtual queues 6 and 7 are forced to grow, because the fluid arrival rate is higher than the fluid departure rate. Thus, z and x can be determined as follows. First, we impose $2z + x = 1$, that is:

$$z = \frac{1-x}{2}$$

Then, recalling that $x \le \lambda$ and comparing virtual queues 6 and 10, we say that $x = \lambda$ is acceptable if:

$$\frac{x}{\delta} < \frac{z}{\beta}$$

otherwise, we suppose $x < \lambda$ (this means that also virtual queue 10 is forced to grow), and obtain x by imposing $\frac{x}{\delta} = \frac{z}{\beta}$. We ignore the first alternative, which leads to all empty queues, and consider the case $x < \lambda$, which holds under condition:

$$\lambda > \frac{1-\alpha-\beta}{1-\alpha+\beta}$$

The values reported in (11) can be easily obtained by substituting the value found for x in the previous expressions. Such values are valid until $x^{(9)}(t)$ and $x^{(12)}(t)$ become null. This happens when:

$$t = t_1 = q\left(\frac{\alpha+3\beta-1}{1-\alpha+\beta} - \lambda\right)^{-1}$$

Considering the rates at which the fluid arrives and is drained at the different queues, the length of the queues at time t_1 is:

$$x^{(k)}(t_1) = \begin{cases} q\dfrac{(\lambda - \dfrac{1-\alpha-\beta}{1-\alpha+\beta})}{(\dfrac{\alpha+3\beta-1}{1-\alpha+\beta} - \lambda)} & k = 10, 11 \\ q\dfrac{(\alpha+2\beta-1)}{(1-\alpha+\beta)(\dfrac{\alpha+3\beta-1}{1-\alpha+\beta} - \lambda)} & k = 6, 7 \\ 0 & \text{otherwise} \end{cases}$$

Proceeding as before, it is easy to verify that, in the new set of values for the $\dot{t}^{(k)}(t)$ after t_1, the only variation consists in $\dot{t}^{(9)} = \dot{t}^{(12)} = \lambda$.

The next relevant event is the emptying of virtual queues 6 and 7 at time $t = t_1 + t_2$, where t_2 is given by the ratio of $x^{(6)}(t_1)$ and $x^{(7)}(t_1)$ and the difference between the draining and arrival rates of fluid at those two queues. At time $t = t_1 + t_2$, the length of the queues is:

$$x^{(k)}(t) = \begin{cases} q\dfrac{\lambda(1-\alpha+\beta) - (1-\alpha-\beta)}{\beta - \lambda(1-\alpha+\beta)} & k = 10, 11 \\ 0 & k \ne 10, 11 \end{cases}$$

By observing the evident symmetry of the network configuration that is reached at $t_1 + t_2$ with the initial condition, it is possible to conclude that the quantities $x^{(10)}(t)$ and $x^{(11)}(t)$ (and the same is true for $x^{(9)}(t)$ and $x^{(12)}(t)$) grow to infinity, so that the fluid model is unstable when:

$$\frac{\lambda(1-\alpha+\beta) - (1-\alpha-\beta)}{\beta - \lambda(1-\alpha+\beta)} \ge 1$$

that proves the assert. With a similar procedure it is possible to obtain the condition when $\beta > \alpha$. ■

If the dynamics of the queueing network correspond to a Markov process, the instability of the fluid model implies the instability (according to Definition 7) of the queueing network. Thus, it is possible to conclude that:

Theorem 7: A queueing network with GPS schedulers, whose evolution can be described by a Markov process, may be unstable (according to Definition 7) under admissible traffic when nominal flow rates at the stations are not matched to the effective average flow rates.

Note that Markov processes, i.e., processes satisfying the Markov property, possibly with continuous state space, are very general models, to which almost all queues and network of queues of practical interest can be reduced. For example, the evolution of the unfinished work in a network of GI/GI/m queues can be described by a (complex) Markov process.

Of course, the instability of the fluid model automatically implies that the GPS service discipline is not universally stable

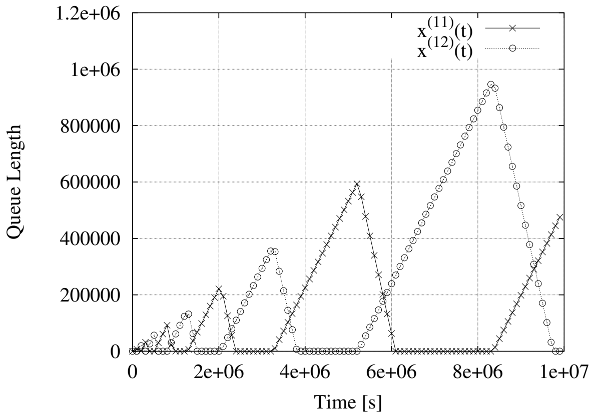

Fig. 3. Evolution of $x^{(11)}(t)$ and $x^{(12)}(t)$ by simulation with strict-priority service

when nominal flow rates are not matched to the effective average flow rates.

B. Networks of Priority Queues

We consider in this section strict-priority/FIFO queueing networks, defined as follows.

Definition 10: A *strict-priority/FIFO queueing network* is a network of queues in which all virtual queues within the same physical queue are associated with a priority level; customers are served according to a strict priority non-preemptive service discipline, i.e., customers are always extracted from one of the non-empty highest-priority virtual queues. When the same priority is assigned to several virtual queues within the same station, ties are broken according to a FIFO service discipline.

We associate each virtual queue k with a priority index $p^{(k)}$ with $1 \leq p^{(k)} \leq |L^{-1}(L(k))|$, in such a way that lower indices correspond to higher priorities. The same priority index is associated with virtual queues with the same priority residing at the same physical queue.

Dai [13] showed that instability phenomena can arise in strict-priority queueing networks, thus reducing the maximum achievable throughput. However, instability was observed in queueing networks where either customers visit several times the same physical queue, or the relative priority order assigned to flows is different in different physical queues. For example, consider again the queueing network depicted in Figure 1, that we have already studied in the case of GPS schedulers. Imagine that a strict-priority service discipline is adopted by servers, and that priority index $p^{(k)}$ is respectively equal to: 1 if virtual queue k is storing packets which are about to leave the network, 2 if it is storing in-transit packets, and 3 if its packets have just entered the network. Simulation results (see Figure 3) under an admissible traffic pattern ($\lambda = \frac{1}{3.25}$ $[s^{-1}]$ and $s = 1$ $[s]$, as in previous subsection) show that the network is unstable. However, in modern multi-service communication networks, packets normally follow simple routes, and priorities are usually assigned to packet flows in such a way that the relative priority order is the same at different nodes. For this reason, we specialize our investigation in this paper to a subclass of strict-priority/FIFO queueing networks, called acyclic strict-priority/FIFO networks.

In order to obtain a precise definition of the class of acyclic strict-priority/FIFO queueing networks, we need to define the Priority Dependency Graph (PDG) induced by a priority assignment.

Definition 11: The *Priority Dependency Graph* is the directed graph $G(V, E)$ satisfying the following properties:

- a vertex $v \in V$ corresponds to each network flow f;
- a directed edge $e \in E$ connects vertices v_m and v_n ($e : v_m \to v_n$), corresponding respectively to flows f_m and f_n, if there exists a physical queue q traversed by flows f_m and f_n, such that $p^{(k_m)} < p^{(k_n)}$, where k_m and k_n are the virtual queues residing at q associated respectively with f_m and f_n.

We say that flow f_m is *father* of flow f_n if the PDG contains an edge from v_m to v_n; in this case f_n is said to be a *child* of f_m.

Now we are ready to formally define the class of acyclic strict-priority/FIFO queueing networks.

Definition 12: An *acyclic strict-priority/FIFO queueing network* is a strict-priority/FIFO queueing network in which

- routes of flows are simple, i.e., all flows can visit at most once any physical queue;
- the PDG is acyclic.

We can now prove a very general and useful result.

Theorem 8: Each acyclic strict-priority/FIFO queueing network is rate-stable under any admissible traffic pattern satisfying the strong law of large numbers.

Proof: The queueing network can be studied through the fluid equations (5) and (6) introduced in Section III-C. With a strict priority service discipline, $T(t)$ satisfies, $\forall k$, the following variation of (7):

$$\int_0^\infty \left[\sum_{h\in L^{-1}_{k^+}(j)} x^{(h)}(t)\right] d\left[t-\sum_{h\in L^{-1}_{k^+}(j)} t^{(h)}(t)\right] = 0 \quad (12)$$

where $L^{-1}_{k^+}(j)$ is the function that returns the set of virtual queues located at station j, whose priority index is not greater than $p^{(k)}$ (note that virtual queue k itself belongs to this set). According to (12), the server must work on the highest priority virtual queue, whenever there is some unit of work to be performed there.

Since the priority dependency graph is acyclic, we can associate an ordinal number o_f with $1 \leq o_f \leq F$ to each flow f in the network satisfying the following rule: for any pair of flows f_m and f_n, with f_m father of f_n in the priority dependency graph, then $o_{f_m} < o_{f_n}$. Note that o_f induces an order relation among virtual queues. However, this order relation is quite weak, since all virtual queues corresponding to the same flow correspond to the same ordinal number. In order to strengthen the previously defined order relation among virtual queues, we proceed as follows. We associate an ordered pair of indices with each virtual queue: the first index corresponds to the ordinal number of the flow that originates the considered virtual queue, while the second index, that we denote with $h^{(k)}$, represents the number of queues that the considered flow has traversed before virtual queue k. Then, a global order relation $\mathcal{GO}$ can be introduced among virtual queues, according to the following rule: virtual queue k_m precedes virtual queue k_n, iff $o_{f_m} < o_{f_n}$, or $o_{f_m} = o_{f_n}$ and $h^{(k_m)} < h^{(k_n)}$ along the path described by $f_m = f_n$.

Now we are ready to proceed with the proof. We first assume that there are no stations where two or more virtual queues have the same priority; then we will relax this assumption. Under that assumption, we claim that:

Claim 1: Given any state vector $X(t)$ whose components are sorted in increasing order according to $\mathcal{GO}$, indicating with k_0 the first non-null component of $X(t)$ and $L(k_0) = j_0$, if t is a regular point for $X(t)$, then:

$$\dot{x}^{(k_0)}(t) = \left(\sum_{k \in L^{-1}_{k_0+}(j_0)} \omega^{(k)} - 1 \right) \gamma^{(k_0)} = \chi < 0$$

Proof: Let f_0 be the flow that originates k_0. For any empty virtual queue k in a regular point t, $\dot{x}^{(k)}(t) = 0$, as proved for a general fluid model in [13]; then, the fluid departure rate equals the fluid arrival rate. By induction, it is immediate to verify that $\dot{e}^{(k)}(t) = \pi^{(k)}$ at each virtual queue k that follows only empty queues along the path described by its traffic flow. But virtual queue k_0, by construction, follows only empty queues along the path described by its traffic flow. Moreover, every virtual queue originated by a father of f_0 must be empty by construction. Thus, for all the higher priority virtual queues k co-located with k_0, $\dot{e}^{(k)}(t) = \dot{d}^{(k)}(t) = \pi^{(k)}$. We can write: $\dot{x}^{(k_0)}(t) = \dot{e}^{(k_0)}(t) - \dot{d}^{(k_0)}(t)$. But $\dot{e}^{(k_0)}(t) = \pi^{(k_0)}$, and $\dot{t}^{(k_0)}(t)$ is equal to 1 minus the sum of $\omega^{(m)}$ for all queues m residing at physical queue j_0, but having a higher priority than queue k_0. Hence $\dot{d}^{(k_0)}(t)$ is equal to $\gamma^{(k_0)}$ minus the sum of $\pi^{(m)}$ at co-located, higher-priority queues. The claim follows immediately, taking into account the fact that $\sum_{k \in L^{-1}_{k_0+}(j_0)} \omega^{(k)} < 1$. ■

Consider now another non-empty virtual queue m. We do not have much information on the behavior of $\dot{x}^{(m)}(t)$; however, the maximum amount of fluid which can arrive at it in a unit of time is bounded, due to the boundedness of the service capacity of all queues that come before virtual queue m along the path described by the considered traffic flow (recall that matrix Δ is bounded). Thus, there exists a constant C such that $|\dot{x}^{(m)}(t)| < C$, $\forall m$. In particular, we can take $C > |\chi|$. Consider the following Lyapunov function:

$$\mathcal{L}(X(t)) = \sum_{k=1}^{N} x^{(k)}(t) \left(\frac{|\chi|}{2C} \right)^{k}$$

To prove stability, we must verify that, whenever $X(t) > 0$, then $\dot{\mathcal{L}}(X(t)) < 0$. Indeed, being empty all queues that come before k_0 (so that for all those queues $\dot{x}^{(k)}(t) = 0$), and being $\dot{x}^{(k_0)}(t) = \chi$, we can write:

$$\begin{aligned} \dot{\mathcal{L}}(X(t)) &\leq \left(\tfrac{|\chi|}{2C}\right)^{k_0} \left[\chi + \textstyle\sum_{i=1}^{N-k_0} C \left(\tfrac{|\chi|}{2C}\right)^{i}\right] = \\ &= \left(\tfrac{|\chi|}{2C}\right)^{k_0} \left[\chi + \tfrac{|\chi|}{2} \textstyle\sum_{i=0}^{N-k_0-1} \left(\tfrac{|\chi|}{2C}\right)^{i}\right] < 0 \end{aligned}$$

Thus, $X(t) = 0$ is the only solution[8] of the fluid model for the initial condition $X(0) = 0$. The fluid model is weakly stable and then the stochastic system is rate-stable for Theorem 2.

With similar arguments, the proof can be extended to the case in which more flows that visit the same physical queue fall in the same priority class. Due to space limitations, we only sketch here the main steps of the proof.

If different flows with the same priority visit the same physical queue, we must partition traffic flows in classes, denoted with K_i, $i = 1, \ldots, L$, where L is the cardinality of the partition, and characterized by the same ordinal number. We order these classes in such a way that if $i < j$ then $o_{f_m} < o_{f_n}$ $\forall f_m \in K_i, f_n \in K_j$. Let K_α represent the highest priority group of flows to which corresponds at least a non-empty virtual queue, and use now k to indicate any virtual queue belonging to a flow in K_α. Moreover, let $i < \alpha$, so that K_i denotes any priority class higher than K_α, and let h indicate any virtual queue belonging to a flow in K_i.

The presence of lower priority virtual queues does not perturb the behavior of $\dot{x}^{(k)}(t)$, because of the strict priority discipline. Moreover, it results $\dot{x}^{(h)}(t) = 0$ at every regular point t. Thus, $\dot{t}^{(h)}(t) = \omega^{(h)}$. This allows us to locally study the behavior of $\dot{x}^{(k)}(t)$: indeed, virtual queues belonging to flows in lower priority classes can be ignored, while every virtual queue h subtracts at its correspondent server $j = L(h)$ an amount of work equal to $\omega^{(h)}$. In other words, the study of local behaviors of virtual queues k can be reduced to the study of the evolution of queue lengths in an underloaded single priority system in which virtual queues are served according to a FIFO discipline.

By applying the same Lyapunov function $\mathcal{L}_{FIFO}$ used in [17] [9] and denoting with $1\!\mathrm{I}_{K_\alpha}$ a $N \times N$ diagonal matrix that has unit diagonal elements in correspondence of virtual queues belonging to K_α and null elements elsewhere, it is possible to conclude that $\dot{\mathcal{L}}_{FIFO}(1\!\mathrm{I}_{K_\alpha} X(t)) < \chi < 0$. For the lower priority classes K_j, there exists a constant $C > 0$ such that $\dot{\mathcal{L}}_{FIFO}(1\!\mathrm{I}_{K_j} X(t)) < C$, since all the growing rates of queues are finite. Thus, defining:

$$\mathcal{L}(X(t)) = \sum_{i=1}^{L} \left(\frac{|\chi|}{2C} \right)^{i} \mathcal{L}_{FIFO}(1\!\mathrm{I}_{K_i} X(t))$$

it results $\dot{\mathcal{L}}(X(t)) < 0$. Then the queueing network is rate-stable, again for Theorem 2. ■

[8] Assume $\mathcal{L}(t) \geq 0$, $\mathcal{L}(0) = 0$, and $\dot{\mathcal{L}}(t) \leq 0$. Consider $\mathcal{L}^2(t) = 2\int_0^t \mathcal{L}(x)\, d\mathcal{L}(x)$. By definition $\mathcal{L}^2(t) \geq 0$, but $\int_0^t \mathcal{L}(x)\, d\mathcal{L}(x) \leq 0$. Hence $\mathcal{L}(t) = 0$, $\forall t$ (see [13]).

[9] $\mathcal{L}_{FIFO} = \sum_{j=1}^{J} \sum_{k \in L^{-1}(j)} \int_t^{t+W_j(t)} \pi^{(k)} h(\dot{d}^{(k)}(s)/\pi^{(k)}) ds$ with $h(x) = x \log(x)$ and $W_j(t) = \sum_{k \in L^{-1}(j)} s^{(k)} x^{(k)}(t)$.

TABLE I

PRIORITY ASSIGNMENTS FOR FLOWS IN THE NETWORK

Source	Destination	Route	Priority	Phase
1	9	(b,n,h)	1	0
1	11	(b,c,d,p,i)	1	1
8	9	(h)	0	1
2	3	(c)	0	2
8	11	(h,o,d,p,i)	1	2
3	11	(d,p,i)	0	3

We notice that a strict parallelism exists among the stability result for acyclic strict-priority/FIFO queueing networks, and the stability result for networks of GPS schedulers under the consistent relative session treatment proved in [2]. The concept of acyclicity in strict-priority networks, indeed, can be considered an extention to the case of strict-priority/FIFO disciplines of the concept of consistent relative session treatmenent defined for GPS schedulers. The extension is, however, non trivial as the following results will show.

We discuss next the property of universal stability for acyclic strict-priority/FIFO queueing networks.

Theorem 9: Acyclic strict-priority/FIFO queueing networks are universally stable under any $[\rho, \phi]$ regulated traffic pattern with arbitrary ϕ and average external arrival rate $\rho < 1$, if no pairs of flows traversing the same station have the same priority.

Proof: This proof is similar to the previous one; we omit it due to the lack of space, but the interested reader can find it in [21]. ■

The extension of the previous result to the case in which several flows share the same priority is impossible, since it was proved [14] by counterexample that the FIFO service discipline is in general not universally stable for $\rho > 0.85$. Note, indeed, that an acyclic strict-priority/FIFO network reduces to a FIFO network when the same priority is assigned to every flow. The unstable example studied in [14], however, requires that some packets follow closed paths (i.e. trajectories that start and end at the same node). Thus, the reader may wonder whether acyclic strict-priority/FIFO networks can be still proved to be not universally stable under the further restriction that packet routes are selected according to "reasonable" routings.

The following generalization of the example reported in [14] shows that instability still arises in acyclic strict-priority/FIFO networks in which packets are routed according to a shortest path routing. It also shows that the adoption of an acyclic strict-priority/FIFO discipline entails a further reduction of the stability region with respect to a pure FIFO discipline.

Theorem 10: Acyclic strict-priority/FIFO queueing networks are **not** universally stable for average external arrival rates ρ greater than 0.8, when assuming all servers have unitary capacity.

Proof: Consider the queueing network in Figure 4, where queues are represented by numbered circles, and labelled edges represent possible customer routes. In Table I we specify customer flow routes and priorities (where smaller numbers mean higher priority).

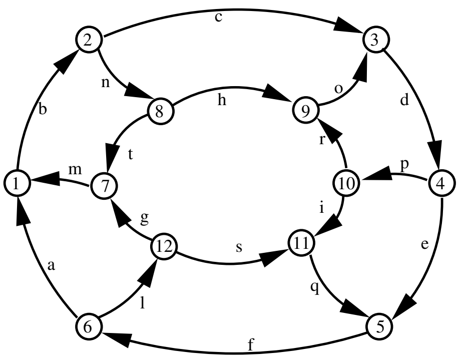

Fig. 4. An acyclic strict-priority/FIFO queueing network that is not universally stable

We will show that a (ρ, ϕ) adversary exists under which some queue lengths in this network grow indefinitely.

We divide the network evolution into periods. Each period is itself sub-divided into phases. Table I also reports the phase in which each flow enters the network. In the following, we consider each phase lasting W_0 time units.

Let us start by examining the network behavior during the first period. We suppose that each server in the network serves flows according to an acyclic strict-priority/FIFO discipline. We suppose that, at the beginning of the period, W_0 packets are stored in the network at node 1 directed to node 9 over path (b, n, h).

During the first phase, ρW_0 packets arrive at node 1, directed to node 11, over path (b, c, d, p, i); in addition, ρW_0 packets are injected at node 8, directed to node 9 over path (h).

During this phase, all the W_0 packets $1 \to 9$ move from node 1 to node 8, since they are older than packets $1 \to 11$; thus, at the end of the phase, all packets $1 \to 11$ are still enqueued at node 1. In addition, at the end of the phase ρW_0 packets belonging to flow $1 \to 9$ are still enqueued at node 8, where the ρW_0 packets belonging to flow $8 \to 9$ could access link h thanks to the higher priority and only $(1-\rho)W_0$ packets $1 \to 9$ coulb be serviced.

During the second phase, ρW_0 packets are injected at node 8 directed to node 11 over path (h, o, d, p, i), and ρW_0 packets are injected at node 2 directed to node 3 over path (c).

During this phase, all the higher priority packets belonging to flow $2 \to 3$ reach their destination, along with the packets belonging to flow $1 \to 9$, that can access link h being older than packets $8 \to 11$. While all the ρW_0 packets $1 \to 11$ have been served at node 1, the residual server capacity at nodes 2 and 8 is exploited by $(1-\rho)W_0$ packets belonging respectively to flows $1 \to 11$ and $8 \to 11$. As a result, at the end of this phase $\rho W_0 - (1-\rho)W_0 = (2\rho - 1)W_0$ packets belonging to flow $1 \to 11$ are enqueued at node 2 and $(2\rho - 1)W_0$ packets belonging to flow $8 \to 11$ are enqueued at node 8.

During the third phase, ρW_0 packets are injected at node 3, directed to node 11 over path (d, p, i). Packets belonging to flows $1 \to 11$ and $8 \to 11$ are served at nodes along their paths until they reach node 3. Thus, all packets in the system converge at node 3: here, priority is given to flow $3 \rightarrow 11$ so that its ρW_0 packets can reach their destination. The residual server capacity at node 3 is exploited by some packets belonging to flows $1 \rightarrow 11$ and/or $8 \rightarrow 11$ (they have the same priority, so packets choice depends on the FIFO order). At the end of the phase, $W_1 = 2(2\rho-1)W_0-(1-\rho)W_0 = (5\rho-3)W_0$ packets directed to node 11 are still enqueued at node 3.

If $\rho > 0.8$, W_1, the number of packets enqueued at node 3 at the end of the first period, exceeds W_0. Note that the initial condition for the second period closely resembles the initial condition for the first period (the topology clearly exhibits a symmetry under which paths $1 \rightarrow 9$ and $3 \rightarrow 11$ are equivalent). Thus, there exists an external arrival pattern such that at the end of the second period $W_2 > W_1 > W_0$ packets are stored at node 5 and directed to node 7. Finally, with similar considerations, it is possible to show that there exists an external arrival pattern such that, at the end of the third period, W_3 packets (with $W_3 > W_2 > W_1 > W_0$) are stored at node 1 and directed to node 9 through path (b, n, h). By induction, we can prove that the number of packets left in the network at the end of each period indefinitely grows, thus the network is not universally stable if $\rho > 0.8$. ■

V. Conclusions

In this paper we have discussed possible underload instabilities due to GPS and strict priority schedulers in packet-switching networks, considering scenarios with acyclic packet routes, and service times that vary only according to channel capacities.

Our analysis extends recent results showing that instability can happen in underloaded queueing networks, loosening the system assumptions in a way that our findings can be applied to the approaches being currently considered for QoS provisioning in the Internet.

Our main results are that: i) GPS schedulers may be unstable when some of the actual packet rates exceed the GPS rates; ii) strict priority schedulers are stable, provided that the priority ordering of packet flows does not change from a router to another; iii) the combination of priority and FIFO schedulers may lead to a weak form of instability for loads larger than 0.8.

In [22], using different techniques, we show that most instability phenomena shown in this paper are not mitigated by an adaptive behavior of traffic sources, as for example in the case of general additive-increase multiplicative-descrease source rate adaptation, and of TCP algorithms in particular.

References

[1] A.K.Parekh, R.G.Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks - The Single Node Case" *IEEE/ACM Transactions on Networking*, Vol. 1, n. 3, June 1993, pp. 344-357.

[2] A.K.Parekh, R.G.Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks - The Multiple Node Case" *IEEE/ACM Transactions on Networking*, Vol. 2, n. 2, April 1994, pp. 137-150.

[3] J.Heinanen, F.Baker, W.Weiss, J.Wroclawski, *Assured Forwarding PHB Group,* RFC 2597, June 1999.

[4] V.Jacobson, K.Nichols, K.Poduri, *An Expedited Forwarding PHB,* RFC 2598, June 1999.

[5] L.Kleinrock, *Queueing Systems*, Vol. 2, John Wiley, New York, 1976.

[6] J.R.Jackson, "Jobshop-like Queueing Systems", *Management Science*, Vol. 10, n. 1, October 1963, pp. 131-142.

[7] F.Baskett, K.M.Chandy, R.R.Muntz, F.Palacios, "Open, Closed and Mixed Networks with Different Classes of Customers", *Journal of the ACM*, Vol. 22, n. 2, April 1975, pp. 248-260.

[8] F.P.Kelly, *Reversibility and Stochastic Networks*, John Wiley, New York, 1979.

[9] S.H.Lu, P.R.Kumar, "Distributed Scheduling Based on Due Dates and Buffer Priorities", *IEEE Transactions on Automatic Control*, Vol. 36, n. 12, December 1991, pp. 1406-1416.

[10] J.G.Dai, G.Weiss, "Stability and Instability of Fluid Models for Re-Entrant Lines", *Mathematics of Operations Research*, Vol. 21, 1996, pp. 115-135.

[11] M.Andrews, L.Zhang, "Achieving Stability in Networks of Input-Queued Switches", *INFOCOM 2001*, Anchorage, Alaska, April 2001, pp. 1673-1679.

[12] M.Ajmone Marsan, E.Leonardi, M.Mellia, F.Neri "On the Maximum Throughput Achievable in Multi-Class Input-Queued Switches and Networks of Switches", *INFOCOM 2002*, New York, NY, June 2002, pp. 1605-1614.

[13] J.G.Dai, *Stability of Fluid and Stochastic Processing Networks,* Miscellanea Publication n.9, Centre for Mathematical Physics and Stochastic, Denmark (http://www.maphysto.dk), January 1999.

[14] M.Andrews, B.Awerbuch, A.Fernandez, J.Kleimberg, T.Leighton, Z.Liu, "Universal Stability Results for Greedy Contention-Resolution Protocols", *Journal of the ACM*, Vol. 48, n. 1, January 2001.

[15] D.Gamarnik, "Using Fluid Models to Prove Stability of Adversarial Queueing Networks", *IEEE Transactions on Automatic Control*, Vol. 45, n. 4, April 2000, pp. 741-746.

[16] P.R.Kumar, S.P.Meyn, "Stability of Queueing Networks and Scheduling Policies", *IEEE Transactions on Automatic Control*, Vol. 40, n. 2, February 1995, pp. 251-260.

[17] M.Bramson, "Convergence to Equilibrium for Fluid Models of FIFO Queueing Networks", *Queueing Systems*, Vol. 22, 1996, pp. 5-45.

[18] A.Borodin, J.Kleimberg, P.Raghavan, M.Sudan, D.P.Williamson, "Adversarial Queueing Theory", *Journal of the ACM*, Vol. 48, n. 1, January 2001.

[19] R.L.Cruz, "A Calculus for Network Delay, Part I: Network Elements in Isolation", *IEEE Transactions on Information Theory*, Vol. 37, n. 1, January 1991, pp. 114-131.

[20] R.L.Cruz, "A Calculus for Network Delay, Part II: Network Analysis", *IEEE Transactions on Information Theory*, Vol. 37, n. 1, January 1991, pp. 132-141.

[21] M.Ajmone Marsan, M.Franceschinis, E.Leonardi, F.Neri, A.Tarello, *Instability Phenomena in Underloaded Packet Networks with QoS Schedulers,* Technical Report, `http://www.tlc-networks.polito.it/˜ emilio/net-rep/instability-tec-rep.pdf`

[22] M.Ajmone Marsan, M.Franceschinis, P. Giaccone, E.Leonardi, F.Neri, A.Tarello, *Instability Phenomena in Underloaded Packet Networks with Elastic Traffic,* Technical Report, available at: `http://www.tlc-networks.polito.it/˜ emilio/net-rep/elastic.pdf`