On the Use of Fixed Point Approximations to Study Reliable Protocols over Congested Links

Michele Garetto, Marco Ajmone Marsan, Michela Meo
Dipartimento di Elettronica, Politecnico di Torino
Corso Duca degli Abruzzi, 24 – 10129 Torino, Italy
Email: {garetto,ajmone,michela}@polito.it

Renato Lo Cigno
Dipartimento di Informatica, Università di Trento
Via Sommarive, 14 – 38050 Povo, Trento, Italy
Email: locigno@dit.unitn.it

***Abstract*— Analytical approaches for the performance investigation of portions of the Internet often consider the behavior of TCP over congested (or bottleneck) links. In several cases, the analysis is based on an iterative Fixed Point Approximation (FPA) to compute the equilibrium point, in terms of packet loss rate and offered load, that represents the operating point of the network. Almost invariably, the FPA is conjectured to converge, but no proof of convergence is provided. This paper proves that a general model of a reliable protocol (such as TCP) over congested links converges to a unique stable solution under mild regularity conditions. This provides a justification of the convergence observed in the literature and a solid base for the further development of analytical approaches based on FPAs.**

I. Introduction

Literature on protocol and network performance concentrates on heavy congestion conditions. There are obvious reasons to this choice, specially when reliable protocol, whose aim is the retransmission of lost or corrupted packets, come into play. In recent years there has been a huge amount of work concerning TCP (a reliable protocol with congestion control features) over Internet congested links (the bottlenecks). Within this framework the analysis under heavy congestion assumes a critical importance, since many Internet links, specially access links and peering connections, tend to exhibit long periods of overload. Fig. 1 shows the utilization of a campus access link measured with MRTG, a monitoring tool widely used throughout the world [1]. This is a daily graph in which each point is the result of a 5 minute average. The link capacity is 16 Mbits/s. These are the conditions where a reliable protocol is most stressed, and hence those where its properties are most interesting.

Concentrating on time periods in which the link is almost saturated, a number of interesting questions, that go beyond throughput analysis, arise: i) what are typical values for packet loss rate when the link utilization is close to 100%? ii) is it possible to push the link utilization to 100% with stable network operation? iii) what are the sustainable values of link utilization without the risk of network instability? The performance analysis of TCP/IP networks, carried out with either analytical approaches or simulations, often gives for granted fundamental stability results, applying steady-state analysis without properly checking its applicability. In this paper we provide an answer to such fundamental questions in a general framework, where only some very elementary properties of the protocols and the network come into play. We give a formal proof of the convergence of the Fixed Point Approximation (FPA) technique often adopted to find the operating point of the network [3], [2]. Analytical predictions are confirmed by simulation experiments with *ns-2*. We also briefly discuss the implications of our results, suggesting original insights into the effectiveness of the congestion control currently adopted in the Internet to prevent network instability.

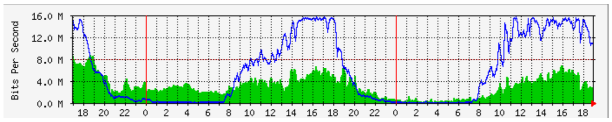

Fig. 1. Typical daily variations on a bottleneck link

To the best of our knowledge, the convergence of the FPA was proved only for the case of long-lived connections [2] with AQM routers. Our work formally extends the applicability of this technique to the important case of short-lived flows, as well as relaxing the hypothesis that AQM techniques are deployed at bottlenecks. The behavior of finite flows under heavy congestion has been considered also in [5], [6] by means of a different approach based on a processor sharing model.

II. Modeling assumptions

We focus on an access link that connects a LAN to the Internet. Users inside the LAN request files from servers around the world. We assume that all traffic traversing the link is carried by TCP connections, and that losses happen only on the access link, which is considered to be heavily congested in the direction incoming into the LAN (the case of Fig. 1). ACKs are never lost on the reverse path. Moreover, the router responsible of dropping packets adopts a simple drop-tail queuing discipline.

We recognize this as being a simplified scenario, but we believe it contains much of the problem complexity that is found in more complicated multi-bottleneck topologies: a deep understanding of the behavior of a single congested link is the fundamental first step before considering more complicated scenarios.

Let μ be the channel capacity expressed in packets per second and B the buffer size in packets. We assume that the amount of data to be transferred by each connection, expressed in number of constant-size packets (the connection size), is taken from a known distribution with average $\bar{S}$.

Let γ be the average connection establishment rate. A realistic arrival process of new connections is complicated to describe analytically. Statistical models for TCP start times that have been proposed in the literature vary from simple

Poisson processes to sophisticated models such as that presented in [7]. In this work we are only concerned with the mean establishment rate of new connections. This value is not constant over long time scales, because it follows the typical daily variations shown in Fig. 1, which is essentially due to the variable number of users accessing the Internet inside the LAN at a given time of the day. Steady-state analysis implies neglecting these slow traffic variations and assuming that the system is at an operating point where the arrival of new connections is described by a stationary random process.

We define as *normalized goodput* the quantity $g = \frac{\gamma \bar{S}}{\mu}$. The notion of *goodput* for a single TCP connection normally corresponds to the rate of packets delivered by the transport layer to the upper layers. Here we consider the average aggregate goodput of all connections traversing the link, and we normalize it by the link capacity. Of course, in a stable system g cannot exceed one.

Let λ be the arrival rate at the bottleneck router of packets incoming into the LAN. The *traffic intensity* ρ is defined as usual in queuing theory as $\rho = \lambda/\mu$. We also introduce the *normalized throughput* as the quantity $t = \rho \cdot (1-p)$ where p is the average packet loss at the bottleneck router. The notion of *throughput* for a single TCP connection normally corresponds to the arrival rate of packets at the receiver. Here we consider the average aggregate throughput of all connections traversing the link, and we normalize it by the link capacity. Since we assume that losses happen only at the bottleneck link, all packets that are not dropped by the access router arrive at the receivers inside the LAN.

In general $g \leq t$, because TCP is not perfect, and sometimes it performs some *unnecessary retransmissions*: a TCP source sends again a segment of data that was already received by the destination or that is still 'in flight' along the path toward the destination.

Packets are dropped at the bottleneck router because of the finite capacity of the buffer storing the packets going through the link. If the buffer were infinite we would have $g \leq t = \rho$, since packets are never discarded. In this case the average queue length would grow to infinity when ρ approaches 1.

Since in real networks it is not possible to guarantee that ρ is always lower than one, it is necessary to limit the buffer size to prevent intolerable delays caused by traffic peaks. With a finite buffer, the packet loss probability is always greater than zero (when ρ is small it is possible to run long simulations without seeing any losses, but any analytical model of the queue with non-deterministic traffic at the input predicts a non-zero packet loss probability) so that we have $g \leq t < \rho$ for any finite value B of buffer size.

In Section III we restrict the analysis to the simple case of no unnecessary retransmissions, so that $g = t$. This corresponds to an ideal protocol that provides a perfect selective retransmission mechanism. This simplification is removed in Section IV.

III. Absence of unnecessary retransmissions

If no unnecessary retransmissions exist, we can establish a simple exact relation for ρ as a function of g and p,

$$\rho = \frac{g}{1-p} \tag{1}$$

which follows immediately from the consideration that TCP is a reliable protocol, and the throughput is equal to the goodput. Unfortunately, p is unknown. Thus we need another relation between ρ and p, that must be given by an analytical model of the packet queue at the bottleneck router buffer. A simple $M/M/1/K$ queue provides a closed-form formula of p as a function of ρ:

$$p = \frac{\rho^B \ (1-\rho)}{1-\rho^{B+1}} \tag{2}$$

The $M/M/1/K$ queue is known to be ill-suited to Internet traffic, because it assumes a Poisson arrival process of packets at the queue, that has been experimentally proved wrong [8]. The main reason is that a Poisson arrival of packets does not account for the burstiness of TCP transmissions within the same window of data. It has been shown instead [4] that an $M^{[X]}/M/1/K$, a queue with a Poisson arrival of batches of packets distributed according to a batch size distribution $[X]$ closely related to the window size distribution of the senders, provides a good estimate of the packet loss probability for a wide range of values of ρ. Unfortunately, the $M^{[X]}/M/1/K$ queue does not allow a closed-form expression of p as a function of ρ. To carry on the analysis we can resort to an approximation using the formula of the $M/M/1/K$: suppose that all batches are of equal size b, and that a batch is completely lost if it cannot be entirely received into the queue. The probability of losing a batch is then given by (2) where B is replaced with $B^* = B/b$, and it is also equal to the loss probability of individual packets. Actually, batches should be variable in size, and not all packets are really lost if a batch cannot be entirely received into the buffer, but we can always find a suitable value B^* to be used with (2) to obtain the correct packet loss probability, at least in the neighborhood of a given value of ρ. In general, $B^* = B$ provides a lower bound to p, because TCP traffic is more bursty than a Poisson process, while $B^* = W_M$, where W_M is TCP maximum window size, provides an upper bound that roughly corresponds to the behavior of earlier versions of TCP [9] before congestion control was introduced [10]. This implies that the true behavior of TCP, whatever version we consider, must be between the upper and lower bounds that can be analyzed in closed form. Moreover, on a highly congested link, the difference between the correct value of p and the one obtained with a simple $M/M/1/K$ becomes smaller and smaller, so that an $M/M/1/K$ queue alone indeed provides a good approximation of the packet loss probability (this is also due the fact that the burstiness of the traffic is reduced because the TCP congestion window is small).

A stable network operating point can be reached only if equations (1) and (2) admit a (single) joint solution. Indeed, network operations closely resemble an FPA solution of the system: new connections start assuming zero loss probability ($p = 0$) and inject traffic in the network; the network provides a new (different) value of p by dropping packets. Sources react to this packet loss probability adjusting their sending rate until convergence is reached.

In the rest of this Section we prove that the solution of this fixed point algorithm (FPA) exists according to (1) and (2), it is unique, and it is stable under the only condition that $g < 1$. Note that g cannot be equal to one, because there is

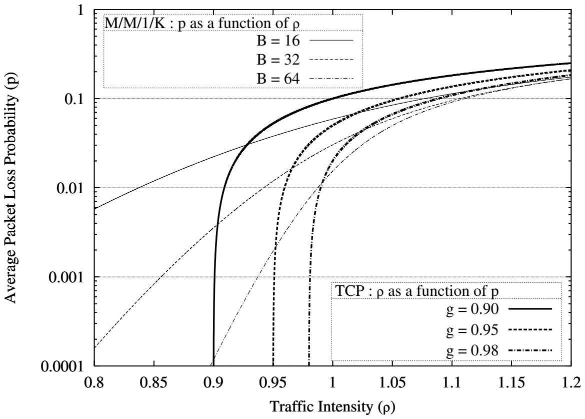

Fig. 2. Graphical solution of the FPA

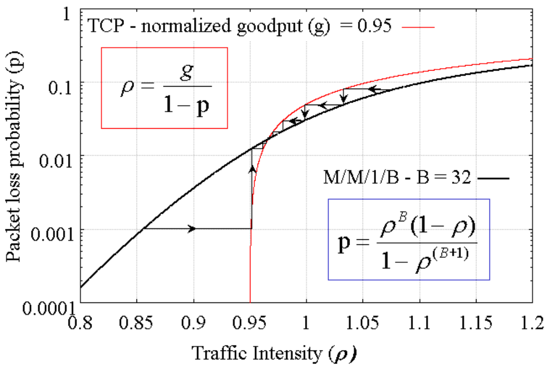

Fig. 3. Example of convergence of the FPA to the only stable solution

always a non-zero probability that the buffer empties, so that the inequality $t < 1$ holds strictly.

A graphical way to obtain the solution is shown in Fig. 2, where we have plotted two families of curves on the plane (p, ρ) according to (1) and (2) for different values of B and g. An admissible solution is given by the intersection of a curve of type (1) and a curve of type (2).

Theorem 1 — The system defined by (1) and (2), admits one and only one solution, which is a stable operating point for a network under the only condition $g < 1$.

Proof — We first prove the existence and uniqueness of the solution for Poisson packet arrivals, then we extend the proof to the general case of a traffic characterized by a burstiness that decreases for increasing values of traffic intensity ρ.

Existence and uniqueness for Poisson packet arrivals — Substituting p into (2) yields

$$f(\rho) = \frac{\rho^{B+1}\,(1-g) - \rho + g}{\rho(1-\rho^{B+1})} \tag{3}$$

and we can study the zeroes $f(\rho) = 0$ in the interval $\rho \geq g$ $(g < 1)$. $f(\rho)$ is continuous over this range of values, as $\rho = 1$ is an apparent singularity that can be factored out. We simply need to find the zeroes of the polynomial

$$h(\rho) = 1 + (g-1)(1+\rho+\rho^2+ ... +\rho^B) \tag{4}$$

There exist one and only one solution since $h(g) = g^{B+1}$ is positive, $h(\frac{g}{1-g})$ is negative and $h'(\rho) < 0$ for any $\rho > 0$. The solution is also stable, because from any initial condition the system converges to the equilibrium point given by the unique solution $(\bar{p}, \bar{\rho})$, since both (1) and (2) are monotonic. □

As we said before, in order to obtain an accurate result using the $M/M/1/K$ queue model, it is necessary to account for the burstiness of TCP traffic. This can be done using a suitable value $B^* = B/b$ for the buffer size, where b is proportional to the average window size of the flows traversing the link. Since the above proof is valid for any buffer size, our result can be immediately extended to the case of a bursty traffic modeled by a constant factor b. However, one may question the uniqueness of the solution in the case in which the traffic burstiness is not constant. Now we will prove that the solution is unique under the assumption that the burstiness is a decreasing function of ρ. This assumption is reasonable, because if we increase the load we increase the multiplexing of the packets, and we also increase the packet loss probability, resulting in a reduced congestion window size.

Uniqueness with bursty traffic — The proof is ab absurdo. Suppose that there are two different solutions for the traffic intensity $\rho_1 < \rho_2$. Since we assume that the burstiness, modeled by the scaling factor b, is a decreasing function of ρ, we have $b_1 > b_2$, and consequently $B_1^* < B_2^*$. Since $\rho_1 < \rho_2$ are both zeroes of $h(\rho)$, we should have

$$1 + \rho_1 + \rho_1^2 + ... + \rho_1^{B_1^*} = 1 + \rho_2 + \rho_2^2 + ... + \rho_2^{B_2^*}$$

but this is clearly impossible if $\rho_1 < \rho_2$ and $B_1^* < B_2^*$. □

IV. Presence of unnecessary retransmissions

The presence of unnecessary retransmissions makes the problem more complex. To proceed in the analysis, we need a model for the amount of unnecessary retransmissions produced by TCP. Unfortunately, this phenomenon is not straightforward and depends on a number of variables (average packet loss probability, flow length, TCP parameters and protocol version). Lacking a better approach we resorted to measure the amount of unnecessary retransmissions in simulation for a few different cases, and to model the phenomenon approximating the points through a simple empirical law. We remark, however, that the FPA convergence property is not sensitive to the unnecessary retransmission law, and a different law would only change the numerical result (operating point), not its existence.

We consider the New-Reno version of TCP, which seems to be the most popular in the Internet nowadays [11]. The SACK option is expected to reduce significantly the amount of unnecessary retransmissions, but we did not consider it because it is still rarely adopted (only about 10% of connections successfully negotiate this option, according to measurements reported in [12]).

Let u be the *normalized rate of unnecessary retransmissions* arriving at the bottleneck router, which is the rate of unnecessary packets sent by the sources, normalized with respect to the link capacity. We also introduce l, the *normalized rate of necessary retransmissions*, defined as

$$l = \frac{g\,p}{1-p} \tag{5}$$

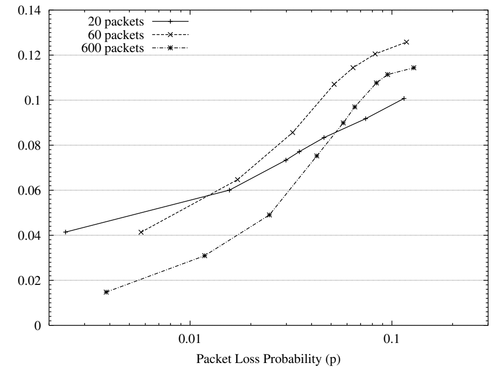

Fig. 4. The coefficient θ as a function of p

which is actually equal to the rate of 'necessary packets' dropped by the queue (a necessary packet, if lost, requires a necessary retransmission). The unnecessary retransmissions u are known to increase with the congestion and the drop rate at the bottleneck. A linear relationship $u = \theta\ l$, where θ is some suitable coefficient, is the simplest relation making u grow with l. Note that we assume u dependent only on the loss rate of 'necessary packets' since dropped repeated packets do not trigger any retransmission.

Fig. 4 reports several values of θ obtained with simulations, plotted against the average drop rate p. Results refer to a bottleneck capacity $\mu = 3500$ packets/s, buffer size $B = 128$, maximum window size 32, and flow size taken from a geometric distribution with average $\bar{S} = 20, 60, 600$ (all parameters expressed in packets). The relationship between θ and the average packet loss probability p is clearly not a constant, but a reasonable approximation for high loads (the most interesting situation) is 0.1. Using a constant value we lose some residual dependency on the packet loss probability and the average flow size, but the analysis is simplified.

The total traffic intensity arriving at the queue is given by

$$\rho = g + l + u \ = \ \frac{g(1+\theta\ p)}{1-p} \tag{6}$$

The *normalized throughput* t derives immediately from (6) as $t = g(1+\theta\ p)$. Since t must be less than 1 under stability, it follows

$$g(1+\theta\ p) < 1 \tag{7}$$

which defines a stability region on the plane (p, g) as shown in Fig. 5. No stable point can exist above the curve defined by (7), as confirmed by simulations (each simulation point corresponds to an experiment in which the number of active connections does not diverge for at least $1000\ s$).

Theorem 2 — The system defined by (6) and (2), admits zero, one or two solutions, depending on the value of g. When a single solution exist, it is a stable operating point for a network; when two solutions exist, only one is a stable operating point for a network, while the other it is not an admissible operating point; in regions where no solution exists, no stable operating point exists.

Proof — We follow the same approach used for the proof of *Theorem 1*. Eliminating p from (6) and (2) yields a single equation, whose solutions are given by the zeroes of the polynomial

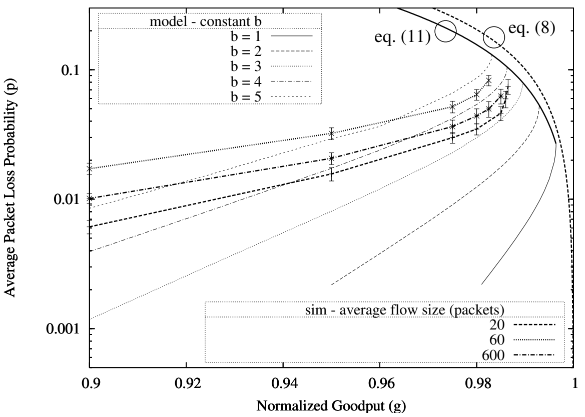

Fig. 5. Stability region and simulation points on the plane (p, g)

$$h(\rho) = 1+(g-1)(1+\rho+\rho^2+...+\rho^{B-1})+\rho^B(g\,\theta+g-1) \tag{8}$$

in the interval $\rho \geq g$, assuming $0 < g < 1$. Note that if $\theta = 0$ we obtain again (4); $h(g) = g^{B+1}(1+\theta)$ is definite positive. The behavior for $\rho \to \infty$ is given by the sign of the coefficient $(g\,\theta+g-1)$ of the highest order term. We have that $h(\rho)$ tends to $-\infty$ if the following inequality holds

$$g \leq \frac{1}{1+\theta} \tag{9}$$

otherwise it goes to $+\infty$.

To determine the number of positive solutions it is convenient to use Descartes' rule of signs. If inequality (9) holds, $h(\rho)$ has only one change of sign in its coefficients, and we conclude that it has exactly one positive solution. This is actually the case considered in Section III. If inequality (9) does not hold, $h(\rho)$ has two changes of sign in its coefficients, and therefore it has either zero or two positive solutions (possibly coincident). More precisely, for $0 < g < 1/(1+\theta)$ there is first a single solution. For $g \leq 1/(1+\theta)$ there are two solutions, initially far apart, that get closer to each other until they coincide into one solution $\bar{\rho}_1 = \bar{\rho}_2 = \bar{\rho}$ of multiplicity equal to two, for $g = g^*$. When $g > g^*$ there is no solution, and the system does not admit a stable operating point.

If there are two solutions $\bar{\rho}_1 < \bar{\rho}_2$, we have $h'(\bar{\rho}_1) < 0$, and $h'(\bar{\rho}_2) > 0$, so that $\bar{x}_1$ (the solution corresponding to ρ_1) is always stable, while $\bar{x}_2$ is necessarily unstable. □

An example of this situation is shown in Fig. 6, where we have used $\theta = 0.5$ to increase the distance between the two curves, for the sake of a better representation. Depending on the starting point, the FPA converges to the stable solution or diverges to infinity. Notice that this results is not dependent on the modeling technique used to represent TCP behavior, but only on the amount of additional traffic (retransmissions) that TCP injects in the network when packets are lost, hence this result implies also that the network itself becomes instable if the initial operating conditions are beyond the unstable solution of the system represented by (6) and (2).

The upper border of the stability region on the plane (p, g) corresponds to the points $(\bar{p}, \bar{g})$ for which the two solutions

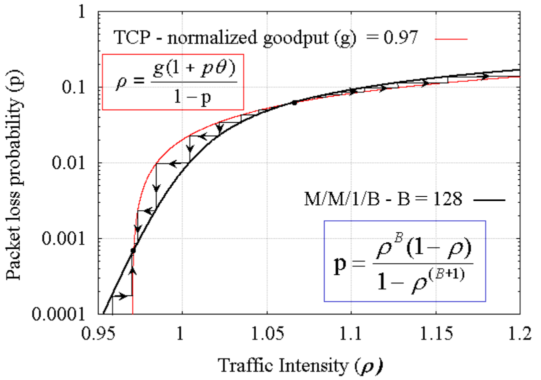

Fig. 6. Example of convergence or divergence of the FPA

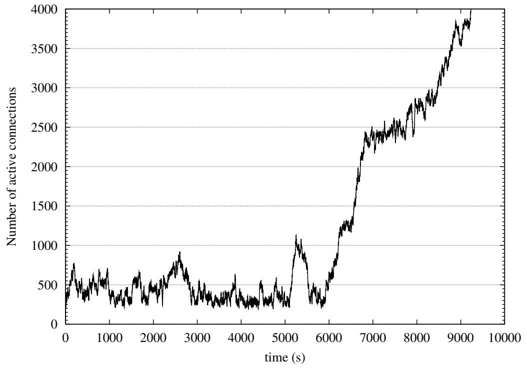

Fig. 7. Simulation trace of the number of connections, $g = 0.986$, $\bar{S} = 20$

$\bar{x}_1$ and $\bar{x}_2$ coincide. This border can be obtained numerically by solving the following system of equations:

$$\begin{cases} h(\rho) = 0 \\ h'(\rho) = 0 \end{cases} \tag{10}$$

where $h(\rho)$ is given by (8). The packet loss probability $\bar{p}$ follows from the solution $(\bar{\rho}, \bar{g})$ using (6).

By varying the value of B^* over the range $[1, B]$, (B is the actual buffer size) we implicitly account for all models of traffic burstiness, so that the curve we obtained is indeed the border of the stability region, independently of the burstiness of the TCP flows (but assuming a constant value for θ). In the case $\theta = 0.1$ the resulting curve is shown in Fig. 5, together with the looser stability bound already obtained using (7). We observe that these curves do not coincide because (7) assumes that the link utilization can be as high as 1, while this is not true, since there always is a non-zero probability that the buffer empties. The solution of (10) accounts for this fact because it incorporates a model of the queue, thus producing a tighter bound for the stability region of the system.

V. Discussion and Conclusions

In this paper we have proved the convergence of the Fixed Point Approximation to obtain the operating point of a bottleneck link loaded by short-lived flows, and we have discovered that there exists a stability region for the behavior not only of TCP models and simulations, but also, under some circumstances, of real-life TCP.

The existence of an unstable region has significant consequences. Even if the system tends to stabilize at the operating point where the traffic intensity is equal to $\bar{\rho}_1$, statistical fluctuations of the traffic load may bring the system beyond the point $\rho = \bar{\rho}_2$. If this happens, the system will diverge. As we mentioned before, this is not only a property of the FPA, but a real case that can be observed by simply running a simulation close to the region of instability. For example, Fig. 7 shows a trace of the number of active connections in the case of average flow size equal to 20 packets and $g = 0.986$. The system keeps itself at the stable operating point for several thousands of seconds, until at time 6000 a traffic peak brings the system out of stability, and the number of active connections begins to increase, quickly drifting toward infinity. It must be noted that this phenomenon requires that the arrival process of new connections is independent from the network load. In real networks users would react to a condition of heavy overload by aborting the existing connections and slowing down the request of new file transfers. Even if human behavior is not considered in this work, our analysis suggests that when the link utilization is close to one, the system operates at a quite unstable point, characterized by high packet loss probabilities (around 10%) and thus rather poor performance is perceived by the users.

Besides, the stability region represented in Fig. 5 clearly defines an upper bound on the sustainable average loss probability. This has major consequences on the design of AQM techniques, since increasing the drop rate beyond the upper bound in the attempt of reducing the queue length may lead the system to instability.

Further work is necessary to extend the analysis to the case of multi-bottleneck scenarios, and to explicitly model the effect of the user behavior in the control loop of the system.

References

[1] "MRTG - the Multi Router Traffic Grapher", `http://people.ee.ethz.ch/~oetiker/webtools/mrtg/`

[2] T. Bu and D. Towsley "Fixed Point Approximation for TCP behavior in an AQM Network," *Proc. of ACM SIGMETRICS 2001,* June 2001

[3] M. Garetto, R. Lo Cigno, M. Meo, E. Alessio, M. Ajmone Marsan, "Modeling Short-Lived TCP Connections with Open Multiclass Queueing Networks," *Proc. of PfHSN 2002*, Berlin, DE, Apr. 22–24, 2002

[4] M. Garetto, D. Towsley, "Modeling, Simulation and Measurements of Queuing Delay under Long-tail Internet Traffic", *Proc. of ACM SIGMETRICS 2003,* pp. 47–57, 2003

[5] S. B. Fredj, T. Bonald, A. Proutiere, G. Régnié, J. Roberts, "Statistical Bandwidth Sharing: A Study of Congestion at Flow Level," *Proc. of ACM SIGCOMM'01*, pp. 111-122, 2001

[6] L. Massoulié and J. W. Roberts, "Arguments in favor of admission control for TCP flows", *Proc. of ITC 16*, Edinburgh, UK, June 7–11, 1999

[7] W. S. Cleveland, D. Lin, D. X. Sun, "IP Packet Generation: Statistical Models for TCP Start Times Based on Connection-Rate Superposition," *Proc. of ACM SIGMETRICS'00*, pp. 166–177, 2000

[8] V. Paxson, S. Floyd, "Wide-Area Traffic: The Failure of Poisson Modeling," *IEEE/ACM Transactions on Networking,* Vol. 3, No. 3, pp. 226–244, June 1995

[9] J. Postel (editor), "Transmission Control Protocol," RFC 793, IETF, 1981

[10] V. Jacobson, M.J. Karels, "Congestion Avoidance and Control," *Proc. of ACM SIGCOMM'88*, Stanford, CA, Aug. 1988

[11] J. Padhye and S. Floyd, "Identifying the TCP Behavior of Web Servers," *Proc. of SIGCOMM'01*, San Diego, CA, USA, Aug. 27–31 2001

[12] M. Mellia, A. Carpani, and R. Lo Cigno, "`Tstat` web page," `http://verza.polito.it/index.html`