

# ANALYSIS OF THE TCP/IP FLOW CONTROL IN HIGH-SPEED WIDE-AREA NETWORKS \*

Eitan ALTMAN, Frédéric BOCCARA, Jean BOLOT and Philippe NAIN  
INRIA, 06902 Sophia-Antipolis Cedex, FRANCE

Patrick BROWN, Denis COLLANGE and Caroline FENZY  
France Telecom - CNET, 06921 Sophia-Antipolis Cedex, FRANCE

## 1 Introduction

We consider the problem of evaluating the performance of TCP (Transmission Control Protocol) over the Internet [3]. Our approach combines experimental and analytic methods, and proceeds in three steps. First, we use measurements taken over the Internet to provide a basis for the chosen analytic model: a TCP connection is modeled by a single node shared with other connections. Non bottleneck nodes are modeled by fixed delays. Second, we consider a single TCP connection sharing a bottleneck node with other connections. The traffic generated by these connections is assumed to be independent from the behavior of the TCP connection under study. We refer to this traffic as *exogenous traffic*, and to these connections as *non-controlled connections*. Third, we consider two TCP connections sharing a bottleneck node. We use fluid models to analyze the behavior of the TCP connections.

## 2 The TCP/IP flow control mechanism

We describe below the Tahoe version of TCP flow control. This is a dynamic window scheme that uses timeouts to detect packet losses. The window size for a connection is the maximum number of unacknowledged packets allowed for this connection at a given time. The control mechanism increases or decreases the window size depending on whether a packet is acknowledged by the destination or is lost (its timeout has expired). Packets are assigned increasing sequence numbers. When it receives a packet, the destination TCP sends an acknowledgement (ack) containing a sequence number indicating the next packet it is waiting for and that all packets with smaller sequence numbers have been correctly received. Packets received out of sequence are buffered but not acknowledged.

The source TCP maintains an estimator of the round-trip time. When it sends a packets, it starts

---

\*This work was supported by France Telecom CNET under Contract 94-5B-012.

a retransmission timer with a timeout equal to the current value of the estimator. If the timer expires and the packet is not yet acknowledged, the packet is considered to be lost. At the source, the window size used to regulate the flow of packets into the network is equal to  $\min\{\text{receive window, congestion window}\}$ . The size of the receive window is defined by the destination and is fixed. Packet losses are used to adjust the size of the congestion window. Throughout, we refer to the congestion window as just the window and we suppose the receive window is always larger than the congestion window.

The TCP window regulation mechanism works in two phases: the *slow-start* (SS) and the *congestion avoidance* (CA) phases. The window is initially set to 1. In the SS phase, it is increased by one every time an ack is received. Therefore, as an ack arrives at the source two packets are generated: one for the received ack and one because the window size is increased by one. This behavior causes a rapid increase of the window size and the amount of data in transit increases rapidly. The SS phase ends when the window reaches a certain level called the *slow-start (SS) threshold* (unless a loss occurs first). At this point the CA phase starts. The purpose of this phase is to slowly increase the utilization so as to adapt to the available bandwidth. This is done by increasing the current size  $W$  of the window by  $1/W$  whenever a packet is acknowledged. This phase ends when a packet is lost. When a loss occurs, the SS threshold is set to half the size of the window, the window is then set to 1, and the cycle restarts.

## 3 Analysis

We model a TCP connection by a single node shared with other connections. This so-called single bottleneck model has been widely used. We have done measurements on Renater [1] and on the Internet [2] which indicate that this model is appropriate in our case. Thus, our reference model is as follows.

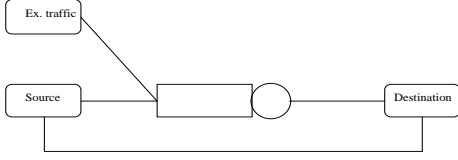


Figure 1: Model for our analysis

We assume that the exogenous stream is a constant rate stream. As already observed in [4], there are two typical types of cyclic behavior; one containing a single SS phase and one containing two SS phases. This turns out to hold also in the presence of exogenous traffic, as shown in Fig. 2 and 3. Yet the computations are more involved, and inter-

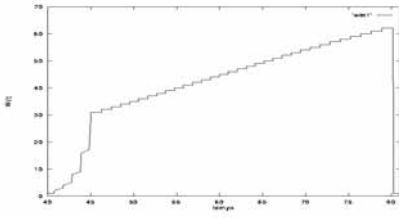


Figure 2: Single slow-start phase

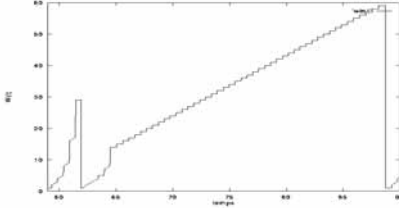


Figure 3: Two slow start phases

esting phenomena arise (Remark 3.1). We analyze both types of behavior and present conditions on the parameters of the network, that indicate when does each type of behavior occur.

### 3.1 Analysis of a single connection with a single slow-start phase

Our approach for analyzing the model is based on a *fluid model*. Similar approaches have been used in [4, 5]. By fluid model we mean that we approximate the dynamics by some averaged flow, and in particular, instead of analyzing the input traffic on the basis of a packet by packet detailed approach, we consider a smoothed inflow where packets are

replaced by a fluid, whose rate is a function of time. Our objective is to compute the average throughput, which we denote by  $\overline{thp}$  (in [1] we also compute the average round-trip delay).

We define a *cycle* as the time interval which starts just after the loss of a packet belonging to the controlled source, given that this loss occurs during the CA phase, till the next time instant that such a loss occurs. During a cycle, the window size drops to one. Furthermore, we note that a cycle always includes at least one SS phase, and that it ends during a CA phase. Let  $C$  denote the duration of a cycle, and let  $N$  be the total number of packets successfully transmitted in a cycle. Then

$$\overline{thp} = N/C. \quad (1)$$

To compute  $N$  and  $C$ , we analyze the dynamic behavior of the window size. Define

$W(s)$  := window size at time  $s$ . We assume  $W(0) = 1$ .

$W_{th}(s)$  := current value of the SS threshold at time  $s$ .

$Q(s)$  := number of packets in the queue (router).

$B$  := buffer size.

$\mu$  := service rate of the queue.

$\tau_1$  := time between the transmission of a packet until it reaches the queue.

$\tau_2$  := time between the departure of a packet from the queue till it reaches the destination.

$\tau = 2(\tau_1 + \tau_2)$  := round-trip delay when the queue is empty (i.e., the time between the transmission of a packet until its ack is received), not including the service time.

$\lambda$  := rate at which exogenous packets are transmitted. We assume that  $\lambda < \mu$ .

$T := \tau + \mu^{-1}$  “sojourn time” of a packet in an empty system, i.e., the round-trip delay plus the service time  $\mu^{-1}$ .

$\beta := B/[(\mu - \lambda)\tau]$  is a normalized buffer size [4].

$W_{max}$  := maximal size attained by the window at the end of the CA phase (before a loss occurs).

We assume that  $W$ ,  $W_{th}$  and  $Q$  are right continuous. When it is clear, we omit the argument  $s$  in quantities such as  $W$  and  $Q$ .

#### Theorem 1

$$W_{max} = (B/\mu + T)(\mu - \lambda) + \lambda/\mu, \quad (2)$$

$$W_{th} = W_{max}/2,$$

$$\overline{thp} = \frac{N_1 + N_2 + N_3}{T_1 + T_2 + T_3}, \quad (3)$$

where  $N_i$  and  $T_i$  ( $i = 1, 2, 3$ ) are given by the following: (i) If  $T(\mu - \lambda) < W_{th}$  then

$$T_1 = T \log(T(\mu - \lambda)), \quad N_1 = T(\mu - \lambda) - 1, \quad (4)$$

$$T_2 = \frac{W_{\text{th}} - T(\mu - \lambda)}{\mu - \lambda/2}, \quad N_2 = W_{\text{th}} - T(\mu - \lambda), \quad (5)$$

$$T_3 = \frac{W_{\text{max}}^2 - W_{\text{th}}^2}{2(\mu - \lambda)}, \quad N_3 = \frac{W_{\text{max}}^2 - W_{\text{th}}^2}{2}. \quad (6)$$

(ii) If  $T(\mu - \lambda) \geq W_{\text{th}}$  then

$$T_1 = T \log(W_{\text{th}}), \quad N_1 = W_{\text{th}} - 1, \quad (7)$$

$$T_2 = T(T(\mu - \lambda) - W_{\text{th}}), \quad N_2 = \frac{T_2^2}{2T^2} + \frac{W_{\text{th}}T_2}{T}, \quad (8)$$

$$T_3 = \frac{W_{\text{max}}^2 - (T(\mu - \lambda))^2}{2(\mu - \lambda)}, \quad (9)$$

$$N_3 = W_{\text{max}}^2 - (T(\mu - \lambda))^2/2.$$

**Proof:** We first discuss the evolution of the window size over time. If an ack arrives at time  $s$ , then

$$W(s) = \begin{cases} W(s^-) + 1 & \text{if } W(s^-) < W_{\text{th}}(s^-) \\ W(s^-) + 1/\lfloor W(s^-) \rfloor & \text{otherwise} \end{cases} \quad (10)$$

where  $\lfloor \cdot \rfloor$  denotes the integer part of the argument. If a loss is detected at time  $s$ , then,  $W_{\text{th}}(s) = W(s^-)/2$  and  $W(s)$  is set to one. Define  $thp_{in}(s) :=$  input rate of fluid originating from the controlled source.

$\lambda_{in}(s) = \lambda :=$  input rate of fluid originating from the exogenous sources.

$thp_{out}(s) :=$  rate of fluid originating from the controlled source, at the output of the queue.

$\lambda_{out}(s) :=$  rate of fluid originating from the exogenous sources, at the output of the queue.

Note that the rate at which acks arrive is equal to  $thp_{out}$ , as long as there is no loss. The total number of packets  $N$  transmitted successfully during a cycle can be expressed as  $N = \int_0^C thp_{out}(s) ds$ . To obtain  $thp_{out}(s)$  we will compute

$$\frac{dW}{dt} = \frac{dW}{dack} \frac{dack}{dt} = \frac{dW}{dack} thp_{out} \quad (11)$$

where  $dW/dt$  is the rate at which the window grows as a function of time. From (10) we get

$$\frac{dW}{dack} = \begin{cases} 1 & \text{if } W < W_{\text{th}} \\ W^{-1} & \text{if } W \geq W_{\text{th}} \end{cases} \quad (12)$$

so that

$$\frac{dW}{dt} = \begin{cases} thp_{out} & \text{if } W < W_{\text{th}} \\ thp_{out}/W & \text{if } W \geq W_{\text{th}}. \end{cases} \quad (13)$$

As long as the queue is empty, we have

$$thp_{out}(t) = thp_{in}(t) = W(t)/T \quad (14)$$

and  $\lambda_{out} = \lambda$ . When it starts building up, then  $thp_{out}(t) + \lambda_{out} = \mu$ . Hence, it follows that the queue starts building up when  $W$  reaches the level  $W(t) = T(\mu - \lambda)$ . When the queue is nonempty, the output rates of both the controlled as well as the exogenous traffic are smaller than the input rates. It is reasonable to assume then that the output rates are proportional to the input rates. Thus,

$$thp_{out}(t) = \frac{\mu thp_{in}(t)}{thp_{in}(t) + \lambda}, \quad \lambda_{out}(t) = \frac{\mu \lambda}{thp_{in}(t) + \lambda}. \quad (15)$$

So,  $thp_{in}(t) = thp_{out}(t)\lambda/(\mu - thp_{out})$ . Another equation that relates the input and output rates of the controlled traffic is obtained by noting that the input rate is the sum of the output rate and the rate at which the window size increases. By using the relation

$$thp_{in}(t) = \frac{dW}{dt} + \frac{dack}{dt} = \left(1 + \frac{dW}{dack}\right) thp_{out}(t)$$

we thus obtain

$$thp_{out}(t) = \mu - \lambda \left(1 + \frac{dW}{dack}\right)^{-1}.$$

Assuming that  $W^{-1} \ll 1$  during the CA phase, we obtain the following simple expression for  $thp_{out}(t)$  when the queue is nonempty:

$$thp_{out}(t) = \begin{cases} \mu - \lambda/2 & \text{during the SS phase} \\ \mu - \lambda & \text{during the CA phase} \end{cases} \quad (16)$$

The above behavior of the throughput (16) is discussed in Remark 1. (When the queue is empty, the throughput is given by (14).) Combining the above with (11), (12) and (13), we get

$$(i) \text{ If } T(\mu - \lambda) < W_{\text{th}} \quad (17)$$

$$\frac{dW}{dt} = \begin{cases} W/T & \text{if } W \leq T(\mu - \lambda) \\ \mu - \lambda/2 & \text{if } T(\mu - \lambda) < W < W_{\text{th}} \\ \mu - \lambda/W & \text{if } W \geq W_{\text{th}}. \end{cases}$$

$$(ii) \text{ If } T(\mu - \lambda) \geq W_{\text{th}} \quad (18)$$

$$\frac{dW}{dt} = \begin{cases} W/T & \text{if } W \leq W_{\text{th}} \\ 1/T & \text{if } W_{\text{th}} < W < T(\mu - \lambda) \\ (\mu - \lambda)/W & \text{if } W \geq T(\mu - \lambda). \end{cases}$$

Let  $\tilde{t}$  be the instant at which  $W_{\text{max}}$  is reached. The queue is then full, so that the number of packets at time  $\tilde{t}$  originating from the controlled source in the queue is  $B thp_{in}(\tilde{t})/(thp_{in}(\tilde{t}) + \lambda)$ .  $W_{\text{max}}$  is then obtained through

$$W_{\text{max}} - \tau_1 thp_{in}(\tilde{t}) - (\tau_2 + \tau/2) thp_{out}(\tilde{t}) - 1 = B thp_{in}(\tilde{t})(thp_{in}(\tilde{t}) + \lambda)^{-1}.$$

Since by (15) and (16) we have  $thp_{in}(\tilde{t}) = thp_{out}(\tilde{t}) = \mu - \lambda$ , and (2) follows. When the queue is nonempty, the number of controlled packets in it is given by

$$W(t) - \tau_1 thp_{in}(t) - (\tau_2 + \frac{\tau}{2}) thp_{out}(t) - 1 = Q(t) thp_{in}(t) (thp_{in}(t) + \lambda)^{-1}.$$

From (17) and (18) we may conclude that there are three periods within a cycle: one in which the window increases exponentially fast, the second when it grows linearly, and the third, in which it grows sublinearly. This will be made more precise below. Define

$T_i$ : duration of the  $i$ th such period,  $i = 1, 2, 3$ .  
 $t_i$ : time at which the  $i$ th period ends. We assume that  $t_0 = 0$ .  
 $N_i$ : number of packets transmitted in period  $i$ .

Clearly  $\overline{thp} = (N_1 + N_2 + N_3)/(T_1 + T_2 + T_3)$  (see (1)), with

$$N_i = \int_{t_{i-1}}^{t_i} thp_{out}(s) ds. \quad (19)$$

Let us compute  $T_1$  and  $N_1$  in the case that  $T(\mu - \lambda) \leq W_{th}$ . Integrating (17) and using the condition  $W(0) = 1$  yields  $W(t) = \exp(t/T)$  as long as the queue is empty. Since the queue starts to build up at time  $T_1$  we have

$$e^{t/T} = W(T_1) = T(\mu - \lambda) \quad (20)$$

so that  $T_1 = T \log(T(\mu - \lambda))$ . Combining now (14) and (20) yields  $thp_{out}(t) = \exp(t/T)/T$  for  $0 < t < T_1$  so that  $N_1 = T(\mu - \lambda)$  from (19).

The derivation of expressions for  $T_1$  and  $N_1$  in the case when  $T(\mu - \lambda) > W_{th}$  is similar to that shown above, as are the derivations for the expressions of  $T_2, T_3, N_2$ , and  $N_3$ . ■

**Remark 1** One of the interesting conclusions from the above analysis is the dynamic behavior of the throughput. From (16) we observe that:

(1) The throughput has a discontinuity when we pass from the SS to the CA phase. This should not be surprising, since there is a discontinuity in the behavior of the window mechanism at that instant.

(2) During the CA phase, the exogenous traffic is seen to behave as if it had full priority over the controlled traffic. Indeed, its throughput is equal to  $\lambda$ , i.e. to the input rate of exogenous traffic. We have observed this behavior in experimentations. That in SS we do not have this effect can

be explained by the fact that at every arrival of an ack, two consecutive controlled packets are transmitted, so that the controller is more “aggressive” in using the available bandwidth at the expense of the exogenous traffic. The fact that during the CA phase the controlled traffic gives up bandwidth to the non-controlled source can be seen as a drawback of TCP/IP. However, this property is interesting when the non controlled traffic is audio and/or video traffic. In this case, we have a natural priority mechanism which in a sense gives priority to the voice and video most of the time (since the CA avoidance phases are typically much longer than the SS phases).

**Remark 2** We have made the assumption in our analysis that losses are detected very soon after they occur. In practice, this can be justified by the mechanism of negative acks, where at every arrival of a packet, the packet that is acknowledged is the last one to have arrived in sequence, and not the current one which just arrived to the destination. If the same packet is acknowledged three consecutive times, TCP/IP understands it to correspond to a loss of a packet. In some applications, this feature of TCP/IP is not implemented. In those cases, the only way to detect a loss is through the expiration of the retransmission timer. The length of a cycle is then approximately  $T_1 + T_2 + T_3 + rto$ , where  $rto$  is the maximum value of the timer (a typical value of  $rto$  is 250msec). On the other hand, a whole window of size  $W_{max}$  can still be transmitted after the loss, and the packets often need not be retransmitted, if they are stored at the destination. Typically, as observed in experimentations, there are no additional losses after the first loss. This can be explained by the fact that a loss occurs when the window size increases, and then two consecutive packets are transmitted one after the other. Thus, losses in the CA avoidance phase occur typically at those “bursts” of two packets, and not between such bursts. (In the CA avoidance phase, many “non-bursty” packets may be transmitted between such bursts). Hence, in the absence of detection of losses through negative acks, we have instead of (3):  $\overline{thp} = (N_1 + N_2 + N_3 + W_{max})/(T_1 + T_2 + T_3 + rto)$ .

We next illustrate the utilization of the above analytical results. We consider the fraction of the available throughput used by TCP, i.e.  $thp/(\mu - \lambda)$ . This quantity indicates how well TCP uses the residual capacity left by the exogenous traffic. In an ideal situation, it should be close to one. A value far below one indicates link underutilization

(as we see is the case when  $b$  is small in figure 5), while a value (even slightly) above one indicates that the TCP connection does not allow for the exogenous traffic to flow. In all cases, we take the unit of time to be one bottleneck queue service time.

Fig. 4 shows the fraction of the available throughput used by TCP as a function of the round-trip time  $\tau$  for different values of  $\lambda$  when the buffer size is  $B = 20$ . For small values of  $\tau$ , TCP uses a higher percentage of the available throughput for higher exogenous traffic rates, even exceeding a ratio of one. For large values of  $\tau$  the opposite is true. In all cases this ratio decreases with  $\tau$  and this at a rate which increases with the exogenous traffic intensity.

In Fig. 5,  $\tau$  is set to 30 while the buffer size varies. The fraction of the available throughput used by TCP is shown for different values of  $\lambda$ . As we saw was the case for high values of  $\tau$ , TCP uses more effectively the available capacity for smaller values of  $\lambda$ . Furthermore, we observe that this is true independent of the buffer size  $B$ . Also, we note that the TCP throughput never exceeds 1. However, for low values of  $B$ , the underutilization can be as high as 30%.

### 3.2 Analysis of a single connection with two SS phases

A second type of periodic behavior observed in simulations was a cycle containing two subcycles, where the first one consists of a single SS phase, and the second consists of three periods, as described in the previous subsection. A more detailed analysis than the fluid model was necessary to describe this behavior [1]. This analysis allowed us to

(i) see when is a loss in a SS phase possible. We showed that the (necessary and sufficient) condition for having such a loss during a SS phase is that  $W_b$  is less than or equal to the SS threshold  $W_{th}$  computed in (2), where  $W_b = (2\mu - \lambda)B/\mu$ . This condition is equivalent to

$$\beta \leq (3 - \lambda/\mu)^{-1}. \quad (21)$$

If the condition is not satisfied, then the approach and calculations used in the previous subsection provides a good description of the dynamics.

(ii) when there are two SS phases, it allowed us to predict at what value of the window size  $W$  and at what time will the loss occur.

The rest of the analysis, for describing the dynamics within each period in the second sub-cycle,

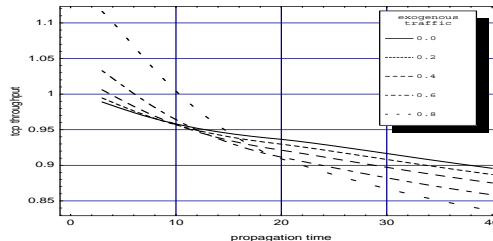


Figure 4: Available throughput utilization vs. propagation time for  $B = 20$  and different  $\lambda$

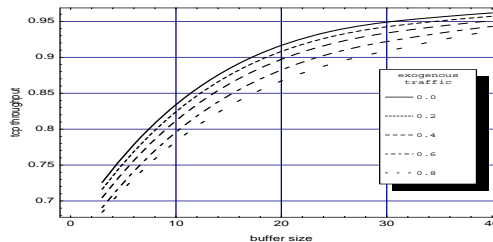


Figure 5: Available throughput utilization vs. buffer size for  $\tau = 30$  and different  $\lambda$

as well as the dynamics in the SS phase of the first sub-cycle, were obtained in [1] by a fluid approach similar to the one mentioned in the previous subsection.

### 3.3 Analysis of two interacting connections

We consider the model depicted in Fig. 6. When several controlled sources share the same bottleneck node, we observed in general quite a chaotic aperiodic behavior. The maximum window size before a loss occurs was also varying in an acyclic way. Such a behavior, as obtained by simulations, is depicted in Fig. 7. In some cases simulations exhibited a cyclic behavior. Using a fluid approach, we were able to analyze the latter case for two controlled sources [1]. This was done by assuming that packets belonging to both sources are lost when

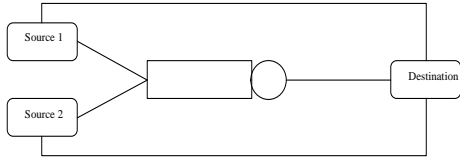


Figure 6: Two controlled sources

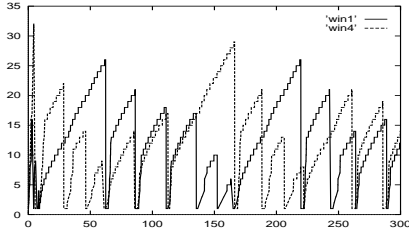


Figure 7: Two controlled sources: a non-synchronized behavior

the queue is full. When this assumption holds, then the window size of both sources will drop to one in a synchronized way, i.e., shortly after the queue is filled. This will result in a relatively simple cyclic behavior, where the window sizes of both sources have the same cycle duration. This “full synchronization” assumption typically holds if the rate at which packets are sent (the throughput) both sources are considerably higher than the service rate, just before the queue is full. An example of such behavior is depicted in Fig. 8. An impor-

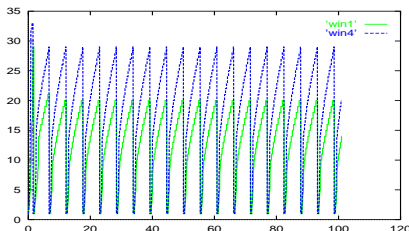


Figure 8: Two controlled sources: a synchronized behavior

tant feature in the case of competing controlled sources is that the information delay is different for both sources; the closer a source is to the destination, the shorter is the round-trip delay, and thus the acks come back faster, so that the window grows faster. This argument shows that sources that are further apart from the destination adapt slower to the available bandwidth, and thus, will get a smaller share of the available bandwidth [4].

## 4 Numerical results

The following calculations and simulation results were obtained in the case of exogenous uncontrolled traffic of rate  $\lambda$ .  $\lambda$  and  $\mu$  are given in packets/sec, where a packet contains 576 bytes, i.e. 4608 bits.  $\tau$  is given in seconds.

Parameters				thp			$\overline{rtt}$		
B	$\mu$	$\lambda$	$\tau$	Anal.	Sim.	%Err.	Anal.	Sim.	%Err.
10	166	66	0.5	57.4	58.4	1.7	0.47	0.50	5.8
20	80	32	1.0	39.2	38.5	1.8	1.07	1.08	1.0
10	125	100	2.0	13.0	12.9	0.7	1.93	2.03	5.0
20	200	160	1.0	29.4	29.7	1.0	1.02	1.03	1.0

The router was assumed to be equidistant from the source and destination ( $\tau_1 = \tau_2$ ). The above 4 cases validate the criterion (21) for deciding whether a single SS phase or two SS phases will occur in a cycle. In cases 1 and 3 we have  $W_b < W_{\max}/2$ , and two SS phases occur, where as in cases 2 and 4, which satisfy  $W_b > W_{\max}/2$ , there is only a single SS phase per cycle. According to the simulations, our fluid model approximates well the average round-trip delay in both regimes (errors less than 6%), and even better the average throughput (errors less than 2%). The precision is even better for the case of a single SS per cycle, where only 1% error was obtained for the average round-trip delay.

**Acknowledgement:** The authors wish to thank M. D. Elouadghiri for helpful discussions.

## References

- [1] E. Altman, F. Baccara, J. Bolot, P. Nain, P. Brown, D. Collange and C. Frenzy, “Performance of TCP/IP over the French Research Network: Measurements and Analysis”, manuscript.
- [2] J.-C. Bolot, “End-to-end delay and loss behavior in the Internet”, *Proc. ACM Sigcomm '93*, 289-298, San Francisco, CA.
- [3] V. Jacobson, “Congestion avoidance and control”, *ACM Sigcomm '88*, Stanford, CA, 314-329.
- [4] T. V. Lakshman and U. Madhow, “Window-based congestion control for networks with bandwidth-delay products and random loss: a study of TCP/IP performance”, *Proc. HPN '94*, 133-147, Grenoble, France.
- [5] S. Shenker and L. Zhang, “Some observations on the dynamics of a congestion control algorithm”, *Computer Communication Review*, 30-39, 1990.