Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Congestion Control and Sizing Router Buffers in the Internet

Saverio Mascolo and Francesco Vacirca

Abstract— **The Internet is made of communication links and packet switching nodes named routers. Routers are equipped with buffers that hold packets during congestion and feed output links with packets during underutilization. A rule largely known in literature is the "bandwidth-delay rule", which states that, in order to guarantee full link utilization, it is necessary to provide each link with a buffer $B = RTT \cdot C$, where RTT is the round trip time and C is the link capacity. The bandwidth delay rule requires buffer size that increases linearly with link capacity. With the recent introduction of 10 Gbps Routers and Ethernet cards, buffer requirements as dictated by the bandwidth-delay rule become extremely large. For instance, a 10Gbps router link with a RTT of 200ms would require a 2Gbits buffer size, which is a challenging requirement for manufacturers. Moreover, such a large buffers would introduce large and time-varying queuing delays that are harmful for time sensitivity traffic such as audio and video. In this paper we investigate the relation between the TCP congestion control and the buffer size required to guarantee full link utilization. We consider two TCP congestion control algorithms: the standard TCP NewReno and the recently proposed TCP Westwood+. Analytical results show that while classic TCP Reno/NewReno requires buffer of order size $\sim 1/\sqrt{n}$, where n is the number of coexisting flows, Westwood+ TCP, in principle, can provide full link utilization for any buffer size. Discrete event simulations and experiment on real 10 Gigabit per second wide area network confirm theoretical results.**

I. Introduction

The Internet is made of communication links connected by switching nodes (i.e. routers), which implement store-and-forward packet switching. Router buffers are important for storing packets during link congestion and for supplying packets during link underutilization. When buffers overflow they cause packet loss, when buffers underflow they cause link underutilization. Buffers provoke time-varying queuing delays that are one of the major sources of uncertainty in the Internet. By quoting [1], "Given the significance of their role, we might reasonably expect the dynamics and sizing of router buffers to be well understood, based on a well-grounded theory, and supported by extensive simulation and experimentation. This is not so. Router buffers are sized today based on a rule-of-thumb commonly attributed to a 1994 paper by Villamizar and Song [2]. Using experimental measurements of at most eight TCP flows on a 40 Mb/s link, they concluded that - because of the dynamics of TCP's congestion control algorithm- a router needs an amount of buffering equal to the average round-trip time of a flow that passes through the router, multiplied by the capacity of the router's network interface. This is the well-known $B = RTT \cdot C$ rule. Requiring such large buffers complicates router design. For example, a 10Gb/s router linecard needs approximately $250ms \cdot 10Gb/s = 2,5Gbits$ of buffers, and the amount of buffering grows linearly with the line-rate". Starting from the paper [3] and the significant contribution in [1], the topic on sizing router buffers has risen an increasing interest ([4] and [5]).

The scope of this paper is to investigate the relation between buffer requirements and the TCP congestion control. In particular we will derive and compare buffer requirements when using standard Reno/NewReno TCP [6] or Westwood+ TCP [7], [8], which is a recent innovation of classic TCP congestion control as proposed by Van Jacobson. The paper is organized as follows: Section II provides background on TCP congestion control; Section III analyzes buffer requirements in the case of a single flow; Section IV investigates buffer requirements in the case of large statistical multiplexing; Section V reports simulations and experimental results, whereas Section VI draws the conclusions.

Saverio Mascolo (mascolo@poliba.it), DEE Politecnico di Bari, via Orabona 4, 70125 Bari, Italy.

Francesco Vacirca (vacirca@infocom.uniroma1.it), Infocom Department, University of Roma "La Sapienza", Via Eudossiana 18, 00184 Rome, Italy.

II. Background on TCP congestion control

In this Section we briefly summarize main features of the TCP congestion control. In particular, we will consider the standard TCP (i.e. Reno/NewReno TCP) and Westwood+ TCP that has been recently proposed in [7], [8] and implemented in Linux [9]. The goal is to analyze and compare buffer requirements using different TCP congestion control algorithms. TCP congestion control is essentially made of a probing phase and a decreasing phase. The probing phase of standard TCP and Westwood+ TCP are identical: it consists of an exponential growing phase (i.e. the slow-start phase) and of a linear increasing phase (i.e. the congestion avoidance phase). The probing phase stops when congestion is experienced in the form of 3 duplicate acknowledgments or a timeout. At this point the behavior of standard TCP and Westwood+ TCP are different. Reno/NewReno TCP implements a multiplicative decrease behavior, whereas Westwood+ TCP implements a decrease behavior based on the estimate of the available bandwidth at the time of congestion. The pseudo-code of Reno/NewReno TCP is:

a) On ACK reception:
   - $cwnd$ is increased accordingly to the Reno algorithm;

b) When 3 DUPACKs are received:
   - $ssthresh = \max(2, cwnd/2)$;
   - $cwnd = ssthresh$;

0-7803-9568-9/05/$20.00 ©2005 IEEE

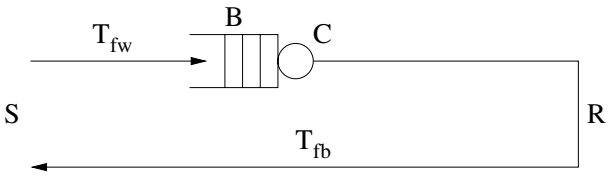

Fig. 1. Schematic of a TCP connection.

c) When coarse timeout expires:
- $ssthresh = 1$;
- $cwnd = 1$;

Regarding Westwood+ TCP, the main idea is to set the control windows after congestion such that the bandwidth available at the time of congestion is exactly matched (see [7] and [8]). The available bandwidth is estimated by properly counting and averaging the stream of returning ACK packets. In particular, when three DUPACKs are received, both the congestion window ($cwnd$) and the slow start threshold ($ssthresh$) are set equal to the estimated bandwidth (BWE) times the minimum measured round trip time (RTT_m); when a coarse timeout expires the $ssthresh$ is set as before while the $cwnd$ is set equal to one. The pseudo code of the Westwood+ algorithm is as simple as reported below:

a) On ACK reception:
- $cwnd$ is increased accordingly to the Reno algorithm;
- an estimate BWE of the available bandwidth is computed;

b) When 3 DUPACKs are received:
- $ssthresh = \max(2, (BWE \cdot RTT_m)/seg\_size)$;
- $cwnd = ssthresh$;

c) When coarse timeout expires:
- $ssthresh = \max(2, (BWE \cdot RTT_m)/seg\_size)$;
- $cwnd = 1$;

III. TCP CONGESTION CONTROL AND BUFFER REQUIREMENTS

The goal of this section is to investigate the buffer size that is required in order to provide full link utilization given a used TCP congestion control algorithm. We start by considering the case of a single TCP connection. The schematic of a single TCP connection is shown in Figure 1, where B is the bottleneck buffer size, C is the link service rate, T_{fw} is the propagation delay from the TCP sender S to the bottleneck buffer, T_{fb} is the propagation delay from the bottleneck buffer to the TCP receiver R and then back to the sender.

We have seen that the standard TCP congestion control algorithm is made of a probing phase that increases the input rate up to fill the buffer and hit network capacity. At that point packets start to be lost and the receiver sends duplicate acknowledgments. After the reception of three duplicate acknowledgments, the sender infers network congestion and reduces the congestion window by half. Let t_1 be the time when 3 duplicate acknowledgments are received by the sender. The congestion window W, i.e. the number of outstanding packets, at time t_1 is then equal to:

$$W(t_1) = B + C \cdot RTT_m + \epsilon$$

where $RTT_m = T_{fw} + T_{fb}$ is the minimum round trip propagation time and ϵ are the lost packets. At time t_1^+ the congestion window is reduced to $W(t_1)/2$ and the sender stops until it gets acknowledgments for $W(t_1)/2$ packets. Since the link service rate is C, the sender will stop for the interval $W(t_1)/2C$. Therefore, in order to get full link utilization, the buffer depletion time B/C plus the time T_{fw} to deal with the packets that are in flight between the sender and the bottleneck router at the moment when the sender stops must be greater or equal to the stop interval $W(t_1)/2C$ plus the forward propagation delay T_{fw}:

$$\frac{B}{C} + T_{fw} \geq \frac{W(t_1)}{2C} + T_{fw} = \frac{B}{2C} + \frac{RTT_m}{2} + \frac{\epsilon}{2C}$$

which turns out:

$$B \geq C \cdot RTT_m + \epsilon \tag{1}$$

The rule (1) states that, in order to get full link utilization, a buffer greater than the bandwidth delay product $C \cdot RTT_m$ is required. The rule (1) is the standard "rule of thumb" reported in [1]. Now we will invoke analogous arguments to investigate buffer requirements when using Westwood+ TCP. Again, by letting t_1 be the time when 3 duplicate acknowledgments are received by the sender, the number of outstanding packets at time t_1 is:

$$W(t_1) = B + C \cdot RTT_m + \epsilon$$

At time t_1^+, the congestion window is now reduced to

$$W(t_1^+) = C \cdot RTT_m$$

because the available bandwidth BWE is equal to C. In this case the sender will stop until it gets acknowledgments for

$$W(t_1) - W(t_1^+) = B + \epsilon$$

packets. Again, since the link service rate is C, the sender will stop for the interval $(B + \epsilon)/C$. Therefore, in order to get full link utilization, the buffer depletion time B/C must be greater or equal to the stop interval $(B + \epsilon)/C$:

$$\frac{B}{C} \geq \frac{B + \epsilon}{C} \tag{2}$$

The inequality (2) is satisfied only when ϵ is equal to zero. To discuss and compare results (1) and (2) we compute the "idle time" id_T that is the time during which the link is idle. The idle time in the case of standard TCP is equal to:

$$id\_T = \left[\frac{RTT_m}{2} + \frac{\epsilon}{2C} - \frac{B}{2C}\right]^+ \tag{3}$$

where $[x]^+ = \max(0, x)$. In this case the idle time is zero if inequality (1) is satisfied. Otherwise, the idle time increase from zero, when (1) is satisfied, up to $\frac{RTT_m}{2} + \frac{\epsilon}{2C}$ when B is zero. In the case of Westwood+ TCP the idle time is:

$$id\_T = \frac{\epsilon}{C} \tag{4}$$

which has the nice property that is small and constant for any value of the buffer size B. This property plays an important role in the case of Gbps links where, due to technological issues [1], the sizing buffer rule (1) cannot be satisfied and buffer are under provisioned.

IV. Buffer sizing in the case of large statistical multiplexing

In this Section we investigate the issue of sizing router buffers in the presence of a very large number of flows that shares the bottleneck.

A. The case of synchronized flows

We have seen that TCP congestion control is characterized by an increasing/decreasing behavior of the congestion window that looks like a "saw-tooth". It is well known that it can happen that many TCP flows experience congestion roughly at the same time so that each TCP congestion window is reduced at roughly the same time. Flows in this condition are said to be synchronized ([1], [11]). The aggregated window size process (i.e. the sum of all the window size process) looks like an amplified version of the single flow. Thus, also in this case the buffer size needs to satisfy the inequality (1) in order to provide full link utilization when Reno/NewReno TCP is used.

B. The case of desynchronized flows

Investigation of conditions for having desynchronized flow is out of the scope of this work. We only refer to studies showing that the absence of synchronization has been shown in real networks ([1], [12]), and that small differences in RTT are sufficient to prevent synchronization ([1], [13]).

In [1] it has been observed that, assuming desynchronized TCP flows so that each window size process W_i is independent of each other, the sum of window sizes $W = \sum_i W_i$ is a bounded random process made of the sum of independent saw-tooth window processes. From the central limit theorem, the sum process will converge to a gaussian process. Since the relation between the queue and the window size W is:

$$Q(t) = \sum_{i=1}^{n} (W_i - C_i \cdot RTT_{m_i}) - \epsilon$$

(i. e. the outstanding packets are in the queue, in the link pipe or are dropped), also the queue has a gaussian distribution. In particular, by neglecting ϵ which is always reasonable if TCP is operating correctly and the buffer is large enough, the relation between Q and W is:

$$Q(t) = W(t) - C \cdot \overline{RTT}_m \tag{5}$$

where $C \cdot \overline{RTT}_m = \sum_i C_i \cdot RTT_{m_i}$. Eq. (5) means that $Q(t)$ inherits the gaussian distribution of $W(t)$ (i.e. the distribution of W shifted by a constant). The relation between averages is:

$$\overline{Q(t)} = \overline{W(t)} - C \cdot \overline{RTT}_m \tag{6}$$

Each TCP Reno/NewReno window process can be modeled as a uniform distribution around the average $\overline{W}_i$ with minimum $(2/3)\overline{W}_i$ and maximum $(4/3)\overline{W}_i$. By recalling that the standard deviation of a uniform distribution is $1/\sqrt{12}$-th of its length, the standard deviation of the single window process W_i is

$$\sigma_{W_i} = \frac{1}{3\sqrt{3}}\overline{W}_i$$

Since $\overline{W}_i = \overline{W}/n$, and for a large number of flows it holds:

$$\sigma_W \leq \sqrt{n} \cdot \sigma_{W_i}$$

the standard deviation of $Q(t)$ is:

$$\sigma_Q = \sigma_W \leq \sqrt{n} \cdot \sigma_{W_i} = \frac{\sqrt{n}}{3\sqrt{3}}\overline{W}_i = \frac{\sqrt{n}}{3\sqrt{3}}\frac{\overline{W}}{n} = \frac{\overline{W}}{3\sqrt{3n}} \tag{7}$$

By considering Eq. (6),

$$\overline{W(t)} = \overline{Q(t)} + C \cdot \overline{RTT}_m \leq B + C \cdot \overline{RTT}_m$$

the Eq. (7) becomes:

$$\sigma_Q = \sigma_W \leq \frac{B + C \cdot \overline{RTT}_m}{3\sqrt{3n}} \tag{8}$$

Considering that the link is fully utilized when the buffer is not empty and when it does not overflow, the utilization can be computed as the probability that $W(t)$ belongs to the interval between $(\overline{W} - B/2, \overline{W} + B/2)$, that is:

$$Utilization = \text{erf}\left(\frac{B}{2\sqrt{2}\sigma_W}\right)$$

In the case of TCP Reno/NewReno, by considering that the standard deviation of the TCP window process is upper bounded by Eq. (8), the utilization can be lower bounded as follows:

$$Utilization_{RENO} \geq \text{erf}\left(\frac{3\sqrt{3}B}{2\sqrt{2}\frac{B + C \cdot \overline{RTT}_m}{\sqrt{n}}}\right) \tag{9}$$

In the case of TCP Westwood+, similar arguments can be applied to derive a lower bound for the link utilization.
Assuming that the buffer is equally shared between all the flows, the TCP window process W_i oscillates between $C_i \cdot RTT_{m_i}$ and $C_i \cdot RTT_{m_i} + B/n$. The process can be modeled as a random variable with uniform distribution and standard deviation equal to:

$$\sigma_{W_i} = \frac{1}{\sqrt{12}}(C_i \cdot RTT_{m_i} + B/n - C_i \cdot RTT_{m_i}) = \frac{1}{\sqrt{12}}\frac{B}{n}.$$

Following same reasoning that has been used to derive Eq. (9), it is possible to show that in case of TCP Westwood+ the link utilization is lower bounded by:

$$Utilization_{WEST} \geq \text{erf}\left(\sqrt{\frac{3}{2}} \cdot n\right) \tag{10}$$

By comparing the lower bound utilization obtained for TCP Reno/NewReno and Westwood+ (i.e. Eq. (9) and Eq. (10) respectively), it is possible to notice that, similarly to the case of synchronized flows scenario, the link utilization of Reno/NewReno TCP congestion control depends on the buffer size, whereas in the Westwood+ case, it does not depend on the buffer capacity B.

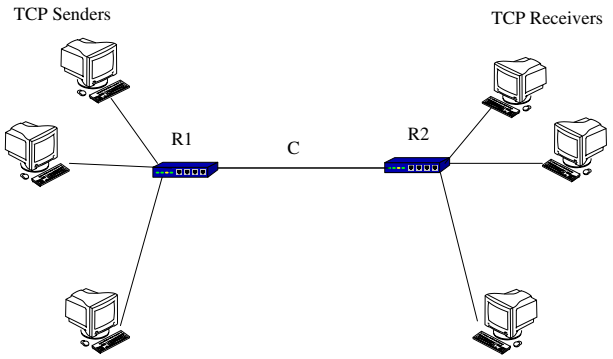

Fig. 2. Simulation scenario.

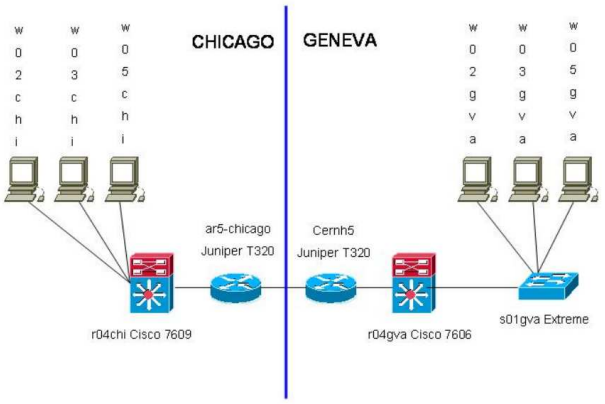

Fig. 3. Experiment scenario.

V. NS2 SIMULATION RESULTS AND INTERNET EXPERIMENTS

To confirm the mathematical analysis developed in previous sections, in this section we report both computer simulation results and real Internet experiments. In particular we reports simulation results collected using the ns2 network simulator [14] and one experiment we have done over a real Gigabit per second path between the CERN in Geneva and the Fermi Lab in Chicago. Currently we are collecting other measurements using the 10Gbps Datatag network [15]. Figure 2 shows the reference scenario employed for collecting computer simulation results. It consists of n TCP Senders establishing a connection with n TCP Receivers. The bottleneck link between the R1 and R2 routers is provisioned with capacity C and has a round trip time propagation time equal to RTT. Propagation delays of all other links are uniformly distributed within the interval (0, 10ms) in order to desynchronize flows. Links between router R1 and TCP senders and links between router R2 and TCP receivers are provisioned with a capacity much greater than C so that the link with capacity C is the bottleneck for each TCP flow.

The reference scenario for the experiments is depicted in Figure 3. It consists of a 10 Gbps connection going from the CERN Lab in Geneva, to the Fermi Lab in Chicago. The link between the Cisco router 7606 and the Extreme router

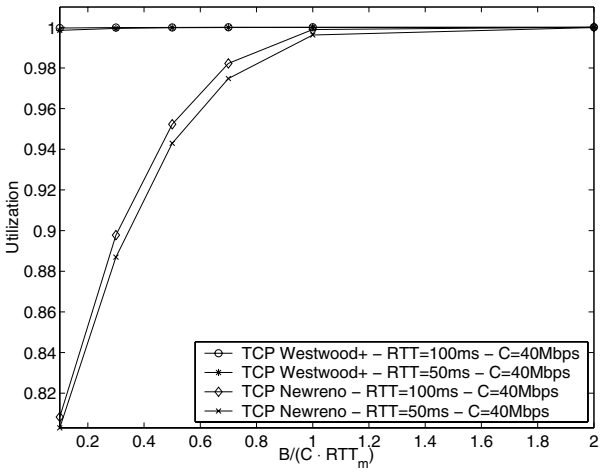

Fig. 4. Link utilization as a function of the bottleneck buffer size in the case of a single TCP source.

$s01gva$ at Geneva is set at 1 Gbps.

A. The case of synchronized flows

The first issue we investigate is the behavior of a single TCP flow when using buffers with different size.

Figure 4 shows the bottleneck link utilization in the case of C=40Mbps and RTT_m equal to 100 ms and 50 ms. Results confirm analysis developed in previous sections, that is, the link utilization obtained using TCP Westwood+ does not depend on the bottleneck buffer size and reaches the full link utilization for any buffer size B. On the other hand, when using TCP NewReno, the full link utilization is achieved only when the buffer is provisioned following the "rule-of-thumbs" $B = C \cdot RTT_m$. For instance, when using the buffer size $B = 0.1 \cdot C \cdot RTT_m$, the link utilization of NewReno TCP falls below the 82%.

Figure 5 depicts the dynamics of the TCP congestion window and of the bottleneck queue in the case of TCP NewReno and TCP Westwood+. In particular, Figures 5(a) and 5(c) show the $cwnd$ and the queue length, respectively, when the bottleneck buffer B is provisioned using the "rule of thumb" $B = C \cdot RTT_m$, whereas Figures 5(b) and 5(d) show the $cwnd$ and the queue length when the buffer size is under provisioned using the rule $B = 0.1 \cdot C \cdot RTT_m$. When $B = C \cdot RTT_m$, it is worth noticing that the two TCP flavors behaves in the same way: in fact, in the case of loss, TCP Westwood+ decreases the congestion window to $C \cdot RTT_m$, which corresponds to the half of $B + C \cdot RTT_m$, that is, it corresponds exactly to the value that is set by TCP NewReno after loss. Also the queue dynamics behaves similarly, and both protocols achieve full link utilization.

When the buffer is underprovisioned with respect to the "rule-of-thumb", behaviors of NewReno and Westwood+ are different. As it is shown in Figure 5(b), NewReno TCP is not able to achieve full link utilization because of the congestion window that oscillates between $B + C \cdot RTT_m = 1.1 \cdot C \cdot RTT_m$ and $(1.1 \cdot C \cdot RTT_m)/2$. On the other hand, TCP Westwood+ congestion control oscillates between

$1.1 \cdot C \cdot RTT_m$ and $C \cdot RTT_m$, which provides a higher level of link utilization. To the purpose it is useful to compare Figures 5(c) and 5(d). When the buffer is underprovisioned (Figure 5(d)), the queue length using TCP NewReno is empty for repeating interval of roughly 20 seconds long during which the link is underutilized. On the other hand, the queue length that corresponds to the TCP Westwood+ congestion control shows that the queue becomes empty only at instants after congestion, which means that the link is fully utilized.

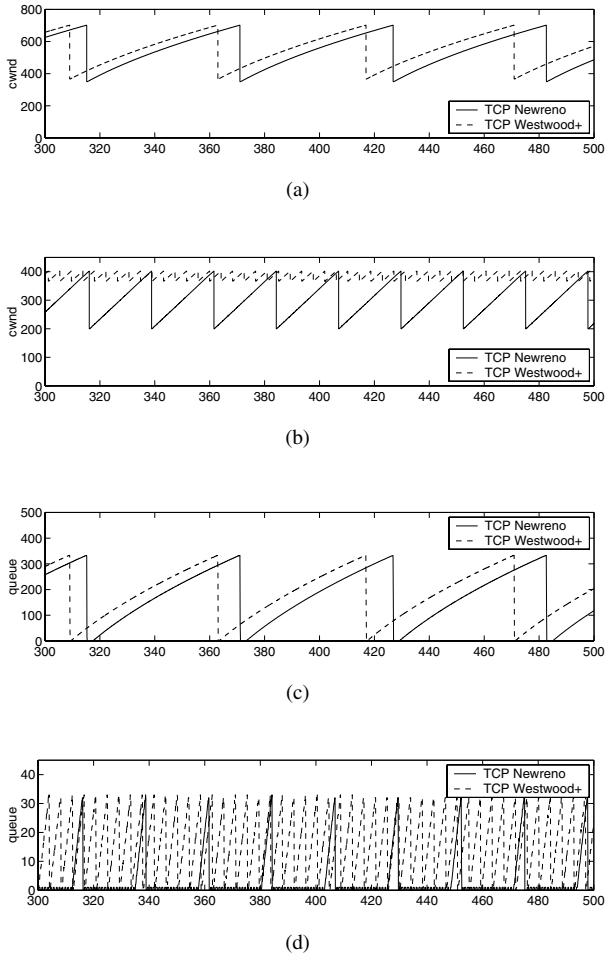

Fig. 5. TCP Congestion Window and queue utilization for a single TCP source scenario. (a) cwnd dynamics with $B = C \cdot RTT_m$; (b) cwnd dynamics with$B = 0.1 \cdot C \cdot RTT_m$; (c) queue dynamics with $B = C \cdot RTT_m$; (d) queue dynamics with $B = 0.1 \cdot C \cdot RTT_m$

In the Internet measurement scenario (see Figure 3), we investigate how three different streams share a 1 Gbps bottleneck link. We first consider 3 NewReno flows and then 3 Westwood+ flows. Figure 6 shows the throughputs in the case of 3 NewReno flows sharing the 1Gbps link, whereas Figure 7 shows same values in the case of 3 Westwood+ flows sharing the 1Gbps link. It is worth noting that the NewReno flows exhibit a classic "sawtooth" oscillatory behavior due to the "by half setting" of NewReno after congestion. On the other hand, it is very interesting to note that the throughput of Westwood+ exhibits an oscillation free behavior because the

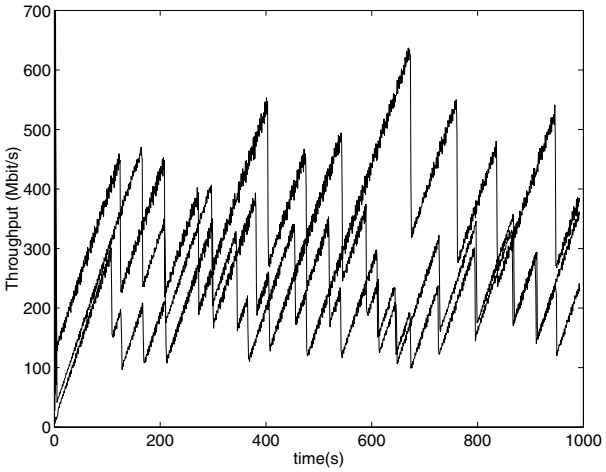

Fig. 6. Throughputs of 3 NewReno flows sharing a 1Gbps link.

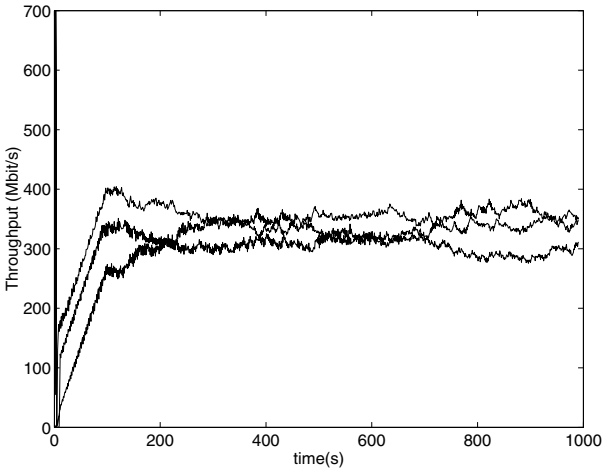

Fig. 7. Throughputs of 3 Westwood+ flows sharing a 1Gbps link.

congestion window is kept around the value of 5000 kbyte during all the test. The average per-connection throughput in the case of NewReno is 270 Mbps, whereas the average per-connection throughput in the case of Westwood+ is 320 Mbps.

B. The case of desynchronized flows

Finally we use ns2 to investigate the link utilization versus the buffer size when many TCP flows share the same bottleneck. Figure 8 depicts the measured link utilization in case of TCP NewReno and TCP Westwood+, when the number of competing flows that share the link n is 200. Link capacity of 200Mbps and RTT of 100 ms and 300 ms are considered. In this case the link utilization is high with both protocols and does not vary with the buffer size. This is consistent with results obtained analytically in Section IV-B, that is, when the number of flows is large, full link utilization can be obtained using buffers of size which is smaller than the one recommended by the "rule-of-thumb".

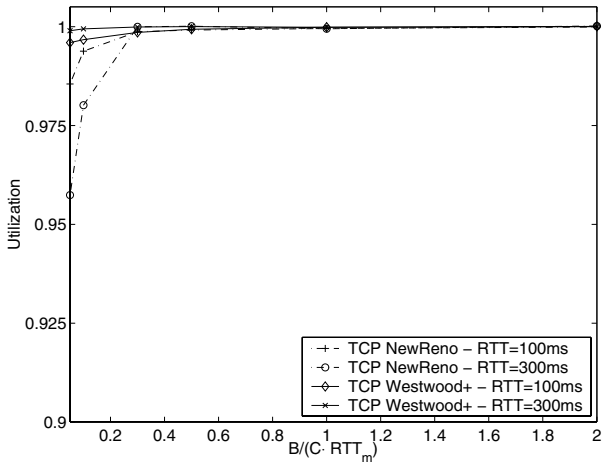

Fig. 8. Link utilization in a 200 fbws scenario with C=200Mbit/s as a function of the bottleneck buffer size.

VI. Conclusions

Mathematical analysis of buffer sizes that are required to provide full link utilization has shown that, differently from standard Reno/NewReno TCP, the utilization provided by Westwood+ TCP does not depend, in principle, on the buffer size. This result is interesting because smaller buffers are required to provide full link utilization in Gigabit networks where, due to technological issues [1], buffers can be underprovisioned with respect to the bandwidth delay product when flows are synchronized or to the $1/\sqrt{n}$ rule when flows are desynchronized and the number of competing flows n is low. Both computer simulations and experimental results have been reported to confirm the theoretical analysis.

References

[1] G. Appenzeller, I. Keslassy, N. McKeown, "Sizing Router Buffers," Proc. of Sigcomm 2004, August 30 - Sept. 3, 2004, Portland, Oregon, USA.

[2] Villamizar, C. and Song C. (1995), "High Performance TCP in ANSNET", ACM Computer Communication Review, vol. 24, no. 5, pp. 45-60.

[3] K. Avrachenkov, U. Ayesta, E. Altman, P. Nain and C. Barakat, "The effect of router buffer size on the TCP performance", in Proc. of the LONIIS Workshop on Telecommunication Networks and Teletraffi c Theory, pages 116-121, St. Petersburg, Russia, January 2002.

[4] S. Jordan, K. Jogi, C. Shi, and I. Sidhu, "The Variation of Optimal Bandwidth and Buffer Allocation With the Number of Sources," IEEE/ACM Transaction on Networking, Vol. 12(6), Dec. 2004.

[5] G. Rainaand D. Wischik, "Buffer sizes for large multiplexers: TCP queueing theory and instability analysis," EuroNGI conference on Next Generation Internet Networks, Rome, Italy, April 2005.

[6] V. Jacobson, "Congestion Avoidance and Control," ACM Computer Communications Review, 18(4): 314 - 329, August 1988.

[7] S. Mascolo, C. Casetti, M. Gerla, M. Sanadidi, R. Wang, "TCP Westwood: End-to-End Bandwidth Estimation for Effi cient Transport over Wired and Wireless Networks", ACM Mobicom 2001, July, Rome, Italy.

[8] L. A. Grieco, S. Mascolo "Performance evaluation and comparison of Westwood+, Vegas and New Reno TCP congestion control", ACM Computer Communication Review, April 2004.

[9] www.kernel.org.

[10] The Westwood+ TCP project. URL "http://www-ictserv.poliba.it/mascolo/tcp%20westwood/tcpwestwood.htm

[11] S. Shenker, L. Zhang, and D. Clark, "Some observations on the dynamics of a congestion control algorithm", ACM Computer Communications Review, pages 30-39, Oct 1990.

[12] G. Iannaccone, M. May, C. Diot, "Aggregate traffi c performance with active queue management and drop from tail", Proc. of Sigcomm Comp. Comm. Rev., 31(3):4-13, 2001.

[13] L. Qiu, Y. Zhang, and S. Keshav. Understanding the performance of many tcp fbws. Computer Networks, 37(3-4):277-306, 2001.

[14] The network simulator. URL: www.isi.edu/nsnam/ns/

[15] Datatag Project. URL http://datatag.web.cern.ch/datatag/.