An Evaluation Study of Data Transport Protocols for e-VLBI

JULIANNE SANSA*
Makerere University / University of Groningen
ARPAD SZOMORU
Joint Institute for VLBI in Europe
And
J.M.VAN DER HULST
University of Groningen

This paper compares TCP-like data transport algorithms in the light of e-VLBI requirements and proposes HTCP with bandwidth estimation (HTCP-BE) as a suitable candidate by simulating its behaviour in comparison with seven existing TCP variants; HighSpeed TCP for large Congestion Window (HSTCP), Scalable TCP (STCP), Binary Increase TCP (BIC), Cubic TCP (CUBICTCP), TCP for Highspeed and long-distance networks (HTCP), TCP Westwood+ (TCPW) and standard TCP (TCP). Using average throughput efficiency and stability as the performance metrics we show that HTCP-BE better suits e-VLBI needs than any of the other seven protocols in environments with random background traffic.

Categories and Subject Descriptors: C.2.2 [**Computer-Communication Networks**]: Network Protocols—Protocol verification; C.2.5 [**Computer-Communication Networks**]: Local and Wide-Area Networks—Internet
General Terms: Data Transport Protocols, Network Performance
Additional Key Words and Phrases: e-VLBI, Radio Astronomy;

IJCIR Reference Format:

Sansa J., Szomoru A. and Van der Hulst J. M. 2007. An Evaluation Study of Data Transport Protocols for e-VLBI. International Journal of Computing and ICT Research, Vol. 1, No. 1, pp. 68 – 75.
http:www.ijcir.org/volume1-number1/article 8.pdf.

1. INTRODUCTION

The European VLBI Network (EVN) is an array of radio telescopes located throughout Europe and as far away as China and South Africa. These radio telescopes produce data at rates of up to 1 Gbps each. Until recently, these data streams were recorded on tapes, nowadays on hard disk drives, and shipped to the correlator located at the Joint Institute for VLBI in Europe (JIVE), in Dwingeloo, the Netherlands. During the last few years JIVE, in collaboration with the European National Research Networks and the pan-European Research Network GEANT, has worked on a proof-of-concept (PoC) project to connect several telescopes across Europe in real-time to the correlator via the Internet (electronic VLBI or e-VLBI). This

* Author's Address: J. Sansa, Department of Networks, Faculty of Computing and IT, Makerere University, P. O. Box 7062, Kampala, Uganda, *sansa@cit.mak.ac.ug*
Kapteyn Astronomical Institute, University of Groningen, Postbus 800, 9700 AV Groningen, The Netherlands, sansa@astro.rug.nl, *www.astro.rug.nl/~sansa/*
A. Szomoru, Joint Institute for VLBI in Europe, Oude Hoogeveensedijk 4, 7991 PD Dwingeloo, The Netherlands, *szomoru@jive.nl*
J.M. van der Hulst, Kapteyn Astronomical Institute, University of Groningen, Postbus 800, 9700 AV Groningen, The Netherlands, *vdhulst@astro.rug.nl*

International Journal of Computing and ICT Research, ISSN 1818-1139, Vol.1, No.1, pp. 68 - 75, June 2007

project has led to an EC sponsored project called EXPReS, which over the next few years will transform the EVN to a fully functional real-time e-VLBI network. During the PoC project it became clear that in spite of the vast capacity of the connecting networks; the actual transport of large data streams poses quite a challenge. By the nature of the e-VLBI technique the amount of data exchanged is very enormous (terabytes of data to be transmitted in hours), which is loss tolerant but delay sensitive, since correlation can only be achieved if signals observed at the same time at the different telescopes are processed together. The Mark5 [haystack] application that handles e-VLBI data uses the Transport Control Protocol (TCP). By its nature, e-VLBI involves transporting huge amounts of data via the Internet over long distances from geographically dispersed telescopes to one central correlator. TCP is somewhat problematic in combination with long distance high-speed links [Allman et al. 1999; Floyd 2003; Wright and Stevens 1994]. Having identified the congestion control algorithm of TCP as the bottleneck under certain circumstance as reported in [Sansa J. et al. 2006], we investigate e-VLBI data transport with several TCP-like transport protocols. Based on the observations we then propose yet a new congestion control algorithm (a minor modification to HTCP), which suits e-VLBI requirements better than the other seven protocols considered.

In the next section we give a brief description about the congestion control algorithms of the TCP-like protocols we are evaluating including our proposed HTCP-BE, followed by an explanation of the simulations setup, after which we present the protocol performance evaluation and finally conclude.

2. TCP-like CONGESTION CONTROL ALGORITHMS

In this section we present the congestion control algorithms and response functions of standard TCP and its recent modifications. We categorize them in two, the non-adaptive and the adaptive algorithms.

2.1 Non-Adaptive TCP algorithms

These are described as the algorithms, which modify the CWND with increase and decrease factors (α and β respectively) that are functions of the value of CWND at the point of modification but independent of all the previous CWND values. This results in CWND values that are changed by specific pre-defined percentage and CWND modifications that are oblivious of each other. In the following subsections we describe the algorithms, which are non-adaptive.

2.1.1 Standard TCP [Wright and Stevens 1994]

The TCP congestion avoidance algorithm is presented in the equations (1) and (2), while the resulting response function is *equation (3).*

On Acknowledgment:

$$CWND \leftarrow CWND + \alpha/CWND \quad (1)$$

On Packet Loss Indication:

$$CWND \leftarrow CWND - \beta \times CWND \quad (2)$$

Where $\alpha = 1$ and $\beta = 0.5$

$$CWND_{average} = 1.2 / p^{0.5} \quad (3)$$

Where p is the steady state packet loss rate.

2.1.2 Highspeed TCP (HSTCP) [Floyd 2003]

HSTCP's congestion avoidance algorithm is similar to that of standard TCP as presented in the *equations (1) and (2)*, but differs in the values of α and β. HSTCP obtains α and β from a pre-computed table based on the network's bandwidth delay product (BDP) in such a way that if its present value is below a certain pre-set threshold termed *low_window*, HSTCP sets $\alpha = 1$, $\beta = 0.5$ thus reverting back to standard TCP congestion control algorithm. However if the BDP value exceeds the *low_window* α and β are then assigned from the table. For our network in steady state the appropriate $\alpha = 26$ and $\beta = 0.22$.
HSTCP then presents a response function shown in *equation (4)*, which is able to reach a higher $CWND_{average}$ than that reached by standard TCP in *equation (3)* but takes a longer time to converge.

$$CWND_{average} = 0.12/p^{0.835} \quad (4)$$

Where *p* is the steady state packet loss rate.

2.1.3 Scalable TCP (STCP) [Kelly 2003]

STCP's congestion avoidance algorithm is a modification of HSTCP and is presented in *equation (5)* for each packet acknowledged and *equation (2)* for each packet lost.
On Acknowledgment:

$$CWND \leftarrow CWND + \alpha * CWND \tag{5}$$

The resulting STCP response function is as shown *equation (6)*, which reaches a higher $CWND_{average}$ than that reached by HSTCP in *equation (4)* for the same steady state packet loss rate. STCP however takes a longer time to converge than HSTCP.

$$CWND_{average} = 0.038/p^{1} \tag{6}$$

Where *p* is the steady state packet loss rate.

2.1.4 Binary Increase TCP (BIC)[Xu et al. 2004]

BIC is a semi-adaptive algorithm, which uses a combination of the binary search increase and the additive increase for incrementing the CWND on arrival of acknowledgments when the current CWND exceeds a preset *low_window* threshold. Should the CWND value be below the *low_window* threshold the standard TCP CWND increment is reverted to. Binary search increase follows the binary search algorithm concept and sets the next CWND value to a value halfway between its current value and currently known maximum. The binary search increase is used when the CWND increment is small while the additive increase is used when the CWND increment is large. Occasionally when the known maximum window is exceeded, BIC goes into a slow start phase in which it probes for a new maximum. The BIC CWND increment is summarised in the *equation (7)*.

$$CWND \leftarrow \begin{cases} CWND + 1/CWND & CWND < lw \\ CWND + (tw - CWND)/CWND & tw - CWND < Sx \\ CWND + Sx/CWND & tw - CWND > Sx \\ CWND + SS\_CWND/CWND & CWND > Wx \end{cases} \tag{7}$$

Where *lw* is the CWND threshold beyond which BIC engages otherwise standard TCP is used, *tw* is the midpoint between the maximum and minimum window sizes, *Sx* is the maximum increment, *SS_CWND* is a variable to keep track of CWND increase during the BIC slow-start and *Wx* is the maximum window size; initially set to a default large integer. BIC CWND increase is semi-adaptive as it follows adaptive patterns implicitly. It progresses through binary increase, additive increase and finally slow-start probing, each of which relies on the previous episode receiving acknowledgments. In addition these functions consecutively increase the CWND in greater proportions than the previous function. On detection of packet loss BIC reduces it's CWND based on multiplicative decrease pattern following *equation (8)*. This decrease function is non-adaptive.

$$CWND \leftarrow \begin{cases} CWND * (1 - \beta) & CWND \geq low_{window} \\ CWND * 0.5 & CWND < low_{window} \end{cases} \tag{8}$$

Where the decrease factor β is fixed at 0.125.

2.2 Adaptive TCP algorithms

These we describe as the algorithms that modify the consecutive CWND differently considering a variable related to the previous packet congestion event not merely the previous CWND value e.g. the time elapsed since the last packet loss event or the throughput attained just before the last packet loss event. They use these variables (duration and /or throughput) to gauge the changing level of congestion, hence modifying

the CWND to a value that indicates the prevailing congestion conditions other than by a fixed percentage. The increase and decrease factors are functions of either the time elapsed since the last congestion event or the throughput just before the last congestion event. In the following subsections we discuss some of the recently proposed algorithms, which are adaptive.

2.2.1 Westwood TCP (TCPW) [Gerla et al. ; Mascolo et al. 2004; Mascolo et al. 2001]

TCPW connection establishment is based on standard TCP (as in *equation (1)*) until a packet loss is encountered. At that point the adaptive decrease is invoked by setting the *ssthresh* (and thus *CWND*) to a value that is larger than the corresponding value set by standard TCP. The *ssthresh* is set based on the available bandwidth estimate *B* for the connection. The adaptive decrease algorithm is defined as follows.

On Packet Loss Indication:

$$\text{ssthresh} = (B * RTT_{min}) / \text{seg size}$$
$$\text{If ssthresh} \leq 2 \text{ then ssthresh} = 2 \quad (9)$$

If the packet loss is detected by 3 duplicate packets

$$CWND_{new} = \text{ssthresh} \quad (10)$$

If congestion is detected with a timeout of an unacknowledged packet

$$CWND_{new} = 1 \quad (11)$$

TCPW setting of the ssthresh value results in the TCPW connection utilising more bandwidth than standard TCP due to two factors. i) When the loss is signaled by 3 duplicate ACKs, TCPW's CWND does not decrease as much as standard TCP decrease. ii) When the loss is signaled by timeout of an unacknowledged packet both TCPW and standard TCP decrease the CWND to just one packet, however since TCPW sets the ssthresh to a much larger value than what standard TCP does, the TCPW connection has a longer slow start phase (in which its CWND grows exponentially) than the standard TCP connection. TCPW increase function is non-adaptive while its decrease function is adaptive. TCPW however presents the challenge of accurate estimation of the available bandwidth. The bandwidth estimation algorithm used for TCPW+ [Mascolo et al. 2004] is more accurate than that used for an earlier version of TCPW [Gerla et al. ; Mascolo et al. 2001].

2.2.2 TCP for High-speed and long distance networks (HTCP) [R.N. Shorten and D. J. Leith 2004]

Uses an adaptive congestion control algorithm in which α_i the increase factor of source *i* is set for each acknowledgment as a function of the time elapsed since the last packet loss event. On receipt of each acknowledgment the CWND increment is defined by the same equation as standard TCP (1), but differs in the assignment of α_i, which follows *equation (12)*.

$$\alpha_i \leftarrow \begin{cases} 1 & \Delta_i \leq \Delta^L \\ 1 + 10(\Delta_i - \Delta^L) + (\frac{\Delta_i - \Delta^L}{2})^2 & \Delta_i > \Delta^L \end{cases} \quad (12)$$

Where Δ_i is the elapsed time since the last congestion event experienced by the flow *i*, Δ^L is the time duration used as the threshold for switching from the low to high speed regimes. In the first case, α_i is increased in the same manner as the standard TCP algorithm. In the second case however α_i is increased by a function relative to the elapsed time since the last congestion event, whose response function is similar to that of HS-TCP.

HTCP's decrease factor β_i for a particular source *i* is computed relative to that flow's throughput just before and after the last congestion event. The decremented CWND is thus set according to *equation (13).*

$$CWND \leftarrow \beta_i * CWND \tag{13}$$

Where

$$\beta_i(k+1) \leftarrow \begin{cases} 0.5 & \frac{B_i^-(k+1)-B_i^-(k)}{B_i^-(k)} > 0.2 \\ \frac{RTT_{min,i}}{RTT_{max,i}} & otherwise \end{cases} \tag{14}$$

Where $B_i^-(k)$ is the throughput of flow *i* immediately before the k'th congestion event, $B_i^+(k)$ is the throughput of flow *i* immediately after the k'th congestion event, $RTT_{min,i}$ is the minimum RTT experienced by the i'th source and $RTT_{max,i}$ is the maximum RTT experienced by the i'th source. The current implementation of HTCP uses *equation (15)* to estimate $RTT_{min,i}$ / $RTT_{max,i}$

$$\beta_i(k+1) = \min_j \beta_i(j) \frac{B_i^-(j)}{B_i^+(j)} \tag{15}$$

2.2.3 Cubic TCP (CUBIC)[Rhee and Xu 2005]

CUBIC is an algorithm that aims at improving and simplifying the BIC algorithm; hence its resulting response function is similar to that BIC. It is however a fully adaptive algorithm unlike BIC since CWND modifications are computed relative to the time elapsed since the last packet loss event in a similar manner to HTCP. The CWND of CUBIC is determined by *equation (16).*

$$CWND = C\,(t - K)^3 + max\_win \tag{16}$$

Where *C* is a scaling factor, *t* is the elapsed time from the last window reduction, *max_win* is the window size just before the last reduction, and $K = \sqrt[3]{(max\_win * \beta / C)}$, where β is a constant multiplication decrease factor applied for window reduction at the time of the loss event.

2.2.4 Our Proposed Algorithm (HTCP with Bandwidth Estimation (HTCP-BE))

Basing on the observations from the above stated algorithms [Floyd 2003; Kelly 2003; Xu et al. 2004; Mascolo et al. 2004; R.N.Shorten and D.J.Leith 2004; Rhee and Xu 2005] as well as some comparative studies [Bullot et al. 2004; Li et al. 2005; Ha et al. 2006; Weigle et al. 2006], we propose an algorithm that combines the strengths of HTCP and TCPW. With this algorithm a connection will be established following HTCP-like rules, however at a packet loss, we propose that the TCPW's adaptive decrease be evoked. From our simulations with NS-2, on a packet loss event a TCPW flow sets its CWND to a value averagely 17% greater than that set by an HTCP flow. The increase factor will be controlled by the HTCP adaptive increase, which controls increment relative to the elapsed time since the last packet loss event. From our simulations the HTCP increase function yields much faster growth of the CWND than that obtained with TCPW. The aim is to use the tested bandwidth estimation mechanism employed by TCPW algorithm and combine it with adaptive increase mechanism of HTCP. The expectation is that this will ensure higher link utilisation than what is possible with any of TCPW or HTCP. The weaknesses of HTCP in short RTT flows as well as parallel flows revealed in [Ha et al. 2006] we propose to solve by employing the tested TCPW adaptive back-off .The non-adaptive nature of TCPW increase factor will be taken care of by the HTCP adaptive increase factor. Our proposed algorithm is summarised in *equation (17)* and *(18).*

On Acknowledgment:

$$CWND \leftarrow CWND + \alpha_i / CWND \tag{17}$$

Where α_i is the same as *equation (12)*

On Packet Loss event:

$$CWND \leftarrow B * RTT_{avg} \tag{18}$$

Where B, the bandwidth at the time of the packet loss is obtained using the estimation algorithm used for TCPW+ [Mascolo et al. 2004]

3. SETUP

Using NS-2 simulator [Breslau et al. 2000; ns], we run simulations for 300 seconds, three times for each protocol per background traffic level. We then compute the average throughput efficiency and coefficient of variation (which is a measure of protocol stability, also used by [Floyd et al. 2000; Rhee and Xu 2005]). We use four background traffic levels, which we label 0 through 3. The background traffic mix follows observations in [Luo and Marin 2005], with the exception of non-TCP traffic. With higher background traffic level the amount of background traffic introduced is increased. The simulations (scripts available on request), are for the typical e-VLBI setting within the EVN with round trip times varying between 10ms and 40ms and bottleneck links of 1 Gbps.

4. PERFORMANCE EVALUATION

While quite a number of metrics exist for evaluating a transport protocol such as throughput efficiency, protocol fairness [Chiu and Jain 1989], stability, impact on CPU, impact on RTT, queuing delay, packet drop rate and packet re-ordering rate we have chosen the two most important ones that would affect the e-VLBI application. Some of these metrics have been used in other related studies [Bullot et al. 2004; Li et al. 2005; Ha et al. 2006; Weigle et al. 2006]. Next we present our results of the two chosen metrics.

4.1 Throughput Efficiency

For each of the protocols we plot the average throughput achieved across varying background traffic environments as shown in Figure 1. The higher this value the more efficient the protocol is.

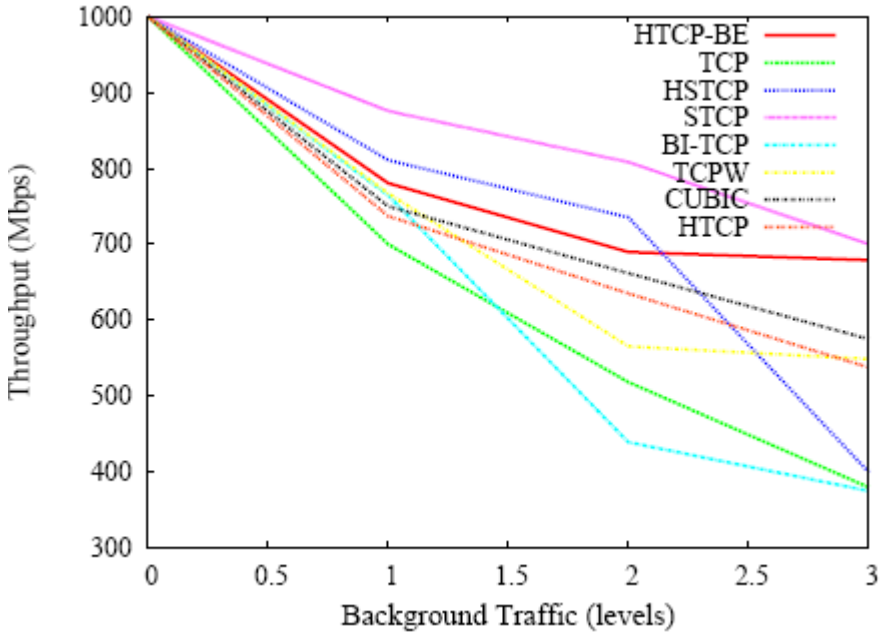

Figure 1. Throughput Efficiency of the eight protocols across varying background traffic

All the protocols achieve less throughput with increasing background traffic. Clearly HTCP-BE reduces least with in higher regimes.

4.2 Stability

Similarly for each of the protocols we plot the average coefficient of variation across varying background traffic environments as shown in Figure 2. The lower this value the more stable the protocol.

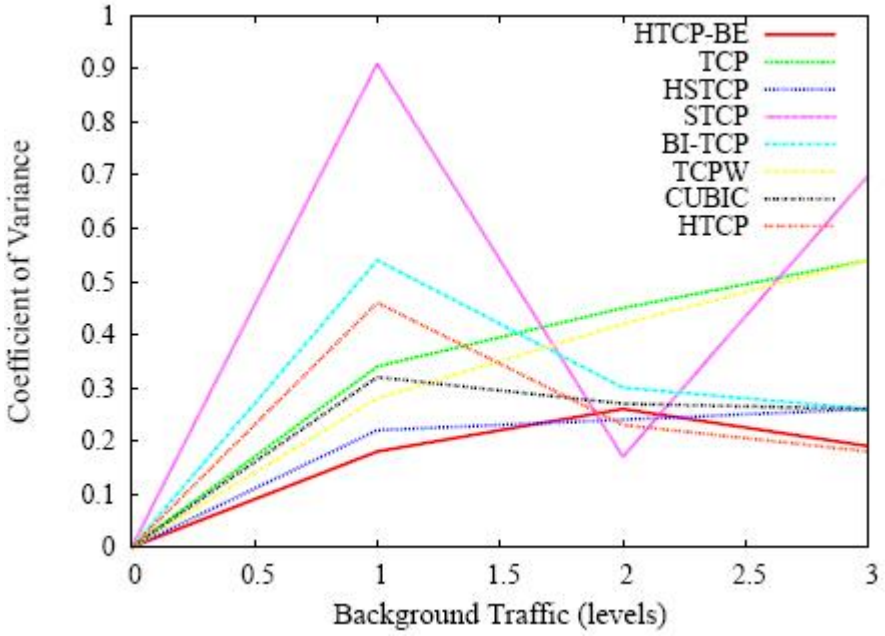

Figure 2. Protocol Stability of the eight protocols across varying background traffic

All the protocols with the exception of STCP exhibit a good stability across the different levels of background traffic. With in the higher regimes HTCP-BE particularly shows the best stability.

5. CONCLUSION AND FUTURE WORK

Our results show that HTCP-BE is of high stability across varying background traffic environments and maintains a high throughput, thus suitable for the delay sensitive e-VLBI application.
To further improve data transport for e-VLBI we shall embark on the following;

- Vary Background traffic to include non-TCP flows because much as the most Internet traffic is TCP a significant portion is non-TCP [Luo and Marin 2005] and it may affect the performance differently.
- Considering the caution in [Allman 1999; Floyd and Paxson 2001] about a balance between simulations and live Internet tests, we will implement HTCP-BE as a kernel module and run e-VLBI tests on a real network.
- Increase the throughput efficiency of HTCP-BE by maintaining a relatively large enough CWND (not reducing at all) in periods of low packet loss and only decreasing it when the packet loss gets out of hand. This is especially useful because e-VLBI is not loss sensitive but delay sensitive.
- Exploring other transport protocols that are not TCP related such as User Datagram Protocol (UDP) and Real-Time Protocol (RTP).

REFERENCES

ALLMAN, A. F. M. 1999. On the effective evaluation of TCP. ACM Computer Communication Review 5, 29.

ALLMAN, M., V., P., AND STEVENS, W. 1999. TCP congestion control. RFC 2581, Internet Engineering Task Force.
BRESLAU, L., ESTRIN, D., FALL, S., HEIDEMANN, J., HELMY, P., MCCANNE, H. S., VARADHAN, K., XU, Y., AND YU, H. 2000. Advances in network simulation. IEEE Computer 55, 5, 59–67.
BULLOT, H., COTTRELL, L., AND HUGHES-JONES, R. 2004. Evaluation of advanced tcp stacks on fast long-distance production networks. PFLDnet .
CHIU, D.-M. AND JAIN, R. 1989. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. Comput. Netw. ISDN Syst. 17, 1, 1–14.
FLOYD, S. 2003. Highspeed TCP for large congestion windows. RFC 3649, Internet Engineering Task Force.
FLOYD, S., HANDLEY, M., PADHYE, J., AND WIDMER, J. 2000. Equation-based congestion control for unicast applications. In SIGCOMM '00: Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication. ACM Press, New York, NY, USA, 43–56.
FLOYD, S. AND PAXSON, V. 2001. Difficulties in simulating the Internet. IEEE/ACM Trans. Netw. 9, 4, 392–403.
GERLA, M., SANDAIDI, M., VALLA, M., AND WANG, R. TCP Westwood with adaptive bandwidth estimation to improve efficiency/friendliness tradeoffs. Computer Communication Journal 2003.
HA, S., KIM, Y., LE, L., RHEE, I., AND XU, L. 2006. A step toward realistic evaluation of high-speed TCP protocols. Technical report, Department of Computer Science, North Carolina State University.
HAYSTACK. Mark5 vlbi data system. Haystack observatory Website. http://web.haystack.mit.edu/mark5/.
KELLY, T. 2003. Scalable TCP: Improving performance in high-speed wide area networks. ACM SIGCOMM Computer Communications Review 33, 2.
LI, Y., LEITH, D., AND SHORTEN, R. 2005. Experimental evaluation of TCP protocols for high-speed networks.
LUO, S. AND MARIN, G. A. 2005. Realistic internet traffic simulation through mixture modeling and a case study. In WSC '05: Proceedings of the 37th conference on Winter simulation. Winter Simulation Conference, 2408–2416.
MASCOLO, S., C. CASETTI, M. GERLA, M. S., AND WANG, R. 2001. Bandwidth estimation for enhanced transport over wireless links. In Mobile Computing and Networking.
MASCOLO, S., GRIECO, L. A., FERORELLI, R., CARMADA, P., AND PISCITELLI, G. 2004. Performace evaluation of westwood+ TCP congestion control. Performance Evaluation 55.
NS. The NS-2 simulation. www.isi.edu/nsnam/ns.
RHEE, I. AND XU, L. 2005. Cubic: A new TCP-friendly high-speed tcp variant. Proc. PFLDnet .R.N.Shorten and D.J.Leith. 2004. H-tcp: TCP for high-speed and long-distance networks. Proc. PFLDnet, Argonne.
SANSA, J., SZOMORU, A., AND VAN DER HULST, J. M. 2006. On network measurement and monitoring of end-to-end paths used for e-vlbi. Chapter in Advances in Systems Modeling and ICT Applications ISBN: 13: 978-9970-02-604-3.
WEIGLE, M., P.SHARMA, AND FREEMAN, J. R. 2006. Performance of competing high-speed TCP flows. Proceedings of IFIP Networking.
WRIGHT, G. R. AND STEVENS, W. R. 1994. TCP/IP Illustrated: The Protocols. Addison-Wesley, Reading, Mass.
XU, L., HARFOUSH, K., AND RHEE, I. 2004. Binary increase congestion control for fast long-distance networks. INFOCOM.