1-1-2007

Managing the Bursty Nature of Packet Traffic using the BPTraSha Algorithm

Karim Mohammed Rezaul

Vic Grout
Glyndwr University, v.grout@glyndwr.ac.uk

Recommended Citation

Rezaul, K.M. & Grout, V.(2007), 'Managing the Bursty Nature of Packet Traffic using the BPTraSha Algorithm'. [Paper presented to the Third Collaborative Research Symposium on Security, E-Learning, Internet and Networking 14th-15th June 2007]. Plymouth: Plymouth University

Managing the Bursty Nature of Packet Traffic using the BPTraSha Algorithm

Abstract

The rapid development of network technologies has widened the scope of Internet applications and, in turn, increased both Internet traffic and the need for its accurate measurement, modelling and control. Various researchers have reported that traffic measurements demonstrate considerable burstiness on several time scales, with properties of self-similarity. The self-similar nature of this data traffic may exhibit spikiness and burstiness on large scales with such behaviour being caused by strong dependence characteristics in data: that is, large values tend to come in clusters and clusters of clusters and so on. Several studies have shown that TCP, the dominant network (Internet) transport protocol, contributes to the propagation of self-similarity. Bursty traffic can affect the Quality of Service of all traffic on the network by introducing inconsistent latency. It is easier to manage the workloads under less bursty (i.e. smoother) conditions. In this paper, we examine the use of a novel algorithm, the Bursty Packet Traffic Shaper (BPTraSha), for traffic shaping, which can smooth out the traffic burstiness. Experimental results show that this approach allows significant traffic control by smoothing the incoming traffic. BPTraSha can be implemented on the distribution router buffer so that the traffic's bursty nature can be modified before it is transmitted over the core network.

Keywords

Self-similarity, Long-range dependence, Auto-correlation function, Hurst parameter

Disciplines

Computer and Systems Architecture | Digital Communications and Networking | Hardware Systems | Systems and Communications

Comments

This paper was presented at the Third Collaborative Research Symposium on Security, E-Learning, Internet and Networking (SEIN 2007), 3rd International NRG Research Symposium, 14-15 June 2007, which was held by University of Plymouth and the symposium proceedings are available at http://www.cscan.org

Managing the Bursty Nature of Packet Traffic Using the BPTraSha Algorithm

Karim Mohammed Rezaul and Vic Grout

Centre for Applied Internet Research (CAIR), University of Wales, NEWI, Wrexham, UK
e-mail: {karim|vic}@cair-uk.org

Abstract

The rapid development of network technologies has widened the scope of Internet applications and, in turn, increased both Internet traffic and the need for its accurate measurement, modelling and control. Various researchers have reported that traffic measurements demonstrate considerable burstiness on several time scales, with properties of self-similarity. The self-similar nature of this data traffic may exhibit spikiness and burstiness on large scales with such behaviour being caused by strong dependence characteristics in data: that is, large values tend to come in clusters and clusters of clusters and so on. Several studies have shown that TCP, the dominant network (Internet) transport protocol, contributes to the propagation of self-similarity. Bursty traffic can affect the Quality of Service of all traffic on the network by introducing inconsistent latency. It is easier to manage the workloads under less bursty (i.e. smoother) conditions. In this paper, we examine the use of a novel algorithm, the Bursty Packet Traffic Shaper (BPTraSha), for traffic shaping, which can smooth out the traffic burstiness. Experimental results show that this approach allows significant traffic control by smoothing the incoming traffic. BPTraSha can be implemented on the distribution router buffer so that the traffic's bursty nature can be modified before it is transmitted over the core network. This paper continues the work of Rezaul and Grout (2007a).

Keywords

Self-similarity, Long-range dependence, Auto-correlation function, Hurst parameter, BPTraSha.

1. Introduction

A number of factors, such as a slow start phase of the congestion window, packet losses, ack-compression of TCP traffic and multiplexing of packets at the bottleneck rate, can cause either short- or long-term burstiness in the behaviour of TCP flow (Aggarwal *et al*., 2000). Park *et al.* (1997) investigate how various versions of TCP congestion control affect network performance when traffic is bursty. They shows a significant adverse impact on network performance attributable to traffic self-similarity and, while throughput declines gradually as self-similarity increases, queueing delay increases more drastically. Self-similarity is closely related to the phenomenon of heavy-tailed distributions, where the tail index of the distribution declines as a power law with small index (less than 2). TCP represents the dominant transport protocol of the network (e.g. Internet), which contributes to the propagation of self-similarity. It was shown by Veres *et al.* (2003) that TCP itself inherits self-similarity when it is combined with self-similar background traffic in a bottleneck buffer through the transform function of the linear system.

Guo *et al*. (2000) investigated the relationship between TCP's congestion control mechanism and traffic self-similarity under certain network conditions. They also demonstrated (Guo *et al*., 2001) that, when a TCP connection is going through a highly-lossy channel - and the loss condition is not affected by this single TCP connection's behaviour, TCP starts to produce packet trains that show pseudo-self-similarity, i.e., traffic is self-similar over a limited range of time scales. In fact, when the loss rate is relatively high, TCP's adaptive congestion control mechanism generates traffic with heavy-tailed-off or idle periods (i.e. inter-arrival time), which in turn introduces long-range dependence into the overall traffic. Sikdar and

Vastola (2001) analysed the traces of actual TCP transfers over the Internet and reported that individual TCP flows, isolated from the aggregated flow on the link, also have a self-similar nature. Also, the loss rate experienced by TCP flow is an important indicator of the degree of self-similarity in the network traffic. A natural construction of the extremely bursty nature of TCP traffic comes from timeouts (representing 'silent' periods) that lead to losses and, consequently, losses increase the burstiness - and higher loss rates thus lead to a higher degree of self-similarity, i.e. higher values of Hurst parameter (Sikdar and Vastola, 2001). It has been shown (Peha, 1997) that, if packets were to arrive according to the well-behaved Poisson process, simple retransmission mechanisms can make traffic appear self-similar over time scales and be a possible source of long-range dependence. Retransmission mechanisms can make a network congestible, because these mechanisms often cause network inefficiencies which cause throughput to degrade specifically in periods when load is already high.

One of the major drawbacks of TCP/IP is the lack of true Quality of Service (QoS) functionality. QoS in networks, in simple terms, is the ability to guarantee and limit bandwidth appropriately for certain services and users. *Traffic shaping* is the term used for any system by which traffic is constrained to a specific speed. Traffic shaping is an attempt to control network traffic in order to optimize, attempt to optimize or guarantee performance, low-latency and bandwidth. Traffic shaping deals with concepts of classification, queue disciplines, enforcing policies, congestion management, QoS and fairness. Shaping is the mechanism by which packets are delayed before transmission in an output queue to meet a desired output rate. This is one of the most common requirements of users seeking bandwidth control solutions. The basic principle of traffic shaping is based on the fact that the outgoing traffic from the FireBrick or router is scheduled. (A FireBrick is a network appliance with a rich feature set, including a stateful firewall, router, managed switch, traffic shaping, tunneling, multilink handling, and much more.)

Each packet has a time stamp, stating when it is to be sent, and all traffic is normally sent in order and not before its time. This method is used to deliberately slow responses from reject and bounce filters, as well as for speed lanes. When sending a packet, its length is considered and the transmission time added to time for the next packet to be sent. This ensures packets can only actually leave at the designated rate and no faster. Shapers can smooth out bursty traffic and attempt to limit or ration traffic to meet, but not exceed, a configured rate (e.g. packets per second or bits/bytes per second). However, earlier research (Neidhardt and Erramilli, 1996 and Vamvakos and Anantharam, 1998) reports that the strong robustness of self-similarity properties that exist in traffic cannot be removed by shaping.

The rest of this paper is organised as follows. Section 2 highlights research related to shaping traffic. Section 3 describes the definitions of self-similarity, long-range dependence and the autocorrelation function. Section 4 introduces the algorithm BPTraSha and its purpose. Section 5 discusses the performance and complexity of BPTraSha by experimental analysis. Section 6 elaborates on how BPTraSha algorithm reduces the long-range dependence of traffic. Finally we draw conclusions and suggest future work in section 7.

2. Related research

Several researchers have shown how to control the network in situations where the distribution tail of the traffic flow process cannot be altered. Pruthi and Popescu (1997) claim that, by incorporating shapers and policers at the edges of the networks, huge buffers are needed that result in large delays and may thus be unacceptable in practice. Darlagiannis *et* *al.* (2003) present a Burst Shaping Queueing (BSQ) algorithm, which can minimize the burstiness of traffic on packet switched routers by interleaving packets that are going to follow different links on next hops. Molnár and Vidács (1997) discuss issues of shaping and simulated queueing performance of ATM traffic. In this work, a leaky bucket shaping method is used and the shaping effect surprisingly results in higher values for the estimated Hurst parameter (the degree of self-similarity) - that is, the estimated Hurst parameter is increased due to shaping. It is also noted that the interpretation of the estimated Hurst parameter is problematic in practice.

Xue and Yoo (2002) propose an optical packet assembly mechanism to function as a traffic shaper and its impact on self-similar traffic characteristics at the edge router is investigated. Simulation results demonstrate that the optical packet assembly mechanism can reduce traffic correlation and the degree of self-similarity. Bushmitch *et al.* (2003), present three different traffic shaping techniques: thinning, striping and shuffling, which can improve the queueing characteristics of data by decreasing the short-term burstiness and diminishing short-term correlations. However, none of these processes are shown to decrease the degree of Long-Range Dependence (LRD) in data. Christensen and Ballingam (1997) propose a dual leaky bucket technique for shaping the web traffic, reducing the intensity of the long duration traffic bursts, which, in turn, reduces the Hurst parameter. The 'leaky bucket' procedure (Turner, 1986) is also employed in (Harmantzis *et al.*, 2001) to examine the effectiveness of shaping in the case of α-stable fractal traffic and it is found that shaping and policing mechanisms do not eliminate self-similarity.

3. Self-similarity, long-range dependence and the autocorrelation function

It is especially important to understand the link between self-similarity and long-range dependence of network traffic and performance of the networks because such characterization can be potentially applied for control purposes such as traffic shaping, load balancing, etc. In general two or more objects having the same characteristics are called self-similar. A phenomenon that is self-similar looks the same or behaves the same when viewed at different degrees of magnification or on different scales on a dimension and is bursty over all time scales. Self-similarity is the property of a series of data points to retain a pattern or appearance regardless of the level of granularity used and is the result of long-range dependence in the data series. If a self-similar process is bursty on a wide range of timescales, it may exhibit long-range dependence. In general, lagged autocorrelations are used in time series analysis for empirical stationary tests. Self-similarity manifests itself as long-range dependence (i.e., long memory) in the time series of arrivals. The evidence of very slow, linear decay in the sample lag autocorrelation function (ACF) indicates the nonstationary behaviour (Brocklebank and Dickey, 1986). Long-range-dependence means that all the values at any time are correlated in a positive and non-negligible way with values at all future instants. A continuous time process, $Y = \{Y(t), t \geq 0\}$, is self-similar if it satisfies the condition (Willinger *et al.*, 1998) that

$$Y(t) \overset{d}{=} a^{-H} Y(at), \quad \forall a > 0, \quad and \quad 0 < H < 1 \tag{3.1}$$

where H is the index of self-similarity, called the Hurst parameter, and the equality is in the sense of finite-dimensional distributions. The stationary process X is said to be a long-range dependent process if its autocorrelation function (ACF) is non-summable (Cox, 1984), meaning that $\sum_{k=-\infty}^{\infty} \rho_k = \infty$. The details of how the ACF decays with k are of interest because the behaviour of the tail of the ACF completely determines its summability. According to (Leland *et al.*, 1994), *X* is said to exhibit long-range dependence if

$$\rho_k \sim L(t)\, k^{-(2-2H)}, \quad as \quad k \to \infty \qquad (3.2)$$

where $\frac{1}{2} < H < 1$ and $L(.)$ slowly varies at infinity, i.e., $\lim\limits_{t \to \infty} \frac{L(xt)}{L(t)} = 1,$ for all $x > 0$. (3.3)

Equation (3.2) implies that the LRD is characterized by an autocorrelation function that decays hyperbolically rather than exponentially fast. LRD processes are characterised by a slowly decaying covariance function that is not summable. When network performance is affected by LRD, the data are correlated over an unlimited range of time lags and this property results in a scale invariance phenomenon. Then no characteristic time scale can be identified in the process: they are all equivalent for describing its statistics, i.e., the part resembles the whole and vice versa.

4. BPTraSha: an algorithm for controlling bursty traffic

```
T = timestamp
B = Packet size in bytes
TT = transmission time
bps = Bit per second
Delt = Delay in second
Tmod = Modified time
Tmod_cng = change in modified time
bps_mod = Modified bit per second
Ld = Longest delay
Sd = Shortest delay
S = sample count (e.g.  number of packet sequences)
C = link speed

1.  Capture B for corresponding T (i.e.  T and B)
2.  Count S
3.  For k = 0 to (S-1)
    a)   if (k = 0)
             bps[k] = B[k] *8 / (T[k]+TT[k])
                where TT[k] = B[k] *8 / C
         else bps[k] = B[k]*8 / (T[k]-T[k-1]+TT[k-1])
    b)   if (k = 0)
             Delt[k] = 0
             Tmod[k] = T[k]
         else
             i)    Delt[k] = T[k]-(Tmod[k-1]+TT[k-1])
             ii)   if (Delt[k] >= 0)
                       Tmod[k] = T[k]
                   else
                       Tmod[k] = T[k]-Delt[k]
4.  For k = 0 to (S-2)
    i)  if (k = 0)
            Tmod_cng[k] = Tmod[k]
        else
            Tmod_cng[k] = Tmod[k+1]-Tmod[k]
    ii)  set    bps_mod[k] = B[k]*8 / Tmod_cng[k]
    iii) if (Delt[k] <0)
            find out Ld      // Longest delay
            find out Sd      // Shortest delay
5.  Exit
```

Figure 1: The algorithm, BPTraSha

Let us assume that client networks, C_1, C_2, C_3,….., C_n, are connected to the main router of an Internet service provider (ISP). The packet sequences (i.e. packet size in bytes) from different sources are queued at the router buffer. The packet sequences arrive at the router buffer with a timestamp in seconds (or milliseconds). Therefore, we have a packet size in bytes for the corresponding timestamp. For the experimental analysis, we used Lawrence Berkeley Laboratory (LBL) TCP data which are publicly available from ITA (2002). The bursty nature of packet sequences arriving at the router will be shaped at the fixed rate by the shaper algorithm BPTraSha. Here we mean the link speed as the desired fixed rate (i.e. capacity, C) at which the packets would be transmitted. In other words, bursty traffic in the input will be regulated at the fixed rate before they pass through the network. The algorithm is described in Figure 1. Table 1 illustrates a sample of trace files that the BPTraSha algorithm uses. The algorithm is implemented both in Java and Matlab programming language.

Length of samples	Timestamp (T_i)	Packet size in byte (B_i)
1	0.008185	41
2	0.010445	42
3	0.023775	42
4	0.026558	41
5	0.029002	82
6	0.032439	55
7	0.049618	41
8	0.052431	42
9	0.056457	42
10	0.057815	454
11	0.072126	40
12	0.098415	95
13	0.104465	55
14	0.122345	40
15	0.12449	40
16	0.125228	41
17	0.138935	41
18	0.13995	104
19	0.14093	41
20	0.146912	72
: :	:	:
N	T_n	B_n

Table 1: Sample of a trace file

The performance of the algorithm is depicted in Figures 2 to 10. These show how the bursty nature of the traffic is smoothed out by the algorithm. The length of packet sequences used for these experiments is N =65,536. We used various types of TCP data for the experiment, but due to space limitations we provide here results from using LBL-TCP3-packet, LBL-TCP4-packet and LBL-TCP5-packet data. The link capacities (i.e. desired rates) applied here are C = 5 Mbps, C = 10 Mbps and C =15 Mbps. Figure 11 illustrates the expected longest delay observed for different link speeds with the variation of length of packet sequences. It is clear from the Figures that higher capacity yields less delay and thereby provides better quality of service. Figure 12 shows the expected shortest and longest delay for different link speeds while the length of sequences is varied. The shortest delay is found to be from

0.000001 seconds to 0.000004 seconds. The longest delay is observed to be from 0.00056 seconds to 0.15927 seconds depending on the link speed (C) and length (N) of the packet sequences. The higher the link speed the shorter the observed delay.

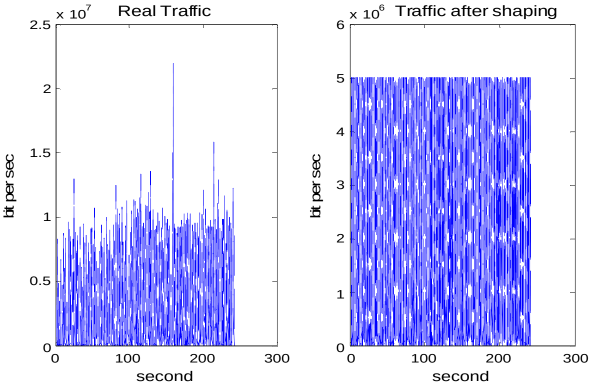


Figure 2: LBL-tcp3-pkt, C = 5 Mbps, N = 65536

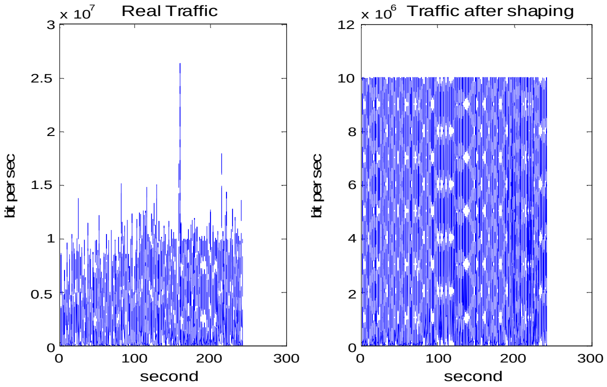


Figure 3: LBL-tcp3-pkt, C = 10 Mbps, N = 65536

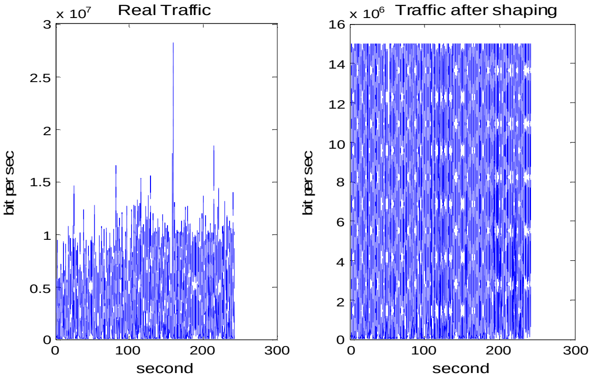


Figure 4: LBL-tcp3-pkt, C = 15 Mbps, N = 65536

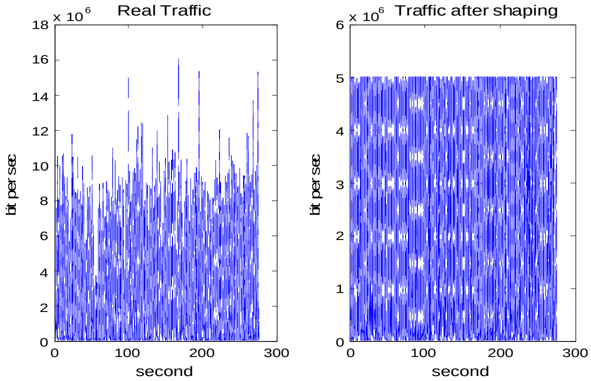


Figure 5: LBL-pkt-4_tcp, C = 5 Mbps, N = 65536

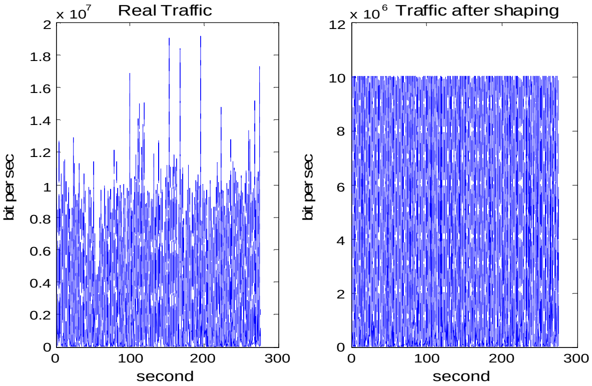


Figure 6 : LBL-pkt-4_tcp, C = 10 Mbps, N = 65536

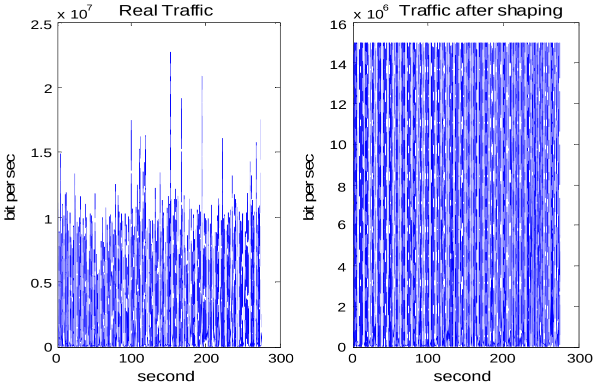


Figure 7: LBL-pkt-4_tcp, C = 15 Mbps, N = 65536

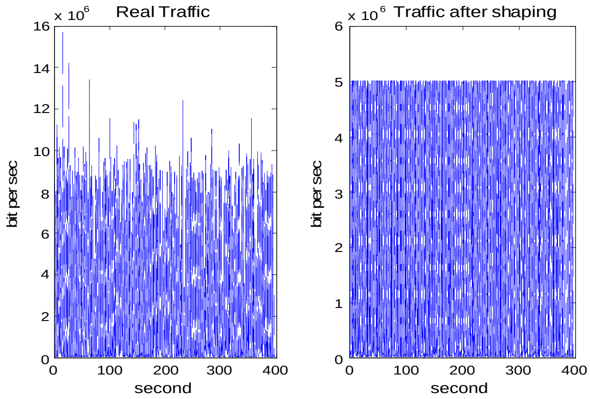


Figure 8: LBL-pkt-5_tcp, C = 5 Mbps, N = 65536

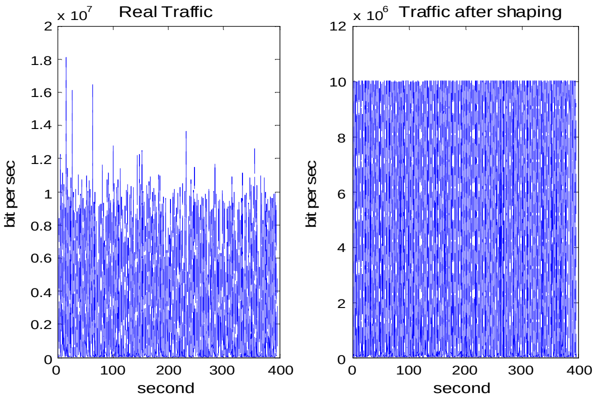


Figure 9: LBL-pkt-5_tcp, C = 10 Mbps, N = 65536

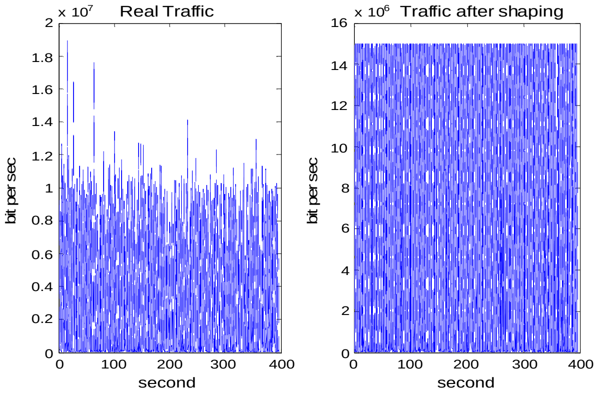


Figure 10: LBL-pkt-5_tcp, C = 15 Mbps, N = 65536

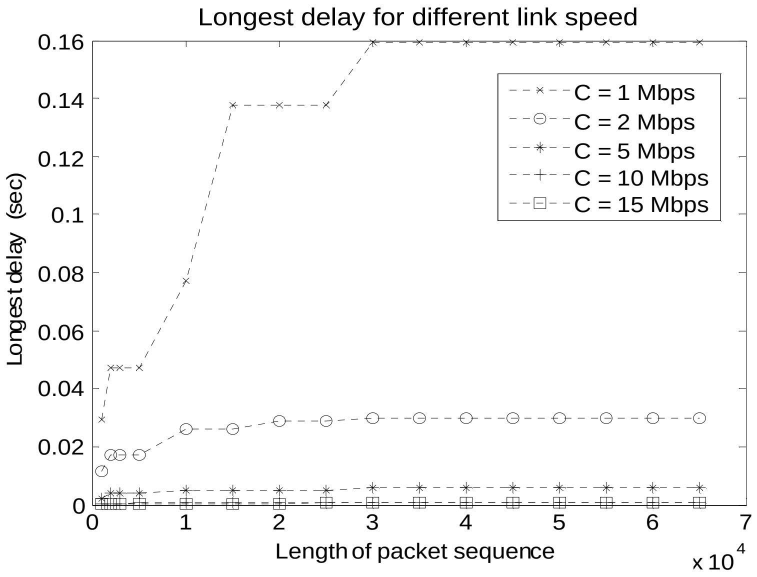


Figure 11: Performance of BPTraSha algorithm: Observation of longest delay. Variation of link speed with different length of packet sequences.

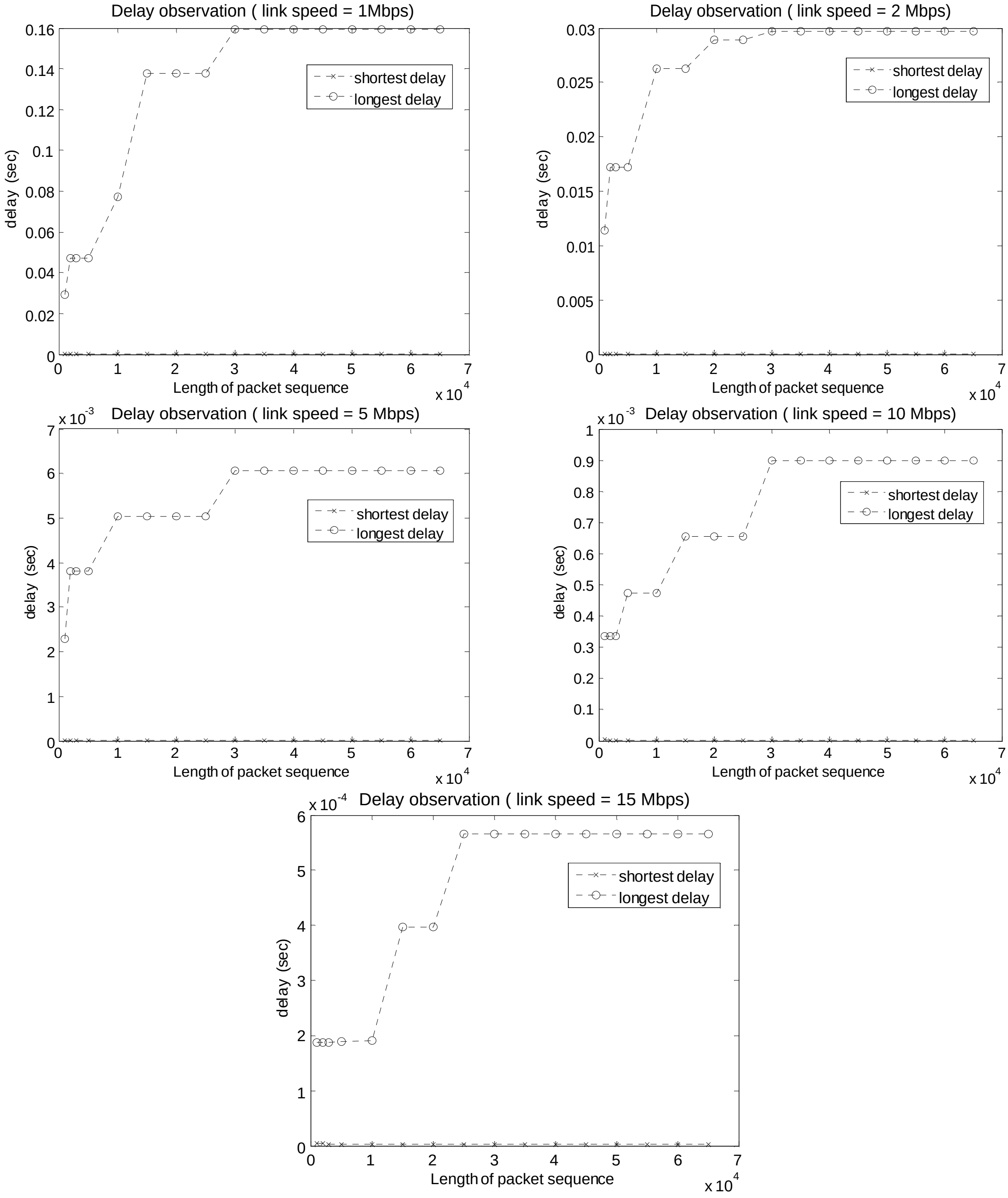


Figure 12: Performance of BPTraSha algorithm: Observation of shortest and longest delay, for different length of packet sequences.

5. Complexity of the algorithm, BPTraSha

To explore the complexity of BPTraSha, we chose six workstations with different specifications which are represented in Table 2. We investigated several lengths of packet sequences such as N = 1,000, N = 2,000, N = 3,000, N = 5,000, N = 10,000, N = 15,000, N = 20,000, N = 25,000, N = 30,000, N = 35,000, N = 40,000, N = 45,000, N = 50,000, N =

55,000, N = 60,000 and N = 65,000. In our research, we mainly concentrate on the time complexity of the algorithm.

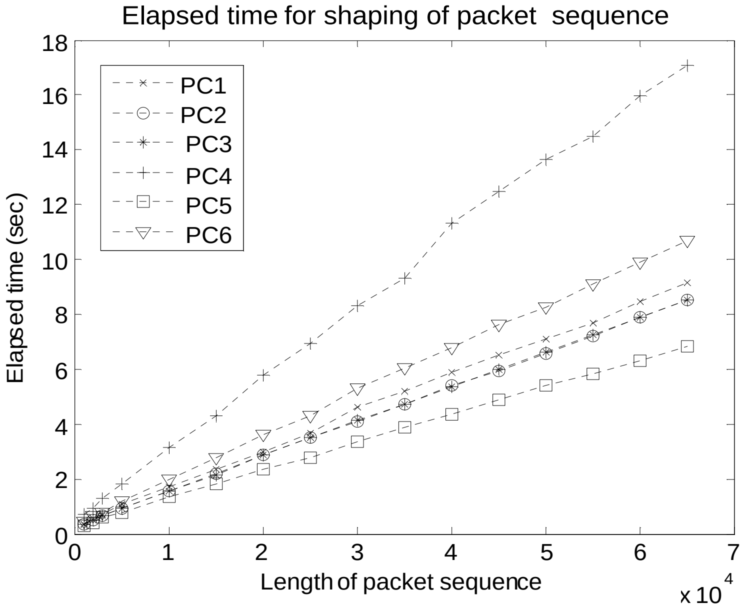


Figure 13: Observation of elapsed time for different length of packet sequences with different PC's

Figure 13 depicts the observed elapsed (execute) times for different lengths of packet sequences for different PCs. It is obvious that PC5 yields better performance as it possesses higher specifications. Figure 14 shows a percentage of affected packets due to delay for different length of packet sequences. Here, higher capacity (C) signifies better performance due to less affected packets. But the elapsed time for executing the algorithm does not significantly vary for different link speeds with the variation of length of packet sequences, which can be observed in Figure 15.

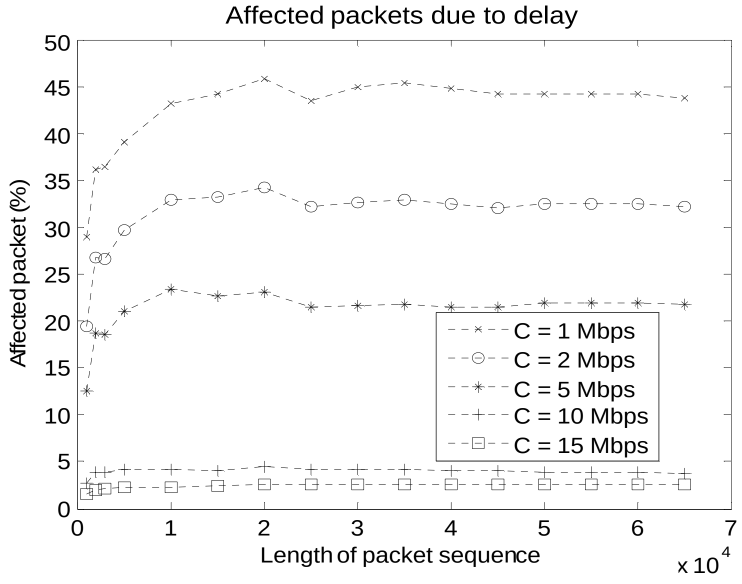


Figure 14: Affected packets due to delay for different length of packet sequences

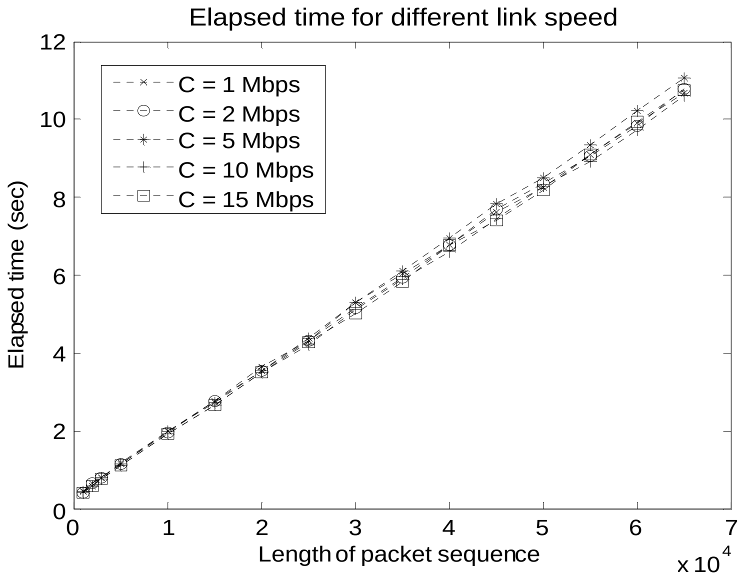


Figure 15: Elapsed time for different length of packet sequences with the variation of link speed.

Work station	Specification
PC1	Intel Pentium (R) 4, CPU 2.4 GHz, 512 MB of RAM
PC2	Intel Pentium (R) 4, CPU 3.0 GHz, 0.99 GB of RAM
PC3	Intel Pentium (R) 4, CPU 3.0 GHz, 504 MB of RAM
PC4	Intel Pentium (R) 3, CPU 866 MHz, 384 MB of RAM
PC5	Intel Centrino Duo Core, CPU T2250 @ 1.73 GHz, 1024 MB of RAM
PC6	Intel Pentium (R) 4, CPU 1.80 GHz, 256 MB of RAM

Table 2: Workstations with different specification

6. Checking LRD by BPTraSha algorithm

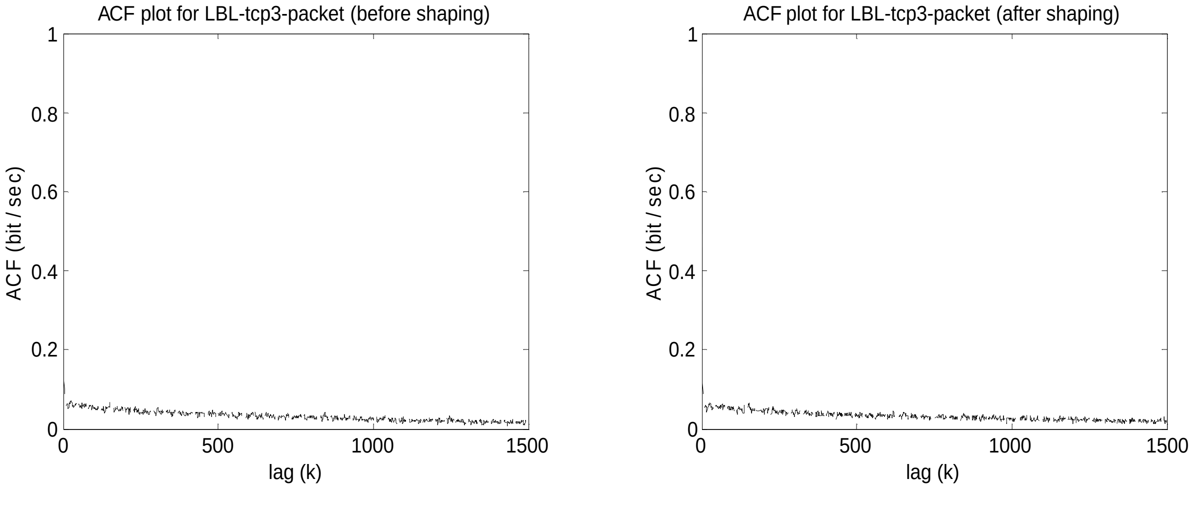


Figure 16: Autocorrelation function plot before and after shaping the traffic. LBL-TCP3-packet, C = 10 Mbps. Before shaping, H = 0.66 and after shaping, H = 0.64.

The autocorrelation function is a very useful tool in traffic engineering problems, especially when measuring the duration of existing traffic in the network. Figure 16 illustrates the ACF plot before and after shaping LBL-TCP3-packet traffic with link speed (C) at 10 Mbs. Note that the degree of LRD (i.e. the Hurst parameter) changes from 0.68 to 0.66 due to shaping the traffic. A similar scenario is observed for LBL-TCP4-packet and LBL-TCP5-packet traffic as shown in Figures 17 and 18.

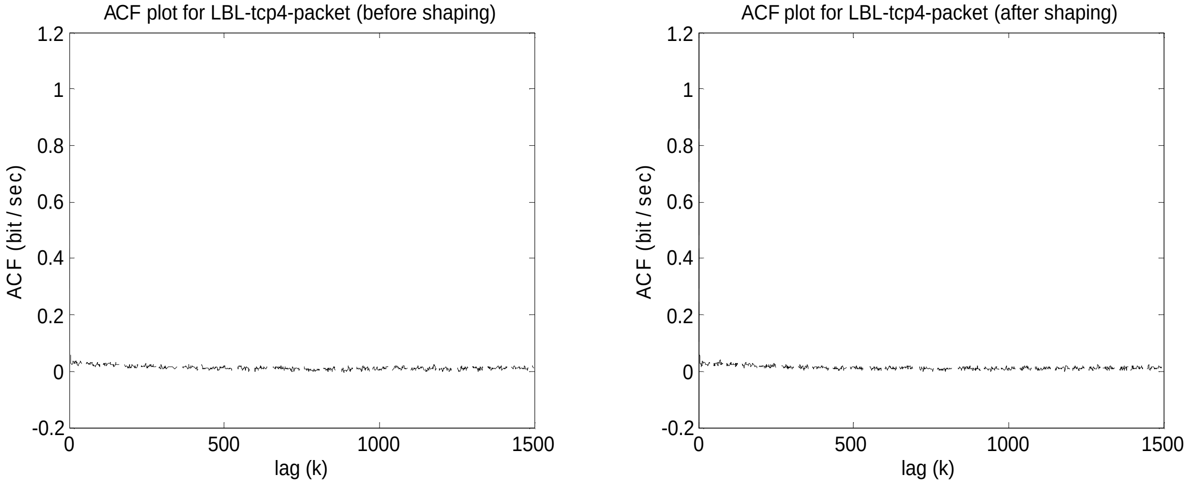


Figure 17: Autocorrelation function plot before and after shaping the traffic. LBL-TCP4-packet, C = 10 Mbps. Before shaping, H = 0.68 and after shaping, H = 0.66.

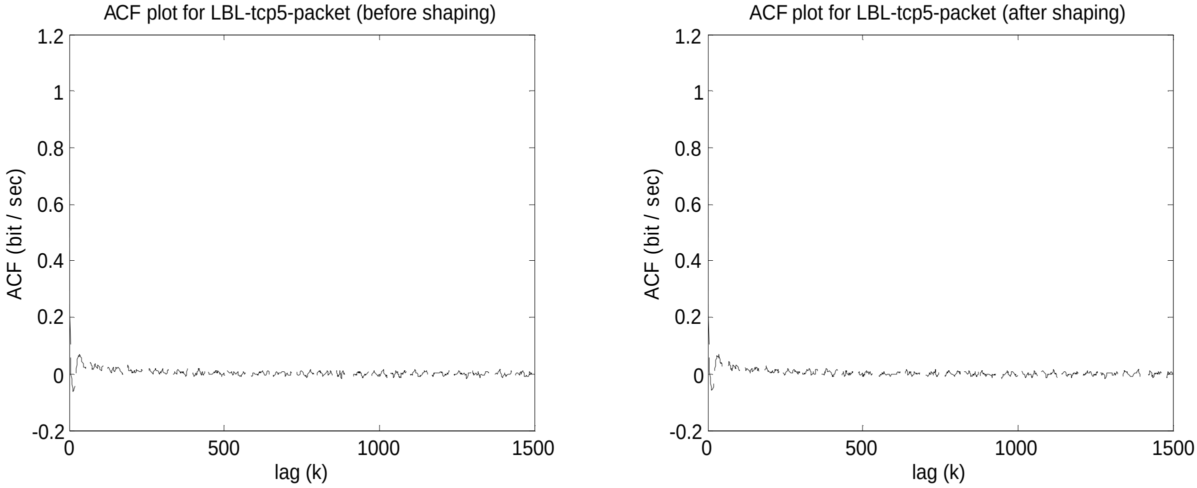
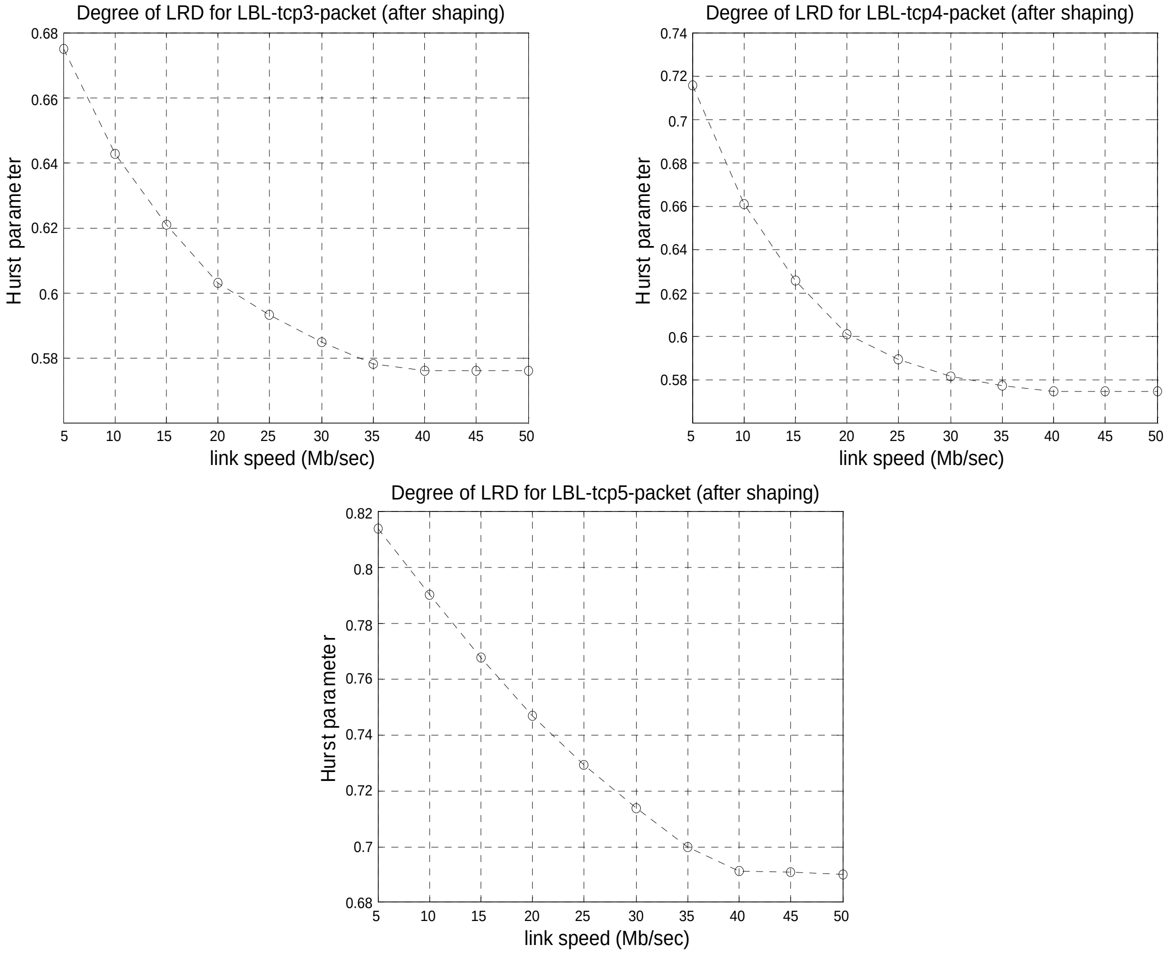


Figure 18: Autocorrelation function plot before and after shaping the traffic. LBL-TCP5-packet, C = 10 Mbps. Before shaping, H = 0.7968 and after shaping, H = 0.7905.

Figure 19 depicts the variation of the degree of LRD with different link speeds (C) while shaping the traffic. Before shaping, the estimates are H = 0.66, H = 0.68 and H = 0.7968 for LBL-TCP3-packet, LBL-TCP4-packet and LBL-TCP5-packet respectively. Clearly the Hurst parameter (H) decreases with increasing link speed (C), meaning that the long-range dependent traffic can be reduced by the BPTraSha algorithm. Note that at a certain limit (C = 40, 45 and 50 Mbps) the Hurst parameter remains unchanged, that is, no reduction of LRD is possible anymore. The Hurst parameter is estimated here by HEAF(2) (Rezaul *et al*., 2006 and Rezaul and Grout, 2007b).



Degree of LRD for LBL-tcp3-packet (after shaping)

Degree of LRD for LBL-tcp4-packet (after shaping)

Degree of LRD for LBL-tcp5-packet (after shaping)

Figure 19: Variation of the degree of LRD with different link speed (C). Before shaping, the estimated H = 0.66, H = 0.68 and H = 0.7968 for LBL-TCP3-packet, LBL-TCP4-packet and LBL-TCP5-packet respectively.

7. Conclusions and Future Work

In this research, we discuss a novel algorithm, BPTraSha, to control the bursty nature of network traffic. Experimental results show that the BPTraSha algorithm is capable of smoothing out the bursty nature of traffic packets received at the router buffer before they are transmitted to the core network (Internet). Also, it is clear from Figure 19, that LRD can be reduced by BPTraSha with increasing link speeds. As the main function of BPTraSha is to shape bursty packet traffic, it can contribute to reducing the network load and lead to the improvement of QoS in future Internet performance. Future work will include an evaluation of the applicability of the BPTraSha algorithm to real-time implementation at the FireBrick or router.

8. References

Aggarwal, A., Savage, S. and Anderson, T. (2000) "Understanding the Performance of TCP Pacing", *Proc. of the IEEE INFOCOM 2000 Conference on Computer Communications*, March 2000, pp. 1157 - 1165.

Brocklebank J. and Dickey, D. (1986) "SAS System for Forecasting Time Series", *SAS Institute Inc.* Cary NC. 1986.

Bushmitch, D., Panwar, S.S. and Pal, A. (2003) "Thinning, striping and shuffling: Traffic shaping and transport techniques for variable bit rate video", In *proceedings of the IEEE Globecom 2003*, vo.2, November 17-21, Taipei, pp.1485-1491.

Christensen, K. and Ballingam, V. (1997) "Reduction of Self-Similarity by Application-level Traffic Shaping", *In Proc. IEEE 22nd Annual Conference on Local Computer Networks*, November 1997, pp. 511 – 518.

Cox, D. (1984) "Long-Range Dependence: a Review", H. A. David and H. T. David (eds.), *In Statistics: An Appraisal, Iowa State Statistical Library*, The Iowa State University Press, 1984, pp.55-74.

Darlagiannis, V., Karsten, M., and Steinmetz, R. (2003) "Burst Shaping Queueing", *In Computer Networks and Distributed Systems (WMC)*, SCS, January 2003, pp. 65-70.

Guo, L., M. Crovella, and Matta, I. (2000) "TCP congestion control and heavy tails", *Technical Report*: BUCSTR -2000-017, Computer Science Dept - Boston University, 2000.

Guo, L., M. Crovella, and Matta, I. (2001) "How does TCP Generate Pseudo-Self-Similarity?", *In Proceedings of the International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunications Systems (MASCOTS~'01)*. Cincinnati, Ohio, pp. 215-223.

Harmantzis F.C., Hatzinakos D. and Katzela I. (2001) "Shaping and policing of fractal α-stable broadband traffic", *Proc. Canadian Conf. on Elec. and Comp. Engineering (CCECE)*, Toronto, Canada, May 2001, pp. 697-202.

ITA (2002), *Internet Traffic Archive*: http://ita.ee.lbl.gov/html/traces.html, (visited on 30th March 2002.)

Leland W.E., Taqqu, M.S., Willinger, W. and Wilson, D.V. (1994) "On the Self-similar nature of Ethernet Traffic (Extended version)", *IEEE/ACM Transactions on Networking*, February 1994, Vol. 2, No. 1, pp. 1-15.

Molnár, S. and Vidács, A. (1997) "On Modeling and Shaping Self-Similar ATM Traffic", *Proc. in 15th International Teletraffic Congress (ITC15)*, Washington, USA, July, 1997.

Neidhardt, A.L. and Erramilli, A. (1996) "Shaping and policing of fractal traffic", *In 10th ITC Specialists Seminar on Control in Communications*, 1996, pp. 253-264.

Park, K. Kim, G. and Crovella, M. (1997) "On the effect of self-similarity on network performance", *In Proceedings of the SPIE International Conference on Performance and Control of Network System*, November 1997, pp 296-310.

Peha, J.M. (1997) "Protocols can make traffic appear self-similar", *In Proceedings of the 1997 IEEE/ACM/SCS Comm. Networks and Distributed System. Modeling and Simulation Conference*, Jan 1997, pp. 47-52.

Pruthi, P. and Popescu, A. (1997) "Effect of Controls on Self-Similar Traffic", *In Proceedings of the 5th IFIP ATM Workshop*, Bradford, UK, July 1997.

Rezaul, K.M. and Grout, V. (2007a) "BPTraSha: A Novel Algorithm for Shaping Bursty Nature of Internet Traffic", *Proceedings of the 3rd IARIA/IEEE Advanced International Conference on Telecommunications (AICT 2007)*, May 13-19, 2007, Mauritius.

Rezaul, K.M. and Grout, V. (2007b) "Exploring the Reliability and Robustness of HEAF(2) for Quantifying the Intensity of Long-Range Dependent Network Traffic", *International Journal of Computer Science and Network Security*, Vol. 7, No. 2, February 2007, pp. 221-229.

Rezaul, K.M., Pakštas, A.., Gilchrist, R. and Chen, T.M. (2006) "HEAF: A Novel Estimator for Long-Range Dependent Self-similar Network Traffic", Y. Koucheryavy, J. Harju, and V.B. Iversen (Eds.): *Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN)*, May 29 - June 2, 2006, LNCS 4003, pp. 34 – 45.

Sikdar, B. and Vastola, K.S. (2001) "On the Contribution of TCP to the Self-Similarity of Network Traffic", Lecture Notes In Computer Science, *Proceedings of the Thyrrhenian International Workshop on Digital Communications: Evolutionary Trends of the Internet*, Vol. 2170, 2001, Springer-Verlag, London, UK, pp.596-613.

Turner, J. (1986) "New directions in communications, or which way to the information age?", *IEEE Communications Magazine*, vol. 24, 1986, pp. 8-15.

Vamvakos, S. and Anantharam, V. (1998) "On the departure process of a leaky bucket system with long-range dependent input traffic", *Queuing Systems: Theory and Applications*, vol. 28, 1998, pp. 191-214.

Veres, A., Kenesi, Z., Molnár, S. and Vattay, G. (2003) "TCP's Role in the Propagation of Self-Similarity in the Internet", *Computer Communications, Special issue on Performance Evaluation of IP Networks and Services,* Vol. 26, Issue 8, May 2003, pp. 899-913.

Willinger, W., Paxson, V. and Taqqu, M. (1998) "Self-similarity and Heavy Tails: Structural Modeling of Network Traffic", Adler, R., Feldman, R., and Taqqu, M.S., (editors), *In A Practical Guide to Heavy Tails: Statistical Techniques and Applications*, Birkhauser, 1998.

Xue, F. and Yoo, S.J.B. (2002) "Self-similar traffic shaping at the edge router in optical packet-switched networks", *Proc. IEEE International Communication Conference (ICC 2002)*, vol.4, April 28- May 2, New York, 2002, pp.2449-2453.