

Effect of Ingress Buffering on Self-Similarity of Optical Burst Traffic

Rui Huang and Gergely V. Záruba

Computer Science and Engineering Department
The University of Texas at Arlington
416 Yates, 300NH, Arlington, TX 76019, USA

ABSTRACT

Recently, optical burst switching and aggregated optical packet switching have gained significant exposure as possible future mechanisms for routing aggregated IP traffic over all-optical core networks. However, the limited buffering capacity in all-optical networks presents a major challenge, as current IP traffic displays strong self-similar properties. Reducing the burst loss rate of such long-range dependent traffic can be costly requiring a significant increase in either the network bandwidth or the buffer size of optical cross connects. In this paper, we revisit the possibility of using buffers to reduce self-similarity before the traffic is routed onto the all-optical core. The aim of this paper is to increase the understanding of the effect of packet/burst aggregation on the self-similarity measure of the traffic. In particular, we implement a simple burst assembly mechanism with two parameters, the maximum burst length L and the maximum burst delay d , so that incoming traffic is smoothed with a guaranteed delay bound. Unlike previous works, we simulate the burst assembler using more realistic input traffic sources, and analyze the results using both R/S plot and discrete wavelet analysis methods. Our detailed results show that buffering indeed reduces traffic self-similarity (an area of research controversy) when parameters L and d are set appropriately.

Keywords: optical packet/burst switching, traffic analysis, self-similarity

1. INTRODUCTION

Self-similarity (or fractal behavior) refers to a common phenomenon in nature where objects or processes retain certain characteristics regardless of the observation's scale. In the domain of network engineering, it has been discovered that Internet traffic is self-similar in the sense that the traffic displays a similarly ragged pattern (burstiness) when observed during smaller time intervals (such as hours) as well as during longer intervals such as days.¹ More precisely, it has been shown that the traffic shows bursty behavior indifferent of the time scale. This finding is somewhat against the common intuition that the raggedness of a traffic pattern during smaller time intervals can be smoothed with buffering at longer intervals.

Mathematically, a continuous-time process $Y(t)$ is defined as self-similar if for all $a > 0$ and $t \geq 0$,

$$Y(t) = a^{-H} Y(at).$$

Here, the variable $H(0 \leq H \leq 1)$, the Hurst parameter, determines the amount of self-similarity. In particular, when $H = 0.5$, $Y(t)$ becomes an uncorrelated Brownian Motion process. When $0.5 \leq H \leq 1$, the process implies a long-range dependence and becomes self-similar. The degree of self-similarity increases with H .

The self-similar phenomenon makes network engineering more difficult because most networks have been previously modeled by Poisson arrival processes instead of self-similar processes. In particular, while a queuing system based on Poisson arrival processes is capable of exponentially reducing the queue length with increasing buffer sizes, such assumption will not hold for self-similar arrival processes. This indicates that increasing the forwarding buffer size has significantly less effect on self-similar traffic than on a Poisson process modeled traffic.

Further author information: (Send correspondence to Gergely V. Záruba)

Rui Huang: E-mail: rxh1725@omega.uta.edu,

Gergely V. Záruba: E-mail: zaruba@uta.edu, Address: 416 Yates, 300NH, Arlington, TX 76019, USA

In other words, to obtain the same packet blocking rate for self-similar and Poisson traffics, a significantly larger buffer size is needed for the former. In all-optical burst and packet switched networks, the effect created by this increased buffer size requirement is more significant since ‘buffers’ are commonly implemented using ‘inflexible’ fiber delay lines. Therefore, it would be particularly cost-effective if the amount of self-similarity could be reduced before traffic is routed onto the optical network core.

The two main proposed approaches for switching IP packets directly over optical networks are optical packet switching (OPS) and optical burst switching (OBS). To reduce the contention between optical packets (or bursts) and to enable easy buffering, OPS networks are likely to be synchronized (all inputs of the optical cross connects are synchronized amongst themselves) with time slots corresponding to a maximum (optical) transmission unit. To efficiently use the bandwidth defined by these time slots, packets going towards the same direction may be aggregated into the same slot (optical packet) at the ingress cross connects. We will refer to such bursts as aggregated packets to avoid confusion with burst switching. In OBS, the extent of packet aggregation is usually greater, i.e., there are more packets ‘stuffed’ into a burst. Additionally, in OBS the routing and forwarding functions of switches are separated, i.e., packet forwarding is not based on a packet header but on a rapidly switched circuit (lightpath). Yet, both OPS and OBS aggregate packets into bursts thus the burstiness of traffic may be affected similarly by the burst assembly mechanisms.

It has been suggested that the buffering scheme implemented for accumulating/creating optical bursts at edge routers can effectively reduce the self-similarity of the burst traffic compared to an unbuffered traffic case.² Here, the buffering scheme contains a maximum delay parameter to provide an upper bound of the queuing delay for bursts; input traffic is set to be self-similar using a Pareto distributed on-off traffic flow with a Hurst parameter $H = 0.9$. The corresponding Hurst value of the buffer output is measured using R/S plots.¹ The simulation consists of two scenarios when output burst size is 25 times and 50 times of the average packet size. The results show that H is less than 0.9 for both scenarios, and that H can be further reduced with increasing the burst size (i.e., the buffer size).

Contradictory to such findings, other experiments indicate that buffering is not effective in reducing traffic self-similarity.³ In those experiments, the effect of buffering using both Poisson and self-similar traffic is studied. For self-similar traffic sources, the buffering scheme is different in that it contains an additional burst length limiting parameter L . When a burst reaches the maximum size - L - it is transmitted immediately regardless of the delay of the first packet in the burst. The Hurst parameter H is chosen to be 0.75 for the input traffic, slightly different from the well-accepted 0.9 value. The simulation examines the Hurst value of bursts using R/S plots under two different extreme load conditions. The results show that the buffering scheme is only effective in reducing self-similarity with extreme heavy loads (100 percentile), but it is ineffective for light loads (1 percentile).

Concerned with the controversy^{2,3} regarding the effectiveness of buffering on self-similarity, this paper aims to attack the problem using different methods for measuring burstiness of aggregated traffic. Our goal is to provide with further insights to the truth of the above controversy by a thorough, well-explained and extensive simulation study. Since previous works do not always describe the complete set of assumptions, network and simulation parameters, it is not possible to compare them directly. Instead, this paper opts to document the complete simulation so that results can be easily duplicated elsewhere. Furthermore, comparing our work to previous studies, we adopt a more realistic traffic source that follows the packet size distribution from a recent empirical study.⁴ While previous works compare self-similarity solely based on the Hurst values obtained by R/S plots, we incorporate additional approximations based on a frequency domain conversion technique, i.e., discrete wavelet analysis.⁵ To eliminate the possible bias caused by the packet loss at the buffer itself, we assume the availability of nearly infinite buffers (with no packet loss induced due to insufficient buffer size) so that the pure effect of buffering on traffic self-similarity can be obtained. Our simulation results show that traffic self-similarity can be generally reduced by burst buffering under our assumptions.

The remainder of the paper is organized as follows. Section 2 introduces the network model including the input traffic’s distribution and the burst assembly policy. Section 3 describes the two techniques of measuring and comparing the self-similarity of output bursts: the R/S plot (3.1) and the discrete wavelet analysis (3.2). Results from our simulation studies are presented in Section 4 along with a detailed description of the simulation parameters. Section 5 concludes the paper and identifies future research directions.

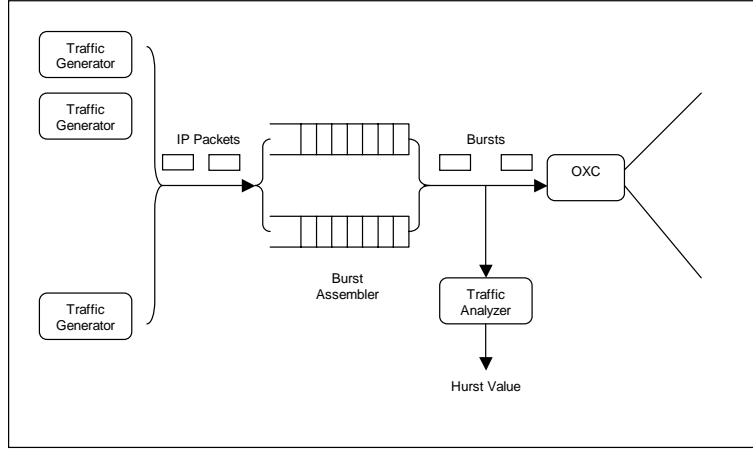


Figure 1. Network model to analyze the self-similarity of optical bursts.

2. NETWORK MODEL

Figure 1 depicts the basic model used in this paper to analyze self-similarity of the optical burst traffic. A set of traffic generators are employed to simulate end users. Each traffic generator generates the incoming IP traffic using a Pareto distributed on/off model.⁶ The IP traffic is routed to a burst assembler, which is responsible of assembling the IP packets into optical bursts, based on a certain scheduling policy. After the bursts are assembled, they are forwarded onto the all-optical network via the local Optical Cross Connect (OXC). A traffic analyzer intercepts the optical bursts and analyzes the self-similar properties of the traffic. The analyzer performs two types of analyses: 1) rescaled adjusted range plot (R/S plot) analysis and 2) discrete wavelet analysis. The results from the analyzer can then be compared to study the impact of burst assembly on the traffic self-similarity.

To capture the characteristics of actual Internet traffic more realistically, we designed a on/off traffic flow model where the *on* and *off* durations are governed by heavy-tailed Pareto distributions with parameter $\alpha = 1.2$.⁶ The cumulative density function (CDF) of the Pareto distribution is given as

$$D(x) = 1 - \frac{1}{x^\alpha}$$

By aggregating those independent Pareto-distributed traffic sources, self-similar traffic can be generated.⁶ The Hurst parameter is related to the alpha value of the source Pareto distribution in that $H = (3 - \alpha)/2$. During (such I.I.D. Pareto distributed) *on* flows, IP packets are generated with an exponentially distributed inter-arrival time with a rate of λ . During an *off* flow, no traffic is generated and the line is considered idle. More precisely, the average duration of *on* and *off* flows is set to the same value, i.e., a value in which a Pareto distributed random amount of (exponentially distributed) packets can be generated. The destination of the IP packets is selected I.I.D. uniform randomly over a total number of $V - 1$ OXC destinations for an all-optical mesh network with V number of OXCs. The size of IP packets follows the characteristics documented in the latest research on Internet traffic trends.⁴ In particular, it is found that the most common sizes are of 44, 552, 576, and 1500 bytes. The packets of 40-44 bytes are mostly TCP control packets such as TCP acknowledgments, SYN, FIN, and RST packets. Telnet sessions also generate small packets since each keystroke is sent as an individual packet. The 552 to 576 bytes sized packets are generated by the default maximum segment size (MSS) for non-local IP destinations, while 1500 bytes is the maximum transmission unit (MTU) of Ethernet. MTU sized packets are usually generated by a session that involves file transmission such as FTP or HTTP. Statistically, the average packet size is 403 bytes. Based on the findings, we devised a simplified distribution that closely resembles the actual distribution of the empirical data. Figure 2 shows the simplified packet size distribution used in this work.

The burst assembler consists of $V - 1$ number of queues, one for each of the destination OXCs. The IP packets coming from the generators are inserted into one of these First-Come First-Serve (FCFS) queues based on their

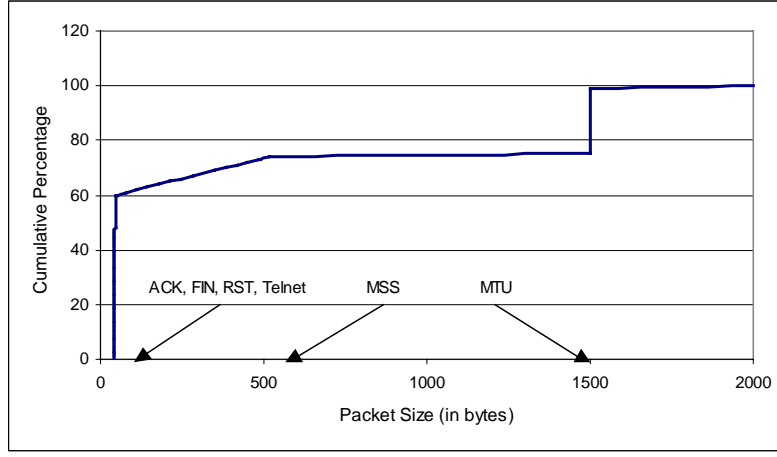


Figure 2. Cumulative distribution of packet sizes.

destination address. The behavior of the burst assembler is controlled by two threshold values: d (maximum queuing delay) and L (maximum burst size), i.e., bursts are assembled using the following scheduling policy. When the total (accumulated) size of the IP packets for a particular destination has reached the threshold L , the packets will be de-queued and assembled into an optical burst. Alternatively, if the wait time of the first (at the head of the queue) IP packet in the queue has reached the threshold value d , all packets in the queue will be assembled as in the other case, even though the capacity of the burst might not be at its maximum. Once a burst has been assembled, it will immediately be forwarded on to the local OXC to be transmitted through the all-optical network. Since our study is restricted to the impact of burst assembly to the self-similarity of traffic, we will assume that the bandwidth of the local OXC is large enough to handle all incoming bursts so that no burst is dropped at the burst assembler. Obviously, the thresholds d and L within the burst assembler represent a performance tradeoff between the transmission delay and the bandwidth utilization. We will find that d and L also affect the self-similarity of the traffic as demonstrated in the following section.

3. HURST PARAMETER APPROXIMATION

The self-similarity of the arrival of assembled bursts is examined and measured by the traffic analyzer (see Figure 1). In particular, two types of analyses are performed to approximate the Hurst parameter of assembled bursts (indicating the extent of self-similarity). Both methods evaluate the traffic self-similarity by examining and comparing the traffic trace on both large and small time scales. This section is devoted to describing the two Hurst approximation methods in detail.

3.1. Rescaled Adjusted Range Plot

The rescaled adjusted range plot (R/S plot) is a commonly used method to detect long-range dependency within a system, and to derive an approximate value for the Hurst parameter.⁶ To produce the R/S plot over a traffic trace, we first need to identify our observation data. Here, we define the observation data as the burst arrival rate for the instance. We use a predefined number of timeslots ts and designate the *actual* number of burst arrivals at each time slot to be our observation. Thus, there are total of ts observations for each simulation run. To eliminate the possible inaccuracy resulting from the simulation start/stop (transients), we discard the first 100 and the last 100 observations and adjust the ts value accordingly. Let us label the remaining observations X_p , where $p = 1, 2, \dots, ts$.

The method of finding the R/S plot of the derived observation data works as follows.¹ The set of observed data X_p is divided into K non-overlapping buckets with size n each, so that the rescaled adjusted range $\frac{R(t_i, n)}{S(t_i, n)}$

can be calculated for each bucket with t_i being the starting position of the i^{th} bucket in X_p . To calculate $\frac{R(t_i, n)}{S(t_i, n)}$, we use the following equation

$$\frac{R(t_i, n)}{S(t_i, n)} = \frac{1}{S(t_i, n)} (\max(0, W_{t_i}, W_{t_i+1}, \dots, W_{t_i+n-1}) - \min(0, W_{t_i}, W_{t_i+1}, \dots, W_{t_i+n-1})),$$

where

$$W_{t_i+j} = (X_{t_i} + X_{t_i+1} + \dots + X_{t_i+j}) - j\bar{X}(t_i, n).$$

Here, $\bar{X}(t_i, n)$, and $S^2(t_i, n)$ are the sample mean and sample variance of the sequence of n observations $X_{t_i}, X_{t_i+1}, \dots, X_{t_i+n-1}$, respectively. We can now calculate $\frac{R(t_i, n)}{S(t_i, n)}$ for each of the K non-overlapping buckets, i.e., for $i = 1, n+1, 2n+1, \dots, (K-1)n+1$. These results (from all K buckets) are averaged to produce a single value for the bucket size n , i.e.,

$$\frac{R(n)}{S(n)} = \frac{1}{K} \sum_i \frac{R(t_i, n)}{S(t_i, n)}.$$

The base-10 logarithm of the produced value is then plotted versus the base-10 logarithm of n to produce a single point in the R/S plot. By repeating the above procedure for different bucket sizes ($1 \leq n \leq ts$) a complete R/S plot is generated. To ensure that all points in the R/S plot have the same weight (when performing the linear regression as described below), we increase the bucket size by $10^{0.11}$ at each step until n reaches the maximum bucket size ts . Thus, after taking the logarithm, the points in the final R/S plot are evenly distributed over the x -axis.

The Hurst parameter can be derived from the R/S plot generated earlier. By the Hurst law, the Hurst parameter, H , is related to the R/S values before rescaling with the logarithms in $E[\frac{R(n)}{S(n)}] \propto cn^H$, where c is a constant, independent of n .⁷ H can be approximated by the slope of the line formed by the resultant R/S plot after applying the base-10 logarithms. To locate the linear (and its slope), we use a simple linear regression method by finding the least square fit of the plot.⁸ Once the line is located, the slope of the line provides an approximation to the Hurst parameter of the optical burst traffic. Using the least squares fit method, the approximated Hurst parameter, \hat{H} , is estimated as

$$\hat{H} = \frac{b \sum_n \log(n) \log\left(\frac{R(n)}{S(n)}\right) - \sum_n \log(n) \sum_n \log\left(\frac{R(n)}{S(n)}\right)}{b \sum_n \log^2(n) - \left(\sum_n \log(n)\right)^2},$$

where b is the total number of points in the plot for the linear regression. The slope of the R/S plot becomes stable as n increases. Therefore, only the points with $\log(n)$ ranging from 2 to 6 are selected to perform the regression. Since $\log(n)$ is incremented by 0.1 at each step, there are a maximum of 40 points used in the linear regression, i.e., $b = 40$.

3.2. Discrete Wavelet Analysis

Wavelet analysis allows time sequenced data to be analyzed in the frequency domain. Wavelet analysis involves wavelet transformation, a tool that has been widely applied in applications such as fault diagnosis and image compression. The technique has been applied to self-similarity analysis.⁵ Using wavelet analysis, the Hurst parameter can be approximated given the time sequenced data. It has been shown that discrete wavelet analysis tends to approximate Hurst values with a higher accuracy than the R/S plot method.⁵

Similarly to the R/S plot analysis, the first step of the discrete wavelet analysis is to define the observation data. Again, we use the burst arrival rate within each timeslot as our observation. Let ts be the number of predefined timeslots. Thus, each timeslot is $\frac{T}{ts}$ seconds long for a total simulation duration of T seconds. Let the actual number of burst arrivals be our observation, while discarding the first and the last 100 observations (for the same stability reasons as with R/S plots). The remaining observations X_p , where $p = 1, 2, \dots, ts$, will be used in the wavelet analysis.

We can now apply a discrete wavelet transformation (DWT) to our observation data, which analyzes the data at different resolutions. Essentially, DWT allows the original observation data to be described by the relationship between the approximation of a finer and of a coarser resolution. More precisely, the approximation at resolution j contains twice as much detail as the approximation at $j + 1$. For a total of J number of resolutions, the original data point X_p can be approximated using DWT as

$$X_p \approx approx_0(p) = approx_J(p) + \sum_{j=1}^J detail_j(p).$$

Here, X_p is represented by the approximation of resolution 0 (highest), which can be re-formulated as the sum of the approximation of a lower resolution J and all the detailed *changes* in the approximations as the resolution becomes finer. For our purpose of analyzing self-similarity, we are interested in comparing $detail_j(p)$ at various resolutions.

Using DWT, the above terms can be formally defined in the following way:

$$approx_J(p) = \sum_k a_X(J, k) \phi_{J,k}(p);$$

$$detail_j(p) = \sum_k d_X(j, k) \psi_{j,k}(p),$$

where $\phi_{J,k}(p)$ and $\psi_{j,k}(p)$ are derived from the *scaling function* ϕ_0 and the *mother wavelet* ψ_0 in that

$$\phi_{j,k}(p) = 2^{-j/2} \phi_0(2^{-j}p - k);$$

$$\psi_{j,k}(p) = 2^{-j/2} \psi_0(2^{-j}p - k).$$

For each resolution j and the k^{th} approximated data point within resolution j , the DWT algorithm generates the coefficients $a_X(j, k)$ and $d_X(j, k)$ as the output. The actual choices of ϕ_0 and ψ_0 are defined by the choice of the wavelet. We use the orthogonal Daubechies wavelet⁹ that is constructed via the lifting scheme.¹⁰ The mother wavelet ψ_0 is recursively defined from the Daubechies scaling function ϕ_0 in that

$$\psi_0(p) = -\frac{1+\sqrt{3}}{4} \phi_0(2p-1) + \frac{3+\sqrt{3}}{4} \phi_0(2p) - \frac{3-\sqrt{3}}{4} \phi_0(2p+1) + \frac{1-\sqrt{3}}{4} \phi_0(2p+2).$$

The actual DWT is implemented using a fast filter-bank based pyramidal algorithm.⁶ For each resolution $j : J \rightarrow 1$, the DWT algorithm calculates the detailed change $d_X(j, k)$. Let n_j be the total number of coefficients $d_X(j, k)$. At a resolution of j , the logarithmic average of $d_X(j, k)$ can be given as:

$$\mu_j = \log_2 \left(\frac{1}{n_j} \sum_{k=1}^{n_j} |d_X(j, k)|^2 \right).$$

μ_j can be plotted against j to illustrate the amount of self-similarity (similar to the R/S plot). The slope of such plot is $\alpha = 2H - 1$. The Hurst parameter, H , can be approximated from the plot using the weighted linear regression technique⁸:

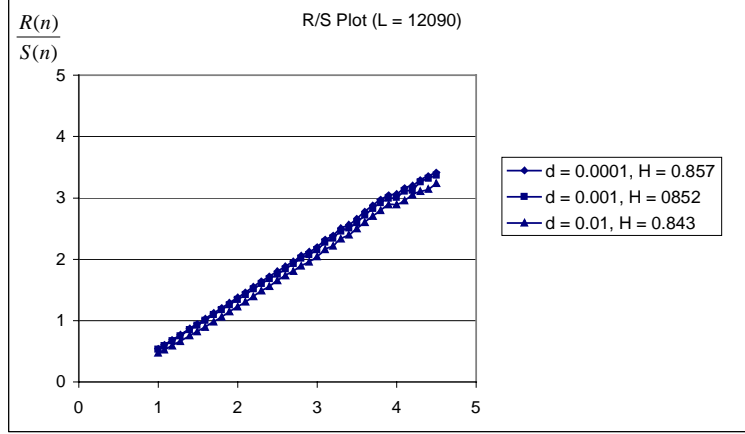
$$\hat{H} = \frac{1}{2} \left[\frac{\sum_j S_j j \mu_j \sum_j S_j - \sum_j S_j j \sum_j S_j \mu_j}{\sum_j S_j \sum_j S_j j^2 - \left(\sum_j S_j j \right)^2} + 1 \right],$$

where the weight $S_j = (n_j \ln^2 2) / 2^{j+1}$ is the inverse of the variance of μ_j .

The actual regression is performed over a selected range of resolution j . Note that the regression here uses \log_2 based data, whereas the regression in the R/S plot is \log_{10} based. To be consistent with R/S plot, we select the points within the range of $j \in [5.0, 16.0]$ when performing the weighted regression.

Table 1. Simulation Parameters

Simulation time T	3600 seconds
Hurst index of <i>on</i> flow α_{on}	1.2
Hurst index of <i>off</i> flow α_{off}	1.2
Packet arrival rate during <i>on</i> flows	625 packets per seconds
Total number of destinations V	13
Maximum burst length	403 ~ 28210 bytes
Maximum queuing delay	0.0001 ~ 0.01 seconds
Number of timeslots ts	36000
R/S plot bucket size n	1, $\lfloor 10^{0.1} \rfloor$, $\lfloor 10^{0.2} \rfloor$, ..., ts
Effect range for R/S regression	[2.0, 6.0], \log_{10} -based, unweighted
Wavelet type for DWT	Daubechies
Effect range for DWT regression	[5.0, 16.0], \log_2 -based, weighted

**Figure 3.** R/S plot with varying d when $L = 12090$.

4. SIMULATION RESULTS

Table 1 lists the simulation parameters used by us so that the experiments can be easily duplicated elsewhere (the authors will be happy to supply the simulator to interested parties upon request). The input traffic pattern is generated by 80 traffic generators, each of which independently generates heavy-tailed packet flows using the on/off model with $\alpha = 1.2$, providing a Hurst value of 0.9 to the aggregated input traffic. Within each *on* flow, the packet arrival rate λ is set to 625, and the size of each packet follows the distribution given in Figure 2. The burst assembler aggregates the packets into optical bursts with a maximum queuing length of L and a maximum queuing delay of d . For each burst size and assembly timeout combination, the simulator generates the optical bursts for a total of 3600 seconds of simulation time.

We first show the actual data plots generated by the R/S plot and DWT analysis described earlier. Figure 3 and 4 show the plots for the cases when the maximum burst length $L = 12090$ and the maximum queuing delay d varies from 0.0001 to 0.01. Both figures suggest that there are enough data points available to produce fairly accurate Hurst value approximation. For the R/S plot, the approximated Hurst value is simply the slope of the data trace. For the DWT plot, the slope is α , and H can be easily derived since $\alpha = 2H - 1$. In the DWT plot, the data points tend to diverge from the trace when j increases, i.e., when the resolution becomes coarser. But because of the weighted scheme used during the linear regression, the divergence does not greatly affect the overall approximation. The data points of a larger j (coarser resolution) are weighted less than those of a smaller j (finer resolution).

We now describe our results obtained by the R/S plot technique in term of the Hurst value. Figures 5 and 6 show that in general the Hurst value decreases with an increasing maximum queuing delay (d) and/or maximum

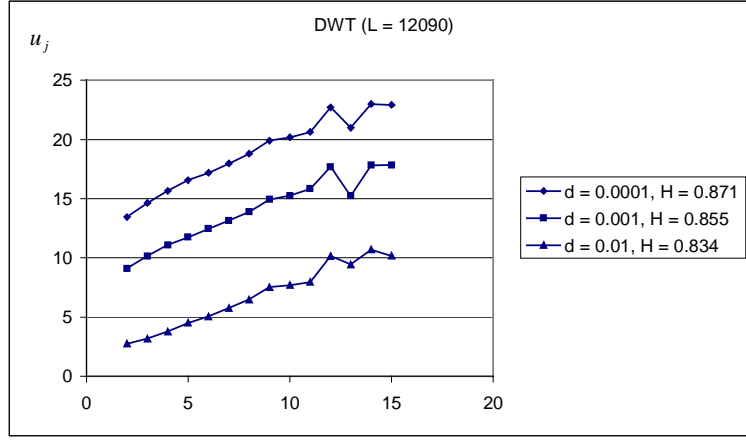


Figure 4. DWT plot with varying d when $L = 12090$.

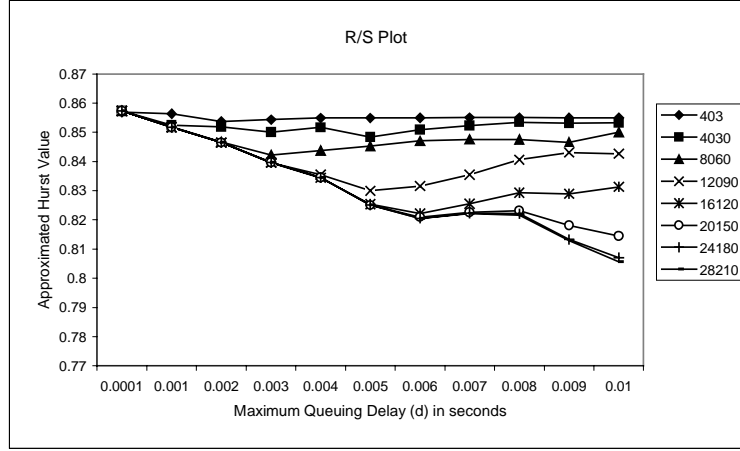


Figure 5. Hurst value approximated using R/S plot by varying d .

burst length (L). Note that the largest Hurst value occurs when the two buffering thresholds $d = 0.0001$ seconds and $L = 403$ bytes (the average packet size) are essentially equivalent to no buffering at the assembler. In fact, as indicated by the near flat line of $L = 403$ in Figure 5 and that of $d = 0.0001$ in Figure 6, virtually no packets are buffered when *either* threshold is small enough to dominate the assembler's behavior. Larger bursts are generated with larger d and L values, reducing the Hurst value of the output traffic. The lines in Figure 6 eventually become flat as the maximum allowable buffer size L becomes so large that lesser number of optical bursts are generated with the maximum size, and/or the delay threshold d dominates the assembler behavior. Overall, the results from the R/S plot analysis of Figure 5 and 6 show that the self-similarity of traffic can be reduced when comparing to no buffering as the thresholds d and L at the burst assembler work together to generate larger optical bursts. These results are in-line with the findings by Ge.²

Our goal with performing similar simulations with an entirely different Hurst approximation method is to clear any doubts about the validity of the impact of burst assembly on self-similarity. Figures 7 and 8 show results of the DWT Hurst approximation approach. As it can be observed the DWT approach indeed validates the results obtained the R/S plot experiments of ours and that of Ge.² The almost perfect overlap between the results of the two methods strongly indicate the existence of burst aggregation effects on traffic self-similarity.

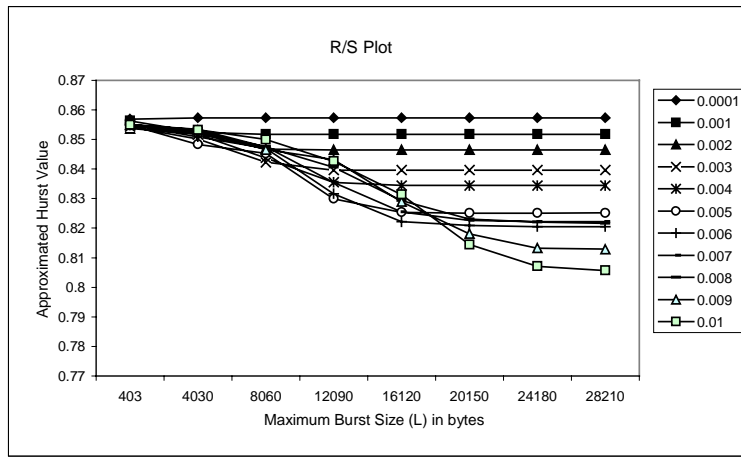


Figure 6. Hurst value approximated using R/S plot by varying L .

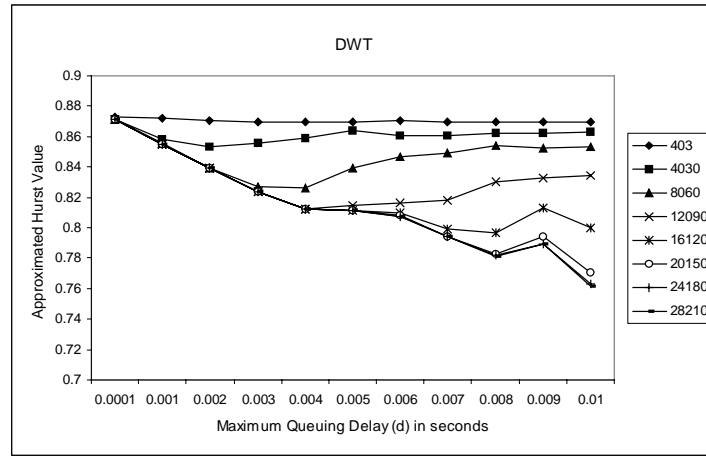


Figure 7. Hurst value approximated using DWT by varying d .

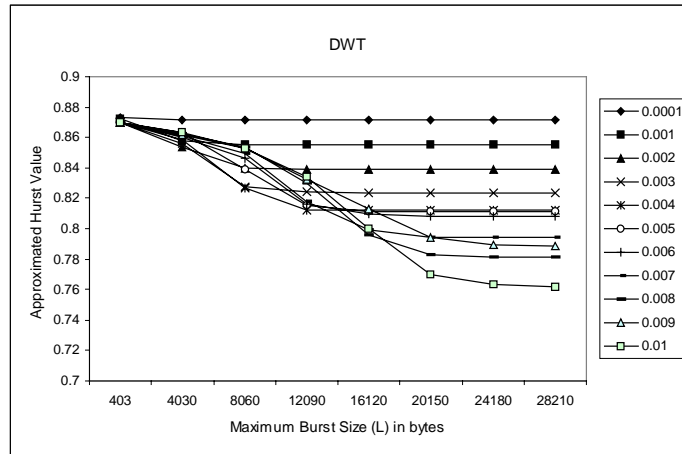


Figure 8. Hurst value approximated using DWT by varying L .

5. CONCLUSIONS

It is well known that self-similarity of network traffic presents with a great challenge in modern network design. This paper studied the effectiveness of buffering at edge routers in reducing traffic self-similarity of aggregated packet/burst switched optical networks. The packet/burst assembler employs two parameters, the maximum burst size L and the maximum queuing delay d , to smooth the incoming self-similar traffic while preserving an upper bound on the queuing delay. The self-similarity of the output burst flow has been analyzed using both R/S plot and discrete wavelet analysis methods. Our results from both analysis methods confirm that by manipulating the L and d parameters, the self-similarity index of the output traffic can be reduced when compared to the self-similarity of the input traffic.

Our future work involves the implementation of the on-line version of the wavelet analysis technique.¹¹ By this the self-similarity of optical burst arrivals can be measured on even longer time scales. This enables a more accurate measurement of the long-range dependency of bursts. We expect that our future research will provide even more conclusive results on the effectiveness of buffering.

REFERENCES

1. W. E. Leland, "On the self-similar nature of ethernet traffic (extended version)," *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, 1994.
2. A. Ge, F. Callegati, and L. S. Tamil, "On optical burst switching and self-similar traffic," *IEEE Communications Letters*, vol. 4, no. 3, 2000.
3. X. Yu, Y. Chen and C. Qiao, "A study of traffic statistics of assembled burst traffic in optical burst switched networks," *Opticomm 2002*, Boston, 2002.
4. K. Claffy, G. Miller and K. Thompson, "The nature of the beast: Recent traffic measurements from an Internet backbone," in *Proceedings of the Internet Global Summit Conference*, San Diego, CA, 2003.
5. P. Abry and D. Veitch, "Wavelet analysis of long-range-dependent traffic," *IEEE Transactions on Information Theory*, vol. 44, no. 1, 1998.
6. K. Park and W. Willinger, "Self-similar network traffic: An overview," in *Self-Similar Network Traffic and Performance Evaluation*, K. Park and W. Willinger, editors, Wiley Interscience, New York, 1999.
7. H.E. Hurst, "Methods of using long-term storage in reservoirs," in *Proceedings of the Institution of Civil Engineers*, vol. 116, pp. 770-799, 1951.
8. S. Chatterjee, A. Hadi, and B. Price, *Regression Analysis by Example*, 3rd edition, Wiley, New York, 2000.
9. A. Cohen, I. Daubechies, J. C. Feauveau, "Biorthogonal bases of compactly supported wavelets," *Communications on Pure and Applied Mathematics*, vol. 45, no. 5, pp. 485-560, 1992.
10. W. Sweldens, "The lifting scheme: A new philosophy in biorthogonal wavelet constructions," in *Wavelet Applications in Signal Processing III, Proceeding of SPIE 2569*, pp. 68-79, 1995.
11. M. Roughan, D. Veitch, P. Abry, "Real-time estimation of the parameters of long-range dependence," *IEEE/ACM Transactions on Networking*, vol. 8, no. 4, pp. 467-478, 2000.