Thinning, Striping and Shuffling: Traffic Shaping and Transport Techniques for Variable Bit Rate Video

Dennis Bushmitch,
Panasonic Information Networking Technologies Lab (PINTL)
Princeton, New Jersey, USA

Shivendra Panwar, Anandabrata Pal
Polytechnic University
Brooklyn, New York, USA

Abstract[1]
In this paper we investigate the transport layer processing intended to improve the communication of multimedia data over wireless and wireline networks. Specifically we consider data *striping* and *thinning* techniques, which are applicable to the multipath / multiflow transmission of multimedia (e.g., MPEG4). We also introduce the *shuffling* procedure, which reorders the data at the network edges. We show that all three techniques break up short-term correlations ind ata traffic thus improving its queueing performance. We demonstrate that while both Long Range Dependence (LRD) and Short Range Dependence (SRD) influence the queueing performance for some timescale of the queueing system, it is the short-range statistical properties of multimedia traffic within the Critical Time Scale that are *dominant* in determining the buffer efficiency of the queue. We also show that estimation of LRD for thinned and striped data may lead to a misleading notion of LRD reduction, when none is warranted. We further outline the ideas for a new transport layer protocol that explores the combination of thinning, striping and shuffling approaches to multimedia data transmission.

I. Introduction

Shortly after the discovery of Long-Range Dependence (LRD) in Ethernet-based data and Web traffic [1]-[4], there began an ongoing debate on the significance of LRD's presence in input traffic for queueing performance. Some of the first papers to argue the effects of the LRD in input data on queueing performance were [5][6]. It has been shown since that LRD has negative effects on the performance of the telecommunication queues, causing excessive datal oss and buffer inefficiency [7]-[10]. LRD sources have been shown, among other things, capable of inducing traffic starvation of the lower priority traffic. It has been also described that traffic aggregation generally doesn't reduce the degree of LRD [1], [15].

Thus some have proposed several wayso f decreasing LRD by traffic shaping [13]-[15]. In [13] the burst aggregation method is proposed at the edges of the network, and is shown to decrease the self-similarity in data while satisfying the delay guarantees in traffic delivery. In [14] the authors propose a technique of dual leaky bucket shaping of web traffic, decreasing the intensity of the long-duration traffic bursts. Since those are shown to be the main contributors toward self-similarity in traffic, the proposed processing decreases the Hurst parameter H and, as a result, improves the queueing performance. In [15] the proposed traffic smoothing algorithm lowers both LRD and short term correlation of queueing data.

At the same time the applicability of LRD-based modeling for real time traffic sources has been questioned [16]-[21]. In [20] and [21] the authors argue that only correlations within the relatively small Critical Time Scale (CTS) contribute to the queue's performance, making long term correlations irrelevant. These findings suggest that Short-Range Dependent (SRD)-based modeling that captures the correlation structure up to CTS can be accurate. We note that certain levels of aggregation were present in those experiments, while buffer sizes were kept small (~10 ms) (e.g., ATM like network dimensioning). Results were less conclusive when the level of aggregation was kept low, and queueing performance at longer time scales (~100ms) was analyzed. In [11], Willinger argues that SRD model can always approximately fit the LRD process over a specific time scale, explaining results in [16]-[18]. However, such models break down when a wider range of time scales is considered, and are usually highly parameterized.

The findings of self-similarity in temporal media (e.g., compressed video) [12] quickly followed its discovery in data traffic. Several rather sophisticated models [22],[23] that capture both LRD and SRD properties of compressed video have emerged since. These models are scene-based, and typically incorporate several contributing random processes. In [19] authors classify these models as having an intermediate position between SRD and LRD-based models in terms of the applicable degree of the buffer dimensioning and the model's accuracy for different timescales. Unfortunately, an accurate and simple single-source model for video traffic, which reflects both its LRD and SRD properties, is still absent. Such modeling is particularly important for: (1) access link QoS engineering, (2) protocol rate control design and (3) single source traffic description for the purpose of QoS support (e.g., RSVP).

In the present work we concentrate our attention on the most recent type of temporal traffic: MPEG4-based high quality video (~1 Mbps). MPEG4's self-similarity was recently described in [24]. In our work we analyze the same family of diverse MPEG4 traces. We concentrate on a single MPEG4 traffic source without aggregation. We look at queueing

[1] This work is supported in part by the New York State Center for Advanced Technology in Telecommunications (CATT) at Polytechnic University, and in part by the NSF under grants ANI0081527 and ANI0081357.

0-7803-7632-3/02/$17.00 ©2002 IEEE

performance on the time scales characteristic of the multimedia client-server video streaming environment (~100-300 ms buffer sizes). As we evaluate the queueing performance, we consider the buffer overflow probability (BOP) (i.e., the probability of exceeding a certain threshold in an infinite buffer queue) versus the buffer size. We find that SRD properties of such traffic are still dominant in influencing its queueing performance. We further suggest and evaluate several techniques that improve the characteristics of this traffic in short range region: (1) thinning, (2) striping and (3) shuffling.

The rest of this paper is organized as follows. In section 2 we introduce data thinning and show its effect on LRD and SRD properties of traffic and video traffic in particular. In section 3 we discuss the data striping technique, which can be viewed as a combination of thinning and interleaving. In section 4 we describe the shuffling approach and its effects on LRD and SRD in multimedia traffic. In section 5 we propose the transport-layer protocol that would support and utilize all of the above traffic-handling techniques. We then conclude our paper with some directions for future research.

II. Thinning and Block-based Thinning of multimedia Data

Consider the process of creating a new media stream by means of picking the elements of the original traffic process X[n], n=0,1,2,..., in their increasing order and periodically skipping every S elements in between. In other words we create a new "thinned out" sequence $X^{\sim}[n]$ from the original data samples separated by the distance S+1 with the rest of the elements dropped. Multiple "substreams" can be created from the original stream X[n] by offsetting the starting point of the thinning procedure, creating (S+1) distinct thinned sequences. In other words:

$$X^{\sim}[k] \triangleq X[(S+1)k] \quad k=0,1,2,\ldots \qquad (2.1)$$

Let ρ[k] be the autocorrelation function (ACF) of the original wide sense stationary sequence X[n] and $\rho^{\sim}[k]$ be the autocorrelation of the thinned $X^{\sim}[n]$ sequence. If S is the thinning *skip* value, the following holds for the *thinned* sequence X[n]:

$$\rho^{\sim}[k]=\rho[(S+1)k] \qquad (2.2)$$

Let us also introduce the *block-based thinning* as thinning on the block basis. Divide X[n] into equal-sized blocks of length B. Assemble a block-based thinned sequence $X^{\sim}[n]$ from the blocks of X[n] in their increasing order and skip every S blocks in between. This makes the spacing distance between the original blocks of data to be $S+1$.

$$X^{\sim}[k]\triangleq X\left[\left\lfloor \frac{k}{B}\right\rfloor(S+1)B+k \bmod B\right] \quad k=0,1,2,\ldots \qquad (2.3)$$

Let again ρ[k] be the ACF of the original WSS sequence X[n]. Under careful examination, the *block thinned* $X^{\sim}[n]$ sequence is 2nd order cyclostationary with the period of cyclostationarity equal to B, meaning that its ACF $\rho^{\sim}[m, k]$ is:

$$\rho^{\sim}[m,k]=\rho^{\sim}[m+lB,k] \text{ for } l\in\mathbb{Z}$$

For lags k: k mod B = 0, the following holds for the ACF of *block thinned* $X^{\sim}[n]$:

$$\rho^{\sim}[m,k]=\rho[(S+1)k] \text{ for } \forall k: k \bmod B=0 \text{ and } \forall m \qquad (2.4)$$

From (2.2) and (2.4) one can see that $\rho^{\sim}[k]$ is essentially a "squeezed" version of ρ[k. In Figure 1 we measure the ACF of the original and the thinned FGN synthetic data to illustrate this observation.

If X[n] is LRD with Hurst parameter H, its ACF decays hyperbolically as:

$$\rho(k)\sim C_{\rho}k^{-\beta}, \text{ as } k\to\infty, \text{ where } \beta=2-2H. \qquad (2.5)$$

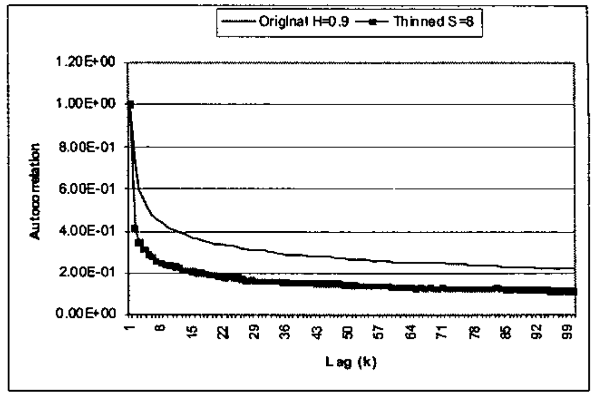

Figure 1 **ACFs of the Original and Thinned FGN Traces**

Then ACF of thinned $X^{\sim}[n]$ is also LRD with the *same* Hurst parameter H since from (2.2):

$$\rho^{\sim}(k)\sim C_{\rho}\left[(S+1)k\right]^{-\beta}\sim C_{\rho}^{\sim}k^{-\beta}, \text{ where}$$

$$C_{\rho}^{\sim}=C_{\rho}(S+1)^{2H-2}$$

This can also be seen in Figure 1 as the curve $\rho^{\sim}[k]$ vs. k has the same asymptotic rate of decay as ρ[k], indicating the invariability of the Hurst parameter upon thinning of a sequence.

We can expect that if the observed ACF reduction occurs within the range of queueing system's critical time scale (CTS), one should observe an improvement in queueing performance. For an approximation of BOP, given the ACF model and system's scale, consult [20],[21].

Because LRD is an asymptotic notion, one has to be careful in not claiming the reduction in LRD, as the actual measurements of H for the thinned sequences are observed. We generated a very long (1 million samples) synthetic Fractional Gaussian Noise (FGN) sequence and thinned it with various skip values, while estimating Hurst parameter and its confidence intervals using Whittle's estimator (see [26]). Figure 2 shows the decrease in Hurst of the thinned sequence as the value of skip S increases. We attribute such reductions to the decreasing sample count of the thinned sequences.

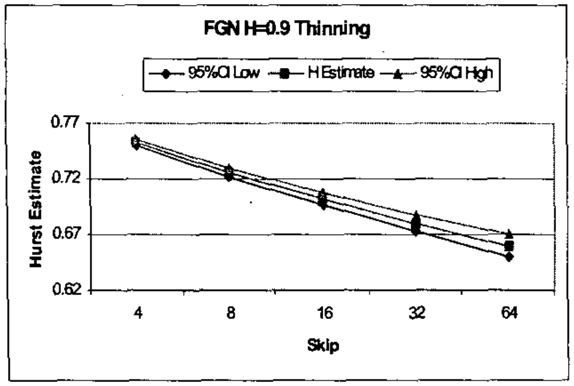

Figure 2 **Thinning FGN Synthetic Traces**

We further studied aspects of thinning on queueing performance by thinning a "Silence of the Lambs" MPEG4 trace with different values of S as shown in Figure 3. The substreams' performance should be compared in pairs, e.g., S=8 vs. S=32 and S=16 vs. S=64. This is because the periodic structure of the Group of Pictures (GOP) in MPEG4 (GOP=12 in our case) caused thinning to "thin out" a disproportionate number of B or P frames vs. I frames. Thinning with S=16 and S=64 actually increased the average bitrates of those traces by equal amounts for the reason above, making the S=8 curve have a"b etter" performance than S=16. This would not be the case if S+1 was chosen to be a number which is relatively prime to the GOP value (e.g., S+1 = 7,11, etc.). *Lesser* improvements in queueing performance hold for a block-based thinning of the same MPEG4 trace, as shown by Figure 4. This is caused by the short-term correlations (up to lag B), which are mostly preserved by block-based thinning. We note that there is no disproportionate thinning in this case, as expected. We also take care of having left a sufficient number of samples for simulation purposes after thinning takes place, by extending the length of the original trace to 1 million samples by means of looping.

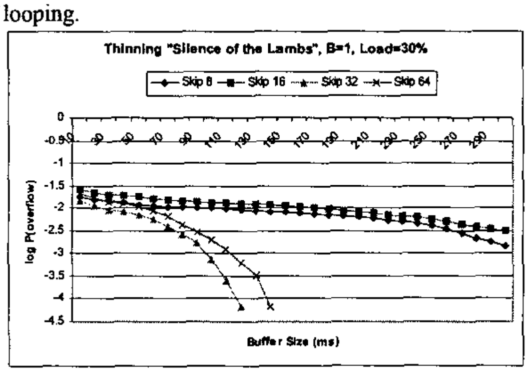

Figure 3 **Thinning MPEG4 "Silence of the Lambs" Trace**

III. Striping

The striping process for L streams $X_i[n]$ ($i \in [1, L]$) into N striped substreams is defined as follows: (1) Each data stream $X_i[n]$ is partitioned into blocks of equal size (B), (2) for each block the destination *striped substream* $X^{\sim}[n]_k$ ($k = 1,\ldots,N$)is determined by using the round robin partitioning $k = j \bmod N$, where jis the block's index within each stream and N is the total number of striped substreams,

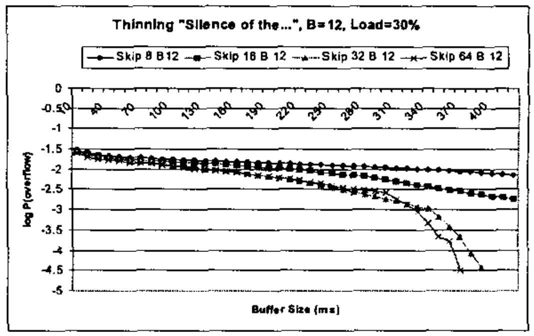

Figure 4 **Block-based Thinning of "Silence" MPEG4 trace**

(3) blocks are added from multiple streams X_i to the substream $X^{\sim}[n]_k$ in their increasing order in simultaneous interleaving fashion. Streams $X_i[n]$ can also be created from single long sequence X[n] by means of breaking it into L distinct separate regions. As an example, consider the process of storing a contiguous media stream across multiple (N) storage elements (SEs), as it is done in RAID0 technology [27], thus forming N substreams. Do this for multiple streams $X_i[n]$ ($i \in [1, L]$). Now consider the output of a single SE when multiple streams are accessed in a uniform and homogeneous manner. It again constitutes a *striped substream* $X^{\sim}[n]_k$. Striping can be viewed as the block-based thinning combined with interleaving with other block-thinned streams. Striping is identical to round robin data striping widely used in distributed storage architectures for scalability purposes [28].

Because another streams' data has no correlation with a given stream's data, we expect the ACF to fall sharply after the initial set of lags [1,..,B]. Given that the original stream is LRD, ACF will "climb up" somewhat after B*N lags, and will do so to a lesser degree each subsequent B*N lags. If the CTS of the system exceeds B and is less than B*N, we expect a significant improvement to BOP as a result of striping stream X[n] over N substreams. We show the ACF of the striped FGN data superimposed with equally parameterized ACF of the block-based thinned trace in Figure 5. Striping should also result in better queueing performance than thinning, at the expense of reconstructing data from multiple striped substreams.

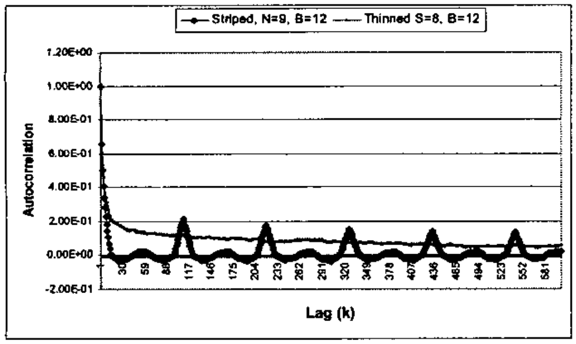

Figure 5 ACF of Striped FGN data

To simulate the process of striping, we fix the number of substreams to N=4, which keeps constant the block *skip* distance S. We use 4 randomly shifted (on a large scale) copies of the same trace to simulate 4 different streams $X_i[n]$ of similar characteristics (i.e., L=N). We vary the block size B in this simulation, controlling in this way the number of short-term correlations of the original streams that affect queueing performance. Because in data striping B is usually measured in bytes, not in frames, we translate its size (Kbytes) into a frame count by using the average frame size. Simulated striping is thus performed on a sample (frame) level. The load is uniform, thus each stream is represented equally in a substream. The results of striping "Jurassic" at 50% server utilization are shown by Figure 6. These results show a dramatic decrease in performance when size 16 and above (16=64Kbyte÷3.8Kbyte/frame (i.e., average sample size)) correlations are preserved by a striping transformation due to larger block size selection.

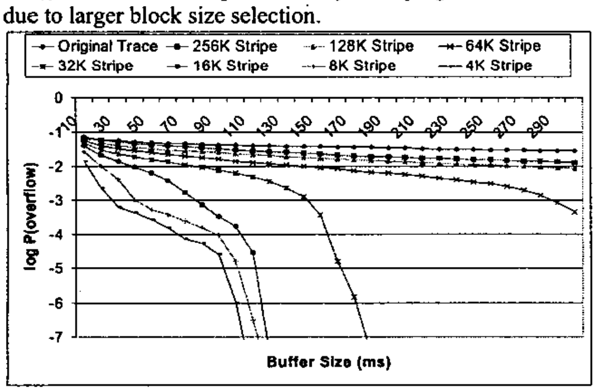

Figure 6 **Striping of Jurassic Park, ρ=0.5**

The estimated Hurst Parameter values using Whittle's estimator for substream $\breve{X}[n]$ are as following:

2K	4K	8K	16K	32K	64K	128K	256K	Orig.
0.60	0.60	0.56	0.53	0.52	0.53	0.51	0.42	0.65

Notice that the smaller Hurst values (and thus assumed lesser degree of LRD) resulting from the separation of individual samples by a larger block boundaries (akin to increasing S) do not necessarily improve the queueing performance. Indeed quite the *opposite* is observed. This is another indication of the dominant contribution of SR statistics of the data on the queueing performance. This also indicates that the Hurst parameter H alone is insufficient in characterization of multimedia streams The striping results are similar for the other two traces.

IV. Shuffling

The shuffled data stream $\breve{X}[n]$ is constructed by means of a *shuffling* permutation rule, which is defined by two parameters *S* (skip) and *R* (restart) and proceeds as following for each block size (S+1)(R+1) of the original sequence X[n]: (1) Samples of $\breve{X}[n]$ are obtained by taking X[n] samples spaced by a *Skip* distance S in their increasing order: {$\breve{X}[0]$, $\breve{X}[1]$, $\breve{X}[2]$, $\breve{X}[3]$,...} = { X[0], X[0+(S+1)], X[0+2(S+1)], X[0+3(S+1)], ...}. (2) After the number of so-called "skips" reaches the *Restart* value R, the process restarts at the first yet unused value of X[n], i.e., X[1], as: {$\breve{X}[1+R]$, $\breve{X}[2+R]$, $\breve{X}[3+R]$, $\breve{X}[4+R]$,...} = { X[1], X[1+(S+1)], X[1+2(S+1)], X[1+3(S+1)], ...}. Following this second round of "skips" and after reaching the restart value of R again, the process restarts by picking the next lowest unused member of X[n], i.e., X[2]. The process continues in this fashion until all samples X[n] are used in a block size (S+1)(R+1). Then the process proceeds by permuting samples within the next block of X[n]. If $\breve{X}[k]$ = X[n] defines the shuffling permutation, then:

$$k=\left\lfloor\frac{n}{S+1}\right\rfloor \bmod (R+1)+(R+1)\{n \bmod (S+1)\}+\left\lfloor\frac{n}{(S+1)(R+1)}\right\rfloor(S+1)(R+1) \quad (4.1)$$

Block-based shuffling is defined as shuffling of equal size blocks (B) rather than individual samples. The permutation between k and n is more complex: $k = j \cdot B + n \bmod B$, where

$$j=\left\lfloor\frac{i}{S+1}\right\rfloor \bmod (R+1)+(R+1)\{i \bmod (S+1)\}+\left\lfloor\frac{i}{(S+1)(R+1)}\right\rfloor(S+1)(R+1)$$

and $i=\left\lfloor\frac{n}{B}\right\rfloor$ (4.2).

When R→∞, we obtain the case of thinning.

Let again ρ[k] be the ACF of the original WSS sequence X[n] and $\breve{\rho}[m, k]$ be the autocorrelation of the shuffled sequence $\breve{X}[n]$. Again we find that $\breve{X}[n]$ sequence is 2nd order cyclostationary with the period of cyclostationarity equal to (S+1)(R+1), meaning:

$$\breve{\rho}[m,k]=\breve{\rho}[m+l(S+1)(R+1),k] \text{ for } l\in\mathbb{Z}$$

Again, for lags k : $k \bmod (S+1)(R+1) = 0$

$\breve{\rho}[m,k]=\rho[k]$ for k: $k \bmod (S+1)(R+1)=0$ and ∀m (4.3)

For the *block-based shuffling*,

$\breve{\rho}[m,k]=\rho[k]$ for k : $k \bmod B(S+1)(R+1) = 0$ and ∀m

We show the plot of ACF for the original and shuffled (S=16, R=16) traces in Figure 7, assuming WSS stationarity of shuffled sequence. The period of overlap between the two functions following (4.3) is 17*17=289. This periodic overlap in autocorrelation values between the original and the shuffled sequences makes the asymptotic rate of decay of the two ACFs to be the same. This makesth e Hurst parameter (H) of the original sequence *equal* to the Hurst parameter $\breve{H}$ of the shuffled data. We observe this *immutability* of the

Hurst parameter by measuring the Hurst parameter of shuffled synthetic FGN data usingW hittle's estimator, as shown by Figure 8.

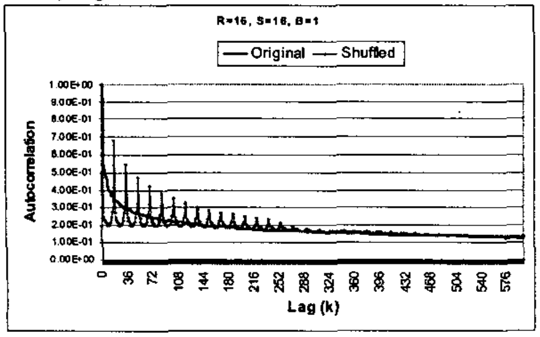

Figure 7 Autocorrelation of the Original and Shuffled FGN Data (S=16, R=16, B=1)

We also note that variance coefficient $a \triangleq \sigma^2 / m$, where σ^2 is the variance and m is the mean of X[n], of the shuffled data is the same as the variance coefficient of the original data, since neither the variance nor the average rate of the increment process m changes upon shuffling. This implies no change in the performance of the LRD traffic if the common FBM modeling [29] is adopted as an approximation.

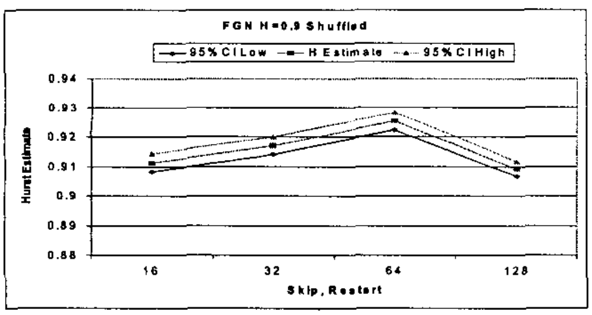

Figure 8 Hurst of Shuffling Synthetic FGN Traces

However, shuffling combats the buffer inefficiency well, due to the reduction of the short-term correlations in data within the CTS of the system, as shown by Figure 9 assuming WSS stationarity of shuffled sequence. The results for the queueing performance of the shuffled Jurassic trace are shown by Figure 10 for B=1 and by Figure 11 for B=12. Notice that single frame shuffling (B=1) performs significantly better than the block-based shuffling for sufficiently large shuffles (e.g., S=64). Just like in the striping case, this is explained by the short-term correlations (up to a lag of B) preserved by the block-based shuffling. However, this reduction is insignificant for small skip values, since they don't affect short-term autocorrelation structure much. The results for the other two MPEG4 traces are similar and were not shown here due to space limitations.

Because compressed video is not stationary (i.e., has a strong periodic component with the period of 1 GOP) and because shuffling introduces cyclostationarity, one has to be careful about how shuffling is conducted and analyzed. If the CTS of the system is not much greater than the size (S+1)(R+1) of the permutation block, the cyclostationarity of thesh uffled sequence cannot be neglected, and new analytical tools will need to be developed to treat it. Certain shuffling patterns may also, for example, cluster I frames together, introducing long periods of buffer overflow in the queueing system. This makes the alignment of GOP and shuffling permutation block an important issue.

If the data transmitted via the network is shuffled,two buffers are required at the sender and the receiver, as shown by Figure 12. The sender's shuffling buffer (size M =(S+1)(R+1)B-2B) serves as a temporary storage for the traffic till it is transmitted in a shuffled order, the receiver's re-shuffling buffer of the same size reconstructs the original order of traffic. These two buffers introduce additional transmission delays and memory expenditures.

However, when the number of intermediate queues (N) is sufficiently large, the overall saving in buffer space, bandwidth and the delay across all nodes combined would compensate for these expenditures. Thus the true benefits of shuffling may not always be realized in single hop transmissions.

V. Protocol

We propose combining the described above data manipulation processes "under the umbrella" of one new transport layer protocol. Data thinning, block-based thinning and striping can be utilized to create substreams, which could be used for multipath / multiconnection transport [30][31]. In addition, we demonstrated that these processes also improve the characteristics of individual substreams in terms of short-term burstiness.

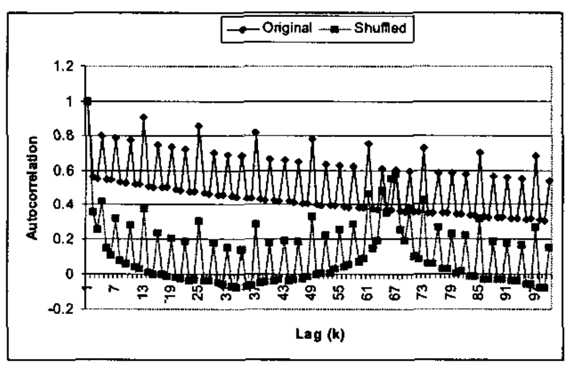

Figure 9 Autocorrelation of the Original and Shuffled (S=64, R=64, B=1) "Jurassic" Trace

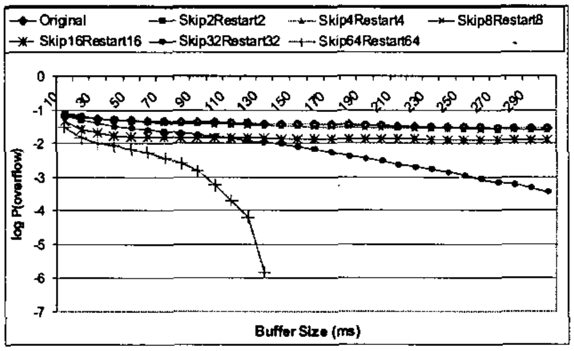

Figure 10 Shuffling "Jurassic Park" Trace, ρ=0.5, B=1

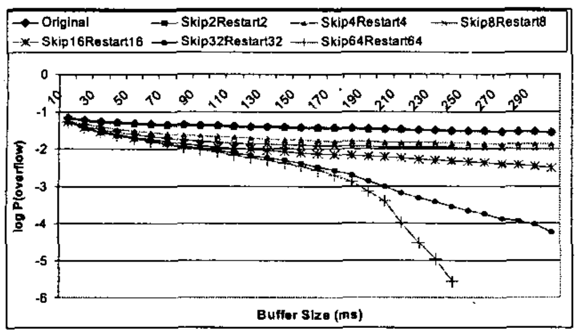

Figure 11 Shuffling "Jurassic Park" Trace, ρ=0.5, B=12

Such a protocol would operate at the layer above UDP and provide connectionless real time transport services to an application. The proposed protocol would have a dynamic mode selection (e.g., striping alone vs. shuffling combined with thinning, etc.) Shuffling can be deployed on both levels: the individual substream and the substream aggregate. Thinning can precede shuffling, or alternatively may follow it. All protocol parameters (S, R, B, N) would be fully configurable by the application framework at any time. Their choice can be dynamic as well as dependent on the networking conditions of the underlying transmission paths / flows. The proposed protocol should be fully compatible with today's multimedia streaming environments, where protocols like SIP could provide session management for it.

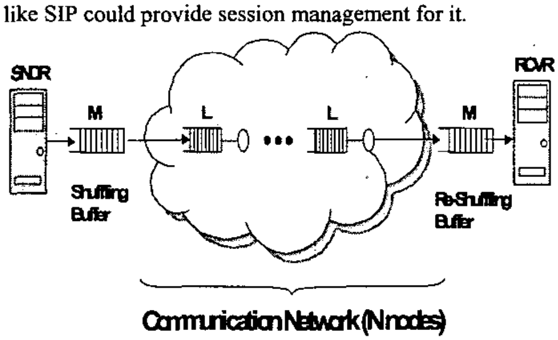

Figure 12 Multi-hop Network Transmission of the Shuffled Data

VI. Conclusion

In this paper we presented three different traffic shaping techniques and considered their effects on handling MPEG4 data. Thinning, striping and shuffling all have been shown to improve the queueing characteristics of data, by decreasing short-term burstiness and diminishing short-term correlations. We have also demonstrated that, for a single-source traffic with practical scale buffer provisioning, the short term statistics are the dominant factor in defining the queueing system's performance. None of these processes are shown to decrease the degree of LRD in data. We also show that LRD estimates can be misleading and no claims of LRD reduction by this kind of traffic shaping can be made.

Nevertheless, observed performance improvement makes these processes good candidates for a transport layer protocol engine. In the near future we are planning to undertake more sophisticated striping simulations. We expect that the increase in the number of substreams would improve queueing performance further. More shuffling experiments are also warranted. In particular, it is necessary to investigate the effects of *skip* and *restart* values on queueing performance. More generally, we can study other shuffling permutations. We are also planning to prototype the proposed protocol engine and evaluate its performance in handling real time media streaming in diverse networking environments.

During this work, we have also created a graphical simulator (written in Java for multiple platforms), which combines all the simulations and statistical tools utilized in this research. We hope that some of these tools are general enough to be found useful by others. The package it available online to any interested reader [25].

Acknowledgement

The authors thank Dr. Rauf Izmailov (NEC)for his valuable contributions to this work.

References

[1] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of Ethernet traffic," *IEEE/ACM Trans. Networking*, vol. 2, pp. 1—15, Feb. 1994.

[2] V. Paxson, S. Floyd, "Wide Area Traffic: The Failure of Poisson modeling", *IEEE/ACM Transaction on Networking*, vol. 3, no. 3, June 1995.

[3] Z. Liu, N. Niclausse and C. Jalpa-Villanueva, ``Web Server Benchmarking and Web Traffic Modeling", In *System Performance Evaluation: Methodologies and Applications*, E. Gelenbe (Ed.), CRC Press, 1999.

[4] M. Crovella, A. Bestavros, "Self-Similarity in World Wide Web traffic: Evedence and Possible Causes", *IEEE/ACM Transaction on Networking*, vol.5, no.6, pp. 835-846,1997

[5] H. Fowler, W. Leland, "Local area network traffic characteristics, with implications for broadband congestion management," *IEEE JSAC*, vol. 9, pp. 1139-1149, Sept. 1991.
[6] A. Erramilli, O. Narayan, W. Willinger, "Experimental Queueing Analysis with Long-Range Dependent Packet Traffic", *IEEE/ACM Transactions on Networking*, vol. 4, no. 2, April 1996.
[7] Z. Sahinoglu, S. Tekinay, "On Multimedia Networks: Self-Similar Traffic and Network Performance", *IEEE Communications Magazine*, January 1999.
[8] K. Park, G. Kim, M. Crovella "On the relationship Between File Sizes, transport protocols, and self-similar network traffic", *Proceedings of 4th ICNP*, 1996.
[9] K. Park, G. Kim, M. Crovella, "On the Effect of Traffic Self-Similarity on Network Performance", *SPIE International Conference on Performance and Control of Network Systems*, November 1997.
[10] P.R. Morin "The impact of Self-Similarity on Network Performance", *Ph. D. dissertation*, Carleton University, December 1995.
[11] W. Willinger, "Scaling Phenomena and Data Network Traffic: Lecture 2", *AT&T Labs Research*, June 2000.
[12] M. W. Garrett and W. Willinger, "Analysis, modeling and generation of self-similar VBR video traffic," in *Proc. SIGCOMM*, London, U.K., 1994.
[13] A. Ge, F. Gallegati, L. Tamil, "On Optical Burst Switching and Self-Similar Traffic", *IEEE Communications Letters*, vol.4, no.3, March 2000.
[14] K. Christensen, V. Ballingam, "Reduction of Self-Similarity by Application-level Traffic Shaping", in *Proc. IEEE 22nd Annual Conference on Local Computer Networks*, pp. 511 – 518, November 1997.
[15] F. Xue, S. Ben Yoo, "Self-Similar Traffic Shaping at the Edge Router in Optical Packet-switched Networks", *Proc. IEEE ICC*, New York, 2002.
[16] D. Heyman, "The GBAR source model for VBR video conferences," *IEEE/ACM Trans. Networking*, vol. 5, pp. 554-560, 1997
[17] D. P. Heyman and T. V. Lakshman, "What are the Implications of Long Range Dependence for VBR-Video Traffic Engineering?," *IEEE/ACM Trans. Networking*, vol. 4, pp. 301—317, June 1996.
[18] D. P. Heyman and T. V. Lakshman, "Source Models for VBR Broadcast-Video Traffic", *IEEE/ACM Trans. Networking*, vol. 4, No. 1,p p. 40-48, 1996.
[19] E. Casilari, A. Reyes, A. Diaz-Estrella, F. Sandoval, "Characterization and modeling of VBR Video Traffic", *Electronics Letters*, vol. 34, No. 10, 1998.
[20] B. Ryu, E. Elwalid, "The Importance of Long-Range Dependence of VBR Video Traffic in ATM Traffic Engineering: Myths and Realities", in *Proc. ACM SIGCOMM'96*, Stanford, August 1996.
[21] D. Eun, H. Ahn, B. Roh, J. Kim, "Effects of LRD of VBR Video Traffic on Queueing Performances", *IEEE GLOBECOM*, 1997.
[22] D. Liu, E. Sara, W. Sun, "Nested Auto-regressive Processes for MPEG-Encoded Video Traffic Modeling", *IEEE Trans. In Circuits and Systems for Video Technology*, Vol. 11, No. 2, February 2001.
[23] M. Krunz, S. Tripathi, "On the Characterization of VBR MPEG Streams", *Proc. of the 1997 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, Vol. 25, Iss. 1, June 1997.
[24] F. Fitzek, M. Reisslein, "MPEG4 and H. 263 Video Traces for Network Performance Evaluation", *Technical University of Berlin*, Technical Report TKN-00-06, Berlin, October 2000.
[25] D. Bushmitch, A. Pal, "GUI Framework for Video Traffic Analysis and Manipulation: VTAM", *www.bushmitch.com/vtam.html*, Princeton, 2001.
[26] J. Beran. Statistics for Long-Memory Processes. Chapman and Hall, 1994.
[27] P. M. Chen, E. K. Lee, G. A. Gibson, R. H. Katz, D. A. Patterson, "RAID: High Performance Reliable Secondary Storage", *ACM Computing Surveys*, vol. 26, no. 2, pp. 145-185, June 1994.
[28] Aref, W.; Bushmitch, D.; Kamel, I.; Mukherjee, S., "An Inexpensive, Scalable, and Open-Architecture Media Server", *Proceedings of First IEEE/RPS Joint Conference on Internet Technologies and Services*, Moscow, Russia, October 1999.
[29] I. Norros, "On the Use of Fractional Brownian Motion in the Theory of Connectionless Networks", *IEEE J. Sel. Areas in Communications*, August 1995.
[30] R. Chaw, C. Lee and J. Liu, "Traffic Dispersion Strategies for Multimedia Streaming", *8th IEEE Workshop on Future Trends in Distributed Computing Systems*, 2001.
[31] N. Gogate, S. Panwar, "Supporting Video/Image Applications in a Mobile Multihop Radio Environment Using Route Diversity", *IEEE International Conference on Communications*, pp. 1701 –1706, vol.3, 1999.