DTMW: A New Congestion Control Scheme for Long-Range Dependent Traffic*

C. Huang[a], I. Lambadaris[a], M. Devetsikiotis[a], P. W. Glynn[b], and A. R. Kaye[a]

[a]Department of Systems & Computer Engineering, Carleton University,
1125 Colonel By Drive, Ottawa, Ontario K1S 5B6, Canada

[b]Department of Engineering – Economic Systems and Operations Research,
Stanford University, Stanford, CA 94305, U.S.A.

Recent measurements based on long empirical traces have revealed that many important types of traffic (e.g., LAN, WAN and VBR video) possess long range dependent (LRD) characteristics. Studies have shown that traffic streams with LRD will suffer higher cell loss rates than traditional models predict. Therefore network performance can be significantly degraded.

To reduce the cell loss rate, schemes based on predicted service have been proposed in the literature. Predicted service uses traffic monitoring to predict future bandwidth requirements and dynamically allocate bandwidth whenever it is necessary. In this paper, we propose a new traffic predictor called the *Double Threshold Moving Window Detector* (DTMW). Our analytical and simulation results show that DTMW can detect and predict bandwidth requirements for LRD traffic in a robust manner. DTMW is not sensitive to the marginal distributions and short term characteristics of individual traffic streams.

1. INTRODUCTION

Traditional traffic models have in common that they are typically Markovian or, more generally, *short-range dependent* (SRD) in nature, that is, their autocorrelation function is summable (typically decays asymptotically exponentially fast). On the other hand, measurements from modern networks appear to give rise to empirical traffic processes that are generally non-Markovian in nature and exhibit *long-range dependence* (LRD). In other words, empirical traffic processes are characterized by slowly decaying autocorrelations (hyperbolic or power decay) which, in turn, resemble *self-similar* or "fractal" traffic [1–4]. The most striking feature of LRD traffic is that burstiness is displayed across several time scales (i.e., from milliseconds to years [1]). This burstiness can drive queueing systems into overflow more frequently than traditional models have predicted [5,6].

While there are numerous congestion control schemes proposed in the literature, they can be generally classified into three types [7]: The first type is called *guaranteed* service. In recent years, several guaranteed service based algorithms have been developed (see,

*This research was supported by grants from the Telecommunications Research Institute of Ontario. To be presented at the 15th International Teletraffic Congress, *ITC 15*, Washington, D.C., June 1997.

for example, [7] and references therein) that require certain burstiness constraints [8] at the access node. A well-known such constraint is the so-called (σ, ρ) constraint which can be implemented by a "leaky bucket". Therefore, a network that uses guaranteed service typically operates in a low utilization region.

The second type of congestion control approach does not provide for the worst-case scenario, instead it guarantees a bound on the probability of lost packets based on statistical characterizations of traffic (see, for example, [9]). Such approaches focus on finding statistical results for the steady state of the network through *a priori* characterization of flows based on a statistical model. For LRD traffic, steady state results may still play an important role in network planning or long term performance prediction. But as a tool to predict the QoS of a single user session, steady state results cannot be used as in traditional models. As our simulations indicated in [4], the cell loss rate of a single user session may be far from the steady state results due to the inherent LRD structure. This poses significant challenges to the probabilistic service based approach [10]. In addition, analytical and simulation results in [6] have shown that, when LRD traffic is present, the call acceptance region is nonlinear, in violation of the philosophy underlying effective bandwidths. Therefore, effective bandwidth solutions for LRD traffic become a loose bound and result in low network utilization again.

To overcome the problems discussed above, applying a dynamic bandwidth allocation scheme seems unavoidable. A third approach, namely that of *predicted service*, is therefore proposed in the literature. This approach takes into account recent measurements of the traffic load in order to estimate future requirements on network resources, and dynamically allocates network resources to minimize the *post facto* delay bound and maximize throughput [7]. Several protocols providing predicted service have been proposed (see, for example, [7,11,12] and references therein). While these approaches differ from each other in how to react to congestion, they all require a well-behaved *traffic predictor*. For example, in [12], an RCBR (Renegotiated CBR) service discipline is introduced. Users of RCBR service are given the option to renegotiate their service rate at any time. There is a clear trade-off between buffer size, requested rate and the frequency of renegotiation. For interactive applications, the renegotiation schedule cannot be calculated in advance. Instead, the authors of [12] proposed a heuristic AR(1) filter to predict future bandwidth requirements. Although this AR(1)-based heuristic shows improvement over static bandwidth allocation, it is still far from optimal as is pointed out in [12]. Typically, AR filters may introduce long response times which are sensitive to individual traces.

In [13] a predictor for Fractal Gaussian Noise (FGN) is devised and shown to have good performance. Unfortunately, it requires *a priori* knowledge of the Hurst parameter and is limited to the FGN process only. As shown in [3,4], real traffic has an arbitrary marginal distribution which may deviate far from the Gaussian distribution. Furthermore, in interactive applications, it is not possible to estimate the Hurst parameter beforehand.

In summary, the detection of significant and long-term changes in the characteristics of a connection are key in providing users with the type of performance they expect while allowing for efficient usage of network resources. In the following sections, we will propose a new traffic prediction algorithm which can help solve some of previous problems. We will initiate the presentation with intuitive arguments and then justify our solution based on analytical and simulation results.

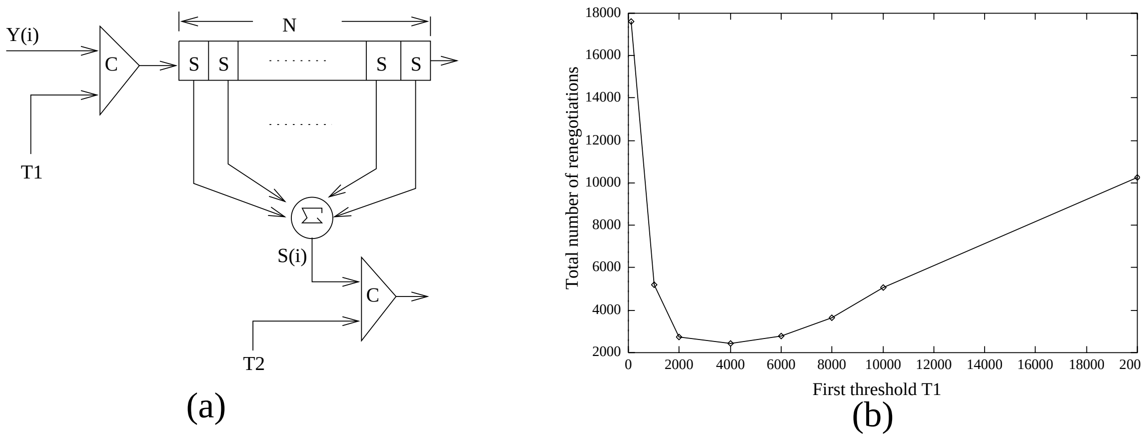


Figure 1. (a) Implementation of DTMW; (b) Frequency of renegotiations versus threshold T_1 for the video trace "Last Action Hero".

2. THE DTMW SCHEME

Broadly speaking, for a process with LRD, high values are more likely to be followed by high values and low values are more likely to be followed by low values. Therefore, in a sense, it is possible to predict future sample values based on past values. Because self-similar traffic displays burstiness at all time scales and because the buffer at the access node can smooth efficiently the burstiness at small time scales, our goal is to predict the burstiness at relatively larger time scales. Therefore, the predictor should not be too sensitive so as to respond to short-term burstiness. Based on these observations, we introduce the *Double Threshold Moving Window* (DTMW) detector in the following.

Let N denote the window length, and T_1 and T_2 denote values of the first and second threshold, respectively. Define the detection quantity $S_N(i)$, $i = N, N+1, \ldots$ as $S_N(i) = \sum_{j=0}^{N-1} I(Y(i-j) \geq T_1)$ where $I(.)$ is an indicator function and $\mathbf{Y} = \{Y(i),\ i = 1, 2, \ldots\}$ is the input process. If $S_N(i) \geq T_2$, we set the detector output to 1, meaning that congestion is about to occur in the near future, otherwise we set the detector output to 0.

It is easy to see that only a comparator, a single-bit shift register and a counter are needed. An illustrative diagram is shown in Fig. 1(a). The first threshold T_1 is set to detect those high values that are larger than T_1. The T_2-out-of-N criterion is for smoothing burstiness at small time scales and detecting the existence of burstiness at large time scales. In the following paragraphs, we justify the above intuition using an analytical approach.

A general analytical solution for the performance of the DTMW under arbitrary input processes can be difficult if not impossible. But, for certain types of sources, it is possible to give analytical results. An example is the CBR model. It is easy to see that if the input process $\mathbf{Y}$ is a CBR process, then DTMW will work as predicted in the last section with a maximum initial response delay of N. In what follows, we analyze the performance of the DTMW under the "*marginal-transformed process*" (MTP) model we proposed in [4]. It has been shown in [4] that this model is general enough to include most of the existing models (e.g., FGN, F-ARIMA) and can match empirical data up to second-order statistics.

The following Proposition applies and further clarifies the DTMW algorithm. Due to space restrictions the proof is omitted (see [14]).

Proposition 1. Let $\mathbf{X} = \{X_i, i = 0, 1, \ldots\}$ be a zero mean, unit variance Gaussian process defined on a probability space $(\Omega, \mathcal{F}, P)$ with autocorrelation function $(r_X(k) : k \geq 0)$. Suppose $r_X(k) \sim k^{2H-2}L(k)$ when $k \to \infty$, where $L(k)$ is a slowly varying function of k and $1/2 < H < 1$. Let $Y = g(X)$, where $g : \mathbf{R} \longmapsto \mathbf{R}$ is a nondecreasing function. If $0 < \Pr(Y \geq T_1) < 1$ and $T_1 > 0$, then the process $\mathbf{Z_N} = \{Z_N(j) = (S_{jN}(jN) - \Pr(Y \geq T_1)jN)/d_N : j = 0, 1, \ldots\}$ with $d_N^2 \sim \frac{2}{(2H-1)2H} N^{2H} L(N)$ converges weakly as $N \to \infty$ to $J(1)B_H(j)$ where $J(1) = \Pr(Y \geq T_1)$ and process $\mathbf{B_H} = \{B_H(t) : t \geq 0\}$ is a Fractional Brownian Motion (FBM) process with parameter H. $\Box$

For the MTP model [4], function $g(.)$ will be the marginal transformation function which will always be nondecreasing. Other conditions in Proposition 1 will be generally satisfied except in degenerate cases. From the above proposition, we have the following conclusions:

(1) For N large enough, the quantity $(S_N(i) - \Pr(Y \geq T_1)N)/d_N$, $i \geq N$, can be approximated by a Gaussian distribution with variance equal to $J^2(1)$. This means that, the detection quantity $S_N(i), i \geq N$ is totally decided by the LRD structure and $\Pr(Y \geq T_1)$. The influences of SRD structure and marginal distribution are removed by the summation procedure in DTMW.

(2) For N and T_2 large enough, the higher the Hurst parameter H, the larger the probability with which DTMW outputs a value of 1.

These two conclusions justify the intuition given at the beginning of this section. Although in practice N is always finite, Proposition 1 can still be a useful guide for tuning T_1, T_2 and N. While we will demonstrate the usage of DTMW through RCBR service and real-time video applications in the following sections, it should be noted that DTMW can be applied to other types of LRD traffic as well as protocols (e.g., ABR) which also focus on predicted services [7,11].

3. INTRODUCTION OF THE DTMW INTO THE ACCESS NODE

The selection of the measure of resource congestion has broad implications for the implementation complexity, stability and performance of the corresponding system. While our conclusions in the last section are based on monitoring the traffic rate process, they are also valid for monitoring the *queueing* increment process based on the following facts:

Consider a slotted-time single server queue with deterministic service rate μ and an arrival rate process $\mathbf{A}$. Let Q_k denote the size of the queue at time $k = 0, 1, \ldots$. Assuming $Q_0 = 0$, we have the following Lindley equation: $Q_i = \langle Q_{i-1} + A_i - \mu \rangle^+ = \langle Q_{i-1} + Y_i \rangle^+$, for $i = 1, 2, \ldots$ where we define the process by $\mathbf{Y} = \{Y_i : Y_i = A_i - \mu, i = 1, \ldots\}$ as the *work load* process. Let us define $\delta\mathbf{Q} = \{\delta Q(i) = Q(i) - Q(i-1), i = 1, 2, \ldots\}$ as the increment process of the queueing process which is denoted by $\mathbf{Q} = \{Q(i) : i = 1, 2, \ldots\}$, we have that $\delta Q(i) \geq T_1$ *if and only if* $A(i) - \mu \geq T_1$. Therefore monitoring $\delta Q(i)$ using DTMW is the same as monitoring $A(i)$ differing only by a constant. Thus, our conclusion in the last section can be applied to the queueing increment process.

When we apply the DTMW to the queueing increment process $\delta\mathbf{Q}$, we are in fact

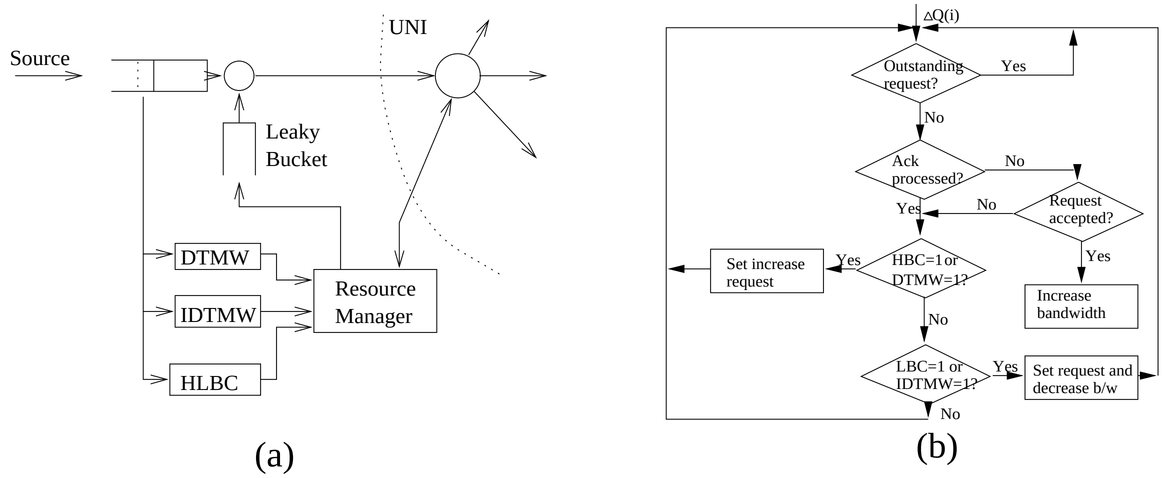


Figure 2. Integration of DTMW with RCBR: (a) structure; (b) flowchart.

monitoring the increasing rate of buffer occupancy. When DTMW outputs a value of 1, it indicates that buffer occupancy is increasing at a rate larger than T_1 per slot. To interpret this in another way, it means that the server needs at least T_1 more bandwidth to keep the buffer occupancy constant. This allows us to integrate DTMW smoothly with RCBR where T_1 can be used as the granularity of the RCBR bandwidth reallocation.

Similarly, we can build a detector which can detect the *decreasing* rate of buffer occupancy as follows:$S_N(i) = \sum_{j=0}^{N-1} I(\delta Q(i-j) \leq T_1)$ where T_1 must be a negative value. If $S_N(i) \geq T_2$, we set the detector output to be 1. We will call this detector the *Inverse Double Threshold Moving Window* (IDTMW) detector.

While DTMW can monitor increasing rates which are larger than T_1, rates smaller than T_1 can still drive the buffer into overflow although in a slower fashion. To further control overflow, we can set a “high buffer” boundary. When buffer occupancy reaches this high buffer boundary, we will always request a bandwidth increase by the quantity T_1. Similarly, to increase the bandwidth utilization, we need to monitor a “low buffer” boundary (which is typically zero), therefore when buffer occupancy reaches the low boundary we will request a bandwidth decrease by the quantity T_1.

The final integrated structure is shown in Fig. 2(a), where HLBC refers to high/low buffer occupancy control. The flow chart for the whole algorithm is shown in Fig. 2 (b), where HBC refers to “high buffer boundary check”. When buffer occupancy is equal or larger than the high buffer boundary, then $HBC = 1$, otherwise $HBC = 0$. Similarly, LBC refers to “low buffer boundary check”. When buffer occupancy is equal or smaller than the low buffer boundary, then $LBC = 1$, otherwise $LBC = 0$.

While bandwidth decrease requests can always be granted without delay, bandwidth increase requests always suffer from a round trip delay. We have taken these factors into account in Fig. 2(b). While the last bandwidth increase request has not been acknowledged, a new bandwidth increase request is not permitted. This prevents excessive bandwidth increase requests being generated due to the round trip renegotiation delay.

4. SIMULATION RESULTS

To test our scheme described in the last section, we simulate our algorithm using four empirical video traces, namely, "Last Action Hero", "Ghost", "Star War", and a segment from BBC News that was digitized in the Broadband Networks Laboratory at Carleton University. All empirical traces are collected at the frame level. In the following, a slot is equal to a frame interval, i.e., approximately 33ms. Buffer occupancy values are always normalized by the mean arrival rates of corresponding sources. We consider separately video segments with intraframe compression (I frames) only and video segments with interframe compression (I, B, P frames). For segments with I, B, and P frames, only "Last Action Hero" and "Ghost" traces are available.

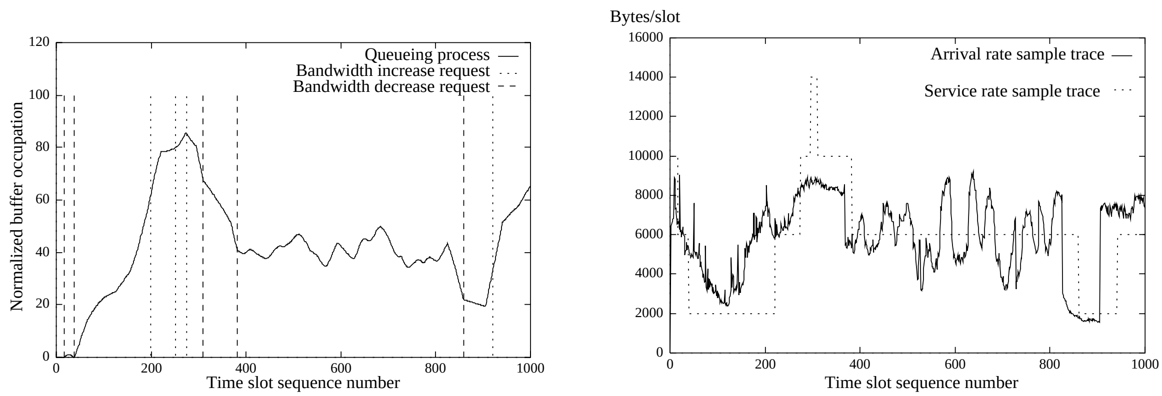
4.1. Video with Intraframe Compression Only

As a scenario, in all cases, we will try to control the normalized buffer occupancy to a value below 100. Assume the round trip delay for bandwidth increase renegotiations to be equal to 20 frame intervals (approximately 0.66s) and that bandwidth reallocation requests are always granted.

First we discuss how to select the control parameters in DTMW. Unlike the AR(1) heuristic, DTMW results in a bounded response delay which is equal to the window size N. To control the normalized buffer occupancy within 100, we need to select the window size N plus the round trip delay to a value smaller than 100 so that the response will not be too late. But an excessively small N will reduce the effect of smoothing out the SRD structure. As a compromise, we set $N = 15$. Because N cannot be set too large, we have to use T_2 to reduce the influence of short term fluctuation. As shown in Proposition 1, the higher the value T_2, the less frequent the renegotiations. So we set $T_2 = N = 15$. For controlling the buffer occupancy below 100 and accommodating a round trip delay at 20 frame periods, the high buffer boundary is set to 80. The parameters of IDTMW are set exactly the same except that the T_1 is negative. The low buffer boundary is set to zero.

The first threshold T_1 of DTMW is more difficult to set. A small T_1 will cause the algorithm to react more frequently therefore introducing more overhead in terms of bandwidth renegotiations. A large T_1 will make DTMW ineffective most of the time and leave the control burden to the high buffer check which may overreact due to the large T_1. In Fig. 1(b), we plot the frequency of renegotiations versus T_1, for the movie "Last Action Hero". Fig. 1(b) shows that there is a low value around 4000. Notice, however, that, around $T_1 = 4000$, the frequency of renegotiations is in general not sensitive to the value of T_1 over wide ranges. We will further illustrate this conclusion later by using different movies. Based on the results in Fig. 1(b), we set $T_1 = 4000$. To simplify the simulations, the bucket size of the Leaky Bucket in Fig. 2 is set to zero.

Fig. 3 to Fig. 4 depict the results for the movie "Last Action Hero". Fig. 3 (a) shows the bandwidth increase/decrease requests versus the queueing process. When the buffer occupancy increases very fast, DTMW requests a bandwidth increase earlier than the time that the buffer occupancy reaches its high boundary. Fig. 3(b) shows the corresponding arrival and departure processes. It can be clearly seen that the service rate tracks the long term arrival rate closely. The histogram of the queueing process is shown in Fig. 4(a). Comparing with Fig. 4(b) which uses CBR service and no control mechanism, the improvement is significant. In Fig. 4(a), most of the normalized buffer occupancy values



(a) (b)

Figure 3. (a) Queueing process and bandwidth increase/decrease requests; (b) Arrival rate process and service rate process. Utilization=1, RCBR service, DTMW/IDTMW control.

Table 1
Results for video traces with intraframe compression, using DTMW.

Video name	Total frames	Mean rate (bytes/frame)	Mean Uti.	Max. buffer	Freq. (1/sec)
LAH	238000	6005.2	0.9997	340	0.31
Ghost	217000	9909.0	0.9978	200	0.26
Star War	170000	27791.0	1.000	140	0.54
BBC News	26000	12709.0	0.9993	103	0.26

are within 100, according to our target and the maximum buffer size without any loss is only 350, which is in stark contrast to 18000 under static CBR in Fig. 4(b). Because the probability of the buffer occupancy larger than 100 now is much smaller than in Fig. 4(b), for a cell-loss tolerant application, the overflow traffic can be dropped.

We applied the same set of control parameters to the other three video traces and list all the results in Table 1. From Table 1, we can see that, for all video clips, the mean utilization is very close to 1 and the maximum buffer occupancy without loss is very close to our target. This shows that DTMW is robust with respect to different traffic streams.

To compare the performance of DTMW with the heuristic AR(1) approach, we apply the AR(1) approach also to the above video traces. As for the simulation of DTMW approach, we chose the parameters for AR(1) approach based on the video trace of "Last Action Hero" and applied the same set of parameters to the other video traces in order to examine the robustness of the AR(1) approach. Similar to the case of DTMW, we set $B_h = 80$ and $B_l = 0$ (see [12] for definitions of B_h, B_l and T) where B_h and B_l are all normalized by the mean source arrival rate. While the maximum bandwidth increase range for DTMW is the same as the granularity of bandwidth allocation (i.e., the first threshold T_1), they can be significantly different for the AR(1) approach. Higher

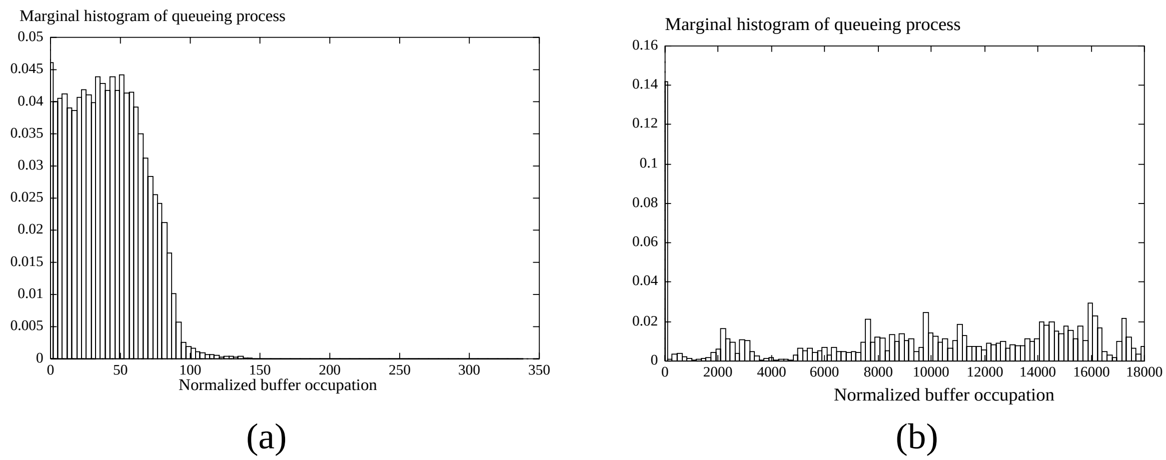


Figure 4. (a) Histogram of queueing process: Utilization=1, RCBR service, DTMW/IDTMW control; (b) Histogram of queueing process: Utilization=1, CBR service, no control.

Table 2
Results for video traces with intraframe compression, using the heuristic AR(1).

Video name	Max bandwidth increase range	Mean Uti.	Max. buffer	Freq. (1/sec)
LAH	4000	0.6019	248	0.27
Ghost	4700	0.6386	190	0.36
Star War	11200	0.7743	176	0.51
BBC News	3300	0.6644	140	0.42

bandwidth increase ranges are more likely to be rejected by the network.

To provide a fair comparison, we set the maximum bandwidth increase range of the AR(1) approach to be the same as for the DTMW by appropriately tuning the value of T. This results in choosing $T = 5000$. A comparison of a source arrival sample trace with the corresponding service rate sample trace is shown in Fig. 5(a) where the slower response of the heuristic AR(1) approach is clearly shown. Final results are also summarized in Table 2.

4.2. Video with both Intraframe and Interframe Compression

Video traces with both intraframe and interframe compression typically exhibit a strong periodic structure associated with their group of picture (GOP) structure. While this may indicate a non-stationary property, we can still treat it as a stationary process with strong short term burstiness in most cases where GOP sizes are small. DTMW is designed to predict fluctuations in long-range dependent streams, therefore it should also work for video streams with both intraframe and interframe compression. To test this conclusion, we apply DTMW to video streams with both intraframe and interframe compression.

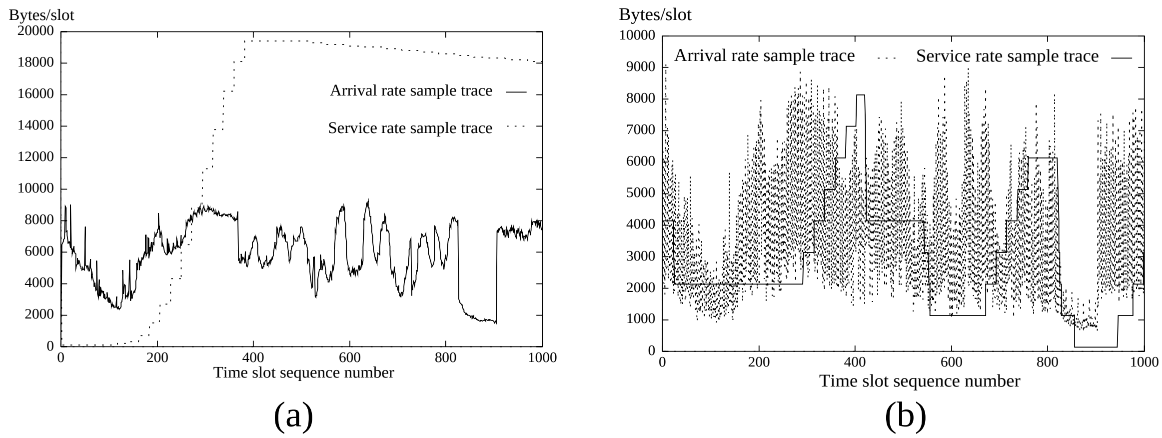


Figure 5. Arrival rate process and service rate process:(a) Utilization=0.6, RCBR service, heuristic AR(1) control, intraframe compression only. (b) Utilization=1, RCBR service, DTMW/IDTMW control, $T_1 = 1000$, intraframe and interframe compression.

Table 3
Results for video traces with intraframe and interframe compression, using DTMW.

Video name	Total frames	Mean rate (bytes/frame)	Mean Uti.	Max. buffer	Freq. (1/sec)
LAH	238000	2481	0.9940	495	0.60
Ghost	217000	4077	0.9944	420	0.50

While keeping all other parameters the same as in the last section, we only optimize again the parameter T_1 through a similar search based on the video trace of “Last Action Hero” as in the last section. The resulting service rate process is shown together with the source arrival rate process in Fig. 5(b). From Fig. 5(b) we can see that the service rate process tracks the arrival rate process closely.

For the two available video traces with both intraframe and interframe compression, the results are listed in Table 3 where the control parameters are the same for both movies. From Table 3, we can see that all utilization values are very close to 1 and the robustness of DTMW is satisfactory.

5. CONCLUSIONS

Predicted service provides dynamic bandwidth allocation for traffic streams with tolerant delay and loss requirements. The crucial part of realizing predicted service schemes is a good predictor that can measure traffic and predict future bandwidth in real time. The DTMW scheme proposed here can predict bandwidth requirements of LRD traffic through on-line measurement. Analytical and simulation studies employing real video traces indicate that DTMW is flexible and robust to different traffic streams in terms of the setting of control parameters. The DTMW algorithm does not require declaration of detailed traffic parameters. Finally, DTMW gives close to 100% bandwidth utilization and buffer occupancy values that are significantly lower compared to the static CBR case.

ACKNOWLEDGEMENTS

The authors are indebted to Dr. Mark Garrett of Bellcore for providing the trace of MPEG compressed "Star Wars".

REFERENCES

1. W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the Self-Similar Nature of Ethernet Traffic (Extended Version). *ACM/IEEE Transactions on Networking*, 2(1):1–15, Feb. 1994.
2. J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger. Long-Range Dependence in Variable-Bit-Rate Video Traffic. *IEEE Trans. on Comm.*, 43(2/3/4):1566–1579, Feb./Mar./Apr. 1995.
3. M. W. Garrett and W. Willinger. Analysis, Modeling and Generation of Self-Similar VBR Video Traffic. In *Proc. ACM SIGCOMM '94*, London, U. K., Aug. 1994.
4. C. Huang, M. Devetsikiotis, I. Lambadaris, and A. R. Kaye. Modeling and Simulation of Self-Similar VBR Compressed Video: A Unified Approach. In *Proc. ACM SIGCOMM '95*, Boston, August 1995.
5. N. G. Duffield and N. O'Connell. Large Deviations and Overflow Probabilities for the General Single-Server Queue, with Applications. Technical Report DIAS-STP-93-30, Dublin Institute for Advanced Studies, 1993.
6. C. Huang, M. Devetsikiotis, I. Lambadaris, and A. R. Kaye. Fast Simulation for Self-Similar Traffic in ATM Networks. In *Proc. IEEE ICC '95*, Seattle, June 1995.
7. D. D. Clark, S. Shenker, and L. Zhang. Supporting Real-Time Applicatiopns in an Integrated Services Packet Network: Architecture and Mechanism. In *Proc. ACM SIGCOMM '92*, 1992.
8. R. L. Cruz. A Calculus for Network Delay, Part I: Network Elements in Isolation. *IEEE Trans. Inform. Theory*, 37(1):114–131, Jan. 1991.
9. A. I. Elwalid and D. Mitra. Effective bandwidth of general Markovian traffic sources and admission control of high speed networks. *IEEE/ACM Trans. Networking*, Vol. 1(3), June 1993.
10. J. Kurose. On Computing Per-Session Performance Bounds in High-Speed Multi-Hop Computer Networks. In *Proc. ACM SIGMETRICS '92*, 1992.
11. The ATM Forum Technical Committee. *Traffic Management Specification Version 4.0*, Apr. 1996.
12. M. Grossglausser, S. Keshav, and D. Tse. RCBR: A Simple and efficient Service for Multiple Time-Scale Traffic. In *Proc. ACM SIGCOMM '95*, pages 219–230, 1995.
13. I. Norros. On the Use of Fractional Brownian Motion in the Theory of Connectionless Networks. *IEEE JSAC*, 13(6):953 – 962, 1995.
14. C. Huang. *Long Range Dependent Traffic: Modeling, Simulation and Congestion Control*. PhD thesis, Carleton University, Jan. 1997.