# ADMISSION CONTROL AND ROUTING OF SMOOTHED VIDEO STREAMS

CHRIS BEWICK, RUBEM PEREIRA, MADJID MERABTI

{C.Bewick, R.Pereira, M.Merabti}@livjm.ac.uk
*Distributed Multimedia Systems Research Group, School of Computing & Mathematical Sciences, Liverpool John Moores University, Byrom Street, Liverpool, L3 3AF. UK*

**Abstract:** The emerging service-based digital networks for applications such as Video-on-Demand depend on packet delivery in real-time. The problem is compounded when a number of streams are sent simultaneously. This paper describes a novel connection admission control mechanism that works with smoothed video from our Network Constrained Smoothing algorithm. Our approach is two-pronged: network rerouting of particular smoothed intervals and connection renegotiation. We alter the routing on a per-interval basis when admission control cannot accept new connections that break aggregate network throughput thresholds. We present simulation results that show quantitative benefits - enhanced network performance from improved spatial and temporal load balancing using our admission control and smoothing techniques.

*Keywords:* Traffic Smoothing, Admission Control, Video Streams, Multimedia Networking, Real Time Traffic.

## 1. INTRODUCTION

In recent years, much research has treated the handling and characterising of digital multimedia traffic. The voluminous multimedia data is usually compressed, possibly in real-time, using sophisticated algorithms that remove spatial and temporal redundancy to make possible the highest compression-to-quality ratios. One industry-standard, MPEG-4, is scalable from low bit-rate ISDN connections at 64kbps to broadcast quality transmissions running in excess of 1Mbps. Intuitively, to provide a quality networked multimedia service, we must first capture the character of multimedia data and describe effectively the transmission requirements to the network. The compression ratio constantly changes to maintain consistent playback quality, generating a *variable bit-rate* (VBR) traffic pattern when transmitted. This type of compression is used in the *DVD* format and *Windows Media Player*. However, both these technologies have performance issues when sent over a distribution network such as the Internet. This is despite the enormous reduction in bandwidth requirements from using high-ratio MPEG compression techniques on untreated multimedia streams. Thus, the challenge is more complex than it may first appear, and we must investigate deeper than the application or transport layers, i.e. into the network infrastructure.

Our vision for multimedia networking is services and networks that share resources fairly between a potentially large number of long-lived, real-time multimedia connections without degrading audio/visual quality. In this paper, we address the problem of inefficient utilisation of network and client/server resources at peak times of demand. Moreover, our motivation for this is the open research issue of resource allocation for network equipment vendors – how to best make use of information about resource-hungry connections such as streaming multimedia.

The remainder of the paper is organised as follows. In the next section, we further review related work and highlight the gaps we address in this paper. In section 3, we discuss the techniques used in our *network constrained smoothing* (NCS) algorithm. In section 4, we discuss our admission control and routing mechanism for NCS smoothed intervals of video. In section 5, we present results and evaluate. We discuss conclusions and further work in section 6.

## 2. RELATED WORK

The support for the QoS required for compressed video is a non-trivial issue due to the highly variable bandwidth requirement. The MPEG-4 compression, reviewed in [Koenen, 1999], compresses media into scenes composed of variable bit rate encoded objects. The total transmission size of the stream is greatly reduced, however exhibits very bursty characteristics that vary over multiple time-scales. For the network operator, it is possible to use statistical multiplexing to allocate resources and perform admission control. The *effective envelope*, proposed in [Boorstyn et al, 2000], places bounds on aggregate traffic requirements over a single node with high certainty. However, they do not show results using smoothed video traffic. The network utilisation decreases when the parameters are inefficiently selected for the shape of the traffic. We note the benefits of using these techniques together

with smoothing to shape independent traffic streams and statistical multiplexing to allocate resources from aggregate bandwidth requirements.

With pre-recorded media, it is possible to use *a priori* knowledge about video frame sizes to perform traffic smoothing in a technique called *work-ahead smoothing* [Salehi et al, 1998]. Buffers are required to store the video data at the client and server. The client receives data in advance of playback, and the server follows a transmission schedule that generates a series of *constant bit rate* (CBR) segments at different rates. In this way, traffic spikes are reduced and the variability is less. Salehi et al [1998] propose and prove an algorithm for *optimal smoothing* that for a given buffer size, the schedule has minimum variability and number of rate changes. We note that *optimal smoothing* incorporates no notion of any QoS negotiation, and we propose an integration of these negotiations at connection set-up time with traffic smoothing. The on-line smoothing research published by [Rao and Ravanghan, 1999] combines the speed and flexibility of on-line algorithms with the benefits of "optimal" buffer constrained *a priori* knowledge of pre-recorded video. The real attraction of smoothing on-line at connection set-up time is that the network can make negotiations of for both admission control and routing. Intuitively, equal-sized CBR intervals are inherently advantageous for network connection management for reason of simplicity.

The algorithm presented in [Hadar and Cohen, 2001] structures the smoothed CBR segments into equal-sized intervals. They present results that consider using buffers that are several-fold larger than the 64k-1MB range of buffers used in the analysis in [Salehi et al, 1998] of the *optimal smoothing* algorithm. The larger buffers enable the research to focus on criteria such as interval frequency and multiplexing gains. They extend the work in [Hadar and Greenber, 2000] and describe the *enhanced Piecewise Constant Rate Transmission and Transport* (e-PCRTT). The schemes for CBR interval-based smoothing fit the networks that allow hosts to renegotiate their traffic requirements on-the-fly after initial connection set-up, for example *Renegotiated Constant Bit Rate* (R-CBR) proposed in [Grossglauser et al, 1997].

However, none of these smoothing algorithms use information about network utilisation and they do not consider the implications for admission control in the network. In this paper, we present the admission control mechanism that works together with NCS to increase network utilisation and avoid congestion at peak times.

## 3. NETWORK CONSTRAINED SMOOTHING

Resources allocation in the network subsystem requires an accurate but manageable description of multimedia traffic. Compressed MPEG video exhibits rate variability on multiple time-scales due to movie scene changes, and the frame size fluctuates regularly according to the GOP (Group Of Pictures) sequence of I, P and highly compressed B frames.

We have developed a flow specification that can capture traffic variability in a sequence of CBR intervals of equal length. For this paper, we use an interval size of 1500 frames, which represents 1-minute of PAL video at 25 fps. The implications of allocating resources using CBR intervals are significant for the long-term behaviour of aggregate network loading. This is particularly remarkable if the multimedia streams are synchronised, since all the rate changes occur around the same time.

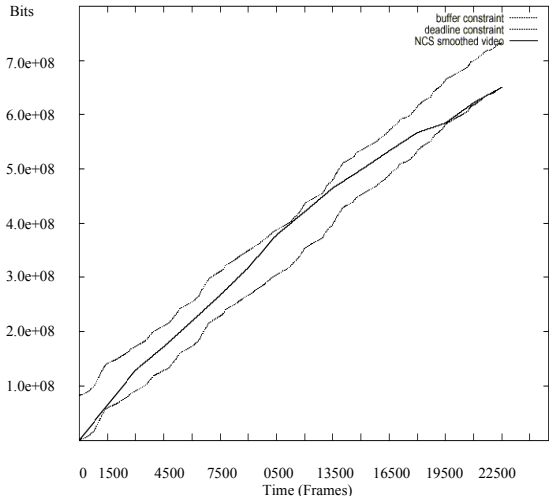


Figure 1: Cumulative bit rate of Network Constrained Smoothing

In Figure 1, we show an example of a smoothed stream, using the NCS algorithm. The lower line is the original pre-smoothing stream's delivery deadline, and the difference between the upper line and the lower line indicate the buffer space allocated. The line in the middle is the smoothed schedule, which should fall fully between the lower and upper lines.

The number of frames in the video may not fit exactly into the intervals, and will most likely begin at a time other than an interval start time. Our scheme for 4 intervals is shown below in Figure 2.

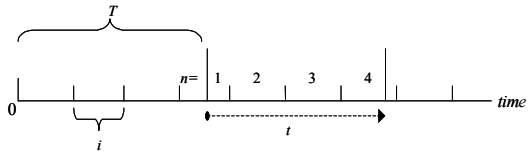


Figure 2. Video stream with shorter starting and ending intervals

The video playback begins at time *T*, the number of frames since the system started, thus we can calculate the length of the first and last intervals using the interval length *i*. All intervals in-between are of equal-size, *i* frames.

For any video traffic smoothing algorithm, including NCS, the usefulness and accuracy of the network loading measurements are an important issue. So far, we have explored the use of equal-sized intervals within a smoothing algorithm. However, we extend this definition to the network loading aggregate. Thus, the change from one interval to the next will occur at the same instant for *all* video streams. This means that, for each stream, all the intervals except the first and last are equal-sized. This has some implications for connections over multi-hop networks, where independent connections may each experience a different delay. To address this, we developed a method of synchronising the connections, and this became a integral part of the tools that we required for the NCS algorithm.

### 3.1 Synchronising Aggregate Video Traffic

The transmission schedule for NCS is made of equal-sized time intervals. However, when multiplexing a number of streams, there are an undefined number of changes in the aggregate traffic level. If interval changes are synchronised for all streams, then the aggregate network loading will only change at interval boundaries. We proposed a novel method for synchronising aggregate video traffic for use with our smoothing algorithm in Bewick et. al [2002]. In this way, the intervals from independent streams are forced to begin and end at set times and increases in resource requirement now always occur at the same instant in time.

The intervals are easy to manage within tables of information about future network link-states. However, new streams may start or finish in the middle of intervals, thus the loading is the *maximum* predicted traffic loading for each interval. When new connections arrive we perform deterministic admission control to accept or reject video connections.

Often two independent streams experience different delays on part of the network because they originate from servers located some distance or number of hops away. Here, network propagation and queuing delays cause synchronised streams to intercept on a link at slightly different times. We have developed a mechanism to overcome this. In essence, we introduce a halt to server transmission for a short time period between segments - the *minor segment*, of the order of hundreds of milliseconds. The larger segments we call *major segments*, and during which the video server transmits the video interval.

### 3.2 A Model for Network Load Prediction

NCS requires information about predicted or future network loading. We store this loading in network link-state tables that are generated using the aggregate of the synchronised intervals from all supported streams. We exploit the notion that multimedia streams are long-lived by using tables that contain time-variant information about link-states on a per-interval basis. In addition, it is possible to use predictions for user behaviour, although this is not addressed in this paper. For example, we wish to allocate extra bandwidth and routing resources in the early evening when more users connect to watch the daily news bulletins.

Consider the VoD server with a connection to a network, shown in Figure 3 connected to node 1 in this topology. A user who is connected to node 5 requests a movie from the server. The links from router A to E are core network and we use the link-state information on the path between nodes 1 and 5 for smoothing and admission control.

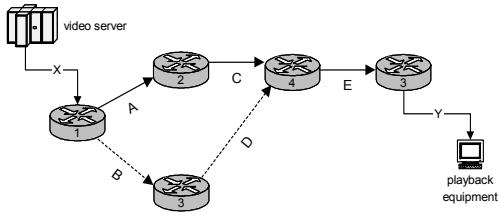


Figure 3: Network loading along the video connection path

The network prediction model describes values for the expected network loading over *N* equal-sized time intervals in terms of the available bandwidth. We use a simple table of data to store a network loading value for each interval in the future until *N*. The table is easy to manage, and can be updated when new video streams connect or there is information about pre-booked or scheduled video transmissions. The table in Figure 4 contains the values for the total bandwidth minus the sum of the smoothed traffic from all supported streams on the network for each interval at time $t_n$ to $t_{n+N}$ where *n* is the current interval.

| Interval Link | $t_{1-3}$ | $t_{4-9}$ | $t_{10-12}$ | $t_{13-15}$ |
|---|---|---|---|---|
| A | **7.50*** | 7.00 | 4.00 | 6.00 |
| C | 8.50 | **6.25*** | **2.50*** | 7.00 |
| E | 9.00 | 9.00 | 9.00 | **5.00*** |
| **Critical link** | **A** | **B** | **B** | **E** |

Figure 4. Calculating the critical link

In the table, the lowest entry for each interval column, marked with an asterisk, is a network bottleneck. The table shows the network bottlenecks for each link. Note that the bottleneck is not always on the same link. We have previously introduced the notion of the *critical-link*: the link bottleneck for a particular time interval. Our load balancing strategy is to handle network performance trouble spots, and ideally we wish to use the values for the *critical link* for network resource allocation and admission control. By using tables at network admission control nodes, we capture the shape of aggregate network loading over time for the route between servers and clients. For example, for a 3-hour video, 180 intervals would be needed in a table with space for each of the 3 links from node 1 to 5 in the path from Figure 3.

### 3.3 The Network Constrained Smoothing algorithm

The smoothing algorithm selects the CBR rate for each interval of video data in the server transmission schedule using information about the character of future network traffic. The careful selection of CBR intervals results in lower bit-rate intervals at busy times on the network, thus the network can support more connections. One key requirement of our algorithm was to avoid pushing complexity into the core of the network, where low time-complexity processing is critical for routers. The NCS is performed on the video server, and the network edge-nodes police the aggregate bandwidth allocated to the server using thresholds set in by the Service Level Agreement (SLA). Furthermore, NCS scales to a high number of streams because there is a cumulative effect that reduces peaks and smoothes the *aggregate* network loading.

The NCS algorithm is explained in more depth in Bewick et al. [2002]. In essence, NCS works to increase the number of simultaneous connections that a network can support without service degradation through overloading. Admission control maintains the QoS level provided by the network.

## 4. ADMISSION CONTROL AND ROUTING

Commercial-grade multimedia services require the network to provide QoS to a finite number of streams. The number of streams is controlled at the network edge by a Connection Admission Control (CAC) mechanism. In our example, a video server has a 10-Mbps uplink to the network, and when videos are requested, the server issues a request for bandwidth and resources from CAC. We require a mechanism that can check the network link-state table and take action when a pre-defined threshold of throughput is reached during one or several of the intervals. Our CAC scheme either refuses the connection or selects another route for this interval or set of intervals. In this way, a single video connection can have more than one route through the network. We mark individual packets at the network-edge with a *flow-id* so that intervals can be correctly routed along different paths.

Our CAC mechanism is a sequence of steps starting with a connection request:

1. Server requests connection based on connection-duration metrics, e.g. average throughput
2. Route selection
3. Pass critical links to server
4. Smooth video connection into *N* intervals using NCS
5. Admission control for each interval, $t_n$ up to $t_{n+N}$
   - If available bandwidth on critical link is less than threshold *T* :
     - Reroute interval *n* if possible
   - Mark interval *n* with *flow-id*
   - Update link-state tables
6. Video transmission starts

The algorithm above will reject connections in two cases: if the bandwidth threshold is exceeded on a rerouted critical link, or rerouting is not possible when there is no alternative route. The information from the network loading prediction model is used to generate the link-state tables. In our example network above, we have a shape, shown in Figure 5, for the predicted network loading along the route selected as links A,C, and E.

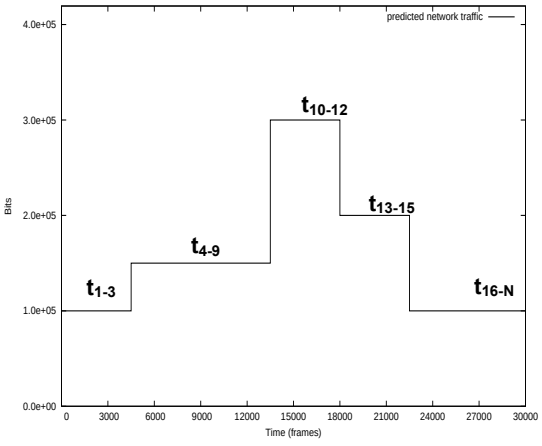


Figure 5 Predicted network loading

The loads on the links other than the critical link, marked with an asterisk, are shown in Figure 6. The values for available bandwidth along links B and D are also shown during the intervals $t_{10\text{-}12}$. The shaded boxes represent the route for each interval.

| Interval / Link | $t_{1\text{-}3}$ | $t_{4\text{-}9}$ | $t_{10\text{-}12}$ | $t_{13\text{-}15}$ |
|---|---|---|---|---|
| A | **7.50*** | 7.00 | 4.00 | 6.00 |
| B | | | 5.00 | |
| C | 8.50 | **6.25*** | **2.50*** | 7.00 |
| D | | | 3.50 | |
| E | 9.00 | 9.00 | 9.00 | **5.00*** |

Figure 6. Bandwidth available - calculated from network loading

When the video server requests a new connection in the network, our admission control mechanism performs tests to see whether the network can support it. If intervals break a throughput threshold, they are rerouted along a different path that is less congested, for example links B and D in the table above. If the connection passes the admission tests for the new path, then the connections' bit rate for each interval is subtracted from the available bandwidth of the respective interval in the network loading tables.

The CAC mechanism and the rerouting technique together achieve enhanced spatial traffic distribution on the network. Furthermore, the traffic is smoothed to lower aggregate throughput at time of peak loading when used with NCS intervals of video.

## 5. EVALUATION

In this section, we evaluate the admission control using NCS and MPEG-4 compressed video. We simulate a video server with a fixed service contract for network bandwidth. We present comparative results of NCS and Optimal Smoothing and also evaluate the advantages of rerouting NCS intervals at peak times and enhanced spatial network load balancing.

The experiments in this paper are based on a library of MPEG-4 video traces, which were presented and discussed in [Fitzek and Reisslein, 2001]. These traces are part of an on-line archive available at: http://www-tkn.ee.tu-berlin.de/research/trace/trace.html.

The simulation measures the network utilisation during the streaming of the smoothed multimedia data across the network of 10-Mbps links, shown in Figure 3.

### 5.1 Simulation

We developed a simulator written in *C* for testing scenarios. There are two main aspects of the workload characterisation in the simulation: Firstly, the video request arrival pattern at the video server, and secondly the predictions for background traffic loading at the system start time. The simulator runs the NCS algorithm in an easily configurable environment that is set up in a configuration file that we edited for scenario description, parameters such as the buffer size, the names of the trace files and their start times. The trace data is from MPEG-4 video, stored as a simple list of per-frame sizes in bits, and a video-server would retrieve this from the locally attached storage-device. The NCS algorithm uses this data and the video server per-stream buffer size for deadline and buffering constraints respectively. When a new connection begins, the network loading table is updated by adding the total of the smoothed traffic. Thus, it is the smoothed traffic schedules from new connections that are part of the input for the network load prediction model.

During a simulation run, the NCS algorithm processes the streams when they are scheduled to appear in the scenario configuration file. The new stream is smoothed into intervals that are synchronised to the start and end of the other time intervals supported in the simulated video server. After a complete simulation run, the results show the performance of the NCS algorithm as the streams connect from start to finish.

### 5.2 Results

Firstly, we consider the results comparing NCS with Optimal Smoothing. The graphs in Figures 7(a) and 7(b) show a number of benefits of using Network Constrained Smoothing over video smoothing techniques that do not consider the loading on the network. The network bandwidth is better utilised at times of peak loading. For example, during the bottleneck interval 10 the aggregate video traffic that our video server sends is 20% lower, reduced from 2.85-Mbps to 2.26-Mbps. In the case of optimal smoothing, rate changes may occur at any moment in time, and there can be substantially more rate changes than the synchronised intervals generated by network constrained smoothing. Under NCS, this facilitates the handling of loading information for network edge-routers that calculate admission control and update network link-state tables.

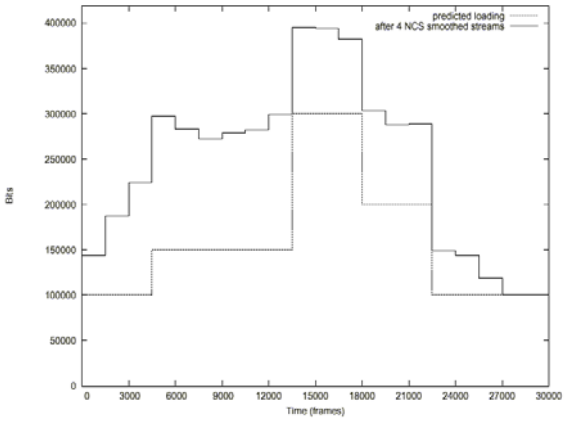


Figure 7(a): Predicted loading and NCS aggregate traffic

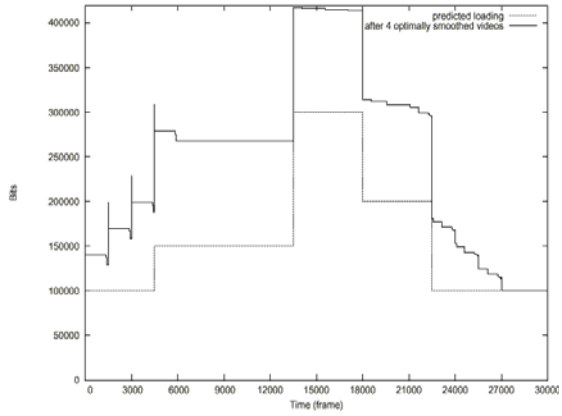


Figure 7(b): Predicted loading and Optimal Smoothing aggregate traffic

Another way to consider the effect of NCS is to observe the buffer occupancy at the client side. In the graph presented in Figure 8, we compare the NCS with Optimal Smoothing regarding the buffer occupancy during transmission of their respective streams.

The NCS will utilise buffer space to choose lower traffic rate for when the network is experiencing a high load. When the throughput is reduced, the video data is retrieved more rapidly from client buffer for playback. Thus, with NCS we expect to see a decrease in buffer occupancy at times when network loading is relatively high.

Other smoothing algorithms ignore the effect of network loading, and instead use constraints such as variability and number of rate changes. For example, *optimal smoothing* does not use information about network loading, and for a large buffer size there are only a few CBR intervals in the transmission schedule. Thus, when the buffer is of a large size, above 4-MB in this case, the buffer is under-utilised and increasing the buffer size has no impact on the allocation of intervals during smoothing.

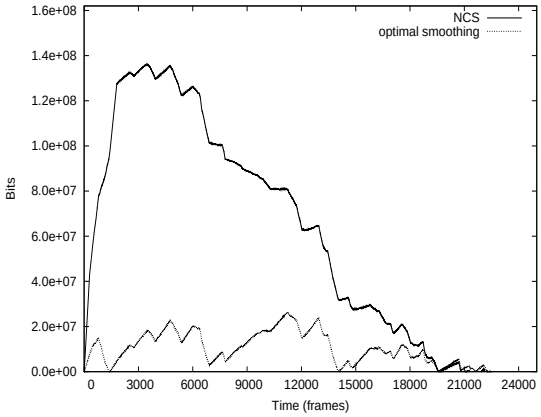


Figure 8 Buffer Occupancy during transmission of video under NCS and Optimal Smoothing.

The buffer occupancy peaks at less than 20%, as shown in the figure above. This leaves much of the buffer resources unused. We have shown how NCS uses larger buffers to store more *work-ahead* video data. Our results show how both buffer and network utilisation are improved during times of high loading with using the NCS.

Next, we discuss the benefits of rerouting individual intervals. The videos used in these experiments are approximately 15-minutes long and each requires more than 0.75-Mbps of average bandwidth. We used the predicted loading on the network from the table shown in Figure 6 to smooth the first video request appearing in the simulation scenario file. For example, the NCS algorithm generated a drop in transmission bit-rate at intervals $t_{10\text{-}12}$ because the network prediction model has the lowest available bandwidth at this time. This effect is cumulative as more streams are added to the system. We run the simulation for a scenario configuration file that schedules 3 more streams, each arriving 1,500 frames apart, and we show the result is Figure 9 below. The dotted lines show the network loading and prediction before rerouting.

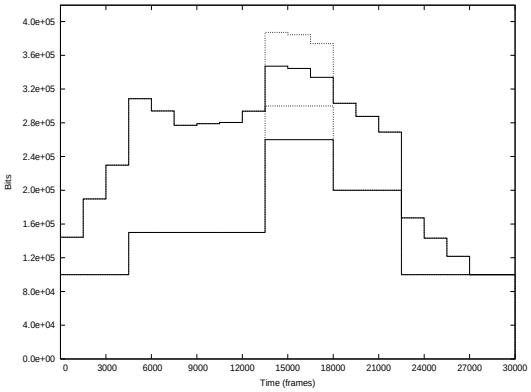


Figure 9. Aggregate traffic when re-routed

We show a number of benefits of using an admission control mechanism that uses rerouting. The bandwidth requirements are reduced at times of peak loading, permitting more video connections over the network. The intervals mean that the loading information is manageable for admission control and the update of network link-state tables is simple, shown in Figure 10.

| Interval Link | $t_{1-3}$ | $t_{4-9}$ | $t_{10-12}$ | $t_{13-15}$ |
|---|---|---|---|---|
| **A** | **2.50** | 3.75 | 2.00 | 3.50 |
| **B** | | | 5.00 | |
| **C** | 3.50 | **3.00** | **0.50** | 4.50 |
| **D** | | | 3.50 | |
| **E** | 4.00 | 6.75 | 7.00 | **2.50** |

Figure 10. Bandwidth available when 4 videos are transmitted

The admission control mechanism reroutes links A and C because the available bandwidth on link C drops below 1.5-Mbps. The alternative route is links B and D, either at or above the threshold of 1.5 Mbps, shown in Figure 11 as shaded boxes for the links in the re-routed path. This is a successful load balancing that improves network utilisation when the intervals $t_{10-12}$ are rerouted, shown as the solid lines below the dotted lines in Figure 9.

| Interval Link | $t_{1-3}$ | $t_{4-9}$ | $t_{10-12}$ | $t_{13-15}$ |
|---|---|---|---|---|
| **A** | **2.50** | 3.75 | 4.00 | 3.50 |
| **B** | | | 3.00 | |
| **C** | 3.50 | **3.00** | 2.50 | 4.50 |
| **D** | | | **1.50** | |
| **E** | 4.00 | 6.75 | 7.00 | **2.50** |

Figure 11. Bandwidth available after re-routing intervals $t_{10-12}$

Using our admission control mechanism and NCS, we have shown the advantages of intervals that are smoothed on the temporal axis, and load balanced spatially on two routes during the connection duration.

## 6. CONCLUSIONS AND FURTHER WORK

In this paper we have presented a novel mechanism for connection admission control of smoothed video traffic. Our mechanism works on two levels: we smooth traffic according to the critical links that are most heavily loaded. Secondly, the admission control mechanisms checks for links along the route that are over a throughput threshold. We have discussed rerouting these links to another path under less traffic load, and the evaluation shows the advantage of spatial load balancing. NCS maps the QoS requirements and the buffer limitations at the client/server side to smoothed intervals of video data that are manageable for the network. This is important for traffic policing nodes at network edges, and load-profiling in the backbone links. We have shown how our method for synchronising the traffic of each video connection improves aggregate traffic manageability and copes with the delays along different network paths. The results show that it is possible to take advantage of workahead smoothing and admission control within a scheme for traffic manipulation that best suits the expected network loading.

The further work is an extension of our admission control mechanism; we are integrating the NCS algorithm into a management and control framework for load balancing for networks that support the QoS for video traffic. Specifically, we are considering a new management component that is responsible for distributing the *critical-link* information to NCS control components operating at the network edge. We expect results to show scalable link-state tables for interval-based information about future network loading.

We have shown that our scheme allows the network to support more connections than was possible with previous smoothing schemes. This leads to more efficient use of QoS networks, thus promoting the development of distributed multimedia services.

## REFERENCES

Bewick C., Pereira R and Merabti M. 2002 "Network Constrained Smoothing: Enhanced Multiplexing of MPEG-4 Video". 7th IEEE International Symposium on computers and Communications July 2002. Taormina, Italy. Pp 144-119.

Boorstyn R., Burchard A, Liebeherr J., Oottamakorn C. 2000, "Effective Envelopes: Statistical Bounds on Multiplexed Traffic in Packet Networks". in Infocom 2000. Tel Aviv, Israel. Pp 1223-1232.

Fitzek F.H.P. and Reisslein M 2001, "MPEG-4 and H.263 Video Traces for Network Performance Evaluation." IEEE Network, Dec. 2001, **15**(6). Pp 40-54.

Grossglauser, M.,. Keshav S. , and D. Tse 1997, "RCBR: A Simple and Efficient Service for Multiple Time-Scale Traffic." IEEE\ACM Transactions on Networking, Dec. 1997. Pp 741-755.

Hadar O. and Cohen R. 2001, "PCRTT Enhancement for off-line Video Smoothing." The Journal of Real Time Imaging, June 2001 **7**(3).Pp 301-314

Hadar, O. and Greenberg S. 2000,. "Statistical Multiplexing and Admission Control Policy for Smoothed Video Streams Using e-PCRTT Algorithm." in International Conference on Information Technology: Coding and Computing (ITCC'00). 2000. Las Vegas, Nevada.Pp 272-277

Koenen, R. 1999, "MPEG-4 - Multimedia for Our Time." IEEE Spectrum, 1999. **36**(2): p. 26-33.

Rao S.G. and Raghavan S.V. 1999, "Fast Techniques for the Optimal Smoothing of Stored Video". Multimedia Systems, 1999, **7**(3): p. 222-233

Salehi, J., Zhang Z., Kurose J., Twosley D. 1998, "Supporting Stored Video: Reducing Rate Variability and End-to-End Resource Requirements Through Optimal Smoothing". IEEE/ACM Trans. Networking, 1998. **6**(4).Pp 397-410

Chris Bewick is a research student currently finishing his PhD Thesis at Liverpool John Moores University, UK. His research interests include QoS networking, video traffic characterisation, admission control, wireless, and peer-2-peer video distribution.

Dr. Rubem Pereira is reader in Multimedia Networking at Liverpool John Moores University. He completed his PhD in performance evaluation of Computer Networks at the Manchester Metropolitan University in 1998.

Prof. Madjid Merabti is Director of the School of Computing and Mathematical Sciences, Liverpool John Moores University. He has published over 80 conferences and journal papers in Distributed Multimedia, Computer Networks, Security and related areas.