Global Time for Interactive Applications over *Global* Packet Networks

Mario Baldi[1,2] and Yoram Ofek[1]
[1] Synchrodyne Networks, Inc., New York, NY
{baldi,ofek}@synchrodyne.com
[2] Torino Polytechnic, Computer Engineering Department, Torino, Italy
mario.baldi@polito.it
www.polito.it/~baldi

Abstract. This work presents that *global time* (a.k.a. time-of-day or coordinated universal time – UTC) is essential in maximizing user perceived quality of service, while eliminating both switching bottlenecks — critical in very high capacity network core — and communications link bottlenecks — at the low speed access (e.g., wireless and DSL). Global time obtained, for example, from GPS (Global Positioning System) or Galileo, is used in the design of all streaming media applications such as toll quality telephony, videotelephony and videoconferencing. The proposed solution, that can be applied to the Internet without changes to any of the existing protocols, provides a guaranteed quality service to each application without requiring nodes to keep state information on microflows.

1 Introduction

The deployment of new high bandwidth multimedia applications will boost network traffic and consequently the deployment of very high capacity transmission technologies, such as Wavelength Division Multiplexing (WDM). On the other side, since multimedia services will have to be widely available, various “low speed” access technologies, such as wireless, DSL, and cable modem will be deployed. In this scenario networks will be characterized by (*i*) *electronic switching* bottlenecks and (*ii*) *communications link* bottlenecks that are created by the bandwidth mismatch between high capacity core technologies and low speed access technologies.

Global time (a.k.a. time-of-day or coordinated universal time – UTC) enables to implement *time-driven priority* (TDP) forwarding which eliminates communications link bottlenecks since it completely avoids congestion, also in bandwidth mismatch points. Moreover, the design of TDP-based packet switches is highly scalable because it is based on simple switching fabrics without any speedup with respect to input link capacity.

Many interactive applications, such as, telephony, videotelephony and videoconferencing, require at the receiver *continuous playing* of the samples captured at the sender, as shown in Figure 1. Continuous playing requires a *constant delay* service to be provided by the application layer, i.e., where samples are acquired and played.

Figure 1 shows the generic model of an application requiring continuous playing and highlights the components of the end-to-end delay.

- The processing delay (**P**) is introduced on the multimedia desktop side. It encompasses the time needed for analog-to-digital conversion and coding (e.g., voice or video compression).
- The network delay (**N**) is the time needed to move packets across the network; it also includes the shaping and packetization delays.

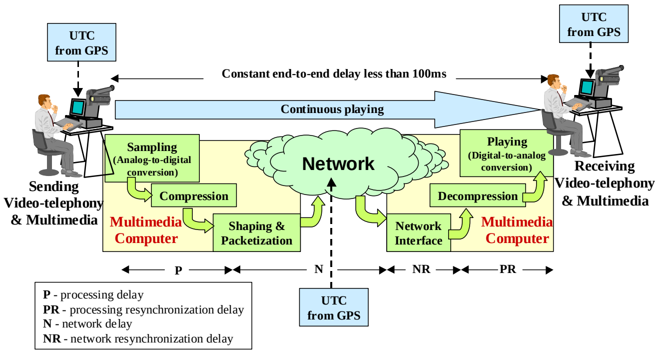

Figure 1: Generic Model of an Application Requiring Continuous Playing

Since the above delay components can vary during a session, some specific action is required at the receiver to keep constant the end-to-end delay between the application layers, thus enabling continuous playing. Before samples are played, delay variations should be "smoothed out" by buffering the samples that have experienced (in the network and in the decoder) a delay shorter than the maximum. This introduces two *resynchronization delay* components that are typically the time spent in a *replay buffer* [4]. This time is such that all samples on exiting the replay buffer have experienced the same delay since the time they were acquired at the sender side. Such an overall delay is equal to or larger than the *delay bound* the system can guarantee.

- The processing resynchronization delay (**PR**) cancels delay variations introduced by the sample coding process.
- The network resynchronization delay (**NR**) cancels variations of the delay experienced in the network (e.g., the delay jitter due to queuing in the network).

The processing delay (**P)** depends on the signal processing performed, which differs, for example, for voice and video. A delay below 100 ms gives human communicators the feeling of live interaction. Since in a global network the propagation delay alone is about 100 ms, every other delay component should be kept as short as possible. This paper shows that global time enables the other delay components (**N, NR,** and **PR**) to be minimized independent of the packet technology (e.g., ATM or IP) deployed and the rate of sessions.

The two resynchronization components, **PR** (processing resynchronization delay) and **NR** (network resynchronization delay), can be kept small, e.g., 25-125 μs. The network delay (**N**) is propagation delay plus a small additional delay per switch, e.g., 50-250 μs. Thus, global time enables the end-to-end delay bound to be minimized [5].

Global time is used in two ways:

1. To implement time-driven priority (TDP) forwarding of packets in global networks, which
   i. guarantees a maximum queuing delay of a few milliseconds, independent of the flow rate and the network load, also in bandwidth mismatch points;

ii. enables the implementation of efficient packet switch architectures based on low complexity switching fabrics. This increases the scalability of switches and eliminates the electronic switching bottleneck.

2. To synchronize the acquisition of samples at the sender (e.g., video capture card) and their continuous playing at the receiver (e.g., video display) with one another and with the TDP forwarding.

2 Network Architecture and Deployment

According to various provisioning models, such as, Integrated Services [6] over the Internet and ATM User Network Interface (UNI), applications signal their Quality of Service (QoS) requirements to the network. If the network has enough resources to satisfy the request, they are reserved and packets transmitted by each application are handled in a way that QoS is guaranteed to their flow (usually called *micro-flow*). Most of the queuing algorithms used to implement such packet handling have to maintain status information for each micro-flow, which is recognized not to be scalable. Time-Driven Priority (TDP) forwarding does not require per micro-flow information in intermediate nodes. Thus, TDP has similar provisioning scalability as forwarding equivalent class (FEC) in IP/MPLS traffic engineering (TE).

In accordance to the Differentiated Services [7] model over the Internet, micro-flows should be aggregated in the network core in order to improve scalability by increasing the granularity with which switches handle packet flows.

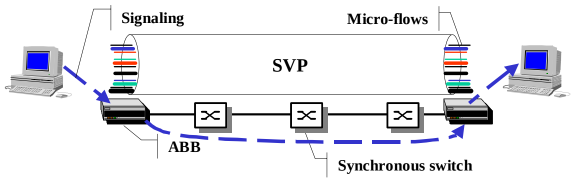

Figure 2: Synchronous Virtual Pipe (SVP) and Access Bandwidth Brokers (ABBs).

Synchronous Virtual Pipes (SVPs) can be set up over networks deploying global time in order to aggregate multiple micro-flows, thereby relieving core nodes from participating in micro-flow level signaling. An SVP can be regarded as a virtual leased line. In order to deterministically guarantee QoS to single micro-flows, Access Bandwidth Brokers (ABBs) at the edges of an SVP handle signaling requests from the applications whose packet micro-flows are to traverse the SVP, and determine the availability of resources within the SVP. If a request is accepted, the ABB reserves a fraction of the SVP resources to the corresponding micro-flow. As shown in Figure 2, intermediate switches are not involved in the signaling operation, but the micro-flow will receive deterministic QoS guarantees, even though intermediate switches on the SVP do not have any awareness of the micro-flow.

The Internet (as well as ATM networks) is based today on asynchronous packet (cell) switches which do not feature TDP forwarding. Thus, especially in the initial deployment phase, TDP switches will coexist and interoperate with current asynchronous packet switches. Figure 3 shows a scenario, likely to be common in the early days of TDP deployment, in which end stations connected to asynchronous local area or access networks communicate through a TDP backbone. Synchronous *boundary* *nodes* control the access to SVPs set up on the synchronous backbone performing both policing and shaping of packets flows– i.e., they *synchronize* packet forwarding. TDP provides the minimum delay bound when deployed end-to-end, but it can be beneficial even when its use is confined to subnetworks. The node at the ingress of a TDP subnetwork, which shares the global time reference, eliminates the delay variation experienced by packets in the asynchronous network; then packets benefit of the controlled delay service provided by the synchronous subnetwork.

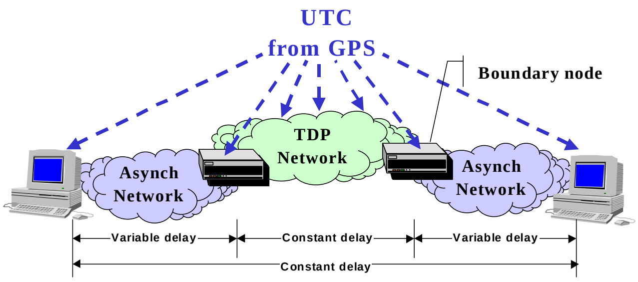

Figure 3: Interoperation between TDP Networks and Asynchronous Networks

The delay jitter introduced by asynchronous segments can be completely canceled in the end-to-end communication if end stations use the common global time to attach time stamps to the packets they generate. The Real-time Transport Protocol (RTP) [8] can be used to carry the time stamp. The source end system generates the time stamp according to the common global time. The node at the ingress of a TDP subnetwork can take advantage of the time stamp to eliminate the delay variation across the asynchronous network. For example, assuming a known delay bound on the asynchronous subnetwork, say of k time units, the ingress TDP switch determines the time the packet should be forwarded by adding k time units to the time stamp value.

Packets travel through the TDP cloud with controlled delay and no loss. With reference to Figure 3, when packets arrive to the destination they have experienced a variable delay in the asynchronous access subnetwork to which the destination is connected. By knowing the delay bound, say of b time units, on the asynchronous segments traversed by the packet, the end station can completely eliminate the jitter by determining the replay time as the time stamp value plus b.

The same approach can also be applied when packets traverse more than one TDP subnetwork and more than two asynchronous networks. SVPs can be set up over multiple synchronous subnetworks interconnected by asynchronous ones. Access devices at the ingress of synchronous segments resynchronize packets that have experienced variable delay across asynchronous subnetworks so that the overall delay throughout the SVP is constant.

3 Global Time and Periodic Forwarding: Time-Driven Priority

All packet switches are synchronized to a global common clock with a basic time period that is called *time frame* (TF). The TF duration is derived from the UTC (coordinated universal time) second received, for example, from a time distribution system such as GPS, GLONASS, TWSTFT (Two-Way Satellite Time and Frequency Transfer), and, in the future, Galileo. For example, by dividing the UTC second by 8,000, the duration of each time frame is T_f = 12.5 to 125 μs; however, the time frame duration can be set as needed.[1]

TFs are grouped into a *time cycle*; Figure 4 shows an example of a time cycle that contains 100 TFs, i.e., there are 80 time cycles in a UTC second. Time cycles are further organized in *super cycles*, each of which typically equals one UTC second. This timing structure is useful to perform resource reservation in order to provide guaranteed services.

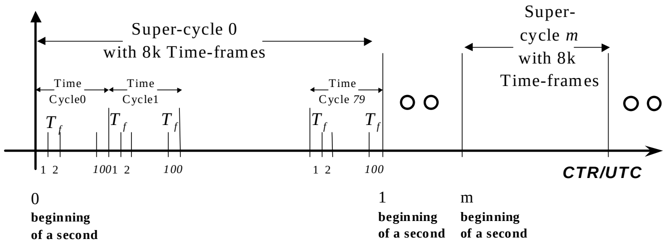

Figure 4: Global common time reference

Thus, all switches around the globe have an identical time structure that is collectively called a *Common Time Reference* (CTR). The CTR can be used to coordinate the acquisition of samples at the sender with the playing of them at the receiver. Moreover, the CTR enables the implementation of *Time-Driven Priority* (TDP) [1] [2] for periodically forwarding real-time packets, for example inside IP and ATM networks, as shown in Figure 5.

Periodic forwarding indicates that the forwarding pattern repeats itself in every time cycle and in every super cycle. TDP guarantees that the end-to-end delay jitter is less than one TF and that reserved real-time traffic is transferred from the sender to one or more receivers with no loss due to congestion.

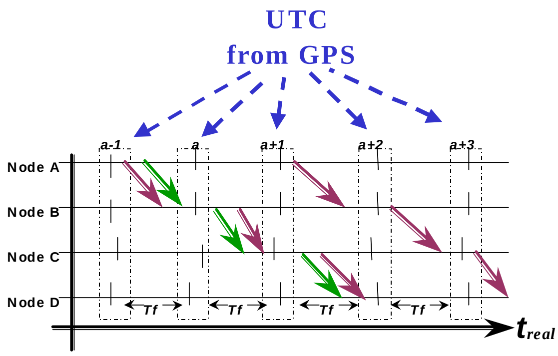

Figure 5: Periodic forwarding with time-driven priority (TDP)

[1] UTC receivers from GPS are available from many vendors for a low price (for example, the price of a one PPS (pulse per second) *UTC* clock, with accuracy of 10-20 nanoseconds, is about $200). By combining UTC from GPS with local Rubidium or Cesium clocks it is possible to have a correct UTC (±1 μsecond) without an external time reference from GPS for days (with Rubidium clock) and months (with Cesium clock).

The simple TDP operation is generalized by adding the following two conditions:

(i) All packets that should be sent in TF *i* by a node are in its output port before the beginning of TF *i*, and

(ii) The delay between an output port of one node and the output port of the next downstream node is a constant integer number of TFs.

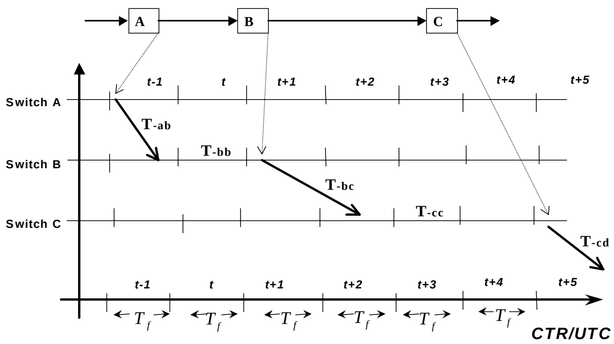

Figure 6: Generalized TDP with arbitrarily bounded link and switch delays

The generalized TDP forwarding, exemplified in Figure 6, is important because of the possibly long lasting software protocol processing in heterogeneous multi-protocol internetworking environments. In this case, a predefined, but fixed, number of TFs will be added in some intermediate switches with an increase in the end-to-end delay; the end-to-end delay jitter will remain constant.

In Table 1 some of the unique properties of TDP is compared with four types of communication networks.

Table 1: A comparison with other methods

<table>
<tr><th colspan="2">Communication methods
Service capabilities</th><th>Circuit Switching or PSTN</th><th>Single Async. Priority w/no reservation</th><th>Multiple Asynchronous Priorities – DiffServ</th><th>Time-driven Priority</th></tr>
<tr><td colspan="2"><u>Data:</u> mail, ftp, etc.</td><td>No</td><td>Yes</td><td>Yes</td><td>Yes</td></tr>
<tr><td rowspan="2"><u>Interactive</u> – on a Global Scale</td><td>Phone</td><td>Yes</td><td>No</td><td>Not proven: depends on scheme</td><td>Yes</td></tr>
<tr><td>Video-phone</td><td>Yes</td><td>No</td><td>Not proven: depends on scheme</td><td>Yes</td></tr>
<tr><td colspan="2"><u>Utilization</u> vs. <u>Loss</u>: with a mixture of high and low speed links</td><td>Full utilization and No loss</td><td>either Low utilization or High loss</td><td>Utilization can be low, and loss can be high depends on the specific scheme – Requires overprovisioning</td><td>Full utilization and No loss Easy to schedule</td></tr>
<tr><td colspan="2"><u>Experience</u></td><td>100+ years</td><td>25+ years</td><td>New technology</td><td>New technology</td></tr>
</table>

4 Periodic Bursty Services: Videotelephony and Videoconferencing

Videotelephony and videoconferencing, like telephony, rely on continuous playing at the receiver of samples acquired at a fixed rate at the sender. Samples are both voice and video frames captured by a camera and digitized by a frame grabber. Video frames have two main differences from voice samples.

1. The sampling rate is usually lower, from a few to 30 frames per second – versus the 8,000 voice samples per second required for voice encoding.
2. The amount of bits required to encode each video frame sample is much larger, at least a few kilobits – versus the 8 bit or less used for a single voice encoding.

When circuit switching is used to transfer video frames, the encoder is operated in such a way that it produces a constant bit rate flow. This is required in order to fully utilize the channel allocated to the session. Consequently, the transmission delay of a single video frame is the time between two successive video frames. This is because the transmission of the current video frame should continue, in a constant rate, until the next video frame is ready. For example, if the sampling rate is ten video frames a second, the transmission delay alone is 100 ms, which is unacceptable in interactive applications.

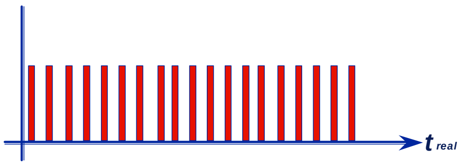

Figure 7: Periodic Bursty Transmission of a Video Stream

The elimination of such a long transmission delay is achieved by transmitting the captured video frame as a short burst. Packet switching allows burst transmission of video frames in packets, i.e., as shown in Figure 7, a video frame is captured, put into a packet, and then transmitted as a burst into the network. Therefore, the only way to transmit video frames for interactive applications with minimum delay is over packet-switched network.

The next question is how to ensure that each transmission of a video frame will reach its destination with no loss and with minimum delay bound. Since video frames are captured periodically, in order to minimize the delay bound, *periodic resource allocation* with *periodic transmission synchronized with their capture* are required. TDP is the only known way to satisfy those requirements, while guaranteeing no loss with minimum delay bound, as shown in Figure 8.

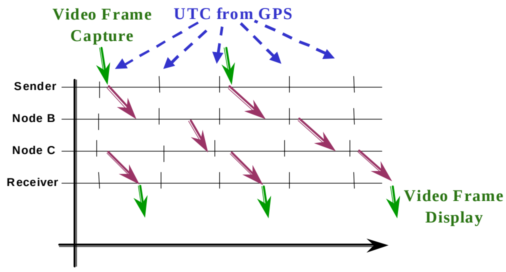

Figure 8: Periodic Capture, Transmission and Display of Video Frames

It is worth noting that even though all the video frames are not encoded with exactly the same amount of bits, the capacity reserved on the links is not wasted since it used to forward “best effort” (i.e., non-reserved) traffic. No loss due to congestion is guaranteed to all the video frames, provided that the amount of bits encoding them does not exceed the reservation.

4.1 Complex Periodicity: MPEG Video

Some video encoding schemes, like MPEG, encode frames with significantly different amounts of bits in a *periodic* fashion. MPEG encodes pictures in one of two different ways[2]:

Intra-frame Coding eliminates spatial redundancy inside pictures and the resulting encoded picture is called *I-frame*.

Predictive Coding eliminates temporal redundancy between a picture and the previous one through motion estimation. The obtained encoded picture is called *P-frame* and it is typically from 2 to 4 times smaller than an I-frame. The more similar two subsequent pictures, the smaller the amount of bits produced for each P-frame. Subsequent pictures are similar if the scene is slow moving, thus not changing much from a video frame period to the other. In summary, predictive coding delivers more compression on slow scenes, such as those captured in videoconferences.

It may be inefficient to transfer such a compressed video stream over a constant bit rate channel, e.g., the one provided by a circuit switched network (see [4] for a detailed discussion). If the encoder is operated in such a way that it produces a constant bit rate flow, it can introduce a delay up to the time between two successive I-frames; such a delay, which can be on the order of 500 ms, is obviously unacceptable for interactive applications.

Complex periodicity scheduling allows MPEG video frames to be transmitted as soon as they are encoded, analogously to what is described in Section 4 for fixed size video frames. TDP together with global time facilitates the realization of complex periodicity scheduling, which provides deterministic quality of service guarantees to variable bit rate traffic. In complex periodicity scheduling the amount of transmission capacity reserved on the links traversed by a session varies in a repetitive manner. Thus, with complex periodicity scheduling an MPEG video stream can be transmitted as shown in Figure 9.

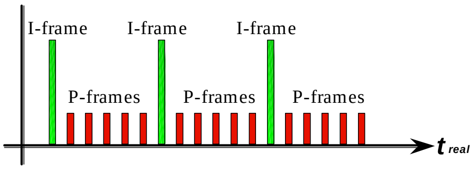

Figure 9: Complex Periodicity Scheduling of MPEG Video Stream

Thus, TDP with complex periodicity scheduling enables transmission of MPEG encoded video without the need of introducing a shaping delay in the encoder, with effective utilization of reserved network resources, with network delay basically equal to the propagation delay, with virtually no jitter, and without loss due to network congestion for all the video frames, provided that their size does not exceed the

---

2 Actually, a third type of encoding, called bi-directional predictive coding exists. Before a picture can be coded, a reference subsequent picture must be captured and coded. This introduces a delay of a multiple video frame periods that is not acceptable given the 100 ms end-to-end delay bound requirement. Thus, this type of compression is not considered here.

allocation in the corresponding TF. In a related study [5] it was shown that an MPEG encoder can be successfully implemented in a way that encoded video frame size never violates the resource allocation.

5 Scalability of Synchronous Switches

The deployment of global time to control packet forwarding has a twofold impact on switch scalability.

TDP allows controlling traffic patterns across each switch since it bounds the maximum number of packets to be moved during each TF to the same output from every input. This can be leveraged in the switch design: optimal input-output switching is obtained with low (x2) speed up in the switching fabric, or even without speed up at all. Instead, asynchronous switches require high speed up in order to achieve high throughput. Thus, given the state of the art aggregate switching fabric capacity, synchronous switches can accommodate higher capacity inputs than asynchronous switches. In other words, since the throughput of a P port synchronous packet switch with no fabric speedup is roughly the same as an asynchronous packet switch with a speed up of P, the interfaces mounted by the former can be P times faster than the latter.

The ability to control traffic patterns across switches can be benefited even further. A non-blocking fabric allows any possible input-output connection *at any time*; a simpler fabric allows only a limited number of simultaneous input-output connections. When TDP switches are deployed, packet arrival can be controlled in a way that incompatible input-output connections are not required during the same TF, thus avoiding unfeasible switching configurations. In other words, thanks to the increased flexibility introduced by the time dimension, synchronous packet switches with blocking fabrics can achieve the same throughput as asynchronous ones with non-blocking fabrics. For example, a non-blocking PxP crossbar requires P^2 crosspoints, while a self routing blocking fabric requires only $2P\log_2 P$ crosspoints. As a consequence, by using the latter fabric, a synchronous switch can accommodate as much as $\frac{1}{2}P/\log_2 P$ more ports than an asynchronous one that uses a crossbar.

5.1 CTR/UTC accuracy

The requirement of CTR accuracy, and hence UTC accuracy, has a direct impact on cost, stability, and implementation complexity. With a time frame delimiter the UTC accuracy requirement is $\underline{\boldsymbol{\frac{1}{2}\cdot T_f}}$ (i.e., UTC±1/2·(12.5μs to 125μs)). The reason for such a relaxed requirement is that the UTC is not used for detecting the time frame boundaries, as they are detected by the delimiters (e.g., unused code-words in the serial bit-stream). Consequently, the only function of UTC is enabling the correct mapping of the incoming time frames from the input channel to the CTR time frames. It is easy to show that up to $\underline{\boldsymbol{\frac{1}{2}\cdot T_f}}$ timing error can be tolerated while maintaining the correct mapping of time frames. (Today, a time card with 1 pps (pulse per second) UTC with accuracy of 10-20 ns is available from multiple vendors. The card is small and costs $100-200.)

6 Conclusions

This work shows how global time can be used to minimize the end-to-end delay for applications that require at the receiver continuous playing of samples captured at the sender. Specifically, global time eliminates the resynchronization delay (see Figure 1). Deployment of global time as a common time reference among switches to implement time-driven priority (TDP) forwarding of packets enables the provision of a service with the following characteristics:

- Deterministic absence of loss;
- Quality independent of the connection rate;
- Per switch delay of two time frames;
- End-to-end jitter of one time frame.

The service with such characteristics can be provided to any application generating one of the following types of traffic:

- Constant bit rate (e.g., voice telephony),
- Variable bit rate with periodic burstiness (e.g., videotelephony),
- Variable bit rate with complex periodicity (e.g., MPEG).

When a TDP network is deployed to carry voice calls, compression can be fully benefited to reduce the amount of link capacity used by each call. This is not the case with other asynchronous queuing schemes that possibly require overallocation to satisfy end-to-end delay requirements.

When dealing with videotelephony, encoded video frames are transmitted in bursts of packets with controlled delay and no loss. The delay perceived by users is lower than the one obtained by carrying video calls over a circuit switching network which requires delay to be introduced for smoothing out the burstiness of the video source.

Global time and TDP forwarding offer the only solution for the transmission of video frames also when they are encoded with a highly variable amount of bits, such as with MPEG. Each video frame can be transmitted in a burst of packets as soon as it is encoded with no shaping delay and no loss due to congestion. Through complex periodicity scheduling, resource reservation is fitted to the size of encoded video frames thus leading to efficient resource utilization.

References

[1] C-S. Li, Y. Ofek, A. Segall and K. Sohraby, “Pseudo-Isochronous Cell Switching in ATM Networks,” *Computer Networks and ISDN Systems*, No. 30, 1998.

[2] C-S Li, Y. Ofek and M. Yung, “Time-driven Priority Flow Control for Real-time Heterogeneous Internetworking,” *IEEE INFOCOM'96*, Apr. 1996.

[3] M. Baldi and F. Risso, "Efficiency of Packet Voice with Deterministic Delay," *IEEE Communications*, Vol. 38, No. 5, May 2000.

[4] M. Baldi and Y. Ofek, “End-to-end Delay Analysis of Videoconferencing over Packet Switched Networks,” *IEEE/ACM Transactions on Networking*, Aug. 2000.

[5] M. Baldi, Y. Ofek, “End-to-end Delay of Videoconferencing over Packet Switched Networks,” RC 20669 (91480), IBM - T. J. Watson Research Center, Yorktown Heights, NY, USA, Dec. 1996.

[6] R. Braden, D. Clark, and S. Shenker, “Integrated Service in the Internet Architecture: an Overview,” RFC 1633, Jul. 1994.

[7] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, “An Architecture for Differentiated Services,” RFC 2475, Dec. 1998.

[8] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications,” RFC 1889, Jan. 1996.