Proceedings of the 12th IEEE International Packet Video Workshop 2002
Pittsburg, PA, April 2002, http://www.pv2002.org (unformatted version)

SYMBIOTIC VIDEO STREAMING BY TRANSPORT FEEDBACK BASED QUALITY-RATE SELECTION

Javed I. Khan, Qiong Gu and Raid Zaghal
Networking and Media Communications Research Laboratories
Department of Computer Science, Kent State University
233 MSB, Kent, OH 44242
javed|qgu@kent.edu

Abstract

In this paper we present a congestion response mechanism designed for time-sensitive traffic based on the principle of direct protocol interactivity. We envision a transport mechanism, which is interactive and can provide event notification to the subscriber of its communication service. We then show a friendly adaptive MPEG-2 video transcoding scheme, which directly interacts with the transport protocol and adjusts its production with the events in the transport layer. In this paper we present the application side symbiotic mechanics, and report potential dramatic improvement in time-bounded video delivery.

Key Words: netcentric applications, TCP interactive, transcoding, MPEG2, temporal QoS.

1. Introduction

Congestion is one of the most actively researched areas in networking. However, the mainstream schemes focus on adjusting the delay-bandwidth product of communication and they work fully inside network. Application packets are delayed either in routers or at the network entry-point to cope with occasional congestions. For example, form the point of view of applications a TCP windowing mechanism acts as a network gatekeeper [BrOP94, Jaco88, AlPa99, Tene96]. It eventually performs some form of traffic shaping and introduces time distortion. Such distortion in temporal dimension is considered to be harmless to normal traffic. However, this is not always the case with time sensitive traffic.

Indeed, for time sensitive traffic a congestion control scheme based on delaying traffic in many cases may mean a mere shift in the point of packet discard.

The situation of a time sensitive application packet in need of transport can be clarified with an analogy to a patient in need of an ambulance. Instead, of being dropped inside the network, in the classical TCP scheme packets are waited at the entry buffer, when the link is congested. A time sensitive packet (such as an audio or video) is often in effect rendered useless at the source. It resembles a situation where the paramedics draw satisfaction from the fact that the patient is not dying in their ambulance, although the packet dies right at the TCP entry buffer waiting for the transport. TCP window buffer spreads a backlog in time. To make the matter worse, the ambulance however returns at some later point in time and picks up the delayed traffic. Effectively this is non usable from application point of view. For time sensitive communication, it not only spells doom for the current data but for packets those follow. Clearly, one of the critical problems in provisioning an integrated solution is that in the current arrangement the applications are not at all being notified of the congestion or of any other network impairment. Rather applications are put to sleep by the network/operating system process. Many of the time sensitive video system therefore avoid TCP and prefer to use raw UDP. Unfortunately, the random packet loss in UDP under congestion can create equally adverse effect[1]. Experiment has shown that if about 10-20% of the UDP packets are randomly lost at congestion, then most video streams become effectively unusable. Because in reality all real video transport packets contain many important header fields with deep inter packet dependency. It hardly improves the situation for an application if it is lost inside network or at the TCP/UDP sending buffer. Any congestion control scheme based on the principle of time distortion can result in similar problem.

In this paper, we show a new friendly adaptive MPEG-2 video transcoding scheme which is congestion adaptive. The interesting aspect of the scheme is that it directly interacts with the transport protocol and adjusts its production in synch with the impairment events in the transport layer. This feedback allows the system to trade-off spatial quality of video with temporal quality. Since, this is done in the application level with deep application level knowledge the results are much more tolerable from the final application performance point of view. In this paper we present the application side symbiotic mechanics, and report potential dramatic improvement in time-bounded video delivery. The overall scheme is network wise very simple and yet effective. The effectiveness is derived from the clever synchronization of the multimedia rate control mechanism of MPEG-2.

1.1. Related Works

Congestion control for time-sensitive traffic is a difficult problem. Most of the classical strategies are based on delaying traffic at various network points. The schemes vary from simple packet dropping in network, to admission control (delaying at network egress points), to graceful delaying by prioritization. For last few years it has been felt that applications have to be more integrated in the solution. Particularly promising are the research in the new TCP friendly paradigm [KeWi00, ReHE00, SiWo98, PrCN00]. [SiWo98] presented a TCP rate-based pacing mechanism that particularly takes note of document transfer characteristics. [ReHE00] discussed a general framework where applications can control rates based on their end-to-end measurements (similar end-to-end technique is used in RealPlayer). There are also fully application level proposals. Due to the lack of convenient means to obtain network states several works suggested [BrGM99, Wolf97] sending multilevel redundant information for video. Also several other works investigated combining application specific information from several streams into one clearinghouse architectures for aggregated congestion control. For example, recently proposed Congestion Manager [ABCS00, BaRS99] is a system layer component. It provisions aggregated congestion control when multiple streams from the same end-point attempt to send via a separate program called Congestion Manager. [SiWo98] proposed building TCP friendly application where application relies on real-time transport protocol (RTP) mediated end-to-end measurement. [PrCN00] used multiple probing mechanics for aggregate congestion control.

While there has been several promising work on network or system level issues to increase TCP friendly-ness, relatively very few work exists that seriously looked into the corresponding issues that arise in an actual time-sensitive application while taking advantage of the suggested 'friendliness'. Notably, the paradigm fundamentally shifts a major part of the congestion management responsibility to the applications. Time sensitive applications themselves have substantial complexity in adapting. Rate adaptation for any advanced application in general is quite complex. It requires sophisticated layer 4+ techniques. Unlike the network layer only paradigm of congestion management, a key research problem in this paradigm is the design of such rate adaptation techniques.

In this paper, we focus precisely here and present the symbiosis mechanics of a MPEG-2

[1] At the first glance, it may seem that UDP is immune. TCP and UDP grams both are treated essentially equally by IP routers and suffer almost same fate at the hands of the time-distortion based congestion control schemes, except that the UDP does not develop backlog due to retransmission. But, it develops backlog due to over production.

ISO-13818-2 [ISO96] video streaming system [KYGP01, KhYa01, KhGu01]. We demonstrate

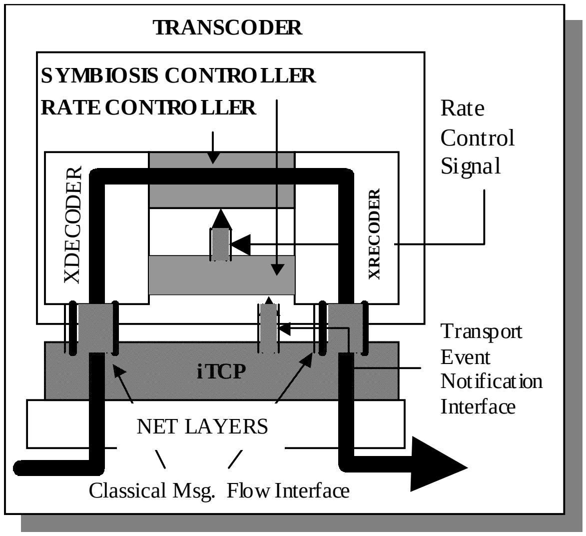

Fig-1 Interactive transport and symbiotic transcoder

a full application domain congestive rate adaptation and the interface mechanics to an interactive transport layer. We then share some interesting performance results.

The general principle we follow is simple and intuitive. It seems an effective delay conformant solution for time sensitive traffic may be built if the original data volume can be reduced by its originator-- the application. The particular scheme we propose here has several novel aspects compared to other recent works. First it depends on an active and direct notification mechanism by the underlying transport protocol, rather than indirect end-to-end feedback. If there is any congestion, we propose an interactive transport protocol, which can directly notify the application.

Secondly, we have designed a transcoder system rather than an encoder. This transcoder actively participates in a lazy symbiotic *exponential-back-off and additive-increase* like scheme [PeDa00]. (This is also one of the first to our knowledge). The advantage of this design is that it isolates the video server operation from the congestion management. It has been designed to sit either at the transport entry-point and perform conventional end-to-end paradigm based video transport like a conventional encoder. Or, it can also sit inside a network using technology such as a proxy for targeted and localized congestion management. The transcoding mechanism observes the local transport layer characteristics and can accordingly adjust the outgoing MPEG-2 stream bit-rate. Such a configuration can facilitate video communication between network segments with widely different bandwidth. However, a transcoding system is computationally more challenging than a conventional encoder. It is required to match the frame rate, and it should be much faster than typical encoding. The advantage is that it subsumes the functionality of encoder based system.

In this paper, in the next section, we first provide the system overview. In section 3 we then present the symbiotic rate control mechanism-- the key application component that provides the key network aware solution. The model has been developed by closely following the MPEG-2 Test Model 5 (TM5). MPEG-2 TM-5 signifies a real video coder with substantial complexity of itself. While the detail can be found in [Mpeg00], in this paper we describe the salient part of the rate control architecture that is critical to this symbiosis. Finally, in section 4 we share performance of the scheme. The results to be presented has been obtained using a real implementation of the symbiotic Transcoder [KYGP01, KhYa01], and letting it run on a simulated version of the proposed TCP interactive.

2. Rate Adaptive Transcoder

2.1. System Configuration:

We have developed a three-part system model-- *server, transcoder* and *the client*. The middle component *transcoder* [KPOY01, KHFH96] can be placed in a suitable network junction point, which intercepts the stream. This is slightly different from encoder-decoder *server-client* system model. This approach has several advantages as opposed to implementing the rate adaptation at the end-point (encoder). It subsumes the functionalities of server-client model. In addition, it allows rate adaptation on video stream that is already encoded and thus enables serving stored video at a dynamically selected rate. This decoupling also has the benefit that the transcoder can be made to auto

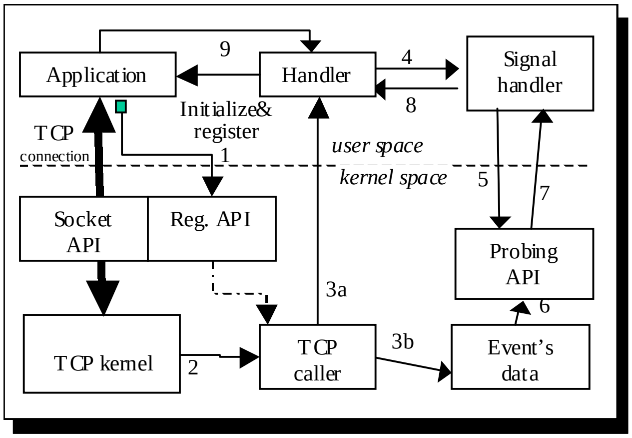

Fig-2 the TCP interactive extension. The added registration API allows demanding applications to subscribe to events and probe additional event data.

sense local asymmetry in link capacities and can be dynamically deployed inside network for streaming. For example it can sit at a node splicing a fiber and a wireless network, and thus can downscale an incoming high-bandwidth video multicast stream for an outgoing low-capacity wireless links. Additionally, there also exists the possibility of bring down transcoding operation inside network by emerging technologies such as active networking [TSSW97, GuTe98]. The transcoder sits on top of the interactive transport control layer-- TCP Interactive. Fig-1 explains the system arrangement.

2.2. Transport Control:

Unlike conventional TCP, this interactive transport layer, when there is an internal timer-out event, passes on the current window resize event to the subscriber layer. The interface is almost identical to the TCP classic, except, upon opening the socket, the application binds an interrupt handler routine to the designated socket event. When, the event occurs the TCP triggers the handler. The binding is optional. If application chooses not to bind any handler the system defaults to the silent mode identical to TCP classic.

The internal architecture of the TCP interactive is shown in Fig-2. The added registration API helps applications to subscribe to TCP events, in this case the timer out. We have added a simple extension to TCP kernel. The main unit is called TCP Caller unit. It is activated if an application subscribes. It keeps track of the TCP timeout event. More inquisitive applications can also probe into selected TCP states. When a timeout event is detected the kernel initiates 2. 3A via TCP caller then invokes the handler. Optionally, applications can probe additional event data via signal handler/ and additional API (4,5,6,7,8,9). If event data is subscribed then 3B occurs concurrently with 3a.

2.3. Transcoder Architecture:

The transcoder unit has a decoder, and a re-encoder[2]. The re-encoder has a feedback rate control mechanism, which is capable of working in two modes: *normal* mode and *frugal* mode. In frugal mode the rate can be controlled at frame level. The actual control signal to the rate controller is generated by an application unit called *symbiosis controller*. The symbiosis controller accepts input signal from the transport layer to realize the symbiosis. Below we describe the MPEG-2 transcoder rate and symbiosis control mechanisms that we have developed for this experiment.

3. Rate Control Mechanism

The rate control mechanism is illustrated in Fig-3. The complexity of the system arises from several reasons. Due to the *variable length coding* (VLC), it is not possible to predict the exact amount of bits that will be produced from a macro-block for a given choice of coding parameters. Secondly, the perceptual content and activity in a particular picture area also dictates the inherent amount of bits that may be required to encode it. Also the bit requirements per macro-block depends on the picture type (I, B or P) and other subjective factors. The proposed mechanism is also a double-loop feedback control mechanism where the output bit-rate is continually sensed to stabilize at piecewise constant rate, with dynamic allowance for variations in the frame/picture type, similar

[2] A number of recent techniques (including ours) have been identified for accelerated fast full logic MPEG-2 transcoding significantly under cutting the cost [KPOY01, KHY01].

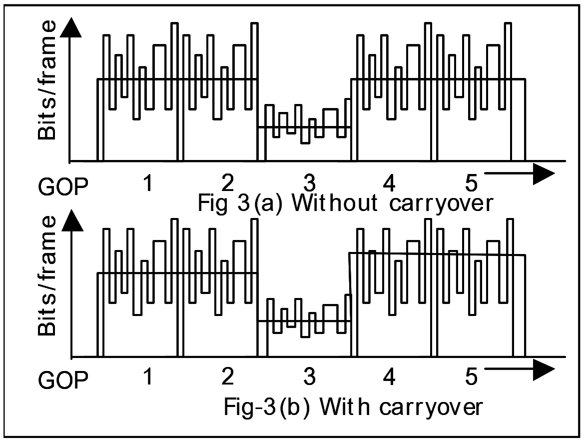

Fig 3(a) Without carryover

Fig-3(b) With carryover

to TM-5. A second internal feedback loop further tracks the efficacy of key conversion factors/constants for additional stability.

The output bit-rate is controlled by the quantization-step. After motion estimation and compensation, the prediction errors for each 8x8 blocks are computed. These 64 pixel differences are then transformed into 64 DCT coefficients. Each of the DCT coefficients is however, quantized using a separate step, because the human visual system responds differently to distortion in various DCT coefficients. However, to control the overall output bit rate MPEG-2 in its linear quantization mode uses a scale factor called *mquant* to determine the actual quantization steps, which are applied on these DCT coefficients. As *mquant* increases, the effective quantization steps become larger, more information is lost, encoding requires lower bits, and also the quality of the picture degrades, and vice verse. To account for few of these factors, in the topmost level the value of *mquant* for each macroblock is calculated as a product of two primary factors (a) the *buffer fullness* and (b) the *macroblock activity*. The *mquant* for the jth frame is computed as a product of two parameters: $mquant_j = Q_j \times N\_act_j$. The final value of $mquant_j$ is coded either in the slice or in the macroblock header [ISO96]. The part that is relevant[3] for this experiment is the Q_j. It is a modulation parameter, that determines how the allocation of frame-bits itself is varied.

[3] The motivation behind the *activity factor* is that human visual perception is less sensitive to distortions in noisier textured areas and more sensitive to distortion in image areas with uniform texture. We used a slightly modified region based activity assignment algorithm.

3.1. Feedback Quantization Mechanism:

The system has two modes of operation: *normal* mode and *frugal* mode. In normal mode, the objective of feedback system is to maintain the output bit rate at piece-wise per GOP (group-of-picture). In frugal mode, it moves into a variable-rate encoding mode with proper proportioning for frame types, and the macro-block activity however, without any carryover. The saving earned during the frugal mode, however, is stored and can be (optionally) carried over to the point where normal mode is resumed to attain overall target rate. The control mechanism maintains three virtual buffers for separately tracking the bits consumed by the I, B, and P frames. To encode a frame of type x, for each macroblock, first a quantity called buffer fullness d_j^x of its corresponding buffer is determined. This is then used to determine the modulation factor Q_j.

$$Q_j = \left[\frac{31 \times e_j^x}{r}\right] \text{ where,} \qquad \ldots(1)$$

$$r = \left\lfloor \frac{2 \times c(t)}{frame\_rate} + 0.5 \right\rfloor$$

Here, r is called *reaction parameter* and is estimated from the current overall bit rate goal c(t). The quantity e_j^x is the *effective buffer fullness* and is computed from *virtual buffer fullness* d_j^x. The notation refers to the jth macroblock inside x type frame. These quantities are determined as following:

$$e_j^x = d_j^x - d_0^x \cdot S(t) \text{, and} \qquad \ldots(2)$$

$$d_j^x = d_0^x + B_{j-1} - \frac{(j-1) \cdot T^x(t)}{mb\_count}$$

In normal mode the *effective buffer fullness* is given by the *virtual buffer fullness*, but during frugal mode it is decoupled from initial buffer fullness, and is only estimated based on the frugal state target bit rate. A value of 1 to the state function *S(t)* moves the system to the frugal state, and zero to normal state. In the frugal mode, the bit generation temporarily reduces. However, the virtual buffer fullness quantity is continually updated. This enables the carryover of the savings made during frugal mode operation when the system returns to normal mode.

3.2. Buffer Fullness Estimation and Carryover:

Virtual buffer fullness is determined from three quantities: (i) the number of bits generated so far by encoding previous j-1 macroblocks inside this frame (B_{j-1}), (ii) the initial fullness of buffer before beginning the encoding of this frame (d_j^0), and (iii) the target bits allocated to this frame (T^x). The initial values for the buffer fullness are computed at the beginning of encoding a frame. For the encoding of first frame of a GOP these are given by:

$$d_0^I = 10 \times \frac{r}{31}, \quad d_0^P = k_P \cdot d_0^I, \quad and \quad d_0^B = k_B \cdot d_0^I.$$

Here k_B and k_P are universal constants and depend on the quantization matrices. For standard MPEG-2 quantization matrix their values are k_P =1.0 and k_B =1.4. For subsequent frames the final fullness of the previous frame is passed on as the initial fullness of the next frame buffers. During the encoding of a frame for each macro-block the actual amount of bits produced is measured immediately after it's encoding. Thus, Once DCT is done, all subsequent coding of the current macroblock including VLC have to be completed before the next macroblock can be quantized.

3.3. Target Rate Proportioning:

To calculate the target bit for each frame, at the beginning of each GOP, first a rough allocation for the entire GOP is estimated. This is estimated from the target stream bit rate, frame rate and the total number of frames in the GOP. Each GOP initially has one I and n_B and n_P B, and P frames respectively.

$$R_{GOP} = \left\lfloor \frac{(1 + n_{P-remaining} + n_{B-remaining}) \times c(t)}{frame\_rate} + 0.5 \right\rfloor \qquad \text{…(3)}$$

To account for the variations in the frame types complexities, a TM-5 like adjustment is made. This is performed with the quantities called *global complexity measures* [X_I: X_P: X_B]. These are computed by averaging the actual quantization values used during the encoding of all the macroblocks including the skipped ones) and the actual number of bits generated S_X, where $X_X = S_X \cdot Q_X$. These averages are maintained for each frame type (x=I, P, and B) and updated at the end of each frame encoding. Finally, the actual target bit rate for each frame type is computed using the following usual TM-5 models (where k's are various defined constants):

$$T^I(t) = \left\lfloor \frac{R(t)}{1 + \frac{n_P \cdot X_P}{k_P \cdot X_I} + \frac{n_B \cdot X_B}{k_B \cdot X_I}} + 0.5 \right\rfloor \qquad \text{…(4a)}$$

$$T^P(t) = \left\lfloor \frac{R(t)}{n_P + \frac{n_B \cdot k_P \cdot X_B}{k_B \cdot X_P}} + 0.5 \right\rfloor \qquad \text{…(4b)}$$

$$T^B(t) = \left\lfloor \frac{R(t)}{n_B + \frac{n_P \cdot k_B \cdot X_P}{k_P \cdot X_B}} + 0.5 \right\rfloor \qquad \text{…(4c)}$$

Once each frame is encoded the bits used is measured and the encoded frame is subtracted from the initial GOP size ($R_{new} = R - S_X$) to estimate the remaining available bits. Also, the number of frames n_B or n_P gradually decreases. The target size for subsequent frames in the GOP, which are either type P or B, are estimated from the remaining bits R, and the remaining number of frames. Finally $Q_j = \lfloor 31 \times d_j^x \cdot r^{-1} \rfloor$ is computed by dividing the buffer-fullness by the TM-5 *reaction parameter*. When the system is in normal mode the rate control mechanism does not need to sense the target bit rate at every frame. However, when it moves into frugal mode it senses the current target-rate per frame. Fig-3 (a) and (b) shows the rate control mechanism without and with frugal saving carryovers.

3.4. Symbiotic Rate Determination:

The transcoder only focuses on the rate adaptation. However, the actual values of the rate dynamics are controlled by a separate mechanism called *symbiosis controller*.

In this lazy binary-back-off symbiotic model, the control parameter of the rate controller *target bit-rate* c(t) is determined by a two variable min/max mechanism. The idea is to closely mimic the rate provided by the underlying transport layer, however, it is done in a way that safeguards the applications requirements. In this experiment we have designed a symbiosis, which responds to a timeout event. Let the target bit rate during normal mode generation is given by C_{max}. When, a time-out event occurs in the channel (designated by an event variable $\xi = 1$), we let the subscriber rate retract to a smaller but yet non zero quantity. We define this point by the parameter called *rate retraction ratio* ρ. The idea is that based on the specific video instance and a tolerance level on its quality the system should still be able to generate video however, with lesser visual quality based on precise quality/ delay tradeoff boundaries of the video. Based on the tolerance we define a ratio called *rate retraction ratio:*

$$\rho = \frac{C_{min}}{C_{max}}.$$

For symbiosis with the underlying TCP, we define a running generation threshold function as following:

$$c_T(t) = \frac{1}{2} c(t-1) \; when \, \xi = 1 \qquad \ldots(5)$$
$$= c_T(t-1) \; otherwise$$

It retracts to half its current size when fault occurs. The running control function c(t) is then given by:

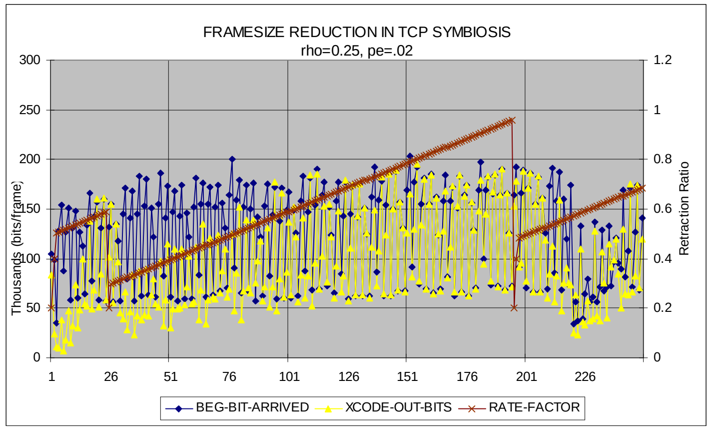

Fig-4 The binary-back off rate reduction. There are three timeout events here. The figure plots the incoming frame sizes, the target rate (retraction ratio) specified by the symbiosis unit, and the resulting output from the transcoder.

$$c(t) = \rho \cdot c_{max} \; when \, \xi = 1 \qquad \ldots(6)$$
$$= 2 \cdot c(t-1) \; when \, c(t) \geq \frac{1}{2} c_T(t-1)$$
$$= \min[C_{max}, x(t-1)+1] \; when \, c(t-1) \geq c_T(t-1)$$

The control function performs lazy *binary-exponential-backoff* and *additive increase* within the limits given by generation parameters ρ and normal mode target bitrate C_{max}. The system enters the frugal state $S(t) = 1$, when then loss event occurs (i.e. $\xi = 1$), and stays in the frugal state until the control (target bit-rate) recovers to the normal target bit-rate.

4. Experiment Results

We have implemented an MPEG-2 DTV symbiotic video rate transcoder that uses the above model. This experiment describes the performance for the case of a MPEG-2 ISO/IEC 13818-2 broadcast DTV (704x480) resolution video encoded with base frame rate of 4 Mbps at main level@main profile on this symbiotic transcoder. We have chosen 704x480 resolution video (typical broadcast quality video) for this experiment. It is much higher resolution that the QSIF or SIF video typical in most contemporary internet applications.

Setup: For these set of experiments we simulated the transport subsystem to operate both in the classical transport mode (labeled as TCP) and in interactive transport mode (iTCP).

We let the video generator (transcoder) feed into the video stream. The underlying channel time-out was induced artificially with a uniform random distribution of error. We further assumed that the resulting packet loss event is independent of the video stream size (accounting for uncorrelated congestion deep inside the network). In the classical mode, we switched off the improvements and let the transcoder operate in conventional error unaware mode. The transcoder generated the video using the conventional TM-5 [MPEG00] rate control at the target rate of 4 Mbps. Transport protocol buffered the generated data while the transport layer exercised binary back-off and additive recovery at time-out events. In the interactive mode, we switched on the interactive mechanism in the transport layer and the symbiosis mechanism of the transcoder. The transcoder according to the symbiosis controller varied the video rate for interactive TCP.

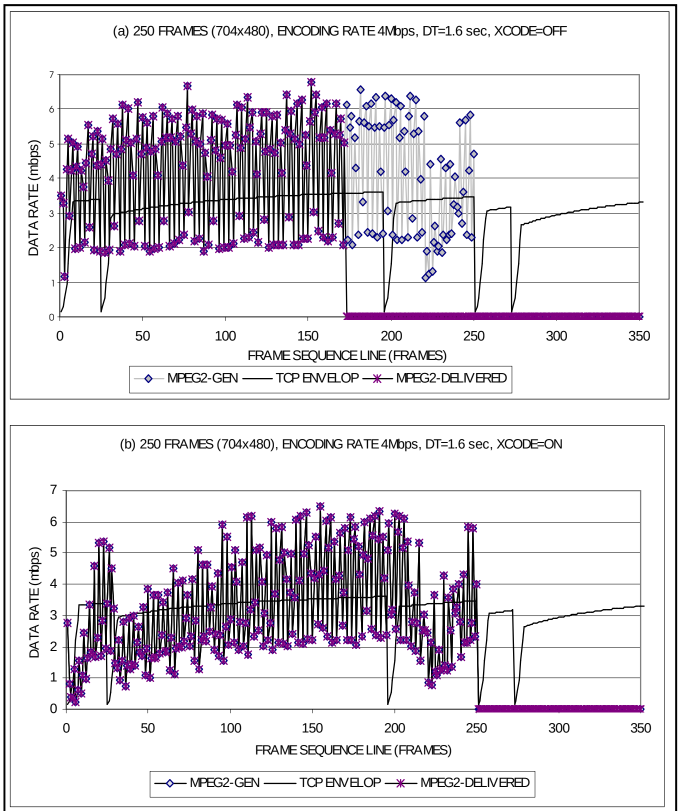

Fig-5 MPEG-2 level performance. (a) The MPEG-2 frames delivered and received in time in TCP, and (b) in iTCP scheme.

The video data was received into an analyzer. The transcoder and the analyzer both recorded the entry and delivery time of each frame date as they were transported according to their coding sequence. A frame is considered 'failed' if its delivery time exceeds a given *discard threshold* (d).

Symbiotic Rate Control: Figure-4 shows the symbiotic frame rate transcoding that occurred due to the joint rate specification at the rate control logic at the symbiosis unit and in the transcoder. It plots the incoming video frame sizes, the target *rate retraction ratio* specified by the symbiosis controller, and the resulting outgoing frame rate generated by the transcoder. The timer out events (in this case there are three) at the TCP resulted in the symbiosis unit to modify the rate according to the *lazy-binary-back-off* rule. A retraction ratio of 0.25 was used. Though, the final generation rate varied widely from frame to frame due to their frame type), but the general trend followed the specified target.

MPEG-2 Frame Transport Efficiency: Now we show the impact of TCP interactivity. In the first experiment, we took frame wise detail event trace of what happens to the first 250 of the frames of this video at both sending and receiving ends. For a given discard threshold d we also traced which frame was successfully received or not at the receiving end of the MPEG-2 player. For comparison we traced both the transport unaware (Fig-5(a)) as well as transport aware operations (Fig-5(b)). In both the figures the x-axis plots the time line in terms of video frame (coding) sequence. At 30 fps rate each frame is approximately spaced 33 ms apart. The first two curves on this plot respectively show (i) the generation function and (ii) the dynamic transport window envelop at the sending end. The third curve shows the frames those were actually delivered to the player with the discard threshold d=1.6 second. The variation in the generation rate depicts the typical nature of video encoding. Due to the frame type difference the allocated bit-rates generally varies sharply between consecutive frames based on their frame type (I, B, or P). Typically I frames require most bits, and B frames the least. An I frame can be 2-3 times larger than B frames in the sequence. As evident, the window went through few back-off and additive increase phases during the shown interval. However, in unaware mode, this back-off resulted in buildup and buffer delay. Correspondingly severe loss started near the 170th frame, and then it could never recover (resulting in loss of all frames afterward). Fig-5(b) shows the corresponding plots for the transport aware mode. As evident, in this mode the transcoder being aware of the underlying window bandwidth, begun adjusting the generation function. As evident, it resulted in dramatic improvement where almost no frame suffered discard. The symbiotic rate control was evident in the low frequency varying nature of the generation function.

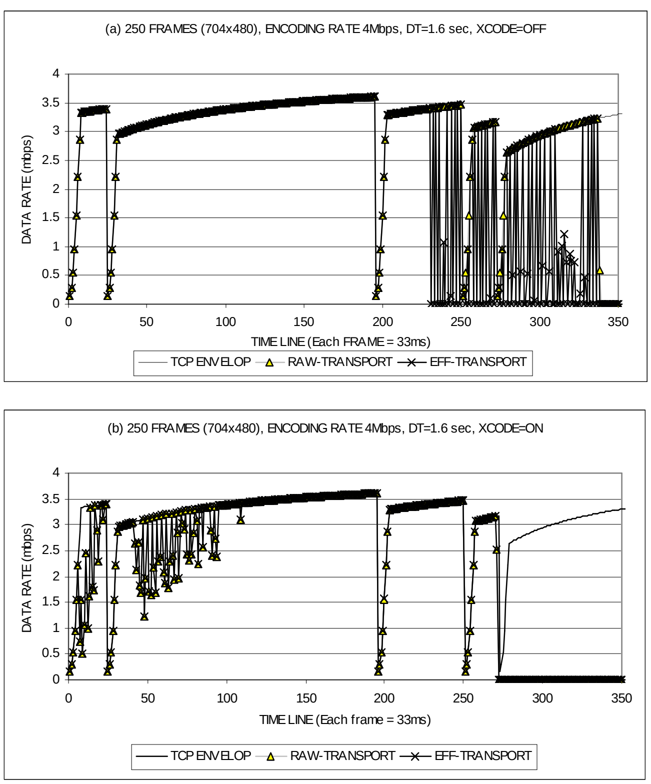

Fig-6 Transport level performance. (a) The MPEG-2 frames delivered and received in time in TCP and (b) in iTCP scheme.

Frame Discard vs. Delay: While the trace shows the general mechanism of the improvement, we were also curious to see how the rate of video frame discard would vary with various choices of the threshold. We were also curious to see how the discard rate would vary with the allocated transport bandwidth (sending window size).

Correspondingly, we varied the maximum transport bandwidth (maximum sending window) bandwidths from 100-110% over the coding bandwidth. Fig-7(a) plots the dramatic difference between the performances of the two mechanisms. It plots the number of failed frame (y-axis) with various discard thresholds (x-axis). The top three curves (P1, P2, P3) show the frame loss for the transport unaware channel for three path bandwidths (+0%, +5% and +10% over encoding target rate). As can be seen, even if the acceptable delay is set as high as 1.2 seconds, more than 50 (+20%) frames are lost for both P1 and P2. Curves P4, P5 and P6 respectively now show the improvement in performance from the iTCP integrated solution for the same three cases. Even at much smaller 0.7 seconds cutoff delay, complete recovery has been possible. No frame was lost. The trend is same even if more channel bandwidth is allocated channel bandwidth. When we allowed greater allowance between target generation rate (4Mbps in this case) and the allowable bandwidth respectively (+0%, +5% and +10%) it lowered the frame loss. This extra 5-10% bandwidth typically is not used in normal mode operation. However, after a back-off it allows for faster transmission of backlogged data resulting in lower delay.

Observation at Transport Layer: While, the above experiment logged the performance at MPEG-2 transport layer (frames exiting from MPEG-2 transcoder and entering at the decoder), we were also curious to see how the events look from the transport layer. For the same experiments shown above, we therefore also traced the events at the entry and exit of the TCP transport layer. Fig-6(a) and

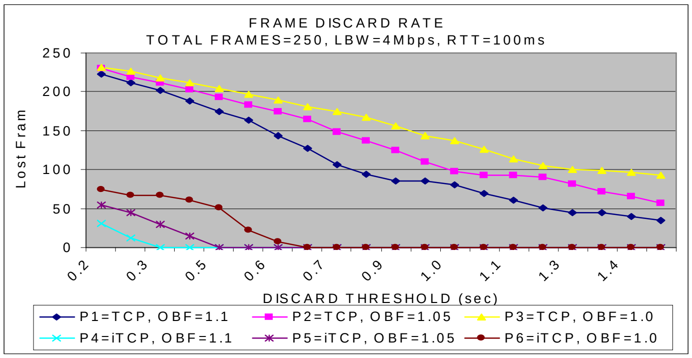

Fig-7(a) Frame loss due to delay.

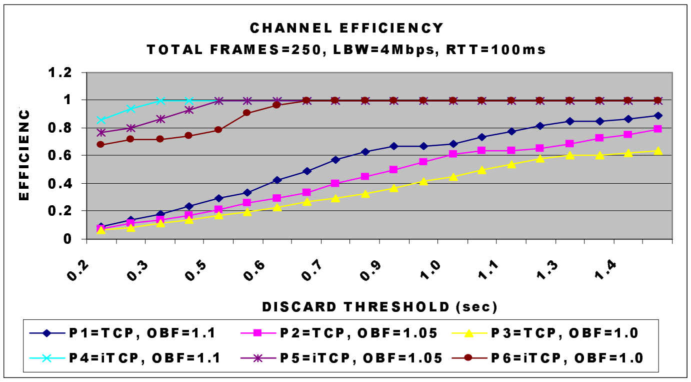

Fig-7(b) Effective channel efficiency in interactive TCP.

6(b) shows the transport layer trace. It first shows when and how much frame data begun their journey from the sending buffer (raw transport- shown by triangular data points). How much of these arrived safe to the application was a different matter. As the transport protocol was delaying delivery, therefore, large segment of this data become stale. We traced back the discarded frame data into the packets. The third curve (effective-transport—shown by cross hair data points) shows the part of data that actually was not stale and thus was actually was still useful to the application. As evident in the unaware mode (Fig-6(a)), initially packets were carrying valid data (overlapping triangles and crosses), but eventually they were not. Though the network was working in full strength (triangles) but the effective transport (crosses) was very poor. As evident, almost 30% of the data that network carried was effectively useless from the application point of view. Fig-6(b) now shows the transport events under the iTCP mode, as evident (by the coincidence of the triangles and crosshairs) this time the results were dramatically better. Correspondingly, a symbiotic mechanism can dramatically relieve the load of the network itself.

Without interactivity, a lower service level will always run into the risk of working hard unknowingly that it is doing useless work! To quantify this waste, we define the quantity *channel efficiency* as the ratio of the total bytes carried at transport layer to the application bytes delivered with service conformance. Fig-7(b) plots the channel efficiency for the same six cases (P1 to P6) of Fig-5. As evident, the channel efficiency of TCP is less than 30% below discard threshold .6 second. While it is over 80% for the proposed iTCP.

Observation at Application Level: In the above two experiments we illustrated how the symbiosis mechanism plays from the video transport protocol (MPEG-2) and the network transport protocol (TCP) layers beneath it. In this final plot we will illustrate how this mechanism appears from the very top-- in the application layer itself. An application receives and delivers uncompressed frames. The performance metric this end-system uses is the temporal and spatial quality difference between the transmitted and the reproduced uncompressed video frames at two ends. The underlying MPEG-2 transport protocol and the network layer TCP together provides the transport. The specific compression, windowing etc. are external techniques to the visual system.

In Fig-8(a) each of the frames are plotted as a point in the video quality/ frame delay plane. As can be seen from the region of the two QoS distributions, in classical TCP, although frames have been generated with SNR quality ranging between 55-38 dB, but many of these frames are lost in transport, and was never delivered. In contrast, the proposed TCP interactive can

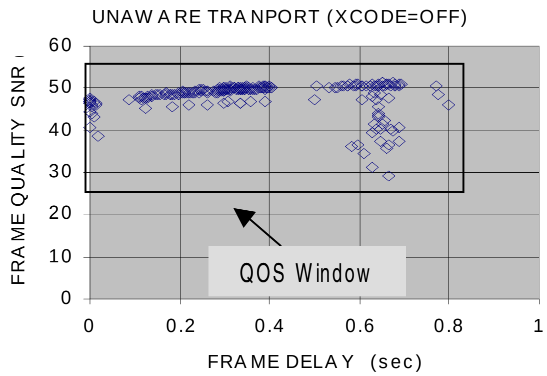

Fig-8(a) TCP Frame QoS performance

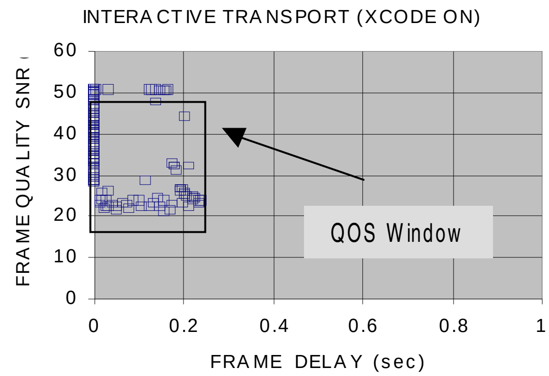

Fig-8(b) iTCP Frame QoS Performance

deliver all the frames with 0.6-0.7 sec delay guarantee at 55-20 dB quality[4].

Fundamentally, what **TCP interactive** has offered is a qualitatively (as opposed to the quantitative improvements offered by any unaware solution) new empowering mechanism, where the catastrophic frame delay can be traded off for acceptable reduction in SNR quality.

5. Conclusions and Current Work

In this paper, we have presented a case of rate symbiosis mechanism in line with current advances in TCP friendly systems. We have presented the case through a simple 'interactive' generalization of the classical transport control protocol, and a novel implementation of a symbiotic MPEG-2 transcoder. The proposed *principle of protocol interactivity* can enable fundamentally new solutions to many of today's hard to tackle problems. In this paper we have demonstrated the case of quality conformant congestion control for time-sensitive traffic.

The approach exposed the overall advantage of network 'friendly' applications. However, is also departs significantly from the mainstream TCP friendly systems those has been suggested recently in two senses. First, it does not consider adding new major component in network software structure. One of the principal strength of the proposed scheme is its relative simplicity at network layers –yet its effectiveness. It only expects some form of interactivity directly from the concerned network protocols as a general interface feature. Thus there is no expectation of or conflict with additional services (such as combined congestion control from multiple applications). Secondly, it is not dependent on indirect probing tools or separate new network utilities. Nor it excludes their use when available. Interestingly some of the information measured by the external tools suggested by other approaches might be already available (or are being estimated/tracked) at lower layers anyway. At least this is the case with TCP congestion. The suggested direct protocol interactivity thus seems to be the logical path that can avoid potential duplication of efforts.

Nevertheless, the approach will add lesser but yet some complexity in the network layer. The augmentation of the notification feature increases the normal mode delay of TCP even it is slight. The actual cost will depend on the intensity of coupling. Designer of application symbiosis unit must be aware of the potential cost of tight coupling between handler and caller. However, as shown by the results-- with a prudent design the impact on the network level transfer rate (based on low layer measurement), if any, can be widely surpassed by the gain made at application layer. However, an interesting aspect of this scheme is that a wrong design will only affect the application at fault and will have no effect on others. Also, notably, the entire scheme is less invasive than many other recent approaches in congestion management (such as ECN or RED, which require router intervention and/or IP layer intervention across network). However, the

[4] Interested viewer can retrieve both versions of the transported video from our website [KhGu01] for perceptual comparison.

proposed interactivity is not an alternate to these, rather is a complimentary scheme.

The work has been supported by the current DARPA Research Grant F30602-99-1-0515 under its Active Network initiative.

6. References

[ABCS00] D. Andersen, D. Bansal, D. Curtis, S. Seshan, and H. Balakrishnan, System Support for Bandwidth Management and Content Adaptation in Internet Applications, Proceedings of the Fourth Symposium on Operating Systems Design and Implementation, OSDI 2000,

October 23-25, 2000 San Diego, California.

[AlPa99] Allman, Paxson, et al. TCP Congestion Control, RFC 2581, April 1999.

[BaRS99] Balakrishnan, H., Rahul, H., and Seshan, S., An Integrated Congestion Management Architecture for Internet Hosts," Proc. ACM SIGCOMM, Cambridge, MA, September 1999.pp.175-187.

[BrGM99] Hector M. Briceño, Steven Gortler and Leonard McMillan, NAIVE--network aware Internet video encoding, Proceedings of the seventh ACM international conference on Multimedia, October 30 - November 5, 1999, Orlando, FL USA, Pp 251-260.

[BrOP94] Brakmo, L.S., O'Malley, S.W., and Peterson, L.L, TCP Vegas: New Technique for Cogestion Detection and Avoidance, Proc. SIGCOMM' 94 Conf. ACM, pp-24-35, 1994.

[GuTe98] Wetherall, Guttag, Tennenhouse, "ANTS: A Tool kit for Building and Dynamically Deploying Network Protocols", IEEE OPENARCH' 98, San Francisco, April 1998. Available at: http://www.tns.lcs.mit.edu/publications/openarch 98.html

[ISO96] Information Technology- Generic Coding of Moving Pictures and Associated Audio Information: Video, ISO/IEC International Standard 13818-2, June 1996.

[Jaco88] Jacobson, V, Congestion Avoidance and Control, Proc. SIGCOMM' 88, Conf, ACM, pp-214-329, 1988.

[KeWi00] Jun Ke and Carey Williamson / University of Saskatchewan, Towards a Rate-Based TCP Protocol for the Web, Proceedings of the 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2000.

[KhGu01] Javed I. Khan, Q. Gu, Network Aware Symbiotic Video Transcoding for Instream Rate Adaptation on Interactive Transport Control, IEEE International Symposium on Network Computing and Applications, IEEE NCA' 2001, October 8-10, 2001, Cambridge, MA, pp.201-213

[KHFH96] Keesman, Gertjan, Hellinghuizen, Robert Hoeksema, Fokke Heideman, Geert, Transcoding of MPEG bitstreams Signal Processing: Image Communication, Volume: 8, Issue: 6, pp. 481-500, September 1996,

[KYGP01] Javed I. Khan, Seung Su Yang, Qiong Gu, Darsan Patel, Patrick Mail, Oleg Komogortsev, Wansik Oh, and Zhong Guo Resource Adaptive Netcentric Systems: A case Study with SONET- a Self-Organizing Network Embedded Transcoder, Proceedings of the ACM Multimedia 2001, October 2001, Ottawa, Canada, pp617-620

[KhY01] Javed I. Khan & S. S. Yang, Resource Adaptive Nomadic Transcoding on Active Network, International Conference of Applied Informatics, AI 2001, February 19-22, 2001, Insbruck, Austria, [available at URL http://medianet.kent.edu/, also mirrored at http:// bristi.facnet.mcs.kent.edu/medianet] (accepted).

[KPOY01] Javed I. Khan, Darsan Patel, Wansik Oh, Seung-su Yang, Oleg Komogortsev, and Qiong Gu, Architectural Overview of Motion Vector Reuse Mechanism in MPEG-2 Transcoding, Technical Report TR2001-01-01, Kent State University, [available at URL http://medianet.kent. edu/ technicalreports.html, also mirrored at http:// bristi.facnet.mcs.kent.edu/medianet] January, 2001]

[Mpeg00] MPEG-2 Test Model 5, [URL: http://www.mpeg.org/MPEG/MSSG/tm5/, last retrieved December 25, 2000]

[PeDa00] 19 L. L. Peterson and B. S. Davie. Computer Networks, 2nd edition, Morgan-Kaufmann, 2000.

[PrCN00] Prashant Pradhan, Tzi-cker Chiueh and Anindya Neogi Aggregate TCP Congestion Control Using Multiple Network Probing, Proceedings of the The 20th International Conference on Distributed Computing Systems, ICDCS 2000. 2000.

[ReHE00] Reza Rejaie,Mark Handley, Deborah Estrin, Architectural Considerations for Playback of Quality Adaptive Video over the Internet, Proceedings of the IEEE International Conference on Networks (ICON' 00), 2000.

[SiWo98] Dorgham Sisalem, Adam Wolisz, Towards TCP-Friendly Adaptive Multimedia Applications Based on RTP, Proceedings of the The Fourth IEEE Symposium on Computers and Communications, 1998.

[Tene96] Tanenbaum, A. Computer Networks, 3rd edition, Prentice Hall, 1996.

[TSSW97] Tennenhouse, D. L., J. Smith, D. Sincoskie, D. Wetherall & G. Minden., "A Survey of Active Network Research", IEEE Communications Magazine, Vol. 35, No. 1, Jan 97, pp 80-86

[Wolf97] Bernd E. Wolfinger, On the potential of FEC algorithms in building fault-tolerant distributed applications to support high QoS video communications, Proceedings of the sixteenth annual ACM symposium on Principles of distributed computing , 1997, Pages 129 - 138