Accepted for publication in the
Proceedings of the IEEE International Symposium on Network Computing and Applications, 2001, October 2001, Cambridge, MA (unformatted version).

NETWORK AWARE SYMBIOTIC VIDEO TRANSCODING FOR IN STREAM RATE ADAPTATION ON INTERACTIVE TRANSPORT CONTROL

Javed I. Khan and Qiong Gu

Internetworking and Media Communications Research Laboratories
Department of Computer Science, Kent State University
233 MSB, Kent, OH 44242
javed@kent.edu
June, 2001

Abstract

In this paper we present a new approach of application integrated congestion management. In contrast to classical UDP or TCP, we present a transport mechanism, which is interactive and provides event notification to its subscriber-- and a network aware adaptive MPEG-2 video transcoding scheme, which adjusts its production in synch with the impairment events in the transport layer. We report a novel symbiotic mechanics, which provides dramatic improvement in time-bounded video delivery. The approach provides a new degree of freedom to network aware applications in quality of service conformant communication.

Key Words: netcentric applications, transport protocol, congestion control, quality-of-service.

1.Introduction

Congestion is one of the most actively researched areas in networking. Numerous ingenious solutions have been proposed so far. The end-to-end techniques include traffic shaping, timeout management in the TCP sliding window protocol (such self-clocking, slow start, fast retransmit and fast recovery, packet pairing, etc.). Many of the end-to-end mechanisms also have contributed to the improvement of TCPs (TCP Tahoe, Reno, Vegas, etc). Also, techniques have been proposed which takes active help from the network embedded components. These include congestion notification via mechanisms such as random early detection (RED) [FlJa93], or more explicit DECbit, or ECN [PeDa00, RaJa93]. Also, rigorous studies have been made in network resource management for congestion avoidance or to ease the impact, such as resource reservation, fair queuing [StZh97, Tenn96]. Most of current schemes adjust the delay-bandwidth product. In current network based application development paradigm, in response to congestion, the TCP window appears as a network gatekeeper and implements policies, which in the end translates to some form of traffic shaping [BrOP94, Jaco88, Tene96]. This adjustment is harmless for time insensitive traffic. However, it has been pointed out that this delay is inconvenient for time sensitive traffic. Indeed such buffering in many cases merely shifts the point of packet discard. Instead, of being dropped inside the network, in the classical TCP scheme packets are restricted at the entry points. A time sensitive packet (such as 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 which effectively 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 on video. 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 the video stream protocol contains many important header fields.

It seems a potential and more effective delay conformant solution for time sensitive traffic may be built if the original data volume can be reduced by its originator-- the application, rather than subjecting them to random UDP loss inside network (or delaying them beyond acceptable limit at TCP buffer). The scheme requires however, an active notification mechanism by the underlying transport protocol. In this research we show the potential advantage of such an active TCP windowing mechanism. If there is congestion, we demonstrate a novel network, which is now able to notify the application, and the application then actively participates in a symbiotic *exponential-back-off and additive-increase* like scheme [PeDe00] leading to a much effective quality conformant congestion management. The mechanism can potentially facilitate building a new class of network aware applications. In this paper we present one such concept application-- a network aware video transport mechanism. In this scheme we demonstrate an interactive version of TCP socket and a smart network aware rate adaptive video transmission system in the context of MPEG-2 video transcoding [ISO96].

The transcoder has been designed to sit at the entry point or inside a network using technology such as active net or multi-protocol switch. It offers a rate adaptation service to facilitate video communication in a network with widely different bandwidth. The transcoding mechanism observes the local transport layer characteristics and can accordingly adjust the outgoing MPEG-2 stream bit-rate. Fundamentally a new class of solutions to the classical problem of network congestion and quality-of-service provisioning is feasible. In this paper we show the qualitative and quantitative improvement that we have observed from its concept proto-type. Before we present our proposed scheme and the results-- below we provide a brief overview on the current research in network transport.

1.1. Related Research Works

IP has recently undergone major redesign resulting into IPv6. The transport layer of the Internet has seen lesser change. Though in the mid-90s a working group named TCP Next Generation (TCPng) has been formed but any RFP is yet to come. Most of the research in TCP deals with TCP timer management. In recent years a number of proposals have been made particularly in the face of wireless integration. Increased network asymmetry makes the time management even more difficult. Also, the improvement of the power efficiency (TCP assumes all nodes are always up) has been a goal of recent research. Selective Acknowledgment TCP [MaM96a], Forward Acknowledge TCP [MaM96b], and TCP Probing [TsBa00], Split TCP (proposed in WAP) [Baba94, AlPa99, Mont00], are just some of the techniques proposed. Also quite a few proposals has been made to keep TCP intact but to augment support mechanisms at Data Link Layer such as snoop [BPSK96], Automatic Repeat Request (ARQ) [DesC93], Transport Unaware Link Improvement Protocol TULIP [PaGa00]. In general contemporary research in transport layer has targeted increasing its bulk delivery capacity by way of improving the techniques within and beneath transport layer. However, the application interface has been kept fixed. No significant functionality has been added in this layer. One of the novel aspects of this work is that we make a case for such functional improvement for the transport service—interactivity.

In the application side recently various quarters have proposed research in complementing the shortcoming of traditional transport service by way of devising ingenious application layer technology. Particularly active areas are scalable video communication [BrGM99, KhYa01, Rhee98, Wolf97, GhJo00] and distributed simulation [RiFa00], web caching and prefetching [KhTa01, JulC00, FCDG99, JaCa98, KrLM97], and another very recent wave has been sparred by mobile information systems (anytime-anywhere-systems) [Bagg98, KrWa00, SVSB99, YeJK98, GhPS00]. While, such wide interest indicates the importance of network aware applications technology, but due to the lack of any network layer support for sensing network states, most solutions tend to focus on operability across a wide variation in transport conditions. However, the lack of any direct feedback from transport layer requires these solutions to be dependent of various indirect probing tools (often unspecified) or simply add redundancy, which can blindly survive wide variation in transport loss characteristics. Less they depend on using information about network local state—or network awareness.

In the above context, we propose impairment management approach that utilizes direct feedback from the transport layer. We demonstrate that the technique enables devising precise and targeted solutions, which are otherwise not easy to achieve. As we will demonstrate, in general, it will increase the efficacy of the solutions and will be more resource efficient. We describe the MPEG-2 transcoder rate control mechanism with symbiotic rate adaptation. We also propose a novel qualitative enhancement of the transport layer, which can provide event notification to facilitate this symbiosis process of the transcoder. We show how this network-application integrated strategy with rich domain knowledge (in this case MPEG-2) can make the best use of the available bandwidth offered by the impaired network. There is no easy way of attaining such optimality just from network layer strategy. In this paper we show the symbiosis mechanism that we have implemented. In the next section we first outline the system model. In section 3 we then present the symbiotic rate control mechanism-- the application component that provides the network aware solution. The model has been developed by closely following the MPEG-2 Test Model 5 (TM5). While the detail can be found in [MPEG00], in this paper we describe the salient part of the rate control architecture that enables the symbiosis. Finally, in section 4 we share performance of the scheme. The analysis has been performed using a real implementation of the symbiotic transcoder, and letting it run on a simulated interactive TCP.

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- however at a dynamically selected rate. This decoupling also has the benefit that the transcoder can be made to auto 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

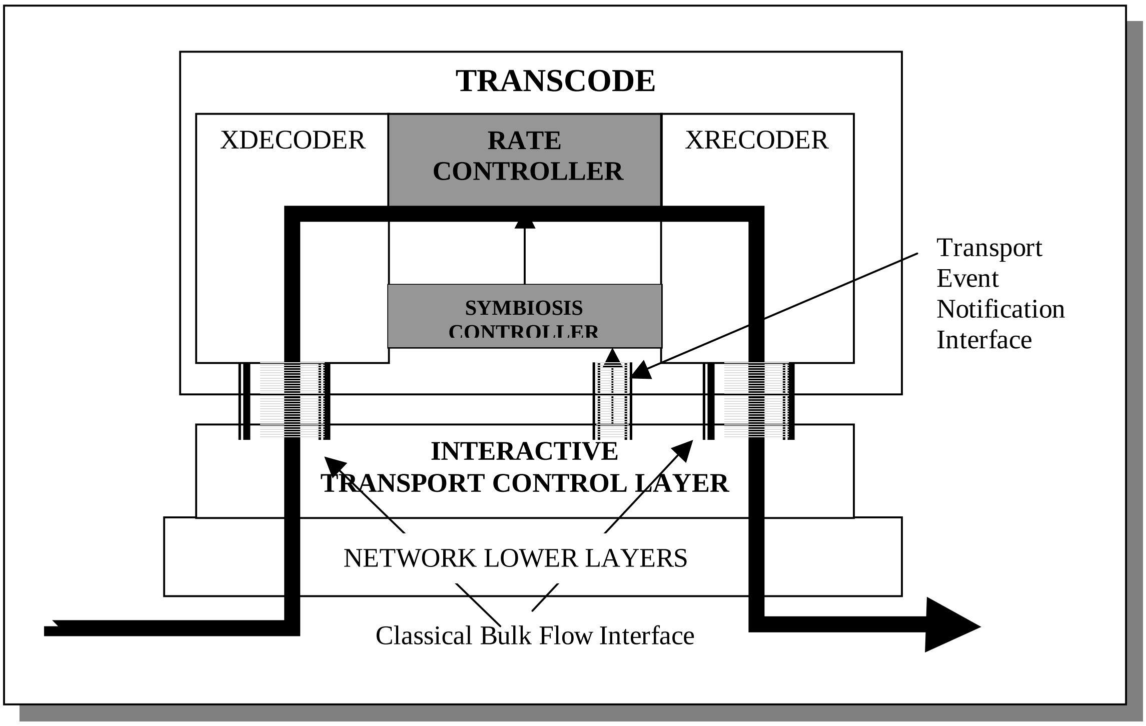

Fig-1 Interactive transport system and smart transcoder

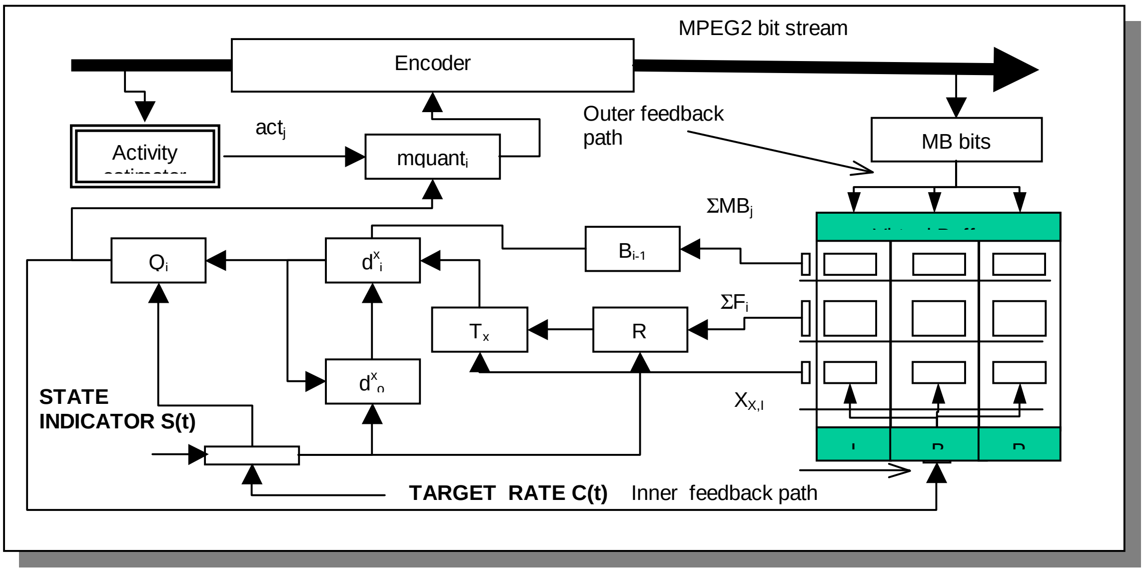

Fig-2 MPEG-2 feedback based symbiotic rate control system

active networking [TSSW97, GuTe98]. Fig-1 explains the system components.

2.2. Transport Control:

The transcoder sits on top of the interactive transport control layer-- TCP Interactive. Unlike conventional TCP, this novel 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.

2.3. Transcoder Architecture:

The transcoder unit has a decoder, and a re-encoder[1]. 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-2. 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) as well other subjective factors. The proposed mechanism is also a double-loop feedback control mechanism where the output bit-rate is continually sensed to determine overall piecewise constant rate, with appropriate accounting for variations in frame/picture type like 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

[1] It is in general full logic MPEG-2 transcoding is a daunting computational task particularly because of the encoding. However, a number of recent techniques (including ours) have been identified for accelerated transcoding. We note that primary encoding at encoder and the secondary encoding employed in the transcoder are not the same. In the re encoding many information are available adhoc. Several computations (such as motion vector estimation) can be bypassed-- significantly under cutting the transcoding cost [KPGOG01].

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. The quantized output for intra- and non-intra frames are respectively given by:

$$y = \frac{f(x, quant\_step) + .75 \times mquant}{2 \times mquant}$$

$$y = \frac{16 \times f(x, quant\_step)}{mquant}$$

Here x is the DCT coefficient, $y=f(x, quant\_step)$ is determined from ISO/IEC 13818-2 tables [ISO96]. 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[2] for this experiment is the Q_j. It is a modulation parameter, that determines how the allocation of frame-bits itself is varied.

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.

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

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

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, though, the bit generation temporarily reduces however, the virtual buffer fullness quantity is continually updated, enabling 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

[2] 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 for estimation of N_act_j. This is like TM-5 but in addition, it allows spatial distribution of the bits to be controlled for a given allocation of frame bits.

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.

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

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):

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

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

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

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 $Q_j = \left[31 \times d_j^x \cdot r^{-1}\right]$ 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.

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*. 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 maximizes applications requirements. In this experiment we have designed a symbiosis, which responds to a timeout event[3]. 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 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

[3] Current TCP implementations further fine-tune the envelop characteristics.

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:

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

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

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

The control function performs *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

For our experiment we have built a simulator for the proposed interactive TCP and subjected it with real MPEG-2 transcoder generated traffic. The simulator allows the underlying data link layer to have various impairment conditions. Below we share some representative results. 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. In this section we also show the frame wise detail event trace of what happens to the first 250 of the frames. We simulated both the classical transport channel (labeled as TCP) as well as interactive transport channel (iTCP). We let the video generator (transcoder) feed into the video stream. We simulated channel time-out event using a uniform random distribution. We further assumed, that this event is independent of the video stream size (due to congestion deep inside the network). In the classical mode, we let the transcoder operate in error unaware mode and generate the video using TM-5 [MPEG00] rate control at 4 Mbps. Transport control protocol

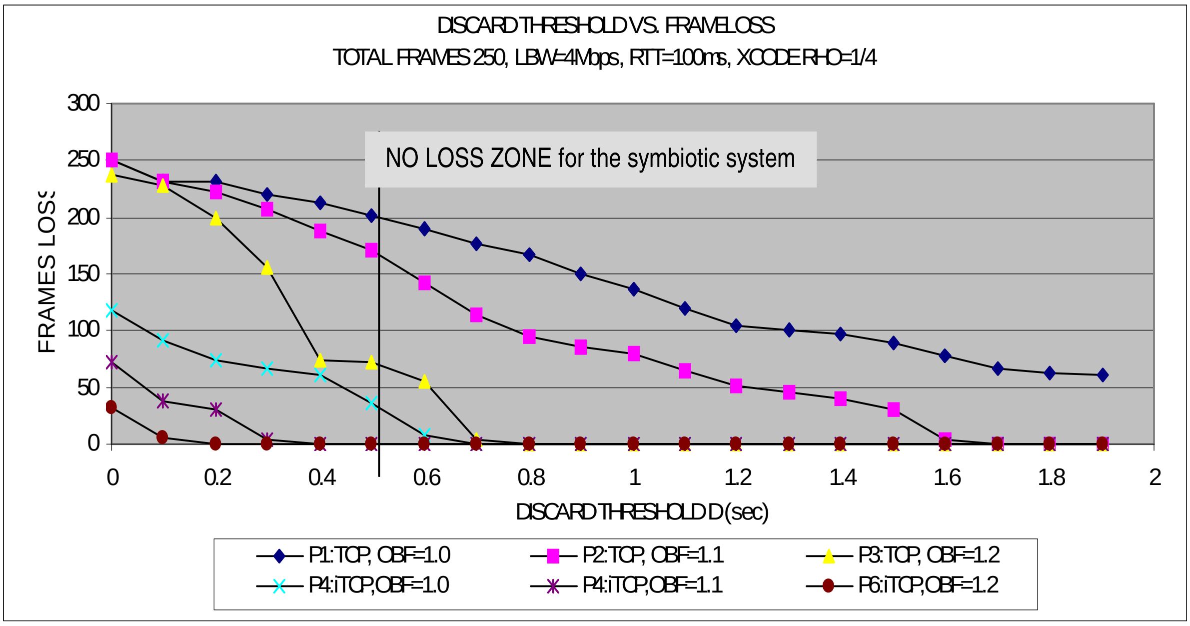

Fig-3

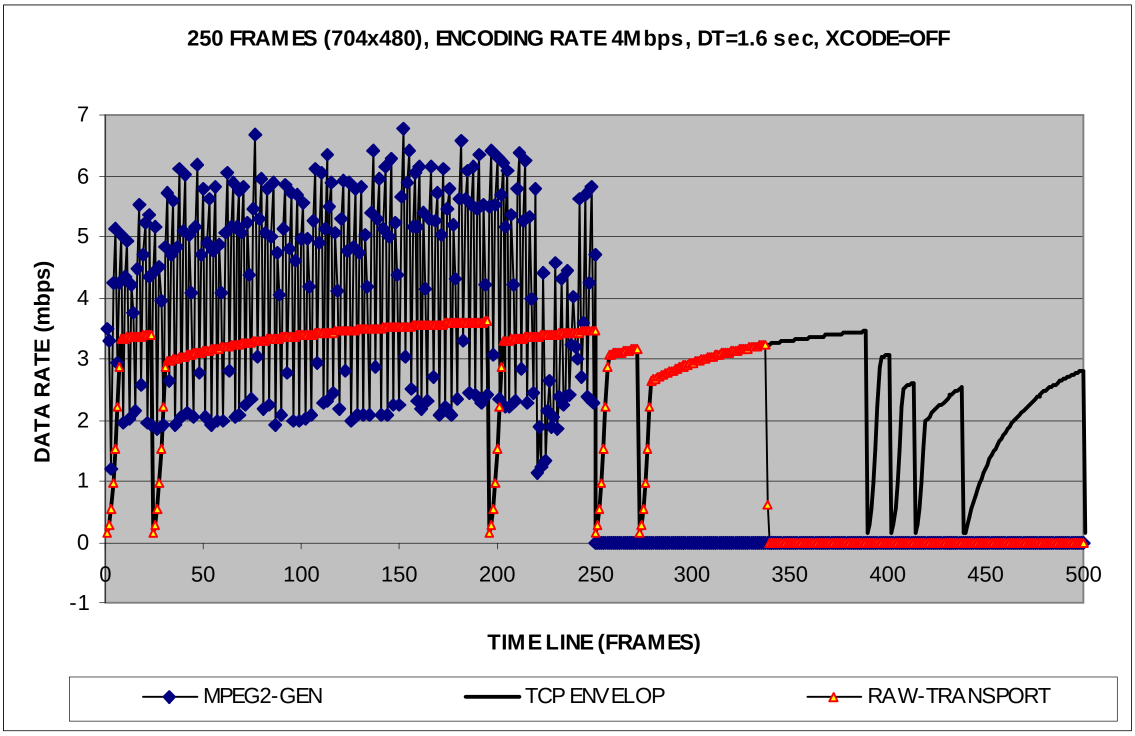
250 FRAMES (704x480), ENCODING RATE 4Mbps, DT=1.6 sec, XCODE=OFF
DATA RATE (mbps)
7
6
5
4
3
2
1
0
-1
0
50
100
150
200
250
300
350
400
450
500
TIME LINE (FRAMES)
MPEG2-GEN
TCP ENVELOP
RAW-TRANSPORT

Fig-4(a)

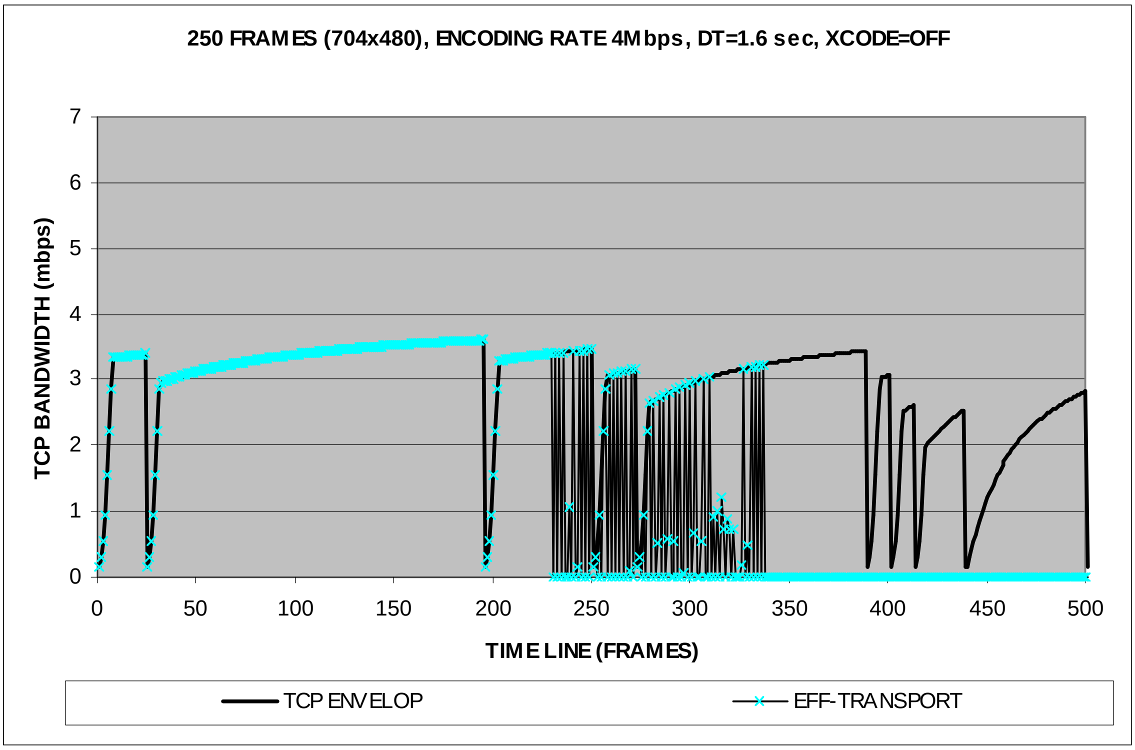
250 FRAMES (704x480), ENCODING RATE 4Mbps, DT=1.6 sec, XCODE=OFF
TCP BANDWIDTH (mbps)
7
6
5
4
3
2
1
0
0
50
100
150
200
250
300
350
400
450
500
TIME LINE (FRAMES)
TCP ENVELOP
EFF-TRANSPORT

Fig-4(b)

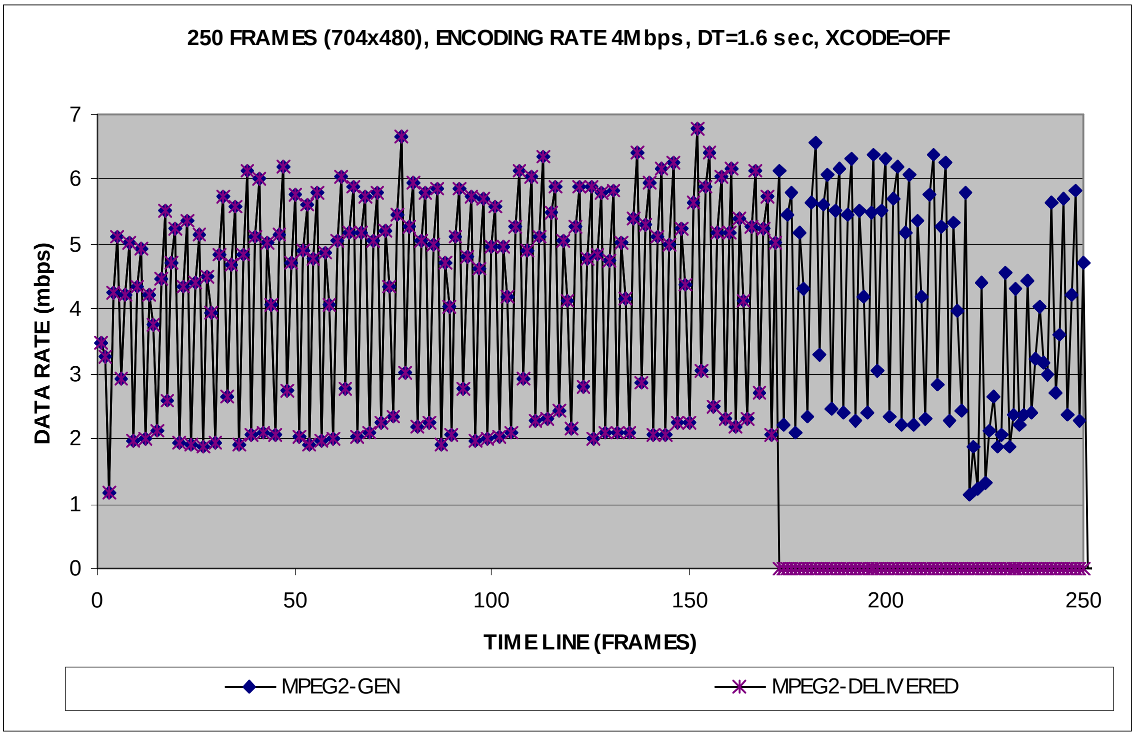
250 FRAMES (704x480), ENCODING RATE 4Mbps, DT=1.6 sec, XCODE=OFF
DATA RATE (mbps)
0
1
2
3
4
5
6
7
0
50
100
150
200
250
TIME LINE (FRAMES)
MPEG2-GEN
MPEG2-DELIVERED

Fig-4(c)

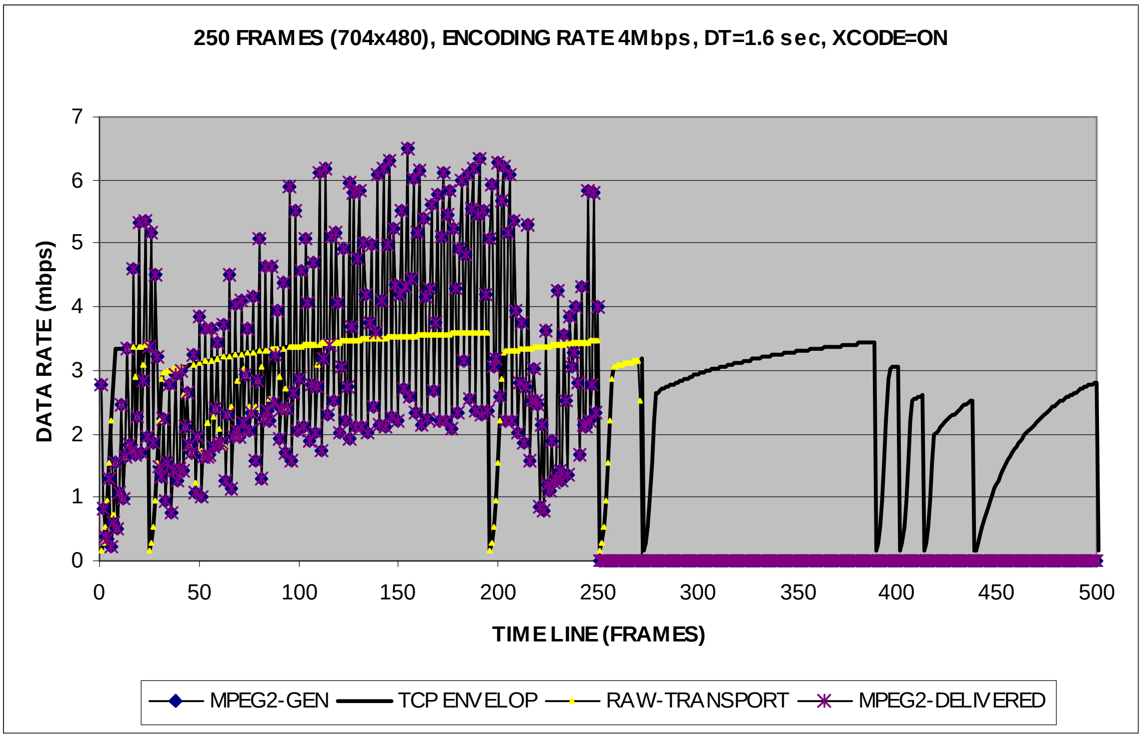
250 FRAMES (704x480), ENCODING RATE 4Mbps, DT=1.6 sec, XCODE=ON
DATA RATE (mbps)
0
1
2
3
4
5
6
7
0
50
100
150
200
250
300
350
400
450
500
TIME LINE (FRAMES)
MPEG2-GEN
TCP ENVELOP
RAW-TRANSPORT
MPEG2-DELIVERED

Fig-4(d)

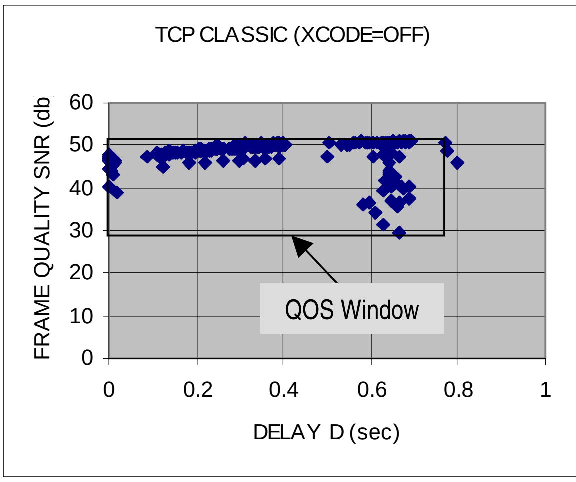

Fig-5(a)

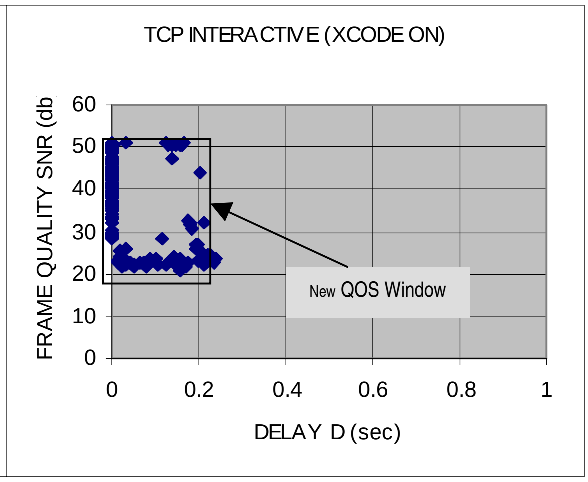

Fig-5(b)

buffered the generated data while the transport layer exercised binary back-off and additive recovery at time-out events. In the interactive mode the transcoder according to the symbiosis controller varied the video rate for interactive TCP. The video data was received into an analyzer. The transcoder and the analyzer both recorded the 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.

The objective of our first experiment is to observe how the rate of frame discard varies with various choices of the threshold. We were also curious to see how the discard rate varies with the allocated transport bandwidth. Correspondingly, we varied the channel (link layer) bandwidths from 100-120% over the coding bandwidth. Fig-3 plots the dramatic difference between the performances of the two channels. It plots the number of failed frame (y-axis) with various cut-off delays (x-axis).

In this symbiosis we used ρ=0.25. The top three curves (P1, P2, P3) show the frame loss for classical channel for three path bandwidths (1.0, 1.1 and 1.2 times 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.

Fig-4(a)-(d) explain the internal mechanics of this entire arrangement. 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. Fig-4(a) plots. Fig-4(a) shows the data generation rate, the TCP window envelop, and the network transfer rate (that occurred within this envelop). As evident, in the unaware scheme due to the back-offs the buffer congestion propagated in time. The congestion grew worse over time. The last of the frames was transported almost near the time-line point 350. How much of the data that was transported was actually useful to the applications? To illustrate the situation, in Fig-4(b) we have plotted the effective transfer rate for a discard threshold 1.6 sec. As evident, although almost 30% of the data that network carried was effectively useless from the application point of view. Fig-4(c) plots the same data at application level. It shows in terms of frame line, which has been generated and which was effectively delivered. Fig-4(d) now re-plots all four quantities for the TCP interactive experiment. It shows the data generated by the transcoder, the TCP envelop (same as before), the network transport rate, and the effective frame delivery. As evident the generated MPEG-2 data now resembles closely the TCP envelop. The MPEG-2 generated matches identically with the MPEG-2 delivered. Consequently, there was no loss in the effective transfer rate.

The application level trade-off that occurred in this experiment is now illustrated in the 2-dimensional plots for these two situations. In Fig-5(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 deliver all the frames with .6-.7 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, resulting in revolutionary advance compared to the almost primitive and time insensitive ways that congestion is generally handled today.

5.Conclusions and Current Work

In this paper, our principal focus has been to demonstrate the potential benefits that a new generation of network aware applications can bring. We have outlined the benefits of a novel 'interactive' generalization of the classical transport protocol—that can bring fundamentally new solution to many of today's hard to tackle problems. In this paper we have demonstrated one such case- how quality conformant congestion control—one of the most difficult problem in current networking research, can be handled in the case of an interactive transport control protocol and a symbiotic MPEG-2 transcoder. It perhaps provides a new dimension in the approaches, which have been proposed in recent literature in congestion management and also in the quality of service provisioning.

Though it is generally casually mentioned that video information can withstand loss, unlike say a text file- but it is not true in its simple sense for most real video stream. While loss of few DCT component can be absorbed, but loss of few bits from a critical header can render an entire sequence useless. More precisely a video stream carries information with various levels of importance. While, some can be sacrificed, but some are as important as the sync bits. Experiment has shown that random dropping of just 20% of the packets makes most video streams useless. In sharp contrast, application level knowledge (such as the one shown in this transcoder) can offer much wider rate scalability thus the symbiosis. Essentially, the demonstrated transcoder rate control mechanism performs a form of optimum stream packing. When the transport capacity dwindles to minimum, the transcoder even then can pack only the very important header bytes—on which the decoding of future frames depend, even if it has to forgo the rest of the pixel data. Such discrimination is impossible to achieve at network level. In this paper, we have demonstrated it is feasible if the application is conveyed the transport impairment states.

It is understood that all applications may not be willing to use notification. While, for simpler and less demanding applications (such as email) current time tested transport protocol TCP will remain adequate, but with the advent of complex network wide applications with requirement for sophisticated inter-process communication perhaps solutions have to include the applications.

Also, interesting is the fact that the classical demarcation between the 'network' and 'application' processes is increasingly blurring. While, current transport protocol was initially designed for end-to-end application modules, now even many 'network' system components are using them. In this rather interdependent system, such lack of interactivity between the service provider and the service subscriber processes will increasingly make it difficult to build high-level strategies for low-level impairments.

With in the scope of this paper, we have only focused on the potential pay-off from such application/network symbiosis, and did not discuss the transport implementation. Nevertheless, implementation of the required out-of-band and out-of-order communication at transport layer is non trivial. However, some of the issues have already been visited during the initial implementation of 'urgent point' and 'push' in TCP, though most current implementations do not support them directly [Come00]. The key concern is how much the realization of notification feature will impact the efficiency of its normal mode operation. We have recently anticipated and presented a more generalized architecture of an interactive programmable channel formalism on active network. Formalism can be found in [KhYa00, KhYa01]. We have also just concluded additional work that provides optimum symbiotic states for a given transport system and target delay

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

guarantee, and will be presented in a forthcoming publication.

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

6.References

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

[Bagg98] Aline Baggio, System support for transparency and network-aware adaptation in mobile environments; Proceedings of the 1998 ACM symposium on Applied Computing, 1998, Pages405 – 408.

[BPSK96] Hari Balakrishnan, Venkata N. Padmanabhan, Srinivasan Seshan and Randy H. Katz, A comparison of mechanisms for improving TCP performance over wireless links;; Conference proceedings on Applications, technologies, architectures, and protocols for computer communications, 1996, Pages 256 - 269

[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.

[Come00] Comer D. E., Internetworking with TCP/IP, Principles, Protocols, and Architectures, 4th Ed, Pretice Hall, New Jersey, USA, ISBN- 0-13-018380-6, 2000

[DesC93] DeSimone, Chuah, et al. "Throughput Performance of Transport-Layer Protocols over Wireless LANs". In Proceedings of Globecom '93, December 1993.

[FCDG99] A. Feldmann, R. Caceres, F. Douglis, G. Glass and M. Rabinovich, Performance of Web Proxy Caching in Heterogeneous Bandwidth Environments, Proceedings of INFOCOM 99, 1999.

[FlJa93] Floyd, S, & V. Jacobson, Random early detection gateways for congestion avoidance, IEEE/ACM Transactions on Networking, 1(4), August 1993, pp397-413.

[GhJo00] Ghose, Debasish; Kim, Hyoung Joong, Scheduling Video Streams in Video-on-Demand Systems: A Survey, Multimedia Tools and Applications, Volume: 11, Issue: 2, June 1, 2000, pp. 167-195.

[GhPS00] Vittorio Ghini, Giovanni Pau and Paola Salomoni, Integrating notification services in computer network and mobile telephony; Proceedings of the 2000 ACM symposium on Applied computing 2000 (volume 2), 2000, Pages 549 – 553.

[Gopa99] Gopalan Suresh Raj, A Detailed Comparison of CORBA, DCOM and Java/RMI, [URL: http://www.execpc.com /~gopalan/ misc/compare.html,retrieved on December 1999]

[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/openarch98.html

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

[JaCa98] Q. Jacobson, Pei Cao, Potential and Limits of Web Prefetching Between Low-Bandwidth Clients and Proxies, 3rd International WWW Caching Workshop, Manchester, England, June 15-17 1998.

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

[JuLC00] J. Jung, D. Lee, and K. Chon, Proactive Web Caching with Cumulative Prefetching for Large

Multimedia Data. Procs. of the 9th International World Wide Web Conference, Amsterdam, Netherlands, May 2000.

[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,

[KhGu01] Javed I. Khan and Q. Gu, TR2001-03-01 Interactive TCP and the Transcoder Symbiosis Report, 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] March, 2001]

[KhTa01] Javed I. Khan and Qingping Tao, Partial Prefetch for Faster Surfing in Composite Hypermedia, Proc. of USENIX Symposium on Internet Technology and Systems, San Francisco, March 2001, (in press).

[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).

[KhYa00] Javed I. Khan & S. S. Yang, Made-to-order Custom Channel for Netcentric Applications over Active Network, Proc. of International Conference on Internet and Multimedia Systems and Applications, IMSA 2000, November 20-23, 2000 Las Vegas, USA, pp22-26.

[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

[KrLM97] T. Kroeger, D. D. E. Long & J. Mogul, Exploring the Bounds of Web Latency Reduction from Caching and Prefetching, Proc. Of USENIX Symposium on Internet Technology and Systems, Monterey, December 1997, pp-319-328.

[KrWa00] Balachander Krishnamurthy and Jia Wan, On network-aware clustering of Web clients, Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, 2000, Pages 97 – 110.

[MaM96a] Mathis, Mahdavi, et al. TCP Selective Acknowledgment Options RFC 2018, April 1996.

[Mam96b] M. Mathis and J. Mahdavi. "Forward acknowledgment: Refining TCP congestion control." In Proceedings of the ACM SIGCOMM, August 1996, Pages 281 – 291.

[Mont00] Montenegro, Dawkins, et al. Long Thin Networks, RFC 2757, January 2000.

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

[PaGa00] C. Parsa and J.J. Garcia-Luna-Aceves, "Improving TCP Performance over Wireless Networks at the Link Layer", Mobile Networks and Applications, 5,1, March 2000, pp57-71.

[Pede00] 19 L. L. Peterson and B. S. Davie. Computer Networks, 2nd edition, Morgan-Kaufmann, 2000.

[Pent00] Kostas Pentikousis; Can TCP be the transport protocol of the 21st century?; Crossroads 7, 2 (Dec. 2000), Pages 25 – 29.

[RaJa93] Ramakrishnan K, and R. Jain, A binary feedback scheme for congestion avoidance in computer networks with a connectionless network layer, ACM Transactions on Computer Systems 8(2), August 1993, pp158-181.

[Rhee98] Injong Rhee, Error control techniques for interactive low-bit rate video transmission over the Internet; Proceedings of the ACM SIGCOMM '98 conference on Applications, technologies, architectures, and protocols for computer communication, 1998, Pages 290 - 301

[RiFA00] George F. Riley, Richard Fujimoto and Mostafa H. Ammar, Network aware time management and event distribution;; Proceedings of the fourteenth workshop on Parallel and distributed simulation, 2000, Pages 119 – 126.

[StZh97] Stoica, I, and H. Zhang, A hierarchical fair service curve algorithm for link sharring and priority services, Proceedings of the SIGCOMM 97, October 1997, pp29-262.

[SVSB99] Prasun Sinha, Narayanan Venkitaraman, Raghupathy Sivakumar and Vaduvur Bharghavan, WTCP: a reliable transport protocol for wireless wide-area networks; Proceedings of the fifth annual ACM/IEEE international conference

[Ten96] Tanenbaum, A. Computer Networks, 3rd edition, Prentice Hall, 1996.

[TsBa00] 24 V. Tsaoussidis, H. Badr, "Efficient Energy/Throughput Tradeoffs of Reliable Transport Protocols using Probing Mechanisms", in Proceedings of PDPTA 2000, CSREA Press, Las Vegas, June 2000.

[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[Find Related Articles] [HTML | PDF | PS]

[YeJK98] Tao Ye, H.-Arno Jacobsen and Randy Katz, Mobile awareness in a wide area wireless network of info-stations; The fourth annual ACM/IEEE international conference on Mobile computing and networking, 1998, Pages 109 - 120