Technical Report 2003-09-01
Internetworking and Media Communications Research Laboratories
Department of Computer Science, Kent State University
http://medianet.kent.edu/technicalreports.html

# Dynamic Transport Enhancement for Time Elastic Traffic with Transientware

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

17 April 2003

*Abstract—* **The paper presents the concept of transientware—a mechanism by which sensitive applications can launch application level targeted and dynamically adaptive enhancements of transport service. The proposed transientware can be used with current transport protocols with interactivity enhancement. As a proof of concept we have recently designed and implemented a functionally enriched transientware enabled TCP on FreeBSD Unix. We then show the performance of a TCP friendly adaptive MPEG-2 video transcoder, which can directly interact with this transport and adjust its outgoing video rate to satisfy temporal quality constraint of the stream via a dynamic rate adaptive scheme. We report potential dramatic improvement in time-bounded video delivery of elastic traffic from live experiment.**

*Index terms*-- **netcentric applications, TCP interactive, transcoding, temporal QoS.**

## 1 INTRODUCTION

### 1.1 *Transient-ware for Dynamic Adaptation:*

The interface between and application and network is emerging as a critical research problem in internet base advanced applications research. In this paper we discuss a method where demanding applications themselves can optionally append lightweight transient coupler processes called "T-ware", which can respond to dynamic network events. It provides application aware smart transport enhancements on the transport services provided by network layer. The T-ware solution is particularly attractive for emerging areas with non-traditional transport needs such as space networking, high performance grid applications, and instrument networking. The approach does not require a static middleware in the traditional sense in the data path between application and network. Rather, these are optional and transient and only invoked conditionally when required. Nor it requires any new or any complex change in conventional transport mechanisms. Also, the transport enhancers can be very effective and efficient as they can implement domain knowledge enriched fully application aware solutions targeted to meet the applications specific need. These 'T-wares' does not interfere with the Internet's network dynamics. Rather it enables applications to be more network-aware for its own good. The 'T-ware's enhancers can be used to implement smart solutions to many network transport deficiencies of current transport systems by application programmer themselves (or third party enhancement experts). As a proof of concept we have recently implemented the 'T-ware' provision on a FreeBSD Unix. In this paper we present how the T-ware helps an application in its effort of dynamic adaptation in the face of dynamic network congestion while sending time sensitive elastic traffic. The solution has been recently tested for high fidelity video streaming over worldwide network over ABONE with dramatic improvement in time bound quality [KhZh03]. This paper presents the conceptual framework for the T-ware.

### 1.2 *Example Case: Congestion Response for Time Sensitive Elastic Traffic*

The problem to which we show a smart solution via T-ware is quite challenging. Coping with the dynamically varying characteristics is an open problem in network applications. In asymmetric networks, such as wireless, very long haul networks, or mobile or power aware device integrated networks, a subset of links in the datagram pathway can show substantial dynamically varying delay and capacity characteristics. Such link dynamism is very hard on the classical transports. For last few years it has been increasingly felt that applications, when they are time sensitive, have to be more integrated in the solution for any effective congestion control [BrGM99, Wolf97]. A particularly noteworthy development along this direction is the new TCP friendly paradigm. Example works are [KeWi00, ReHE00, SiWo98, PrCN00]. [SiWo98] presented a TCP rate-based pacing mechanism that particularly takes note of document transfer characteristics. For example [Rama00] investigated a Real Time Protocol for multicast. [SiWo98] proposed building TCP friendly application where application relies on real-time transport protocol (RTP) mediated end-to-end measurement. [ReHE00] discussed another framework where applications can control rates based on their end-to-end measurements (similar end-to-end technique is used in RealPlayer). Also several works further investigated combining application specific information from several streams into one clearinghouse architectures for aggregated congestion control. For example, Congestion Manager [ABCS00, BaRS99] is a middle 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 (CM). CM tries to minimize congestion by coordination between multiple sending streams. [PrCN00] used multiple probing mechanics for aggregate congestion control.

In this context, the approach that we will present is based on dynamic yet local feedback between the application sending end-point and the sending-end-point of the transport layer, via an auxiliary application module. It is different from the end-to-end communication based adaptation between the local and remote application or middle-layer end-points. The application strategy of response is then simple and intuitive. Delay conformant communication for time sensitive elastic traffic over a channel pathway with dynamically varying links is possible if the original data volume can also be dynamically adapted by its originator-- the application with respect to the transport capacity[1].

The adaptation is applicable for traffic where it is possible to dynamically adjust the data generation rate. We call it ***elastic traffic***. Most perceptual data, such as audio, video streams generally belongs to this traffic class. The benefit of elastics adaptation has been demonstrated in [KhGZ02].

### *1.3 Transient-ware Approach*

It is important to note that techniques such as elastic adaptation, typically require complex application level knowledge besides the dynamic state information about the transport impairment. Most of the current suggestions attempted to handle by growing the network layer directly or by adding static system level middleware. Unfortunately, it means network or system need to know substantially about all the complex application specific techniques. Despite some initial success the field is still wide open. It seems finding a uniform way of supporting any QoS model at this intermediate stage is harder than anticipated.

This work tests a hypothesis if it can be performed

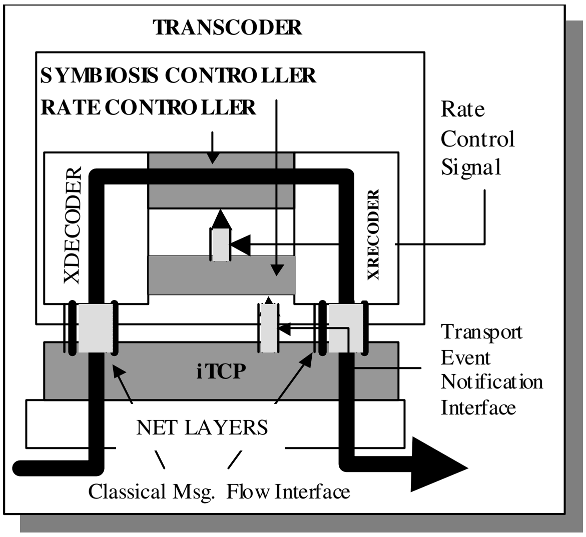

Fig-1 Interactive transport and symbiotic transcoder

---

[1] It is interesting to note that the significance of 'interactivity' has been envisioned in the original TCP specification. RFC 793 states status notification as a general requirement of TCP interface. However, it seems this vision was lost in subsequent implementations.

in the other way. Let the actual transformation be performed at the application level with application specific technique, but pass on the network state to them. We call it **transport interactivity**. Therefore, instead of performing core adaptation in the network layer we pass on network state to a special application component. However, what are the challenges?

A critical problem in current implementations of transport protocols is that these do not have any convenient mechanism by which dynamic states can be communicated back. Indeed, in TCP-friendly solutions lack of such communication has forced applications to use end-to-end techniques to 'guess' the network problem states. We suggest removing bulk of this guesswork via direct *protocol interactivity* and let application specified transient units handle the symbiosis or elastic adaptation with application level knowledge.

To test the efficacy of the principle, we have recently designed and implemented an enhanced TCP kernel which can provide feedback to its subscriber application. We have also suggested an enhanced socket interface via which an application can subscribe, and register the T-wares. We have also implemented a corresponding video rate transcoder system that works in symbiosis with the network to dynamically respond to network congestion. This transient-module can help the transcoder to actively participate in a custom symbiotic *exponential-back-off and additive-increase* like scheme in application layer with deep application level knowledge.

The resulting scheme is similar in spirit to the TCP friendly approaches. However, the novelty is in how it is done. We expect network (or system) layers to remain as simple as possible. The means and techniques for rate reduction remain with the producer application. The responsibility of the network layer is simply to pass on only selected end-point events to the applications.

As, we will show the scheme is not only intuitive and simple, but also surprisingly effective compared to many other recently proposed schemes which involve much more complex system/network layer reorganization.

Naturally, one of our important concerns in this direct feedback based system is the impact of close coupling between the application and the network. Extension of network/application interfacing has been traditionally avoided by network protocol designers. We will also present some encouraging results on this issue.

The results presented in this paper are not a simulation, rather report from a real implementation of the concept system. The implementation has two components. (i) First is an **interactive transport**. We have extended the sending end-point of Free BSD Unix TCP. Interface and state wise it is fully compatible with all classical TCP peers as well as classical applications. The second is (ii) a novel **symbiotic MPEG-2 full logic transcoder** [KYGP01, KhYa01], which is capable of working in tandem with the interactive transport. The emphasis of this paper is performance of this implemented.

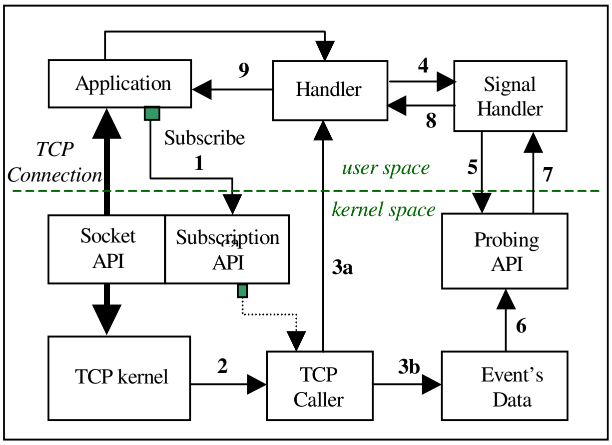

Fig-2. The TCP interactive extension. The added registration API allows demanding applications to subscribe to events and probe additional event data.

The paper is organized in the following way. In the next section, we first provide the overview of the T-ware symbiosis system. Section 2 presents the design of the interactive transport. Section 3 then presents the MPEG-2 transcoder and in particular the symbiotic rate control mechanism--the key application component that provides the key network aware solution. Finally, in section 4 we share performance of the scheme.

## 2 SYSTEM OVERVIEW

The application system has been developed as a three-part system -- *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. The system has been designed to sit either at the transport entry-point and perform conventional peer-to-peer conventional encoder based rate adaptation. Or, it can also sit inside a network using technology such as active router or a service proxy for targeted and localized congestion management. This approach has several advantages. It subsumes (i) the functionalities of server-client model. In addition, (ii) it allows rate adaptation on video stream that is already encoded and thus enables serving stored video at a dynamically selected rate. (ii) The 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 dynamically varying-capacity wireless links using techniques such as active routers [TSSW97, GuTe98]. The outgoing video stream from the transcoder, sits on top of the new iTCP layer (shows in Fig-1). The iTCP sending end features the event notification mechanism. The symbiosis unit of the transcoder performs the elastic adaptation.

## 3 INTERACTIVE TRANSPORT CONTROL

### 3.1 Architecture

The transcoder sits on top of the interactive transport control layer-- TCP Interactive. Fig-1 explains the system arrangement. 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 identical to the TCP classic. Except upon opening the socket, an adaptive application may bind an interrupt handler (T-ware) routine to the designated socket event. When, the event occurs the TCP triggers the T-ware. The T-ware is optional. If application chooses not to bind any, the system defaults to the silent mode identical to TCP classic.

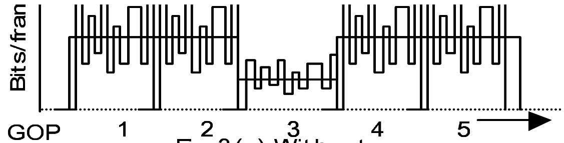

Fig 3(a) Without carryover

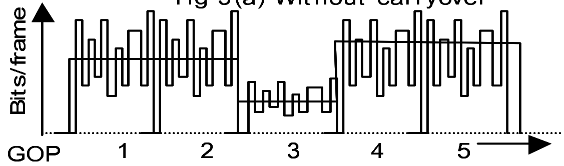

Fig-3(b) With carryover

The internal architecture of the TCP interactive is shown in Fig-2. The added *subscription API* helps applications to subscribe to TCP events, in this case the timer out. We have added a simple extension to the TCP kernel. The main unit is called *TCP Caller* unit. It is activated if the application subscribes. It keeps track of the TCP timeout events. 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.

### 3.2 Event Model

TCP state diagram reduces to 7 external events. We have made 4 of them accessible namely (i) retransmission timer out, (ii) congestion window *snd_cwnd* has reached the advertised window size *snd_wnd*, (iii) A new ACK was received, and (iv) A third duplicate ACK was received. A protocol is expected to make only a subset application accessible. Even a subset of it is expected to be subscribed by a particular application.

### 3.3 Compatibility & Interoperability

In the design of the TCP Interactive protocol we have retained three important protocol engineering principles namely *(i) network functional compliancy* (ii) *state-transition* compliancy, and (iii) *default-to-classical extension* interface model. These principles provide important advantages to the scheme. Its interface mechanism with IP layer remains functionally identical with TCP classic. Thus, it remains fully operational with all other currently deployed TCP remote-hosts. Only the server/transport hosts which are interested to run

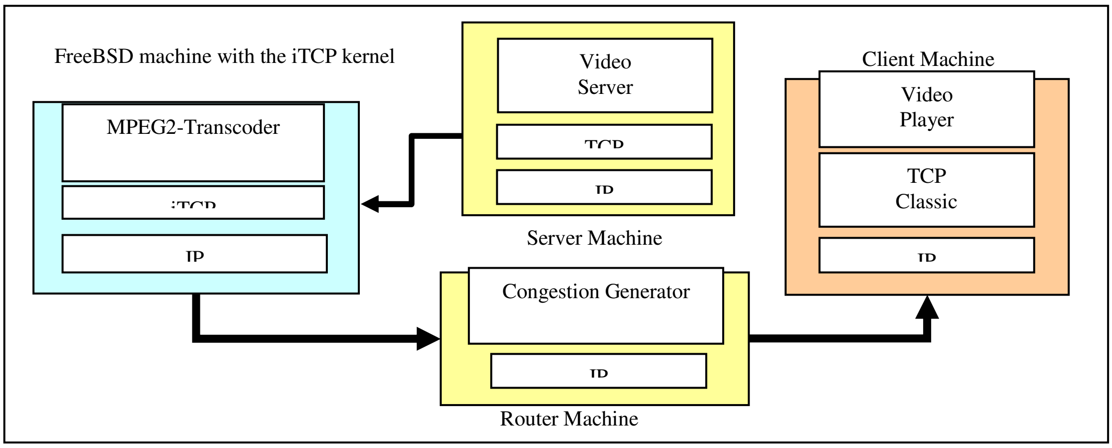

Fig-4. Experiment Setup.

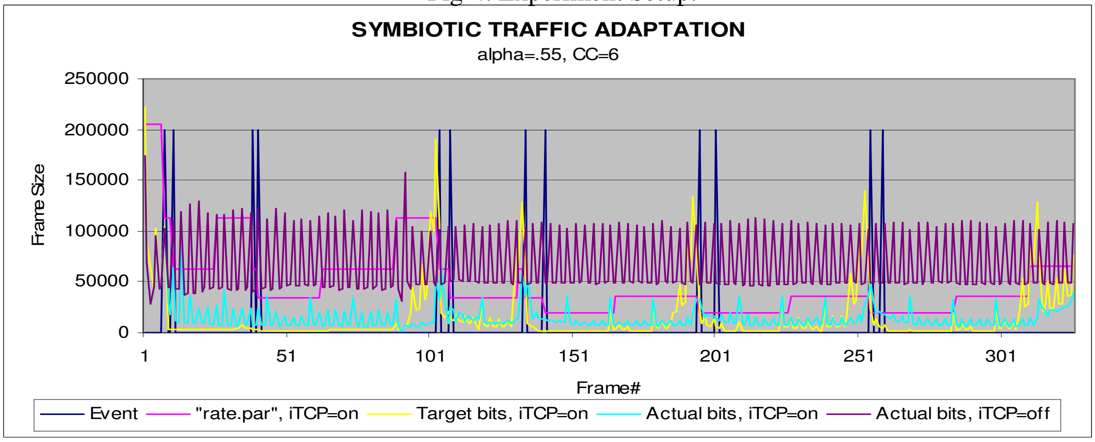

Fig-5. The binary-back off rate reduction in transcoder. There are six congestion bursts. The figure plots the incoming frame sizes, the event driven target rate (retraction ratio) specified by the symbiosis unit, and the resulting output rate from the transcoder.

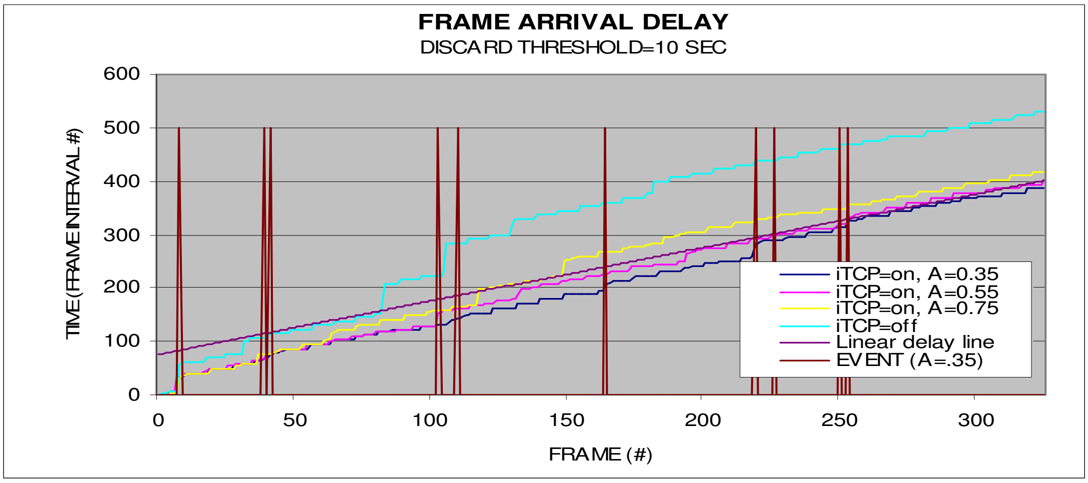

Fig-6(a). The figure plots the arrival time of the frames. For ideal case it should be linear. However, with each timeout event the backlog increased (observed as the step jumps in delay). The iTCP helps in eliminating these step increase in delays. The events correspond to A=.35 experiment.

adaptive software applications can locally upgrade to the extended model. It does not require any other host to modify anything on their site. Secondly, its *state-transition* behavior also remains identical to that of TCP classic. Thus, all other network embedded transparent elements which rely on certain assumptions about TCP behavior (such as congestion control schemes), will also not be affected. Thirdly, the '*default-to-classical*' extension of application programming interface enables the TCP interactive host to concurrently run all other existing applications without any code modification. Thus, it keeps all legacy applications 100% compatible even on the updated host. Notably, many other suggested congestion management potentially violate these critical protocol extension principles.

## 4 SYMBIOTIC MPEG-2 TRANSCODER

### *4.1 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.

### *4.2 Transcoder Rate Control:*

The rate control in the video transcoder is a complex process which involves sophisticated domain knowledge. 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. In our system we have designed a mechanism which performs rate control in two modes. In normal mode it operates as MPEG-2 TM-5 model. In frugal model it reduces the bit rate as per dynamic target, but remembers the saved bits (allocation not used) and optionally carries them over to the over next normal mode phase. Fig-3 illustrates the frame wise bit generation without and with carryover in these two modes. The detail rate control mechanism has been presented in [KhGR02].

### *4.3 Symbiotic Rate Determination*

The transcoder focuses on the mimicking a given target rate. The target rate dynamics are controlled by the T-ware unit-- 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* $\rho$. 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

---

[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, KHYa01].

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:

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

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.

## 5 EXPERIMENT RESULTS

### 5.1 *System Setup:*

We have implemented an MPEG-2 DTV symbiotic video rate transcoder that uses the above model. We have also implemented a TCP Interactive transport on a FreeBSD system. Fig-4 shows the experiment setup. In this scheme a transcoder has been placed on an intermediate node between the server and the client. Between the transcoder and the client we have also placed a congestion injector.

### 5.2 *Congestion Model:*

The congestion injector alters the routing table periodically. The congestion has been modeled as a sequence of bursts. The injector allowed the time between each consecutive burst and its duration to be individually specified. Two vectors *InterBurstTime*[], and *BurstDuration*[] are used for that purpose.

In our experiment, we used bursts with 3 seconds duration and random inter-burst time. To simulate various intensity of congestion we allowed the congestion to be increasingly more frequent (CCOUNT) within the transmission time of the test video. We let the congestion happen 3, 6 and 9 times within the transmission interval. However, the InterBurstTime[i] was generated randomly to avoid any effect of periodicity.

### 5.3 *Traffic Characteristics:*

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 2 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.

### 5.4 *Setup:*

For these set of experiments we run 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. 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

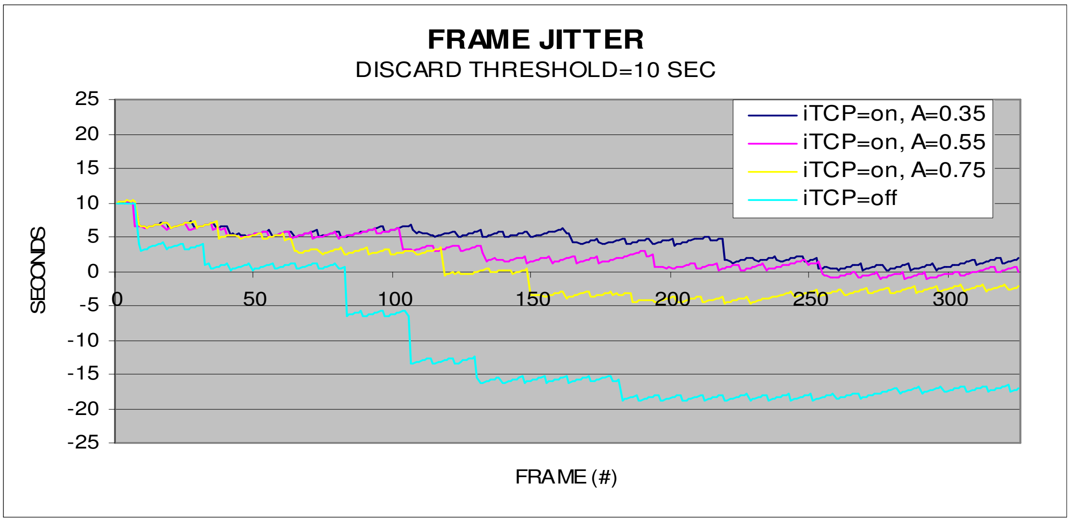

Fig-6(b). The figure plots the per frame jitter for the same set of experiments. A positive jitter however, means the frame arrived earlier than its ideal time. As shown, the classical TCP fell behind with each timeout event.

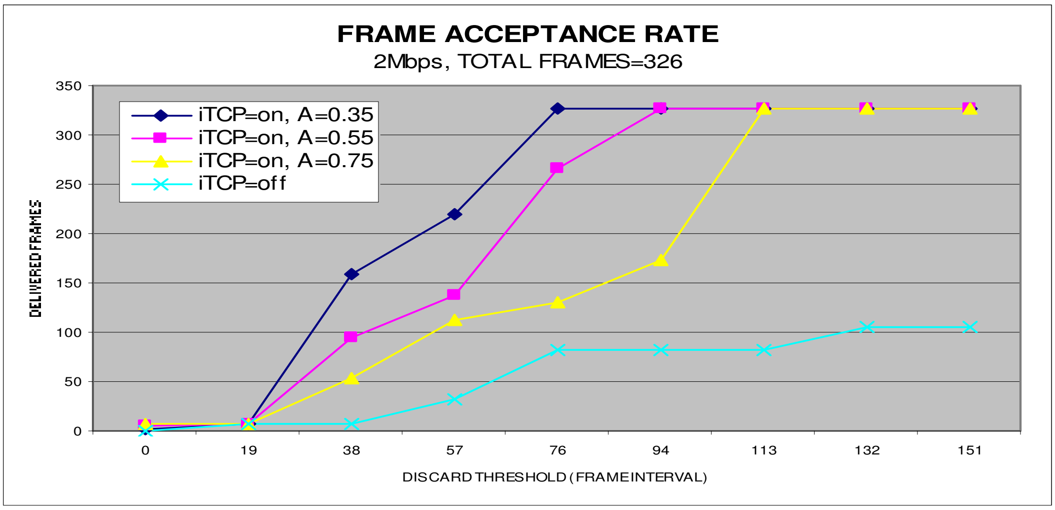

Fig-6(c). Frame loss due to delay as a function of discard threshold. If the discard threshold increases more frames are usable.

Table-1 The average delay and average picture quality for Y, U, and V components.

| | Average Delay | SNR(Y) | SNR(U) | SNR(V) |
|---|---|---|---|---|
| **iTCP=on, A=0.35** | 10.31 | 16.14 | 9.08 | 9.92 |
| **iTCP=on, A=0.55** | 12.81 | 15.61 | 8.67 | 9.46 |
| **iTCP=on, A=0.75** | 10.50 | 21.59 | 12.84 | 14.01 |
| **iTCP=off** | 32.27 | 25.31 | 15.90 | 17.21 |

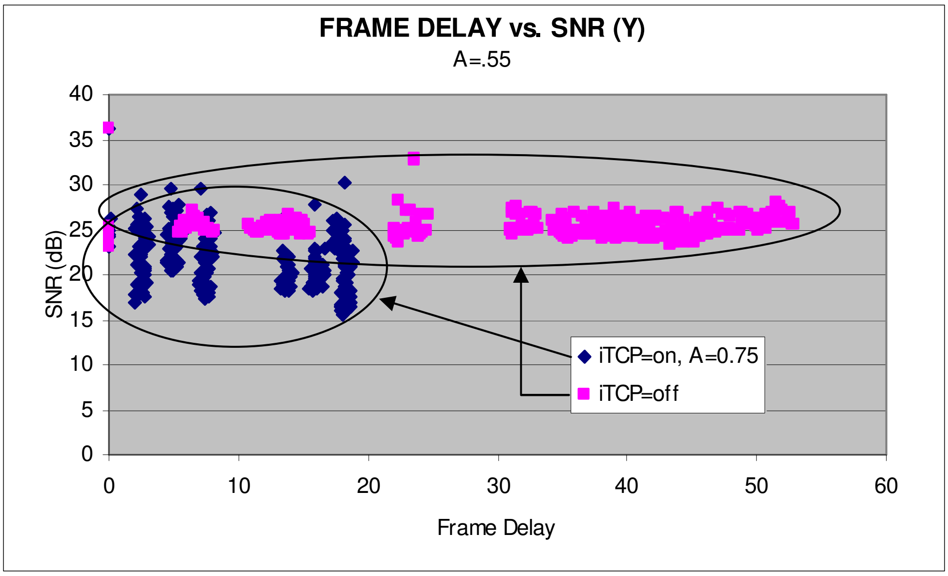

Fig-7. The two clusters shows the quality, delay tradeoff offered by the iTCP. The iTCP dramatically reduced frame delivery delay by controlled trade-off of the SNR quality.

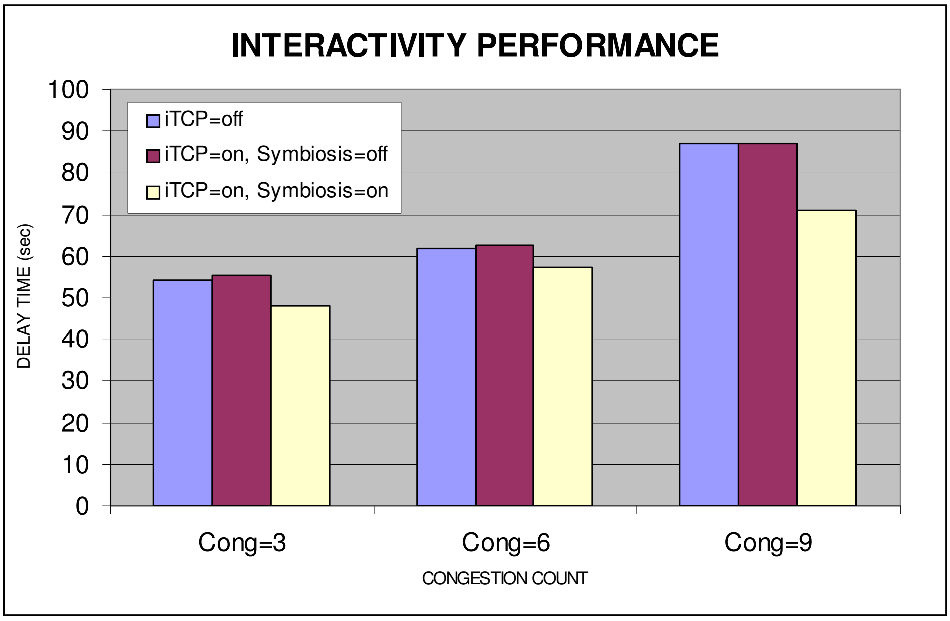

Fig-8. Compared to the overall transport time without interactivity (iTCP=OFF), there is a small overhead associated with iTCP event trapping mechanism (iTCP=ON, Symbiosis=OFF). However, this small overhead is vastly outweighed by the symbiosis (iTCP=ON, Symbiosis=ON).

[MPEG00] rate control at the target rate of 2 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.

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

### *5.5 Symbiotic Rate Control:*

Fig-5 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 12 timeout events generated by six bursts) at the TCP resulted in the symbiosis unit to modify the rate according to the lazy-binary-back-off rule. A retraction ratio (A) of 0.55 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.

### 1.1 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 326 of the frames of this video at both sending and receiving ends. For a given discard threshold time in the receiving end 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 and transport aware cases. In Fig-6(a) we plot the delay experienced by the video frames in terms of frame arrival interval. To avoid any impact of the frame coding and generation delay, we let the transcoder first run in local mode and output its stream to a local disk as soon as it is produced. We then computed the frame delivery time with respect to the frame interval delay in local mode transcoding. Fig-6(a) now plots the frame arrival time for classical TCP (iTCP=off), and three cases of interactive TCP (iTCP, A=.35,.55 and .75). For comparison in the graph we also show the ideal expected frame delivery time based on linear generation rate. As can be seen the iTCP outperformed the classical TCP. The figure also shows the events times (for the case of A=.35). As it can be shown after each congestion burst, TCP continuously fell behind. The delay built up and hardly it could recover. This is shown by the step jumps in the delay line. The iTCP also suffered some step buildup, but it was much smaller and it could recover after few seconds.

In Fig-6(b) we plotted the jitter experienced by the frames. We took the difference between the expected ideal arrival time and the actual arrival time for each frame. A positive jitter means the frame arrived earlier than expected. As shown the iTCP also drastically reduced the jittery behavior.

### *5.6 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 let the player wait for a grace period. After that it considered the frame as lost frame and took application level measures to extrapolate the frame. Fig-6(c) plots the dramatic difference between the performances of the two mechanisms. It plots the number of delivered frames (y-axis) with various discard thresholds (x-axis). The top three curves show the frame loss for the transport unaware channel and the transport aware channel for the three retraction cases. As can be seen, in the cases where transport unaware scheme lost 75% frames, interactive scheme could send all of them in time after 100 frame delay threshold

### 5.7 *Observation at Application Level:*

In the above two experiments we illustrated how the symbiosis mechanism worked from the video transport protocol (MPEG-2) and the network transport protocol (TCP) layers beneath it. In this 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 both ends. The underlying MPEG-2 transport protocol and the network layer TCP together provides the transport. The specific compression, windowing etc. and other detail mechanisms are external techniques to the end systems.

In Fig-7 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 22-29 dB, but many of these frames were lost in transport, and were never delivered. In contrast, the proposed TCP interactive can deliver all the frames with 15-17 delay guaranteed at 15-26 dB quality[3]. This plots shows the Y block quality. Table-1 shows the average for all Y, U, and V blocks. 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.

### 5.8 *Overhead of Event Service:*

The dramatic advantage in application level performance came at a cost since the event tracking mechanism added some overhead. We were also curious to find out the overhead of the event mechanism. To track the overhead, we recorded the total data transmission time under the following three conditions. First, we let the transcoder send the video over the classical TCP implementation of our BSD system. The left most bar of Fig-8 plots the transport time for the three congestion levels. To observe the overhead of the event trapping service, in second run we activated the iTCP implementation however, we stopped the symbiotic reduction so the transport layer handled the same amount of data. As can be noted the event trapping service itself added very little overhead. However, in the third run we activated the event delivery and symbiosis. As can be seen, the delay dropped now. This is because although there were slight increase in the event trapping and delivery overhead, but it was vastly offset by the application level symbiosis technique. The advantage the application gained from the event delivery was much bigger than the overhead.

## 6 CONCLUSIONS AND CURRENT WORK

The proposed *principle of protocol interactivity* via the proposed T-ware mechanism can enable fundamentally new and interesting 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 elastic video. Also, it can be used to augment interesting features to current transport services.

T-ware is not a general case of network middle layer. T-ware is a specific investigation that focuses on two critical aspects: (i) embedding application specific techniques with improved application/network interface and (ii) allow dynamic mode of adaptation while the data communication is in progress. This type of adaptation is not covered under setup time contract based QoS mechanisms. On the other hand the initial rates (such as $C_{max}$) can be negotiated by other contract based QoS schemes.

---

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

The importance of these two factors can be emphasized by the fact that many transport impairments such as fault, or congestion are practically dynamic. Also, in many cases even applications may not know in advance what is its' capacity requirements. The proposed interactivity is not an alternate to other network level schemes, rather is a complimentary scheme.

The augmentation of the notification feature increases the normal mode delay of TCP slightly. The actual cost depends on the intensity of coupling. Designer of application symbiosis unit must be aware of the potential cost of tight coupling between T-ware 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 safeguard of this scheme is that a wrong design will only affect the application at fault and will have no effect on the network or on others.

The work has been supported by the DARPA Research Grant F30602-99-1-0515.

## 7 REFERENCES

[ABCS00] D. Andersen, D. Bansal, D. Curtis, S. Seshan, and H. Balakrishnan, System Support for Bandwidth Management and Content Adaptation in Internet Applications, Proc. of the 4th 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/openarch98.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, Procs. of the 8th Intl. Symp. on Modeling, Analysis and Simulation of Computer and Telecomm 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

[KhZu03][ Javed Khan and Raid Zaghal, "Performance Analysis of TCP Interactive based World Wide Video Streaming over ABone: Symbiotic Control and Video Quality", Technical Report TR2003-02-04, Kent State University, [available at URL http://medianet.kent. edu/ technicalreports.html] March 2003.

[KhGZ02] Javed I. Khan, Qiong Gu and Raid Zaghal, Symbiotic Video Streaming by Transport Feedback based Quality rate Selection, Proceedings of the 12th IEEE International Packet Video Workshop 2002, Pittsburg, PA, April 2002, http://www.pv2002.org (electronic proceeding)

[KHFH96] Keesman, Gertjan, Hellinghuizen, Robert Hoeksema, Fokke Heideman, Geert, Transcoding of MPEG bitstreams Signal Processing: Image Communication, Volume:8, Issue: 6, pp.481-500, Sep. 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://www.medianet.kent.edu/, also mirrored at http:// bristi.facnet.mcs. kent.edu/medianet]

[KhGR02] Javed I. Khan, Qiong Gu and Raid Zaghal, Symbiotic Video Streaming by Transport Feedback based quality rate selection, Proceedings of the 12th IEEE International Packet Video Workshop 2002, Pittsburg, PA, April 2002, http://www.pv2002.org .

[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, 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 4th 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, Proc. of the 16th ACM symposium on Principles of distributed computing, 1997, pp.129- 138

[Rama00] Suchitra Raman, A Framework for Interactive Multicast Data Transport in the Internet, Ph.D. thesis, University of California, Berkeley, May 2000