Proceedings of the 4th International Conference on Wireless and Optical Communication, WOC03,
July, 2003, Banff, Canada. Editor: Lambertus Hesselink, ISBN: 0-88986-374-1 (383) (unformatted version)

DYNAMIC QOS ADAPTATION FOR TIME SENSITIVE 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

ABSTRACT

The paper presents the concept of transientware—a mechanism by which sensitive applications can launch targeted, and dynamically adaptive enhancements of transport service. The transientware mechanism can be used with current transport protocols with interactivity enhancement and can be very useful for internet multimedia applications We have recently implemented the system on FreeBSD. In this paper we discuss the idea of T-ware and show its superior performance in time-bounded video delivery of elastic traffic from real experiment.

Keywords-- netcentric applications, transcoding, TCP Interactive.

1. Introduction

In this paper we discuss a novel 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 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 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. 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). T-ware based adaptation is particularly attractive for asymmetric network environment such as wireless and adhoc networks where the available and the quality of transport service and links can vary dynamically over a wide range.

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 problem to which we show a smart solution via T-ware is quite challenging. Coping with the dynamically varying link characteristics is an open problem in network applications. 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. A particularly noteworthy development along this direction is the new TCP friendly paradigm [SiWo98, Rama00, KeWi00, ReHE00, PrCN00]. The suggestion is to use more application level feedback. For example [ReHE00] discussed a framework where applications can control rates based on their end-to-end measurements. Also several works further investigated combining many application streams fro a single host and their specific information into one clearinghouse architecture, for aggregated congestion control. An example is Congestion Manager [ABCS00, BaRS99] which tries to minimize congestion by coordination between multiple sending streams and adding a middle layer.

2. T-ware Approach

In this context, the T-ware 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.

It is important to note that any sufficiently advanced adaptation technique (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 these 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. The attempt to contain them in a permanent system layer as a perfect protocol module can be hopelessly deceptive. Indeed the middle layer approaches seem to be failing to attract follower because of their relatively high

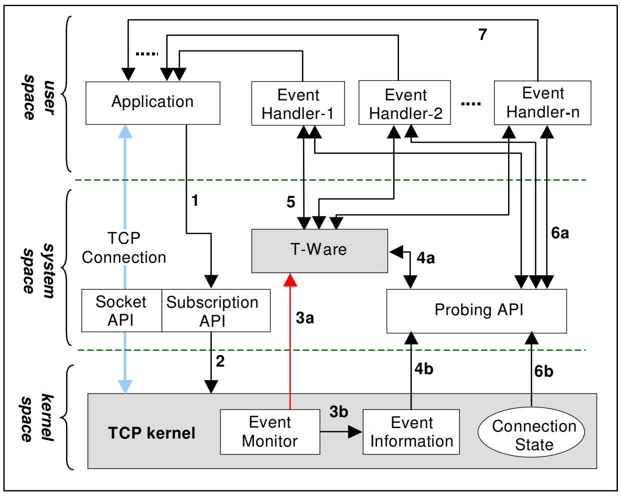

Figure-1. The TCP-interactive extension. The added registration API allows demanding applications to subscribe to events and probe additional event data.

complexity and bloatedness compared to their effectiveness.

This work tests a hypothesis if it can be performed in the other way. Let the actual transformation be performed at the application level with disposable yet highly targeted 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?

What is the critical system requirement towards such solution? A critical obstacle in current implementations of transport protocols is that these do not have any convenient mechanism by which dynamic states can be communicated back. We suggest 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 and a symbiotic T-ware that binds them. The 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.

In this paper, in the next section, we first provide the overview of the T-ware symbiosis system and the design of the interactive transport. In section 4 we then present the MPEG-2 transcoder and in particular the symbiotic rate control mechanism-- the T-ware component that provides the key network aware solution. Finally, in section 5 we share some performance of the scheme.

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.

3. System Overview

The application system has been developed as a three-part system -- *server, transcoder* and *the client*. The middle component *transcoder* [KhGu01] 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 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 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.

3.1 Interactive Transport Control

3.1.1 Architecture

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

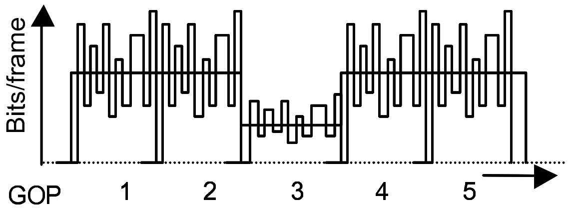

Fig 2(a) without carryover

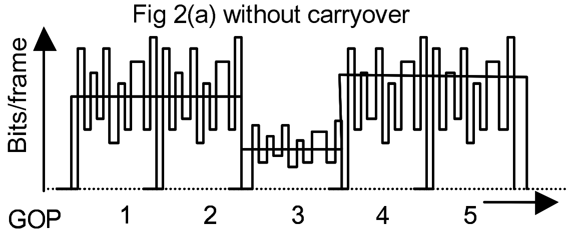

Fig 2(b) with carryover

Figure-2. Transcoder frame-wise bit generation with and without carryover.

TCP, interactive transport layer, when there is an internal timer-out event, passes on the current window resize event to the subscriber layer. The interface allows an adaptive application to optionally bind a T-ware routine to the designated socket event. When the event occurs, the TCP triggers the T-ware. If application chooses not to bind any T-ware the system defaults to the silent mode identical to TCP classic.

The internal architecture of the TCP interactive is shown in Fig-1. 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 T-ware 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, T-wares can probe additional event data via signal handler/ and additional API (4,5,6,7). If event data is subscribed then 3b occurs concurrently with 3a.

3.1.2 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.

3.1.3 Compatibility & Interoperability

TCP Interactive protocol retains 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. First it remains fully operational with all other currently deployed TCP remote-hosts. Only the server/transport hosts which are interested to run adaptive T-ware can locally upgrade to the extended model. Secondly, its *state-transition* behavior also remains identical to that of TCP classic. 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.

4. Application Architecture

4.1 Classical Modules: Transcoder

The transcoder unit has a decoder, and a re-encoder. Classical transcoder accepts a target rate as input and generates the bit-stream at specific rate. We have enhanced it to become capable of working in two modes: *normal* mode and *frugal* mode to accommodate requirements in two rate ranges. It is shown in Fig-2. In frugal mode the rate can be controlled at frame level. The particular rate control techniques applied in the transcoder makes complex provisions for frame type, perceptual activity etc. and are based on complex theoretical consideration. The details are not relevant for this paper but can be found in [KhGR02]. Essentially we have enhanced the TM-5 model so that two sets of bit accounting techniques operate in two modes. In *normal mode* it operates as MPEG-2 TM-5 model. In *frugal mode* 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.

4.2 Symbiotic Rate Determination

The new element in the application system is the T-ware unit—*symbiosis controller*. This is the unit which is invoked by the iTCP on event(s) and generates the target rate for the rate controller. In the test system it follows a lazy binary-back-off symbiotic model using a two variable min/max mechanism. Let the target bit rate during normal mode generation is given by C_{max}, and the minimum acceptable rate for application is given by C_{min} 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. 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

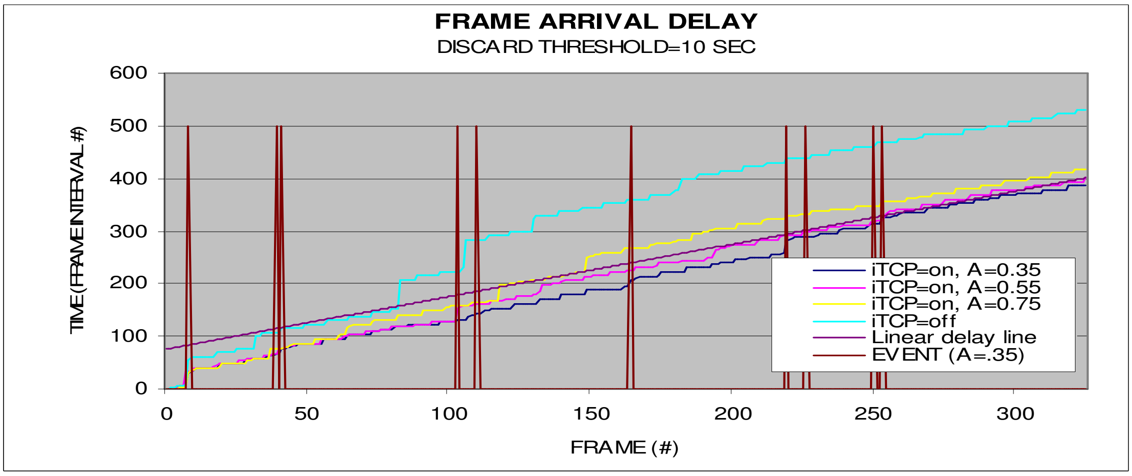

Fig-3. The arrival times of the frames. For ideal case it should be linear. However, with each timeout event the backlog increased (observed as the step jumps in delay). The T-ware helps in eliminating these step increase in delays. The spikes are the events for A=.35 experiment. Note the correlation between the spikes and the jumps in delay.

on precise quality/ delay tradeoff boundaries of the video. Based on the tolerance we define a ratio called *rate retraction ratio:*

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

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

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

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

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

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

We have implemented a full T-ware system with (i) an MPEG-2 DTV symbiotic video rate transcoder system (ii) a TCP Interactive transport on a FreeBSD and the (iii) lazy binary back-off symbiotic T-ware. For experiment we placed a transcoder on an intermediate node between the server and the client. Between the transcoder and the client we have also placed a congestion injector. We modeled the congestion as a sequence of bursts with various frequencies. We used bursts with 3 seconds duration and random inter-burst time. 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.

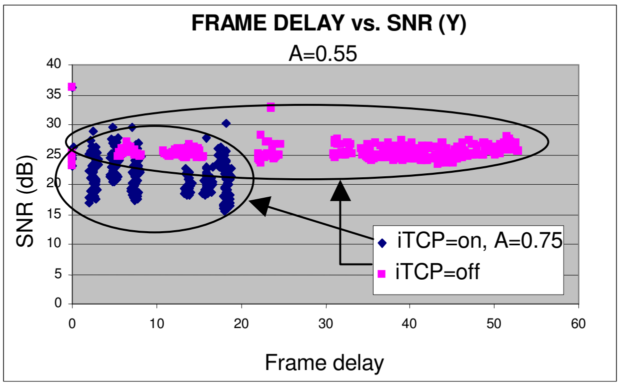

Fig-4. The two clusters show the quality, delay tradeoff offered by the iTCP. The iTCP dramatically reduced frame delivery delay by controlled trade-off of the SNR quality.

5.1 Transport Delay and Jitter

The first experiment is shown in in Fig-3. It plots the delay experienced by the video frames in terms of frame arrival interval with and without T-ware. It plots the frame arrival time for both 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. It also shows the congestion bursts. 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 and backlog of data grew in sending end buffer. The resulted in delay built up and hardly it could recover. This is evident in the step jumps in the delay line. As evident the iTCP also suffered some step buildup, but it was dramatically smaller and it could recover after few seconds. This dramatically improved the jitter and timely delivery of traffic.

5.2 Application Level Quality of Service

In Fig-4 now 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. This plots shows the Y block quality. Fundamentally, what T-ware solution 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.

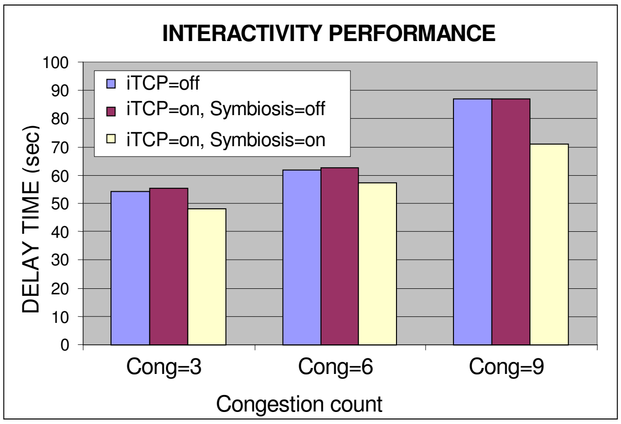

Fig-5. Compared to the overall transport time without interactivity, there is a small overhead associated with iTCP event trapping mechanism (iTCP=on, Sym=off). However, this small overhead is vastly outweighed by the symbiosis (iTCP=on, Sym=on).

5.3 Overhead of T-ware

What is the cost of T-ware invocation? We were 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-5 plots the transport time for various intensities of congestion events. 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 needed by the t-ware to operate 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

T-ware mechanism can enable fundamentally new and interesting solutions to many of today's hard to tackle problems. It can also be used to augment interesting features to current transport services.

T-ware is not a general case of network middle layer. T-ware focuses on two critical aspects: (i) embedding application specific techniques and (ii) dynamic adaptation while the data communication is in progress. The importance of these two factors can be emphasized by the fact that many transport impairments such as fault, or congestion are practically dynamic. Almost by definition these require runtime intervention. For example, reservation based QoS provisioning schemes will not address this problem. What if despite the contract, the situation changes at run time due to change either in network or at application? Some form of dynamic mechanism such as T-ware will be needed to handle dynamic provisioning. T-ware is particularly attractive in wireless and adhoc networks where the available and the quality of the physical layer regularly fluctuate drastically.

What are the potential diverse impacts of T-ware? Security can be one. T-ware is a potential solution which should allow advanced application designers to add temporary system level components in a controlled manner. Notably, T-ware itself cannot directly modify the behavior of the transport protocol. Thus any potential threat is passive. Another concern might be coupling. The designer of T-ware units must be aware of the potential cost of tight coupling. 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 network or others.

The research has been supported by the current 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,Oct 23-25, 2000 San Diego, CA.

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

[KeWi00] Jun Ke and Carey Williamson / University of Saskatchewan, Towards a Rate-Based TCP Protocol for the Web, Proceedings of the 8th Intl. Symposium on Modeling, Analysis and Simulat-ion 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 NCA' 2001, Oct 2001, Cambridge, MA, pp.201-213.

[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.

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

[ReHE00] Reza Rejaie, Mark Handley, Deborah Estrin, Architectural Considerations for Playback of Quality Adaptive Video over the Internet, Procee-dings of the IEEE ICON 2000.

[SiWo98] Dorgham Sisalem, Adam Wolisz, Towards TCP-Friendly Adaptive Multimedia Applications Based on RTP, Proceedings of the The 4th IEEE Symp-osium on Computers and Communications, 1998.

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