Towards an Adaptive Transport Protocol for Efficient Media Delivery over Wireless Networks

Sumathi Gopal, Dr. Dipankar Raychaudhuri, WINLAB, Rutgers University

I. Introduction:

The past decade has seen tremendous research, development and usage of local and wide area wireless networks. There is now a high penetration of cell-phone usage along with the evolution of powerful new mobile platforms, smaller wireless devices, and micro-browsers for small handhelds. Many important wireless data communication standards such as 802.11 for LANs and GPRS, 3G for WANs have also materialized. With this accelerated evolution of high-speed wireless data networks, the next generation "killer applications" are expected to be multimedia based.

But supporting multimedia services particularly video delivery over wireless networks, is distinctively challenging due to constraints such as random time-varying network interference, stringent QoS requirements of media and limited battery power of mobile devices. To maintain video quality, strict upper bounds on bulk end-to-end delay and jitter need to be met. Video delivery also requires fairly high bandwidth availability. Wireless LANs in vogue today – the IEEE 802.11 networks, do have bandwidth availability but are characterized by bursty errors, caused by dynamic variations in channel conditions with different types of fades.

Real-time Transport Protocol (RTP) [18], is the most widely used of media delivery protocols over packet networks and is highly optimized for the wired Internet. It does make provision for feedback from the receiver in the form of RTCP, but in order to handle it, the application needs to implement rate control itself. Despite this application-level complexity, this approach has gained much prominence, because of the lack of a transport protocol that can do so.

The two transport protocols in vogue today, TCP and UDP [1] are two extremes in a spectrum of possibilities. TCP fashions a reliable channel for higher layers, with its rate control, error control and congestion control mechanisms highly optimized to suit the wired Internet. UDP on the contrary supports very basic transport functionality, with no error, rate or flow control. Media applications typically use the UDP protocol and handle rate and flow control mechanisms on their own and of late in conjunction with RTP/RTCP [18]. This approach to solving media-related problems is thus a major impediment to new media applications, with them having to handle intricacies of the transport layer functionality. There is thus a clear need for a new transport protocol that can dynamically adapt to the reliability and flow control needs of the overlying applications. The need is further emphasized with wireless networks, with their wide variation in channel characteristics and bursty errors.

The transport protocols in use today are mostly flat, a result of independent layers and modularity. Modularity is popular because of the flexibility it offers for the choice of protocols in each layer, with well-defined service points between adjacent layers. But it also results in disadvantages such as redundancies in the communication architecture, enough to degrade the performance of the system. For example, [20] reports an 'incompatibility' problem among TCP connections in an IEEE 802.11b multi-hop network caused due to TCP timeouts during 802.11b MAC layer retransmissions. This problem can be easily avoided if some form of 'information-exchange' is incorporated between the two layers. We hence believe that to some extent, inter-layer adaptation and optimization will significantly improve efficiency of data delivery over wireless networks. With delay and jitter sensitive multimedia traffic, inter-layer optimization becomes all the more important.

II. Inter-layer optimization:

As a first step towards optimizing the communication architecture, we propose to evolve a control parameter plane that would exist in parallel to the data plane. This control plane will contain control parameters from all layers in the communication stack. Below is a list of all possible parameters that can be extracted from the various layers [2] [3].

Physical layer:

1. RSSI (Received Signal to Sender Input);
2. Available maximum bit rate(depends on the modulation technique used)
3. Bit Error Probability (possibly derived from previous bit error rates)
4. Length of a slot time (aSlotTime parameter in 802.11b)

MAC/Data Link Control layer:

1. Bit Error Rate,
2. Frame loss rate (frames ACKed/frames sent),
3. Effective throughput (over both short and long durations),
4. Variance in throughput
5. Variance in bit error rate
6. Link congestion indicator
7. Maximum frame size
8. Frame overhead (as a percentage value)
9. Retransmission attempt rate
10. Frame error rate
11. Congestion Indicator

Network layer:

1. Routing cost (may be quantified using round-trip-time/bulk delay/ number of hops to destination host etc)
2. Delay jitter (variation of delay between arriving packets irrespective of order of transmission)

Transport layer:

1. Missing packet rate
2. Packet out of order rate
3. Packet retransmission rate
4. Application layer packet size
5. Maximum data rate handled by the receiver
6. Effective network data rate
7. Maximum network transmission unit
8. Transport layer delay overhead
9. Round-trip-time average
10. Round-trip-time variance
11. Fragmentation rate
12. End-to-end bulk delay
13. End-to-end Packet Jitter
14. End-to-end hop count
15. *Application specified parameters if any*

The layers above the transport layer are application specific and hence we will not consider those parameters in our design.

III. Proposed Research:

The Basic functions of a transport protocol are

a. Multiplex packets sent by multiple processes onto a single byte channel and

b. Demultiplex packets at receiving end and deliver them to waiting processes

The UDP protocol provides this basic functionality. Improved media delivery performance is achieved by providing additional adaptive features.

We propose to develop a transport protocol that will:

1. Accept service specification parameters from higher layers (applications) (QoS requirements). These specifications may be:
    a. Tolerable Packet Loss rate
    b. Tolerable end-to-end bulk delay
    c. Maximum delay jitter
2. Utilize a suitable conversion algorithm to translate application-specifications to suitable error-control, flow-control and congestion-control mechanisms.
3. Carryout a handshake procedure – equivalent of *Connection Establishment* in TCP, to inform and negotiate if necessary, with the peer transport layer at the destination about the service requirements of the particular flow.
4. Periodically assess network condition based on control parameters available from various underlying layers and itself. The result of this assessment may be to classify the channel as "good", "intermediate" or "bad". We should be careful to not makes these decisions over very short time-frames

The transport protocol should be able to use this information to calculate upper and lower bounds on available throughput, delay jitter and other such network characteristics, which affect the quality of media traffic.

5. Adapt to the channel condition by tweaking 'adjustable knobs' such as On-Off ARQ, TPDU size etc.
6. Access its own performance in the course of data transfer, to determine if its meeting the application specifications.
7. Report this status to the application, so that in case needed, the application can re-specify less-stringent parameters, so that the transport protocol can meet it.
8. Dynamically adapt to new application specifications during the course of data transfer.

This new transport protocol will thus incorporates inter-layer optimization, to improve overall transmission efficiency of media traffic over wireless networks. With this transport protocol, media applications can be oblivious to mechanisms of rate control, error control etc and can concentrate perfecting their own functionality.

The next section details the literature survey carried out to gain insight into the design of this desired transport protocol.

IV. Literature Survey:

An exhaustive literature survey of topics relevant to transport protocols for media delivery and wireless networks, revealed a broad variety of approaches to solving various challenges in the field. Transport protocols proposed for wireless networks are mostly extensions of TCP to cater to applications that require reliable data service [8] [10] [21]. Transport protocols for media delivery are adaptive in nature, in that the application can specify requirements at the start of transmission, and are primarily geared towards the wired Internet [13]. Some adaptive transport protocols proposed are also ATM where QoS can be supported more easily [12]. Some try to solve the problem of 'delayed deployment and standardization of new protocols' by taking a generalized approach. Protocol Boosters [7] and ADAPTIVE [5] are some such that provide for dynamic protocol extension for new functionality by adding new software/hardware modules.

The POC service [13] and Generic Transport Protocols [17] describe research closely related to our proposed transport protocol. [13] elaborates on a new transport protocol that accepts service specifications from the overlying application at the beginning of a session and adapts likewise. The service specification is in the form of the acceptable out-of-order behavior and loss for each packet. [17] proposes a new family of Generic Transport Protocols (GTP) that go one step further and accept even temporal specification from applications, i.e. maximum acceptable delay for each packet.

Below is a more detailed description of different research work (GTP, POC service, ADAPTIVE, Protocol Boosters, HPF, I-TCP, WTCP) from which we may draw significant insight.

POC transport protocol for improved performance of DMS systems

This paper [13] proposes the *POC Transport Architecture* to provide for the stringent communication service requirements of Distributed Multimedia Systems (DMS). Distributed Multimedia Systems (DMS) are a natural outcome of the explosive growth in multimedia computing, communications and applications. But transport protocols in vogue today are not geared towards supporting these systems. They do not cater to the quality of service needs of the media applications that operate these systems.

The Partial Order and partial reliability Connection (POC) proposed in this paper, provides a service authorized to support both bounded losses and bounded disorder, which are key to media applications. POC supports QoS specification by higher layers (applications) in terms of reliability ***r*** and order ***o***, enabling them to specify acceptable ranges of parameters ***r*** and ***o***. For a given service specification $S = (O, R)$, the order dimension O represents a set of all *acceptable* orders for a given set of objects. The reliability dimension R, represents the subset of objects that *must* be delivered by the service S. Thus the compliment of R represents the set of objects that are authorized to be lost.

An overlying application is enabled to specify it's the communication requirements, i.e. the POC service, using the TSPN (Time Stream Petri Net) model [14], which provides the framework for applications to specify the desired order, reliability (acceptable losses) and temporal behavior. Retransmissions are made for those objects whose loss is not permissible.

An MPEG-I video server is used to test the protocol, as it is one of those applications that require a continuous service but can tolerate losses and disordered packets. Firstly, the elements in the video sequence whose loss or out-of-sequence-delivery is impermissible are identified (such as I-Frame or an entire Group-Of-Pictures (GOP)). The authors infer out of order permissibility for MPEG-I video from [16].

Then the POC transport service for MPEG-I video is specified in two phases – the definition of Transport Service Data units (TSDUs) and the specification of the POC service using TSPN formalism. The former step is required in order to enable the user process (in this case, the MPEG-I decoder) to optimally manage losses and disordered information. These TSDUs differ from application to application depending on their QoS requirements. In this case, two types of TSDUs – TSDUs involving *slices* (parts of an MPEG *frame*) and TSDUs involving *headers* - are created, as they have separate reliability and order constraints. The POC service for each type of TSDU is separately specified.

Comparison between performance of the application while using each of TCP/IP, POC and UDP/IP, indicated POC to have a far better than either of the other two, in terms of received video quality (delay jitter, and packet loss). Of course, here the MPEG decoder at the receiver end was specifically designed to handle losses and out-of-order delivery of packets.

Their website http://dmi.ensica.fr/poc contains demonstrations of the video quality improvement using of POC instead of plain UDP.

GTP: A set of Generic Transport Protocols

This work [17] extends POC service transport protocol, generalizing for any type of application. The paper begins by identifying that there is a clear lack of transport protocols that offer service intermediate to the extreme services offered by TCP (fully reliable and ordered) and UDP (no reliability or order guarantees). GTP comprises of a new family of protocols that provide for an intermediate service while enabling applications to specify QoS parameters in the form of packet ordering, reliability and time constraints.

Applications are enabled to use a temporal extension of Hierarchical Time Streaming Petri Nets (HTSPN) model [19] to specify parameters to the transport layer. This approach differs from Application Level Framing (ALF) mechanism (RTP [18] is based on these principles) where the application needs to be aware of the intricacies of the communication procedure. GTP protocols instead consider QoS parameters specified by the application and adapt with changing network conditions to meet application requirements. GTP thus adds an additional feature of temporal constraint specification by applications over POC [15]. The paper describes in detail the parameter specification and delivery procedures, which is out of scope here.

They thus provide a weakly synchronous transport service (TPOC) to multimedia applications that delivers multimedia information units according to time related QoS parameters derived from application level requirements. This greatly aids in the reduction of complexity of distributed multimedia applications while significantly improving the use of network and communication resources.

GTP was first implemented in the Java programming language, to augment the framework of the 5th IST European project GCAP (Global Communication Architecture and Protocols for new QoS services over Ipv6 networks), which aims to develop improved end-to-end multicast and multimedia transport protocols. It uses a *pull* approach where the receiver initiates the connection and termination, while the sender awaits a request. The paper mentions couple of experiments to demonstrate the advantage of GTP over either of TCP or UDP. The first experiment transmits JPEG images and compares end-to-end delay as seen with UDP, TCP* (GTP's emulation of TCP) and GTP with a full-reliable-no-order service specification. In the second experiment, performances of TCP* and GTP with 70%-reliability-specification are compared by transmitting an MJPEG video stream with 411 frames. Both experiments demonstrated improved performance with the use of GTP's partial reliability and ordering features.

The ADAPTIVE Framework [4]

ADAPTIVE is a proposed mechanism to overcome the *Throughput preservation problem* that occurs due to a disproportionate advancement in VLSI-Fiber Optic technologies and the rest of the communication architecture. Thus drastic improvement in *channe*l throughput, has not been met by a significant gain in *system* throughput, primarily due to the transport system bottleneck. Transport system factors that add to the transmission overhead include: (1) process management (eg., context switching, synchronization, scheduling overhead etc) (2) Message management (memory-to-memory copying, dynamic buffer allocation etc) (3) Multiplexing and demultiplexing (4) protocol processing tasks (such as checksumming, segmentation, retransmission timer, flow control, connection management etc) (5) network interface hardware.

ADAPTIVE (an acronym for "A Dynamically Assembled Protocol Transformation, Integration and evaluation Environment") provides a flexible framework to develop and experiment with alternate process architectures. A process architecture binds communication protocol entities to logical and/or physical processing elements. Protocol entities include abstractions such as layers, tasks, connections and/or messages. *The term 'process' is used here to mean a thread of control executing within a single address space.*

The architectures proposed in this paper are broadly divided into three general categories: *horizontal*, *vertical* and *hybrid*. *Horizontal* process architectures include *Layer Parallelism* that associate "process-per-protocol-task" (such as presentation layer, transport layer, network layer), and *Task Parallelism* that associate "process-per-protocol-task" (for example, flow control, error detection, routing etc). Vertical process architectures include *Connectional Parallelism* that associate "process-per-connection" and *Message parallelism* that associate "process-per-message". *Hybrid* process architectures may include a combination of one or more of the above-mentioned parallelisms.

Some process architectures may utilize certain system resources (say multiple CPUs) more effectively than others. The contrary might also happen. For example, certain process architectures may increase the overhead of inter-process communication and memory-to-memory copying, whereas others may increase the overhead of synchronization and/or context switching. In general, the suitability of a process architecture would depend on factors such as (1) the type of traffic generated by applications (2) the architecture of the hardware and operating system (such as message passing vs. shared memory) (3) the underlying network environment. Overall, these process architectures are said to significantly impact the performance of applications and transport systems.

It is claimed that ADAPTIVE's modularity also increases its portability, allowing it to run on multiple underlying kernel and protocol family architectures. The paper focuses on a version of ADAPTIVE that is hosted in the UNIX STREAMS environment [5]. It concludes by demonstrating the use of ADAPTIVE to implement and evaluate specific protocol machines customized for several classes of multimedia applications (such as real-time video applications), running over several different networks (Ethernet, FDDI etc).

Protocol Boosters [6]

This paper describes a methodology for protocol design that uses incremental construction of the communication protocol from elements called "protocol boosters" on an as-needed basis. They were evolved for the Internet, but may be adapted to suit wireless networks also. A protocol booster, which may be a software or hardware module, is designed to enable dynamic protocol customization to heterogeneous environments and rapid protocol evolution. It is a supporting agent and is not by itself a protocol. It is also transparent to the protocol being boosted. The end-to-end protocol messages are not modified and the functionality of the end-to-end protocol is not replicated.

The paper provides examples of error and congestion control boosters for UDP and also demonstrates initial results from booster implementations. Sample implementations include a Two-element Jitter-Control Booster for IP, One-Element Error Detection Booster for UDP, etc. Advantages of protocol boosters include their independence from standardization, and need for minimum resources for design. They are said to be a free market approach to protocol and network design. Further, the paper compares and contrasts Protocol Boosters with other architectural alternatives. It claims that ONLY the boosters take advantage of higher layer information (unlike in the link-layer-adaptation approach), do not alter message syntax (unlike protocol conversion, as seen in Van Jacobson's TCP header compression [10]) and do not modify semantics of the protocol (protocol termination as in I-TCP [7]).

HPF: An adaptive transport protocol for multimedia communication

The paper [12] proposes a transport protocol called HPF that can improve multimedia communication over the Internet. It identifies the following requirements of multimedia traffic to be of primary importance:

1. Priority levels (I-frames more important than B-frames, audio over video etc) for guarantee of service during congestion.
2. Each flow may contain multiple interleaved sub-streams; each having separate needs with respect to reliability, sequencing and timeliness.

The protocol is designed to provide for these requirements by supporting heterogeneous packet flows consisting of interleaved sub-steams with different needs (level of reliability etc). Applications are enabled to tag data with a priority level, so that intermediate nodes can discard low-priority packets in favor of higher ones in a given stream when congestion occurs. Of course, this implies modification of the link-scheduler in intermediate nodes.

Congestion control and reliability mechanisms are decoupled so that the two questions "how many packets can be transmitted next" and "which packet needs to be transmitted next" can be answered independently. In order to achieve this decoupling, end-to-end congestion control is based on feedback from the receiver about the fraction of received packets in the current window, rather than the sequence number of the last received ACK.

The paper indicates HPF performance measurements in their experimental testbed, to have provided effective support for heterogeneous packet flows in the presence of dynamic networking resources.

Indirect-TCP (I-TCP) for Mobile hosts:

Bakre and Badrinath [7] proposed this protocol to alleviate problems such as slow and unreliable links, faced by Mobile Hosts (MH) while trying to exchange data with a fixed host (FH). The Indirect-TCP is built to utilize the fact that the link between the MH and FH can be clearly separated into wired and wireless parts, with a Mobile Service Router (MSR) as the center point. The wired and wireless parts differ immensely with respect to reliability and I-TCP tries to improve overall throughput under different situations, by tackling the problems in the wireless link.

When a MH wishes to communicate with some FH using I-TCP, it sends a request to its current MSR, to open a connection with the FH on *behalf* of the MH. The transport protocol between the MH and MSR is a variation of TCP that is tuned for wireless links and is mobility-aware. On receiving such a request, the MSR establishes a new connection to the FH using the same IP address and port number used by the MH to open a connection to it. All data packets between FH and MH flow via the MSR.

The FH sees only an *image* of its peer MH that it interprets as residing on the MSR.

When the MH moves to another cell during the lifetime of the current I-TCP connection, the MSR hands over all ongoing TCP connections and their states of the MH, to the new MSR. The new MSR creates the two sockets corresponding to the I-TCP connection with the *same* endpoint parameters that the sockets at the previous MSR had associated with. During this handoff period, the old MSR stores packets destined for the MH and forwards them to the new MSR once the handoff is complete. The new MSR in turn forwards them to the MH.

Thus with no change in the endpoint parameters, the FH is oblivious to the indirection at the central MSR and the handoffs that occur. Some specific changes are made to the IP input routine in the MSR to enable movement of the connection states. Mobility support is made possible by careful implementation of I-TCP state handoff at the MSR. As can be inferred, with I-TCP, acknowledgements are not end-to-end but instead there are separate acknowledgements for the wired and wireless parts of the connection. Thus the overall throughput is better than regular TCP, only if there are no MSR failures, and the handoff period is small.

Modifications to the transport protocol are made only on the MH and MSR nodes, thus confining any changes only to the wireless link. No changes are made to the hosts on the fixed network, thus making the protocol scalable. The I-TCP protocol was found to yield significant throughput improvement over that of TCP when there was mobility between both overlapping and non-overlapping cells. Gain in throughput was better in WANs than in LANs.

Wireless-TCP:

This work does not contribute directly to our research topic, but does offer some important clues. It brings out the flaws of TCP for transmission over wireless networks. For example, it explains the reason why retransmission timeouts are detrimental to overall throughput. We may also utilize their approach for congestion and rate control in our own experiments.

WTCP [10] is designed for Wireless-WAN networks, particularly the CDPD network that was in existence a few years ago. WTCP attempts to provide a reliable transport service while trying to overcome in-built network problems such as long blackouts (due to long fading), very low bandwidth, high latency etc. The CDPD network typically has channel capacity not more than 12 Kbps and suffers from high delays (800 ms to 4 sec) and latency. The paper argues that the TCP's congestion control and rate control mechanisms are inappropriate for the WWAN, and hence contemporary approaches that try to improve TCP throughput by either improving reliability at the link layer, providing TCP-aware smarts in the base station or splitting the TCP connection into two parts [8], do not solve the root problem. Instead they propose new schemes to provide rate control and congestion control separately. WTCP distinguishes the cause of packet loss and adjusts transmission rates accordingly. The paper outlines reasons for the failure of TCP in W-WANs due to some of TCP's mechanisms not being compatible to CDPD's channel arbitration methodology. To overcome problems faced by TCP, WTCP employs rate-based congestion control instead of TCP's window-based congestion control and Inter-packet delay to determine rate adaptation.

For *rate control*: The receiver maintains a record of previous rates requested and the corresponding adaptation by the sender, and performs the rate adaptation computation. In this process, it uses information that the sender sends in every data packet, such as last received ACK sequence number, sending rate etc. In every ACK, the receiver requests a new calculated data rate and the sender adapts likewise.

The receiver calculates the most appropriate sending rate at that time, considering both long-term and short-term ratios of observed sending rate at the receiver to the actual sending rate at the sender.

For *reliability*: WTCP implements *Selective Acknowledgements*, by which the sender can set frequency of ACKs and the receiver sends cumulative and selective ACKs likewise. The sender decides the frequency based on perceived conditions such as observed ACK loss at the sender, half-duplex or full-duplex nature of the WWAN channel and average deviation in the inter-ACK separation observed at the sender. It further tunes the ACK frequency such that it receives at least one ACK in a threshold period of time. Further, WTCP does not use *retransmission timeouts (RTO)*, considering the deterioration of TCP performance caused by erroneous RTO estimation. This is achieved by use of *Probe* packets that are sent if the sender receives no ACKs for a threshold period of time. These packets are meant to elicit ACKs from the receiver and recover from the blackout. The *Probe packet mechanism* is also used for loss recovery, eliminating the need for timeout-based retransmissions in WTCP.

WTCP suffers from the disadvantage that, in order to deploy the protocol, changes need to be made even to fixed end hosts. This is a humongous task considering the widespread use of TCP. But the protocol addresses various problems faced by TCP in wireless networks. This paper lays a good foundation to our work on an Adaptive Transport protocol. Another observation is that the study uses a flat uniform error rate model for wireless channel errors. This is a simplistic assumption considering results obtained in "A Markov-based Channel Model Algorithm for wireless networks" [22].

Other publications

A continuous media and orchestration service [4] proposes a transport service that operates as an intermediate layer between the continuous media application and TCP/IP to provides synchronization services for multiple related media entities. This is a precursor to the Real-time transport protocol [18] that is often used by media applications today.

An *ATM based QoS adaptive system* in [11] is an adaptive transport layer called METS over the ATM infrastructure, which can adapt to varying QoS in the network and hide these fluctuations from the higher layers by the use of mechanisms such as flow scheduling, flow shaping and jitter filtering.

V. Conclusion and direction for future work

Some observations made regarding POC [13] and GTP protocols [17] deserve specific mention. They enable the application to specify communication requirements to the transport protocol using the TSPN and HTSPN models respectively [19]. The transport protocol schedules packet delivery based on the service requirements of each packet as defined by the application. But it does not have a mechanism to accept a modified specification for an existing transport session. This feature is desirable considering the dynamism of the wireless network. The only feedback from the receiver is about the receipt of a packet. There is no connection establishment procedure before start of data transmission. Such a handshaking procedure would indeed aid either end associating buffers and states to packet types. Overall, we may use this work as the basis for our research and develop further.

The prioritized scheme proposed by the HPF transport protocol [12] despite its efficient use of network resources, requires significant modification in network layer functionality of intermediate nodes.

The Protocol Boosters and ADAPTIVE algorithms define new protocol architectures to decrease delay overhead caused by not-so-efficient transport and other protocols in the communication stack. We may choose from these approaches in the design and implementation of dynamically loadable modules in our transport layer.

I-TCP deals with handling TCP connection states in wireless multi-hop networks. Our proposed transport protocol will also have connection states, although of a different nature and knowledge derived from I-TCP research might benefit. WTCP was designed for the low-bandwidth CDPD wireless network that is not even in existence today. The results derived in this work do not seem directly related to the wireless channels under consideration in our project.

VI. Bibliography

[1] Tanenbaum A. S., "Computer Networks", 3rd. ed., pages 479 – 476, Prentice Hall, NJ, USA 1996

[2] Tanenbaum A. S., "Computer Networks", 3rd. ed., chapters 2, 3, 4, 5, 6; Prentice Hall, NJ, USA 1996

[3] Bertsekas D., Gallager R., "Data Networks", 2nd Edition, Prentice Hall, 1991

[4] "A continuous Media Transport and Orchestration Service", Andrew Campbell, Geoff Coulson, Francisco García and David Hutchison, 1992.

[5] Schmidt D. C., Suda T., "ADAPTIVE: a Framework for Experimenting with High-Performance Transport System Process Architectures", Proceedings of the 2nd International Conference on Computer Communications and Networks, ISCA, San Diego, California, June 28-30, 1993, pp.1-8.

[6] D. Ritchie, "A Stream Input-Output System", AT&T Bell Labs Technical Journal, vol. 63, pp 311-324, Oct 1984.

[7] Bakin D.S., Feldmeier D.C., Marcus W.S., McAuley A.J., Raleigh T.M., Smith J.M., "Protocol Boosters", IEEE Journal on Selected Areas in Communication, April 1998

[8] Bakre A, Badrinath B.R., "Indirect TCP for mobile hosts", Proceedings of the 15th International Conference on Distributed Computing Systems, 30 May-2 Jun 1995

[9] V. Jacobson, "TCP/IP Compression for Low-Speed Serial Links." RFC 1144, February 1990.

[10] Sinha P., Venkitaraman N., Sivakumar R. and Bharghavan V., "WTCP: A Reliable Transport Protocol for Wireless Wide-Area Networks", Wireless Networks, March 2002, Volume 8 Issue 2/3.

[11] Campbell A.T., Coulson G., "A QoS Adaptive Transport System: Design, Implementation and Experience", Proceedings of the Fourth ACM Multimedia Conference (MULTIMEDIA'96). pp.117-128. New York, NY, USA. Nov 1996

[12] Bharghavan V., Dwyer D., Jia-Ru Li, Sungwon Ha, "An adaptive transport protocol for multimedia communication", Multimedia Computing and Systems, 1998. Proceedings. IEEE International Conference on , 28 Jun-1 Jul 1998

[13] Dairaine L., Diaz M., Rojas-Cardenas L.; Senac P., "An adaptive transport service for multimedia streams", IEEE International Conference Multimedia Computing and Systems, 1999., Volume: 2 , Jul 1999

[14] Diaz M., Drira K., Lozes A., Chassot C., "On the definition and representation of the quality of service for multimedia systems", 6th IFIP Internation Conference on High Performance networking (HPN'95), Spain, September 1995

[15] Amer, Chassot C., Connolly T. and Diaz M., "Partial order transport service for multimedia applications", IEEE/ACM transactions on networking, vol. 2, No. 5, October 1994

[16] Rojas-Cardenas L., Chaput E., Dairaine L., Senac P., Diaz M., "Video Transport Over Partial Order Connections", HIPPARCH 98, London June 1998

[17] Senac P., Exposito E., Diaz M., "Towards a New Generation of Generic Transport Protocols", Proceedings of Thyrrhenian International Workshop on Digital Communications (IWDC 2001), Taormina, Italy, September 17-20, 2001. http://www.laas.fr/GCAP/docs/publications/IDWCSenacExpositoDiaz.pdf

[18] Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V. RTP: A Transport Protocol for Real-Time Applications. InternetDraft, IETF, draft-ietf-avt-rtp-new-09.txt, March 2001.

[19] Senac P., De Saqui-Sannes P., Willrich R., " Hierarchial Time Stream Petri Net: a model for hypermedia systems" in Application and Theory of Petri Nets(IEEE), Lecture Notes in Computer Science, Springer Verlag Editor 1995; http://citeseer.nj.nec.com/snac95hierarchical.html

[20] Saadawi, T., Xu S., "Revealing TCP incompatibility problem in 802.11-based wireless multi-hop networks" , Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE , 2001, Page(s): 2847 -2851 vol.5

[21] Nahrstedt, K.; Servetto, S.D.; "Video streaming over the public Internet: multiple description codes and adaptive transport protocols", Proceedings. 1999 International Conference on Image Processing,1999. Volume: 3, 1999, Page(s): 85 -89 vol.3

[22] Konrad A., Zhao B. Y., Joseph A. D., Ludwig R., "A Markov-Based Channel Model Algorithm for Wireless Networks", CS Department, University of California, Berkeley; http://www.cs.berkeley.edu/~almudena/publications/MSWiM.pdf