International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 18 (2015) pp 39724-39730
© Research India Publications. http://www.ripublication.com

Optimization of Real-Time Video Over 3G Wireless Networks

A.S.Syed Navaz[1], **P.Jayalakshmi**[2], **N.Asha**[3]

[1]*Assistant Professor, Department of Computer Science, Muthayammal College of Arts & Science, Namakkal, India*
[2]*Assistant Professor Senior, SITE, VIT University, Vellore, India*
[3]*Assistant Professor, SITE, VIT University, Vellore, India*
a.s.syednawaz@gmail.com, pjayalakshmi@vit.ac.in, nasha@vit.ac.in

Abstract

The current 3G technologies provide many other services like video telecast, video conferencing, broadcasting TV programs, movies etc. This is achieved by feedback adaptation, whereby the media being sent is adapted in real time according to feedback information about the observed network state and application state. For the success of such adaptive schemes, the feedback must: 1) arrive in a timely manner and 2) carry enough information to effect useful adaptation. In this paper, we exploit the use of feedback adaptation for media streaming in 3G wireless networks, where the media servers are located in wired networks while the clients are wireless. The argument is that end-to-end feedback adaptation using only information provided by 3G standards is neither timely nor contain enough information for media adaptation at the server. The introduction of a streaming agent (SA) at the junction of the wired and wireless network provides useful information in a timely manner for media adaptation. Henceforth optimization algorithms can be designed to take advantage of SA feedbacks to improve performance. The improvement of SA feedbacks in peak signal-to-noise ratio is significant over non agent-based systems

Keywords: 3G wireless networks, multimedia communication, congestion control, WCDMA.

Introduction

In this mobile communication era, mobile phones are common and are mandatory for human survival. They are used for voice communication, messages, multimedia messages, browsing internet on move etc.

Even mobile service providers provide many value added service other than basic services such as weather forecasting, news, score updates etc. The goal of this paper is to improve real-time video transport over 3G wireless networks. By real-time video transport, we mean a piece of video content being delivered from a server in a wired network to a mobile client via a last-hop wireless link, to be decoded and viewed by the client before the entire content has been downloaded. This video streaming service must be compliant with the 3GPP packet streaming service (3GPP-PSS) [1], where the server uses RTP [2] for media transport and each client sends only RTCP reports as feedback to its server.

One common objective of media adaptation is congestion control whereby video sources reduce their transmission rates in reaction to deduced network congestion [3]. For paths involving wired and wireless links, that end-to-end feedback information alone is ineffective for congestion control purposes since it is not possible to identify where losses occur. Specifically, if losses occur in the wireless link due to poor wireless condition, it is not helpful for the sources to reduce their transmission rate. On the other hand, if losses occur in the wired network due to congestion, the sources should reduce their transmission rate.

One effective mechanism to provide additional information that allow sources to take appropriate actions is the RTP monitoring agent [4]—a network agent located at the junction of the wired and wireless network that sends *statistical feedbacks* (RTCP reports in particular) back to the sender to help the sender determine the proper action. However, the limited information contained in such statistical feedbacks is often insufficient for fine-grained application-level streaming optimization schemes [5]–[7].

Another limitation of using only end-to-end feedbacks is the long time for the feedbacks to arrive. In today's 3G wireless network, typical one-way delay of radio links is quite large—on the order of 100 ms—without link layer retransmissions. Thus, the actual end-to-end delay in practice can be quite large, especially with wireless link-level retransmissions implemented. Such long delay can severely impede the effectiveness of feedback information for the purpose of congestion control and beyond.

Both problems above can be solved simultaneously using a special agent called a streaming agent (SA) [8], located at the junction of the wired network and wireless link. Unlike the RTP monitoring agent [4], which provides only statistical feedbacks such as average roundtrip time (RTT) and packet loss rate, SA sends timely feedbacks, such as acknowledgment packets (ACKs), that tell the sender whether each packet has arrived at SA correctly and on time. We call such information provided by SA the wired application state, in contrast with information provided by RTP monitoring agent, which we term wired network state. Obviously, using fine-grained timely feedbacks one can derive wired network state as well as wired application state. Since most of the delay is in the wireless link, SA can provide much faster response about the condition of the wired network so that congestion control can react faster to alleviate network congestion. Furthermore, by providing the wired application state rather than just the wired network state, SA allows senders to have much more flexibility in media adaptation than it is possible with wired network state alone. Armed with SA's extra feedbacks, we next design application-level streaming systems that can take advantage of these feedbacks.. It is a complexity-scalable automatic retransmission scheme that capitalizes on SA feedbacks in estimating the success delivery probability of transmitted packets.

Related Work

The idea of inserting agents at carefully chosen locations in the network is not new, and it has been reported in [9] to increase web traffic performance and in [13] to monitor network services. In contrast, our agent-based approach focuses on the delivery of delay-sensitive media content over 3G wireless networks. Prior research related to wireless media streaming are extensive.

A. Network Protocols

The focus of works on wireless data transport has been on optimizing TCP over last-hop wireless networks [10]–[11]. As an example, [10] proposed a Snoop protocol that improves the performance of TCP over a last-hop wireless link connection. In brief, it is a TCP-aware link layer protocol that caches and retransmits TCP packets and suppresses negative acknowledgments from senders. Since we are focusing on streaming media content that are highly delay sensitive, the unpredictable transmission delay and delay variance of TCP over wireless links mean TCP is not appropriate for our application scenario. The IETF has been active in extending the current RTP specification [2] to enable better streaming quality, including proposals to extend RTCP to include timely feedbacks [12], and the use of forward error correction (FEC) to the standard RTP stream [13].

In related protocol development, HP labs Bristol has introduced UDP Lite [14] that offers more flexibility than the current UDP specification so that a packet with checksum errors will not be dropped automatically before being passed upward to the application. Our work is orthogonal to these developments and our proposed streaming agent and associated optimizations can potentially be modified to work with these new protocol specifications.

B. Media Optimization

An extensive body of previous research [5], [6], [7], [15], designs optimized media transmission schemes assuming the entire delivery path is one packet independent channel. Our work differs in that we separate the channel into two parts: a wired network and a last-hop wireless link. Our work can also be viewed as an extension of [6] and [7]: we show how SA timely feedbacks—information along the delivery path—can be used to enhance end-to-end streaming performance in a rate distortion sense.

Earlier works on media streaming optimization [16] for wireless links have focused on one or a few characteristics of the unique medium that differ from our work when optimizing media streams. The work in [17] assumed a packet loss model for the wired network and a bit error model for the wireless link and discussed an FEC scheme that offered packet-level and byte-level protection, respectively.

In contrast, we assume a packet loss model for both the wired network and the wireless link as it is common in current 3G networks. The work in [17] focused exclusively on the wireless last-hop.. As stated earlier, we assume a packet loss model for both wired network and wireless link.

C. Contemporary Work

The work in [21] and [22] employs a *proxy* at the wired/wireless boundary that caches media packets and intelligently requests packet retransmission from the streaming server after performing application-level rate-distortion optimized streaming procedures. In contrast, our *agent*-based approach is inherently a network level service provided by a wireless network operator, and hence has the following contrasting

characteristics: 1) our agent-based approach does not require fine-grained feedbacks from the clients, hence it is 3GPP-PSS compliant where clients send only RTCP statistical reports; 2) the agent does not interpret the payload of the media stream nor buffer any RTP packets, and hence it has much lower complexity overhead than a proxy; 3) it avoids security issues since payload can be encrypted without affecting operation correctness; 4) as a network service, the agent's feedbacks are useful for a hose of streaming applications at the server, one of which will be discussed in detail later; and 5) it retains the soft state property similar to the Snoop agent, where soft state is defined as the outage of a network element—temporary out-of-service due to equipment failure, etc.—will cause a tolerable degradation of performance instead of a catastrophic breakdown of the streaming session.

Streaming Agent

SA [8], an enhanced version of the RTP monitoring agent [2], is a network agent installed by the wireless network provider to provide network services to the server/client pair that are not possible with endpoints alone. It is located at the intersection of the wired core network and the transmitting wireless link (see Fig. 1 for an illustration). During a server-client RTP streaming session, SA identifies the stream by packet classification: look for matches at selected fields of RTP headers of incoming IP packets such as source and destination addresses and source and destination port numbers. SA periodically sends timely feedbacks (SA-FB) to the sending server in sub-second intervals, reporting the arrival status of the last RTP packets of a stream. The frequency of SA-FB and the value can be preset by the network operator or by the sender prior to the start of the streaming session via a message exchange between the sender and SA.

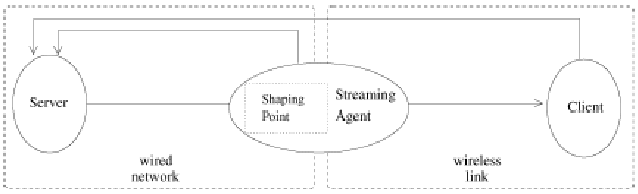

Figure 1 Using Streaming Agent (SA) to provide timely feedback.

In order not to overwhelm the wireless link, a shaping point, located just prior to SA, is used to limit the sending rate so that the packet rate is no larger than the wireless link bandwidth. Essentially, a layer-3 IP packet queue stores packets waiting to be fragmented and transmitted in lower layers (see Fig. 2 for an illustration). If the wireless link condition is poor, the number of retransmissions until successful transmission will be large, causing the IP packet queue to build up.

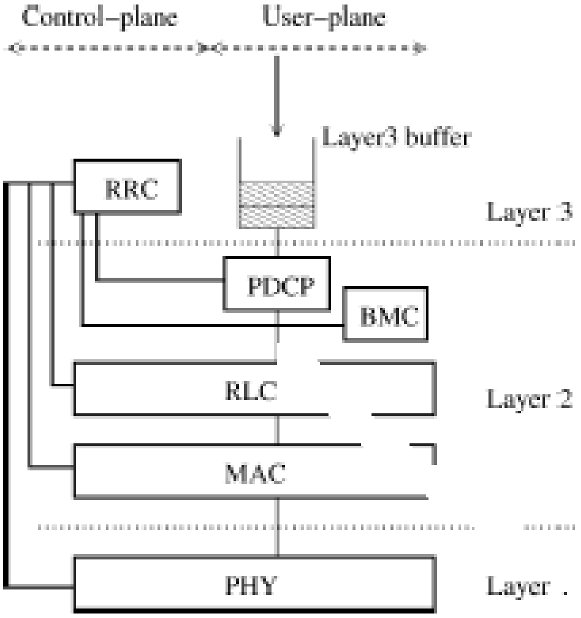

Figure 2 IMT-2000 Protocol Stack.

The shaping point reacts to the fullness of the queue by pre-dropping packets that would have been dropped anyway due to eventual queue over-flow. This way, packets that are dropped at the wireless link are dropped solely due to poor channel condition. Details of the shaping point are discussed in [4].

A. Using SA in 3GPP-WCDMA

We discuss here the implementation of SA in a wideband code division multiple access (WCDMA) communication system, one of the main 3G air interface technologies to be deployed in Europe and Asia, including Japan and Korea.

Two possible locations for SA within a WCDMA system are the radio network controller (RNC), which is responsible for the control of the radio resources of the radio access network, and the transmitting base station (node B) (see Fig. 3 for an illustration). Node B handles layer 1 processing such as channel coding and interleaving, rate adaptation, spreading, etc. RNC, on the other hand, performs layer-3 packet processing such as header compression. Hence, it is logical to place the functionalities of SA at RNC. Several link layer transmission modes are available for WCDMA. As an IP packet is passed down from layer 3 to layer 2, the RLC provides segmentation and retransmission services (see Fig. 2 for an illustration). Whether and how many retransmissions are done depends on the RLC modes. There are three modes: transparent, unacknowledged and acknowledged. For acknowledged mode, an automatic repeat request (ARQ) is used for error control. The tradeoff between link quality and link delay can be set by adjusting the number of retransmissions, set by radio resource control (RRC) during configuration.

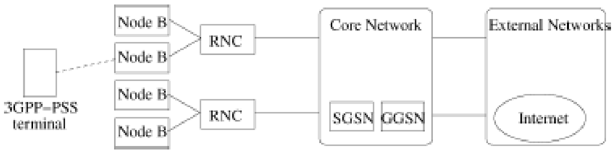

Figure 3 3GPP-WCDMA network elements.

Typical implementations use interleaving, channel codes and other techniques in lower layers to reduce raw packet loss rate to a reasonable level under normal conditions (less than 5%). If a small number of link-layer retransmissions is used in addition, then the resulting wireless link loss is very small. SA-FB can be used to deduce wired network state—essential for the sender to perform proper congestion control. SA-FB can also be used to deduce wired application state. Ideally, the server wants timely feedbacks directly from the client (C-FB) to reconstruct the *end application state*. In contrast, wired application state is at best an estimated end application state. So why are SA-FBs still desirable?

First and foremost, the current 3GPP-PSS [1] specification has dictated the use of existing RTP and RTCP specifications [2] only. That means one can rely on compliant 3GPP-PSS handsets to provide only RTCP feedbacks—statistical feedbacks only. Second, even if mobile client can be modified to provide fine grained feedbacks, it may have severe power constraint due to limited battery life. Having client send frequent feedbacks while receiving streaming video may not be desirable. Finally, if the intended streaming application can tolerate a fixed initial delay up to several seconds, we can elect to use acknowledgment mode as the wireless link layer transmission mode of choice to perform a small number of link-layer retransmissions. Doing so means the resulting wireless link loss is very small, and so wired application state closely mimics end application state, and therefore SA-FBs are essentially a much faster version of C-FBs, given the common hundreds of millisecond delay of the 3G wireless link.

SA Application: Complexity – Scalable ARQ

A. Problem Formulation

We next focus on a scheme that employs SA for video streaming optimization. We call the scheme complexity-scalable ARQ. In short, we derive a rate-distortion optimized application-level retransmission scheme, leveraging on previous work [6], [7], that adapts to SA and possibly client feedbacks to optimize video quality.

The basic problem framework is the following. There is a predicatively coded (IPPP…) video sequence with I-frame frequency L. At any given optimization instance, an optimization window equal to *M*-frame time is selected. The window is defined to be the set of frames whose delivery deadline (to be discussed) falls within start time and end time. Frames are brought into the optimization window at time end(t). Frames in the window expire at time when they cannot reasonably be expected to be delivered to the client on time. The slope of both functions—the rate at which they advance in time—is the playback speed at the client. We follow a standard congestion control protocol by adjusting the packet spacing in time, using the equation-based TCP-friendly congestion control in [3]

$$T_t = \mu_t\sqrt{\frac{2\epsilon_t}{3}} + 3(\mu_t + 4\phi_t)\epsilon_t(1 + 32\epsilon_t^2)\sqrt{\frac{3\epsilon_t}{8}} \qquad (1)$$

where $€_t$, μ_t, and Φ_t,are the updated estimates at time of packet loss rate, mean round trip time and round trip time variance, re- spectively. Suppose the optimization can be run at the server no more frequently than every P_{min} seconds, we select the optimization period at time t, P_t, as

$P_t =[P_{min} / T_t] T_t$ (2)

The number of packets that can be selected for transmission at optimization instance t-the bandwidth at time t, is $R_{t=}$ Pt / T_t

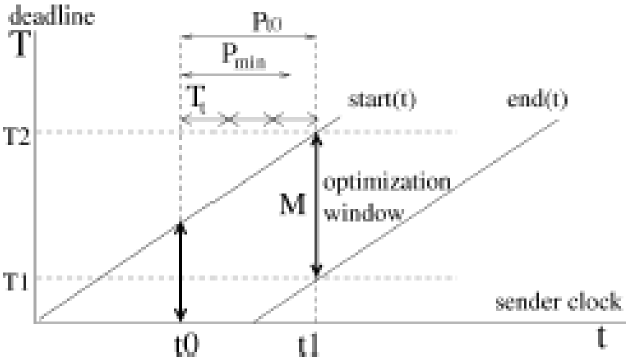

Figure 4 Optimization window.

i. Source Model:

Each frame is represented by one *data unit* (DU_i). Data unit is the smallest atomic unit considered during the optimization. Each DU_i is characterized by three numbers: delivery deadline T_i, size in number of RTP packets B_i, and reduction in distortion ι_i. DU_i points to a set of DUs that is dependent on for correct decoding. Specifically, for a predictively coded (IPPP…) sequence, DU_i is correctly decoded iff each DUj, k<j<i is correctly delivered by the delivery deadline to the client, by d_i. Otherwise, reduces distortion by 0.

For d_1 of the first I-frame, we calculate the peak signal-to-noise ratio (PSNR) of encoded I-frame against the original frame 1, *plus* the PSNR of encoded I-frame against the original frame 2 through. The reason is that in the event of a frame decoding failure,i>1, as a simple error concealment strategy we can display correctly decoded frame 1 during playback time of frame i.This resulting reduction in distortion wired application state needs to be accounted for in d_1.

To calculate di of frame i>1, we calculate the PSNR of encoded P-frame I against the original frame i, plus the PSNR of encoded P-frame i against the original frames i+1 through as we did for the first frame, *minus* the PSNR of encoded frame i-1 against the original frame i through L. The reason for the last term is that if I frame is correctly decoded, that it should be more similar to future frame than frame j>1 than frame i-1, then using frame I as display time of frame j>I will cancel out the benefit of using frame i-1 for error concealment of frame j.

ii. Network Model:

Essentially, the wired and wireless parts of the network are each modeled by a time-invariant independent packet erasure model with constant delay, where packet loss rates are α and β in the wired network and wireless link, respectively. Let π_i be the number of transmissions for in the current optimization instance. Let $\Phi_{i=}$ {n_i,a_i, b_i }be the history of DU_i in all previous optimization instances: DU_i's total number transmissions to-date,, and the number of ACKs received from the SA and client, a_i and b_i respectively.

iii. Mathematical Formulation:

Given a window of data units, the problem is to determine the optimal retransmission scheme for data units in the window. Like [7], the problem is formulated as a minimization of end-to-end distortion subject to a transmission rate constraint. The optimizing variable is set $\pi=\{\pi_{i\ldots\ldots}\pi_m\}$ where π_i is the number of times is transmitted in the window. By transmission policy then, we mean how many times each data unit is transmitted for this optimization period of duration P_t.

B. Dynamic Programming (DP) Solution

To solve (4), we employ a DP technique inspired by [8]. To simplify discussion, we assume for now that the size of the optimization window is, and we are optimizing a sequence of 1 I-frame plus dependent P-frames.We denote $\theta(k,\pi)$ by the additional distortion reduction from frame k to L provided by policy vector given the first frames are correctly decoded.

$$\Theta(k,\pi) = \sum_{i=k}^{L} d_i \prod_{j=k}^{i} (1 - \epsilon(\phi_j, \pi_j)).$$

The DP solution can be derived from the above eqution naturally by defining as the optimal $\theta(k,\pi)$ given the rate budget for frame to is packets. Assuming we bound the maximum number of transmissions for a given data unit in the current optimization window to be N.

Complexity Scaling*:* Like [6], a complexity of $O(NLR_t)$ means the algorithm is pseudopolynomial. In our case, it means a large running time for large R_t. To reduce running time, we can solve an approximate instance of the problem by performing a rounding operation by for the sizes of data units B_i's and rate constraint R_t

$$B_i^r = \left\lceil \frac{B_i}{K} \right\rceil \qquad R_t^r = \left\lfloor \frac{R_t}{K} \right\rfloor.$$

A. *Simulation Setup*

We performed simulations using Network Simulator 2 [24].The setup is shown in Fig. 5. Three nodes were constructed, *n*0, *n*1 and *n*2, representing the three locations of the server, SA and the wireless client, respectively. To connect these nodes, two links were constructed. Link n0-n1, simulating the wired network between the server and SA, had constant propagation delay and uniform loss rate

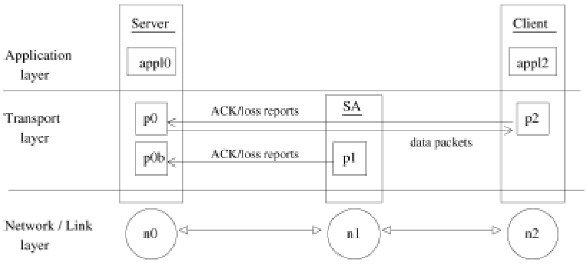

Figure 5 Simulation Setup.

For link n1-n2 that simulated the wireless link of rate 144 kbps, we implemented link-layer retransmission as follows: a network layer packet was fragmented and grouped into transport blocks of 180 bytes each that spanned 10 ms. Groups of transport blocks were interleaved (spread) to give a one-way delay of. In reality, larger spread reduces the probability of error due to fading at the cost of a larger end-to-end delay. Each transport unit was transmitted through the wireless link with success probability. When link-layer retransmission was used, each transport unit was retransmitted when the sender timed out on the receiver's link-layerACK. The maximum number of retransmissions was set at 20.

The transport layer had a duplex connection (p0-p2) from the server n0 to the client n2 and a simplex connection (p1-p0b) from SA n1 to the server. p0-p2 was for endpoint data transmission from server to client. p1-p0b was for feedbacks from SA to server; a filter was placed at link n1-n2 to sniff out packets targeted to the client and to forward them to p1, who then sent ACKs to the server.

A server application, app0, sat at sender node and sent packets to the client using the connection p0-p2. Each packet had a sequence number in the packet header indicating the frame it contained; we assume each frame maps to one RTP packet.

For each video sequence, a distortion matrix d(j,i) was generated offline and loaded into both server and client before playback began. Server derived distortion value for each data unit using (3), and client calculated the visual PSNR during each frame playback time: if frame I is correctly decoded, then PSNR is d(I,i) ; if not, then the most recently correctly decoded frame was used for display for frame I, and we used d(j,i) as PSNR. If no such frame was available, then PSNR was 0.

B. *SA Application: Complexity-Scalable ARQ*

We will compare our proposed complexity-scalable ARQ scheme using SA with two schemes. The first is a rate-distortion optimized streaming scheme [7] based solely on client's RTCP feedbacks. We term this scheme *No Agt*. We will see soon that *No Agt* suffers severely from two shortcomings: 1) not having fine-grained client feedbacks means *No Agt* cannot accurately estimate success delivery probability of transmitted packets and 2) not being able to distinguish between wired and wireless loss and wired and wireless delays means *No Agt* performs unnecessary wired network congestion control, sending at rates much lower than necessary.

The second is a proxy scheme proposed in [26], [27] that perform a hybrid sender/receiver-driven rate-distortion optimized streaming optimization. For convenience, we term the proxy scheme *CCG* after the authors. In brief, *CCG* essentially performs a receiver-driven version of [7] at the proxy to request media packets from a streaming server, caches the packets, then performs a sender-driven version of [7] to transmit packets to the client. Recall that we have already pointed out the differences between our agent approach and the proxy approach, including security concern, complexity overhead and soft state property in Section II-C. Nevertheless, we will compare their performance in this section.

We use H.263 Version-2 video codec (TMN10) to encode two 300-frame MPEG test sequences, container and foreman. They are coded in QCIF (176 144) format at 120 kbps, 30 frames/s and at one I-frame every 25 frames. The resulting average PSNR1 for the compressed streams under noiseless conditions are 38.49 and 32.46 dB, respectively. For each data point, the video sequence was replayed upon completion until 600- second playback time has been reached. This was done for an averaging effect. For the wireless link, acknowledgment mode is set up as described in Section Simulation Setup, with wireless loss rate fixed at 04.

For complexity-scalable ARQ using SA, termed *SA* below, we assume a 2-s delay between server start time and client playback time. The size of the optimization window is 10-frame time. Optimization is performed every second, i.e., we assume a sending rate of 1 packet per optimization period.

This means the procedure of rounding by a factor of, discussed in Section IV-B1, is not necessary, as is already the smallest integer possible. For *CCG*, we assume a 1-s buffering delay at the proxy, and a 1-s delay buffering delay at the client. This is roughly equivalent to the 2-s delay employed at the client for the *SA* case. The proxy also had an optimization window of ten frames, and it performed the same congestion control protocol (1) as *SA* as described in Section Problem Formulation.

(a) Wired Delay (b) Wireless Delay

(a) Wired Delay (b) Wireless Delay

Figure 6 Performances with Streaming Agent (SA).

Conclusion and Future Work

In this paper, we proposed the use of a Streaming Agent at the junction of the wired and wireless networks to provide additional timely feedbacks to the streaming servers. We discussed one specific streaming optimization, complexity-scalable application-level retransmission that exploits such feedback to demonstrate potential benefits. Through simulation, it is shown that significant PSNR improvement can be maintained over non agent-based systems.

For future work, extensions of the streaming agent that provide more services is possible. Given the network agent already monitors media flows, it is perhaps sensible for it to perform network policing, for example, to make sure it does not operate at a higher sending rate than it deserves. Information collected at the network agents can also be used for core network analysis for possible network-wide optimizations.

References

[1] 3GPP TS 26.233 Transparent End-to-End Packet Switched Streaming Services PSS); General Description (Release 4) (2001, Mar.). *[Online]. Available: ftp://ftp. 3gpp.org/Specs/2001-03/Rel-4/26_series/ 26 233-400.zip.*

[2] H. Schulzrine, S. Casner, R. Frederick, and V. Jacobson, “RTP: A Transport Protocol for Real-Time Application*,” IETF RFC 1889, 1996.*

[3] S. Floyd, M. Handley, J. Padhye, and J. Widmer, “Equation-based congestion control for unicast applications,” *in Proc. ACM SIGCOMM, Stockholm, Sweden, Aug. 2000.*

[4] T. Yoshimura, T. Ohya, T. Kawahara, and M. Etoh, “Rate and robustness control with RTP monitoring agent for mobile multimedia streaming,” *in Proc. IEEE Int. Conf. Communication, New York, Apr. 2002.*

[5] M. Podolsky, S. McCanne, and M. Vetterli, “Soft ARQ for Layered Streaming Media,” *Univ. California, Berkeley, Tech. Rep.UCB/CSD-98-1024, 1998.*

[6] V. Chande and N. Farvardin, “Progressive transmission of images over memory less noisy channels,” IEEE J. Select. Areas Commun., vol. 18, no. 6, pp. 850–860, Jun. 2000.

[7] P. Chou and Z. Miao, ““Rate-Distortion Optimized Streaming of Packetized Media,” *Microsoft Research Technical Report,”, Tech. Rep.MSR-TR-2001-35, 2001.*

[8] G. Cheung and T. Yoshimura, “Streaming agent: A network proxy for media streaming in 3g wireless networks,” *in Packet Video Workshop, Pittsburgh, PA, May 2002.*

[9] M. Margaritidis and G. Polyzos, “Mobiweb: Enabling adaptive continuous media applications over 3g wireless links,” *IEEE Pers. Commun. Mag., vol. 5, no. 6, pp. 36–41, Dec. 2000.*

[10] H. Balakrishnan, V. Padmanabhan, S. Seshan, and R. Katz, “A comparison of mechanisms for improving TCP performance over wireless links,” IEEE/ACM Trans. Networking, vol. 5, no. 6, pp. 756–769, Dec. 1997.

[11] Liu, D. Goeckel, and D. Towsley, “TCP-cognizant adaptive forward error correction in wireless networks,” in *Proc. INFOCOM*, New York, NY, Jun. 2002.

[12] J. Ott, U. Bremen, S. Wenger, S. Fukunaga, N. Sato, K. Yano, A. Miyazaki, K. Hata, R. Hakenberg, and C. Burmeister. (2002) Extended RTP Profile for RTCP-Based Feedback. IETF. [Online]. Available: draft-ietf-avt-rtcp-feedback-02.txt.

[13] J. Rosenberg and H. Schulzrinne, “An RTP Payload Format for Generic Forward Error Correction,”, IETF RFC 2733, 1999.

[14] L.-A. Larzon, M. Dagermark, and S. Pink, “UDP Lite for Real Time Multimedia Applications,” HP Laboratories, Bristol, Tech. Rep. HPLIRI- 1999-001, 1999.

[15] P. Chou and A. Sehgal, “Rate-distortion optimized receiver-driven streaming over best- effort networks,” in *Packet Video Workshop*, Pittsburg, PA, Apr. 2002.

[16] Q. Zhang, W. Zhu, and Y.-Q. Zhang, “Network-adaptive scalable video streaming over 3g Wireless network,” in *Proc. IEEE Int. Conf. Image Processing*, Thessaloniki, Greece, Oct. 2001.

[17] H. Matsuoka, T. Yoshimura, and T. Ohya, “Design, implementation and performance measurement of multimedia streaming protocol (MSP),” in *Proc. Asian Int. Mobile Computing Conf. (AMOC’2002)*, May 2002.

[18] J. Chakareski and P. Chou, “Application layer error correction coding for rate-distortion optimized streaming to wireless clients,” in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 3, Orlando, FL, May 2002, pp. 2513–2516.

[19] J. Chakareski, P. Chou, and B. Aazhang, “Computing rate-distortion optimized policies for streaming media to wireless clients,” in *Proc. IEEE Data Compression Conf.*, Snowbird, UT, Apr. 2002, pp. 53–62.

[20] J. Chakareski, P. Chou, and B. Girod, "Computing rate-distortion optimized policies for hybrid receiver/sender driven streaming of multimedia," in *Proc. Asilomar Conf. Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 2002.

[21] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizes," *IEEE Trans. Acoustic., Speech, Signal Process.* vol. 36, Sep. 1988.

[22] Lee, G. Chan, Q. Zhang, W.-W. Zhu, and Y.-Q. Zhang, "Optimal allocation of packet- level and byte-level FEC in video multicasting over wired and wireless networks," in *Proc. GLOBECOM*, San Antonio, TX, Nov. 2001.

[23] A.S.Syed Fiaz, R. Pushpapriya, S. Kirubashini, M. Sathya, "Generation and allocation of subscriber numbers for telecommunication", International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR), pp. 257-266, Vol No: 3; Issue No: 1, March -2013.

Author's Biography

A.S.SYED NAVAZ received M.Sc in Information Technology from K.S.Rangasamy College of Technology, Anna University, Coimbatore, M.Phil in Computer Science from Prist University, Thanjavur, M.C.A from Periyar University, Salem and Pursuing Ph.D in the area of Wireless Sensor Networks. He researched and published in International journals and working as Editorial Board Member & Reviewer for International journals also Member in 13 International Social Bodies. Currently he is working as an Assistant Professor in the Department of Computer Science at Muthayammal College of Arts & Science, Namakkal, India. His Research areas are Wireless Sensor Networks, Mobile Computing & Image Processing.

P.JAYALAKSHMI received the M.Tech degree in IT/Networking from VIT University in 2006. She is an Assistant Professor Senior at VIT University in Vellore, Tamil Nadu. Her research interests include Computer Networks and Mobile Adhoc Networks. She published papers in international journals to her credit. She is a member of the IEEE WIE and the CSI and her contact e-mail id is pjayalakshmi@vit.ac.in

N.ASHA working as Asst. Professor in School of Information Technology and Engineering (SITE), VIT University, Vellore District, Tamil Nadu, India. She completed her master degree-(M.E) in Computer Science Engineering. She is pursuing Ph.D in the area of software testing. Her main research interests include Computer Networks, Software Engineering, Software Testing, Re-engineering. She has many publications in national and international journals and conferences to her credit. Asha can be contacted by e-mail at nasha@vit.ac.in