International Conference on Network Applications, Protocols and Services 2008 (NetApps2008)
21 - 22 November 2008
Executive Development Center, Universiti Utara Malaysia

A Survey on Buffer and Rate Adaptation Optimization in TCP-Based Streaming Media Studies

A. Suki M. Arif, Osman Ghazali, Suhaidi Hassan
InterNetworks Research Group
Graduate Department of Computer Science
UUM College of Arts and Sciences
Universiti Utara Malaysia
06010 UUM Sintok, Malaysia
{suki1207, suhaidi, osman}@uum.edu.my

***Abstract*-Contrary to the popular conventional wisdom that the best transport protocol for the streaming media is UDP, many findings found that most of the transport protocols used nowadays are TCP. Two main reasons that UDP is not being used widely are it is not friendly to other flows and some organizations are blocking this protocol. In the meantime, TCP is naturally reliable and friendly to other flows. But with so many controls inbuilt in the protocol; such as congestion control, flow control, and others with the heavy acknowledgement mechanism, resulting delays and jitters. Thus it's naturally not friendly to the streaming media. But with all the inherited weaknesses, we have seen explosive growth of streaming media in the Internet. With these contrasting premises, it is very interesting to study and investigate the streaming media via TCP transport protocol, specifically on buffer and rate adaptation optimization.**

I. INTRODUCTION

Recent years have witnessed explosive growth of the Internet and increasing demand for multimedia information services. Streaming media via the Internet has received tremendous attention. The old way of accessing multimedia information via the Internet is by *downloading mode*, in which the users download the entire video or audio file. Now, with the streaming mode, there will be a very short buffering-downloaded period, and then the remainder of the media is obtained across the network as the media is played-out. Therefore the users don't have to wait long to obtain the services.

To gain good quality streaming media, several weaknesses need to be addressed. In term of transport protocol, the inherited problems are lacking of throughput guarantees, variabilities in bit rates, delays or jitters, and packet loss [1, 2]. Those characteristics are not "friendly" to the streaming media. Streaming media is favor timeliness to reliability. That means, streaming media is able to compromise packet losses but sensitive to packet delays.

Conventional wisdom holds that UDP is a better transport protocol than TCP for the streaming media [3, 4]. This wisdom is easy to be understood because UDP is a best-effort delivery service. Theoretically, there will be less delay and provides better bit rates (throughput). Unfortunately, this best-effort transport protocol potentially impedes the performance of other applications that employ TCP, or worse, endanger the stability of the Internet [5]. In addition to that, some organizations are blocking these types of application [1, 6].

On the other hand, TCP is complex and dynamic; employs congestion control schemes that adapt dynamically to network conditions, retransmit loss or time-out packet, thus it often yields variable transmission rates and packet delays [7]. Its reliability is inherently unsuitable for streaming media, which is a time-sensitive application [2].

Although the above-mentioned premises seems to indicate that the present transport protocol, namely TCP and UDP, are not suitable for streaming media, several studies found that media streaming is significantly growing in the Internet traffic [8-10]. Many studies also found that TCP is a more popular transport protocol than UDP [6, 11].

Several authors have stated their arguments on high promising potential of running streaming media over TCP transport protocol [3, 6, 11-13]. The researchers have found that the bandwidth are adequate to access multimedia streaming for many broadband users by using direct TCP [11]. Direct TCP means without further improvement to the application technology or the protocols and can be run smoothly. For example, the study by [3] reveals that TCP may not be as delay-unfriendly as is conventionally believed. The authors have justified their finding by giving a reason on congestion control mechanism used by TCP regulates rate as a function of the number of packets sent by the application. Consequently it biases towards flows with small packet sizes. Since streaming application uses constant and small packet sizes [14], it gives more advantage than application with longer packet sizes.

In addition to that, the studies by [11] also found that direct TCP streaming generally provides good performance when the available bandwidth (achievable TCP throughput) is roughly twice the video bitrates. The other studies have found that streaming video clips on the Internet today are encoded at bit rates 89-300 Kbps [8, 15]. We are witnessing the rapid deployment of broadband connectivity which support download rates of 750 Kbps – 1 Mbps [11]. These studies also have been supported by [4]. They reveal that streaming application is able to be transported via present TCP with good performance in a wide variety of scenarios, especially when the available bandwidth is more than the video bitrates.

The other interesting studies by [16] also found that TCP retransmission or lost packet recovery mechanism in TCP protocols is not too severe for streaming application (with appropriate buffer size). They found that 94% of "recoverable" lost packets returned to the client before their decoding deadlines and almost 95% of all recovered packets were recovered in a first retransmission attempt.

The researchers are looking at many angles to improve further the streaming media over TCP. As stated previously, the problem of TCP for streaming media are delays, jitters, data losses and bandwidth throughput fluctuation. Almost all of the previous mentioned problems can be solved by using appropriate buffer size at the client [2, 17] and variable transmission rate media [4, 18, 19]. The appropriate buffer at the client is able to store some level of media data, thus it can absorb the delays, jitters and bandwidth throughput fluctuation, should they occur. Therefore streaming applications can gain more tolerance towards unpredictable changes of network conditions. Whereas the variable media bit rates mean the media is encoded into several quality levels. Should the bandwidth degrade, the server will switch to the lower quality transmission rate. Therefore, the users still be able to enjoy a smooth streaming media.

The rest of the paper is organized as follows. Section 2 discusses the development in the usage of buffer at client in order to absorb the TCP bandwidth fluctuation. In section 3 we will elaborate at what researchers are doing in implementing adaptive bit rates. In Section 4 we will discuss roadmap for the ongoing and future studies for streaming media. On the last section we will conclude our paper.

II. TCP WITH BUFFER OPTIMIZATION

In streaming media, the data has to be continuously available at the client to prevent any lapses in the network bandwidth or any playback disruption. As stated in the introduction section, by using appropriate buffer size, the delays or network throughput fluctuation, can be absorbed. Thus, it will allow the client to continue playing back the content during lapses in network bandwidth and compensate for short-term variations in packet transmission delay [14].

The buffer level will increase or decrease depending on the ratio between the arrival rate of media data (transmission rate) and the coding rate for playback purpose (refer fig.1). Therefore, in the case of the arrival rate greater than the coding rate, the buffer level will increase, and vice versa. Should there is lapse in network bandwidth (the Internet) for some duration (the arrival rate throughput will decrease), the buffer level will decrease as well. It may leads to buffer level drain-out and the playback may be halted.

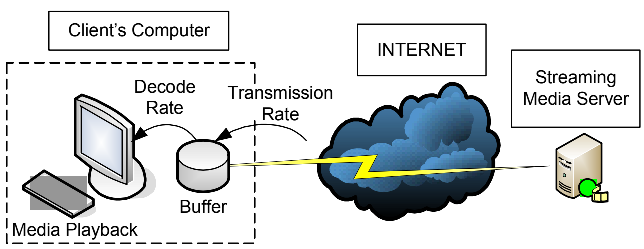

Figure 1. Buffer level increasing or decreasing mechanism.

The issue is what is the appropriate buffer size [20] and how to ensure enough media is filled into the buffer, consistently? If it is too big, there will be delay in the playback start but contribute towards robustness as well as high and nearly constant media streaming quality. If it is too small, it will easily drain-out, unable to sustain uninterrupted playback and rebuffering will be repeated many times. For "thin-clients", such as mobile devices, they have very limited buffer space. Consequently, they may be unable to store enough media to be able to compromise network throughput fluctuations or delays. Others, for example interactive streaming applications are very time sensitive applications. Furthermore, the Internet is highly dynamic and throughput fluctuation will occur regularly, thus consistent buffer-filling is almost impossible.

Buffer Optimization with Faster Trasmission Bit Rate

Popular commercial streaming media products nowadays (utilize more than 90% of streaming media traffic [15]), namely Windows media services and RealNetworks media services, are using a creative method to solve this problem. They are using faster transmission bit rate than media encoding rate, if the available bandwidth permits. By introducing this technique, they allow clients to use big buffer size without the need to wait for a relatively long period for the media to be played-out.

The authors in [6] argue on the effectiveness of using faster transmission bit rate than encoding rate, specifically in Windows media service's Fast Streaming. The weaknesses of faster transmission rate is over-utilizing of bandwidth and CPU resources. According to them, Fast Streaming over supplies of media data to the client is up to 54.8%. This is due to clients' early terminations. Some of them will find that the clip is not interested after a few seconds viewing, therefore they will abort the connection as well. The finding in [15] also reveals that 20% of the downloading traffic comes from aborted connections, in which they conclude that it wastes a significant amount of the Internet bandwidth. They also stated that 44% of streaming sessions are terminated before 10% of the object is played. There are only 11% of the users play the entire objects.

Buffer Optimization with Other Methods

The other way of optimizing playback quality that relates to buffer is done by [21]. They propose a scheme that dynamically allocates more bandwidth to streaming media session in which the client buffer level is lower compared to others that share the same bottleneck link. They call their scheme as Playback-Buffer Equalization (PBE). They utilize QoS in the network to classify the packets from many clients based on the label contained in a packet. The label is inserted in each packet corresponding to the buffer level of the streaming session the packet belongs to. By implementing this scheme, the low playback buffer will get higher priority than others, especially the one with excess buffered data.

The researchers in [22] have introduced Dynamic Buffer Active Tuning (DBAT). DBAT is a send buffer (at server side) and it will provide feedback (to the server application) before the send buffer overflows. By introducing this method, the number of lost packets as a result of buffer overflow is able to be reduced and it will improves overall throughput.

III. TCP WITH ADAPTIVE BIT RATES TRANSMISSION

Perhaps the most major technical challenges in streaming media is to adapt to changing network conditions. Unfortunately, in the best effort network, there is no way to select unique target rate ahead of time. Should the bandwidth transmission rate degrade continuously, the media bit rate could not afford the same media quality, and will leads to stall in media display. In term of user satisfaction, they prefer continuous lower quality display than good quality display with intermittent pause. The strategy of adaptive streaming is to adjust the media rate according to network conditions [23]. Therefore, many commercial streaming media nowadays rely on multiple bit rates or adaptive bit rates to adapt to the changing network conditions.

In practice, the range of adaptation should be discrete rather than continuous. The user perceives quality will suffer if there is frequent changes. Thus, another aspect to be considered is to maximize the range of supported rates and quality levels, and to provide the finest granularity of realizable points within that range. The greater the range and the finer the granularity, the more the freedom in the media rate-matching with the network bandwidth [23].

One of the methods of utilizing adaptive bit rates in response to changing network conditions is to use media scaling or content adaptation [1]. Media scaling is a method to reduce or increase the quality of media; for example by changing the video frame rate, video resolution, different encoding layers etc. Typical media scaling techniques are temporal scaling, quality scaling, and spatial scaling. Therefore by changing the quality of media, we can change the transmission bit rates as well, to accommodate the changing network conditions.

One of the scalable adaptive encoding scheme is called Fine Granularity Scalability (FGS) [24]. In this scheme, a movie is encoded into base picture and extra quality picture; such as adding resolution either temporal or spatial. As the scheme is called (fine granularity scalability), the video can be scalabled to the nearest bandwidth availablity (than having only a few static quality media) [17]. Thus, it will utilizes optimal bandwidth or sends the best quality image as permitted by bandwidth availability. One of the recent video codec standard is ISO/IEC14496-10 or H.264/AVC.

Adaptive Transmission with the Assistance of RED/ECN

Some of the researchers are applying ECN (Explicit Congestion Notification) [25] in order to gain better rate adaptation decision for streaming application. ECN will use active participation of the router to detect the incipient of congestion before the queue overflows, and provide an indication of this congestion to the TCP end nodes (refer fig. 2a and fig. 2b). Thus, it can reduce unnecessary queuing delay for all traffic sharing that queue. The original TCP's congestion control and avoidance algorithms are based on the notion that the network is a black-box. The network's state of congestion or otherwise is determined by end-systems probing for the network state, by gradually increasing the packets on the network. The TCP source detects dropped packets either from the receipt of three duplicate acknowledgments or after the time-out of a retransmit timer.

One of the studies that utilizes ECN in order to improve rate adaptation decision is done by [26]. They adopted "a network-aware approach" phrase to express the notion that the active involvement of network routers to give feed-back on the network status. Thus, they will gain better network status information compare to the other studies in which infer network status based on end-to-end black-box constrain information.

By inferring network status based on end-to-end black-box only information, the status may not be correct. When there are many duplicate acknowledgements or spurious timeout are observed, that does not mean congestion has occurred [27]. It may happen as a result of packet reordering, packet duplication, or a sudden delay increase in the data or the ACK path. In their report, Raghuveer, A. et. al. [26] wrote that the transmission rate will be increased when network fluctuation is detected (specifically when the client experiences a low arrival rate for considerable amount of time) but there is no incipient congestion in the network. There is no arbitrary increase the transmission rate, but there will be "bandwidth probing" prior step in which to check if there is enough available bandwidth to support higher quality video. Therefore, they will not add unnecessary congestion to the path.

Other novel effort by [26] is to introduces target arrival rate calculation for the client, which will avoid the buffer underflow. Then they will seek for the transmission rate that will result in the targeted arrival rate. Besides that, they also do not change the video quality when the transmission rate changes. Therefore, the user's perceived quality is maintained.

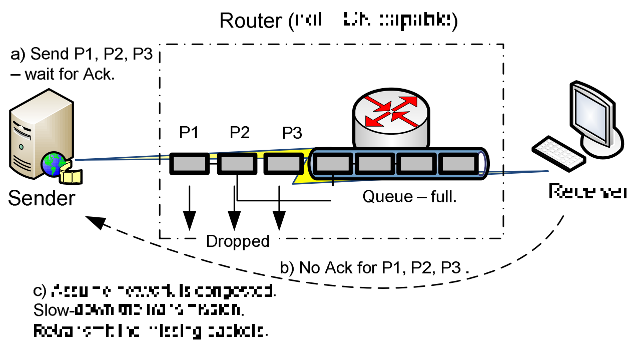

Figure 2a. TCP without ECN.

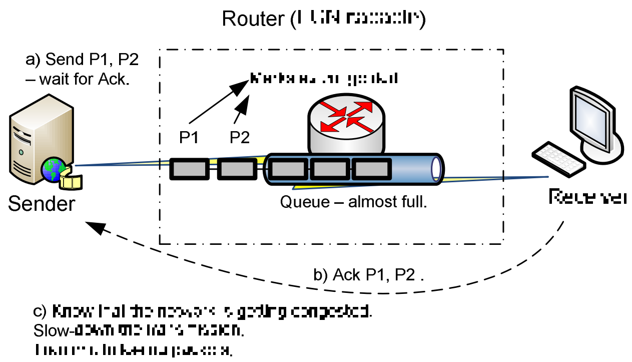

Figure 2b. TCP with ECN.

Other Rate Optimization Methods

One of the creative methods in optimizing the transmission rate is by performing an adaptive playout. By implementing this mechanism, we don't have to select the closest pre-sets bit rates (as in transmission rate adaptation). Even, if we use a fine-grained rate-scalable codec, we still do not take full advantage of the bandwidth available. To the best of our knowledge, there is no widely accepted or truly efficient rate-scalable adaptation.

One of the mechanisms is called Adaptive Media Playout (AMP) by [28]. They claimed that by adjusting the video playout quality at the client side, for example by simply adjusting the duration that each frame is playing back, the transmission rate will be able to full-utilized the available bandwidth. They also claimed that by slowing down the playout rate up to 25% it is often un-noticeable.

The other method that is being extensively research is to use Quality Oriented Adaptation Scheme (QOAS) [24, 29-33]. This method will depend to client agent to monitor and periodically gives feed-back to the server regarding the status of transmission quality. The status of transmission quality is computed base on network metrics such as loss rate, delays and jitters. Then the server will reduce or increase the transmission quality to adapt to existing network fluctuation.

IV. ROADMAP FOR STREAMING MEDIA STUDIES

Many of the streaming media problems can be addressed by implementing appropriate buffer size and effective adaptive transmission rate. Unfortunately, to implement the ideas is not an easy solution. For instance, to ensure consistent buffer-filling in dynamic changing network conditions. One of the solutions is to implement higher transmission rate than media rate. But by implementing this mechanism, there will be over supply of data and resources. Same as with the challenges to implement effective transmission rate adaptation. It is not a trivial task to determine either the network quality degradation is not a brief-fluctuation. Should the rate changes frequently, the users' perceive quality will be lower.

Moreover, the applications implementation still pose many problems for low delay and interactive streaming applications. Besides that, the researchers are looking at many other angles to improve further the streaming media over TCP. Among others are the problems that are inherited in TCP protocol algorithm. For example, reliability mechanism in which dependent on retransmission algorithm will potentially increase wait time (delays) between consecutive retransmission. Meanwhile, there will be frequent bandwidth throughput fluctuation due to TCP's AIMD mechanism. This mechanism form a saw-tooth pattern. When congestion occurs the rate is decreased almost half before it is increased back (refer fig. 3), thus it is rather not suitable for streaming media or it may lead to annoying jerky playback.

The Roadmap for TCP-Friendly Streaming Media

Although UDP is not favorable protocol for the streaming media due to unresponsiveness or unfriendliness to the other TCP flows, UDP initial design is for real-time (time-sensitive application) or with unreliability mechanism.

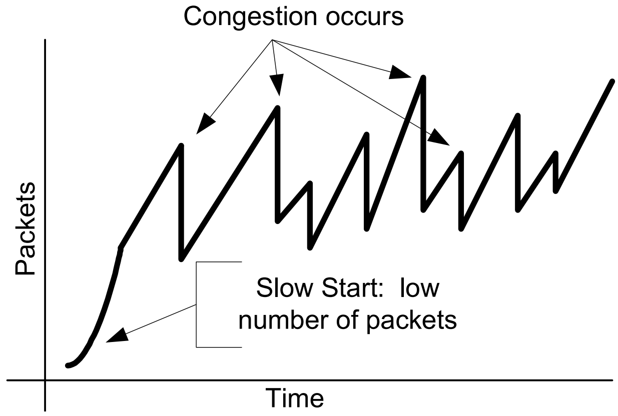

Figure 3. AIMD sawtooth behavior – probing for bandwidth.

Besides adding friendliness to the UDP, the researchers also optimize the protocol to well-suite with streaming media applications. Base on above premises, some researches are being done to utilize UDP as transport protocol for streaming media and at the same time adding rate control mechanism [34].

Most of the researches in this champ are creating or modifying UDP layer, or at upper layer, so that it will be friendly to other flows, also attempts to mimic TCP throughput and without instantaneous fluctuations of TCP's AIMD algorithm [34]. These kinds of protocols are introducing rate control or congestion control to the UDP. They are called TCP-friendly protocol. The protocols is said TCP-friendly when its bit rates does not exceed the maximum bit rates from a conformant TCP connection under equivalent network conditions [5] or when it does not reduce the long-term throughput of any coexistent TCP flow than another TCP flow on the same path under the same network conditions [35].

Various methods have been studied and introduced. Some of the methods are emulating, or even applied directly, TCP's AIMD behaviour to be friendly to other TCP flows [35]. Although they mimic the TCP behavior, the associated reliability mechanism are not included [35] and consequently it reduces delay. Therefore the best of both worlds are gained, friendly to other TCP flows and at the same time does not creates unnecessary delay to the time-sensitive streaming media.

Perhaps the most promising protocol in this category is the Datagram Congestion Control Protocol (DCCP) [36]. It is a recently proposed unreliable transport protocol incorporating end-to-end congestion control, which is similar to TCP. Phrase as congestion control without reliability [37], it has been motivated by the need for a new transport protocol that offer a reliable delivery; accommodates alternative congestion control algorithms; accommodates the use of explicit congestion notification (ECN); and requires minimal overhead in packet size and CPU processing at the data endpoints [38].

It has been created by the very persons who have been involved in the development of the transport protocol, usability and the stability of the Internet. Moreover, only minimum but very much similar TCP's congestion control have been added to this protocol. It promises a stable, applicable and future-proof protocol that can be used to replace UDP. In addition to that, it has been designed to accommodate alternative modular congestion control mechanisms, namely TCP-like, TFRC and future, extensible or further form of congestion control. By doing this, it offers any application to negotiate suitable congestion control without need to embed in the application layer.

However, the challenges are still there. As pointed by authors in [37], though they tried to resemble TCP as similar as possible (to take advantage of TCP full-proof implementation stability), but without reliability mechanism as in TCP, they had to reconsider almost every aspect of TCP's design. The intertwine semantics relationship between congestion control and reliability mechanism are very close, make it impossible to separate both mechanism without changing something. Therefore, to a certain extent, we still need to test the protocol in action [22].

Implementation wise, TCP-friendly's UDP-based streaming media is not much of the protocols' mechanisms but rather on industries acceptance of the protocols. To deploy such protocol in the global Internet are likely if they offer vastly improved performance with today's Internet Infrastructure [35]. Some of the mechanisms introduce are just pass simulation phase, sometimes with limited performance metric only. Others are offering limited performance improvement. Other algorithms are empirical and often fail in a dynamic network.

Overcome Shortcoming of TCP Reliability Mechanism for Streaming Application

Another way of optimizing TCP for streaming media is to overcome the inherit TCP weaknesses for the streaming media, specifically the inbuilt reliability mechanism. For instance, abrupt jump in media transmission rate is unavoidable and it will be problematic for any congestion control algorithm [39]. The abrupt jump is caused by "motion compesation", in which codecs will effectively transmit changes from one video frame to another. There will be frequent little changes and major motion interchangebly from one frame to another, resulting abrupt jump in media transmission rate. At the same time the principle for almost all congestion control algorithms in transport protocol should increase gradually. These mismatch mechanisms pose inefficient streaming media applications.

Although we mentioned studies by [16] which indicate that the current TCP retransmission or lost packet recovery mechanism is not too problematic for streaming application, it still poses many problems for low delay or interactive streaming applications (such as video conferencing or telephony). When a packet lost or timeout occurs, retransmission mechanism is triggered. One of the important parameter here is retransmission timeout (RTO) estimation. Should the value is small, it may trigger spurious timeout. On the other hand, if the value is large, the chances is that the retransmitted packet will be late and useless for decoding (at client side).

Loguinov D. and Radha H [40] have evaluated several RTO estimators in the context of a video streaming application; namely, early TCP-like RTO estimators by Jacobson [41] and modifications of that as written in RFC2988. The studies in [40] have found that previous TCP-like estimators is not quick enough to detect lost packets.

Sinha R. and Papadopolous C. [42] have introduced timer-less to eliminate RTT estimation and timer-triggered events. By eliminating time-based calculation, the estimation will be more accurate in term of reducing high variation in RTTs which commonly occur in the Internet. The other advantage of timer-less estimation is in the aspect of reducing problem of the timer coarse granularity, which commonly exist in protocol implementation. They claimed that their implementation is able to generate more retransmission within useful media playback decoding period, supressing unnecessary retransmission requests and retransmissions.

As mentioned in the introduction section that one of the bandwidth throughput fluctuation problem is caused by TCP's AIMD mechanism. The mechanism may delay delivering the streaming data by introducing throughput fluctuations when congestion occurs [3]. Therefore, one of the solutions to overcome this kind of weakness is to use other congestion control mechanism. For example, use other TCP flavors that do not use AIMD as the mechanism to control the congestion. Boyden et al. in [4] have suggested that TCP Vegas is better than TCP NewReno in many cases. According to them, being delay-based rather than loss-based, does not create the AIMD's saw-tooth pattern. Thus, this flavor shows better advantage in scenario of high random loss, and when there is much competing background traffic.

V. CONCLUDING REMARK

Many measurement studies indicate that more and more Internet users are experiencing streaming application. Active researches in this area show that the Internet users are expecting better performance. Many of the streaming media implementation weaknesses can be addressed by implementing appropriate buffer size and effective adaptive transmission rate. However, many more challenges need to be done to implement that ideas and to minimize further other weaknesses. For example, how to ensure consistent buffer-filling in dynamic changing network conditions? Therefore, many more fine-tuning or optimization need to be done to enable greater users satisfaction; especially in low-delay streaming application environment.

REFERENCES

[1] M. Prangl, I. Kofler, and H. Hellwagner, "Towards QoS Improvements of TCP-Based Media Delivery," presented at Networking and Services (ICNS 2008), pp. 188-193, 2008.

[2] H. Sun, A. Vetro, and J. Xin, "An Overview of Scalable Video Streaming: Research Articles," *Wireless Communications & Mobile Computing*, vol. 7, pp. 159-172, 2007.

[3] E. Brosh, S. A. Baset, D. Rubenstein, and H. Schulzrinne, "The Delay-Friendliness of TCP," *To appear in SIGMETRICS*, 2008.

[4] S. Boyden, A. Mahanti, and C. Williamson, "TCP Vegas Performance with Streaming Media," presented at Performance, Computing, and Communications Conference (IPCCC 2007), pp. 35-44, 2007.

[5] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the Internet," *IEEE/ACM Trans. Netw.*, vol. 7, pp. 458-472, 1999.

[6] L. Guo, E. Tan, S. Chen, Z. Xiao, O. Spatscheck, and X. Zhang, "Delving into internet streaming media delivery: a quality and resource utilization perspective," in *Proceedings of the 6th ACM*

SIGCOMM conference on Internet measurement. Rio de Janeiro, Brazil: ACM, 2006, pp. 217-230.

[7] B. Wang, W. Wei, J. Kurose, D. Towsley, K. R. Pattipati, Z. Guo, and Z. Peng, "Application-layer multipath data transfer via TCP: Schemes and performance tradeoffs," *Perform. Eval.*, vol. 64, pp. 965-977, 2007.

[8] M. Li, M. Claypool, R. Kinicki, and J. Nichols, "Characteristics of streaming media stored on the Web," *ACM Trans. Inter. Tech.*, vol. 5, pp. 601-626, 2005.

[9] M. Chesire, A. Wolman, G. M. Voelker, and H. M. Levy, "Measurement and Analysis of a Streaming-Media Workload," in *Proc. of USITS*, 2001.

[10] K. Sripanidkulchai, B. Maggs, and H. Zhang, "An analysis of live streaming workloads on the internet," in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*. Taormina, Sicily, Italy: ACM, 2004, pp. 41-54.

[11] B. Wang, J. Kurose, P. Shenoy, and D. Towsley, "Multimedia streaming via TCP: an analytic performance study," *SIGMETRICS Perform. Eval. Rev.*, vol. 32, pp. 406-407, 2004.

[12] S. Boyden, A. Mahanti, and C. Williamson, "Characterizing the Behaviour of RealVideo Streams," presented at SCS Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), Philadelphia, PA, USA, 2005.

[13] C. Krasic, K. Li, and J. Walpole, "The Case for Streaming Multimedia with TCP " in *Proceedings of the 8th International Workshop on Interactive Distributed Multimedia Systems* Springer-Verlag, 2001 pp. 213-218

[14] T. Phelan, "Strategies for Streaming Media Applications Using TCP-Friendly Rate Control," Internet Engineering Task Force (IETF), Internet Draft, 2007, work in progress.

[15] L. Guo, S. Chen, Z. Xiao, and X. Zhang, "Analysis of multimedia workloads with implications for internet streaming," in *Proceedings of the 14th international conference on World Wide Web*. Chiba, Japan: ACM, 2005, pp. 519-528.

[16] D. Loguinov and H. Radha, "End-to-end Internet video traffic dynamics: statistical study and analysis," presented at INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, pp. 723-732 vol.2, 2002.

[17] J. Chakareski and P. Frossard, "Adaptive systems for improved media streaming experience," *Communications Magazine, IEEE*, vol. 45, pp. 77-83, 2007.

[18] C. Huang, P. A. Chou, and A. Klemets, "Optimal Control of Multiple Bit Rates for Streaming Media," Microsoft Corporation, Redmond WA, Technical report, 2004.

[19] N. Ozbek and E. T. Tunali, "On optimal receiver buffer size in adaptive Internet video streaming," *Journal of Zhejiang University SCIENCE A*, vol. 7, pp. 112-118, 2006.

[20] T. Kim and M. H. Ammar, "Receiver Buffer Requirement for Video Streaming over TCP," *Silicon.com Whitepaper*, 2005.

[21] W.-t. Tan, W. Cui, and J. Apostolopoulos, "Playback-Buffer Equalization for Streaming Media using Stateless Transport Prioritization," presented at Packet Video Workshop, Nantes, France, 2003.

[22] S. Hasan, L. Lefevre, Z. Huang, and P. Werstein, "Cross Layer Protocol Support for Live Streaming Media," presented at Advanced Information Networking and Applications (AINA 2008), pp. 319-326, 2008.

[23] C. Krasic, "A Framework for Quality-Adaptive Media Streaming: Encode Once - Stream Anywhere," in *School of Science & Engineering*: Oregon Health & Science University, 2004.

[24] D. Lopez, F. Gonzalez, L. Bellido, and A. Alonso, "Adaptive multimedia streaming over IP based on customer oriented metrics," presented at Computer Networks, 2006 International Symposium, pp. 185-191, 2006.

[25] K. Ramakrishnan, S. Floyd, and D. Black, "The addition of explicit congestion notification (ECN) to IP," in *Request For Comment (RFC 3168)*: Internet Engineering Task Force (IETF), 2001.

[26] A. Raghuveer, E. Kusmierek, and D. H.-C. Du, "A Network-Aware Approach for Video and Metadata Streaming," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 17, pp. 1028-1040, 2007.

[27] R. Ludwig and R. Katz, "The Eifel algorithm: making TCP robust against spurious retransmissions," *SIGCOMM Comput. Commun. Rev.*, vol. 30, pp. 30-36, 2000.

[28] M. Kalman, E. Steinbach, and B. Girod, "Adaptive media playout for low-delay video streaming over error-prone channels," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 14, pp. 841-851, 2004.

[29] G.-M. Muntean, "Efficient delivery of multimedia streams over broadband networks using QOAS," *Broadcasting, IEEE Transactions on*, vol. 52, pp. 230-235, 2006.

[30] G.-M. Muntean, P. Perry, and L. Murphy, "Subjective Assessment of the Quality-Oriented Adaptive Scheme," *Broadcasting, IEEE Transactions on*, vol. 51, pp. 276-286, 2005.

[31] G.-M. Muntean and N. Cranley, "Resource Efficient Quality-Oriented Wireless Broadcasting of Adaptive Multimedia Content," *Broadcasting, IEEE Transactions on*, vol. 53, pp. 362-368, 2007.

[32] G.-M. Muntean, "Effect of Delivery Latency, Feedback Frequency and Network Load on Adaptive Multimedia Streaming," presented at Local Computer Networks, 2007. LCN 2007. 32nd IEEE Conference on, pp. 421-427, 2007.

[33] G.-M. Muntean, P. Perry, and L. Murphy, "A new adaptive multimedia streaming system for all-IP multi-service networks," *Broadcasting, IEEE Transactions on*, vol. 50, pp. 1-10, 2004.

[34] J. G. Apostolopoulos, W.-t. Tan, and S. J. Wee, "Video Streaming: Concepts, Algorithms, and Systems," in *Handbook of Video Databases, Design and Applications*: CRC Press, 2004.

[35] J. Widmer, R. Denda, and M. Mauve, "A survey on TCP-friendly congestion control," *Network, IEEE*, vol. 15, pp. 28-37, 2001.

[36] E. Kohler, M. Handley, and S. Floyd, "Datagram Congestion Control Protocol (DCCP)," in *Request For Comment (RFC 4340)*: Internet Engineering Task Force (IETF), 2006.

[37] E. Kohler, M. Handley, and S. Floyd, "Designing DCCP: Congestion Control Without Reliability," in *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*. Pisa, Italy: ACM, 2006, pp. 27-38.

[38] H. Elaarag and M. Bassiouni, "An Internet Friendly Transport Protocol for Continuous Media Over Best Effort Networks," *International Journal of Communication Systems*, vol. 15, pp. 881-898, 2002.

[39] T. Phelan, "Strategies for Streaming Media Applications Using TCP-Friendly Rate Control," Internet Engineering Task Force (IETF), Internet Draft (work in progress), 2007.

[40] D. Loguinov and H. Radha, "On retransmission schemes for real-time streaming in the Internet," presented at INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies., pp. 1310-1319, 2001.

[41] V. Jacobson, "Congestion Avoidance and Control," presented at ACM Sigcomm '88, Stanford, CA, pp. 314-329, 1988.

[42] R. Sinha and C. Papadopoulos, "An adaptive multiple retransmission technique for continuous media streams," in *Proceedings of the 14th international workshop on Network and operating systems support for digital audio and video*. Cork, Ireland: ACM, 2004, pp. 16-21.