DEALING WITH WIRELESS LINKS IN THE ERA OF BANDWIDTH DEMANDING WIRELESS HOME ENTERTAINMENT

Gustavo Marfia, Marco Roccetti

Dipartimento di Scienze dell'Informazione
Universita' di Bologna, Bologna, Italy, 40126
Email:{marfia|roccetti}@cs.unibo.it

ABSTRACT

Wireless home entertainment is strongly emerging in the consumer market. This is happening thanks to various number of home appliances that can connect wireless to the Internet, as witnessed by the recent launch of Sony Bravia. For this reason there is an interest, in the research community, to design and develop the protocol architecture of Home Entertainment Centers (HECs). HECs are entertainment hubs capable to handle heterogeneous media, thus behaving as gateways, and to support the main multimedia services (e.g., TVs, game consoles, etc.) at home. As the interactivity of these services is crucial to guarantee a satisfactory quality of service to the end user, it is important to understand which protocols HECs have to use to exchange data and how these perform on a home wireless link. Following recent findings that confirm that TCP still occupies the lion share of Internet residential traffic, the scope of this paper is to show that among the most used and popular candidates, TCP Westwood is capable to best adapt to a home scenario that entails an increase of the bandwidth requirements and a widespread use of wireless connections.

Keywords— Home, Entertainment, TCP, Wireless, Westwood

1. INTRODUCTION

The demand of interactive multimedia contents, under the form of online games and Internet Protocol Television (IPTV) at home, has set the stage for the development and marketing of wireless Home Entertainment Centers (HEC). Many high end appliances now integrate both a wireless interface and the capability of connecting directly to the Internet (e.g., the Sony Bravia displays content from Facebook, YouTube and Netflix [1], for example). In a near future, HECs will play the role of intermediaries between the multimedia servers and home appliances.

Essentially, a HEC represents a hub for all in-home entertainment experiences (Figure 1); it connects to the Internet and performs as a gateway between home client devices (e.g. Sony Bravia) and various sources of multimedia content. However, a HEC is more than a simple media gateway, combining services such as IPTV, Web radio, game console, picture viewer, electronic program guide, DVR, CD/DVD/video player, music jukebox, web browser, email handler and instant messenger. Contents related to these services can be locally available or dynamically retrieved from the Internet based on user's needs. Several streams are thus produced by these active services, all distributed by the HEC throughout the home. TiVo [3], Apple TV [4] and Windows Media Center [5] are all exemplars of what Home Entertainment Centers (HECs) can be.

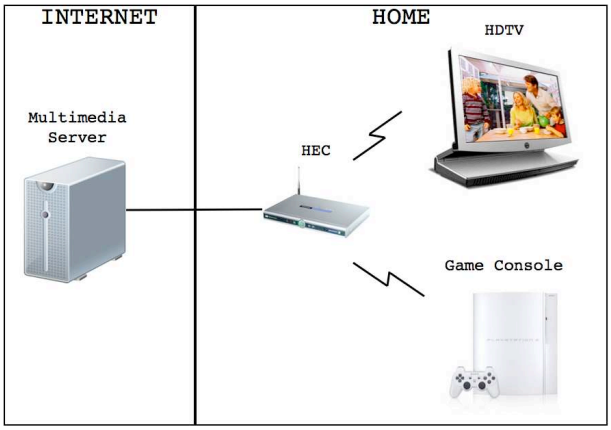

Fig. 1. Architecture of an interactive multimedia system based on a wireless Home Entertainment Center (HEC).

In general, a HEC can employ many different communication technologies to connect to client devices, however, wireless technologies today dominate over wired technologies due to their flexibility and mobility features and their high speed data rates. For this reason, HECs integrate an Access Point (AP), just as shown in Figure 1, which is exploited to guarantee wireless connectivity to user devices.

In such scenario, where the majority of multimedia traffic is distributed over a wireless last link at home, there is much interest to understanding how well the currently deployed protocol architecture supports an efficient delivery of multimedia streams. Surprisingly, while in this arena many experts and practitioners expected that UDP would have played as the main transport protocol of multimedia content on the Internet

This work was partially supported by the Italian FIRB DAMASCO project and by a grant of the Computer Science Department, University of Bologna.

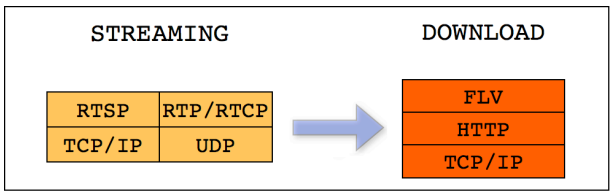

Fig. 2. Stack for multimedia services evolution: from RTP over UDP flows to FLV over HTTP messages.

at home, recent findings contradict such prediction [6]. In fact, even though TCP was born to guarantee a reliable and ordered transfer of packets and lacks of any support for applications with real-time constraints, recent measurement studies unveil that TCP is gaining momentum as the leading option for the streaming of multimedia [8]. Such phenomenon follows from the widespread choice made by system designers and developers who use Flash Video as the main container file format to deliver video over the Internet, as well as HTTP as the main application layer protocol for the progressive streaming and download of Flash Video data. As HTTP relies on TCP at the transport layer, TCP has grown to be, by far, the main protocol used for the transfer of multimedia content at home [6–8].

In particular, Figure 2 shows how the protocol stack employed to support a great amount of multimedia applications has evolved. While the IETF [10] had defined, with its working groups, a protocol stack for multimedia streaming mainly based on the use of RTP over UDP, the most amount of multimedia traffic does not, today, follow such protocol paradigm. RTP based multimedia flows, instead, amount to only the 1.5% of residential traffic, while this percentage grows to 15% in the case of FLV (i.e., Flash video) over HTTP streams [6]. This trend clearly depends on the widespread use, by popular websites such as YouTube and Google Video, of the Flash video technology. Moreover, this figure is bound to grow, as the Microsoft Internet Information Services [7] is performing similar protocol choices (i.e., HTTP) in its deployment of the Smooth Streaming technology to support HDTV flows.

However, moving one step back, TCP has been for long studied and analyzed for its inability to scale and efficiently utilize fat pipes and for its inefficiency on wireless links. In particular, its inefficiency on a wireless link derives from the fact that a random loss is interpreted as a congestion loss, resulting, in such case, in a useless reduction of its sending rate. Although the design flaws of the legacy TCP have led to endless debates on which is the best candidate to substitute it, the legacy TCP congestion control algorithm (TCP SACK [9]) still dominates the Internet scene [6]. Only two congestion control variants, BIC and CUBIC TCP [13, 14], have so far pushed their way to subsequently become the default congestion control schemes into the Linux networking stack since kernel versions 2.6.13 and 2.6.18, respectively. However, these algorithms still address the scalability issues of the legacy TCP on fat pipes, rather than its critical issues on wireless links.

Fortunately, the problem of delivering multimedia streams

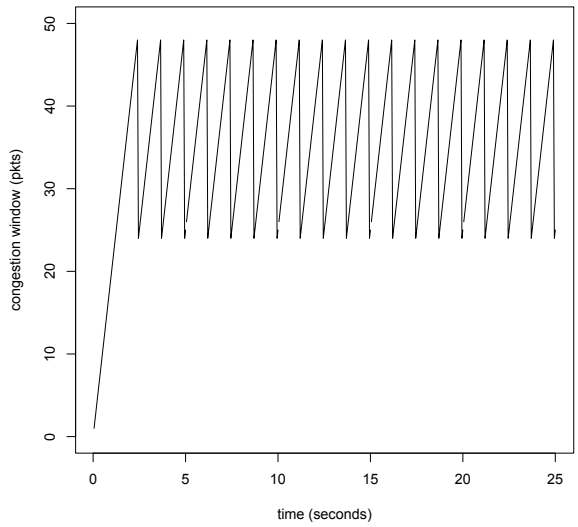

Fig. 3. Typical TCP SACK congestion window evolution, when no random error losses are experienced during the connection.

at home can be easily addressed within a HEC. In fact, as all streams flow through a HEC, and most of these streams are encapsulated into HTTP packets, a HEC can overcome the problems deriving from the use of TCP SACK in multimedia servers by implementing a HTTP proxy and a TCP algorithm optimized for wireless links. This approach simply requires: i) the use of a HTTP proxy, which is straightforward to deploy, and; ii) the choice of the most adequate TCP scheme to be used at the HEC.

Since the latter is trickier, the remainder of this paper will mainly focus on describing how we selected what we believe is the best TCP choice for a wireless HEC.

Therefore, the objective of this work is twofold: (a) provide an understanding of the inefficiencies that are caused at wireless HECs by the use of the legacy TCP protocol, and; (b) formulate a proposal for an architecture, based on an advanced TCP version, optimized for wireless transmissions, which proves viable to increase the utilization of the wireless bandwidth by a HEC.

To perform (b), we will rely on analytical results which give the average throughput of TCP Westwood Buffer and Bandwidth Estimation (TCPW-BBE) [18], and therefore explain why this TCP flavor may represent a good candidate as a HEC TCP version.

This paper is organized as follows: in Section 2 we provide a background on the research that has involved wireless HECs in the past; in Section 3 we will reveal the details of our solution, discussing both analytically and experimentally, why we believe that TCP Westwood may be the right HEC TCP candidate. Section 4 provides simulations results which confirm the validity of our work. Finally, Section 5 concludes and suggests future directions of research in this domain.

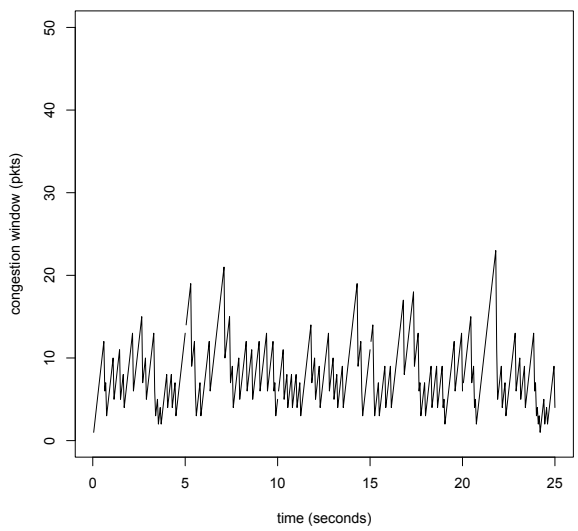

Fig. 4. Congestion window degradation when the connection experiences a significant number of random losses.

2. BACKGROUND AND PROBLEM STATEMENT

An amount of research in the scientific community has focused its attention to the delivery of multimedia content through wireless Home Entertainment Centers. An interesting result is reported in [20], where the authors study the interactions between UDP and TCP flows at HECs and highlight that an excessive aggressiveness of TCP flows can hurt media flows based on UDP, thus causing interactivity problems. In particular, they showed that the sending window of competing TCP flows can reach values that create severe UDP segment losses and delays. To reduce the aggressiveness of TCP, they implement a mechanism which overwrites, in order to limit its maximum value, the advertised window of a TCP connection at an access point. In this way, both UDP flows and TCP flows are accommodated through a HEC, freezing the growth of the legacy TCP sending window when this hurts the interactivity required by multimedia flows. However, this work expected that the main streaming and gaming architectures would have entailed a content delivery mainly based on the use of RTP over UDP flows. Instead, as discussed in Section 1, the challenge today is not of how to protect UDP flows from aggressive TCP flows at the last wireless hop, rather how to guarantee an efficient delivery of multimedia content, over a wireless link, with TCP.

TCP flows, in fact, do not perform as one would hope over wireless links. When a segment is sent, a TCP sender waits for a returning ACK to increase its sending window. Depending on how many ACKs are lost, and after retransmitting the lost segments and receiving their acknowledgements, TCP adjusts its sending window to one of two possible values. If many segments were lost during the connection and this triggered a retransmission timeout, the sending window is set to one, as the loss is interpreted as a heavy congestion event, caused by pushing too much data on the data pipe. If, instead, the segments were recovered by the Fast Recovery, Fast Retransmit algorithm of TCP, the loss is interpreted as due to a light congestion event, in which case the sending window is set to half of its original value.

TCP Protocol	Average throughput	Congestion measure
TCP SACK	$\propto \sqrt{\frac{1-\bar{\lambda}}{\bar{\lambda}}}$	$\bar{\lambda}$ = loss probability
CUBIC	$\propto \sqrt[4]{\frac{1}{\bar{\lambda}^3}}$	$\bar{\lambda}$ = loss probability
BIC TCP	$\propto \frac{1}{\bar{\lambda}^d}, \quad 1/2 < d < 1$	$\bar{\lambda}$ = loss probability

Fig. 5. Main TCP versions behavior.

In either case, a segment loss is interpreted as a congestion event. This is, in fact, the most frequent cause of packet loss over wired networks, where a random loss occurs with a very low probability (e.g., 10^{-8} on optical links). On the contrary, a random loss becomes much more probable when links are wireless, clearly due to interference problems that may origin from multi-path and other disturbances. When such events occur, it is of no use to lower the TCP sending window, as the cause of loss and the sending window value are uncorrelated. To further clear the issue, please consider the TCP SACK congestion windows plotted in Figures 3 and 4. Both of these figures represent the sending window evolution at a TCP SACK source, therefore the number of sent packets (y-axis) as a function of time (x-axis). In Figure 3, the sawtooth is the typical pattern of TCP SACK when losses are only due to reaching the maximum amount of packets that can fit through the pipe. In Figure 4, instead, there is a 2% chance that a packet may be randomly dropped. Comparing the two figures, it is clear that the second scenario inefficiently uses the wireless resource, as the amount of data sent oscillates around a very low value because the TCP sender assumes that losses are due to congestion, while this is not the case.

The Table shown in Figure 5 summarizes the behavior expected from the main TCP versions when a loss occurs. As we can see, both TCP SACK, CUBIC and BIC TCP tune their sending rate as a function of the loss rate λ, without enforcing any mechanism which distinguishes a random loss from a congestion loss. This means that the sending rate of such protocols can be hurt on wireless links. In order to combat such problem, protocols such as TCP Westwood [15] introduce a mechanism where, observing the dynamics of the round trip time of TCP segments, random and congestion losses can be distinguished.

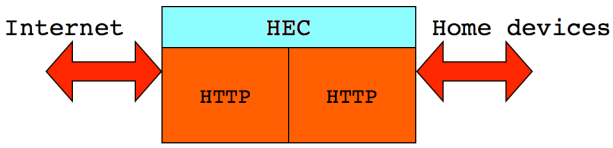

Fig. 6. Proposed HEC architecture.

In our scenario, one step forward can be made with the use of a HEC. In the following we will show how a HEC can be simply adapted to optimize the performance of multimedia flows on a wireless link. In particular, we will show how a careful design of a HEC can lead to a better utilization of the available bandwidth and, therefore, to an increase efficiency. In the following we will define a HEC protocol architecture and show, both analytically and experimentally, that this protocol architecture also represents a good compromise between simplicity and efficiency.

3. A PROTOCOL SOLUTION FOR A WIRELESS HEC

The TCP inefficiency problems on wireless links can be simply overridden deploying a HTTP proxy at a HEC. In fact, in this way, any TCP connection between a multimedia server, in the Internet, and a home device is broken at the HEC. The advantage of such solution is that the TCP sending window on the wireless link is governed by the TCP version deployed at the HEC. Figure 6 shows a protocol stack where the home HTTP connection with the Internet is broken at the HEC. This implies that, by deploying a TCP variant at the HEC, it is possible to drive the amount of packets that are sent over the home wireless link.

The problem, at this point, becomes a TCP variant selection problem. In the following we will show why TCP Westwood Buffer and Bandwidth Estimation [18] represents a good candidate to deal with wireless links.

3.1. TCP Westwood Buffer and Bandwidth Estimation

TCP Westwood names a family of protocols [15–18] designed with the main scope of improving the efficiency in wired-wireless combined networks with non-negligible random packet losses, and also friendliness to legacy protocols, such as TCP SACK. All the algorithms in this family are sender side only modifications to the legacy TCP scheme. For the scope of this paper, we refer to TCP Westwood Buffer and Bandwidth Estimation (TCPW-BBE), first presented in [18]. Unlike preceding versions of TCP Westwood, TCPW-BBE dynamically determines the effective bottleneck buffer by estimating the maximum round trip time of the connection, RTT_{max}, and uses this information to limit its aggressiveness towards TCP SACK. Thanks to this peculiarity, TCPW-BBE is both efficient when utilizing wireless links and friendly to the legacy TCP protocol. In the remainder of this Section we will briefly describe how the algorithm works, and, then, provide the result of our analytical study of TCPW-BBE, which guarantees it effectively well behaves when a random loss occurs.

3.2. Algorithm

TCPW-BBE, whose mechanism is in detail described in Algorithm 1, works as TCP SACK when increasing its sending window (line 1.10). When a packet loss is detected from three duplicate acknowledgements (lines 1.13-1.15), instead, TCPW-BBE differs from TCP SACK in attempting to distinguish a random loss from a congestion loss. This is performed in a very simple way: when the experienced packet round trip time is high, this is interpreted as a congestion loss. In fact, high round trip time values typically coincide with full network buffers and, therefore, congestion. If, instead, the round trip time is low, the loss is interpreted as a random loss. In the following we explain in more detail how this is implemented.

1.1 $cwnd_n \Leftarrow$ congestion window at step n
1.2 $RTT_n \Leftarrow$ round trip time at step n
1.3 $k \Leftarrow$ reduction factor
1.4 $u \Leftarrow$ random loss indicator
1.5 $BE \Leftarrow$ narrow link capacity estimate
1.6 $RE \Leftarrow$ achieved rate
1.7 $ssthres_{n+1} \Leftarrow$ new slow start threshold
1.8 $cwnd_{n+1} \Leftarrow$ new congestion window
1.9 **if** *a packet is successfully delivered* **then**
1.10 | $cwnd_{n+1} \Leftarrow cwnd_n + \frac{1}{cwnd_n}$
1.11 **end**
1.12 **else**
1.13 | **if** *3 DUPACK* **then**
1.14 | | $cwnd_{n+1} \Leftarrow$ $RTT_n \times k \times (u \times BE + (1-u) \times RE)$ $ssthres_{n+1} \Leftarrow$ $RTT_n \times k \times (u \times BE + (1-u) \times RE)$
1.15 | **end**
1.16 | **if** *timeout* **then**
1.17 | | $cwnd_{n+1} \Leftarrow 1$ $ssthres_{n+1} \Leftarrow cwnd_n/2$
1.18 | **end**
1.19 **end**

Algorithm 1: Congestion Window Update in TCPW-BBE

In Algorithm 1 we have two important parameters, k and u. k (line 1.3) is a reduction factor, that ranges between $1/2$ and 1. Basically, when the round trip time at step n is close to the maximum round trip time experienced during the connection, $k = 1/2$. In the opposite case, $k = 1$. u (line 1.4), as k, depends on the actual, minimum and maximum round trip time estimates and plays the role of distinguishing a random loss from a congestion loss. In fact, u ranges between 0 and 1 and when a random loss occurs $u = 1$, while if a congestion loss occurs $u = 0$. It is then clear that u modulates the sending window between two values, BE and RE (line 1.14). In fact, as also mentioned in Algorithm 1, BE (line 1.5) represents the narrow link capacity (i.e., the maximum achievable sending rate), while

RE (line 1.6) represents the average rate so far experienced by the TCP connection. So, if a random loss occurs, TCPW-BBE attempts to push as many packets as possible through the pipe, while if a congestion loss occurs, it safely rolls back to the average sending rate.

Finally, if a severe loss of packets occurs and this is observed through a timeout event, TCPW-BBE behaves as TCP SACK, halving the slow start threshold and setting the congestion window to one (lines 1.16-1.18).

As a result of this description, it should now be clear that TCPW-BBE has been designed to well behave when a random loss occurs. In fact, when a random loss occurs, and this is detected as such (i.e. $RTT = RTT_{min}$), the congestion window is set to a value close to the maximum achievable sending rate. This is a very useful feature for a HEC, as a higher bandwidth utilization translates into a higher number of multimedia flows that can be supported at home.

3.3. Measuring the Throughput

We now motivate why TCPW-BBE is so successful when dealing with wireless links and, therefore, why it could be a good choice for a HEC networking stack. To do this, we quantify the average throughput achieved by a TCPW-BBE flow when packet losses are mainly due to congestion (i.e., $RTT = RTT_{max}$) and when, instead, they are mainly random (i.e., $RTT = RTT_{min}$). For the sake of clarity, we skip the derivation of these formulas which may be found in [23] and directly step to the results.

When the $\bar{RTT} = RTT_{max}$ we have that:

$$\bar{x}_{RTT_{max}} \approx \frac{1}{3 \times RTT_{max}} \sqrt{\frac{6}{\bar{\lambda}}}, \quad (1)$$

while, when the $\bar{RTT} = RTT_{min}$, the average sending rate is:

$$\bar{x}_{RTT_{min}} \approx \frac{\bar{BE}}{2} + \frac{1}{2} \times \sqrt{\bar{BE}^2 + \frac{4}{\bar{\lambda} \times (RTT_{min})^2}}, \quad (2)$$

where we see that, in case packets are lost due to wireless random errors, $\bar{x}_{RTT_{min}} > \bar{BE}$. This surprising result, as obviously the rate cannot exceed the narrow link capacity (i.e., the maximum physical capacity of the data route), means that TCPW-BBE in practice ignores random losses.

This analysis lets us conclude that TCPW-BBE scales, attempting to occupy the entire pipe when the $RTT = RTT_{min}$, and, does not suffer from the inefficiencies due to an erroneous reduction of the sending rate when random losses occur. It is, then, a valid candidate for a HEC implementation, as we will show in practice in the following.

4. EVALUATION

Let us now first qualitatively see how TCPW-BBE performs on the same scenario exploited in Figure 4 and then move on to a practical example. Figure 7 shows how the congestion window evolves, in the same situation experienced in Figure 4. TCPW-BBE is capable to distinguish a congestion loss from a random loss, and using this information, is capable of sustaining an average congestion window that is visibly higher than the one experienced with TCP SACK.

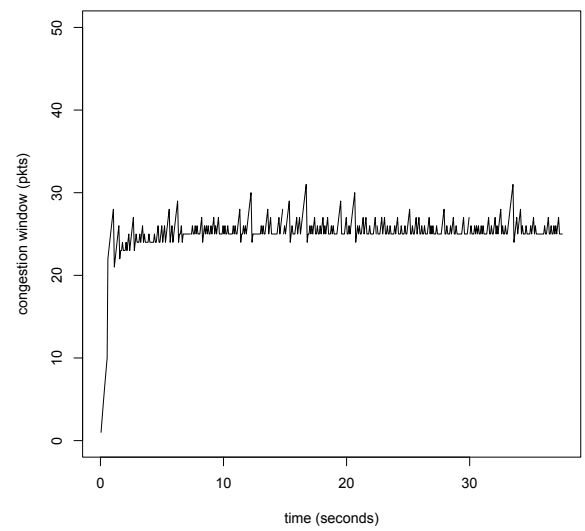

Fig. 7. TCPW-BBE congestion window in the same scenario of Figure 4.

We now set plausible values for a scenario as the one depicted in Figure 6. In general, the two-way propagation delay between a HEC and a home device is very low, we set this value to 1 millisecond. With a 100Mbps connection, we find that the average improvement, in terms of throughput, is of about 30%. We may also confirm this analytically. In fact, assuming $\bar{\lambda} = 2 \times 10^{-2}$, $\bar{BE} = 100Mbps$, Equations 1 and 2 we find:

$$\bar{x}_{TCP\ SACK} = 5741pkt/s = 68.9Mbps, \quad (3)$$

and

$$\bar{x}_{TCPW-BBE} = 8600pkt/s = 103.2Mbps. \quad (4)$$

Clearly, it is not possible that the average rate of TCPW-BBE exceeds the physical capacity of the connection, as TCPW-BBE pushes more packets, both $RTT(t)$ and $\lambda(t)$ will increase, giving higher average values and consequently an average rate that is below 100Mbps.

5. CONCLUSION

We here presented multimedia content delivery architecture, including a wireless Home Entertainment Center, based on TCPW-BBE. Using an analytic approach, we demonstrated that TCPW-BBE can be the ideal choice for the delivery of multime-

dia content over wireless links, as it is capable of distinguishing a random loss of a packet from a congestion loss.

6. REFERENCES

[1] http://news.cnet.com/8301-17938_105-10282740-1.html

[2] http://www.netflix.com/NRDInfo/Wii

[3] http://www.tivo.com

[4] www.apple.com/appletv/

[5] http://www.microsoft.com

[6] Backstrom, L ., Huttenlocher , D., Kleinberg, J ., Maier , G., Feldmann, A., Paxson, V., Allman, M., "On dominant characteristics of residential broadband internet traffic". In *Proc. of the 9th ACM SIGCOMM Conference on internet Measurement Conference (IMC'09).*

[7] http://www.iis.net/expand/SmoothStreaming

[8] Qian, F., Gerber, A., Mao, Z. M., Sen, S., Spatscheck, O., Willinger, W., "TCP revisited: a fresh look at TCP in the wild". In *Proc. of the 9th ACM SIGCOMM Conference on internet Measurement Conference (IMC'09).*

[9] Fall, K., Floyd, S., "Simulation-based comparisons of Tahoe, Reno and SACK TCP". In *SIGCOMM Computer Communication Review*, 26, 3 (Jul. 1996), 5-21.

[10] http://www.ietf.org

[11] http://www.youtube.com

[12] http://video.google.com

[13] Xu L., Harfoush, K., Rhee I., "Binary increase congestion control (BIC) for fast long-distance networks". In *Proc. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'04).*

[14] Ha, S., Rhee, I., Xu, L., "CUBIC: a new TCP-friendly high-speed TCP variant". In *SIGOPS Operating Systems Review*, 42, 5 (Jul. 2008), 64-74.

[15] Mascolo, S., Casetti, C., Gerla, M., Sanadidi, M. Y., and Wang, R. 2001. "TCP westwood: Bandwidth estimation for enhanced transport over wireless links". In *Proc. of the 7th Annual international Conference on Mobile Computing and Networking (MobiCom'01).*

[16] Wang, R., Valla, M., Sanadidi, M.Y., Ng, B. K. F., Gerla, M., "Efficiency/Friendliness Tradeoffs in TCP Westwood". In *Proc. of the Seventh IEEE Symposium on Computers and Communications (ISCC'02)*, Taormina, Italy, 1 - 4 July, 2002.

[17] Wang, R., Valla, M., Sanadidi, M.Y., Gerla, M., "Adaptive Bandwidth Share Estimation in TCP Westwood". In *Proc. IEEE Global Communications Conference (Globecom'02)*, Taipei, Taiwan, R.O.C., November 17-21, 2002.

[18] Shimonishi, H., Sanadidi, M.Y., Gerla, M., "Inproving Efficiency-Friendliness Tradeoffs of TCP in Wired-Wireless Combined Networks". In *Proc. of the IEEE International Conference on Communications (ICC'05).*

[19] R. Srikant, "The Mathematics of Internet Congestion Control". A Birkhauser book, 2004.

[20] Palazzi, C. E., Ferretti, S., Roccetti, M., Pau, G., Gerla, M., "What's in that Magic Box? The Home Entertainment Center's Special Protocol Potion, Revealed". In *IEEE Transactions on Consumer Electronics*, IEEE Consumer Electronics Society, vol. 52, no. 4, Nov 2006, 1280-1288.

[21] Kelly, F., "Charging and Rate Control for Elastic Traffic". In *European transactions on telecommunications*, 1997, 8 (1) : p.33-37.

[22] Palazzi, C., Kaiser, A., Achir, N., Boussetta K., "A Preliminary Evaluation of Backup Servers for Longer Gaming Sessions in MANETs", in *Proc. of the International Workshop on DIstributed SImulation & Online gaming (DISIO 2010) - ICST SIMUTools 2010*, Torremolinos, Spain, Mar 2010.

[23] Marfia, G., "P2P Intelligent Transportation Systems: Mobility, Fairness and Scalability", Ph.D. Thesis in Computer Science, University of California, Los Angeles, 2009.

[24] http://www.akamai.com/