TCP At Last: Reconsidering TCP's Role for Wireless Entertainment Centers at Home

Gustavo Marfia, and Marco Roccetti

Abstract— *The amount of investments and research that led to the launch of new high definition TV sets and game consoles witness the importance gained by the home entertainment segment of the consumer market. Moreover, many entertainment appliances connect wireless to the Internet, providing users with the freedom of enjoying them at their preferred location at home. As this happens, there has been a widespread interest for the design of Home Entertainment Centers (HECs). HECs play, in fact, a key role as they are the media gateways between the home appliances and the Internet, thus enabling the distribution of various services based on different media streams at home. As most media flows are today based on the Transmission Control Protocol (TCP), and most entertainment services are distributed via wireless, wireless HEC designers face the problem of devising a protocol architecture that avoids the disruption of media flows and, consequently, of multimedia services. We here propose a solution to this problem and support its effectiveness with both analytical and simulation results.*[1]

Index Terms — Wireless Home Entertainment Center, Consumer Multimedia, TCP for Wireless Multimedia.

I. INTRODUCTION

The marketing of wireless Home Entertainment Centers (HECs) stems from an on growing demand of interactive multimedia content, which includes home gaming and TV over the Internet Protocol (IPTV). In fact, IPTV and many gaming devices today support an exchange of data based on wireless interfaces and enable, via wireless, both the capability of interacting with multimedia servers and the possibility of browsing any type of content from the Internet. Many are the exemplars of such phenomenon which can connect wireless to the Internet and display very popular contents (including movies and games) coming from various distribution platforms [1]-[3].

Basically, a HEC is a home media gateway which: (a) connects to the Internet, and; (b) provides a unique interface for home client devices. Therefore, it is the single point of reference for all multimedia contents received from the network and, sent from home. As such, it can support heterogeneous services ranging from TV streaming to web browsing, from web radio broadcasting to digital video recording, from instant messaging to social networking, for example. It can also act as a cache of multimedia contents, thus allowing a device to locally retrieve that content, hence reducing the latency (in case it is very popular, for example). Several commercial examples of these tools are already available on the consumer market and are gaining larger shares of customers [4]-[6]. Among these, some already support all the main audio, video and image coding formats, are compatible with all main widescreen HDTVs and provide hundreds of Gigabytes of disk space for the caching of multimedia contents at home.

A HEC can generically support different wireless or wired technologies to connect to the home devices, but the use of wireless technologies is preferred because of its flexibility. As an example, new gaming and HDTV devices equipped with a WiFi 802.11g/n standard interface can easily connect to commercial HECs, which can, therefore, easily enable multimedia services everywhere within a home. As this is the current state of the art for many commercially available HECs, we here assume that HECs are generally integrated with wireless access.

It is now clear that a key point to improve the quality of the experience users enjoy is to understand how to optimize the distribution of heterogeneous contents through a wireless HEC at home. In the past, an amount of research has been carried out in this field, based on the assumption that the majority of multimedia streams would have been carried over the Internet employing the User Datagram Protocol (UDP) at the transport layer [7]. However, this prediction has proven to be incorrect, as recent findings show that the majority of audio and video streams use the Transmission Control Protocol (TCP) [8]. Although TCP has been designed long ago to provide a reliable scheme for elastic (e.g., downloading) applications, and therefore does not well apply to the case of real time streams, recent measurements have shown that it is still the leading choice for streaming multimedia content to residential networks [9].

In practice, the lion share bandwidth exploited by TCP flows is due to a widespread use of the Flash Video (FLV) format as the main container for the delivery of multimedia streams at home [8]. As FLV is often encapsulated into the HyperText Transfer Protocol (HTTP), TCP has

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

G. Marfia is with the University of Bologna, Bologna, 40127 Italy (e-mail: marfia@cs.unibo.it).

M. Roccetti (reference author) is with the University of Bologna, Bologna, 40127 Italy (e-mail: roccetti@cs.unibo.it).

Contributed Paper
Manuscript received 08/31/10
Current version published 12/23/10
Electronic version published 12/30/10.

consequently become the leading transport protocol for multimedia streams. In support of this, consider that the majority of video streams over the Internet rely on the FLV Video and also many software companies are adopting similar protocol solutions to encapsulate HDTV flows into HTTP [10], [11].

Fig. 1 illustrates how multimedia contents could be transported: either i) exploiting a protocol stack based on the Real Time Protocol (RTP) encapsulated into UDP packets, as suggested by the Internet Engineering Task Force (IETF) organization, or ii) based on HTTP with TCP as transport protocol, as explained before.

As it often happens within consumer markets where new *de facto* solutions often override accepted standards, the protocol stack defined by the IETF has been beaten by a less efficient, but simpler solution [11]. To witness this, recent measurement studies have unveiled that the total RTP share amounts to only the 5% of residential Internet traffic, whereas FLV over HTTP alone gains more than the 15% of overall traffic [8].

It is then clear that TCP is the main transport protocol of multimedia contents towards our homes and, therefore, it is necessary to overcome its inefficiencies that can lead to a poor user experience or to service disruption. In fact, TCP was born with the main objective of supporting the transfer of data over wired networks. For this reason it has been designed assuming that a packet loss was always due to a congestion loss. As this statement is generally true for wired networks, where the chance of incurring in a random loss is as low as once every one hundred million, this is false for wireless networks. Here, instead, the main cause of packet loss is interference and not congestion.

Although new schemes have been deployed with the scope of slowly replacing the legacy TCP congestion control algorithm with algorithms that more efficiently utilize the available network resources, these do not address the TCP inefficiencies over wireless links [12]. For example, new TCP variants, as those recently adopted into the Linux networking stack, are not able to distinguish a congestion loss from a random loss [13], [14].

At this point, the true problem is to find a TCP version that distinguishes congestion losses from random losses and, combats the latter proficiently. The scope of this work, then, is to describe a TCP version, which results the most suitable to be deployed at HECs. In particular, we will discuss a HEC protocol architecture, which, deploying a suitable TCP, can combat the service inefficiencies caused by wireless random losses.

Indeed, we will provide both analytical and experimental evidence that TCP Westwood can be a good candidate for this scope [15]. In particular, we will show it supports higher data rates than the legacy TCP when random losses dominate.

The remainder of the paper is organized as follows. Section II provides the necessary research background on wireless HECs and its typical protocol problems, as well. Section III discusses the HEC protocol architecture we propose based on TCP Westwood, along with a mathematical analysis revealing the main characteristics of this TCP variant. Section IV confirms with simulation results the validity of our approach.

II. BACKGROUND AND PROBLEM STATEMENT

In the past, for the reasons explained in Section I, the most amount of research in the wireless HEC domain has focused on optimizing a multimedia distribution architecture where multimedia streams were delivered with UDP. An interesting result, in this direction, is given by the work presented in [7]. The authors, in fact, highlight that fairness problems arise from the coexistence of TCP flows and UDP flows on the same wireless link. As they assume that multimedia flows are

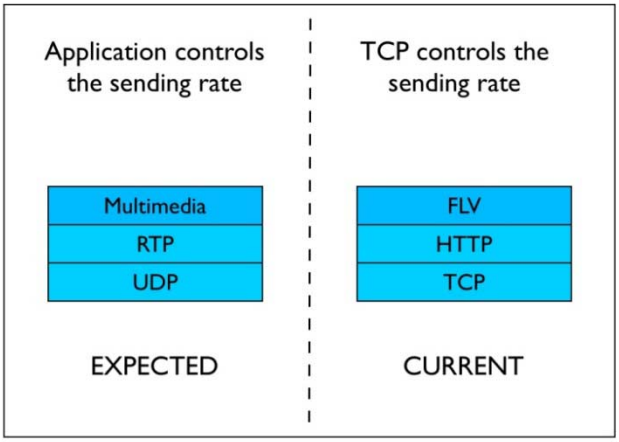

Fig. 1. Multimedia services stack evolution: RTP over UDP vs. FLV over HTTP.

based on UDP, they show that these flows can be hurt by TCP streams. To deal with this problem they engineer a mechanism that overrides the TCP advertised window at HECs and, therefore, limits the TCP sending window growth. They show their scheme is effective, as the interactivity required by the stream transported by a UDP protocol is restored to the same values achieved when no TCP connections compete on the wireless link.

Another prominent approach amounts to the Video Transport Protocol (VTP), which introduces an end-to-end mechanism that efficiently tunes the sending of UDP packets over wireless links at home [16]. In particular, they exploit a sender side scheme which: a) ensures a friendly coexistence of UDP and TCP flows, and; b) increases the multimedia flows utilization of a wireless link. These results are achieved implementing, at the application layer, a set of ad hoc mechanisms devised with this aim.

However, both of the approaches mentioned before fall short in recognizing that the majority of multimedia flows are currently transported by TCP, and not by UDP. Consequently, a better utilization of the home wireless links needs to be rethought in the light of this.

There is a major problem with TCP. In fact, TCP flows were not designed to efficiently utilize wireless links. The legacy TCP addresses two major issues: reliability and

congestion control. To achieve the second goal, TCP adapts the sending rate to avoid network overflow or service starvation.

In particular, the most popular TCP version, termed TCP SACK, implements a congestion control algorithm, which falls into the AIMD (Additive Increase, Multiplicative Decrease) family of algorithms. The very basic concept can be summarized as follows: (a) when a packet loss is detected, the TCP sender halves its sending window; (b) when a packet is successfully delivered, the TCP sender increases its sending window by one packet. In summary, the sending rate of TCP is determined by the rate of incoming packet acknowledgments (ACKs). Consequently, at the steady state, the sending rate of a TCP source will match the arrival rate of the related ACKs. Therefore, the congestion control mechanisms triggered by a packet loss guarantee that the protocol automatically detects congestion, a reaction to which is the decrease of the TCP sending rate.

Unfortunately, we have a big problem here: as a packet loss is always interpreted as a congestion sign, a TCP SACK sender will always lower its sending window when a loss is detected. While this behavior is acceptable on wired networks, where congestion is the primary cause of packet losses, this is not the case with wireless links, where interference and multi-path are often the causes of packet loss (i.e., random loss). Clearly, in such case, the behavior of TCP determines as a negative effect that the sending window is halved with each single random loss, thus deteriorating the total throughput.

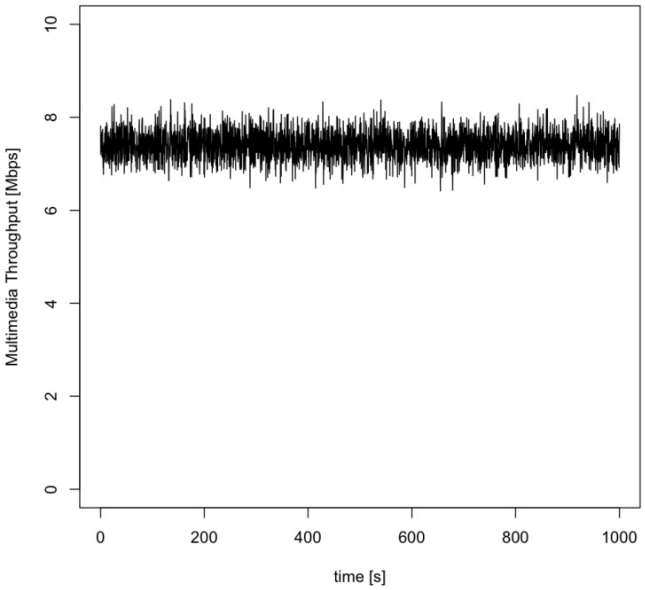

Fig. 2. Multimedia throughput with TCP SACK and no random losses.

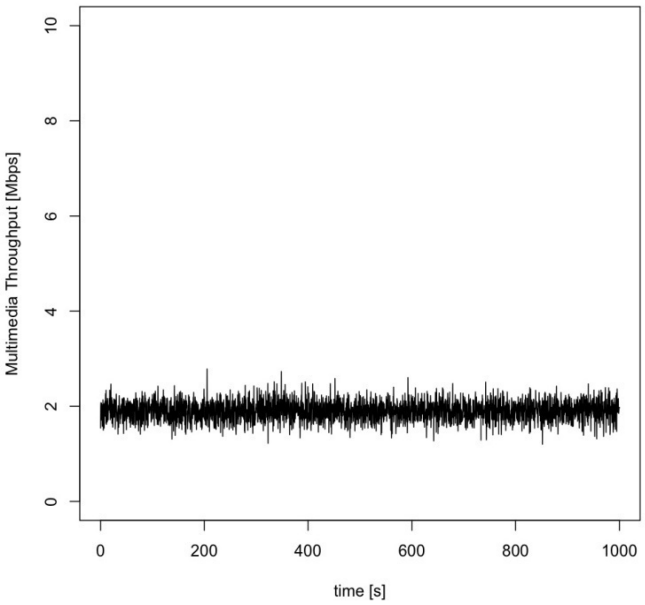

Fig. 3. Multimedia throughput with TCP SACK and random losses.

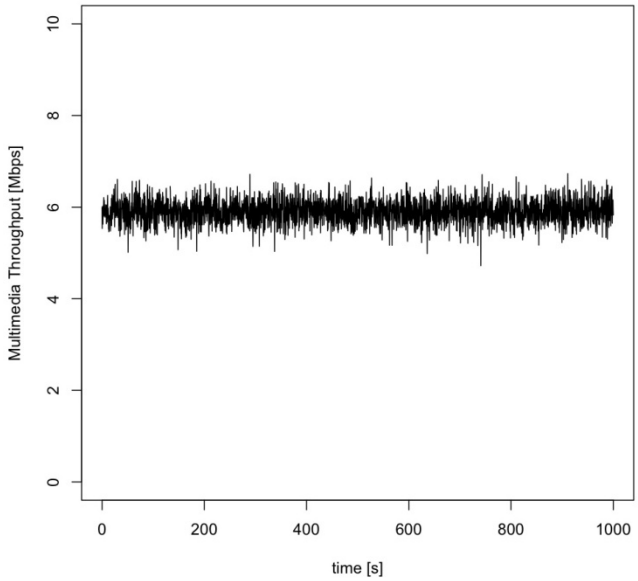

Fig. 4. Multimedia throughput with TCP Westwood and random losses.

To better clarify this phenomenon, the behavior of TCP SACK on a 10Mb/s wired and on a 10Mb/s wireless connection is respectively shown in Fig. 2 and in Fig. 3, where a single legacy TCP flow dynamics is represented.

While the throughput achieved by TCP SACK in a wired network meets the expectations (8Mb/s), as shown in Fig. 2, instead over a wireless link with only the 2% of random losses the throughput dramatically drops to 2Mb/s, as depicted in Fig. 3. Unfortunately, almost all TCP proposals have been designed so far with a notion of packet loss strongly dependent on congestion in mind. Hence, each of them suffers from throughput degradation, which is a function of the overall packet loss probability. Without providing further details on this issue, it suffices here to mention that it has been proven that the throughput of the most popular TCP versions in general deteriorates as the square root of packet loss [12]-[14], [17].

At this point, the question is if any hope exists to exploit a TCP variant, which is able to distinguish between random, and congestion losses. A positive answer to this question would allow us to incorporate such TCP implementation into a HEC architecture without any performance degradation when multimedia flows are distributed throughout the home. We have some good news: the right TCP candidate exists. Its name is TCP Westwood. We anticipate here the positive effect that TCP Westwood can have in the wireless HEC scenario by resorting to Fig. 4, which show that TCP Westwood can achieve a throughput which meets the expected quality standard (e.g., 6Mb/s) with the same conditions as those of Fig. 3.

In the following Section III we will explain how TCP Westwood works and why this is the right choice for architecting wireless HECs.

III. Architecting a Wireless HECs: A Viable Solution Based on TCP Westwood

We have already anticipated in the previous Section that TCP Westwood is a proper answer to the problem of architecting a wireless HEC that does not suffer of throughput degradation due to random losses. From now on, we will explain why.

In particular, we will here refer to a specific variant of TCP Westwood, termed TCP Westwood Buffer and Bandwidth Estimation, TCPW-BBE for short [15]. Although TCP Westwood groups a family of protocols whose main scope is to improve the efficiency in wired-wireless combined networks with a non negligible random packet loss, we look at TCPW-BBE because, among the many TCP Westwood versions, it shows the best combination in terms of friendliness to the legacy TCP and efficiency on wireless links [15],[18]-[20].

Anticipating the main issue, TCPW-BBE (from now on TCPW for the sake of brevity) limits its aggressiveness in lowering the sending window by exploiting information concerning the maximum round trip time of the connection. For this reason, and because when a random loss is detected TCPW sets the sending window equal to the connection pipe size (i.e., the maximum amount of packets that may be sent without causing a congestion loss), it is easy to imagine that TCPW can be considered a potential candidate to fix our problems, as discussed in the following sections.

Before showing how TCPW works, it is now worth mentioning that deploying any TCP variant is not a problem in this framework. In fact, it is possible to separate the Internet from the home network by simply exploiting the HEC to install a HTTP proxy at home. In this way, all the HTTP based traffic breaks at the proxy: the legacy TCP drives the packets received from the Internet to the proxy, while the TCP version deployed at the proxy drives the packets sent onto the home wireless network.

A. TCP Westwood at Work

We here show how TCPW works by discussing the algorithm that implements it (referred to as Algorithm 1 and reported in Table I).

Let us move directly to the core of the algorithm that is the congestion avoidance phase. When a TCPW sender receives an acknowledgement, it behaves as a legacy TCP sender, increasing the sending window by one every round trip time (line 10 in Algorithm 1 of Table I). When congestion is revealed by three duplicate acknowledgements (lines 12-14), TCPW differs from TCP SACK in implementing a mechanism that distinguishes a random loss from a congestion loss.

The rationale behind this mechanism is as follows. Intuitively, as the amount of congestion increases, the network buffers occupancy rate increases and, therefore, the packet round trip time grows. For this reason, when a loss occurs and the corresponding packet round trip time is high, TCPW interprets the loss as a congestion loss and behaves as the legacy TCP (both the sending window and the slow start threshold are halved). Instead, when a loss is experienced but the round trip time is low, TCPW interprets this as a random loss. In this case, the sending window and the slow start threshold are set equal to the pipe size. The algorithm takes into account this phenomenon by tuning two key parameters: the reduction factor k and the random loss indicator u.

TABLE I
ALGORITHM 1: SENDING WINDOW UPDATE IN TCPW

Line	Statement
1	$CW_n \Leftarrow$ sending window at step n
2	$RTT_n \Leftarrow$ round trip time at step n
3	$k \Leftarrow$ reduction factor
4	$u \Leftarrow$ random loss indicator
5	$BE \Leftarrow$ bandwidth estimate
6	$RE \Leftarrow$ rate estimate
7	$ssthresh_n \Leftarrow$ new slow start threshold
8	$CW_{n+1} \Leftarrow$ new sending window
9	If a packet is successfully delivered then
10	$CW_{n+1} \Leftarrow CW_n + 1/CW_n$
11	Else
12	If 3 DUPACK then
13	$CW_{n+1} \Leftarrow RTT_n k(uBE + (1-u)RE)$
14	$ssthresh_{n+1} \Leftarrow RTT_n k(uBE + (1-u)RE)$
15	If timeout then
16	$CW_{n+1} \Leftarrow 1$
17	$sstthresh_{n+1} \Leftarrow CW_n/2$

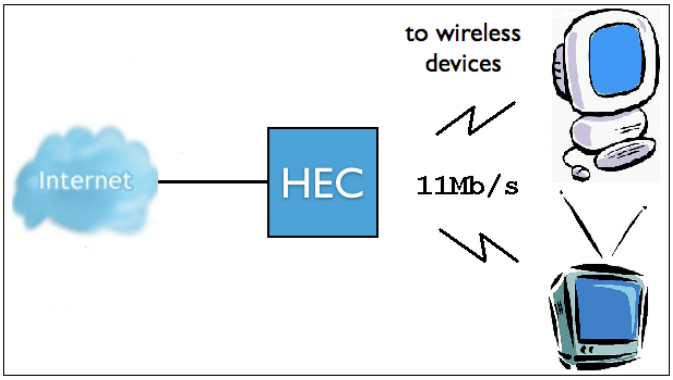

Fig. 5. Content delivery path.

In particular, k (line 3), ranges between 1/2 and 1. It equals 1/2 when the round trip time is close to its maximum ($RTT = RTT_{max}$) and, in this case, halves the sending window and the slow start threshold values (lines 13 and 14). When, instead, the round trip time is close to its minimum ($RTT = RTT_{min}$), it is set equal to 1.

Of great importance is also the role of u. Let us start to explain why, with two other additional parameters: BE and RE. We can easily define both quantities by referring to Fig. 5. The Bandwidth Estimate (BE) is defined as the capacity of the wireless link between a HEC and its clients. For example, in Fig. 5 BE equals 11Mb/s. The Rate Estimate (RE), instead, is equal to the average rate achieved by a connection, hence being influenced by congestion. For example, in Fig. 5, RE equals to 5.5Mb/s as two different flows (i.e., one that ends at the TV and one at the PC, respectively) are assumed to equally share the capacity of the wireless link.

When only one flow is active in Fig. 5, we may expect that packet loss will predominantly be due to random losses on the wireless link, as no other flows congest the path. In such case it is desirable that the throughput matches BE. Instead, when two flows are active, both flows will mainly experience congestion. The two flows should equally share the link, thus achieving RE each.

Parameter u performs the task of determining which is the chance a loss is random rather than due to congestion. BE and RE are weighted accordingly, in order to adapt the values of the sending window and the slow start threshold to the actual situation. Specifically, u, which ranges between 0 and 1, equals 0 when a congestion loss occurs, whereas equals 1 when a random loss occurs.

Only when a severe loss occurs (timeout expires), TCPW behaves exactly as the legacy TCP, setting the sending window to one and the slow start threshold to half its value (lines 15-17).

To make a long story short, TCPW has been designed to well behave when a random loss occurs. To do this, it distinguishes a random loss from a congestion loss by simply assuming that when a loss occurs and the round trip time is close to the minimum observed connection round trip time, the loss is random. This is a very desirable feature for a HEC, as it permits a higher utilization of the bandwidth resource.

B. TCPW and Stable Flows: A Steady State Analysis

While Subsection III-A describes how TCPW distinguishes a random from a congestion loss, and accordingly adapts the sending window, we now show how TCPW flows tend to a stable sending rate, under steady state conditions (i.e., steady bandwidth share and steady round trip time values).

Needless to say, stability is a highly desirable property for any multimedia stream conveyed by a wireless HEC. In fact, if TCPW flows were unstable, any connection change (e.g., a round trip time increase due to the concurrent download of a webpage) could cause abrupt oscillations of the multimedia sending rate, thus resulting in unwanted buffering events while viewing a movie, for example. As a negative result of this, a sudden throughput drop could cause a freeze of the multimedia sequence, as shown in the case of Fig. 6 and Fig. 7. These two Figures present a frame sequence of a given FLV streamed through a HEC with a stable throughput (Fig. 6, TCPW) and with an unstable throughput (Fig. 7).

Fig. 6. Stream progression with sufficient and stable throughput (TCPW).

Fig. 7. Stream progression frozen by buffering events if the stream rate does not stabilize.

In the following, we provide an analytical proof of the fact that TCPW is able to convey stable throughputs through a wireless HEC.

Indeed, the stability of TCPW may be proven in a single step: it is sufficient to show that TCPW implements a gradient algorithm, as this condition is sufficient to guarantee the convergence properties of a TCPW flow, according to the gradient theory discussed in [21], [22].

Furthermore, once this step is taken, it is also possible to compute the stable sending rate of TCPW when the average round trip time approaches its minimum and maximum values, respectively, with the aim of quantifying to what extent the former is larger than the latter.

Let us start now with the first step of our proof, and simply rewrite the equation of line 13 in Algorithm 1 of Table I, with an explicit dependence on time, it goes as follows:

$$CW(t) = RTT(t)\, k\, (u\, BE + (1 - u)\, RE). \quad (1)$$

We now transform (1) to obtain a fluid flow interpretation of TCPW: this will result in a sending rate *x(t)* evolving in time. To understand how this transformation is carried out consider the following arguments.

Specifically, assuming $\lambda(t)$ be the probability a packet is lost, when an ACK is received, TCPW increases its sending window by $1/CW(t)$. As this event happens with a frequency of $x(t)(1-\lambda(t))$ (i.e., the rate of returning ACKs is equal to the sending rate times the probability a packet is received), we have that TCPW increases its sending window at a rate of $(1/CW(t))\,x(t)(1-\lambda(t))$.

Instead, suppose that at the time instant *t*, occurring after a time interval of *RTT*, three duplicate ACKs are received, then TCPW assumes a loss has occurred and, consequently, computes a reduction of its sending window which is equal in size to the difference of the values of the two sending windows at time $(t - RTT)$ and *t*, respectively. This reduction amounts hence to $(CW(t - RTT(t)) - CW(t))$. If then the number of loss events at time *t* is equal to $x(t)\lambda(t)$, we have that TCPW decreases its sending window by the following quantity: $x(t)\lambda(t)\,(CW(t - RTT(t)) - CW(t))$.

After these simple considerations, the differential equation describing the evolution of the sending rate $x(t)$ in time (e.g., $\dot{x}(t)$) can be written as the sum of two contributions: the amount by which the sending rate increases, $(1/CW(t))\,x(t)(1-\lambda(t))$, minus the amount by which it decreases, $(x(t)\lambda(t)\,(CW(t - RTT(t)) - CW(t)))$.

As a result we then have that the fluid flow formulation of TCPW is:

$$\dot{x}(t) = \frac{1}{CW(t)}\,x(t)(1-\lambda(t)) + \\ - x(t)\lambda(t)(CW(t - RTT(t)) - CW(t)). \quad (2)$$

As we are interested in how the sending rate evolves under steady state conditions, this entails dealing with average values.

Hence we set: $RTT(t) = \overline{RTT}$ (which also gives $u = \bar{u}$ and $k = \bar{k}$) and $RE = \bar{x}$, while *BE* can be assumed constant at all times as it equals the bandwidth capacity of the path.

Moreover, as the amount of packets sent at time *t* equals the sending window at time *t*, the sending rate $x(t)$ can be approximated by the sending window divided by the average round trip time, yielding $x(t) \approx CW(t)/\overline{RTT}$.

All these things considered, we can then rewrite (2) as a function of the round trip time at the steady state and of the packet loss at time *t*, $\lambda(t)$:

$$\dot{x}(t) \approx \frac{1-\lambda(t)}{\overline{RTT}^2} - x(t)\lambda(t)(x(t) - \bar{k}(\bar{u}\,BE - (1-\bar{u})\bar{x})). \quad (3)$$

The Gradient theory tells us now that (3) represents a gradient algorithm if it can be written in the following form [21], [22]:

$$\dot{x}(t) = s(x)(\dot{U}(x) - \lambda(t)), \quad (4)$$

where $U(x)$ is the utility function.

It is now important to take a break to explain why we exploit the utility function concept in this framework. Simply, the utility is exploited in economics and in game theory, where the role of a utility function is that of mapping a given variable to a certain measure of benefit (i.e., utility); the same occurs in our TCP domain, where the given variable is the sending rate (i.e., $x(t)$).

Under this metaphor, (4) expresses the evolution in time of $x(t)$ as the difference of the derivative of the utility function and the packet loss quantity $(\dot{U}(x) - \lambda(t))$, times a correction factor $s(x)$.

In very simple words, if, with the passage of time, the number of losses at a given moment *t* is too large, TCPW decreases its sending rate, as this maximizes the general utility function of the system, i.e., the number of successfully sent packets.

Always following the utility theory, it is possible to show that, when $\lambda(t) << 1$, we can obtain $s(x)$ and $U(x)$ in our TCPW domain as respectively given by the following formulas:

$$s(x) = x^2(t) - (\overline{\psi} - \overline{\eta x})x(t), \quad (5)$$

and,

$$\dot{U}(x) = 1/\left((1+\overline{\eta})R\overline{T}T^2x^2 - \overline{\psi}\,R\overline{T}T^2x^2\right), \quad (6)$$

with $\overline{\psi} = (\bar{k}\,\bar{u}\,BE)$ and $\overline{\eta} = \bar{k}(1-\bar{u})$.

If we now replace (5) and (6) into (4) we can simply obtain a gradient version of (3).

Summarizing, as we have been able to finally complete this transformation we have consequently proven the stability of TCPW.

This was only the first part of our objective, but it is not yet enough. We are interested now in evaluating the stable values of the sending rates for both the situation when *RTT* approaches its maximum and minimum values, respectively. Indeed, the difference between these two values will express the real advantage of distinguishing random errors from congestion losses in our HEC framework.

This can be easily obtained by setting $\dot{x}(t) = 0$. The solution of this differential equation will return the $x(t) = \bar{x}$ value we are looking for, in both the case of $RTT_{\max}$ and RTT_{min}.

For the sake of brevity, we do not provide the reader with all the intermediate steps leading to the final values of these two parameters which finally result as follows:

$$\bar{x}_{RTT_{\min}} \approx \frac{BE}{2} + \frac{1}{2}\sqrt{BE^2 + \frac{4}{\bar{\lambda}RTT_{\min}^2}}, \quad (7)$$

$$\bar{x}_{RTT_{\max}} \approx \frac{1}{3RTT_{\max}}\sqrt{\frac{6}{\bar{\lambda}}}. \quad (8)$$

It is easy to understand by analyzing (7) that TCPW ideally yields a throughput value equal to the pipe size (*BE*) with smaller *RTT* values (i.e., random losses), whereas (8) tells us that the throughput decreases following a 1/*RTT* rule when congestion losses occur.

Also of interest is the ratio between (7) and (8) which returns:

$$\frac{\bar{x}_{RTT_{\min}}}{\bar{x}_{RTT_{\max}}} \approx 3RTT_{\max}\, BE\sqrt{\bar{\lambda}/6}. \quad (9)$$

This ratio tells us that TCPW outperforms the legacy TCP when either the bandwidth, or the round trip time, or both increase. We will see this observation confirmed by the results of Section IV.

For all these reasons we can conclude that TCP is a valid candidate for a wireless HEC implementation.

IV. Results

We performed three sets of experiments, using a simulator we developed in C#. We report the average values drawn from running TCPW and TCP SACK 10 times per each set of experiments.

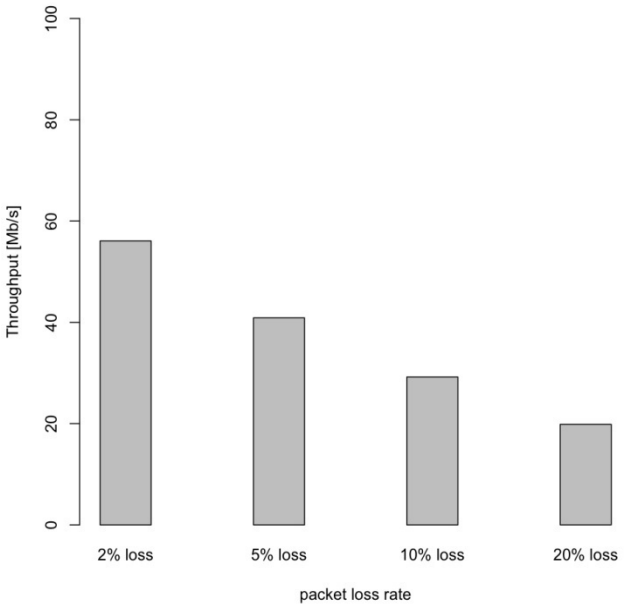

Fig. 8. HEC throughput with TCP SACK.

The first experiment refers to a wireless HEC scenario, where the two way propagation delay between a wireless HEC and a device is very low (one millisecond), and the wireless link bandwidth is 100Mb/s. In Fig. 8 and Fig. 9 we report the throughput achieved using TCP SACK and TCPW, respectively, for a 100 seconds long connection, and under a random packet loss ranging from 2% to 20%.

In particular, Fig. 8 and Fig. 9 show that the higher the packet loss rate is, the greater is the advantage of using TCPW in terms of achieved throughput. In fact, with a 20% packet loss rate the throughput achieved by TCPW is two times larger than that achieved with the legacy TCP.

The second and third class of experiments were developed to show how the performance gap between TCPW and the legacy TCP increases depending on the home connection round trip time value, and on the value of bandwidth at home as well.

The second scenario accounts for a situation where the two way propagation delay is 70 ms, packet loss probability is 2% and the maximum available bandwidth is 10Mb/s. The leftmost part of Fig. 10 compares the values of the throughput achieved by the legacy TCP and TCPW, respectively, under the conditions mentioned before. As expected, we notice that an increase in round trip time favors TCPW which achieves a higher throughput.

Finally, the third scenario accounts for a situation where the two way propagation delay and the packet loss probability remain the same as in the previous case, whereas the bandwidth increases up to 100Mb/s. Again, as expected, TCPW outperforms the legacy TCP (rightmost part of Fig. 10) confirming the findings of Section III.

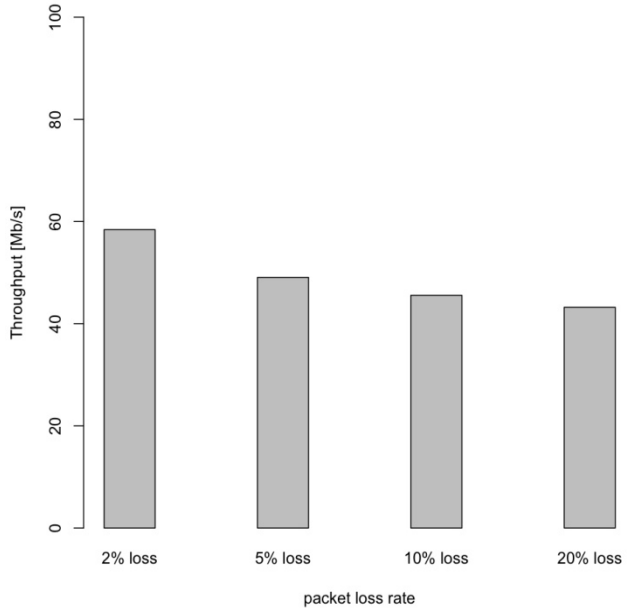

Fig. 9. HEC throughput with TCPW.

V. Conclusion

This paper has addressed the problem of devising a wireless HEC protocol architecture with the aim of identifying the best TCP variant able to maximize the throughput of the multimedia flows distributed throughout the home.

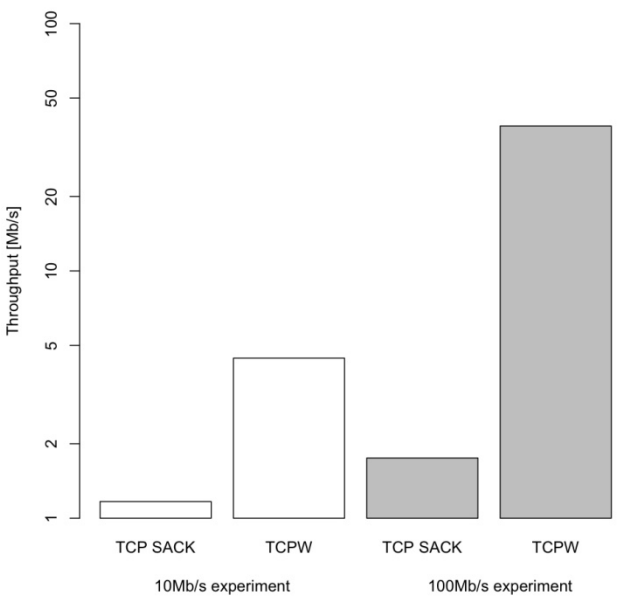

Fig. 10. Left: TCP SACK and TCPW throughput on a 10Mb/s link, with a 2% packet loss rate. Right: TCP SACK and TCPW throughput on a 100Mb/s link, with a 2% packet loss rate.

We have proposed a HEC architecture that exploits TCPW as a means to maximize and stabilize those flows on the home side.

Our solution complies with the legacy TCP used on the Internet side as all the incoming traffic can be broken at the HEC using a proxy-based approach. Analytical and simulative results have confirmed the validity of our approach.

A final recommendation is to mount our HEC protocol stack architecture on top of a WiFi 802.11e layer, specifically devised to support multimedia flows. In particular, 802.11e limits the number of retransmissions when random errors occur, thus favoring the intervention of TCPW.

REFERENCES

[1] K. Keeker, R. Pagulayan, J. Sykes and N. Lazzaro, "The untapped world of video games," *Proc. of the ACM Conference on Human Factors in Computer Systems (CHI'04),* Vienna, Austria, Apr. 2004.

[2] T. Hennig-Thurau, V. Henning, H. Sattler, F. Eggers and M.B. Houston, "The last picture show? Timing and order of movie distribution channels," *Journal of Marketing*, vol. 71, no. 4, pp. 63-83, Oct. 2007.

[3] B. Fogg, "Three possible futures for persuasive technology," *Proc. of the Fourth International Conference on Persuasive Technology (Persuasive'09),* Claremont, CA, US, Apr. 2009.

[4] A. Sentinelli, G. Marfia, S. Tewari, M. Gerla and L. Kleinrock, "Will IPTV ride the peer-to-peer stream?" *IEEE Communications Magazine,* vol. 45, no. 6, pp. 86-92, Jun. 2007.

[5] M. Cha, P. Rodriguez, J. Crowcroft, S. Moon and X. Amatriain, "Watching television over an IP network," *Proc. of the Eighth ACM SIGCOMM Conference on Internet Measurement (IMC'08).* Vouliagmeni, Greece, Oct. 2008.

[6] R.S. Cruz, M. S. Nunes and J. Goncalves, "A Personalized HTTP adaptive streaming WebTV," *Proc. of the First ICST Conference on User Centric Media (UCMedia'09),* Venice, Italy, December 2009.

[7] C.E. Palazzi, S. Ferretti, M. Roccetti, G. Pau and Gerla, "What's in that magic box? The home entertainment center's special protocol potion, revealed," *IEEE Trans. Consumer Electron.*, IEEE Consumer Electronics Society, vol. 52, no. 4, pp. 1280-1288, Nov. 2006.

[8] L. Backstrom, D. Huttenlocher, J. Kleinberg, G. Maier, A. Feldmann, V. Paxson and M. Allman, "On dominant characteristics of residential broadband internet traffic," *Proc. of the 9th ACM SIGCOMM Conference on Internet Measurement (IMC'09),* Chicago, Illinois, Nov. 2009.

[9] F. Qian, A. Gerber, Z. M. Mao, S. Sen, O. Spatscheck and W. Willinger, "TCP revisited: a fresh look at TCP in the wild," *Proc. of the 9th ACM SIGCOMM Conference on Internet Measurement Conference (IMC'09),* Chicago, Illinois, Nov. 2009.

[10] J. Van Der Merwe, S. Sen and C. Kalmanek, "Streaming video traffic: characterization and network impact," *Proc. of the 2002 International Web Content Caching and Distribution Workshop*, Boulder, CO, US, Aug., 2002.

[11] C. Huang, J. Li and K Ross, "Peer-assisted VoD: Making internet video distribution cheap," *Proc. of the Sixth International Workshop on Peer-to-Peer Systems (IPTPS'07).* Bellevue, WA, US, Feb. 2007.

[12] K. Fall and S. Floyd, "Simulation-based comparisons of Tahoe, Reno and SACK TCP," *SIGCOMM Computer Communication Review*, vol. 26, no. 3, pp. 5-21, Jul. 1996.

[13] L. Xu, K. Harfoush and I. Rhee, "Binary increase congestion control (BIC) for fast long-distance networks," *Proc. of the Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'04),* Hong Kong, P.R.C., Mar. 2004.

[14] S. Ha, I. Rhee and L. Xu, "CUBIC: a new TCP-friendly high-speed TCP variant," *SIGOPS Operating Systems Review,* vol. 42, no. 5, pp. 64-74, Jul. 2008.

[15] H. Shimonishi, M. Y. Sanadidi and M. Gerla, "Improving efficiency-friendliness tradeoffs of TCP in wired-wireless combined networks," *Proc. of the IEEE International Conference on Communications (ICC'05),* Seoul, Korea, May 2005.

[16] G. Yang, L. J. Chen, T. Sun, M. Gerla and M. Y. Sanadidi, "Real-time streaming over wireless links: a comparative study," *Proc. of the Tenth IEEE Symposium on Computers and Communications (ISCC'05),* Cartagena, Spain, Jun. 2005.

[17] G. Marfia, C. E. Palazzi, G. Pau, M. Gerla, M. Y. Sanadidi and M. Roccetti, "TCP Libra: balancing ows over heterogeneous propagation scenarios," *Proc. of the 6th International IFIP-TC6 Networking Conference (NETWORKING 2007),* Atlanta, Georgia, US, May 2007.

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

[19] R. Wang, M. Valla, M. Y. Sanadidi, B. K. Ng, and M. Gerla, "Efficiency/friendliness tradeoffs in TCP Westwood," *Proc. of the Seventh IEEE Symposium on Computers and Communications (ISCC'02),* Taormina, Italy, Jul. 2002.

[20] R. Wang, M. Valla, M. Y. Sanadidi, and M. Gerla, "Adaptive bandwidth share estimation in TCP Westwood," *Proc. of the IEEE Global Communications Conference (Globecom'02),* Taipei, Taiwan, R.O.C., Nov. 2002.

[21] R. Srikant, "The mathematics of internet congestion control," Springer Verlag, 2004.

[22] F. Kelly, "Charging and rate control for elastic traffic," *European Trans. Telecomm.*, vol. 8, no. 1, pp. 33-37, Sept. 1997.

BIOGRAPHIES

Gustavo Marfia received a degree in Telecommunications Engineering from the University of Pisa in 2003 and a Ph.D. in Computer Science from UCLA, in 2009, respectively. He is currently a Researcher at the Department of Computer Science, University of Bologna. His research interests include ad hoc networks, consumer multimedia and multimedia streaming.

Marco Roccetti received the Italian degree in Electronic Engineering from the University of Bologna, Italy in 1989. He has been a Professor with the Computer Science Department, University of Bologna since 2000. His research interests include digital audio and video, consumer multimedia and computer entertainment.