Wireless Home Entertainment Center: Reducing Last Hop Delays for Real-time Applications

Claudio E. Palazzi[1, 2], Giovanni Pau[1], Marco Roccetti[2], Stefano Ferretti[2], Mario Gerla[1]

[1] Computer Science Department
University of California, Los Angeles, CA 90095, USA
Tel: +1 - 310 - 825 4367
e-mail: {cpalazzi | gpau | gerla}@cs.ucla.edu

[2] Dipartimento di Scienze dell'Informazione
Università di Bologna, 40126, Bologna, Italia
Tel: +39 - 051 - 209 4503
e-mail: {roccetti | sferrett}@cs.unibo.it

ABSTRACT

Future digital entertainment services available to home users will share several characteristics: i) they will be deployed and delivered through the Internet, ii) a single media center will be exploited to orchestrate all parallel services, and iii) wireless technologies integrated within the home entertainment system will be massively utilized for the transmission of various data streams to networked devices. In this scenario, new effective strategies are needed to regulate the concurrent access to the wireless network when parallel applications generate different but simultaneous UDP/TCP-based flows. In this work, we present a novel technique aimed at guaranteeing a fast and smooth data delivery for real-time streams while maintaining a high throughput for TCP-based applications. Our approach is based on the utilization of a smart Access Point able to exploit available information about the ongoing traffic and existing features of the regular TCP. We compare the performance of our solution with an alternative one that makes use of an optimal setting of the 802-11 MAC layer parameters. Simulation results confirm that our smart Access Point represents an optimal candidate to be exploited in complex wireless scenarios for in-home entertainment.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Performances Attributes.

General Terms

Algorithms, Measurement, Performance, Design, Experimentation, Verification.

Keywords

Media Center, Wireless Networks, Computer-Centered Home Entertainment, Transmission Protocols, Home Entertainment Center.

1. INTRODUCTION

Since its first appearance in 1999, *TiVo* has revolutionized the way millions of customers interact with their TVs. Its integration between TV sets and computers, in fact, enabled users to pause live-TV programs, even for several hours, and watch them at their own convenience, thus substituting the old VCR with a more functional Digital Video Recorder (DVR) [1]. This idea has been further extended by Microsoft through its *Media Center*, which represents a hub for all in-home entertainment experiences [2]. As a matter of fact, more than any other device, computers have evolved gaining the capability to handle heterogeneous media. Both TiVo and Media Center, as well as the future evolution of this technology, can be grouped into the class of *Home Entertainment Center* (HEC).

In this context, Internet is going to play a fundamental role as the wider and wider diffusion of broadband connectivity is helping it in becoming the major vehicle for providing entertainment services. Therefore, a HEC is also designed to be connected to the Internet and perform as a gateway between client devices located in the house (or in a student dorm, for example) and the outside world. A HEC provides all-in-one functionalities and combines several services such as IPTV, Web radio, game console, picture viewer, electronic program guide, DVR, CD/DVD/video player, music jukebox, web browser, email handler, instant messenger. Contents related to these services can be locally available or distributed over the Internet to be dynamically retrieved based on the user's needs. Several streams are thus produced by these active services, all distributed by the HEC throughout the home.

To make the HEC able to communicate with client devices, several transmission strategies can be devised. However, wireless technologies present the flexibility and mobility features to overcome physical barriers and permit an (almost) instantaneous connectivity from every spot in the house. For this reason, we assume that HECs are endowed with an Access Point (AP) in order to guarantee wireless connectivity to the various user's devices (e.g., screens, speakers, joypads).

Indeed, a broad range of entertainment services can be singularly conveyed over IEEE 802.11 (*Wi-Fi*) technologies thanks to their transmission rates. Nominally, 11Mbit/s for the IEEE 802.11b, 54Mbit/s for the IEEE 802.11g, or even 100 Mbit/s for the IEEE 802.11n. Yet, mobility is achieved at the cost of a lower performance than that attainable through the use of a wired network. With respect to the latter, in fact, wireless networks provide lower bit rates which also depend on the distance and obstacles between the client device and its AP. Very limited effort has yet been devoted to study the impact of several heterogeneous streams that simultaneously shares the same (initial or final)

ACE 06, June 14-16, 2006, Hollywood, California, USA.

wireless hop while the question whether Wi-Fi can actually support the intense traffic generated by various and heterogeneous entertainment applications in an in-home environment needs an immediate and definite answer.

Furthermore, networks and related protocols are often developed assuming that the traffic will be mostly TCP-based and that the main goal will be that of guaranteeing high throughputs. These assumptions need a radical reconsideration when services for entertainment come into the picture. Think, for instance, of fast-paced gaming applications, according to which small amounts of data are transmitted at high constant rates via the UDP transport protocol. This kind of application is extremely delay sensitive thus having in low delay latencies for packet delivery its focal requirement. The same holds for IPTV, where TV programs and/or Video on Demand services are offered to clients using the medium of the Internet. In such scenarios, latencies and jitter must be reduced as much as possible in order to ensure an acceptable quality of service.

To facilitate the deployment of networked services through the HEC, we presented in [3] a study of the problem focusing on cross-layer interactions between transport and MAC layers in presence of heterogeneous and concurrent entertainment transmission flows. We showed in particular how even a single persistent TCP connection is able to deteriorate the performance of real-time entertainment applications. We then proposed a MAC layer tune-up for APs in order to find a compromise between the requirements of downloading applications and those of real-time applications. Although it represents an improvement to the current state of the art, this solution is still not optimal.

Aiming at finding an optimal solution, we present now an alternative approach focused on the transport layer and exploiting existing features of regular TCP. In substance, a HEC is centrally located in the house and communicates with the user's devices through the AP. The AP is hence in the position of having a comprehensive sight on all the in-home transmissions and to control them. In particular, the AP can be enhanced to snoop all transiting streams and modify the *advertised window* of TCP packets in order to limit the growth of on-going TCP flows. If appropriately applied, this technique produces a good TCP rate, yet without exceeding the capacity of the channel: losses and consequent halving of the sending rate are, in fact, avoided. At the same time, as the buffer at the AP remains always (almost) empty, the delivery time for each transiting packet is not affected by queuing delay and users perceive a smooth progression of real-time entertainment applications.

We show how the AP associated with the HEC, can dynamically and effectively intervene on the advertised window of transiting TCP packets in order to ensure optimal performances for both downloading and real-time entertainment applications. We compare this solution with the one proposed in [3] by investigating both their efficiency and factual deployability. In our study, we analyzed a realistic scenario with 4 different applications simultaneously run: video stream, online gaming, video chat, and download of multimedia files.

The paper is organized as follows. Section 2 discusses issues concerned with the transmission of multimedia streams over in-home wireless scenarios. Section 3 presents the aforementioned solutions to face these issues. In Section 4, we describe the simulation scenario exploited to assess the efficacy of our devised approaches, while in Section 5 we analyze the obtained results. Finally, Section 6 concludes our work.

2. ON THE IMPACT OF THE WIRELESS MAC / TCP INTERFERENCE

As recently demonstrated by measurements on a real OC48 link, the available capacity over the Internet is generally larger than the aggregate bandwidth utilized by transiting flows [4]. Moreover, more and more providers are offering today guaranteed high speed connectivity to home customers [5, 6, 7]. In essence, tools are available to verify that the customer's connection is factually supporting as much traffic as the bandwidth advertised by the provider.

In this scenario, it is widely accepted that the bottleneck of the connection is generally located at the edge of the path connecting a sender and a receiver; specifically, the last link connecting an edifice with the Internet or the in-edifice wireless link. Since the rapid increase of the bandwidth delivered to homes and the broadband available in locations that intend to gain profit from offering high performance online entertainment to customers, we can assume to have the bottleneck located in correspondence of the wireless link. Indeed, when several contemporary applications share the same wireless link, it might be the case when the AP receives packets at higher rates than its forwarding one. This can happen for several reasons such as, for instance, the fact that the wireless medium allows the transmission of only one packet at a time and is not full-duplex as wired links.

Moreover, interference, errors, fading, and mobility may cause packet losses which are handled by the MAC protocol through local retransmissions. These local retransmissions hide error losses to the TCP and are useful to increment the reliability of the connection. Without them, the TCP would misinterpret error losses as congestion evidences and reduce its sending rate decreasing its performance. On the other hand, retransmissions follow the well known back off mechanism by which an increasing amount of time is utilized to determine whether a packet has been lost and hence retransmit it. The 802.11 MAC protocol performs up to seven retransmissions of short packets (i.e., RTS/CTS, acks) and four retransmissions of long packets (i.e., data packets) [8]. This means that subsequent packets have to wait in queue until the preceding ones or their retransmissions finally reach the receiver and the corresponding acks get to be successfully sent back.

Finally, the same wireless connection might be shared by several devices and applications that increase the congestion level causing queuing. As it is well known, TCP connections have an aggressive behavior and continuously probe the channel for more bandwidth until buffers are fully utilized and overflowed. In presence of persistent TCP connections (i.e., when downloading files) it is hence very likely to happen that buffers were steadily fully utilized, thus periodically slowing down the delivery time of each packet, and deteriorating the performance of time-sensitive applications such as online games, for example.

At the same time large buffers help TCP-based flows in keeping a high sending rate. This happens for several reasons but the most important ones are: i) the link successive to the buffer remains fully utilized for longer periods of time since there are (almost)

always packets in queue that are ready to be sent as soon as possible, and ii) traffic bursts can be more easily accommodated thus reducing packet losses and maintaining higher sending rates for longer periods of time. In essence, a tradeoff relationship exists among the per-packet delay and the total throughput achieved. The solution for this tradeoff depends on the buffer size and on its utilization.

2.1 Problem Statement

As previously mentioned, the impact of the interference between wireless MAC and transport protocols can produce delays which could hit also several tens of milliseconds. This represents a huge waste of time when trying to concurrently deliver real-time information for entertainment services. Think, for instance, of online games which have very stringent requirements on delay latencies, i.e., typically 150ms is considered as the maximum endurable transmission delay to guarantee interactivity [9]. For this reason, effective control schemes must be devised to limit queuing delays when data streams for real-time entertainment applications are simultaneously active.

3. PROPOSED SOLUTIONS

To solve the aforementioned problem we analyze two different possible solutions. The first one involves modifications of the 802.11 MAC layer and, hence, is specifically intended for the wireless media [3]. The second one, instead, exploits existing features of regular TCP and could be extended also to the case where the connection is completely wired. For the proposed solutions we investigate both their efficiency and factual deployability to expose pros and cons.

3.1 IEEE 802.11 Parameters Setting

The first proposal regards the utilization of more appropriate setting for parameters of the IEEE 802.11 MAC protocol. Parameters such as the maximum number of retransmissions and the buffer size were, in fact, determined in a period when the TCP-based traffic was largely predominant in the Internet. The main concerns for designers were hence reliability and high throughputs.

As already mentioned, nowadays, UDP-based real-time entertainment applications are becoming more and more popular and demand for low delays in packet delivery. This kind of applications is resilient to some packet loss while is extremely delay sensitive in packet delivery. For this reason, it is preferable to drop a packet than to waste time in retransmissions.

This obviously partially contradicts the initial assumption that reliability is the most important issue over wireless links. Therefore, 802.11 parameters should be modified to make it more sensitive towards real-time application needs. In particular, the number of local retransmissions could be diminished in order to find an efficient compromise between reliability and low delays in packet delivery.

Furthermore, having a large buffer size at the AP helps TCP connection to maintain large sending rates for longer periods and diminishes the impact of burst traffic. On the other hand, to a larger buffer corresponds a longer queuing time experienced when the buffer is full, thus jeopardizing the performance achieved by time-sensitive applications. By adjusting the buffer size to an appropriate value we can again try to find an optimal compromise between the needs of both traditional TCP-based and real-time entertainment applications.

3.2 Limited Advertised Window

Here, we are aiming at finding the best solution to the tradeoff relationship existing between TCP throughput and real-time application delays. The two types of traffic should be able to coexist without affecting each other and the employed solution should be easily and factually deployable.

Starting from the last point, i.e., deployability, it is evident how a technique that would exploit existing features of the already utilized protocols could be easily implemented in a real scenario. A possible solution could hence be that of utilizing the advertised window to limit the bandwidth utilized by TCP flows.

Indeed, the actual sending rate of a TCP flow depends on its current *sending window*; this value is determined as the minimum between the *congestion window* (continuously recomputed by the sender) and the advertised window (provided by the receiver via returning *ACK* packets) [10]. It is hence evident how the advertised window perfectly embodies a natural upper bound for the sending rate of TCP flows.

Limiting the maximum sending rate of a TCP connection may greatly improve the performance of the HEC. An optimal tradeoff between throughput and low delays, in fact, could be achieved by maintaining the sending rate of the TCP flows high enough to efficiently utilize the available bandwidth but, at the same time, limited in its growth so as to not utilize buffers. In this way, in fact, the throughput is maximized by the absence of packet losses which would halve the congestion window, while the delay is minimized by the absence of queues.

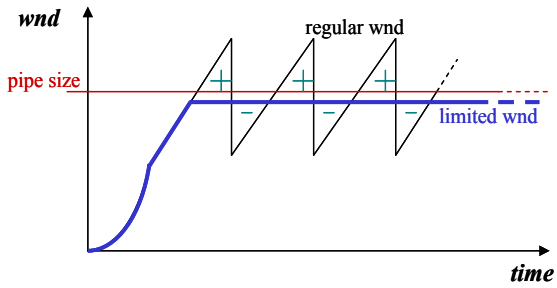

Figure 1. Comparison between regular and limited sending windows (wnd).

To better understand how limiting the sending window could guarantee the same or even a higher throughput with respect to utilizing regular TCP, we show in Figure 1 a general saw tooth shaped sending window of a regular TCP and overlap it with one limited by the advertised window. As it is evident, the latter is more stable since it does not use the buffer at the bottleneck link and consequently experiences no losses. The minus signs in the chart represent situations in which the regular sending window provides TCP with a sending rate that is inferior to the one guaranteed by the limited sending window. The plus signs represent the inverse situation (generally accompanied by having packets queuing on the buffer preceding the bottleneck link). If the upper bound for the sending window is appropriately chosen,

the balance between the plus and minus signs will guarantee to the limited sending window an equal or even superior final throughput with respect to the regular sending window. At the same time, queuing delays will be avoided.

To achieve this desirable result we need first to address two important issues: how to determine an appropriate upper bound and how to apply it in practice to the sending window.

Regarding the first point, the most appropriate formula can be derived from the two main goals we want to achieve: i) full utilization of the available bandwidth and ii) no queuing delays. Real-time traffic generally exploits UDP and this transport protocol has no congestion control mechanism. Some smart UDP-based application, however, implements congestion control at the application layer [11]. In any case, to avoid queuing delays, the aggregate bandwidth utilized by TCP flows cannot exceed the total capacity of the bottleneck link diminished by the portion of the channel occupied by the concurrent real-time traffic.

In essence, the maximum sending rate for each TCP flow at time *t*, namely *maxTCPrate(t)*, is represented by:

$$maxTCPrate\ (t) = \frac{(C - UDPtraffic\ (t))}{\#TCPflows\ (t)} \qquad (1)$$

where *UDPtraffic(t)* corresponds to the amount of bandwidth occupied by UDP-based traffic at time *t*, *#TCPflows(t)* is the concurrent number of TCP flows, and *C* represents the capacity of the bottleneck link and must be accurately determined in order to optimize performances.

The second issue that we need to address is how to practically employ this formula in order to have it working in a real scenario. This means i) identifying the location for its implementation, and ii) proposing a method to compute the value of the various variables in (1).

Regarding the first issue, the advertised window is generally imposed by the receiver; however, this could not represent the most suitable place to set it. Determining the most appropriate value for the advertised window requires a comprehensive knowledge about all the flows that are transiting through the bottleneck. Since all flows have to pass through the AP, this represents the most appropriate node on which implementing our scheme. Indeed, the AP is integrated with the HEC and the mechanism can take advantage of this to retrieve all the necessary information. This approach is also in accordance with other proposals available in literature such as, for example, [12]. However, whereas [12] requires modifications at both the AP and the receiver, our scheme exploits a "smarter" AP.

Focusing on the second issue, in any commercial operating system it is possible to know which kind of connection is in use and which its nominal speed is just by looking at the status of the network interface. Knowing this, in Section 5.3 we empirically find the optimal value for the considered in-home scenario. Through snooping the channel or exploiting information known at the HEC we can also infer the number of active TCP connections and the aggregate amount of current UDP traffic. The AP can hence easily compute the best *maxTCPrate(t)* utilizing (1) and accordingly modify the advertised window included in the transiting acks. From here on we refer to this scheme as *Smart Access Point with Limited Advertised Window* (SAP-LAW).

4. IN-HOME ENTERTAINMENT: A SIMULATION ASSESSMENT

The aforementioned scenario has been analyzed in depth through the well known NS-2 network simulator (version ns-2.28) [13]. In particular, the simulated topology is depicted in Figure 2 where the in-house entertainment environment is represented by four mobile nodes named N1, N2, N3 and N4, and the HEC that incorporates also the AP.

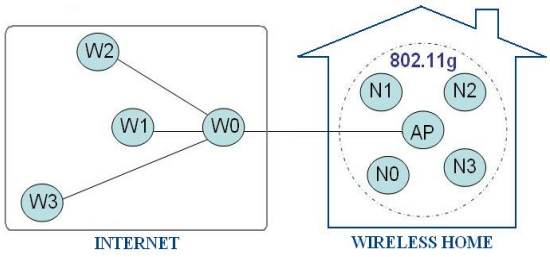

Figure 2. Simulated topology.

The MAC layer parameters have been set accordingly to the IEEE 802.11g standard. The simulation outcomes showed us that we were able to reach a maximum achievable bandwidth of circa 20Mbps. This represents a reasonable value over the declared 54Mbps even in the real world [14].

Table 1. Simulation configuration of the wired links

Node 1	Node 2	Physical Latency	Link Capacity	Buffer Size
W1	W0	10ms	100Mbps	140pkts
W2	W0	20ms	100Mbps	140pkts
W3	W0	30ms	100Mbps	140pkts
W0	AP	10ms	100Mbps	140pkts

Table 2. Simulated application flows

From	To	Flow Type	Transp. Prot.	Start	End
AP	N0	video stream	UDP	0s	180s
W1	N1	online game	UDP	45s	180s
N1	W1	online game	UDP	46s	180s
W2	N2	video chat	UDP	90s	180s
N2	W2	video chat	UDP	91s	180s
W3	N3	FTP	TCP NewReno	135s	180s

Regarding the wireless medium we have adopted the Shadowing Model which is a realistic and widely utilized signal fading model available in NS-2. We followed the directions provided by the official NS-2 manual to represent a home environment partitioned into several rooms. Specifically, in our simulations, the path loss exponent of the Shadowing Model was always set equal to 4, while different shadowing deviation values have been tested to simulate different partition degrees inside the house. The attenuation of the transmitted signal grows with the increase of these parameters; we hence expect to face higher percentages of packet losses over the wireless medium when setting the shadowing deviation to 9.

Focusing on the wired links, their one-way delays and capacities have been configured as listed in Table 1, while their buffer sizes have been set equal to 140 packets. This value corresponds to the

pipe size on the connection starting from W3 and reaching a wireless node.

Table 3. Changing parameters in the simulated configurations

Parameter	Values	Comment
MAC data transmission	1, 2, 3, **4**	standard value is 4
Shadowing deviation	**7**, 9	medium, high
user-AP distance	5m, **10m**	same room, different room
MAC buffer size	50pkt, **100pkt**	common values in commerce

In order to represent a general scenario we have run different kinds of applications which are listed in Table 2. The characteristics of the various simulated flows make them highly realistic. In particular, the video-stream and video-chat flows have been generated by feeding the NS-2 with real trace files of high quality MPEG4 *Star Wars IV* and VBR H.263 Lecture Room-Cam, respectively [15].

Moreover, we have assumed that the player in the house is engaged in one of the very popular first person shooter games with other ~25 players, geographically apart from each other and connected through the Internet. We have hence set NS-2 to generate the corresponding traffic considering the approximations suggested in [16]. Specifically, game events have been generated at client side every 60ms; while the server was transmitting game state updates every 50ms toward the client. The packet size has been set to 42Bytes and 200Bytes for client and server generated game packets, respectively.

Due the space limitation of this paper, we present results only for the throughput achieved by the FTP application and the jitter experienced by the online gaming application. However, no significant information is lost since the per-packet delay and jitter for all the simulated real-time applications showed similar behaviors.

Simulation experiments have been replicated to examine the effects generated by differently setting some of the parameters involved in the scenario. Table 3 lists all the variable parameters in the simulations; each combination of their possible values has been simulated. However, where not differently stated, simulations were run utilizing some realistic default values for the simulative parameters. These values are written in bold in Table 3.

5. EXPERIMENTAL RESULTS

We present here the most relevant results from the extensive set of simulations we have run. In particular, we first demonstrate how concurrent TCP-based traffic can affect the performance of real-time applications when our solutions are not employed. We then compare the outcome with those of our proposed solutions.

5.1 FTP Impact on Real-Time Entertainment Applications

In this Section we discuss results obtained when resorting to standard setting parameters for both MAC and TCP without resorting to our solutions. The various applications start in sequence at precise times. In this way, it is possible to evaluate the impact of every new flow over the preexisting ones. In particular, we expect to witness increasing delays and hence higher jitter in the arrival time of packets as we augment the traffic level.

As it is evident from Figure 3 and Figure 4, the bandwidth requirement of the first starting applications (i.e., the real time ones) in our scenario is well below the effectively available capacity of the IEEE802.11g wireless medium. We have to wait until the FTP flow takes action, quickly saturating the channel and the buffers along the path with its packets, before being able to clearly detect a significant impact on the various real-time flows.

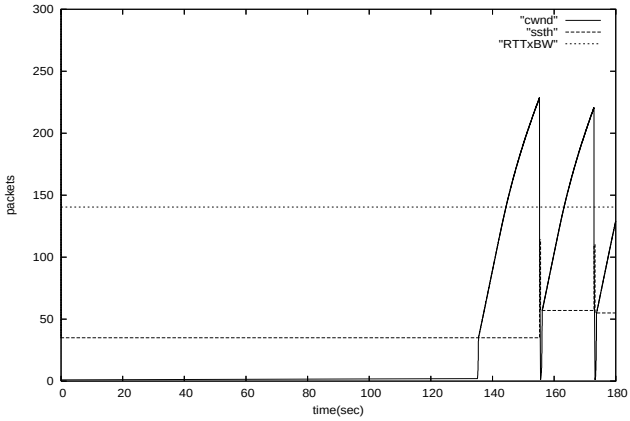

Figure 3. Measured TCP congestion window when the regular TCP NewReno and IEEE 802.11g are employed.

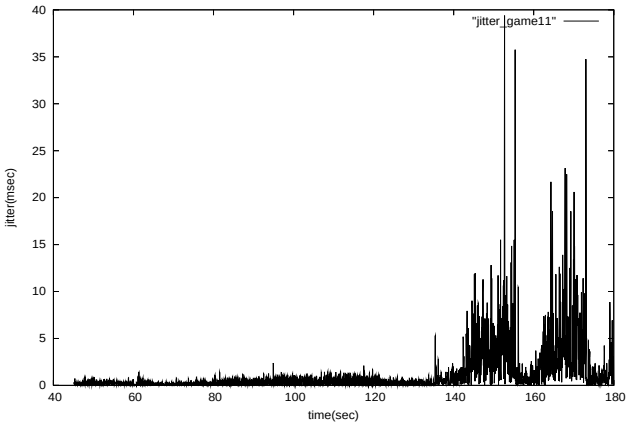

Figure 4. Measured online game jitter with regular IEEE 802.11g employed; from 135s, a regular TCP NewReno flow is competing for the channel.

More in detail, Figure 3 illustrates the congestion window and the slow start threshold for the TCP flow, plus the bandwidth-RTT product for the channel (i.e. the horizontal line in correspondence of 146 packets). As it is evident from the chart, the congestion window steadily surpasses the bandwidth-RTT product, which represents the pipe capacity of the channel, thus generating queuing up of packets at the bottleneck. Consequently, real-time gaming packets experience a steady increase in the delivery time, which is due to higher queuing delays. Corresponding to peaks in

the saw tooth shape of the congestion window (see Figure 3), we have striking amplification of the jitter experienced by the online game application (see Figure 4). Real-time application providers typically deploy mechanisms to ensure low per-packet latencies over the Internet. Clearly, delay increments of tens of milliseconds on very last hop may jeopardize these efforts.

5.2 Solution #1: Appropriately Setting MAC Layer Parameters

To improve the performance of real-time applications, part of the FTP throughput can be bartered with lower queuing delays. Specifically, by utilizing different buffer sizes and/or maximum number of retransmissions at the MAC layer, we can improve the performances achieved by the various real-time applications.

Starting with the first parameter, Figure 5 confirms that having larger buffer sizes at the MAC layer guarantees higher throughputs to TCP. Obviously, there is no difference in the achieved throughput when wireless losses are frequent enough to bind the TCP transmission rate below the pipe size. On the other hand, we have already anticipated that having large buffers along the path may augment the total delay time experienced by packets. In fact, each packet waits in queue for a time which proportionally grows with the number of preceding packets already present in the same buffer at its arrival. In case of intense traffic, buffers tend to be congested and hence queuing delays may become a significant component of the global delays experienced by each packet.

Focusing on the second parameter, although it is true that having a high maximum number of retransmissions at the MAC layer improves the reliability in packet delivery, it also increments the delivery time of packets waiting in queue for being transmitted. Therefore, a more appropriate configuration of the IEEE 802.11g with respect to the traditional one would probably make use of a maximum number of 3 retransmissions, thus guaranteeing a high FTP throughput whilst maintaining a low jitter in packet delivery time. Moreover, when a unique queue is maintained for all the traffic flows, a small size (50 packets at most) should be preferred.

This configuration ensures an elevate TCP throughput and also a reduction of the jitter experienced by real-time packets. As a demonstration, Figure 5 shows that the TCP total throughput during the 45 seconds when the FTP was running is 55371 packets (the congestion window, slow start threshold, and bandwidth-RTT product are shown in Figure 6). The corresponding reduction of the jitter for the online game flow traveling from the server to the client can be noticed by comparing values in Figure 7 (appropriate setting of MAC layer parameters) with those presented in Figure 4 (standard configuration).

An even better jitter could be gained further diminishing the maximum number of MAC retransmissions to 2. However, we advice against this choice because it may sensibly reduce the FTP throughput as can be observed in Figure 5.

5.3 Solution #2: Limiting TCP's Advertised Window

In order to implement SAP-LAW, we have enhanced the simulated scenario by enabling the AP to modify the advertised window (included in returning acks) accordingly with (1). In particular, the average UDP-based aggregate traffic was computed through a simple low-pass filter and the new advertised window was determined every 200ms.

Various values for the parameter C in (1) have been tested and results are reported in Figure 8. In this chart, we can see the average, the variance, and the maximum value for the jitter experienced by the game flow directed from the server to the client. Moreover, Figure 8 also presents the throughput trend of the concurrent TCP-based flow.

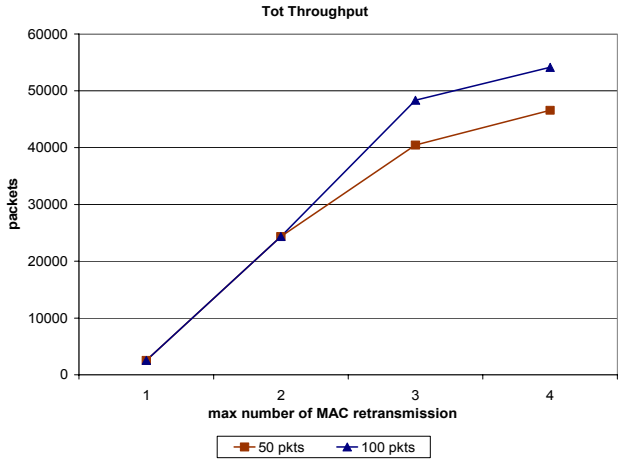

Figure 5. FTP total throughput with different MAC buffer sizes; user-AP distance = 10m, shadowing deviation = 9.

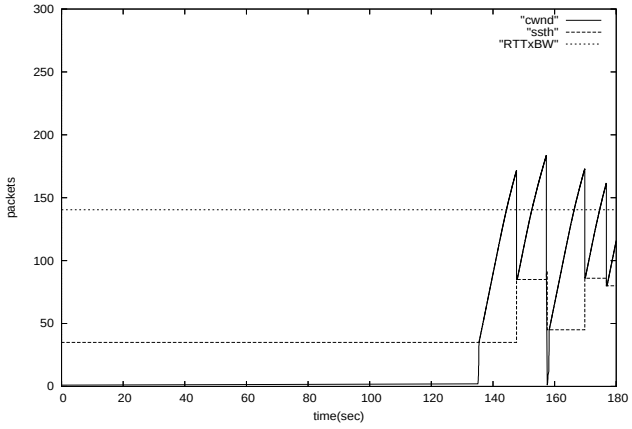

Figure 6. Measured TCP NewReno congestion window with max 3 retransmissions at the IEEE 802.11g MAC layer.

As clearly shown, both the average and the variance of the online game flow increase when we utilize higher values for C. This is coherent with the fact that higher C values decrease the resilience of the scheme to TCP bursts thus leading to some queuing at the AP. While the average results are very low for all C values, the variance sensibly increases with higher values of C thus indicating the presence of many peaks of very high delay in the packet delivery. This is confirmed also by the line representing the maximum delay value experienced by packets.

Figure 8 also demonstrates how the throughput decreases when C

is set too low. Instead, if C is set higher than the maximum achievable throughput on the channel (in this case, 20Mbps), then the sender will be allowed to send more packets than those bearable by the bottleneck link causing queuing delays. Thus, it happens that some packets may overflow the buffer and the consequent losses cause the reduction of the sending window and average throughput.

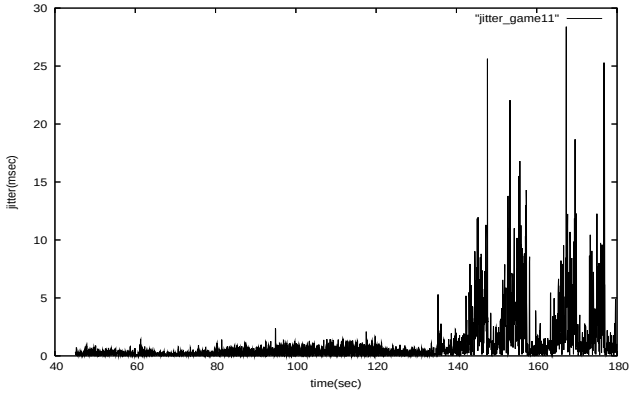

Figure 7. Measured online game jitter with max 3 retransmissions at the IEEE 802.11g MAC layer; from 135s, a regular TCP NewReno flow is competing for the channel.

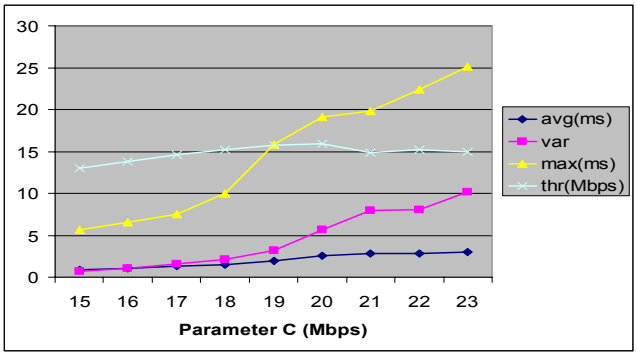

Figure 8. Throughput achieved by the FTP flow and jitter statistics of the game flow when employing SAP-LAW.

Supposing that we want to limit the maximum jitter within 10ms, we have to set C equal to 18Mbps (i.e., the 90% of the maximum achievable bandwidth). Indeed, this seems to be an appropriate choice able to guarantee both low queuing delays and high TCP efficiency. The advertised window exploited by the TCP flow is evident in Figure 9, which also reports the congestion window, the slow start threshold, and the bandwidth-RTT product. We have to keep in mind that the TCP flow starts at second 135 of the simulation time and that the actual sending window is determined as the minimum between the advertised window and the congestion window. This said, we can appreciate from the chart how the AP is able to keep track of the concurrent real-time traffic and determine the most appropriate advertised window. In particular, for this configuration, the final throughput in terms of acknowledged packets over 45 seconds hits 58677, while the jitter experienced by online game packets is kept low (see Figure 10).

Following (1), when only one TCP flow is running, SAP-LAW sets its advertised window close to the difference between the bandwidth-RTT product and the aggregate UDP-based traffic. This difference also represents an estimate of the amount of real-time (UDP-based) traffic present on the channel and, as Figure 9 shows, its value is relatively small if compared to the whole channel capacity. This demonstrates that real-time applications generally do not have to face bandwidth shortage in an 802.11g wireless home, while they still have to deal with high and variable delays.

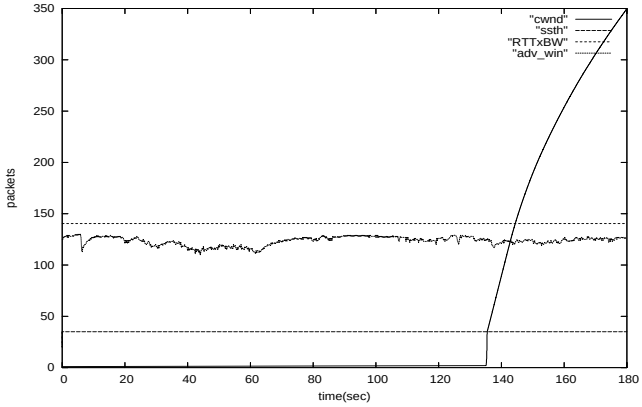

Figure 9. Measured congestion window and advertised window of a SAP-LAW flow with C = 18Mbps; regular IEEE 802.11g employed.

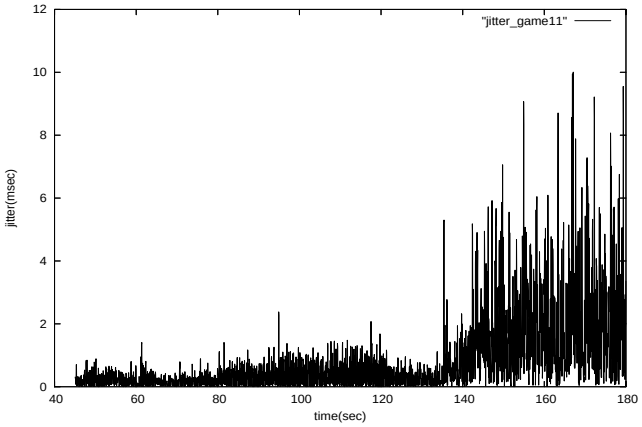

Figure 10. Online game jitter with regular IEEE 802.11g employed; from 135s, a SAP-LAW flow with C = 18Mbps is competing for the channel.

5.4 Summarizing Results

In Figure 11 we summarize statistical results obtained by: i) utilizing regular TCP New Reno on a standard IEEE 802.11g MAC configuration (Regular), ii) appropriately setting the MAC layer parameters (MAC-Setting), and iii) employing SAP-LAW.

The compared statistical parameters are the average, the variance, and the maximum value of jitter experienced by online game packets traveling from the server to the client via the AP. Again, results obtained from the other real-time applications running in the simulated scenario (i.e., video-stream and video-chat) are coherent with the showed ones and need no further explanation; we hence skip to present their outcomes. Rather, we also show the average throughput achieved by the concurrent TCP connection.

As it is evident, employing SAP-LAW to support FTP traffic is the solution that would guarantee the best performance both in terms of lowest per-packet delay and achieved throughput. Moreover, SAP-LAW could be easily implemented as it only requires the presence of slightly "smarter" APs. The modifications to the AP are very limited, thus minimally impacting on their cost and, at the same time, SAP-LAW can perfectly coexist with the current Internet and its employed protocols. Considering this and the remarkable results achieved, SAP-LAW represents the optimal candidate for enhancing computer-centered home entertainment in a wireless scenario.

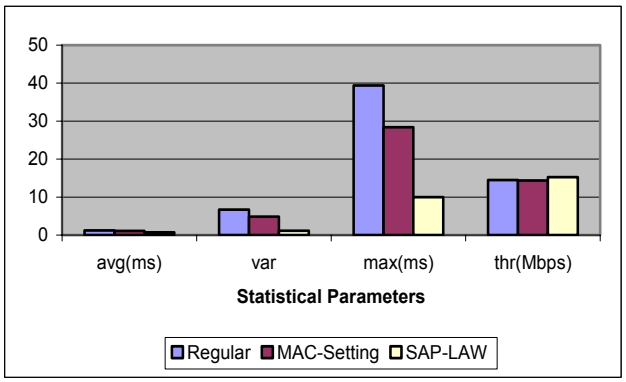

Figure 11. Statistical values of the online game stream for the compared schemes.

6. CONCLUSION

In this paper, we evaluated a scenario involving in-home entertainment delivered to wireless device through a HEC. A discussion has been provided that analyzes the mutual influence among several concurrent transmission streams in this context. We investigated the impact of the underlying wireless technology and showed how even a single persistent TCP connection can conspicuously increase the queuing delay suffered by concurrent real-time entertainment applications.

To solve this problem, we proposed SAP-LAW: a solution that exploits regular features of TCP and an enhanced AP to optimize the performances of both TCP and UDP-based transmission streams. In particular, our scheme snoops the on-going traffic through the AP and appropriately assigns an upper bound to the advertised window of TCP flows. We compared SAP-LAW to a solution that acts at the MAC layer by optimizing parameters setting and showed how the former outperforms the latter by consistently ameliorating the global performance of computer-centered home entertainment services. However, the two solutions are not incompatible with each other and could also be employed together.

Finally, even if in our model we assumed to have the bottleneck located in correspondence of the wireless link, the considerations we expressed in this paper, as well as the results that we showed, could be easily extended to a different scenario where a bottleneck is located before entering the edifice. We reserve to analyze this case as a future work.

ACKNOWLEDGMENTS

Partial financial support for this work is provided supported by: the Italian MIUR (under the ICTP/E-Grid, Interlink, MOMA, DAMASCO initiatives); the National Science Foundation (through grants CNS-0435515/ANI-0221528); and STMicroelectronics (under the UC-CoRe Grant MICRO 05-06).

REFERENCES

[1] The TiVo Homepage. http://www.tivo.com/

[2] Windows XP Media Center Edition 2005 Home Page. http://www.microsoft.com/windowsxp/mediacenter/

[3] C. E. Palazzi, G. Pau, M. Roccetti, M. Gerla, "In-Home Online Entertainment: Analyzing the Impact of the Wireless MAC-Transport Protocols Interference", *in Proc. of IEEE International Conference on Wireless Networks, Communications and Mobile Computing (WIRELESSCOM 2005)*, Maui, HI, USA, Jun 2005.

[4] H. Jiang, C. Dovrolis, "Why is the Internet traffic bursty in short (sub-RTT) time scales?", *in Proc. of ACM SIGMETRICS 2005*, Banff, AL, Canada, Jun 2005.

[5] Broadband Speed Tests. http://www.dslreports.com/stest

[6] Verizon Online DSL. http://www.verizon.com/dsl/

[7] AT&T Worldnet DSL Service. http://www.att.net/dsl/

[8] IEEE, "Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," Specifications, ISO/IEC 8802-11:1999(E), 1999.

[9] L. Pantel, L. C. Wolf, "On the Impact of Delay on Real-Time Multiplayer Games", *in Proc. of the 12th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, Miami, FL, May 2002.

[10] W. R. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*, Addison Wesley, 1994.

[11] A. Balk, M. Gerla, M. Sanadidi, D. Maggiorini, "Adaptive Video Streaming: Pre-encoded MPEG-4 with Bandwidth Scaling", *Computer Networks: The International Journal of Computer and Telecommunications Networking*, *15*, 5, 2004, 415-439.

[12] L. L. H. Andrew, S. V. Hanly, R. G. Mukhtar, "CLAMP: Active Queue Management at Wireless Access Points", in *Proc. of the 11th European Wireless Conference 2005*, Cyprus, April 2005.

[13] The Network Simulator, NS-2. http://www.isi.edu/nsnam/ns/

[14] A. L. Wijesinha, Y. Song, M. Krishnan, V. Mathur, J. Ahn, V. Shyamasundar, "Throughput Measurement for UDP Traffic in an IEEE 802.11g WLAN", *in Proc. of 6th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks (SNPD/SAWN'05)*, Towson, MD, USA, May 2005.

[15] Movie Trace Files. http://www-tkn.ee.tu-berlin.de/research/trace/ltvt.html

[16] J. Farber, "Traffic Modelling for Fast Action Network Games", *Multimedia Tools and Applications, 23*, 1, 2004, 31-46.