Reducing Queuing Delays through VoAP

Claudio E. Palazzi, Armir Bujari
Università degli Studi di Padova
Padova, Italy
{cpalazzi, abujari}@math.unipd.it

Silvia Mirri
Università degli Studi di Bologna
Bologna, Italy
silvia.mirri@unibo.it

***Abstract*— Cloud-based entertainment is becoming a hot topic for researchers and practitioners. In this context, thin client or cloud based gaming promise to attract users with the possibility to dematerialize the console and let the user play any possible game for any platform through the Internet. Basically, the computational burden is left to the cloud while the user enjoys a fat video feed accessed through her/his thin client via the shared wireless gateway. Unfortunately, this model of interaction also raises new complex technical issues that requires specific networking solutions. In this short paper we aim at addressing the heterogeneous flow coexistence problem at the home wireless gateway. We propose a solution based on a TCP Vegas-like congestion control algorithm deployed on top of the home gateway. Our solution works out of the box with the standard protocols at server, router and client level, thereby making deployment straightforward.**

Keywords—Smart home gateway; TCP Vegas; Thin client game

I. INTRODUCTION

Thin client gaming is a representative example of cloud based gaming. Through this technique, the game engine resides in the provider's cloud and the outcome is streamed directly as a video content to the players' devices to be accessed through a thin client [1]-[4]. This allows consumers to access the game without a console, making the end-device unimportant as the computing burden is shifted to the cloud gaming system that collects the users' actions and then renders, decodes and streams back the result.

Unfortunately, due to significant and unsolved technical challenges, this interaction model remains in its early stages without widespread deployment [6], [7]. In this context, the home gateway embodies a potential bottleneck that, if not dealt properly, could hinder the gaming experience. One of the main issues is represented by the contrasting modus operandi of the underlying transport protocols (TCP and UDP) supporting different applications sharing the same wireless channel. Real-time applications such as cloud gaming require fast delivery of gaming events and updates, whereas elastic, download sessions requires high data rates. Furthermore, cloud based games generate fat video streams consuming a lot of the available bandwidth [1], [8]. This jeopardizes the interactivity and responsiveness of gaming sessions, with TCP's aggressive congestion control behavior being the main cause [9], [10].

To address this issue we could rely on delay-based TCP congestion protocols, employing packet Round Trip Time (RTT) rather than losses to prevent congestion. In this realm, TCP Vegas is the main candidate [11]. However, its adoption is hindered by the widespread deployment and use of loss-based protocols. The only way to employ TCP Vegas in practice would be to completely dismiss all loss-based TCP versions and, needless to say, this is not a feasible option.

In this article, we show how to practically exploit the benefits of delay-based congestion control algorithms such as the one employed by TCP Vegas, to ensure a smoother network traffic progression. In summary, the gateway is enhanced with an algorithm that automatically limits TCP flows through their advertised window when the channel is near saturation, thus avoiding long queues (and queuing delay) as well as packet loss, while keeping the TCP-based flows at a data rate that corresponds to a full utilization of the available bandwidth. At a high level, this approach is similar to TCP Vegas, even though the algorithms clearly have significant differences in employment, hence the name Vegas over Access Point (VoAP). As a result, elastic applications achieve high data throughput, while real-time, gaming sessions maintain a low latency. Also, by intervening only at the wireless gateway, we guarantee a feasible deployment of the solution.

II. VEGAS OVER ACCESS POINT (VOAP)

With the regular TCP protocol, the actual sending rate (i.e., the sending window) is computed as the minimum between the sender's congestion window and the receiver's advertised window. Instead, VoAP dynamically modifies the advertised window to limit the growth of the TCP flow's sending rate so as to maintain the sending rate of the TCP flows high enough to efficiently utilize the available bandwidth and, at the same time, limited in its growth so as to not overutilize the buffers. Consequently, the queuing delay experienced by each packet is reduced, while the throughput is kept high by the absence of packet losses (used by TCP to halve the congestion window). Furthermore, by exploiting an existing feature of regular TCP implementations we also limit to the home gateway the modifications required by our solution.

Conversely from regular TCP, with our solution the advertised window is determined by VoAP's algorithm implemented on the home gateway. This is a preferable location as We decide to install VoAP on the home gateway represents the bottleneck of the connection and all communication should pass through this device.

The basic functioning is very simple (and effective). VoAP monitors the ongoing traffic and the current queuing delay experienced by packets transiting through its buffer. When the queuing delay increases above a certain threshold, VoAP proceeds with *appropriate* on-the-fly modifications of the advertised window in TCP ACKs so as to limit TCP flows just below the congestion level.

In essence, taking inspiration from TCP Vegas' congestion control algorithm, VoAP employs two thresholds: α and β. If the measured queuing delay is below the threshold α, the network is definitely not congested and the algorithm linearly increases the window, allowing the TCP to increase its transfer speed. Instead, if the queuing delay lays between the two thresholds, α and β, then we can assume that the network is well utilized but not yet congested; for this reason the flows are not acted upon. Finally, when the delay surpasses β the network can be considered as saturated; thereby the TCP flows are slowed down by linearly decrementing their advertised windows communicated though the ACKs.

The parameters α and β are crucial for our solution. However, a coarse tuning is more than enough to generate positive effects. In particular, α has to be set big enough to let some packets in the buffer so as to not slow down transmissions; we have found that an α of 5 ms was more than enough. On the other hand, β embodies the ideal maximum queuing delay we intend to suffer at the home gateway. Considering that any game packet should be delivered within 150 ms (including transmission, propagation, handling, queuing, etc.), we decided to set β equal to 15 ms [10]-[14].

III. Experimental Evaluation

To evaluate the efficacy of VoAP we have created a realistic testbed exploiting a well-known and reliable simulation environment: Network Simulator 2 (NS2).

In the considered scenario (see Fig. 1) some wireless devices are connected to the home gateway (where VoAP is deployed). We can also see some wired nodes that represent remote resources in the cloud. The wired connections have a 100 Mbps capacity, whereas the wireless channel is a IEEE 802.11g representing the bottleneck of the connection. The home gateway is a Wi-Fi Access Point (AP) configured with a buffer size of 250 packets to realistically represents real implementations [15]. The one way delay between the resources (servers) and clients is 40 ms and the flows we run include a FTP/TCP download and a game flow composed by a server-client game video and a client-server game commands stream.

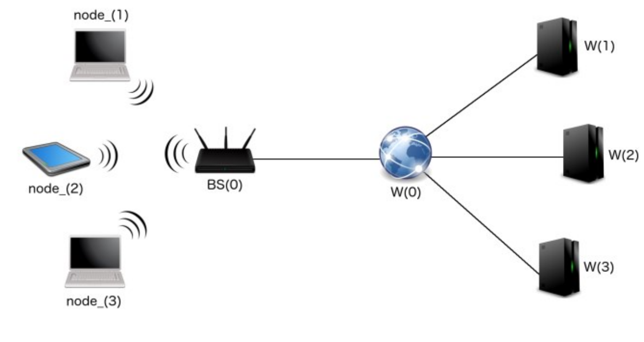

Fig. 1. The simulated scenario in our experiments.

TABLE I. Game Flows Employed in our Tests

Game	Limitation with respect to Unlimited	Bandwidth
UT	2/3	4200 Kbps
UT	1/3	2000 Kbps
Rome	Unlimited	3817 Kbps
Rome	2/3	2500 Kbps
Rome	1/3	1270 Kbps

To increase the reliability of our tests we have employed the real game traffic traces gathered in [1], which have been collected using the OnLive thin client game system with different categories of games. In particular, we have utilized traces corresponding to Unreal Tournament III (UT) and Grand Ages: Rome (Rome). As an important characteristic of thin client game systems is the ability to adapt the quality of the streamed video based on the network conditions, we have considered different possible video rates. In particular, Table I reports the different game flow configurations we have tested along with the required data rate for video streaming. For all of them, the game action stream from client to server was always 125 Kbps for UT and 100 Kbps for Rome.

The outcome of the experiments, averaged over 50 simulation runs, is reported in the charts. In particular, Fig. 2 shows the average throughput achieved by the FTP/TCP flow. As it is evident, even if VoAP limits the sending window of TCP, the final throughput is not decreased. On the other hand, Fig. 3 demonstrates the effectiveness of VoAP in reducing the average queuing delay experienced by any packet transiting through the home gateway.

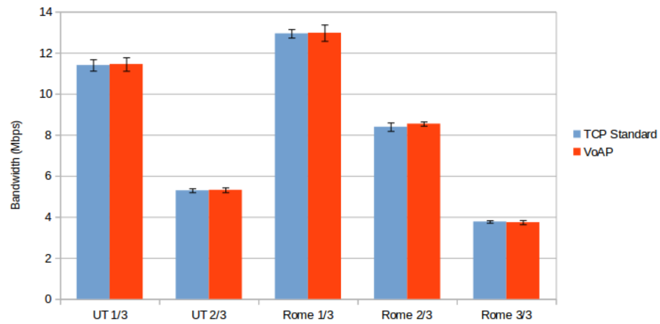

Fig. 2. Average throughput achieved by the FTP/TCP flow.

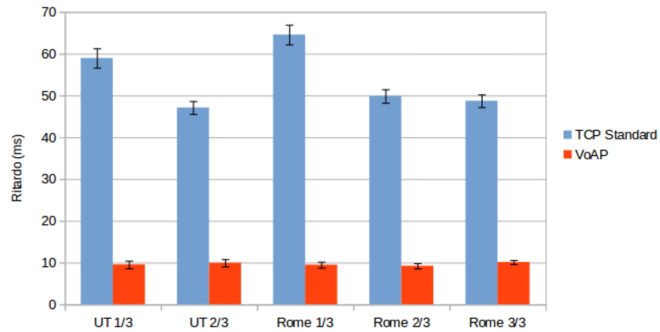

Fig. 3. Average queuing delay experienced at the home gateway.

IV. CONCLUSION

We have considered an in-home wireless scenario with various clients connected to the Internet to run different applications such as download and thin client gaming. We have shown how this coexistence is not easy and generates interactivity harmful queuing delays. We have also discussed a solution named VoAP that is easily implementable and that effectively reduces the queuing delay for online games while preserving the throughput of elastic flows. As future work we plan to study the feasibility of our solution in a mesh network scenario [16].

ACKNOWLEDGMENT

Our sincere gratitude goes to Michele Massaro for his help in the technical development of the testbed.

REFERENCES

[1] M. Claypool, D. Finkel, A. Grant, M. Solano, "Thin to Win? Network Performance Analysis of the OnLive Thin Client Game System", in Proc. of the ACM Annual Workshop on Network and Systems Support for Games (NetGames), Venice, Italy, 2012.

[2] A. Lai and J. Nieh, "On the Performance of Wide-Area Thin-Client Computing," ACM Transactions on Computer Systems, May 2006.

[3] Yu-Chun Chang, Po-Han Tseng, Kuan-Ta Chen, Chin-Laung Lei, "Understanding the Performance of Thin-Client Gaming", in Proc. of IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR), Naples, FL, USA, 2011.

[4] M. Furini, "Mobile Games: What to Expect in the Near Future", in Proc. of GAMEON Conference on Simulation and AI in Computer Games, Bologna, Italy. 2007.

[5] R. Shea, L. Jiangchuan, E. C.-H. Ngai, C. Yong, "Cloud Gaming: Architecture and Performance", IEEE Network 27(4), 2013.

[6] D. Wu, Z. Xue, J. He, "iCloudAccess: Cost-Effective Streaming of Video Games From the Cloud With Low Latency", IEEE Transactions on Circuits and Systems for Video Technology 24(8), 2014.

[7] A. Kaiser, D. Maggiorini, N. Achir, K. Boussetta, "On the Objective Evaluation of Real-Time Networked Games," in Proc. of IEEE Global Telecommunications Conference (GLOBECOM), Honolulu, HI, USA, 2009.

[8] 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 the IEEE International Conference on Wireless Networks, Communications and Mobile Computing (WIRELESSCOM), Maui, HI, USA, 2005.

[9] C. E. Palazzi, S. Ferretti, M. Roccetti, G. Pau, M. Gerla, "What's in That Magic Box? The Home Entertainment Center's Special Protocol Potion, Revealed", IEEE Transactions on Consumer Electronics 52(4), 2006.

[10] S. Low, L. Peterson, L. Wang, "Understanding TCP Vegas: Theory and Practice", Princeton University, Princeton, New Jersey, 2000.

[11] R. Shea, L. Jiangchuan, E. C.-H. Ngai, C. Yong, "Cloud Gaming: Architecture and Performance," IEEE Network 27(4), 2013.

[12] S. Choy, B. Wong, G. Simon, C. Rosenberg, "The Brewing Storm in Cloud Gaming: A Measurement Study on Cloud to End-User Latency," in Proc. of the IEEE Network and Systems Support for Games (NetGames), Venice, Italy, 2012.

[13] M. Gerla, D. Maggiorini, C. E. Palazzi, A. Bujari, "A Survey on Interactive Games over Mobile Networks", Wiley Wireless Communications and Mobile Computing 13(3), 2013.

[14] A. Kaiser, D. Maggiorini, N. Achir, K. Boussetta, "On the Objective Evaluation of Real-Time Networked Games," in Proc. of IEEE Global Telecommunications Conference (GLOBECOM), Honolulu, HI, USA, 2009.

[15] T. Li, D. Leith, D. Malone, "Buffer Sizing for 802.11 Based Networks", IEEE/ACM Transactions on Networking 19(1), 2011.

[16] A. Hamidian, C. E. Palazzi, T. Y. Chong, J. M. Navarro, U. Körner, M. Gerla, "Deployment and Evaluation of a Wireless Mesh Network", in Proc. of the 2nd International Conference on Advances in Mesh Networks (MESH 2009), Athens, Greece, 2009.