

Ensuring Coexistence among Games and Downloads in Multihop Wireless Networks

Claudio E. Palazzi, Armir Bujari
University of Padua
Via Trieste 63 - 35123 - Padova - Italy
Email: {cpalazzi,abujari}@math.unipd.it

Gustavo Marfia, Marco Rocchetti
University of Bologna
Mura Anteo Zamboni 7 - 40127 Bologna - Italy
Email: {marfia,rocchetti}@cs.unibo.it

Abstract—Mobile communication devices are expected to overtake traditional desktop PCs and become the dominant point of entry to the Internet. This technology shift has boosted user demand for multimedia applications and services which are now typically accessed through the wireless interface. Users expect to be always connected and able to run a variety of applications which sometimes impose contrasting requirements. Of particular research interest in this scenario is the coexistence between two popular categories of traffic: elastic and real-time, both of them covering a wide range of user applications. In this paper, we investigate the above issue in the scenario where service connectivity is provided through a multihop wireless network. To this end, we study two representative applications, one for each of the above categories, which are content download and interactive game play. We show by means of extensive simulations, how both the content download’s goodput and the online game’s per-packet delivery delay can be improved through an appropriate solution implemented on the wireless Internet Gateway.

I. INTRODUCTION

Typically, wireless communication involves a single hop; yet, multihop wireless networks are gaining a lot of attraction from the research community [1], [2]. This trend is being driven by the rapid deployment of 802.11 networks and the advantages of relaying packets among nodes. In infrastructure rich areas, relaying can reduce dead spots, lower power consumption [3], [4], and increase network capacity [5]. In rural or developing areas, multihop wireless networks can be deployed more readily and at lower expense than traditional wireless networks. To this purpose, we consider an extension to the classical, single-hop scenario where Internet connectivity is provided via wireless multihop channel and users run heterogeneous applications such as online gaming and content download, which generally pursue different goals [6], [7].

Supporting both real-time and elastic traffic, as online games and content download, respectively, is a tough challenge as they pursue different performance goals: online games require fast delivery of every single game event, whereas high data rate is desirable when downloading a data content. Unfortunately, in the attempt to reach high data rates for content download, the classic Transmission Control Protocol (TCP) periodically saturates links and buffers along its path, generating queuing delays affecting the transmission time of game events [8]. Moreover, flows in multihop wireless networks are affected by self-interference, by which packets of the same flow transmitted on adjacent hops compete for the shared wireless medium.

In summary, the coexistence between TCP based content download and online game flows over multihop wireless networks embodies a tough technical challenge that deserves scientific investigation. Although extensive research has been devoted to improve TCP-based flow performance over multihop networks, little has been done to support real-time applications such as online game flows in this heterogeneous context [9]. Recently, research studies demonstrated that neither classic nor enhanced existing variants of TCP and 802.11 are able to protect real-time traffic in wireless multihop scenarios [1].

Online games have been demonstrated to be a killer application able to attract millions of users. Therefore, we address the multihop wireless scenario where network resources are shared by online game flows and TCP based content transfer. We propose an original solution able to support online gaming performance (i.e., small transmission delays) while preserving TCP performance (i.e., high data transfer goodput).

Our solution is based on an enhanced wireless Internet Gateway that is able to monitor ongoing traffic so as to appropriately modify, on the fly, the advertised window in TCP’s ACKs. Indeed, the Internet Gateway is in a strategic position to gather information about the channel condition and the on-going traffic. The idea is to exploit this gathered information to regulate heterogeneous transmission flows and make them coexist efficiently, even on a wireless multihop connection.

As our solution is based on the utilization of an enhanced gateway, we have named it Smart Wireless Internet Gateway (SWING).

The two main ideas at the basis of SWING are as follow.

- 1) The wireless Internet Gateway continuously snoops all the transiting packets and computes the maximum data rate at which downloading applications can transfer their files without exceeding the factual available bandwidth given the other simultaneous flows and the multihop wireless path.
- 2) A cross-layer approach is implemented to modify on-the-fly the advertised window of each transiting TCP packet so as to make it equal to the aforementioned maximum data rate, and accordingly limit the transfer rate of TCP flows.

This way, download applications produce a smoother traffic efficiently utilizing the available channel without generating queues along the multihop wireless path that would create

losses and delays. In summary, both the elastic applications' goodput and the real-time applications' per-packet delivery delay are preserved. We do not propose new protocols or architectures, thus ensuring factual deployability of our solution. Indeed, we simply design an enhanced gateway with a re-engineered packet forwarding functionality. Customers just need to get our enhanced gateway from their preferred electronics seller and plug-it-in.

Results gathered and discussed in this paper demonstrate that the performance of entertainment applications in wireless multihop environments are improved by SWING with respect to the use of both traditional protocols and state of the art proposals falling in this domain.

The remainder of the paper is organized as follows. In Section II, we provide the motivation behind this study along with the considered scenario. In Section III we review related works in this context, outlining our contribution with respect to existing studies. In Section IV and Section V, we present SWING and the experimental results demonstrating its effectiveness, respectively. Finally, Section VI concludes the paper providing some insights on future research directions we aim to investigate.

II. PROBLEM STATEMENT AND CONSIDERED SCENARIO

Recent advances on wireless technology, the expansion of the broadband Internet, and the increasing demand for digital content call for intelligent devices and networking solutions. These solutions should provide connectivity between devices and efficiently exploit the high-speed access to the Internet, extending its coverage everywhere, directly into the hands of the consumer.

Content download and online gaming represent two killer applications we use in our everyday life [8]. However, they are characterized by different requirements: the former aims at achieving as much goodput as possible, whereas the latter needs very fast per-packet delivery delay (there should be less than 150 ms between the generation of a game event and its delivery to the other players [10]–[12]). This problem and the peculiar challenge represented by the multihop wireless scenario deserve further investigation.

Starting from the transport layer, we can group multimedia applications into two main classes depending on the adopted protocol: the Transmission Control Protocol (TCP) or the User Datagram Protocol (UDP). TCP is reliable and ensures ordered delivery of every packet sent. For this reason, it is utilized by elastic applications that involve content download/transmission. A very important component of TCP is represented by its congestion control functionality, which probes the connection between sender and receiver with higher and higher data rates eventually filling up the channel. At this point, packets are queued at the buffer associated with the bottleneck of the connection until it overflows causing packet losses and consequent reduction of the sending rate. On the other hand, UDP packets are immediately sent toward the receiver with a data rate decided by the sender without any delivery guarantee; yet, its small overhead and lack of retransmissions may generate less delay than TCP in the packet delivery. UDP is hence usually employed by real-time applications that can

tolerate some packet loss but have stringent requirements in terms of per-packet delivery delay (e.g., online games).

The lack of congestion control functionalities has been the reason for considering UDP unfair toward TCP. On the other hand, a problem emerges when real-time applications (UDP-based) coexist with elastic ones (TCP-based) on a wireless channel, causing the former to experience a scattered flow progression. Major causes for this problem can be found in the TCPs congestion control functionality. In particular, TCP continuously probes for higher transfer rates, also queuing packets on the buffer associated with the bottleneck of the connection. If one considers that the same wireless connection might be shared by several devices and applications, thus increasing the congestion level and queue lengths, it is even more evident how packets can be delayed, jeopardizing the real-time requirement.

This negative scenario is further worsened by the following factors related to the wireless interface. First, the wireless medium allows the transmission of only one packet at a time and is not full-duplex as wired links. Packets have hence to wait for their turns to be transmitted. Second, as interference, errors, fading, and mobility may cause packet loss, the IEEE 802.11 MAC layer reacts through local retransmissions which, in turn, cause subsequent packets to wait in queue until the preceding ones or their retransmissions eventually reach the receiver. Furthermore, the back-off mechanism of the IEEE 802.11 introduces an increasing amount of time before attempting a transmission again. These three problems are obviously exacerbated when considering a multihop wireless network as in our scenario.

Flows in multihop wireless networks are also affected by self-interference, by which packets of the same flow transmitted on adjacent hops compete for the shared wireless medium. Indeed, in a network with two adjacent wireless hops, transmissions cannot occur on both links simultaneously, thus halving the available bandwidth. Increasing the number of hops complicates the scenario. The available bandwidth is further reduced due to alternate transmissions on adjacent wireless links, this, until reaching a number of hops greater or equal to four. At this point, with only about 25% of bandwidth actually available, transmissions can be pipelined with two hops of distance thus (theoretically) not further reducing the exploitable bandwidth in a significant way [9], [13].

Here we set on the trail to search for an appropriate network solution able to support multimedia applications, and their coexistence, over a multihop wireless scenario. Our solution is designed to be noninvasive and with real deployment chances. To this purpose, we have considered the scenario depicted in Fig. 1. The case study is composed by a wired Internet part and a multihop wireless part, both interfaced by a wired-wireless gateway. We consider several clients which share a four hop wireless path and the clients are responsible of generating three TCP flows and ten bidirectional game flows interacting with the game server. The rationale behind this deployment follows the argument exposed before, that of the multihop pipelining effect, which gives us an indicative trend for our implementation in a well founded theoretical framework.

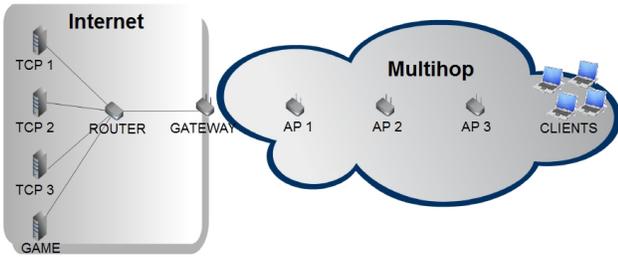


Fig. 1. Considered scenario.

III. RELATED WORK

A classic approach when dealing with networks that include wired and wireless links is that of splitting the connection at the node interfacing the wired and wireless parts, called the Internet Gateway [14]. This way, the wireless link issues are entirely hidden to the rest of the network and, in particular, to the TCP. However, this requires extra capabilities at the Internet Gateway and violates the end-to-end paradigm of TCP.

Similar, link layer approaches try to hide the wireless link layer issues to the TCP. Yet, these approaches generally try to preserve the end-to-end semantics. The Snoop protocol is certainly the most representative example of this class of solutions [15]. With this protocol, a snoop agent on the base station performs local retransmissions on the wireless link using information sniffed from current TCP traffic. Focusing on the multihop scenario, QoS scheduling with priority queues within intermediate nodes of a multihop network has been proposed [16]. Yet, it requires traffic that has been previously marked with the priority level and only guarantees priority to certain type of traffic without actually aiming at an efficient coexistence among heterogeneous traffic flows.

A technique that has emerged as an effective solution in supporting data transmission over wireless multihop networks is pacing. In essence, this class of solutions evenly space, or pace, data to be transmitted into the multihop so as to avoid bursts [17], [18].

In summary, a large amount of research has focused on the optimization of TCP performance in wireless multihop networks. Yet, very few works have focused on the performance of real-time applications, or on the interaction between real-time and elastic flows, in this context [8], [19]. In the following subsection we discuss in detail a solution which is considered the state of the art for supporting the coexistence of real-time and elastic flows in wireless multihop networks.

A. State of the Art: Gateway Adaptive Pacing

The Gateway Adaptive Pacing (GAP) algorithm is a proposal tackling the issue of data transmissions over heterogeneous environments composed of a multihop wireless part [1], [17]. It adopts a rate-based congestion control by tuning the transmission rate adaptively accounting for the spatial reuse constraint of multihop wireless networks and the contention on the network path of the connection.

In essence, the algorithm operates by delaying TCP packets queued at the gateway based on continuous delay measurements of the multihop environment [1]. Accounting for the

spatial reuse constraint is the Four Hop Delay (FHD) metric, which is the time employed by a packet to traverse the four hop wireless path. The estimate depends on the traffic a packet is subject to during its trip; thereby, the FHD changes depending on the path conditions.

To compute the FHD, (1) can be used,

$$FHD = 2 \left(\frac{RTT_{wireless}}{h} + \frac{s_{data} - s_{ACK}}{b} \right) \quad (1)$$

where h , b , $RTT_{wireless}$ are the number of hops in the wireless path, the wireless channel bandwidth and the RTT of the TCP packet in the wireless domain respectively. Instead, s_{data} and s_{ACK} denote the size of the TCP packet and the size of the corresponding ACK.

The rationale behind the FHD metric is the well known fact that in multihop wireless environments nodes cannot transmit in parallel without interfering on immediate neighbors transmissions [9], [13]. GAP ensures that at most one packet is transmitted each FHD, this way it avoids filling intermediate buffers in vain. On the other hand, real-time application packets, but in general all packets, are not subject to additional queuing delays.

IV. PROPOSED SOLUTION: SMART WIRELESS INTERNET GATEWAY

The TCP flow's actual sending rate (i.e., the sending window) is computed as the minimum between the congestion window and the advertised window. We hence propose an original solution based on dynamic modifications of the advertised window so as to limit the growth of the TCP flows' sending rates. A good tradeoff between throughput and low delays could be achieved by maintaining the sending rate of each TCP flow high enough to efficiently utilize the available bandwidth but, at the same time, limited in its growth so as to not utilize buffers. This way, per-packet delays are minimized by the absence of queues along the route from the sender to the receiver, while the throughput is kept high by the absence of packet losses that would halve the sending window and hence the sending rate.

In our solution, the appropriate advertised window is computed and imposed by the gateway between the wireless part of the network and the wired Internet (see Fig. 1). This gateway is in the position to handle all packets transiting to/from the multihop wireless part and hence to modify on the fly the advertised window of TCP ACK packets.

Our proposal is based on the utilization of an enhanced gateway which we have named Smart Wireless Internet Gateway (SWING). With SWING, the appropriate advertised window value expressed in packets for any TCP flow i has to be computed and continuously updated. Its initial value is given by (2), where BW_i , RTT_i and $TCP_{pktSize}_i$ are the bandwidth share, the round trip time (RTT) and the packet size, respectively, for that flow.

$$advwin_i = \frac{BW_i * RTT_i}{TCP_{pktSize}_i} \quad (2)$$

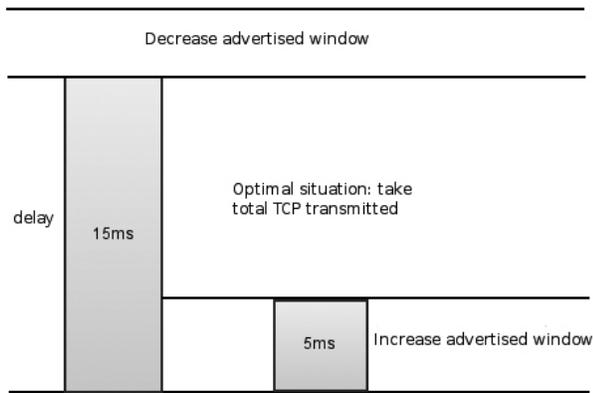


Fig. 2. Actions on the advertised window depending on the queuing delay experienced on the Gateway’s buffer.

More in detail, BW_i can be computed as (3), where C and TF are the total capacity of the channel and the number of TCP flows sharing the channel, respectively. RT is the amount of real-time traffic in the channel identified by UDP based flows (as online gaming). Inspired by [13], H can be computed as the minimum between 4 and the actual number of wireless hops.

$$BW_i = \frac{C - RT}{2 * H * TF} \quad (3)$$

We need only initial rough values for parameters C and RTT : the bandwidth stated by the Internet provider and the result of a ping to the server. After the initial setting, the advertised window is continuously updated to reflect the changes in the channel and in the number of flows sharing the multihop wireless network. To this end, we have taken inspiration from TCP Vegas’ congestion control algorithm.

In essence, the Gateway monitors how long packets stay in its buffer before being forwarded along the multihop wireless path. As depicted by Fig. 2, when this queuing delay is below a threshold th_{min} or above another threshold th_{max} , the advertised window imposed by the Gateway is increased or decreased, respectively, by one packet. While in between th_{min} and th_{max} , the imposed advertised window remains constant.

V. EXPERIMENTAL EVALUATION

We have implemented our solution on the well known ns-3 network simulator. To this aim, we have considered the scenario in Fig. 1: three servers for content download generating TCP-based flows with 100 ms, 140 ms and 240 ms of RTT, respectively, and a game server at 30 ms of one way delay from ten game clients located at the end of the wireless multihop path. Clients are connected to the Internet Gateway through four wireless hops using IEEE 802.11g technology. Following suggestions from [20] in modeling online gaming traffic, each of the ten game sessions corresponds to a bidirectional flow of packets departing every 40 ms with a payload of 40 B. The three TCP-based download flows start at simulated time 0 s and end at 200 s, whereas the ten online gaming flows start at simulated time 0 s and end at 250 s.

We compare SWING against a default setting scenario (classic TCP and UDP, with a regular Internet Gateway) and GAP. In the experiments, SWING’s queuing delay thresholds for modifying the advertised window have been set as $th_{min} = 5$ ms and $th_{max} = 15$ ms. These values are inspired by TCP Vegas so as to always have some packets in queue ready to be transmitted while limiting the queuing delay.

We compare the per-packet delivery delay of a representative game flow, considering default settings (Fig. 3), GAP (Fig. 4), and SWING (Fig. 5), respectively. The charts show how the per-packet delay of the online game flow suffers from the simultaneous presence of TCP flows (from 0 s to 200 s), whereas it goes back to values close to the one-way delay (30 ms) when only the ten online game flows, and not the TCP flows, are active (from 200 s to 250 s). Furthermore, while GAP clearly outperforms the default setting scenario in terms of smoother traffic and lower delays, SWING shows further improvement in these directions.

We also analyze the performance trend of the TCP flows so as to verify that their goodput is not affected by the presence of online game traffic. As shown by Fig. 6, Fig. 7 and Fig. 8 SWING results the only solution among the compared ones which succeeds in ensuring both low per-packet delays and efficient bandwidth utilization even in presence of heterogeneous flows over a multihop wireless network.

To further demonstrate SWING’s ability in avoiding queue utilization, we have measured the evolution of the queuing time spent by packets in the Internet Gateway’s buffer before being transmitted toward the next hop. In Fig. 9, Fig. 10 and Fig. 11 we show this queuing delay as experienced by a single representative game flow. It is evident how both GAP and SWING outperform the default setting scenario, with SWING being the winner, further decreasing the queuing delay.

VI. CONCLUSION AND ONGOING WORK

We addressed the problem of heterogeneous flow coexistence in the multihop wireless scenario, considering TCP (elastic) and UDP-based (real-time) traffic in particular.

TCP’s aggressive behavior coupled with a self-interfering multihop wireless environment can conspicuously increase the queuing delay suffered by real-time applications. Our solution, SWING, consists in an enhanced wireless gateway that does not need to modify existing Internet protocols. We do not propose new protocols or architectures, thus ensuring factual deployability of our solution.

SWING is based on dynamic modification of the advertised window so as to limit the growth of the TCP flows’ sending rate while preserving the per-packet delay of real-time traffic.

We compared SWING to GAP, a state of the art proposal in the domain, showing that our solution outperforms the latter. Yet, we deem necessary to further pursue our investigation into the following directions: (i) dynamic adaptation of th_{min} and th_{max} , depending on the bandwidth; (ii) monitoring TCP flows’ use of the assigned bandwidth share to redistribute unused bandwidth (e.g., in case of Telnet or Web traffic); (iii) comparing our solution against different TCP versions [21]; (iv) considering complex topology also including a multihop real testbed [22].

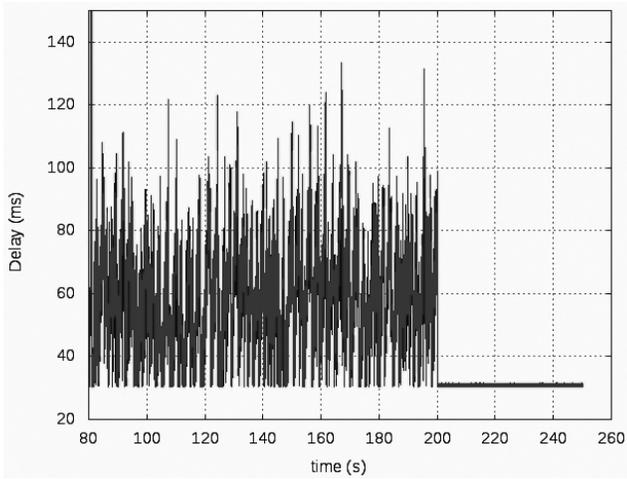


Fig. 3. Game flow: per-packet delivery delay with regular protocols.

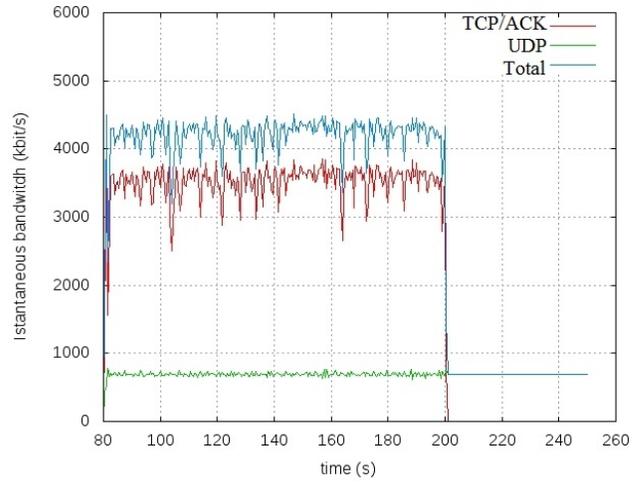


Fig. 6. Flows' goodput with regular protocols.

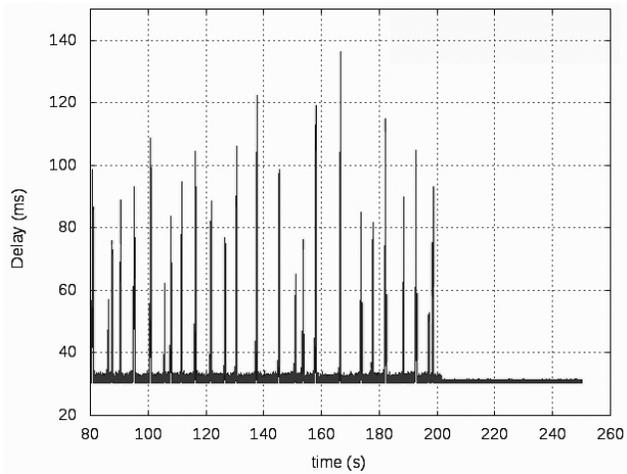


Fig. 4. Game flow: per-packet delivery delay with GAP.

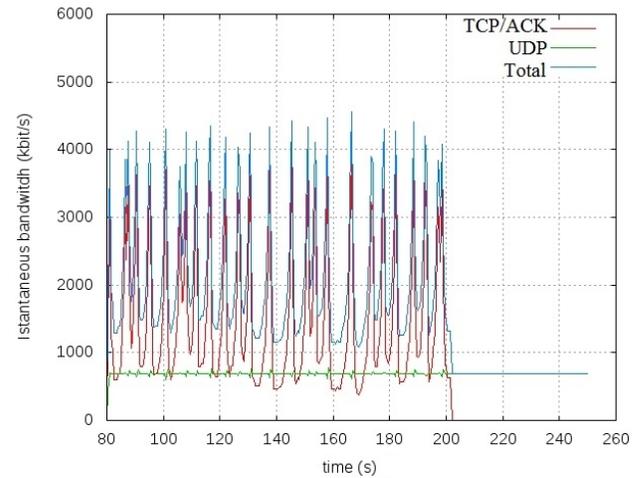


Fig. 7. Flows' goodput with GAP.

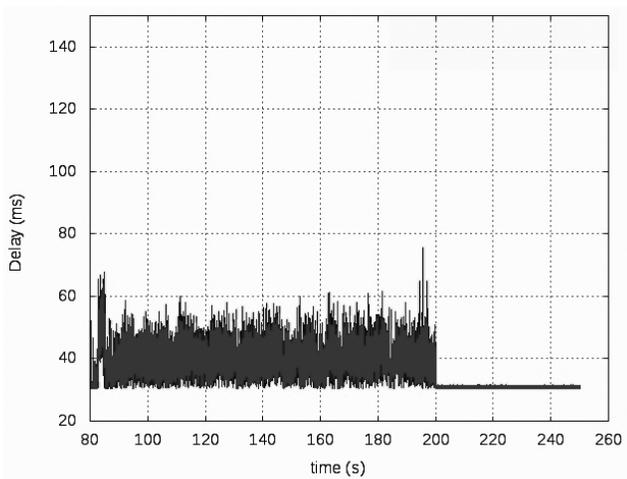


Fig. 5. Game flow: per-packet delivery delay with SWING.

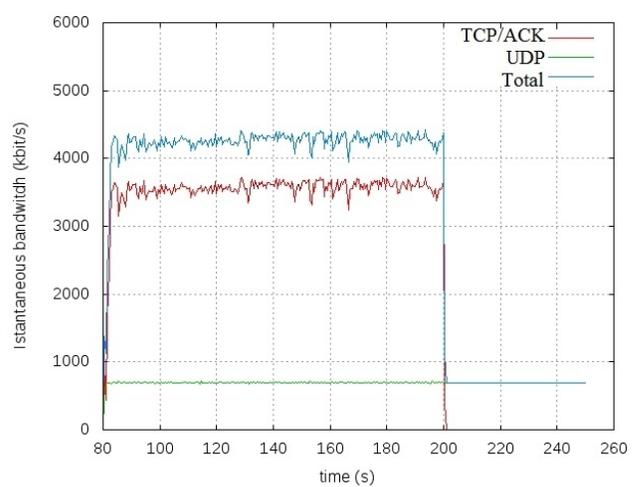


Fig. 8. Flows' goodput with SWING.

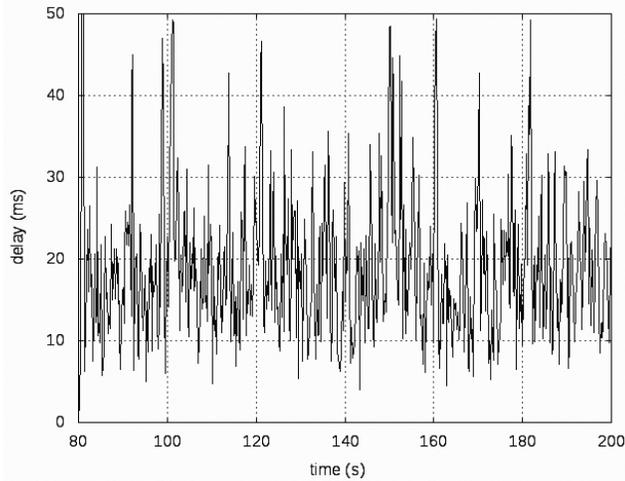


Fig. 9. Queuing delay with regular protocols.

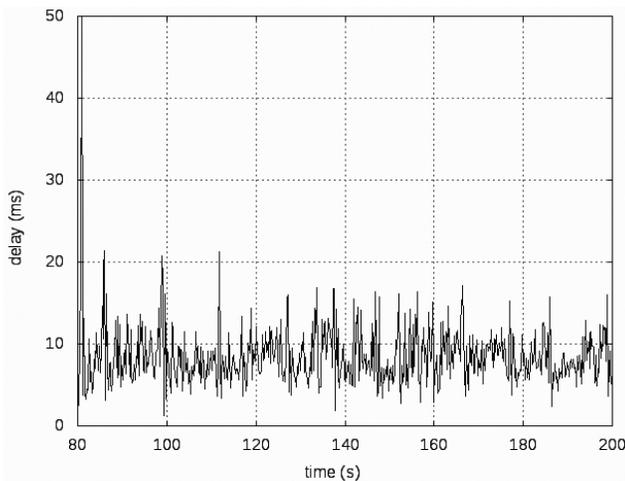


Fig. 10. Queuing delay with GAP.

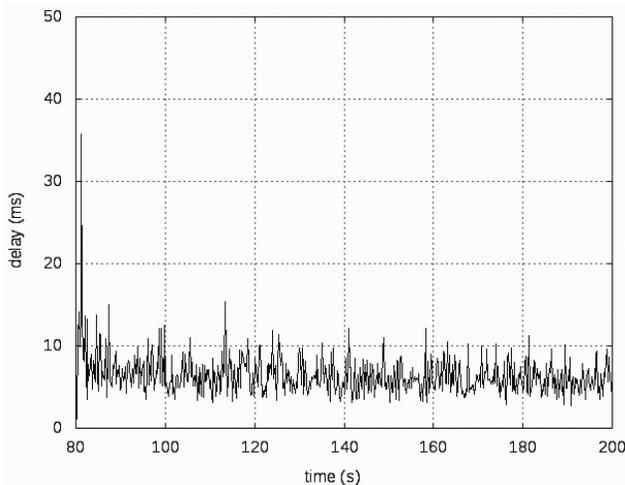


Fig. 11. Queuing delay with SWING.

REFERENCES

- [1] K. Kim, D. Niculescu, and S. Hong, "Coexistence of voip and tcp in wireless multihop networks," *IEEE Communications Magazine*, vol. 47, no. 6, 2009.
- [2] C. E. Palazzi, "Online Gaming and File Download Coexistence over Multihop Wireless Networks," in Proc. of the 32nd IEEE International Conference on Computer Communications (IEEE INFOCOM), 2013.
- [3] N. Swetha, K. Vikas, R. S. Sreenivas, and P. R. Kumar, "Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the compow protocol," in *European Wireless Conference*, 2002.
- [4] L. De Giovanni, C. E. Palazzi, "Optimal Client-Server Configuration of Mobile Ad-Hoc Networks" *Electronic Notes in Discrete Mathematics* 41, 2013.
- [5] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, 2000.
- [6] G. Marfia, C. E. Palazzi, G. Pau, M. Gerla, M. Y. Sanadidi, and M. Roccetti, "TCP Libra: Exploring RTT-Fairness for TCP," in Proc. of the IFIP/TC6 NETWORKING, 2007.
- [7] C. E. Palazzi, G. Pau, M. Roccetti, and M. Gerla, "In-Home Online Entertainment: Analyzing the Impact of the Wireless MAC-Transport Protocols Interference," in Proc. of the IEEE 2005 International Conference on Wireless Networks, Communications and Mobile Computing (IEEE WIRELESSCOM), 2005
- [8] C. E. Palazzi, S. Ferretti, M. Roccetti, G. Pau, and M. Gerla, "What's in that magic box? the home entertainment center's special protocol potion, revealed," *IEEE Transactions on Consumer Electronics*, vol. 52, no. 4, 2006.
- [9] Z. Fu, H. Luo, P. Zerfos, S. Lu, L. Zhang, and M. Gerla, "The impact of multihop wireless channel on tcp performance," *IEEE Transactions on Mobile Computing*, vol. 4, no. 2, 2005.
- [10] A. Kaiser, D. Maggiorini, K. Boussetta, and N. Achir, "On the objective evaluation of real-time networked games," in *IEEE Global Telecommunications Conference*, 2009.
- [11] M. Gerla, D. Maggiorini, C. E. Palazzi, and A. Bujari, "A survey on interactive games over mobile networks," *Wiley Wireless Communications and Mobile Computing*, vol. 13, no. 3, 2013.
- [12] L. Pantel and C. Wolf, "On the impact of delay on real-time multiplayer games," in *International Workshop on Network and Operating Systems Support for Digital Audio and Video*, 2002.
- [13] J. Li, C. Blake, D. S. De Couto, H. I. Lee, and R. Morris, "Capacity of ad hoc wireless networks," in *Proceedings of the 7th annual International Conference on Mobile Computing and Networking*. New York, NY, USA: ACM, 2001.
- [14] A. Bakre and B. Badrinath, "I-tcp: Indirect tcp for mobile hosts," in *International Conference on Distributed Computed Systems.*, 1995.
- [15] H. Balakrishnan, S. Seshan, E. Amir, and R. H. Katz, "Improving tcp/ip performance over wireless networks," in *ACM MobiCom*, 1995.
- [16] S. Xu and T. Saadawi, "Does the ieee 802.11 mac protocol work well in multihop wireless ad hoc networks?" *IEEE Communications Magazine*, vol. 39, no. 6, 2006.
- [17] S. M. ElRakabawy, A. Klemm, and C. Lindermann, "Gateway adaptive pacing for tcp across multihop wireless networks and the internet," in *ACM International Symposium on Modeling Analysis and Simulation of Wireless and Mobile Systems*, 2006.
- [18] S. Rangwala, A. Jindal, K.-Y. Jang, and K. Psounis, "Understanding congestion control in multi-hop wireless mesh networks," in *ACM MobiCom*, 2008.
- [19] T. Bu, Y. Liu, and D. Towsley, "On the tcp-friendliness of voip traffic," in *25th IEEE INFOCOM*, 2006.
- [20] J. Farber, "Traffic modelling for fast action network games," *Multimedia Tools and Applications*, vol. 23, no. 1, 2014.
- [21] G. Marfia and M. Roccetti, "Tcp at last: reconsidering tcp's role for wireless entertainment centers at home," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 4, 2010.
- [22] A. Hamidian, C. E. Palazzi, T. Chong, J. Navarro, U. Korner, and M. Gerla, "Deployment and evaluation of a wireless mesh network," in *2nd International Conference on Advances in Mesh Networks.*, 2009.