Multimedia Transmissions over Vehicular Networks

Armir Bujari, Claudio E. Palazzi, Daniele Ronzani
Dipartimento di Matematica
Università degli Studi di Padova
Padua, Italy
{cpalazzi, abujari, dronzani}math.unipd.it

Abstract—**With vehicular networks quickly becoming the new wireless frontier of the Internet, car passengers embody the next consumers that will be targeted by multimedia content providers. However, researchers and practitioners are still struggling to tackle the technical challenges that raise in this scenario such as, for instance, the continuous variations in the number and type of flows served by the access points along the road. Considering different traveling scenarios in terms of vehicles' speed and location we discuss how access points could be improved to properly address these challenges, ensuring coexistence among heterogeneous types of flow, even in presence of moving vehicles that generate frequent network traffic variations.**

Keywords—Infrastructure multimedia; smart access points; vehicular networks

I. INTRODUCTION

Vehicular networks are no more just a theoretical topic thanks to the factual interest shown by governments and car manufacturers, as well as to the market responsiveness [1]. Indeed, media frequently reports on innovation brought by vehicular connectivity planned by major car manufacturers. Furthermore, it is well known that the US and EU government have reserved the 5.9 GHz frequency spectrum to implement vehicular communication, both in infrastructure and ad-hoc mode, through the DSRC/IEEE 802.11p standard.

Vehicles will use their connections to improve safety but also to provide classic Internet applications such as web-surfing, social networks, email, video streaming, online gaming and file download [2]-[4]. This scenario poses significant technology issues that deserve scientific investigation. In this context, our aim is that of supporting heterogeneous multimedia traffic for passengers traveling in cars.

More in detail, we investigate issues raising when elastic (TCP-based) and real-time (UDP-based) applications coexists in infrastructure-based vehicular networks and propose a solution. In fact, whereas in computer networks classes we learn that the lack of congestion control in UDP-based flows is a potential harm toward TCP-based flows, in reality even the latter represent a source of problems, as persistent TCP-based flows are responsible for performance deterioration of concurrent UDP-based applications [5]. This problem is clearly made worse by the high mobility and network traffic variability characterizing vehicular networks. Indeed, cars traveling from the coverage area of an access point (AP) to another one generates continuous variations in the number and type of flows served by APs along the road.

To address this problem, we discuss here the use of smart APs along the road. The purpose of these upgraded APs is to control heterogeneous transmission flows and make them coexist efficiently. To this aim, each smart AP has to continuously snoop transiting packets and compute the maximum data rate at which each elastic application will be able to transfer their files without incurring in congestion losses. As we will discuss, this is not a hard task for APs as they are in the perfect place, traversed by all transiting flows, to do this. Then, this computed data rate is used to compute an appropriate advertised window that is included on-the-fly in transiting ACKs of TCP flows with the aim of having each TCP flows not exceeding the bandwidth factually available [6].

This solution is named Smart Access Point with Low Advertised Window (SAP-LAW) and its employment smoothens the network traffic. In particular, the available bandwidth is used efficiently while avoiding congestion losses and queuing delay at the AP. The latter is crucial to ensure low per-packet delay to real-time flows. Changes in the number and type of flows served by the considered AP are immediately detected and addressed. Furthermore, SAP-LAW does not require changes in the whole Internet, but just the deployment along the road of APs with SAP-LAW features.

Through a realistic case study, we demonstrate how SAP-LAW represents a valid solution to find the best tradeoff solution between the throughput achieved by elastic applications and the per-packet delay of real-time ones, even with the challenges of the vehicular networks. In particular, we consider different scenarios in terms of vehicles' speed and location and analyze their impact on the achieved performance.

The rest of the paper is organized as follows. In Section II we provide background information about the considered scenario. Section III outlines our solution. The experimental assessment is explained in Section IV and collected results are reported in Section V. Finally, conclusion are drawn in Section VI.

II. QUEUING DELAY

Nowadays a significant part of online traffic requires interactivity more than bandwidth. Consider, as a representative example, the very popular online games. Indeed, game messages (and corresponding network traffic) are generally very small, whereas it is crucial that the time spent between the generation of any game event and its delivery to all interested players remains below a certain time threshold [7]. The typical interactivity threshold for fast-paced interactive games (i.e. vehicle racing, first person shooter) is around 100 ms; however, this value can be increased up to seconds in case of slow paced games (e.g., strategic and role play games) [8], [9].

Since a single game event experiences different overall delivery delays from the source to all the diverse destinations, these values should be kept below the game interactivity threshold so as to not jeopardize the players' interactivity. In particular, the delivery delay is composed by several components: physical latency, queuing time on nodes along the path, and processing time. The queuing time is crucial both because of its impact and because it an unnecessary delay that could be removed from the system.

To make the scenario even worse, the APs along the road experience continuous variations in the number and type of flows that each of them has to serve. Therefore, the delivery delay could be not only high but also highly variable. In fact, players may be able to adapt their gaming action to the delay of the system (e.g., constantly anticipating the steering at each curve when playing car racing games) [9]. Yet, with highly variable delivery delays of game events, this strategy is not applicable.

As recently demonstrated by measurements on a real OC48 link, the capacity of the Internet is generally larger than the aggregate bandwidth utilized by transiting flows and the bottleneck of the connection is generally located at the edge of the path connecting the sender and the receiver [10]. Problems may hence arise at the last hop, which represents the bottleneck in terms of the available capacity for the connection. In fact, it might be the case when the AP receives packets at higher rates that it can forward to destination.

A vehicular network scenario is also featured by interference, errors, fading, and mobility, which may cause a significant amount of packet losses. Aiming at supporting elastic (TCP-based) traffic, the MAC protocol handles these packet losses through local retransmissions that hide error losses to the TCP and improve the reliability of the connection. On the other hand, retransmissions imply that subsequent packets have to wait in queue until the preceding one, or one of its retransmissions, finally reaches the receiver and the corresponding ACK is successfully sent back. In other words, retransmissions generate more queuing delays and hence delivery delays.

Finally, vehicular networks are generally composed by several connected vehicles sharing the same wireless medium; this clearly increases the congestion and queuing levels. Indeed, TCP connections continuously probe the channel for more and more bandwidth until buffers along the path are fully utilized and overflowed. With persistent TCP connections supporting elastic applications, buffers at the bottleneck becomes steadily fully utilized, thus queuing packets, slowing down their delivery time, and jeopardizing the interactivity of online games.

III. TCP AND SAP-LAW

The TCP sending rate is computed as the minimum between the advertised window and the current congestion window [6]. The receiver node determines the advertised window as the number of packets the receiver can handle; its value is then communicated back to the sender through a specific field in ACK packets. Instead, to define the congestion window, the sender takes into account the success of ongoing transmissions. In essence, upon any successful transmission demonstrated by a returning ACK, the congestion window is increased; whereas when one or more packets are lost, the congestion window is halved.

Therefore, the TCP sending rate (or sending window) steadily grows until the congestion window surpasses the advertised window, or until a packet is lost. In the former case, the actual sending rate remains constant and equal to the advertised window. Instead, in the latter case, the sending window is halved before restarting its growth.

A. Limiting the TCP's Advertised Window

The capacity of a channel represents an insurmountable limit to the TCP throughput. If the TCP transmission rate tries to surpass that limit, the sender simply generates packet queuing at the bottleneck that will eventually cause congestion losses and consequent halving of the sending rate. Therefore, TCP's sending rate should be kept high enough to efficiently utilize the available bandwidth but, at the same time, limited in its growth so as to not utilize buffers. This is a strategy that maximizes the throughput thanks to the absence of packet losses, while minimizing the per-packet delay thanks to the absence of queues.

An efficient upper bound for the TCP's sending rate has to be computed also considering other flows sharing the same bottleneck. Specifically, the aggregate bandwidth utilized by TCP flows on a given bottleneck should not exceed the total capacity of the bottleneck link diminished by the portion of the channel occupied by the concurrent real time traffic. Moreover, this aggregate bandwidth should be fairly shared by the TCP flows. Therefore, the maximum sending rate for each TCP flow at time t, namely *TCPubrate(t)*, is represented by:

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

where *UDPtraffic(t)* represents the amount of bandwidth consumed by UDP-based traffic at time *t*, *#TCPflows(t)* is the number of simultaneously present TCP flows, and *C* corresponds to the capacity of the bottleneck link.

Having determined the appropriate upper bound for a TCP flow's sending rate, our Smart Access Point with Low Advertised Window (SAP-LAW) modifies the advertised window included in TCP's ACKs. Since the actual sending window is computed as the minimum between the congestion window and the advertised window, the latter embodies a natural upper bound to the congestion window; even better, it is already implemented in all TCP versions.

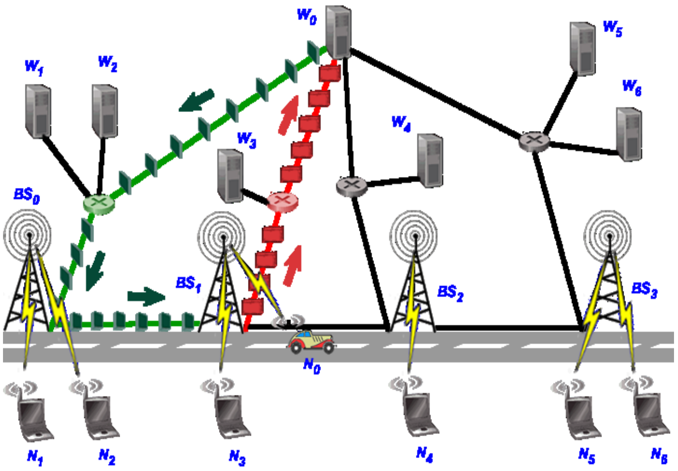

Fig. 1. Experimental scenario.

TABLE I. BACKGROUND FLOWS

From	To	Home Agent (AP)	Flow Type	Transport Protocol
W_1	N_1	AP_0	FTP	TCP New Reno
N_2	W_2	AP_0	Online gaming	UDP
W_2	N_2	AP_0	Online gaming	UDP
N_3	W_3	AP_1	Online gaming	UDP
W_3	N_3	AP_1	Online gaming	UDP
W_4	N_4	AP_2	Video streaming	UDP
W_5	N_5	AP_3	FTP	TCP New Reno
W_6	N_6	AP_3	Video streaming	UDP

The TCP advertised window is usually determined by the receiver; unfortunately, the receiver is not in the best position for the kind of computation and modification we need to perform. Indeed, to define the appropriate advertised window, a comprehensive knowledge about all flows that are transiting through the wireless link (i.e., the bottleneck) is needed. The AP represents the node able to implement our scheme since all flows have to pass through it. By spoofing the channel, the AP can also infer the number of active TCP connections and the aggregate amount of current UDP traffic; it can hence easily compute all the parameters needed in (1). Furthermore, all TCP's ACKs have to transit through the AP, which can hence modify on-the-fly the advertised window field with *TCPubrate(t)*.

Finally, it is worth mentioning that SAP-LAW does not require to have all APs in the network endowed with its mechanism. In fact, APs implementing SAP-LAW can coexist side by side with regular ones; the former bringing advantages to their served flows, without affecting the latter.

IV. EXPERIMENTAL ASSESMENT

We test SAP-LAW through the NS-2 simulator in an urban-like scenario. Along the curb APs provide wireless users with connectivity to the Internet. These wireless users can be static or mobile (i.e., vehicles). Wireless users connect to their closest AP to run different applications: FTP, video stream, and online gaming.

We consider a road segment comprising four APs (AP_0, AP_1, AP_2, AP_3) at 1 Km of distance from each other, seven wired nodes (W_0, W_1, …, W_7), and seven wireless ones (N_0, N_1, …, N_7). Among the wireless nodes, we monitor N_0 while it is driving through the transmission ranges of the various APs with a passenger playing an online game. For the sake of clarity, Fig. 1 shows the considered network topology: a block, or a group of blocks.

Wireless nodes continuously transmit and/or receive data through certain APs; the distance between these wireless nodes and their engaged APs is 100 m. The scenario also includes wireless nodes connected to the various APs so as to have a predetermined background traffic continuously utilizing these APs, thus allowing a clearer understanding of the outcomes. Specifically, the employed application flows are described in Table I.

To increase the realism of the considered multimedia applications, the video streaming corresponds to the actual movie Star Wars IV in high quality MPEG4 format. Online gaming traffic is inspired by real traces of the popular Counter Strike action game, and has i) a server-to-client flow characterized by an inter-departing time of game updates of 200 Bytes every 50 ms and ii) a client-to-server flow characterized by an inter-departing time of game events of 42 Bytes every 60 ms [11].

Since we are considering a vehicular scenario, we have modified the IEEE 802.11 module available for NS-2 to behave following IEEE 802.11p's specifications. This results in having about 750 m of transmission range [1]. MAC layer buffers on the APs are set equal to 100 packets as this is one common value in off-the-shelf APs.

To run their applications, wireless users (both static and mobile) connects to the closest AP; from there, data are transmitted through wired links to the server in the Internet that is providing that service, and *viceversa*. These wired links have a 100 Mbps capacity, whereas wireless ones have a variable capacity (depending on channel interferences) of about 19 Mbps, thus representing the bottleneck of the whole connection. All wired links directly connecting two APs have less than 1 ms of propagation delay (they are only 1000 m far from each other), whereas all other wired links have a propagation delay of 20 ms. The distance between the static wireless nodes and their engaged APs is 100 m. Continuously, all the APs serve background traffic from/to some static wireless node.

Buffers in the wired connections are set equal to 70 packets, which corresponds to the pipe size, i.e. the bandwidth-RTT product. The TCP's advertised window is initially set to a very high value, 550 packets to let the TCP's sending window grow as the capacity of the links permit, and remains constant when regular TCP New Reno and regular APs are employed, whereas SAP-LAW dynamically and continuously sets it by employing (1).

A mobile node, N_0, is traveling along the road passing by the coverage area of each of the APs with a speed of 14 m/s (about 50 Km/h and 32 Mph). When N_0 moves into the coverage area of a new AP, it connects with the new antenna and continues its operations through Mobile IP's packet redirection [12]. The mobile node N_0 runs different applications in different sets of simulations, specifically: a TCP-based FTP or a UDP-based online game. In the former case the data flow is mostly unidirectional, from a server in the Internet to N_0 (plus ACKs on the returning path), whereas in the latter case the data flow is bidirectional, game events go from N_0 to a game server in the Internet and game updates go from the server to N_0.

In our simulation campaign, we have considered two main scenarios: i) an **urban** scenario with vehicle N_0 traveling at 14 m/s (about 50 Km/h and 32 Mph) and ii) a **highway** scenario with vehicle N_0 traveling at 33 m/s (about 119 Km/h and 74 Mph). We have compared the regular TCP scenario with the SAP-LAW scenario. For the latter case, we have considered three reasonable values for parameter C in (1): 18, 19, and 20 (Mbps). For each configuration, results have been gathered averaging the outcome of 50 simulation runs.

V. RESULTS

First, we have verified whether some of the compared solutions could have a negative effect on the TCP's throughput. We have considered the case in which N_0 is downloading a file. As can be seen in Fig. 2 the main difference regards the speed at which N_0 moves as in the highway scenario the experienced goodput may be 20% lower than in the urban scenario. However, if focusing on the urban scenario, the considered solutions achieves very similar goodput values and the same can be said when focusing only on the highway scenario.

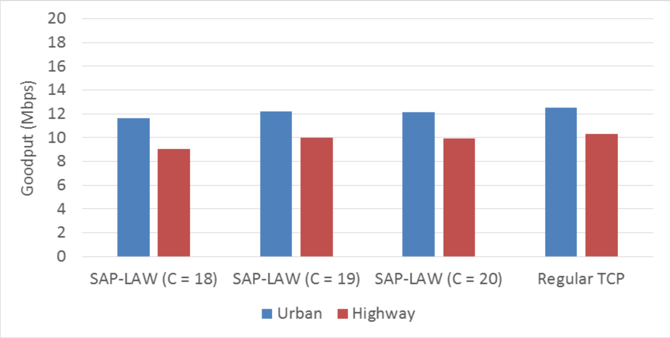

Fig. 2. Average goodput received by the mobile node N_0.

We have considered even the goodput achieved by a static node as N_1 and the outcome is reported in Fig. 3. As can be seen, all solutions brings to very similar goodput values, even regardless of the speed of node N_0. This makes sense as node N_1 is static. For the same reason it is no surprise that the average goodput is higher than in Fig. 2. However, what is most important for us is that no SAP-LAW version harmed the resulting goodput.

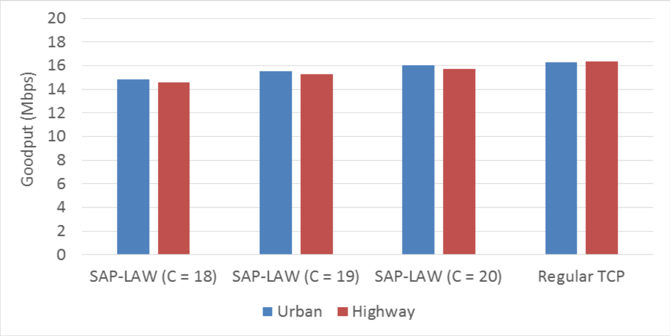

Fig. 3. Average goodput received by the static node N_1.

We have then focused our attention on the queuing delay generated by the various configurations and, in particular, on the jitter. In the experiments, we considered only SAP-LAW with C set equal to 19. We have considered a series of scenarios where the passenger of N_0 is engaged in a Counter Strike online game session. The urban scenario and the highway one are reported in Fig. 4 and Fig. 5, respectively. The charts present average, variance and maximum values of the jitter suffered by N_0's online game flow. The values show that, even if the average jitter of both cases are low, the difference in terms of variance and of maximum values is significant. This is crucial as even one or a few game events experiencing high delivery delay may disrupt the interactivity of the game and annoy the player.

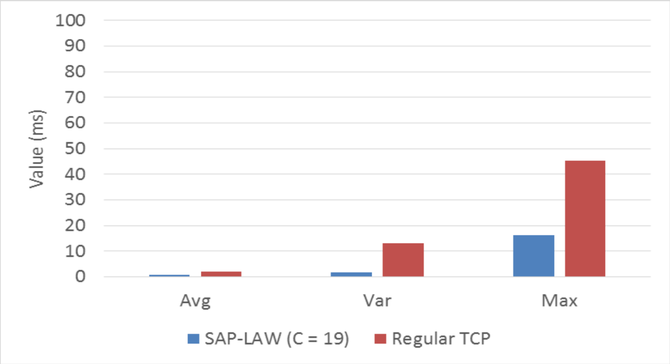

Fig. 4. Jitter of the online game flow received by N_0; urban scenario.

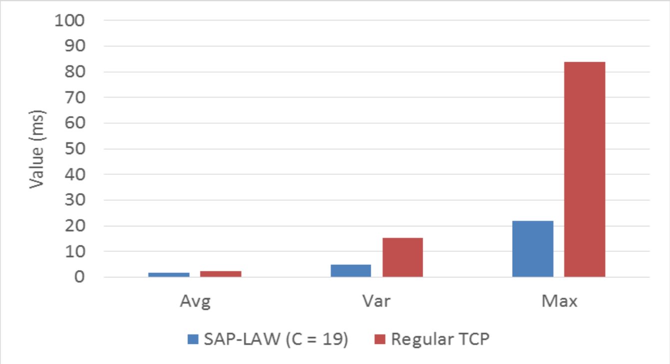

Fig. 5. Jitter of the online game flow received by N_0; highway scenario.

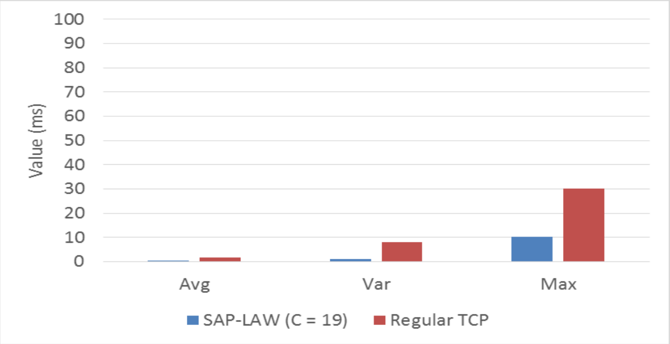

Fig. 6. Jitter of the online game flow received by N_2; urban scenario.

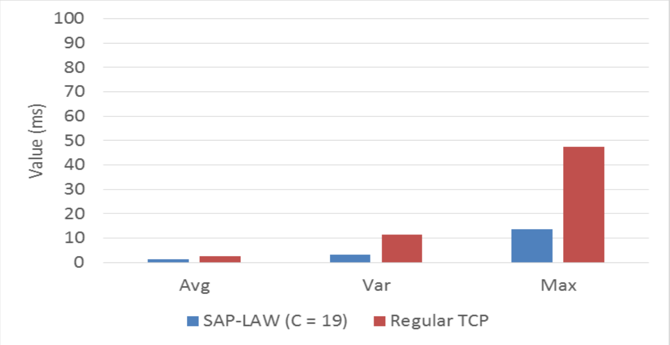

Fig. 7. Jitter of the online game flow received by N_2; highway scenario.

The jitter experienced by static node N_2 (also engaged in an online game) is reported in Fig. 6 and Fig. 7, respectively for the urban scenario and for the highway one. The jitter values are lower than those experienced by N_0 due to the fact that N_0 is traveling whereas node N_2 is static. Beside this, we can still notice the benefits brought by the employment of SAP-LAW.

VI. CONCLUSION

Vehicular networks represent the next frontier in wireless communications. Through APs along the road, car passengers will be able to access the Internet and all their favorite online applications, even games. In this context, we have evaluated a solution, named SAP-LAW, based on the deployment of smart APs able to exploit regular features of transport protocols in order to improve the coexistence among elastic and real-time applications. Preliminary results have shown the effectiveness of the proposed solution. We now intend to enlarge our work by analyzing the performance in more complex scenarios and against different algorithms [13]. We would also like to adapt our solution to support DTN [14]-[17] and alert message propagation [18], [19].

ACKNOWLEDGMENT

Our deep gratitude goes to Salvatore Frizzoli for his technical help with NS-2. This work has been partially funded by the Università degli Studi di Padova, through the projects PRAT CPDA137314 and PRAT CPDA151221.

REFERENCES

[1] Dedicated Short Range Communications (DSRC) Home. [Online]. Available: http://www.leearmstrong.com/dsrc/dsrchomeset.htm

[2] M. Di Felice, L. Bedogni, L. Bononi, "Group Communication on Highways: An Evaluation Study of Geocast Protocols and Applications", Elsevier Ad Hoc Networks Journal 11(3), 2013.

[3] M. Furini, "Mobile Games: What to Expect in the Near Future", in Proc. of GAMEON 2007, Bologna, Italy, 2007.

[4] 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.

[5] C. E. Palazzi, S. Ferretti, M. Roccetti, G. Pau, M. Gerla, "Wireless Home Entertainment Center: Reducing Last Hop Delays for Real-time Applications", in Proc. of ACM SIGCHI International Conference on Advances in Computer Entertainment Technology (ACE 2006), Hollywood, CA, USA, 2006.

[6] V. Jacobson, "Congestion Avoidance and Control," in Proc. of ACM SIGCOMM' 88, Stanford, CA, USA, 1988.

[7] C. E. Palazzi, S. Ferretti, S. Cacciaguerra, M. Roccetti, "Interactivity-Loss Avoidance in Event Delivery Synchronization for Mirrored Game Architectures," IEEE Transactions on Multimedia 8(4), 2006.

[8] G. Armitage, "An Experimental Estimation of Latency Sensitivity in Multiplayer Quake 3," in Proc. of ICON, Sydney, Australia, 2003.

[9] L. Pantel, L. C. Wolf, "On the Impact of Delay on Real-Time Multiplayer Games," in Proc of NOSSDAV'02, Miami, FL, USA, 2002.

[10] 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, 2005.

[11] J. Färber, "Traffic Modelling for Fast Action Network Games," Multimedia Tools and Applications, 23(1), 2004.

[12] C. E. Palazzi, B. Chin, P. Ray, G. Pau, M. Gerla, M. Roccetti, "High Mobility in a Realistic Wireless Environment: a Mobile IP Handoff Model for NS-2," in Proc. of IEEE TridentCom 2007, Orlando, FL, USA, 2007.

[13] A. Bujari, M. Massaro, Claudio Enrico Palazzi, "Vegas over Access Point: Making Room for Thin Client Game Systems in a Wireless Home", IEEE Transactions on Circuits and Systems for Video Technology, 25(12), 2015.

[14] S. M. Tornell, C. M. T. Calafate, J.-C. Cano, P. Manzoni, "DTN Protocols for Vehicular Networks: An Application Oriented Overview", IEEE Communications Surveys and Tutorials 17(2), 2015.

[15] D. Maggiorini, C. Quadri, L. A. Ripamonti, "Opportunistic Mobile Games Using Public Transportation Systems: A Deployability Study", Multimedia Systems Journal 20(5), 2014.

[16] S. Gaito, D. Maggiorini, C. Quadri, G. P. Rossi, "On the Impact of a Road-Side Infrastructure for a DTN Deployed on a Public Transportation System", in Proc. of Networking, Prague, Czech, 2012.

[17] C. E. Palazzi, A. Bujari, S. Bonetta, G. Marfia, M. Roccetti, A. Amoroso, "MDTN: Mobile Delay/Disruption Tolerant Network", in Proc. of 20th IEEE International Conference on Computer Communications and Networks (ICCCN 2011), Maui, HI, USA, 2011.

[18] G. Marfia, M. Roccetti, A. Amoroso, G. Pau, "Safe Driving in LA: Report from the Greatest Intervehicular Accident Detection Test Ever", IEEE Trans. Vehicular Technology 62(2), 2013.

[19] A. Amoroso, G. Marfia, M. Roccetti, "Going Realistic and Optimal: A Distributed Multi-Hop Broadcast Algorithm for Vehicular Safety", Elsevier Computer Networks 55(10), 2011.