A latency analysis for M2M and OG-like traffic patterns in different HSPA core network configurations

Milica V. Popović, *Member, IEEE,* Dejan D. Drajić, *Member, IEEE,*
and Srđan M. Krčo, *Senior Member, IEEE*

Abstract — **In this paper we present an analysis intended to reveal possible impacts of core network features on latency for modelled M2M and Online Gaming traffic. Simulations were performed in a live 3G/HSPA network. Test traffic simulating multiplayer real-time games and M2M applications was generated on 10 mobile phones in parallel, sending data to a remote server. APNs with different combinations of hardware and features (proxy server, different GGSNs and firewalls, usage of Service Awareness feature) were chosen. The traffic was recorded on the Gn interface in the mobile core. The goal of experiments was to evaluate any eventually significant variation of average recorded RTTs in the core part of mobile network that would clearly indicate either the impact of used APN on delay for a specific traffic pattern, or selectivity of the APN towards different traffic patterns.**

Keywords — **Core network delay, HSPA, Latency, M2M, Online Gaming, RTT.**

I. Introduction

MACHINE type traffic and online gaming are types of applications expected to show substantial growth in the upcoming years [1]. They are increasingly using mobile networks to communicate with other devices, central servers and players. In 3GPP [2], machine type traffic is a part of the Machine Type Communication (MTC) framework which describes the exchange of data between two machines, while in ETSI [3] it is referred to as Machine to Machine (M2M). The most challenging problem is the co-habitation of M2M traffic with conventional user traffic coupled with the potential of a rapid increase in the number of devices connected to cellular infrastructure. Together with “traditional traffic” (voice, SMS, web browsing, data download), M2M traffic is introducing new requirements on the underlying mobile networks. The M2M promises huge market growth with expected 50 billion connected M2M devices by 2020 [4]. Support for such a massive number of M2M devices has deep implications on the end-to-end network architecture, thus it is important to understand the corresponding traffic models and their influence on mobile network. An important characteristic of the MTC is the variety of possible communication patterns, with heterogeneous requirements and features [2]. M2M application space is vast and includes security, health monitoring, remote management and control, gaming, industrial wireless automation, and many more. In the Online Games (OG) domain, low latency is particularly critical for so called massive multi-user online first-person shooter type of games.

The work presented herein was done in the context of the ICT FP7 LOLA project, European Academia/Industry collaborative project [5]. The goal of the LOLA project was to provide significant technological advances in terms of minimizing end-to-end latency in wireless systems. LOLA targeted low-latency applications found in M2M communications and highly-interactive services such as gaming or remote control.

Within the LOLA project, comprehensive testing was conducted in order to reveal network response to OG and M2M types of traffic and possible bottlenecks leading to increased latency. For a chosen OG application, measurements of real application traffic were done [6], and traffic models derived by fitting statistical distributions to real traffic data [7]. For M2M traffic, models [7] were derived based on defined application scenarios [8], wherein some applications require low latency [9]. For the purposes of testing, an Android application for traffic generation was developed, further referred to as TG-App, with the goal of implementing the modeled parameters, i.e. different distributions of packet sizes and inter-arrival times. Traffic simulations were done in a live HSPA network, and latency assessed through the analysis of Round Trip Time (RTT).

Previously conducted measurements [10] - [12], using the same traffic parameters as in testing presented herein, had revealed a significant impact these types of traffic might have on the HSPA network. The radio access

Paper received March 15, 2014; revised July 17, 2014; accepted July 25, 2014. Date of publication November 15, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Zorica Nikolić.

This paper is a revised and expanded version of the paper presented at the 21th Telecommunications Forum TELFOR 2013.

This paper describes work undertaken in the context of the LOLA project - Achieving LOw-LAtency in Wireless Communications (www.ict-lola.eu). The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement n° 248993.

Milica Popović, Telekom Srbija a.d. Takovska 2, 11000 Belgrade, Serbia, (email: milicapop@telekom.rs).

Dejan Drajić, Ericsson d.o.o, Milentija Popovića 5a/v, 11070 Belgrade, Serbia, (e-mail: dejan.d.drajic@gmail.com).

Srđan Krčo, Ericsson d.o.o, Milentija Popovića 5a/v, 11070 Belgrade, Serbia, (e-mail: srdjan.krco@ericsson.com).

network proved to be generating the largest part of the overall latency, as expected, but had also exhibited a huge degradation of accessibility key performance indicator during tests on a live, loaded Node B, for both circuit-switched (voice) and packet-switched services. Further research showed that the Node B upgrade led to a better network performance and decreased latency. The intention of additional test cases, presented in this paper, was to analyse the influence of core network features on latency.

Depending on the Access Point Name (APN) chosen, packet data travel along different paths in the mobile core network and different features are applied. Different Gateway GPRS Support Nodes (GGSN) and firewalls may be used, a proxy server may be used or not, Service Awareness feature applied or not, and all of these may have an impact on latency for specific traffic patterns. Herein, several APNs with different combinations of the stated hardware/features are chosen to detect any significant variation of latency generated in the core network. In the mobile network proxy server is normally used to speed up loading of internet pages for users and content filtering, but for M2M and OG types of traffic it might produce some additional delay. Service Awareness feature of the GGSN is used for billing purposes and for QoS control, and it might also yield some delay, as the GGSN may drop some out-of-order packets and perform additional retransmissions with the RNC. Two types of GGSN were used, from the same manufacturer, using two types of firewalls, a software firewall and a hardware firewall. It is important to note that APNs chosen have other different settings, as each is designed for a different type of user access and different purpose.

The paper is organized as follows. Section 2 provides a brief explanation of measurement setup and traffic generation application. In Section 3 a description of test cases is given, while in Section 4 the measurement results are presented and analyzed. Section 5 provides a summary with concluding remarks.

II. Measurement Setup

Measurement setup was topologically the same as in our previous work [10] - [12], and is shown in Fig. 1. Ten test phones with Android 2.2 operating system were used, all of the same make and model. Tests were performed in Telekom Srbija live HSPA network. The Node B under test was located in the highly urban area. Client TG-App was installed on test phones sending test traffic to the remote server, located at an approximate distance of 80 km from the test site. In order to measure the delay in the network, trace recording was performed on phones and in the mobile core network, at the Gn interface connecting the Serving GPRS Support Node (SGSN) and the GGSN.

The Node B, serving both test traffic and regular users' traffic, had the following characteristics:

- 192/192 Channel Elements Uplink/Downlink activated
- Enhanced Uplink (eUL) and High Speed Downlink Packet Access (HSDPA) activated
- two carriers, HSPA traffic going to both carriers
- license for 16 simultaneous HSDPA users in each cell (i.e. 32 in a sector).

All phones were static, served by two sectors of the subject Node B. The TG-App was using the TCP protocol and the delay was measured as the two-way delay, RTT, for each packet sent. For this analysis, traffic traces captured at the Gn interface using the Wireshark [13] application were of interest. The RTT measured at this interface, further denoted as the Gn RTT, was the time interval between the moment a client's packet passed through the interface on its way to the server and the moment its corresponding acknowledgement (ACK) from the server passed the interface in the opposite direction (towards the phone). The exchange of messages between the client and the server was captured on phones as well, using the Shark for root application [14], for verification reasons.

Traffic capturing at the Gn interface was performed by filtering out only the traffic originated from the Radio Network Controller (RNC) of interest. Even with filtering, the amount of traffic crossing this interface was huge, thus only 3-5 minutes of traces were taken. The traces were captured on two physical branches of the Gn interface using two laptops that were not synchronized. Files recorded were then chronologically merged, yielding two main traces for two branches. These traces were analyzed separately since it was determined that the traffic from individual phones did not always go over a single branch, and with no time synchronization between two capturing devices further chronological merge of two traces would not yield valid results. The traces were further filtered by IP addresses of individual phones and delay statistics was derived. In some test cases, for some of the phones it was not possible to obtain any valid delay statistics, as the whole traffic from the phone went over one branch, whereas the whole corresponding traffic from the server to that phone (acknowledgements) went over the other branch.

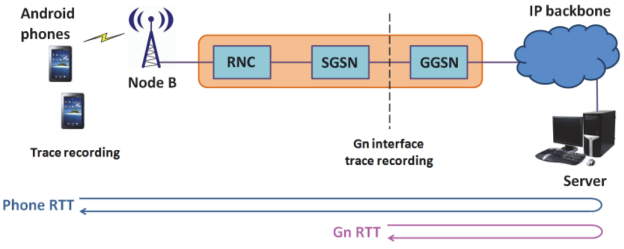

Fig. 1. Measurement setup.

The following online games were simulated:

- Open Arena (OA)
- Team Fortress (TF).

Both simulated games are first person shooter (FPS) games i.e. the genre of video games that features a first-person perspective to the player.

The following M2M applications were simulated:

- Bicycle Race (BR)
- Auto Pilot (AP)
- Team Tracking (TT).

TABLE 1: TEST TRAFFIC CHARACTERISTICS.

Phone Number	Application/ Direction/ Protocol	Packet sizes distribution/ Time distribution	Average packet size [bytes]	Average time between packets [s]	Max throughput [kbps]	Min throughput [kbps]
1	OA, TCP	Gauss (42.2;4.6)B, Uniform (69,103)ms	40	0.086	6.68	1.82
2	TF, UL, TCP	Gauss (76.52;13.9)B, Uniform (31,42)ms	75	0.0365	33.27	5.21
3	OA,DL, TCP	Gauss (0.172;0.05)kB, Uniform (41,47)ms	170	0.044	94.32	0.17
4	TF, DL, TCP	Gauss (0.241;0.06)kB, Uniform (39,46)ms	240	0.0425	117.39	0.17
5	BR, UL, TCP	Constant (1)kB,Uniform (100,500)ms	1024	0.3	80.00	16.00
6	BR, DL, TCP	Constant (1kB),Uniform (100,500)ms	1024	0.3	80.00	16.00
7	AP, UL, TCP	Constant(1)kB, Uniform(25;100)ms	1024	0.0625	320.00	80.00
8	AP, DL, TCP	Constant(1)kB, Constant(1000)ms	1024	1	8.01	7.99
9, 10	TT(GPS Keep Alive), UL, TCP	Constant(0.5)kB, Uniform(1;25)s	512	13	4.00	0.16

Bicycle Race is a machine-aided gaming type of application. The opponents are at different locations and agree about the corresponding length (or time) of the race. To calculate and share the equivalent position of each participant, measurements are taken by sensors (GPS, temperature, humidity, speed, terrain configuration) and are exchanged between the opponents. Auto pilot scenario includes vehicle collision detection and avoidance (especially on highways) and the urgent actions taken in case of an accident. Team Tracking (TT) is a public safety application used to monitor the position of several nodes (team members) in a given environment (e.g. building, stadium) for situation awareness and consequent action scheduling.

Table 1 presents the parameters configured for TG-App on every particular phone for online gaming and M2M simulations [8], [10], [11].

III. DESCRIPTION OF TEST CASES

In order to evaluate the impact of core network hardware and settings on latency for defined traffic patterns, five test cases have been performed. Each test case used a different combination of core network features/configuration, i.e. a different APN. Test cases are described in Table 2.

TABLE 2: CONFIGURATIONS IN TEST CASES.

Test Case	APN name	Using proxy	Service Aware	GGSN	Firewall
1	GPRSw	Yes	Yes	1	CheckPoint (SW FW)
2	GPRSi	No	Yes	1	CheckPoint
3	TEST	No	No	1	CheckPoint
4	3G	No	No	2	Juniper (HW FW)
5	Prepaid	No	Yes	2	Juniper

First, the usage of a proxy server was under test, expected to yield higher latency. Service Awareness feature was expected to yield higher latency, since the GGSN performs ordering of packets in order to comprehend the service and perform right billing, that may result in drops of unordered packets, retransmissions etc. Further, two different GGSNs were used, each using a different firewall - CheckPoint (software firewall) and Juniper (hardware firewall). For each APN used in tests there were sets of parameters configured depending on the purpose of a specific APN:

- Case 1 APN GPRSw was the public APN, used for accessing basic internet services from the phone, going out to the web over web/wap proxy, because of the optimization of speed for older telephone models, billing of certain contents from the portal, online marketing, etc. It was using Service Awareness feature at GGSN.
- Case 2 APN GPRSi was also public, made for the same purposes as Case 1 APN, with the only difference in not using the proxy server.
- Case 3 APN TEST served for test purposes, didn’t use Service Awareness, nor a proxy server.
- Case 4 APN 3G was designed for user access with notebook and tablet computers using a modem, for higher throughputs. The traffic is processed through GGSN2, not using Service Awareness
- Case 5 APN PREPAID was designed for internet access with higher speeds, and was used for prepaid SIM cards. It was using the path through GGSN2 and Juniper firewall, as 3G, but used Service Awareness, unlike 3G.

The comparison of delay statistics for the same traffic pattern and different APNs was expected to show any impact of used features/hardware on latency. Further, the comparison of delay statistics for the same APN and different traffic patterns might reveal any selectivity of the core network in terms of latency for different kinds of traffic.

IV. Result Analysis

The impact of described features on latency was evaluated through the analysis of RTT as seen from the Gn interface. Gn RTT comprises a double delay in the mobile core network – passage through the GGSN, proxy if used, firewall – then a double delay in the public IP network, a double delay in the server network, and processing delay in the server. The public IP network may be considered random and non-selective to certain types of traffic, as well as the server network, thus any significant differences emerging from Gn RTTs comparison would indicate an additional delay generated in the mobile core network. Test cases were performed successively, cases 1 and 2 on the same day, cases 3 and 4 on the next day, and Case 5 on the day after. Thus, the conditions in the core network, the backbone and the server network might have differed to some extent.

The detailed analysis would involve trace recording at the firewall for each test case with subtraction of two-way delays at the Gn interface and the firewall for each phone, as well as trace recording at the server. This is out of the scope of this work, but was performed in other test cases [15], proving that the backbone and the server network do not show clear selectivity toward different traffic patterns.

It is important to have in mind that the Gn trace comprised only about 5 minutes of transmission, due to the large throughput at the interface, even with filtering, and the limitations of capturing methods used. Taking into account retransmissions that might occur between the GGSN and the RNC in the user plane (direct tunnel), buffering at both sides, as well as ordering of packets, possible drops of out-of-order packets etc, occurring at the GGSN due to Service Aware functionalities, these traces were not so straightforward for analysis.

Further, TCP implementation in TG-App is such that for every generated application packet, a sequence of TCP packets is exchanged between the phone and the server [9]. The 1st TCP packet is always a 70 B long message from the client. The 2nd client's message contains the payload of nominal size (according to the defined distribution), enlarged for the TCP header. Both messages are acknowledged by the server, and at the end of sequence, the client acknowledges the receipt of server's last message, as shown in Fig. 2. This sequence is repeated for every application packet. Thus, two RTTs are recorded by Wireshark for each TG-App-generated packet. The server introduces an average 200.5 ms delay (with 4 ms standard deviation) [15] for the first TCP packet, which is verified by trace recording at the server. For 2nd TCP packets, delay at the server was less than 1 ms for all traffic patterns.

Gn traces were filtered out by IP addresses of every phone in 5 test cases, giving 50 separate traces, some of which were not valid, as described in the Measurement setup section. The valid RTT statistics is given in tables 3 and 4, for the 1st and the 2nd TCP packets in sequence respectively.

Average Gn RTTs are graphically represented in Fig. 3, for the 1st (a) and the 2nd (b) TCP packets. It is clear that Case 4 and Case 1 APNs led to the worst results concerning latency, while the best average results were achieved using Case 3 and Case 5 APNs, that were using different GGSNs and firewalls, without a proxy.

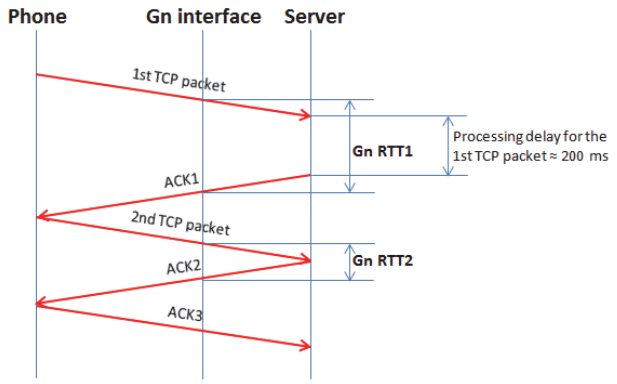

Fig. 2. The sequence of TCP packets exchanged for one application packet.

Table 3: Average RTT for the 1st packet in sequence.

Name	Average Gn RTT for the first packet in sequence [s]				
	GPRSw	GPRSi	TEST	3G	Prepaid
1	0.2349	0.2280	0.2164	-	0.2156
2	0.2504	0.2250	0.2225	-	0.2151
3	0.2346	0.2407	-	-	-
4	0.2326	0.2232	0.2152	0.2425	0.2319
5	0.2353	0.2224	0.2159	-	-
6	0.2338	0.2274	0.2173	0.2577	0.2164
7	0.2336	0.2252	0.2161	0.2561	-
8	0.2354	0.2365	0.2066	0.2430	0.2160
9	0.2338	0.2234	0.2157	-	-
10	0.2327	0.2429	0.2268	0.2470	-

Table 4: Average RTT for the 2nd packet in sequence (main payload).

Name	Average Gn RTT for the second packet in sequence (main payload) [s]				
	GPRSw	GPRSi	TEST	3G	Prepaid
1	0.0326	0.0225	0.0165	-	0.0149
2	0.0536	0.0190	0.0155	-	0.0147
3	0.0336	0.0184	-	-	-
4	0.0320	0.0207	0.0161	0.0394	0.0305
5	0.0354	0.0270	0.0175	-	-
6	0.0356	0.0237	0.0162	0.1385	0.0159
7	0.0358	0.0229	0.0168	0.0562	-
8	0.0341	0.0313	0.0171	0.0818	0.0148
9	0.0349	0.0150	0.0150	-	-
10	0.0355	0.0125	0.0163	0.0403	-

Considering the group of APNs using the same GGSN 1 and software firewall, and differing in the usage of proxy and Service Awareness feature, it seems that the usage of proxy introduced an additional delay, as well as the Service Awareness feature. Looking at the APNs using

GGSN 2 and hardware firewall, it seems that other particular APN settings had a more pronounced impact. Case 4 APN, designed for computer access and achieving high throughputs, should not be used with these types of traffic, while Case 5 APN, designed for prepaid access, gave rather good results.

The average values in a small statistical sample might be misleading, so the time variations of Gn RTTs for individual phones are observed. In order to make comparisons on per phone or per APN basis, all valid traces out of 50 are processed through Wireshark to obtain average values in 1 second intervals. Different traces are then aligned by starting points, i.e. normalized to the same time axes, and displayed on the same graph.

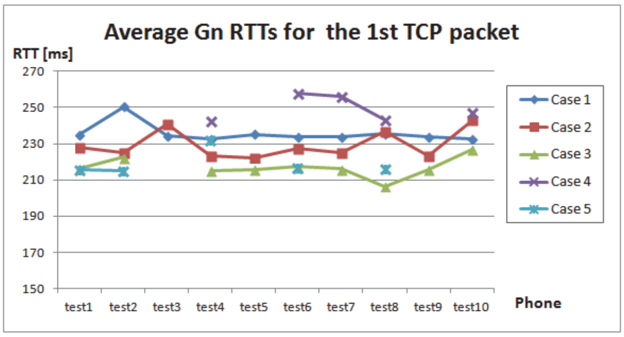

Fig. 3(a). Average Gn RTTs for the 1st TCP packet.

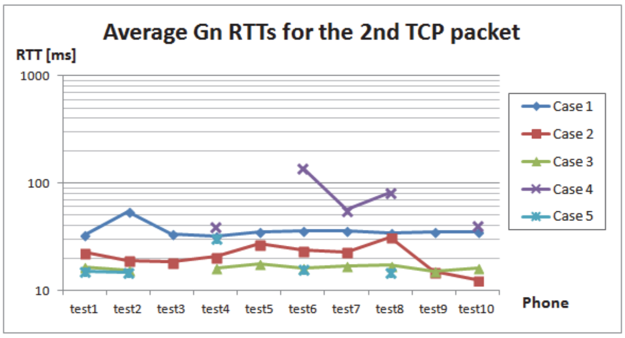

Fig. 3(b). Average Gn RTTs for the 2nd TCP packet.

The delay caused by the usage of proxy may be observed looking at phone traces grouped by Case 1 and Case 2 APNs in time. For clarity reasons, Fig. 4 shows only the average RTT values for 2nd TCP packets, obtained by averaging RTT values of all phones in 1s periods. Since the only difference between the two APNs is the usage of a proxy server, the obvious offset between the averages for two test cases confirms that the proxy server introduced an additional delay of around 10 ms.

Here, high results for all phones at some time instances indicate some problem in the core, backbone or server network, as the alignment of phone traces implies up to 25s time offset (the highest wait time between the generation of two packets). No obvious selectivity of a certain APN towards specific traffic patterns is observed.

It is also interesting to observe the same alignment for Case 4, having the worst results, displayed in Fig. 5. The average values on 1s intervals are rather scattered for all phones with valid results, confirming that there is no stability of the delay and that high values are rather frequent.

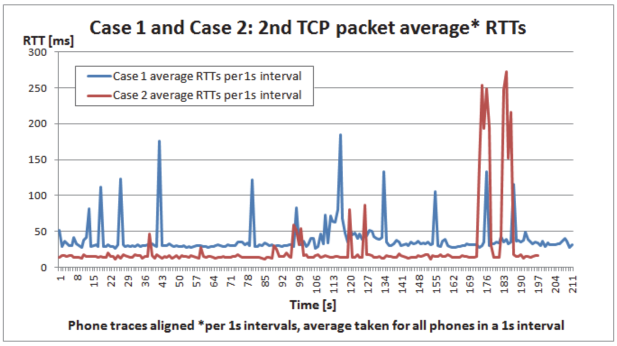

Fig. 4. 2nd TCP packet average RTTs in Case 1 and in Case 2.

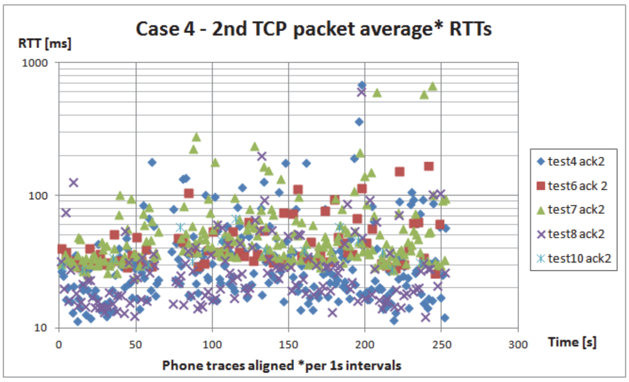

Fig. 5. 2nd TCP packet average RTTs in Case 4.

If we align the traces per phones, aligning the test periods of five test cases per 1s intervals, the impact of specific core network features for a particular traffic pattern may be observed. One should have in mind that such grouping (different APNs) on a single graph means that traces from completely different time intervals, even days, are aligned. Figures 6-8 show these variations for phones test4, test 6, and test8, respectively. The figures confirm that Case 4 APN had the most unstable delays for both types of traffic, M2M and OG.

In Fig. 6 it is visible that the Case 5 APN, generally with the best results for all phones, gave rather stable and high average RTTs for phone test4, for both 1st and 2nd TCP packets, being on the order of Case 1 average values. Looking at the traffic pattern, this phone had the largest variation of throughput, so this might suggest that the Case 5 APN is not suitable for such patterns. Looking at results for phones test6 and test8, in Fig. 7 and Fig. 8, Case 5 gave stable, low RTTs.

From Fig. 7 and Fig. 8, we observe a clear differentiation of Case 1 RTTs, being much higher than the average values for other APNs, except Case 4 APN. It is similar for phone test4, but the results for Case 1 in Fig. 6 are masked with Case 5 results. The comparison of Case 1 and Case 2 results in time for a single phone again confirms that the usage of proxy introduces an additional delay.

Further, Case 2 APN seems to have large average values for all patterns due to instability of the delay, compared to Case 3, which might imply that the Service Awareness feature in case of GGSN1 and specific settings for these APNs generated an additional delay. Moreover, it seems that this instability is manifested more for 1st TCP packets than for 2nd ones, suggesting that perhaps some buffering of small packets might be the cause. The additional delay introduced by this feature is not so pronounced as the one induced by proxy, as the average values suggest.

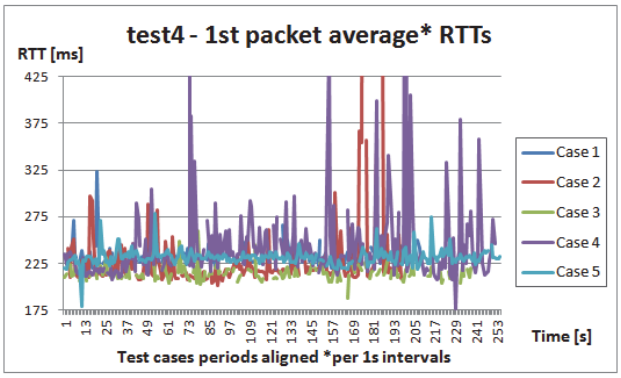

Fig. 6(a). All APNs, test4 – 1st TCP packets.

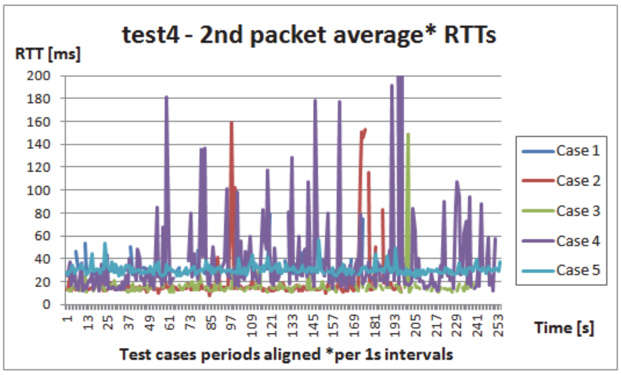

Fig. 6(b). All APNs, test4 – 2nd TCP packets.

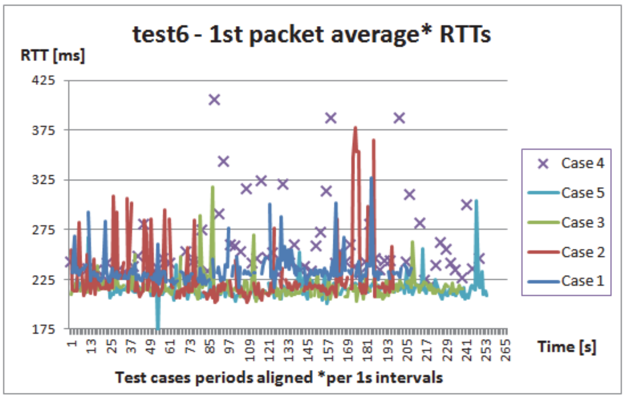

Fig. 7. All APNs, test6 – 1st TCP packets.

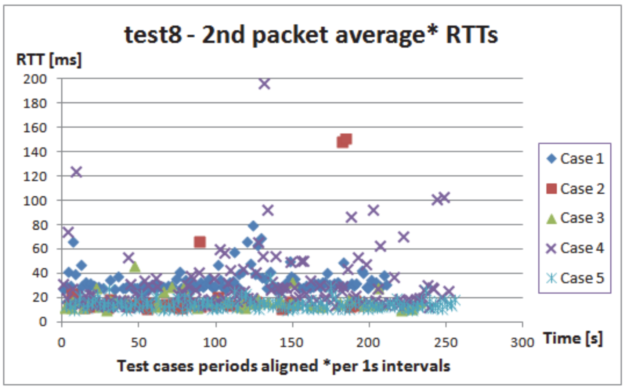

Fig. 8. All APNs, test8 – 2nd TCP packets.

V. Conclusion

In this paper, we presented measurement results showing the impact of core network features on delay for M2M and OG traffic patterns in a live 3G/HSPA network. These effects were observed through the usage of different APNs, using different combinations of core network hardware and features (two GGSNs, two firewalls, proxy server, Service Awareness feature of the GGSN).

Proxy server introduced additional delay in the core for these types of traffic. Service Awareness feature, on the other hand, did not show clear impact on delay.

While in case of GGSN1 it did introduce some instability of delay, in case of GGSN2 the results depended more on other specific APN settings. Further, APN designed for achieving high throughputs with tablets and notebooks showed the worst results concerning latency. Some selectivity towards specific traffic patterns was observed, as the APN with generally the best results showed a stable, much higher delay for the pattern with the highest variations of throughput, of almost three orders of magnitude. Presented results suggest that the APNs for M2M and Online Gaming access should be carefully designed, depending on the application field and specific latency requirements. Although the access network induces the biggest part of delay, the results indicate that the contribution of the core network should not be neglected.

References

[1] Van Der Meer J, Ericsson, Growth Plan 2011-2015. *EricssonVision50B Communicating Devices*. June 2011.

[2] 3GPP TS 22.368 v10.0.0, "Service requirements for machine-type communications (MTC)," March 2010. Avalable at: http://www.3gpp.org/ftp/Specs/archive/22_series/22.368/

[3] ETSI TS 102 689, "Machine-to-Machine communications (M2M); M2M service requirements," Aug 2010.

[4] Orrevad, A., M2M Traffic Characteristics (When machines participate in communication), in Information and Communication Technology2009, KTH Sweden: Stockholm. p. 56.

[5] ICT FP7 LOLA project, http://www.ict-lola.eu/

[6] LOLA Consortium, Deliverable 3.3 Summary of the Traffic measurements, 2010.

[7] LOLA Consortium, Deliverable D3.5 Traffic Models for M2M and Online Gaming Network Traffic, 2012.

[8] LOLA Consortium, Deliverable 2.1 Target application scenario, 2010.

[9] N. Nikaein, S. Krco, "Latency for Real-Time Machine-to-Machine Communication in LTE-Based System Architecture," *European Wireless* 2011, Austria.

[10] D. Drajic, S. Krco, I. Tomic, P. Svoboda, M. Popovic, N. Nikaein and N. Zeljkovic "Traffic generation application for simulating online games and M2M applications via wireless networks," WONS 2012, Italy.

[11] D. Drajic, M. Popovic, N. Nikaein, S. Krco, P. Svoboda, I. Tomic, and N. Zeljkovic "Impact of online games and M2M applications traffic on performance of HSPA radio access networks," ESIOT 2012, Italy.

[12] M. Popovic, D. Drajic, S. Krco "Evaluation of the UTRAN (HSPA) performance in different configurations in the presence of M2M and Online Gaming traffic," to be published in *Transactions on Emerging Telecommunications Technologies*, 2013.

[13] http://www.wireshark.org.

[14] http://www.androidzoom.com/android_applications/tools/shark-for-root_imrr.html.

[15] [15] LOLA Consortium, Deliverable 5.3 Validation results for Testbed 1, 2013

[16] M. Popović, D. Drajić, S. Krčo, "The impact of HSPA core network features on latency for M2M and OG-like traffic patterns," *TELFOR 2013*. Beograd, 26-28 nov. 2013, pp. 291 – 294.