Latency analysis for M2M and Online Gaming traffic in an HSPA network

Milica Popović[1], *Member, IEEE,* Dejan Drajić[2], *Member, IEEE,* Philipp Svoboda[3], *Member, IEEE* ,Navid Nikaein[4], *Member, IEEE*, Srđan Krčo[2], *Senior Member, IEEE,* Markus Laner[3],*Member, IEEE*

[#1]*Telekom Srbija a.d.*Takovska 2, Belgrade, Serbia
milicapop@telekom.rs
[#2]*Ericsson d.o.o., Milentija Popovica 5a, Belgrade, Serbia*
dejan.d.drajic@gmail.com, srdjan.krco@ericsson.com
[*3]*Institute of telecommunications, Vienna University of Technology Gusshausstrasse 25/38A, 1040Wien, Austria*
psvoboda@nt.tuwien.ac.at, mlaner@nt.tuwien.ac.at
[#4] *Mobile Communication Department, Eurecom, 06904, Sophia Antipolis, France*
navid.nikaein@eurecom.fr

Abstract. This paper analyzes latency in a High-Speed Packet Access network appropriately resourced for emulated Machine-to-Machine and Online Gaming traffic. Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are used for conveying application data and the measurement results from the end-user perspective are compared. In TCP case, traffic traces were recorded in two points of the core network as well (Gn interface and the firewall), and the overall delay is dissected to portions belonging to different parts of the network (access, core, backbone). The diversity of the traffic patterns used made it possible to draw conclusions concerning selectivity of certain parts of the network towards different traffic patterns, indicating the direction of future research on reducing latency for subject emerging application domains in legacy networks.

Keywords: *HSPA, Latency, M2M, Online Gaming, RTT, TCP, UDP*

1. INTRODUCTION

High-performance Online Gaming (OG) and Machine-to-Machine (M2M) are emerging applications for cellular networks, expected to create significant amount of traffic and interconnect a huge number of devices over the following years. In case of M2M, this number will exceed the human-to-human communications [1]. It is predicted that these applications, in addition to conventional voice and Internet traffic, will be an integral part of the traffic transported by Long-Term Evolution (LTE) networks.

In 3GPP, machine type traffic is a part of the Machine Type Communication (MTC) framework, which describes the exchange of data between two machines, also called M2M in ETSI [1][2][3]. These applications introduce additional traffic and bring in new requirements to the underlying mobile networks. In order to be able to cater for such increase combined with the change in the user and the node structure, it is important to understand the traffic characteristics. A large class of the traffic generated by emerging M2M applications requires low-latency, especially in the uplink access [4]. Low latency is critical for OG applications as well, in order to ensure the best gaming experience possible [5].

In this paper, we focus on latency performance of a live High-Speed Packet Access (HSPA) mobile network in the presence of M2M and OG applications. The measurements presented herein were done in the context of the ICT FP7 LOLA project [6].

The aim of these measurements was to assess latency per different classes of M2M and OG-like traffic in a live HSPA network, appropriately resourced in order to avoid any effects of congestion and to bring to light the pure network-traffic relations. Furthermore, latency performance of different parts of the network was assessed. Traffic models used in measurements were derived in [7][8]. The OG models were obtained by fitting statistical distributions to recorded traffic of real applications [8], whereas M2M models were defined according to the application scenario traffic specifications presented in [7][9]. The derived traffic models are quite diverse, covering a large number of possible applications, and thus well suited to reveal any network selectivity. The same traffic patterns were used throughout the whole project in order to compare the impact of different configurations and protocols used.

Based on the parameters provided as the modeling results, an Android application was developed, denoted as TG-App [9], with the goal of generating traffic according to the modeled parameters, i.e. different distributions of packet sizes and packet inter-arrival times. Measurements were performed using Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) transport protocols. The round-trip time (RTT) and the network cell statistics were analyzed.

The M2M traffic is mostly uplink-oriented, with various throughputs depending on the type of application [10]. The OG traffic is more balanced, fast, with generally small packets. Both uplink (client-server) and downlink (server-client) traffic of an application were tested in network uplink (UL), creating additional load for a live Node B. Previously conducted measurements [9][12][13] using the same traffic parameters and the same Node B had exposed a significant impact of these types of traffic on the live HSPA network. A large degradation of accessibility key performance indicator (KPI) [14] was observed. Further research showed that a Node B upgrade in terms of increased processing power and increased number of simultaneous HS users led to better network performance and latency decrease [13]. However, although the main network KPIs [14] were inside the regular limits, the latency performance was still not satisfactory and could be linked to certain KPIs deviations [10] [13]. Finally, the network was modernized, introducing shorter transmission time interval (TTI) of 2ms in the UL and the Node B further upgraded in order to eliminate the impact of any network bottleneck. Contemporary networks are designed to support traditional downlink-dominant traffic, so this additional traffic in the UL required the increase of resources, while the same additional traffic in the downlink (DL) would not present an issue.

Results presented in this paper are complementary to the results published in [12][13][15]. While previous papers mainly analyzed network behavior, i.e. the impact of new types of traffic on the network and ways to mitigate these effects, this paper focuses on detailed latency analysis, .i.e. the quality of service for subject applications, in a network appropriately equipped based on previous work results.

The paper is organized as follows. The next two sections describe the measurement setup and traffic patterns used in experiments. In Section 4, the analysis of measurement results is given. The comparison of

results for two protocols and different patterns is given in Section 5, and the comparison of TCP latency statistics at different points in the mobile network is given in Section 6. Sections 7 and 8 outline a summary with main conclusions.

2. MEASUREMENT SETUP

The measurement setup is shown in Figure 1. Setup is topologically the same as for measurements described in our previous work [9][12][13][15]. Traffic load was generated by ten mobile phones with Android 2.2 operating system. In order to minimize variations in delay originated in the device itself, all phones used in simulations were of the same model. Mobile network used for testing was a 3G network supporting High-Speed Downlink Packet Access (HSDPA) and enhanced uplink (eUL). Testing was done in a highly urban area. The test traffic, generated by TG-App applications on phones, was sent to a remote server, located in another town at approximate distance of 80km. Duration of the tests was about 1.5 hours.

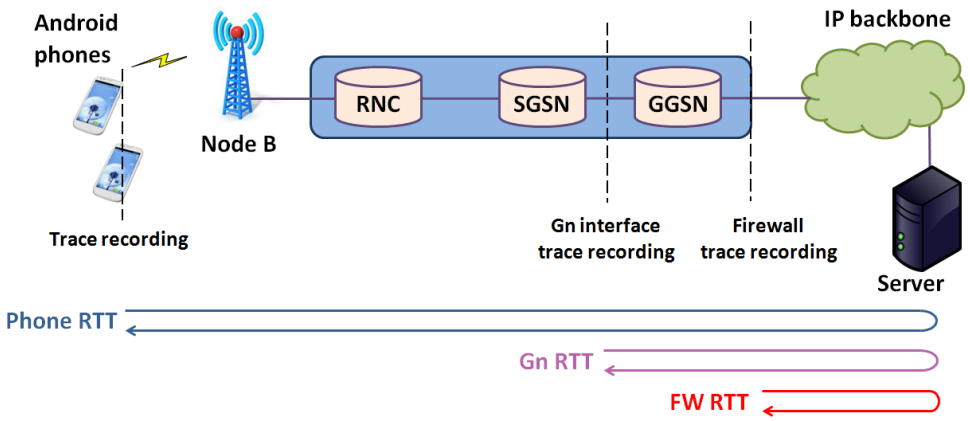

Fig. 1. Measurement setup.

The serving Node B had the following characteristics in the test cases analyzed herein:

- 256/256 channel elements (CE), UL /DL activated
- two carriers, HSPA traffic enabled on both carriers
- license for 32 simultaneous HSDPA users in each cell, and 16 eUL users
- 2ms TTI in UL

The default access point name (APN) was chosen, using proxy and service awareness feature. For data traffic, the direct tunnel functionality was deployed, i.e. a direct connection between the Radio Network Controller (RNC) and the Gateway GPRS Support Node (GGSN) in the user plane [16] was established.

In order to record traffic flow on the phones, the Android application "Shark for root" [17] was used. Traces were taken at the Gn interface and the firewall (FW) (connection point to the IP backbone) using the Wireshark application [18]. The network statistics were also gathered, in order to evaluate the network behavior compared to previous measurements [13].

For TCP traffic simulations, the value of RTT is measured as the time interval between sending a packet from the phone and receiving a corresponding ACK message from the TCP server. Since there was no possibility to measure one-way delay for UDP case, due to the lack of synchronization between tracing points, the TG-App was designed in such way that for every received UDP packet the server application generates a "fake ACK". The client application, upon receipt of a "fake ACK", calculates the "RTT" and records it in its report. Other details of a particular TG-App implementation of TCP and UDP shall be explained in detail in subsequent sections with measurement results. For testing purposes, either TCP or UDP were used for all phones, regardless of the application, while normally UDP would be used for gaming applications and TCP for M2M applications requiring data reliability.

3. TRAFFIC PATTERNS

The generated traffic for online gaming corresponded to the following applications:

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

Both simulated games belong to the First Person Shooter (FPS) class, a genre of video games that features a first-person perspective to the player. The primary design element is combat, involving all kinds of arms. FPS we consider in this paper offer a multiplayer mode, using a common server. For the success of the

players [19] low delay and jitter in the network are crucial and therefore these games can be considered quite challenging.

The generated traffic for M2M applications corresponded to the following applications:

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

Simulated applications are examples of the many possible M2M applications proposed and analyzed within the LOLA project. Bicycle Race is a machine-aided gaming application, where the opponents racing at different locations agree on the corresponding length of a race. In order to calculate and share the equivalent position of participants, measurements are taken by sensors (GPS, temperature, humidity, speed, terrain configuration) and exchanged between the opponents. Auto pilot scenario includes vehicle collision detection and avoidance (especially on highways). Team Tracking is a public safety application used to monitor the position of several nodes in a given environment (e.g. building, stadium) for situation awareness and consequent action scheduling.

Contrary to conventional bursty traffic, the above mentioned applications have continuous activity and varying throughput. Test traffic characteristics in terms of average packet size, time between packets and throughput are summarized in Table 1 for TG-App settings on every particular phone. Each phone had one TCP or UDP connection to the server.

TABLE 1. TEST TRAFFIC CHARACTERISTICS

Phone	Application, protocol	Settings	Average packet size [bytes]	Average time between packets [s]	Max throughput [kbps]	Min throughput [kbps]
test1	OA, UL, TCP/UDP	Gauss (0,04121;0,004497)kB, Uniform(0,069;0,103)s	40	0.086	6.68	1.82
test2	TF, UL, TCP/UDP	Gauss (0,07473;0,013085)kB, Uniform(0,031;0,042)s	75	0.0365	33.27	5.21
test3	OA,DL, TCP/UDP	Gauss (0,16836;0,08381)kB,	170	0.044	94.32	0.17

		Uniform(0,041;0,047)s				
test4	TF, DL, TCP/UDP	Gauss (0,23511;0,07748)kB, Uniform(0,039;0,046)s	240	0.0425	117.39	0.17
test5	M2M, BR, UL, TCP/UDP	Constant(1)kB, Uniform(0,1;0,5)s	1024	0.3	80.00	16.00
test6	M2M, BR, DL, TCP/UDP	Constant(1)kB, Uniform(0,1;0,5)s	1024	0.3	80.00	16.00
test7	M2M, AP, UL, TCP/UDP	Constant(1)kB, Uniform(0,025;0,1)s	1024	0.0625	320.00	80.00
test8	M2M, AP, DL, TCP/UDP	Constant(1)kB, Uniform(0,999;1,001)s	1024	1	8.01	7.99
test9	M2M, TT(GPS Keep Alive), UL, TCP/UDP	Constant(0,5)kB, Uniform(1;25)s	512	13	4.00	0.16
test10	M2M, TT(GPS Keep Alive), UL, TCP/UDP	Constant(0,5)kB, Uniform(1;25)s	512	13	4.00	0.16

4. MEASUREMENT RESULT ANALYSIS

In the following subsections, the analysis of measurement results is given. For UDP measurements, trace recording was performed only on phones, whereas for the TCP case it was performed on phones, at the Gn interface and at the firewall.

4.1 RTT measured by Shark Application on the Phone

4.1.1 UDP Measurement Result Analysis

The “fake ACK” is implemented in such way that the client application waits for it and upon receipt generates a new packet with the new wait time. If the fake ACK does not arrive within 3s, the application considers it lost. This implies that no delay larger than 3s was recorded. The application processing delay on the phone was less than 2 ms, which was verified by comparing UDP application reports and Wireshark traces from phones. Packet size is nominal size plus UDP header. The aforesaid implies that the traffic patterns did not follow the size/time distributions exactly, but very closely. In the Table 2, the obtained RTT statistics is given in the last column.

These statistics show significantly lower RTTs compared to results from previous measurements [12][15], when the same Node B was in a weaker configuration (less CEs, license for less simultaneous

HSPA users, HSPA on one carrier, 10 ms UL TTI). Average RTTs are multiple times smaller than in [12][15], as expected, except for phones 9 and 10 with very sporadic traffic patterns, whose RTTs are of the same order as previously. Obviously, the network upgrade had poor influence on reducing latency for these patterns.

It is clearly visible that traffic patterns of phones test5-test8 had more than 100ms longer RTTs than those of phones test1-test4. If we compare traffic characteristics (Table 1), we may deduce that this is due to the influence of packet length on RTT. Large packets have bigger RTTs. In this group of phones, we further see the influence of inter-arrival time (generated time between packets) – the phone test8 with largest inter-arrival time of 1 second had the highest RTT, while the phone test7, with smallest inter-arrival time got the smallest RTT. These results show that slower traffic patterns have larger RTTs compared to fast traffic.

4.1.2 TCP Measurement Result Analysis

When TCP is used as the transmission protocol, the TG-App exchanges a sequence of packets with the server for each application packet. The example in Figure 2 is given for the phone test9 whose nominal packet size was 512 bytes. The Figure 2 depicts different RTTs recorded for a sequence of packets exchanged between client and server for one application packet. The “First RTT“ recorded by Wireshark is the RTT to the receipt of server’s acknowledgement to the first 70B packet. The ”Second RTT“ is for the server’s response to the 578B packet with main payload. The last RTT recorded in the sequence, the “client ACK RTT”, is the RTT calculated for the backward direction. In the phone trace, it represents the application processing delay, since it is the time, recorded at the phone, between receiving the server’s ACK and sending the client’s acknowledgement for the received ACK to the server. Thus in the phone trace this last RTT recorded for the sequence is not related to the network. In Gn and FW traces, the “client ACK RTT” represents the backward RTT, from the tracing point to the phone and back, so it may be used in analysis for verification.

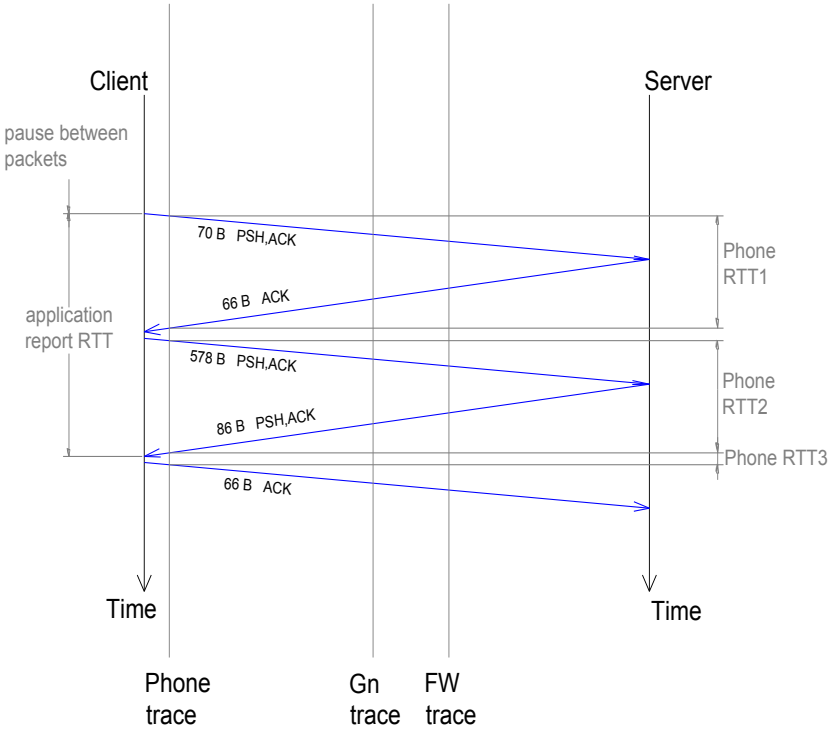

Fig. 2. Different RTTs calculated in the phone trace, for a sequence of packets corresponding to one application generated packet.

At this point, it would be convenient to introduce the following notation for different RTTs - $\overline{RTT}_n^{tracing\ point}$, where *tracing point* may be PH (*phone*), Gn (*Gn interface*) or FW (*firewall*), while *n* denotes which RTT in the sequence is in question, i.e. may take values 1 (*first TCP packet*), 2 (*TCP packet with main payload*) or 3 (*client ACK*).

The generated TCP packets obviously did not strictly follow the defined size/time traffic distributions, but the chosen M2M and online gaming patterns anyway represent only the subset of possible ones. The statistics taken from Shark traces captured on phones are given in Table 2.

Concerning the application processing delay, we observe from $\overline{RTT}_3^{PH}$ statistics that it is negligible, less than 2ms for all phones, as in the UDP case. Further, the average RTT for the 1st packet in sequence is larger than the RTT for the 2nd packet in sequence. The explanation lies in server processing delay for the 1st packet, which is further confirmed through the analysis of Gn and FW traces, and will be more thoroughly explored in the subsection on server processing delay.

We may also notice that in the group of phones with rather „fast“ traffic (test1-test8), largest RTTs for the second packet (main payload) are recorded for phones test5-test8. These phones have larger packets than phones test1-test4, and as in the UDP case, it may be concluded that packet size influences the RTT (bigger packets have larger RTTs).

TABLE 2. TCP AND UDP PHONE RTT STATISTICS TAKEN FROM SHARK TRACES

Phone	$\overline{RTT}_1^{PH}$ [ms]	$\overline{RTT}_2^{PH}$ (main payload) [ms]	$\overline{RTT}_3^{PH}$ (processing delay) [ms]	Average of UDP RTT [ms]
test1	348.6	148.9	1.2	118.0
test2	311.7	113.8	1.0	121.3
test3	309.1	124.3	1.1	131.4
test4	310.0	130.0	1.1	139.0
test5	325.2	230.8	1.2	256.6
test6	307.7	222.0	1.2	279.8
test7	316.9	228.7	1.4	236.7
test8	342.7	231.3	1.1	348.3
test9	1,532.5	439.5	0.6	1,716.5
test10	1,556.2	355.8	0.8	1,601.8

RTTs are generally smaller than for the previous TCP cases in [7][12][13][15] for all traffic patterns, due to the network upgrade. RTTs for phones test9 and test10, with large inter-arrival times, are still high, as in UDP case.

4.2 RTT measured by Wireshark on the Gn interface

Due to the large amount of traffic traversing the Gn interface, even with filtering, and the capabilities of recording devices, captured traces are only 3-5 minutes long in total. The Gn trace was obtained from two branches of the Gn interface, with laptops that were not synchronized. Chronological merge of obtained files was only possible within one branch. If packets originated from a specific phone went over one branch and the corresponding server's responses over the other, it was not possible to extract valid RTTs. This was the case with phones 2 and 3. The statistics for the Gn interface RTT are given in Table 3.

Again, as with phone traces, RTTs for first packets in sequence are large, and RTTs for the second ones are small, which supports the conclusion about significant server processing delay for the 1st TCP packet. Average RTTs for the 1st packet are of the same order, so no selectivity in the core and the backbone

concerning traffic patterns can be observed at first glance. The same is with statistics for 2nd TCP packets. Average client ACK RTTs suggest that the "backway" delay, to the phone and back, was less than or around 100ms.

4.3 RTT measured by Wireshark on the firewall

Firewall trace was captured using filtering by phone IP addresses. Phone test10 was rebooted after the beginning of the experiment so its IP address changed, and consequently no statistics could be reported. The statistics for the RTT measured at the firewall are given in Table 3.

Looking at maximum RTTs, large values (over 1s) are due to retransmissions, but these values occur in several instances, i.e. they do not influence much the average RTT, except for phone test9 with the slowest traffic (less samples). Again, the average RTTs for the 1st TCP packet are around 200ms larger than those for the 2nd TCP packet, which implies some server processing delay. Average RTTs for the 1st and for the 2nd packet show no apparent selectivity in the backbone concerning traffic pattern.

TABLE 3 TCP RTT STATISTICS FROM SHARK TRACES CAPTURED AT THE GN INTERFACE AND THE FIREWALL

Name	$\overline{RTT}_1^{Gn}$ [ms]	$\overline{RTT}_2^{Gn}$ (ma in payload) [ms]	$\overline{RTT}_3^{Gn}$ (Gn to phone and back) [ms]	$\overline{RTT}_1^{FW}$ [ms]	$\overline{RTT}_2^{FW}$ (main payload) [ms]	$\overline{RTT}_3^{FW}$ (FW to phone and back) [ms]
test1	231.8	26.9	128.4	227.0	23.8	125.1
test2	-	-	-	226.7	22.4	76.4
test3	-	-	-	225.2	23.2	78.2
test4	235.7	29.1	74.9	227.0	22.8	74.9
test5	221.2	29.7	72.9	225.7	25.0	81.5
test6	238.4	27.6	73.1	226.7	23.9	75.0
test7	242.3	26.7	79.8	227.7	23.4	79.9
test8	246.1	32.7	72.6	227.9	23.6	82.6
test9*	604.3*	18.3	105.1	235.5	104.4*	82.1
test10	221.3	22.5	81.1	-	-	-

*For phone test9 we have just a few packet sequences recorded properly. Average Gn RTTs are larger due to one retransmission of first packet with 3.3s RTT – without it the average RTT for one application packet would be 0,2345s i.e. of the order of other phones' RTTs. Large average FW RTT is due to a case of 3 retransmissions having 15s RTT – without this instance, the average FW RTT for the application packet would be 0,02554s, i.e. similar to other phones.

4.4 Server Processing Delay

In order to prove that the server processing delay for the 1st TCP packet in a sequence sent for one application packet was significant, around 200ms, as deduced from traces in previous sections, another round of measurements has been performed. Both TCP and UDP measurements were conducted, with trace recording on the server and on phones. The statistics was made for the whole communication of server application with 10 test phones, provided that if standard deviation would have been high, calculation by individual phones would be done.

The average delay within the server for the first TCP packet sent for one application packet was 200.5 ms, with the standard deviation of only 4ms. This proved that the server induced significant delay for the 1st TCP packet, regardless of the traffic patterns of individual phones. The processing delay for the 2nd TCP packet was mostly less than 1ms. In UDP case server processing had small influence on RTT, having 0.4ms delay in average with 0.8ms standard deviation.

5 COMPARISON OF RTT STATISTICS FOR DIFFERENT TRAFFIC PATTERNS

RTTs in UDP case are expected to be smaller than in TCP case, for all traffic patterns, because of UDP characteristics. Being a connectionless protocol, UDP does not establish and maintain a session. There are no retransmissions of packets, so the actual traffic on the link is closer to the defined patterns and RTTs for "successful" packets are smaller. There is no ordering of packets, so there's neither processing of out-of-order packets nor drops that might occur in the core (service-aware GGSN feature). Also, UDP has a smaller header compared to TCP, thus having smaller packets for the same data content, which can also influence the RTT.

The so far analysis has revealed the influence of inter-arrival time and packet size on latency for both TCP and UDP cases. Additional delay is induced in the access network, due to the stateful behaviour of the network [20], i.e. lower layer wireless procedures (e.g. schedulers, Radio Resource Control state machine). Large wait times between packets imply that the phone will often be assigned random access channels [20],

which are shared and have high access times. Further, very long generated inter-arrival time, if larger than the value of corresponding inactivity timers [20] of the WCDMA network (in this case, up to 22s), results in phone going to the Idle state. This means that for sending the next packet, the phone must first reestablish the Radio Resource Control (RRC) connection [21]. This situation occurs with phones test9 and test10, so additional time is being spent on control-plane signaling. Deep phone trace inspection for TCP case reveals that for smaller generated inter-arrival times "first RTTs" for these two patterns are smaller, since the phone maintains the RRC connection, established for previous packets. Large packets have larger delays due to segmentation on the radio interface, as observed for phones test5-test8 in both cases.

Further insight may be obtained through comparison of UDP and TCP results per traffic patterns. One should have in mind that the 1st TCP packet is only 70B long, while the second carries the main payload. For large inter-arrival times, the first TCP packet suffers the influence of accessing the network, i.e. establishing the RRC connection. It also suffers the influence of assigned traffic channel, according to the current throughput requirements and network conditions. The second TCP packet may experience larger delay due to the influence of generated packet size and the assigned channel. In UDP case, every packet sent suffers all impacts, so for this particular TCP/UDP implementation the advantages of UDP described at the beginning of this section disappear.

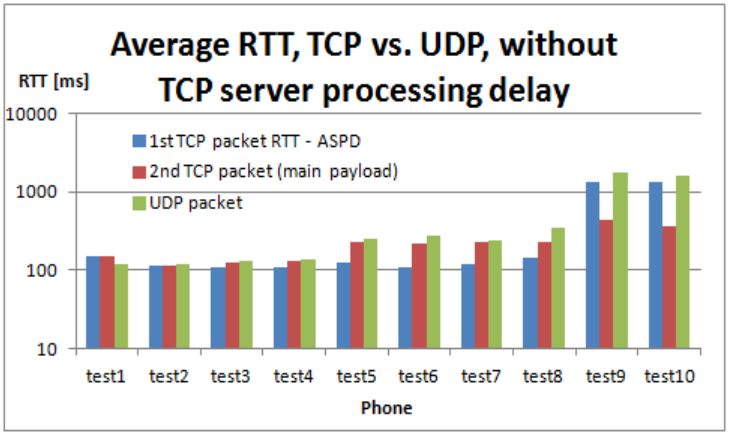

Fig. 3. TCP vs. UDP latency: average recorded RTTs, with average server processing delay subtracted from the 1st TCP packet RTT, logarithmic scale.

The latency statistics given in Table 2 are graphically represented in Figure 3, with average server processing delay (ASPD) subtracted from the average RTTs for the 1st TCP packet. By eliminating this delay inherent only to the 1st TCP packet, all three observed quantities become comparable.

The influence of packet length on latency is visible through RTT increase for phones test5-test8, i.e. traffic patterns with largest packets, for UDP packet and 2nd TCP packet, both carrying the main payload, compared to the average for the 1st TCP packet. Figure 4(a) shows the latency results with traffic patterns aligned per average nominal packet size, and average inter-arrival times displayed. Looking at the results for the 1st TCP packet, 70B long, and the results for the 2nd TCP packet and the UDP packet, carrying the main payload of nominal size, it is clear that large packets have larger latency. Yet, the results for 512B packet size, i.e. for patterns of phones test9 and test10, show that the inter-arrival time has much stronger influence on latency than the packet size.

Figure 4(b) shows the latency results with traffic patterns aligned per average inter-arrival times, and average packet sizes displayed. The influence of the inter-arrival time is mostly visible through results for phones test9 and test10, but also through results for phones with the same packet size of 1024 B – test5-test8. UDP and 1st TCP packet RTTs for phones test9 and test10, with largest inter-arrival time, show that these phones often went to the "Idle" state – inter-arrival time precedes the sending of these packets, and a large generated value results in transition to the "Idle" state. Further, 2nd TCP packet average RTTs are much smaller, but bigger than for other patterns. In the moment of sending the 2nd TCP packet, the phone is for sure in the RRC Connected state. This shows that the phones test9 and test10 were getting random access channels more often than other phones, even if in RRC Connected state, due to sporadic traffic patterns. Looking at the results for phones test5-test8, all with packets of the same size, the influence of longer inter-arrival time is visible through the increase of RTT for the 1st TCP packet and the UDP packet. These phones were in the RRC Connected state all the time, as their inter-arrival times were not long enough to force them to go to the "Idle" state. Again, longer wait times lead to the more frequent assignment of random access channels, inducing bigger latency.

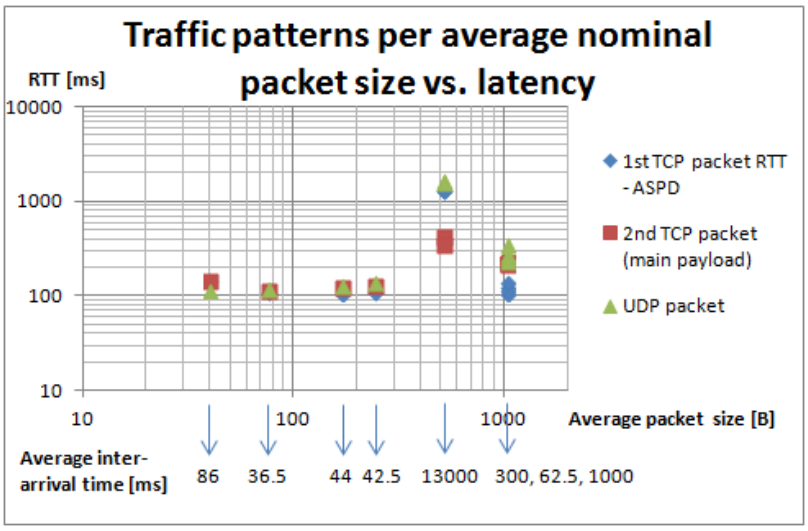

Fig. 4(a). Traffic patterns per average packet size vs. latency.

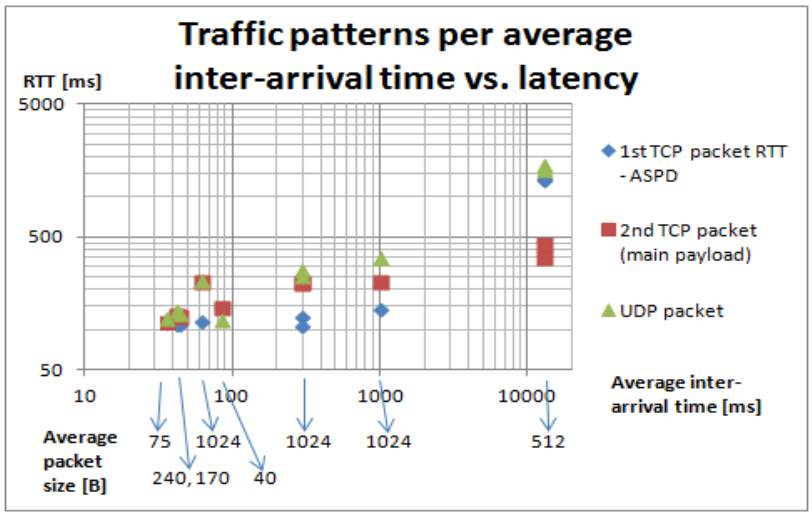

Fig. 4(b). Average inter-arrival time vs. latency.

In Figure 3 we observe that the 2nd TCP packet RTTs for phones test5-test8 are obviously smaller than UDP RTTs. The reason for this behavior, TCP and UDP packets being similar in size, may lie in the inter-arrival time. Inter-arrival time for the 2nd TCP packet is the last RTT, as the packet is sent immediately after the ACK for the 1st one is received, while for UDP packets wait time is the last "RTT" plus nominal wait time (see UDP section). For large packets, these differences in wait times and current throughputs obviously result in larger delay for UDP packets. Similarly, delay for the 2nd TCP packet of phones test9 and test10 is much larger than for other patterns (Figure 4a), since its inter-arrival time is the RTT for the 1st TCP packet, experiencing large delay.

It is clear that the effects of the inter-arrival time and packet size intertwine and that the delay of a packet depends strongly on network response to the previous packet flow.

6 COMPARISON OF RTT STATISTICS AT DIFFERENT POINTS AT NETWORK

For comparison purposes, by subtracting average TCP RTTs at different tracing points (Tables 2 and 3), we may assess latency induced in different parts of the network. The number of RTT samples captured at the Gn interface is much smaller than at other tracing points, so the values obtained by subtracting average values at different tracing points should be considered approximate, and compared accordingly.

By subtracting the average RTTs on the Gn interface from the averages on the phone, we get the approximate average two-way delay in the access network, including the transport network between the Node B and the RNC, and between the RNC and the SGSN and GGSN. These three nodes were physically at the same location, so the latter part of delay is negligible. The values are shown in Figure 5. The latency in the access is highest for phones test9 and test10, with large inter-arrival times. In the group of phones with comparable inter-arrival times, test1-test8, the 2nd TCP packet has larger latency in the access for phones test5-test8 than for others, due to the increased packet size. For phones test9 and test10, the influence of very large inter-arrival time on the 1st TCP packet RTT dominates over the influence of increased packet size on the 2nd TCP packet RTT. The conclusions from section V are valid, and the largest portion of latency is generated in the access network, which is very sensitive towards specific traffic patterns.

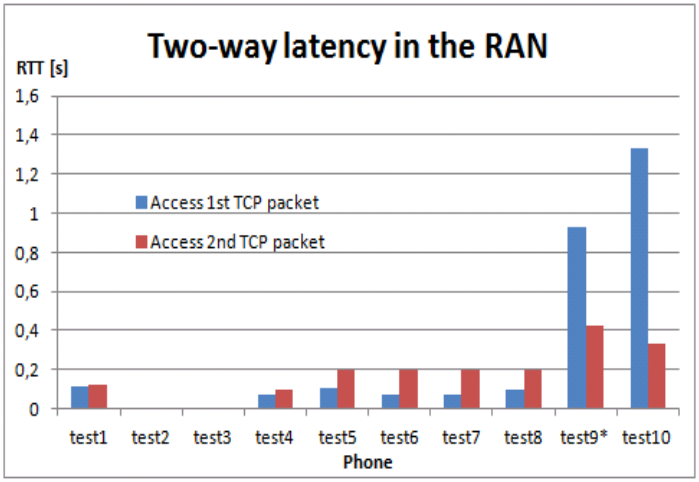

Fig. 5. Two-way latency in the RAN (Radio Access Network).

The two-way latency in the core may be obtained by subtracting average FW RTTs from the Gn RTTs. It is presented in Figure 6 and is generally less than 20 ms. Since the Gn and FW RTTs are of the same order of magnitude, and the Gn trace represents a small statistical sample, two negative values occurred that are omitted from the graph. The two-way latency in the backbone, eliminating the server processing delay for the 1st TCP packet, is shown in Figure 6 as well.

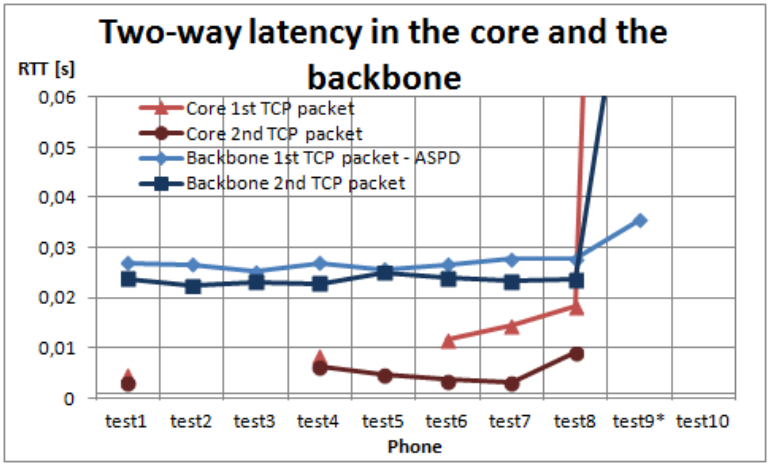

Fig. 6. Two-way latency in the core network and the backbone

For both the core and the backbone, two-way latency for the 2nd TCP packet shows no dependency on packet length. Interesting observation concerns the delay for the 1st TCP packet, being larger than the delay for the 2nd, for almost all traffic patterns (test9 results may be considered invalid as explained before). While in the core the reasons may as well be sought in specific mobile core features, the backbone simply represents a set of routers, with no flow management. Here, one should recall that the 1st TCP packet is always 70 B long, while the average length of the 2nd varies from 40 B (test1) to 1 kB. Thus, this small extra delay for the 1st packet may be attributed to some buffering mechanisms for small packets in the routers. Moreover, looking at the "core" values for phones test1-test8, seems that this extra delay becomes more apparent as main (2nd) packets grow larger (test5-test8), as if the huge variance of packet length for

successive packets contributes to this delay. Anyway, more certain conclusions cannot be drawn without further measurements.

7 DISCUSSION

The expansion of M2M and OG applications coincides with the rise of smartphone applications. Many of the latter generate background synchronization messages whose periodicity resembles the one of sporadic M2M applications, implying transitions to the Idle state. For such M2M applications, latency is usually not crucial, but both sporadic M2M and smartphone applications may create problems in the network (signaling overload) [22]. Nevertheless, for other M2M applications generating both sporadic keep-alive and event-based messages alternatively, low latency ensures a proper reaction.

On the other hand, small packets of latency critical M2M and OG applications resemble latency critical applications such as voice over IP [23]. The small-packets delay in routers re-gains attention in the literature lately, as for small-packets applications in low-latency access networks (e.g. VoLTE: *Voice over LT*E) this delay in the core might be of significance. LTE is a recognized technology for deploying M2M applications, whose population growth might impose issues in the core as well, as the routers' CPUs become overloaded processing excessive number of small packets.

In order to support the required QoS for latency-critical M2M applications, the access network needs to overcome issues with large inter-arrival times. In HSPA networks, introducing Cell/URA_PCH states [21], that would prevent frequent transitions to the Idle state, might solve this problem to some extent. For the mentioned event-based M2M applications, keep-alive messages may also be important since they could ensure the RRC Connected state yielding less signaling and lower latency for the upcoming event-based message. Yet, synchronized keep-alive messages for a large number of connected devices might again impose signaling problems for the network, and any bottleneck in the network implies higher latency.

The network should be planned according to expected traffic patterns, but dimensioning the network designed for downlink-dominant traffic according to uplink-oriented traffic demands implies rather small

network utilization [13]. As in a legacy HSPA network resource allocation schemes cannot be effectively changed, traffic shaping could be of help for reducing inherent access delay for large packets (segmentation at higher layers with smaller inter-arrival times). Traffic aggregation may be applied for mitigating the effects of large inter-arrival times and reducing core delay for small packets.

While the spatial concentration of OG users might be expected to be high only at some events (e.g. sport events), the foreseen ubiquity of M2M devices will inevitably yield large number of connections and huge amount of uplink traffic in some areas. For latency critical applications using legacy networks, due to issues described above, the most probable solution would be to use dedicated networks with M2M gateways, as proposed for LTE-A [24]. For less critical applications and applications with more sparse distribution of devices, some of the above suggested methods may be applied. The reported results are obtained in a properly resourced network, while any additional load in terms of traditional traffic or new types of traffic would challenge the delay figures [13].

8 CONCLUSION

Measurements presented in this paper were performed in an HSPA network suitably resourced to enable the observation of pure relationship between the traffic parameters and latency, without any network bottlenecks that could impact this relation. Main conclusions that can be taken from the analysis performed are as follows.

The largest part of delay is generated in the radio access network. The size of the inter-arrival time (wait time between packets) influences RTT most strongly. This influence is due to inherent properties of the radio access network [20]: sporadic traffic with low throughput is assigned with common channels, which are shared, with a collision risk, offering high access times. Very large inter-arrival time results in UE (User Equipment) going to the “Idle” state, which further imposes excessive signaling in the network as for the next packet the UE has to reestablish the RRC connection. Packet size also influences the delay in the access network, however to a smaller extent. Large packets have bigger RTTs, due to segmentation on the

radio interface. The access network assigns and switches traffic channels [20], having direct impact on latency, according to the current application throughput and the state of buffers at the UE and the RNC, and the analysis has shown that the incurred delay strongly depends on "previous conditions" of the particular traffic stream. These results conform with the recent work in the area [25][26].

On the other hand, small packets induce some small additional delay in the backbone and, more pronouncedly, in the core. This extra delay is on the order of several milliseconds and may be attributed to buffering, but due to the small sample in the Gn interface trace, this should be explored further.

Although the literature focuses on enabling wide deployment of M2M applications in LTE/LTE-A, legacy networks will still play a significant role in the upcoming years [27]. Future research is planned to cover latency measurements for M2M and OG traffic patterns in LTE network and to explore strategies for assigning M2M users to different access technologies according to delay constraints and inherent properties of each technology.

ACKNOWLEDGMENT

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.

REFERENCES

[1] Orrevad A., (2009). M2M Traffic Characteristics (When machines participate in communication). *In Information and Communication Technology, KTH Sweden, Stockholm*, p. 56.
[2] 3GPP TS 22.368 v10.0.0, (March 2010). Service requirements for machine-type communications (MTC). *Avaliable at:* http://www.3gpp.org/ftp/Specs/archive/22_series/22.368/.
[3] ETSI TS 102 689, (Aug 2010). Machine-to-Machine communications (M2M); M2M service requirements
[4] Chen Y., Wang W., (2010). Machine-to-machine communication in LTE-A. *In Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE, 72nd, Sept. 2010*, pp. 1 –4.
[5] Manweiler J., Agarwal S., Zhang M., Roy Choudhury R., and Bahl P, (2011). Switchboard: a matchmaking system for multiplayer mobile games. *In Proceedings of the 9th international conference on Mobile systems, applications, and services, ser. ACM MobiSys,*.
[6] ICT FP7 LOLA project, http://www.ict-lola.eu/.

[7] LOLA Consortium, (2012). Deliverable 3.5 Traffic Models for M2M and Online Gaming Network Traffic
[8] LOLA Consortium, (2010). Deliverable 3.3 Summary of the Traffic measurements
[9] LOLA Consortium, (2010). Deliverable 2.1 Target application scenario
[10]Drajić D., Krčo S., Tomić I., Svoboda P., Popović M., Nikaein N. and Zeljković N, (2012). Traffic generation application for simulating online games and M2M applications via wireless networks. *WONS 2012, Courmayeur, Italy.*
[11]Shafiq M. Z., Ji L., Liu A. X., Pang J., Wang J., (2012). A First Look at Cellular Machine-to-Machine Traffic – Large Scale Measurement and Characterization. *In SIGMETRICS/Performance'12, London, UK*
[12]Drajić D, Popović M., Nikaein N., Krčo S., Svoboda P., Tomić I. and Zeljković N, (2012). Impact of Online Games and M2M Applications Traffic on the Performance of HSPA Radio Access Networks. *IMIS 2012, Palermo, Italy.*
[13]Popović M, Drajić D., Krčo S., (2013). Evaluation of the UTRAN (HSPA) performance in different configurations in the presence of M2M and Online Gaming traffic. *In Transactions on Emerging Telecommunications Technologies, Wiley and Sons.* DOI: 10.1002/ett2738.
[14]ITU-T, Recommendation E.800, (1994). Terms and definitions related to quality of service and network performance including dependability
[15]Drajić D., Popović M., Nikaein N., Krčo S., Svoboda P., Tomić I. and Zeljković N., (2012). HSPA radio access performance evaluation for Online games and M2M applications traffic (TCP vs UDP). *TELFOR 2012*, Belgrade.
[16]3GPP TR 23.919, (2007). 3rd Generation Partnership Project, Technical Specification Group Services and System Aspects; Direct Tunnel Deployment Guideline (Release 7)
[17]http://www.androidzoom.com/android_applications/tools/shark-for-root_imrr.html.
[18]www.wireshark.org.
[19]Claypool M., Claypool K. (2010). Latency can kill: precision and deadline in online games. *In MMSys'10, Scottsdale, AZ,*
[20]Holma H., Toskala A., (2010). *WCDMA for UMTS: HSPA Evolution and LTE (5th Edt.)*. Wiley
[21]3GPP TS 25.331, (2012). 3rd Generation Partnership Project, Technical Specification Group Radio Access Network; Radio Resource Control (RRC); Protocol Specification (Release 6)
[22]Huawei, (July 2012) Smartphone Solutions White Paper. Available at: www.huawei.com.
[23]Stoke Tech Insights, (August 2012). Will Small Packets Degrade Your Network Performance. Available at: www.stoke.com.
[24]Zheng K, Hu F., Wang W., Xiang W., Dohler M., (July 2012) Radio resource allocation in LTE-Advanced Cellular Networks with M2M Communications. *IEEE Communications Magazine*
[25]Laner M., Svoboda P., Rupp M., (2012). Latency Analysis of 3G Network Components. *In EW'12, Poznan, Poland*
[26]Laner M., Svoboda P., Rupp M., (2012). A Comparison Between One-way Delays in Operating HSPA and LTE Networks. *In WiNMee'12, Paderborn, Germany,*
[27]GSMA, (2013). The Mobile Economy 2013. *Available at* www.gsma.com.