Impact of online games and M2M applications traffic on performance of HSPA radio access networks

Dejan Drajic[#1], Milica Popovic[#2], Navid Nikaein[*3,]Srdjan Krco[#1], Philipp Svoboda[*4], Igor Tomic[#1], Nenad Zeljkovic[#2]

[#1]*Ericsson d.o.o.*
Milentija Popovica 5a, Belgrade, Serbia
{dejan.drajic, srdjan.krco, igor.tomic}@ericsson.com

[#2]*Telekom a.d.*
Takovska 2, Belgrade, Serbia
{milicapop,nenadz}@telekom.rs

[#3] *Mobile Communication Department, Eurecom*
06904, Sophia Antipolis, France
navid.nikaein@eurecom.fr

[*4] *Institute of telecommunications, Vienna University of Technology*
Gusshausstrasse 25/38A, 1040Wien, Austria
psvoboda@nt.tuwien.ac.at

***Abstract* — In this paper we presented results of the performance evaluation of the WCDMA/HSPA radio access network when loaded with multiplayer real-time games and M2M applications traffic. Evaluation was done in a live network, on one Node B. Traffic, characteristic for the emerging Machine Type Communication and online multiplayer games, was generated by an application running on 10 mobile phones in parallel, all connected to the Node B under test. The focus of the work was on the RTT (*Round Trip Time*) and radio cell statistics. The main goal of the cell statistics analysis was to evaluate potential impact of the additional traffic on the performance of 3G/HSPA radio access network. It is concluded that system is designed more for continuous traffic patterns with high data rate in general, than for sporadic traffic patterns with very low data rate, like that of M2M or online gaming nodes. Therefore, the cost of signaling and control activity is dominant in comparison to the data rate.**

***Keywords* — M2M, Traffic modelling, HSPA network, performance evaluation**

I. Introduction

Machine type traffic and gaming are types of applications that are increasingly using mobile networks to transfer data to central servers or interact with other devices/machines and players. 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 Machine to Machine (M2M) [1]. Together with additional traffic, these applications are also introducing new requirements on the underlying mobile networks. The market for M2M applications will grow in the upcoming years; according to some estimations 50 billion M2M devices will be active in year 2020 0. In order to be able to cater for such increase and also the change in the user and the node structure, it is important to understand the underlying traffic models.

In the online games domain, low latency is particularly critical for an avatar model of online games with high precision weapons, the so called massive multi-user online first-person shooter type of games. In the near future, we can expect that handheld gaming devices will be equipped with embedded mobile interface, such as HSPA or LTE which will further increase demand on the mobile networks.

M2M is in the focus of the mobile industry for some time now and along with the ongoing activities in the research community, efforts towards understanding the impact of M2M on the mobile network architecture and specification of the relevant standards are under way (for example ETSI M2M and 3GPP). An important characteristic of the machine type communications is the variety of possible communication patterns, with heterogeneous requirements and features (see examples in Appendix B of 3GPP TS 22.368 0).

In this paper, we focus on latency requirements and performance of a live mobile network in the presence of M2M and online game applications. This work is done in the context of the ICT FP7 LOLA project, European Academia/Industry collaborative project [3]. The goal of the LOLA project is to provide significant technological advances in terms of minimizing end-to-end latency in wireless systems. LOLA targets low-latency applications found in machine-to-machine (M2M) communications and highly-interactive services such as gaming or remote control. In [4], a summary of functions and theoretical approaches for M2M traffic nodes based on literature and general ideas is presented together with an approach to modelling network traffic for M2M applications.

In 0, the results for online gaming traffic for four different applications-games are presented: Open Arena, Team Fortress 2, Dirt2 and World of Warcraft. These online gaming applications were defined for measurements due to their high impact on the gaming market. In [6], we presented several scenarios for M2M applications that require low transport network delay. The M2M traffic characteristics are analysed for the following applications: autopilot, virtual race, team tracking, and sensor-based alarm and event detection. In [4], detailed traffic modelling is done for above mentioned applications. Based on the parameters provided as the modelling results, a mobile phone application for packet-level traffic generation, denoted as TG-App, was developed with the goal of implementing modelled parameters, like different distributions of packet sizes and packet inter-arrival times, and to fulfil identified requirements, like TCP or UDP transmission of packets, multi-connection (few parallel TCP or UDP sessions). Using the TG-App, the RTT (*Round Trip Time*) and cell statistics are analysed.

In this paper we present and comment the results of the new experiments carried out after the pilot experiment, which is presented in an earlier paper [7]. In addition, the traces obtained from the Gn interface were analysed (see Figure 1).

This paper is organized as follows. Section 2 provides a brief explanation of measurement setup and traffic generation application. In Sections 3 and 4, the measurement results related to latency and cell statistics are presented and analysed. Sections 5 and 6 provides a summary with concluding remarks.

II. MEASUREMENT SETUP

The measurements described in the [7] were performed with 6 phones of different makes. This heterogeneity impacted the results hence in order to get more reliable results we conducted new simulations with 10 phones with Android 2.2 OS from the same manufacturer and the same model. This way we eliminated the impact of different phone hardware and software used in the experiment. In order to have better understanding of different portions of delays introduced by the network elements, traces are also collected on the Gn interface.

Mobile 3G network used for testing is network with HSDPA/HSUPA support on the serving NodeB, located in the highly urban area.

The NodeB under test had the following characteristics:

- 96/96 CE (Channel elements) UL/DL activated;
- eUL activated (and HSDPA);
- license for 16 simultaneous HS users;
- two carriers, HS traffic going to second carrier.

All phones were accessing Internet through the 3G HSDPA network – NodeB, RNC, SGSN, GGSN - and the server is located at a remote site at approximate distance of 80 km from the phones. The measurement setup is shown on Figure 1.

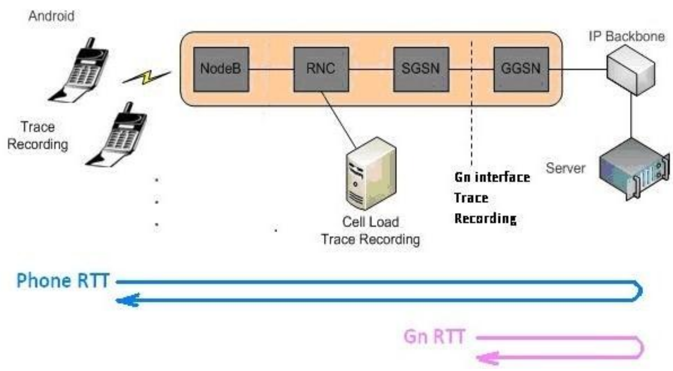

Figure 1: Measurement setup

The NodeB under test was connected to a RNC (RBG01) with a Hybrid IP link – 4E1+18Mbps – where all data traffic (HS, R99) is mapped to IP transport. All phones were static, and not in use by the users, thus excluding influence of user interaction with tests and located in two sectors, cells BGU44B, BGU44C on the first carrier or BGU44J, BGU44K on the second carrier. The cells are the same as the ones used in [7].

To analyze the RTT, an Android-based traffic generator application, TG-App, was used with corresponding server-side functionality [7]. The latter is installed on the server to which the application sends data. For TCP traffic simulations, the value of RTT is measured and displayed for each packet. The measured RTT is an estimation of the time interval between sending a packet from the phone and receiving a corresponding ACK message from the TCP server. TG-App records start/stop time, packet size distribution parameters, inter-arrival time distribution parameters, generated packet lengths, RTTs and serving cell IDs, and makes a report. Traces were taken at phones using Shark reader application, and at the Gn interfaces using Wireshark. For capturing packets on the Gn interface, only the traffic originated from the RNC of interest was filtered, and due to large traffic on this interface, even with filtering, just 3-4 minutes of traces were taken. In order to calculate RTT and analyse traffic on the phone, i.e. exchange of information between phone and server, Shark for root [8] traffic sniffer was used on the phones. After that related logs of the sniffer have been analysed, and results are produced.

TCP multi-connection type of simulation is performed with ten phones: 4 phones simulating online gaming, and 6 phones simulating MTC applications (each phone has only one TCP connection). This was done in order to analyse and understand behaviour of the serving cell(s), for the case of different load with online games and M2M applications and the impact of different traffic patterns on the cell statistics.

The following online games were simulated:

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

Both games are first person shooter (FPS). FPS is a genre of video games that features a first-person perspective to the player. The primary design element is combat, involving all kind of arms. FPS we consider in this paper offer a multiplayer mode, which allows many human players to play on a common server. Considering the nature of the game, low delay and jitters in the network are crucial for the success of the players, therefore these games can be considered as quite challenging for these kinds of network parameters. TABLE I presents the parameters configured for TG-App on every particular phone for online gaming simulations [7][9].

The following M2M applications were simulated:

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

These applications are examples of the many possible M2M applications which are proposed and analysed within LOLA project. Bicycle Race is a machined-aided gaming type of application, where the opponents are at different locations and agree about a circuit and the corresponding length of a race.

TABLE I
SIMULATION PARAMETERS FOR TCP MULTI-CONNECTION, 4 PHONES WITH SIMULATION OF ONLINE GAMING

Phone Number	**Application/ Direction Protocol**	**Packet sizes distribution/ Time distribution**
1	OA, UL, TCP	Gauss (42.2;4.6)B, Uniform (69,103)ms
2	TF, UL, TCP	Gauss (76.52;13.9)B, Uniform (31,42)ms
3	OA,DL, TCP	Gauss (0.172;0.05)kB, Uniform (41,47)ms
4	TF, DL, TCP	Gauss (0.241;0.06)kB, Uniform (39,46)ms

To calculate and share the equivalent position of each participants, measurements are taken by sensors (GPS, temperature, humidity, speed, terrain configuration) and are exchanged between the opponents. Auto pilot scenario includes both vehicle collision detection and avoidance (especially on highways) and how the urgency actions are taken in case of an accident. Team Tracking (TT) 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. TABLE II presents the parameters configured for TG-App on every particular phone for M2M applications [7][9].

TABLE II
SIMULATION PARAMETERS FOR TCP MULTI-CONNECTION 6 PHONES WITH M2M APPLICATION

Phone Number	**Application/Direction Protocol**	**Packet sizes distribution/ Time distribution**
5	BR, UL, TCP	Constant (1)kB,Uniform (100,500)ms
6	BR, DL, TCP	Constant (1kB),Uniform (100,500)ms
7	AP, UL, TCP	Constant(1)kB, Uniform(25;100)ms
8	AP, DL, TCP	Constant(1)kB, Constant(1000)ms
9, 10	TT(GPS Keep Alive), UL, TCP	Constant(0.5)kB, Uniform(1;60)s

The idea behind this new experiment was to re-run test defined in [7] with the phones from the same manufacturer, beside 6 phones with the same traffic parameters as in [7], 4 more phones are added which simulate traffic from M2M applications, namely phones 7 and 8 are running Auto pilot application, while phones 9 and 10 are running TT application. In the TT application we've simulated GPS keep alive messages, which were supposed to be generated uniformly in the interval 1-60 sec. Unfortunately, in the case of longer time periods generated, TCP connection with the server will be lost (on the server side there is TCP connection time-out, and in the case that there is no traffic for a longer period TCP connection will be dropped, and application will failed). Since it happened very often during the simulation after half an hour we changed time distribution to interval 1-3 sec, and after that there were no more application failures. This period is quite shorter than it is supposed to be (3 instead of 60 seconds), but we assume that also period of few seconds is large enough to observe network behavior for the case of M2M applications.

The applications were activated for all phones, and duration of the simulation was 2 hours. During simulation, behaviour of all phones was regular, and we didn't face any difficulties and problems with the application and phones. It was possible to root all phones and to run Shark for root

One of the main characteristics of the above mentioned applications is that they are uplink dominant and generating small packets with low data-rate, i.e., less than 64kbit/s, in contrast to convention downlink-dominant continuous traffic.

III MEASUREMENT RESULTS ANALYSIS

On the Figures 2 and 3 are presented results for phone RTT (see Figure 1) for all phones obtained from the Shark for root traces. On the x-axis the time of the day when the simulation was conducted is presented, and on the y-axis the RTT delay in seconds is given.

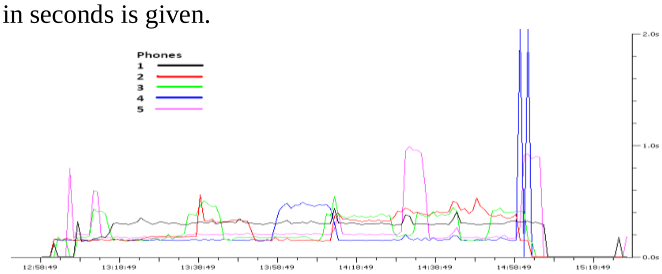

Figure 2: RTT for the phones 1-5

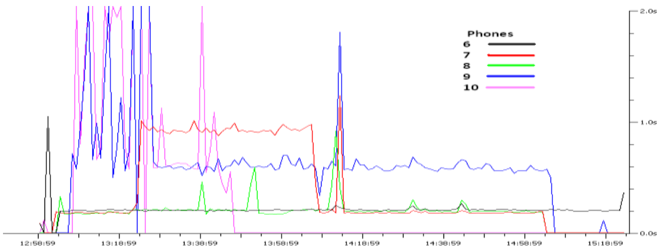

Figure 3: RTT for the phones 6-10

TABLE III presents the basic statistic for RTT and delay variance and median for phones.

TABLE III
AVERAGE, VARIANCE AND MEDIAN OF THE RTT FOR THE DIFFERENT PHONES IN THE MEASUREMENT

Phone	**Average (RTT)**	**Variance (RTT)**	**Median (RTT)**
1	288.535ms	77.800ms	289.393ms
2	223.556ms	93.418ms	148.490ms
3	226.497ms	109.613ms	179.447ms
4	173.554ms	193.135ms	142.123ms
5	216.952ms	146.553ms	246.307ms
6	206.047ms	29.788ms	279.286ms
7	271.226ms	231.850ms	232.158ms
8	205.985ms	125.298ms	269.485ms
9	672.479ms	3595.509ms	434.310ms
10	867.882ms	5567.476ms	403.131ms

Phones 1-4 simulating online games generally show smaller RTTs with less peaks, and with smaller variance values. The phones 5-8 representing M2M applications showed more-or-less the same RTT values as phone simulating online games, but with dual behavior, i.e. showing alternate periods of being in the region of stable RTT values and in the peaks up to one second. Phones 9 and 10 have extremely high peaks in the first half hour, when interval for packet generation was 60 seconds, and after decreasing time interval to 3 seconds RTT value decreased to 600 ms. A big difference between average and median values could be seen for phones 9 and 10, which could be explained with the high peak to average RTT ratio (see Figure 3).

The traffic generation application itself makes text reports, with RTT measured for each packet, as well as the size of the packet and Cell ID of the serving cell. Looking at statistics taken from application text reports, lowest recorded values of RTT are 112ms and 128ms, but for most phones even the lowest values are bigger than the average ones given by Shark trace. It can be concluded that the application itself introduces an extra processing delay when measuring the RTT.

Gn RTT is the time from the moment the packet passes Gn interface on its way to the server until the moment its appropriate ACK returns from the server and passes Gn interface on its way back to the phone. In other words, the Gn RTT comprises delays in the GGSN, proxy (if used, and the selected default APN uses it), firewall, and in the outer network routers and the transport, as well as processing delay in the server application. Though the real exit from the mobile network is the firewall, we will refer to this Gn RTT as to the "outer" network delay, just for purposes of differentiating the portion of RTT that belongs to the "inner" network (radio part, NodeB, transport, RNC, SGSN. From Gn traces, it can be seen that mean RTT value is mostly around 125ms, but some peaks could be noted. Phone 5 has a significant peak over 1s, as well as phone 9. Phones 3, 8 and 10 have peaks going to 500ms. The peaks of Gn RTT are generated either in the part GGSN–proxy-firewall, either in the server part, if we assume that the IP backbone network has a non-selective processing delay. On the Figure 4 Gn RTTs for the phones 6-10 are presented.

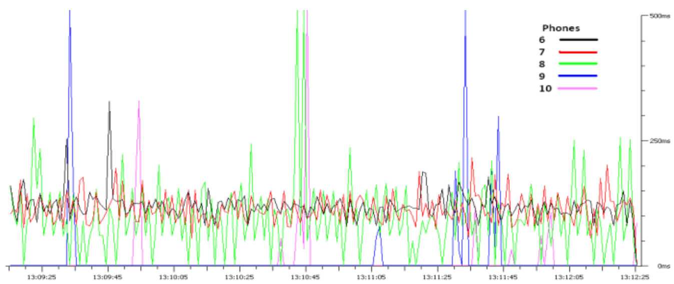

Figure 4: Gn RTT for the phones 6-10

For each test case, and for each phone, Gn traces were filtered by its IP address, then merged chronologically with Shark trace from the phone, and then RTTs on Gn (black line) and phone (red line) are displayed on a graph together, for comparison. The Gn RTT refers to latency in the outer network, i.e. from GGSN to the server and back. By comparing it with phone RTT, we can see which part of phone RTT belongs to the outer network, as well as the difference between RTTs belonging to the "inner" mobile network. On the Figure 5 Gn RTT and phone RTT are presented for the phone 5.

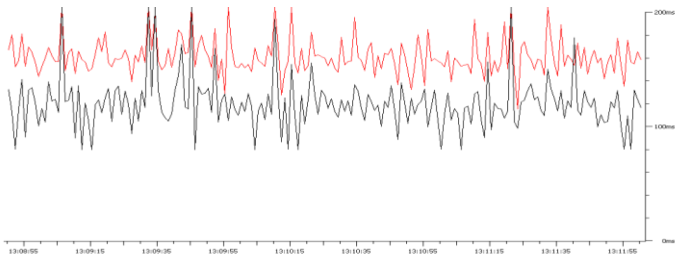
Figure 5: Gn RTT and Phone RTT for the phone 5

From traces, it can be concluded that the greater part of RTT belongs to the "outer" network. Of the overall RTT, around 30ms to 100ms belongs to the mobile network itself, the greater part of it expected to be generated on radio interface and processing delay at NodeB (and eventually transport network). We are looking at average values, since no time synchronization existed between Gn and phone measurements.

Concerning the difference between phone and Gn RTTs, it can be seen that phones simulating online games, have the smallest difference i.e. the smallest delay in the "inner" network, than the phones 5-10 simulating M2M applications. The phones with sporadic traffic patterns (9 and 10) are also more likely to experience higher delays in the "outer" network after the Gn interface. Possible causes for larger Gn RTTs are the usage of proxy, firewall, delay in the server network or server processing delay. In order to evaluate the usage of proxy, we plan to conduct a new test case – with changed APNs, so the proxy is not used.

This behavior is same as the behavior already observed in [7] - low data-rate M2M application experience very large end-to-end delay values, and the reason can be found in the mobility management entity and specific scheduling policy of the UMTS network [10]. Low data-rate applications with sporadic patterns are more likely to be given a random access channel for data communications. These channels offer very high access times at a low data-rate causing very high end to end delay numbers.

IV Cell statistics analysis

Cell statistics is gathered for 4 cells of interest. Serving Node B BGU44 has two carriers deployed in 2 sectors of interest – B and C sector/cell, while J and K denote cells of the second carrier. According to radio network design, the first, basic carrier is used for idle mode camping, call establishments and serving R99 traffic (speech, video calls and R99 data services). The second carrier is used for serving HSPA traffic mainly. Radio parameters and algorithms are set to control UE behavior in a way where all UE's are camping in idle mode on the first carrier. During call establishment procedure, in case that UE is HSPA capable and requesting data service, the call will be transferred through blind handover (coverage relation is defined) to second carrier. All phones used are HSUPA (EUL) capable. Main test traffic was on the uplink, in opposition to regular users' traffic that is dominant on the downlink. Test traffic in the downlink mainly consists of TCP's acknowledgements (ACKs). In given radio conditions (indoor, 150m from the NodeB) there is no dominant serving cell (TABLE IV), as the application reports also suggest. In the following tables, cell IDs reported by the application are given for each phone, with the number of packets sent over a specific cell, and the percentage regarding the overall number of packets sent. The following statistics should be taken with precaution, since regarding the radio conditions, many phones were probably in softer handover, communicating with NodeB over two sectors at the same time, which is not recorded by the application.

TABLE IV TOTAL NUMBER OF PACKETS PER CELL

Cell	Packets	Percentage
BGU44B	30173	43,22%
BGU44C	25019	35,84%
BGU44J	9604	13,76%
BGU44K	5009	7,18%
Total:	69805	

Phones 1 and 2 were mostly using BGU44B (and BGU44C), phones 3 and 4 mostly BGU44C (and BGU44B), 5 and 6 were using alternately cells J and K, while phones 7-10 mostly used B and C cells. As mentioned before, all HS traffic goes to cells J and K (2^{nd} carrier), while R99 traffic is served on both carriers (i.e. all 4 cells, B-J and C-K), depending on thresholds defined by operator. The dominant usage of B and C cells means that phones were mostly getting R99 RABs.

As expected uplink traffic during testing is higher than in the rest of the day. It is notable that with more test traffic, especially in cell B which is mostly used, PsCommon (Packet switch Common channel) is the most dominant RAB. Due to high load on those cells, admission and congestion control are assigning to users the least demanding PS RAB on common channels, to reduce the risk of user being denied service. The second carrier is generally scheduled for data services, especially demanding ones. Here we observe that the Ps64 (64Kbit/s) being the dominant RAB, with small portions of the Ps384(384Kb/s) and EUL. We note again a similar observation as above in that the higher load when the resources are limited leads to acquiring lower RABs. As in [7] measurements, especially with higher regular traffic in the latest measurements, the limiting factor are the NodeB processing power, i.e. the number of active channel elements.

The main impact of additional traffic on the radio network performance was detected in the area of accessibility, defined as the ability of user to get connection for the requested service, where significant performance degradation was observed. On the first carrier, first we have speech accessibility degradation, going down to 80%, mostly due to failure in the RRC connection procedure (figure 6).

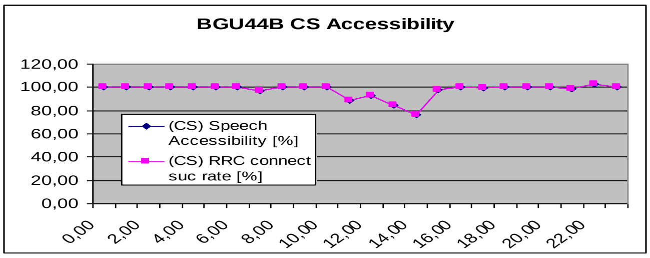

Figure 6. *CS Accessibility – BGU44B*

Second, we also have a huge degradation in PS accessibility, going to 0% – most probably due to lack of NodeB processing power and due to congestion in code three on downlink as well (figure 7).

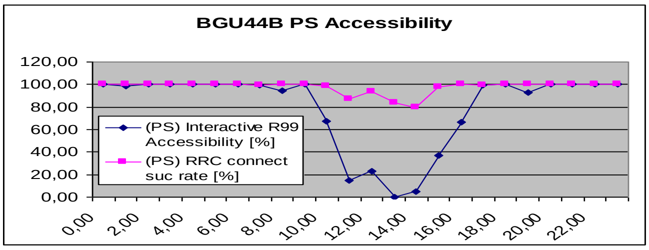

Figure 7. *PS Accessibility – BGU44B*

For the second carrier we have the following results:

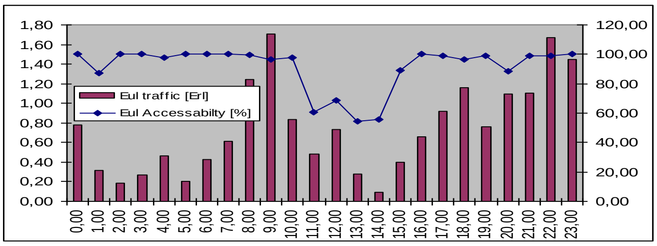

Figure 8. UL traffic and *HS Accessibility – BGU44J*

First, there is a degradation of overall data service accessibility, going lower than 50%. This HS accessibility degradation is mainly due to insufficient licenses for simultaneous active HS users (the actual number being 16 users), but could be due to insufficient channel elements as well. It should be noticed that looking at HSDPA and EUL graphs, we see that in the moments of lowest accessibility, in testing hours, the traffic itself is not in its peak data rate, this behaviour is expected, as in situation when network is congested, many users that would generate a lot of traffic are not admitted, which in turn decreases the total amount of traffic.

In the area of retainability, packet drop rate on BGU44B is about 2%, and slightly lower on BGU44C. On the second carrier, HS drop rate is negligible on both cells, while R99 packet drop rate is highest for K cell, about 0.9%. We can say that impact on retainability is minimal. Also as we saw before during [7] tests, in the area of integrity UL BLER and throughput are slightly affected by additional traffic. To conclude, the major impact of additional traffic on radio network performance is in the area of accessibility.

V Discussion

The results obtained in this paper reveals that the most challenging problem, in view of a rapid increase in the number of online gaming users and machines connected to cellular infrastructure, is the co-habitation of conventional continuous traffic with the emerging sporadic traffic. They suggest that the current cellular system are mainly designed and optimized for continuous data traffics mainly in the downlink and high data rate, at least in terms of the time-scales needed to send several IP packets, which in turn makes the control signalling overhead manageable. In comparison to the file-download or web-browsing types of application M2M applications are mostly uplink-dominant characterized by small and nearly constant packet sizes with low data rate, less than 1KByte/s. In some M2M applications, such packets are very small in size (i.e. few bytes) and extremely low duty cycle, which from the system throughput perspective represents a vanishing data rate.

A large class of the traffic generated by these emerging applications can require low-latency, especially in uplink access [11]. For example, for online gaming application, the low-latency is critical to offer the best game experience as possible [12]. Large portions of M2M applications are expected to be delay-sensitive as to reduce the: (i) latency of the event-driven realtime packets such as alarm notification, and (ii) power consumption of the sensing device. Therefore, the control signalling overhead translates directly to the latency-increase as it becomes the dominant factor when compared to the payload. In addition, such traffic requires uplink resources at Node B to handle large number of simultaneous users and increase the cell accessibility, which in turn improve the latency.

VI Conclusion

In this paper, we presented the results of an analysis of the impact of the emerging MTC and gaming applications on the delay and cell performance in WCDMA/HSPA wireless networks. The initial results of this analysis were presented in [7]. Building on that work, the tests were repeated with 10 identical phones, thus eliminating the negative effect observed in the initial tests with mobile phones from different phone manufacturers. The traffic generation application was also upgraded with new features, allowing better insight into the RTT and cell statistic analysis. Also, the RTT on the Gn interface was measured this time which facilitated better understanding of the delay structure. It was concluded that the inner (network) part of the RTT delay vary from 30-100ms, while the main portion of the total RTT delay is generated outside the mobile operator network. We also observed that with few number of uplink traffic, there is a degradation of overall data service accessibility, going lower than 50%.

Based on the new measurements, it can be seen that the average RTTs, excluding the extreme values, are smaller than RTTs obtained in [7]. This can be explained by introduction of the IP transport for data traffic which leads to smaller RTTs. The cell statistics is significantly deteriorated compared to the initial measurements. One important factor contributing to this is the time of the year when the initial and new measurements were done - the initial measurements were performed during a holiday period, while the new ones were done in October when both students and the working people are back in town. Since the Node B under test is located in a highly urban area, with several commercial and university buildings in the area, the impact of additional users is not negligible. In such an environment, with a base station that cannot handle all data requests adequately, it can be seen from the traces that phones with sporadic traffic patterns have RTTs being very high in some periods, which is, as was concluded earlier, due to mobility management entity of the UMTS network, low data-rate applications with sporadic patterns being given a random access channel for data communications. The main conclusion is that, especially in a situation with lack of network resources, more continuous traffic passes better than sporadic traffic patterns, especially when the data rate is very low. Existing systems are designed more for high throughput in general, than for sporadic traffic patterns from many users, like M2M or online gaming nodes. Therefore, the cost of signaling and control activity is dominant in comparison to the data rate.

The measurements of the NodeB processing power and the number of simultaneous HS users were identified as major bottlenecks in [7]. In the future work, we will explore the impact of these two factors on cell statistics and consequently on RTTs – repeating the measurements with 10 test phones with the same traffic distributions, but with the NodeB under test upgraded with more processing power (doubled Channel Elements) and doubled licenses for simultaneous HS users.

In order to check the RTT peaks on the Gn interface, that were attributed to the “service aware“ functionalities of the GGSN, as well as to proxy and firewall, we will repeat measurements with APN not using proxy, nor “service aware“ features, and using different firewalls. Also in the future work we will analyse transmission, latencies and cell statistics via UDP transport protocol for the same traffic mixture.

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] 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/

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

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

[4] LOLA Consortium, Delierbale. 3.4 Online gaming traffic modeling , 2011.

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

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

[7] D. Drajic, S. Krco, I. Tomic, P. Svoboda, M. Popovic, N. Nikaein and N. Zeljkovic “Traffc generation application for simulationg online games and M2M applications via wireless networks”, WONS 2012, Italy

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

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

[10] A. Barbuzzi, F. Ricciato, and G. Boggia, Discovering Parameter Setting in 3G Networks via Active Measurements, IEEE Communications Letters, Vol. 12, No. 10, October 2008

[11] Y. Chen and W. Wang, “Machine-to-machine communication in LTE-A,” in Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE, 72nd, sept. 2010, pp. 1 –4.

[12] J. Manweiler, S. Agarwal, M. Zhang, R. Roy Choudhury, and P. Bahl, “Switchboard: a matchmaking system for multiplayer mobile games,” in Proceedings of the 9th international conference on Mobile systems, applications, and services, ser. ACM MobiSys, 2011.