Traffic generation application for simulating online games and M2M applications via wireless networks

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

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

Abstract — **In this paper we present results of traffic modeling and simulation of multiplayer real-time games and M2M applications using TCP protocol over Telekom Serbia HSPA mobile network, performed within the FP7 LOLA project. In this experimentation, the RTT (*Round Trip Time*) and cell statistics are analysed. For this purpose, a traffic generation application is developed for Android phones to generate traffic pattern for emerging Machine Type Communication and online multiplayer games in mobile wireless network for different offset of tests. Application is activated by setting the corresponding parameters related to desired test case, i.e. duration of testing, size and frequency of the data packets that the application sends to the server. The main goal of the cell statistic analysis is to evaluate potential impact of additional simulated traffic in view of increasing data-centric users on the performance of mobile wireless radio network.**

Keywords— **M2M, RTT, Traffic modeling, TCP, HSPA network, Android application**

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 part of the Machine Type Communication (MTC) framework which describes the exchange of data between two machines, also called Machine to Machine (M2M). Together with additional traffic, these applications are also introducing new requirements on the underlying mobile networks. In this paper, we focus on latency requirements and performance of a live mobile network in the presence of M2M and online game applications.

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 [2], [3]. 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 High-Speed Packet Access (HSPA) or LTE.

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 [1]). Another key aspect is the design of low-layer signalling which allows for extremely short acquisition times for event-driven traffic and switching between idle and active communication states. One of the most challenging problems is the co-habitation of M2M traffic with conventional user traffic, and moreover this is coupled with the potential of a rapid increase in the number of machines connected to cellular infrastructure in the coming decades.

This work is a follow-up of the results obtained in the context of ICT FP7 LOLA project, European Academia/Industry collaborative project. The goal of the LOLA 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 [8], a summary of functions and theoretical approaches for M2M traffic nodes based on literature and general ideas is presented together with an approach to modeling network traffic for M2M applications. In [6], 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 [3], 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 [8], detailed traffic modeling is done for mentioned applications. Based on parameters provided as the modeling results, Traffic Generation Application (TG-App) is developed with the goal of implementing modeled 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). Through the TG-App, the RTT (*Round Trip Time*) and cell statistics are analysed.

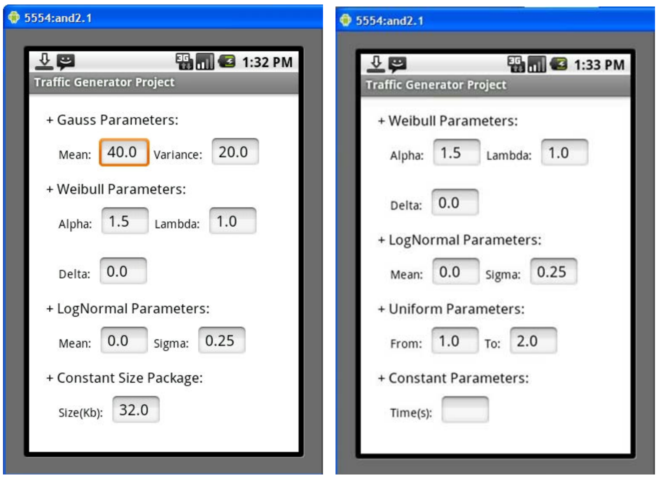
This paper is organized as following. Section 2 provides a detailed explanation of Traffic generation application and its parameters. In Section 3, the measurement setup and emulated traffic patterns are presented. In Section 4 and 5, the measurement results related to latency and cell statistics are presented and analysed. Finally, Section 6 provides a summary with concluding remarks.

II. TOOL (TRAFFIC GENERATION APPLICATION) DESCRIPTION

TG-App is developed for Android-based mobile phones both for the client side as well as the server-side. The latter is installed on the server towards which the mobile phone application will send data. This application provides the following distributions for packet sizes: LogNormal, Gaussian, Weibull distribution and Constant packet size; and the following distributions for packet inter-arrival times: Weibull, Gaussian, LogNormal and Uniform distribution.

Figure 1 provides examples of parameter settings for both packet size and inter-arrival time with appropriate distributions. The values for packet sizes are in kilobyte (kB), and for inter-arrival times are in second. On the left-hand side figure, the default parameters for the packets sizes distributions are given, while on the right-hand side, the default parameters for time distributions are given. As the result of the simulation, both the generated packet size and the time until the next packet generation are displayed on the output screen.

For the case of TCP traffic simulation, the value of RTT is measured and displayed for every packet. The measured RTT is an estimation of the time interval between sending the packet from the phone and receiving the ACK message from TCP server.



+ Gauss Parameters:
Mean: 40.0 Variance: 20.0
+ Weibull Parameters:
Alpha: 1.5 Lambda: 1.0
Delta: 0.0
+ LogNormal Parameters:
Mean: 0.0 Sigma: 0.25
+ Constant Size Package:
Size(Kb): 32.0

+ Weibull Parameters:
Alpha: 1.5 Lambda: 1.0
Delta: 0.0
+ LogNormal Parameters:
Mean: 0.0 Sigma: 0.25
+ Uniform Parameters:
From: 1.0 To: 2.0
+ Constant Parameters:
Time(s):

Figure 1: Selection of packet size and inter-arrival time distributions

III. MEASUREMENT SETUP AND TRAFFIC PATTERNS

The application is designed to generate traffic in mobile wireless network for different types of tests. All phones access Internet through the 3G HSDPA Telekom Network – NodeB, RNC, SGSN, GGSN - and the server is located at a remote site (in another town in Serbia, Novi Sad, at approximate distance of 80 km from Belgrade) with a public IPv4 address. The measurement setup is shown in the figure below.

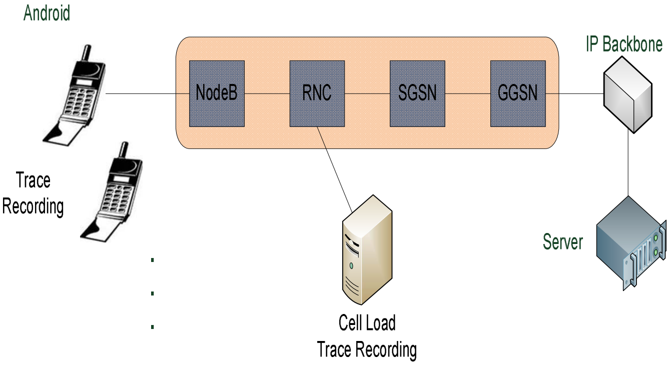


Figure 2: Measurement setup

Mobile phone application is sending packets towards the server according to a predefined traffic pattern. After a successful reception of the packet from a TCP connection, server is sending an acknowledgment (ACK) to the mobile phone application. In this procedure, the RRT is estimated as the total time interval between sending the packet and receiving the acknowledgment. In order to calculate RTT and analyse traffic on the phone, i.e. exchange of information between phone and server, Shark for root [5] traffic sniffer was used on the phones. After that related logs of the sniffer have been analysed, and results are produced. Totally, 6 phones were included in the testing. All 6 phones are supporting R99 data transfer on uplink (i.e. 64kbps or 384kbps Radio Access Bearer, RAB), with no support for HSUPA. It was possible on every mobile phone to install and run the TG-App, but it was not possible to root every phone in order to run Shark for root. We assume this is due to different implementations found in the mobiles from different vendors, and because of different versions of Android OS, and installed applications on the phones.

During the whole simulation period, all phones were static. They were located in two offices, i.e. indoors, relatively near to the serving base station Node B (BGU44), placed on the roof of the building. In the area of interest, Telekom Serbia deployed HSPA mobile wireless network with two WCDMA carriers. Since the two offices are in border coverage area between two sectors of the serving Node B, and having in mind indoor radio conditions and the possibility of softer handover, Telekom Serbia provided comprehensive statistics for four cells – two sectors of the same Node B with two carriers – these are cells BGU44 B, C for the first carrier and BGU44 J, K for the second, B and J cell belonging to one sector, and C and K to the other one. The objective of the cell statistics analysis is to reveal any performance degradations due to additional load in the cell, as test traffic is combined with the traffic generated by real users in a live network.

TCP multi-connection type of simulation is performed with six phones: 4 phones simulated online gaming, and 2 phones simulated MTC applications. 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. Besides, M2M traffic is modeled as an on-off traffic with constant/uniform packets sizes, while traffic of the online games corresponds to different distribution of packets sizes. The following paragraphs analyse the traffic characteristics for two online games and two MTC application.

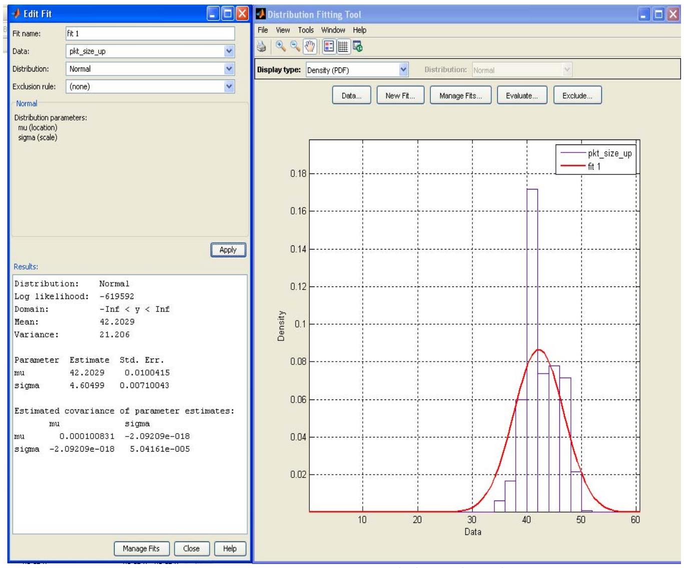
The following online games were simulated:

- 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.

Open Arena (OA) is an open source game based on the Quake3 game engine designed by id-software. It is a classic representative of FPS style games. The application traffic is encapsulated into UDP packets for network transmission. The application itself takes care about packet loss and reordering on its own. The standard network port used in OpenArena is 27960. The processing as depicted in [8] consists of filtering the traffic for this specific port of type UDP and records the packets at the IP layer. In the current release of the TG-App only TCP type traffic is supported. As we generated traffic strictly in one direction we conclude that for this case we can replace UDP with TCP for first analysis of the impact.

Team Fortress 2 (TF) is, similar to OA, a FPS online game. In contrast to OA it is closed source, the source code and the application function cannot be analysed in a direct way. The application is based on the game engine of Half Life 2 (HL2) of the software studio Valve. TF2 is purely focused on online gaming only. The application data is encapsulated into UDP packets for the network transmission. As in the case of OA the application itself has to implement all advanced functions to work in case of packet loss. The classical server port is 27005 and 27015. The typical application datagram size is below 500 Byte, therefore there is no need to apply any reassembly method in the post-processing step. The raw traces are dumped at the IP layer. Here is the example of how the parameters for online gaming simulations are obtained.



Fit name: fit 1
Data: pkt_size_up
Distribution: Normal
Exclusion rule: (none)

Normal
Distribution parameters:
mu (location)
sigma (scale)

Display type: Density (PDF)
Distribution: Normal

Results:

```
Distribution:     Normal
Log likelihood:   -619592
Domain:           -Inf < y < Inf
Mean:             42.2029
Variance:         21.206

Parameter  Estimate   Std. Err.
mu         42.2029     0.0100415
sigma      4.60499    0.00710043

Estimated covariance of parameter estimates:
       mu               sigma
mu       0.000100831  -2.09209e-018
sigma  -2.09209e-018   5.04161e-005
```

Figure 3: Open Arena: Up-Link packet size

The figure is obtained from MATLAB, representing the results of measurements performed in [6] (purple bars) and the statistic distribution that has the best fit (red curve) with measured packet size distribution. For the case of Open Arena, that is Gaussian (Normal) distribution in Uplink with parameters: mean value=42.2029, and sigma=4.60499, i.e. variance=21.206.

In the Table 1, fitted parameters for both uplink (UL) and downlink (DL) are given, which are used in the TG-APP.

TABLE I
PARAMETERS FOR ONLINE GAMING SIMULATIONS

	Open Arena	**Team Fortress**
Packet Size UL	Normal (42.2;4.6)kB	Normal (76.52;179.549)kB
Inter Packet Time UL	Uniform (69-103ms)	Uniform (31-42ms)
Packet Size DL	LogNormal (5.039;0.47)kB	LogNormal (5.43;0.32)kB
Inter Packet Time DL	Uniform (41-47ms)	Uniform (39-46ms)

The following M2M applications were simulated:

- Bicycle Race (BR)
- Auto Pilot (AP)

One example of the many possible M2M games is the virtual race (e.g. virtual bicycle race using real bicycles). The opponents are on different locations, possibly many kilometres away. At the beginning, the corresponding length of a race is agreed (i.e. 10 km or 20 min) between the peers. The measurements are taken by sensors (GPS, temperature, humidity, speed, terrain configuration etc.) and are exchanged between the opponents. They are used by the application to calculate the equivalent positions of the participants and to show them the corresponding state of the race (e.g. "you are leading by 10 m"). The number of competitors may be more than two, and all competitors must mutually exchange information, and the applications must present all participants the state of other competitors. During the race they are informed about the place and the distances from each other (e.g. "you are the 3rd behind the 2nd by 10 m and leading before the 4th by 15 m"). The packets containing GPS and sensors data are on the order of 1 kB. Taking into account the typical speeds (of bicycles) in this scenario (rarely higher than 50 km/h = 13.9 m/s), the packets should be exchanged approximately every 100-500 ms. The traffic is symmetric in both directions, i.e. uplink and downlink.

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. It is based on a M2M device equipped with sensors embedded in the cars and surrounding environment and used in automatic driving systems. These M2M devices (cars, road sign units, highway cameras) send information to a backend collision avoidance system. The backend system distributes notifications to all vehicles in the vicinity of the location of the collision, together with information required for potential actuation of relevant controls in the affected cars. In all receiving cars, the automatic driving systems based on the received information take over the control fully or partially (brakes activated, driving direction changed, seating belts tightened, passengers alerted etc). If there is no such system in a car, the driver is notified and instructed.

In the uplink periodic, low data rate keep-alive messages (GPS, speed, time) from the M2M devices to the backend system (once per minute, in the order of 1kB per message every 25-100ms depends on the car velocity) are sent towards the server. From the server side, if everything is OK on the road, backend system periodically (about every 1s) sends some notification messages to the cars with packet length of 1kB, which is used in this simulation case. In the case of accident which is not simulated here, short bursts emergency signals from the M2M backend to the M2M devices including warning and actuation commands (each burst in the order of 1kB every 10ms) are sent.

In the Bicycle race we assume symmetric traffic on the uplink and downlink, with constant packet size and packet inter-arrival time with uniform distribution, while in Auto Pilot application we have uniform distribution for packet inter-arrival time towards the server, and constant distribution from the server to the car. Since in both scenarios it is likely to have more than 2 participants, and we didn't have enough resources (in the sense of licenses for the number of simultaneous users in the HSPA cell) for the simulation of huge number of competitors/vehicles, we decided to perform simulation of the traffic from 8 competitors/vehicles, aggregated by a gateway. This produces the same traffic load in the cell using one traffic aggregator generating equivalent of 8 KB instead of 8 competitors/vehicles generating load of 1kB. Table 2 presents the TG-APP parameters for both uplink and downlink [6], [8].

TABLE II
PARAMETERS FOR M2M GAMING SIMULATIONS

	Bicycle Race	**Auto Pilot**
Packet Size UL	Constant (8kB)	Constant (8kB)
Inter Packet Time UL	Uniform (100-500ms)	Uniform (25-100ms)
Packet Size DL	Constant (8kB)	Constant (8kB)
Inter Packet Time DL	Uniform (100-500ms)	Constant (1000ms)

In these simulations we can simulate only uplink communications (generating traffic from the phone to the server), and on downlink we only have ACK packets in the case of TCP communication. Actually, virtual race is based on UDP protocol, but current version of the tool is supporting only TCP protocol, which we will use in the following simulations. In order to see system behaviour (cell statistics and RTT) and to observe influence of different traffic patterns and increased uplink load in the cell, we decided to simulate the downlink traffic model as well, but in the uplink, from mobile phone towards the server.

IV. MEASUREMENT RESULTS ANALYSIS

All the parameters configured for TG-App on every particular phone are summarized in the Table 3.

The applications were activated for all phones, with planned duration of 2 hours. Yet, the application on phone 3 was stopped by server earlier, due to extra-large packet received. Phone 2 also terminated before the end of the generation process, most likely due to user interaction. Phone 5 was stopped by core due to shortage of credits. The generation of extra-large packets requires an upper bound on the maximum packet size, which was not provided by the TG-App. Concerning phone 2, since most of the phones were in the regular usage by their owners (application was executing in the background), it could happen that the owner unintentionally stopped or interrupted the application.

As mentioned before, it was not possible to root every phone in order to run Shark for root. Phones 1 and 6 could not be rooted, so traces were captured for phones 2-5. For further delay analysis these traces are used, while for cell statistics influence of all 6 phones was observed and analysed.

TABLE III
SIMULATION PARAMETERS FOR TCP MULTI-CONNECTION, 4 PHONES WITH SIMULATION OF ONLINE GAMING, AND 2 PHONES WITH M2M APPLICATION SIMULATIONS

Phone Number	**Application/ Protocol**	**Packet sizes distribution/ Time distribution**	**Phone Brand and Model**
1	OA, UL, TCP	Gauss (42.2;4.6)kB, Uniform (69,103)ms	Samsung Galaxy S GT-I9000
2	TF, UL, TCP	Gauss (76.52;13.9)kB, Uniform (31,42)ms	Samsung Galaxy S GT-I9000
3	OA,DL, TCP	Gauss (0.172;0.05)kB, Uniform (41,47)ms	Huawei U8110
4	TF, DL, TCP	Gauss (0.241;0.06)kB, Uniform (39,46)ms	Huawei U8110
5	M2M, BR, UL, TCP	Constant (8kB), Uniform (100,500)ms	Samsung Spica I-5700
6	M2M, BR, DL, TCP	Constant (8kB), Uniform (100,500)ms	Samsung Galaxy S GT-I9000

From the Shark for root traces it can be seen that the low data-rate M2M application experience very large end-to-end delay values in the order of several seconds. The delay for the simulated gaming applications is nearly one order of magnitude smaller. The reason for the difference in delay faced by these two applications can be found in the mobility management entity of the UMTS network under test. Low data-rate applications with sporadic patterns are 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. The following table presents the basic statistic for RTT and delay variance for the four different phones.

TABLE IV
AVERAGE AND VARIANCE OF THE RTT FOR THE DIFFERENT PHONES IN THE MEASUREMENT

Phone	**App / Proto**	**Average (RTT)**	**Variance (RTT)**
2	TF, UL, TCP	336.887ms	1.44ms
3	OA,DL, TCP	370.166ms	2.71ms
4	TF, DL, TCP	185.986ms	0.81ms
5	M2M, BR, UL, TCP	4764.94ms	27.9ms

We can also observe in the figure 4 that the delay of Phone 5 is extremely large and could not be explained by the mobility states of UMTS, which would be expected to cause a delay of around 4-5 sec for sporadic network traffic on IP layer [11].

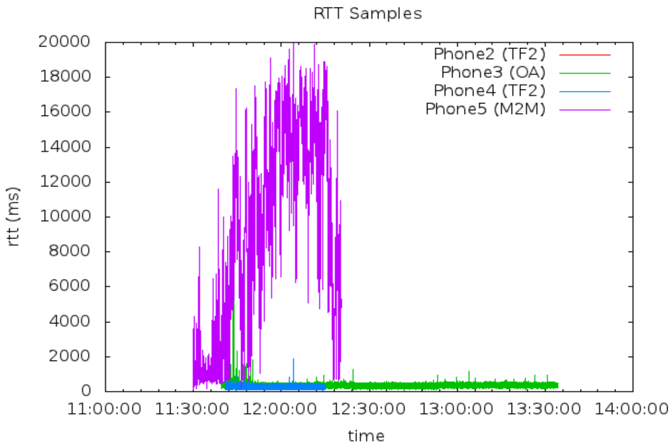


Figure 4: RTT Analysis for M2M and Online Gaming Traffic

A first analysis of the phones hardware lead to the assumption that the hardware of Phone 5 was insufficient to run the (non optimized) TG-App in the background. To verify this assumption we analysed a second trace recorded on another day where two phones generated M2M traffic featuring different, more potent, hardware.

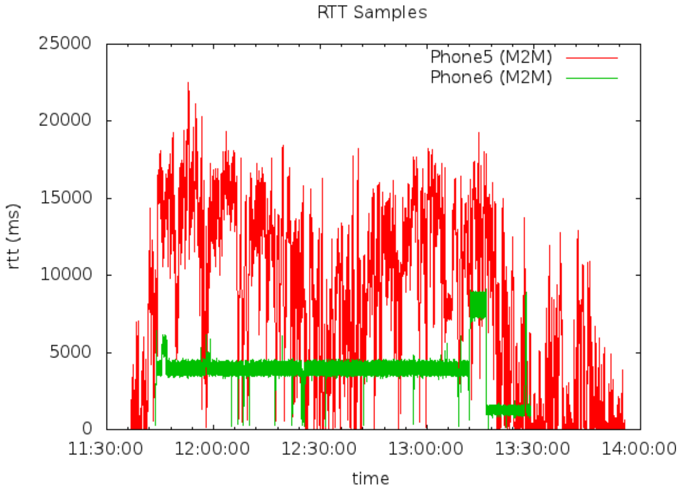


Figure 5: RTT Analysis for Phone 5 and 6 generating M2M traffic

The result of this analysis is presented in the Figure 5. Here we learn that the new phone shows the expected behaviour of 4-5 sec delay, while the Phone 5 again shows a delay pattern similar to the first day. We conclude from this that phone 5 was overloaded with the application. This result reconfirms that the phone capabilities play a key role in the performance of emerging MTC and gaming application and thus the user experience. Typical hardware configuration for this type of experiment should provide at least 800MHz CPU clock and 256MB internal memory.

V. CELL STATISTICS ANALYSIS

The main goal of cell statistics analysis is to evaluate potential impact of additional simulated traffic on mobile wireless radio network performance. The first step is to understand Telekom Serbia HSPA radio network design. In the area of interest Telekom Serbia deployed HSPA mobile wireless network with two WCDMA carriers. It means that same Node B through same antenna system is covering area of interest with two cells operating on two different frequency channels, where bandwidth of each channel is 5MHz. 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. Apart from HSPA data service traffic, there might be some speech handled on the second carrier, in case that user is simultaneously performing speech call and data service (multi RAB service). Radio parameters and algorithms are set to control UE behaviour in a way where all UEs are camping in idle mode on the first carrier. During call establishment procedure, in case that UE is HSPA capable and requesting for data service, the call will be transferred through blind handover (coverage relation is defined) to second carrier.

All UE terminals used in tests for traffic generation are HSDPA capable on downlink, while at the same time they do not support EUL – Enhanced Uplink (also known as HSUPA). Having in mind that the radio network design and applied radio parameter settings and strategy, the generated traffic by the test UEs will start connection on the first carrier and consume first carrier resources for a short period of time, after which they will be moved to the second carrier. Hence, generated traffic will impact performance of both carriers, and analysis need to cover both - first and second carrier cells performance.

Analysis of radio conditions, in the area where testing was organized, showed that there is no dominant server, and two cells are covering area of interest: BGU44B and BGU44C. On the second carrier, pairs to BGU44B and BGU44C are BGU44J and BGU44K cells. It is worth mentioning that all cells of interest belong to the same Node B (two different sectors). Test UEs were in softer handover state (handover between sectors within the same Node B) and when it comes to certain Node B resources like Node B processing power and channel elements, no matter being served by one or another cell, UE uses resources belonging to the same pool.

We can see on figures bellow that generated traffic on uplink is significant, much higher than carried traffic during remaining parts of the day when test phones were not active. It was expected, as the main traffic pattern of regular users is intensive on downlink, while simulated traffic (M2M, gaming) is mainly uplink dominant. Test UEs were served by both cells on the second carrier: BGU44J (with slightly higher amount of generated traffic) and BGU44K.

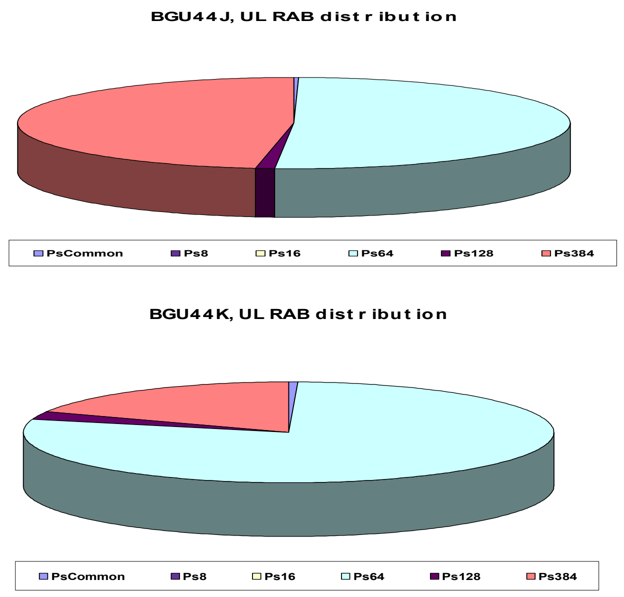


Figure 6. *Ul RAB distribution – BGU44J, BGU44J*

The distribution of used RAB's is showing that some users were able to use preferred 384 kbps RAB on the uplink, while some of them (approximately 50% in BGU44J and 75% in BGU44K) have been using 64 kbps RAB mostly due to bottleneck in available Node B processing power.

The main impact of additional traffic on radio network performance was detected in the area of accessibility (ability of user to get connection for requested service), where significant performance degradation was observed.

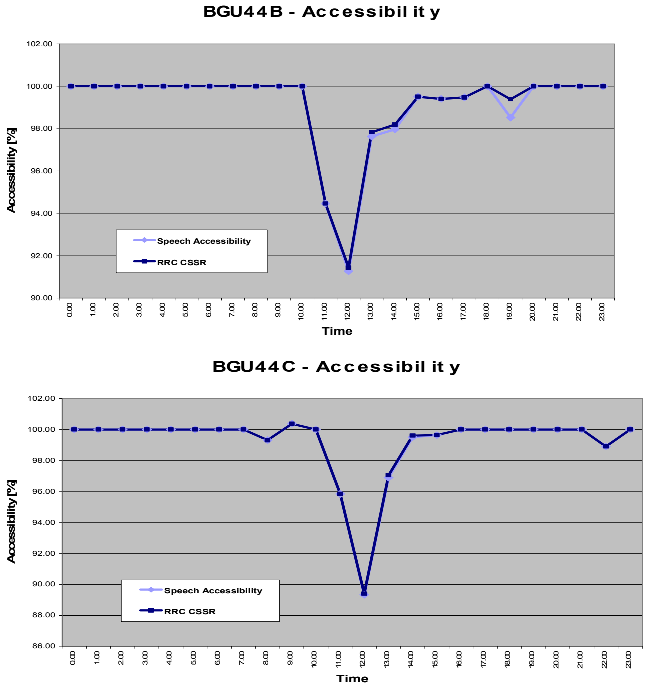


Figure 7. *Speech Accessibility – BGU44B, BGU44C*

Speech accessibility degradation on first carrier was detected, in the area from approximately 100% in regular situation to 90% in the presence of simulated traffic, mostly due to failures in RRC establishment procedure due to congestion on radio interface.

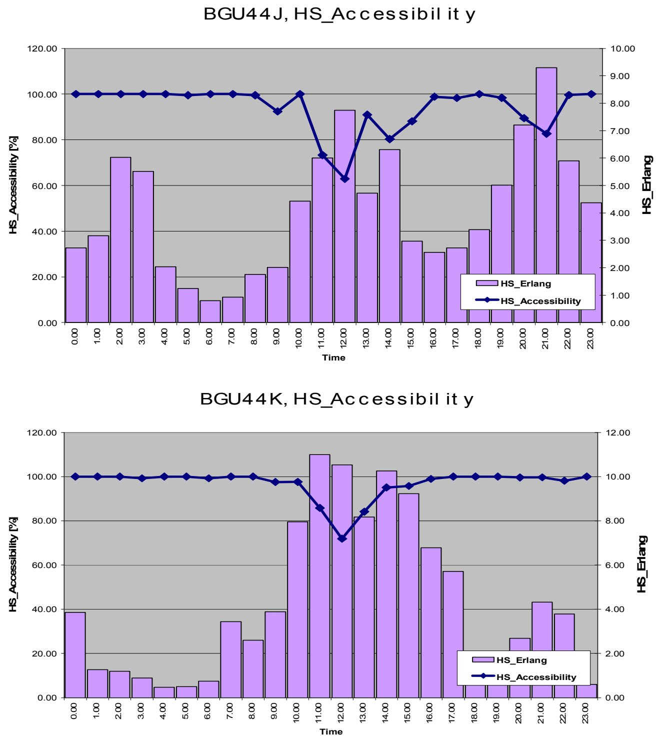


Figure 8. *HS Accessibility – BGU44J, BGU44K*

High Speed (HS) data service accessibility degradation on second carrier was detected, in the area from approximately 100% in regular situation to 60-80% in the presence of simulated traffic, as a result of insufficient license (allowing only 16 simultaneous active HS users at the same time).

Negative impact of additional traffic on the network performance in the area of retainability (ability of the user to keep connection - drop call rate degradation) or integrity (QoS requirements, i.e. Throughput) was not detected.

Another interesting phenomena is the impact of generated traffic on interference on uplink, that leads to cell breathing and coverage shrinking. Measurements on RSSI (Received Signal Strength Indicator) showed uplink interference growth and Noise Raise of 3.5 dB for BGU44J and 6 dB for BGU44K (Figure 9).

One of the most frequently used models for radio propagation modeling in mobile network planning process is Okumura-Hata. According to that model, path-loss is function of several factors: transmitter antenna height (Node B antenna), receiver antenna height (UE Antenna), area type, frequency used and distance from transmitter to receiver. If we assume transmitter antenna height of 30m, receiver antenna height of 1.5m, urban area type and standard used frequency for WCDMA of 2100 MHz, path-loss formula according Okumura-Hata model may be simplified to:

$$L\mathit{path}=a+b\log R\ [\mathrm{dB}]$$

where L*path* stands for Path-loss, a= 134.69, b= 35.22 and R is distance from Node B to UE. We can calculate how noise raise impacts cell coverage. For the case when interference grows by measured value of 3.5 dB, maximum path-loss will decrease by 3.5dB - which corresponds to maximum distance to Node B decrease by 20.5%.

Concluding the cell statistic analysis we can state that the presence of the uplink dominant data traffic generated by four online games and two M2M nodes causes the effective cell radius of the Node B to decrease by 32.5%.

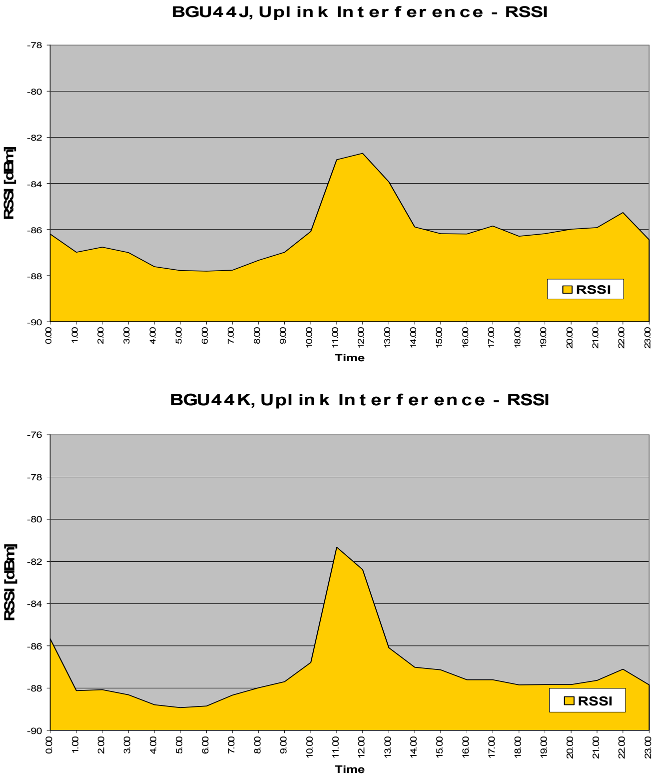


Figure 9. *Uplink Noise Rise – BGU44J, BGU44K*

VI. CONCLUSION

In this paper, we study the impact of emerging MTC and gaming application on the delay and cell performance of wireless network. The RTT analysis showed that one of the main impacts in the delay stems from the mobility management in UMTS causing one order of magnitude higher delay values for sporadic low data-rate traffic patterns (MTC) compared to online gaming data patterns. Analyse of additional data traffic on wireless network performance have shown that main impact can be expected in area of accessibility, where new traffic causes bottlenecks in the system in the area where it wasn't present before. Impact on KPIs in area of retainability and integrity was not shown to be significant with generated traffic. Finally, analysis showed that coverage may be affected as well, where significant cell shrinking was detected.

In the future work we will analyse transmission, latencies and cell statistics via UDP transport protocol for different kind and mixture of online games and M2M applications. In order to get more reliable results, we plan to run simulations with higher number of phones, i.e. 10 phones in parallel from the same manufacturer, thus also eliminating computational capacity of hardware used in the experiment. Furthermore, a new type of traffic emulation for M2M applications will be developed, enabling state emulation of the traffic, where user can define different states of emulation with different distributions and parameters, as the combination of the above mentioned distribution.

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

[4] LOLA Consortium, Deliverable 2.1 target application scenario, 2010R. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, “High-speed digital-to-RF converter,” U.S. Patent 5 668 842, Sept. 16, 1997.

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

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

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

[8] LOLA Consortium, Deliverable 3.4 Online gaming traffic modeling , 2011.

[9] Team Fortress 2 official website, http://www.teamfortress.com/

[10] OpenArena official website, http://www.openarena.ws/

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