LTE core network testing using generated traffic based on models from real-life data

Pál Varga
Dept. of Telecommunications and Media Informatics
Budapest University of Technology and Economics
2 Magyar Tudósok krt., Budapest, Hungary, H-1117
Email: pvarga@tmit.bme.hu

Péter Olaszi
AITIA International Inc.
Telecommunication Division
48-50 Czetz János str., Budapest, Hungary, H-1039
Email: polaszi@aitia.ai

***Abstract*—The aim of this paper is to describe a general methodology for testing the mobile network core with traffic generated from a simulated access-segment. A main element of this methodology is to derive load models and mobility models from real-life traffic patterns, and use these – as well as pre-recorded protocol messages – in the process. The Evolved Packet Core (EPC) of Long Term Evolution (LTE) provides a good base for presenting practical elements of this methodology. This paper demonstrates the workflow of this methodology through a practical example of EPC load testing. This demonstration covers issues related to nodes and interfaces covered during such tests; dialogs used for session and mobility control; and cross-correlating message flows that effect various elements across the architecture.**

I. Introduction

The main aim of this paper is to provide a methodology for testing the performance of mobile core networks by loading them with varying traffic that bears real-life features. The motivation is to identify certain limitations of mobile cores whose performance capabilities under extreme conditions are unclear. Beside laying down the theoretical foundations of an advanced performance testing methodology for mobile cores, this paper was written with strong practical attitude; the methodology stands on real-life-traced messages.

Although load testing through traffic generation is part of current network verification procedures, the advanced model-based approach introduced here is new. Some aspects of the current methodology have appeared in the earlier work of the authors [1], describing their live, deployed traffic analyzers working on 2G and 3G technologies.

There are solutions already available for EPC load testing; nevertheless, they fall behind the methodology described in this paper – as far as realistic traffic composition is concerned. In the case of the Polaris system [2] the test cases are flexibly configurable, but traffic variations are limited to changes in the overall traffic load rate. The TeraVM solution [3] has an impressive capacity of testing up to 1 Tbps, and it provides per-flow analysis features after the test – but its variability is limited to playing existing PCAP-format traces at amplified rates. The IXIA testing system [4] covers service/subscriber differentiation, although does not deal with overall traffic composition and modeling issues. The *ng4T* tester [5], the Torrent6100 [6], the Aricent solution [7], and the Nethawk EAST [8] all cover high variety of control plane procedures, but they all lack the feature of realistic variance in the user plane packet flows. The *ns-3* [9] Discrete-event Network Simulator can also be used for traffic generation – although it is primarily built for research purposes, hence connecting it to live equipment is challenging.

Following the general description of the advanced performance testing methodology for mobile core networks in Section II, its procedures are detailed in Sections III-VII: Data Capture, Data Analysis, Model Creation, Implementation, Verification and Refinement. The EPC-specific use case is described in Section VIII.

II. Methodology

The focus of this paper is the complex load testing of the mobile core by simulating the behavior of the access network at the edge of the core. The principle is to simulate the traffic of hundreds of thousands of users as they attach to the network, initiate user-related communication, and move within the network while producing and receiving traffic.

A major task of the methodology is traffic *analysis*, carried out in order to find proper *models* and parameters that fit this behavior. The models are implemented in the form of a traffic generator, which produces synthetic traffic whose statistical parameters match the *observed* real-life patterns. After *verification* and fine-tuning of the model, the traffic generator can be *deployed* in order to reveal the performance limitations of mobile core networks.

Figure 1 illustrates our methodology through the general cycle of modeling: *observation*, *analysis*, *model creation*, *implementation* and finally *verification* and *deployment*.

1) *Observation*: Capturing of real-life traffic data in an operational network. Traces are collected from both the control plane and the user plane.
2) *Analysis* is performed on the collected data. Types of various network activities are identified. For each activity, relevant features and statistical parameters are identified, and their values are extracted from the data.
3) *Model creation*: Based on the relevant parameters and message samples models are built, which account for the various activities and traffic patterns observed in the network.

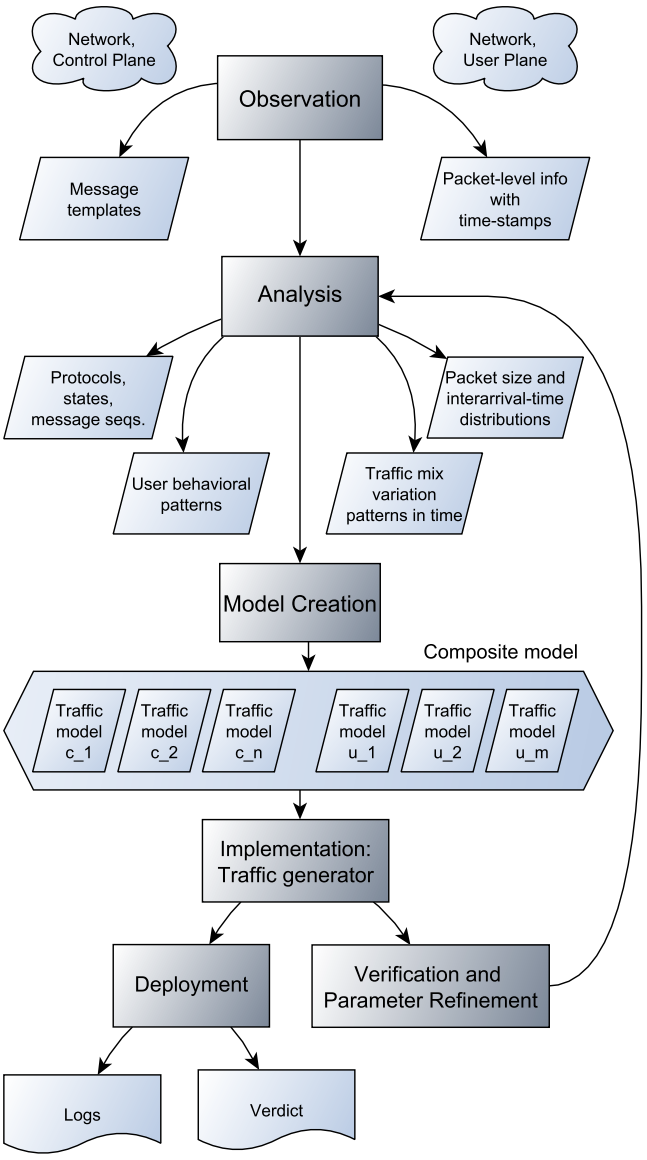

Fig. 1. Tasks and their results in the methodology. Shaded gray rectangles illustrate procedures; parallelograms represent intermediate data.

4) *Implementation*: The models are realized as a traffic generator. The device simulates the operation of a specific network segment through the parameters of the implemented models.
5) *Verification and refinement*: The statistical properties of the synthetic traffic are matched against those observed in real patterns. The models' parameters are refined in an iterative process in order to improve the prediction accuracy of the models.
6) *Deployment*: Once the models are considered sufficiently accurate, the traffic generator can be deployed in a pilot network (or live network) to carry out complex load testing tasks. At this point the models may be operated outside the previously observed realistic parameter range. Thus the device can be used to simulate extreme network activities or boundary conditions, which would otherwise be difficult or impossible to produce in a real-life setup.

Because of their distinct characteristics, control and user planes need to be addressed separately throughout the model creation process.

A. Control plane

Control plane messages are captured bit-by-bit. In the analysis phase, control message sequences and protocol state transitions are identified and stored. Subscriber mobility patterns are analyzed and their relevant parameters are identified.

The models created for the control plane are mainly based on protocol specifications: message sequences and state transitions need to conform to the standards and vendor-specific extensions. Subscriber mobility models, on the other hand, can as well employ statistical parameters.

B. User plane

User plane capture is a process of collecting data packets sent over the network.

The analysis needs to identify categories of user activities (such as voice calls, video streaming, web browsing or email traffic) and define relevant parameters which characterize the activities (e.g. packet delay and jitter for VoIP and video; expected value and variance of packet sizes for email download).

Typically different sets of parameters are identified for different types of observed activities.

C. Composite model

In order to model the traffic in the actual network, a collection of different traffic models need to be elaborated, each one accounting for a particular type of user activity. That is, different models need to be used for characterizing e.g. voice calls, video streaming, web browsing or email traffic. Similarly, on the control plane different models are needed for e.g. describing subscriber attach demand or mobility within the network.

The specific models are then combined into a *composite model* (Figure 2). The composite model characterizes the different types of subscriber activities on the control plane and tells us how particular subscriber activities contribute to the overall observed traffic on the user plane. In this sense the composite model is a super-model, whose parameters are the constituting models themselves.

In the following we describe in detail the process of data capture, analysis, model creation, implementation, verification and deployment.

III. Data capture

The motivation behind traffic capture in the described method is to provide data for analysis and model creation. Two key features of the capture process is *losslessness* and *accurate time-stamping*, which are addressed below. Capture is performed at multiple network interfaces of the control and the user plane.

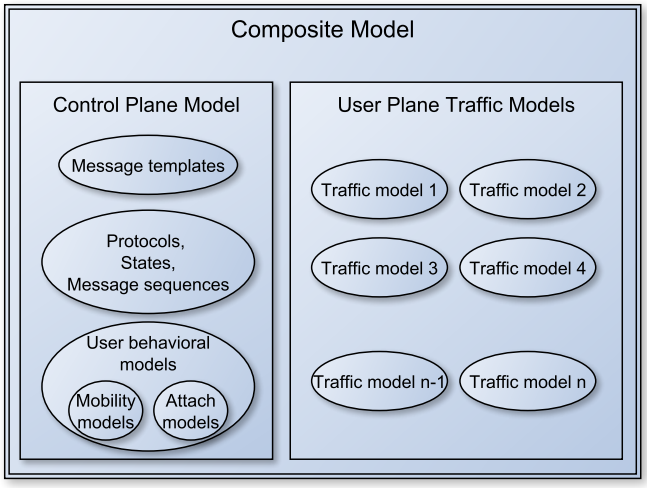

Fig. 2. Elements of the composite activity model. Different types of models describe different activities on the control and user plane.

A. Control plane data capture

The collected data are practically bit-by-bit captures of control plane traffic. The capture contains actual protocol messages along with mobility information. These control messages can be collected into a set of templates, used by the implemented traffic generator in the network testing phase.

Losslessness is a key requirement here. Missing even one control message may lead to missing e.g. a temporary identifier update, which in turn may lead to loss of a whole communication session.

In general, accurate time stamping is not of paramount importance for the control messages, as long as the original order of the messages is preserved. Time stamping of the *mobility* control messages, on the other hand, carry valuable information for model creation. Accurate time stamping is necessary in order to build a mobility model from the statistical properties of how location changes appear within the live network.

Note that control plane data account for a relatively small portion of the total traffic – the counter example being e.g. machine-to-machine communication.

B. User plane data capture

The actual contents of the user packets are not relevant from the model creation point of view. However, in-band signaling in the user plane needs to be captured and stored accurately, just like in the case of control plane messages.

In the case of user plane packet capture, losslessness is a less strict requirement. Actually it may be satisfactory to capture packet headers only. Whether or not headers only would suffice depends on the planned depth of the analysis, and the granularity of the modeling. As an example: traffic generation with random payload does not require payload fingerprint analysis. It is completely satisfactory to chop such packets during their capture, and merely analyze their time-stamp and the 5-tuple of "from-IP", "to-IP", "from-port", "to-port", and "protocol".

The requirement for the accurate time-stamping is also determined by the purpose of the modeling. When the model includes packet inter-arrival time modeling (or even taking long-range dependence into account), the theoretically smallest packet inter-arrival time determines the required time-stamping accuracy for proper capture. (It is 672 ns for 1 Gbps Ethernet, and 6.72 ns for 100 Gbps Ethernet.)

IV. Data analysis

The purpose of data analysis is to categorize activities in the network and identify relevant parameters for the various activities in order to build models from them. We would also like to find typical and non-typical traffic patterns so as to use them for generating varying, real-life-like synthetic traffic. The user and the control plane data should be analyzed separately.

A. Protocol analysis in the control plane

The control plane of the mobile core is responsible for call and session management, mobility management, policy control and charging [10]. Signaling on the control plane uses the message-oriented, reliable SCTP transport protocol. This ensures that the communicating devices always receive complete signaling messages in correct order.

The actual task of protocol analysis is to (i) identify the dialogs to be simulated (examples include the Initial Attach, Periodic Location Update, Normal Location Update and PDP Context Activation procedures), (ii) collect message templates for the dialogs and (iii) identify relevant message parameters.

Some subscriber activities can be characterized by statistical parameters. Such are the probability of changing location within the network, activating a PDP context or detaching from the network.

B. Traffic analysis in the user plane

Categorization of network activities is important for successful model creation. Data traffic is a result of parallel user activities: users are simultaneously engaged in voice calls, view video streams, browse the web, read emails, and so on.

Different user activities produce different traffic patterns. Each traffic pattern can be characterized by various parameters. These parameters are chosen so that they help building an effective traffic model for the particular traffic pattern.

From this point of view traffic analysis is a complex deconvolution problem. In the aggregate traffic flow of the user plane one needs to identify the various user activities and extract the relevant parameters, which characterize the traffic pattern produced by the particular type of activity.

User plane packets account for a significant portion of the overall traffic volume. Relevant parameters (such as packet type, size, source and destination addresses) are extracted from the data flow. The actual contents of the user plane packets can be discarded.

V. Model creation

The composite model described in this paper incorporates building blocks from earlier published modeling methods, briefly summarized in the following references.

Chandrasekaran [11] gives an overview of the various traffic models. The Cisco paper [12] on VoIP traffic patterns compares various traffic models for voice calls, including Erlang B and C, Extended Erlang B, Engset, Poisson, EART/EARC, Neal-Wilkerson, Crommelin, Binomial and Delay.

The UMTS Forum Report 44 [13] is built on a traffic model which distinguishes service categories (video/audio streaming, mobile gaming, etc.), device categories (smartphones, tablets, connected embedded devices, etc.) and various subscriber activity patterns. These components contribute to the overall traffic models through parameters such as traffic per service/device, device mix, upload/download direction, period of the day/week/year, and others.

Aalto et al. [14] compared various scheduling algorithms from link delay and fairness aspects, and found that scheduling algorithms in the access network have their impact on the observed traffic in the core network.

A. The composite model for the traffic generator

This section describes the creation methodology for the composite traffic model, which consists of independent, classical traffic models. The parameters of the composite model include:

- the total number of simulated subscribers in the system,
- number of subscribers taking part in each activity (i.e. number of subscribers for each traffic model),
- triggering events/probabilities to change any parameter in any of the traffic models and
- the rate at which subscribers start/stop particular activities (i.e. the change rate in the number of subscribers simulated by a particular traffic model, such as attach/detach rate).

In the composite model we assume that one subscriber is engaged in one activity at a time. That is, a simulated subscriber is used in only one traffic model at a time. In the pool of simulated subscribers, each subscriber is assigned to a given traffic model for a set period of time, in an on-off manner. Figure 3 depicts an example of how the user traffic models contribute to the composite model in time.

In order to simulate a limited number of subscribers, we found it practical to introduce an *Idle* traffic model. This model acts as the subscriber pool, which holds available users for the various activities simulated by the models.

The traffic model parameters are considered as target values only, in the context of other model parameters. The target may not be met if fewer subscribers are simulated than required by a model or the traffic generator's hardware and software resource limitations are reached, or the resources of the Network Under Test are exhausted. In a load testing scenario the latter may actually be a desired event – provided that the aim is to discover the limitations of the system.

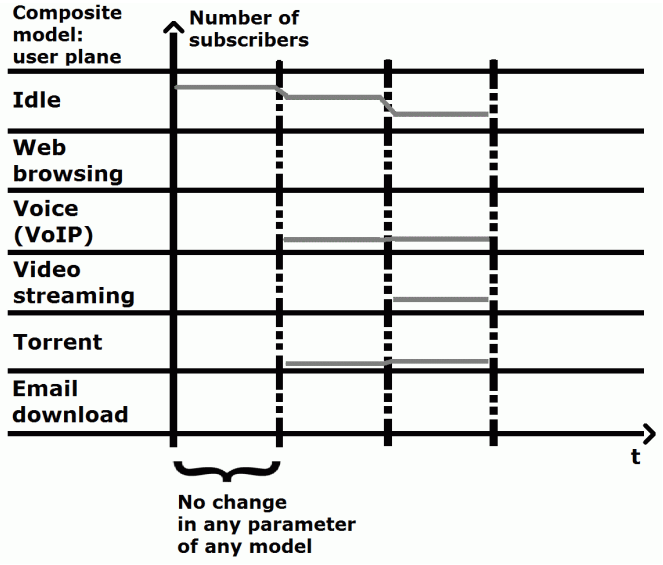

Fig. 3. Superpositions of user traffic models in time

B. Further considerations on modeling

Traffic modeling has a huge literature, and the modeling based on arrival processes, on-off behavior, long-range dependent behavior, and heavy-tailed file size distributions are researched deeply. To get the traffic generator's composite model realistic, these traffic-features should also be considered – both during the modeling and the verification phases.

Arlitt et al. published a comprehensive workload characterization study of Internet Web servers in 1997 [15] which they repeated ten years later [16]. Although traffic volume increased 30-fold over the elapsed years, they found that some key workload characteristics seem to persist over time.

Park and his colleagues [17] found that self-similarity in network traffic can arise due to the reliable transfer of files drawn from heavy-tailed distributions. Crovella et al. [18] concluded that the heavy-tailed nature of transmission and idle times is not primarily a result of protocols or user preference, but rather some basic properties of information storage and processing: both file sizes and user "think times" are strongly heavy-tailed. Shaikh et al. [19], examining the on-off model, propose a wavelet-based criterion to differentiate between the network-induced traffic gaps and user think times.

Bregni and Jmoda [20] exhaustively analyze long-range dependency and provide an accurate estimation for the Hurst parameter.

Andreev et al. [21] propose a practical traffic generation model based on the discrete-time batch Markovian process with the intent to fill the gap between the analytically complex models discussed in academic publications and the simpler, more practical models preferred by standardization bodies.

Terdik and Gyires [22] believe that it is time to reexamine the Poisson traffic assumption, because the amount of Internet traffic grows dramatically, and any irregularity of the network traffic, such as burstiness, might cancel out because of the

huge number of different multiplexed flows.

VI. Implementation

In the control plane messages, the network-, service- and user-specific parameters are filled out as required. In practice, a relatively huge part of these messages never change: many parameters in call-setup, session or mobility management are set to exactly the same value. Network- and service- specific parameters (e.g. point codes, service capabilities, expected QoS parameters) have no or limited variance; user-related parameters (e.g. endpoint identifiers, temporary codes), on the other hand, vary a lot.

From this viewpoint it is possible to build a protocol message pool in which each kind of dialog is represented by a set of message templates. In these templates there are

- parameters filled out with *fixed values*,
- variable parameters, whose values are *chosen from a range* (based on certain rules), and
- variable parameters *matching the protocol logic* (i.e. temporary identifiers, sequences, etc.).

In the user plane, individual dummy packets are generated as defined by the composite model. Actual packet lengths, and inter-arrival times are also derived from the composite model in use.

VII. Verification and refinement

The general purpose of making a model is to give predictions. In our case, after the composite model is implemented, the traffic generator device is deployed and its output is matched against the earlier-captured real-life traffic.

In the control plane, the generated traffic needs to match the captured traffic as set out by the protocol standards. The statistical properties of subscriber mobility patterns need to match that of the captured patterns.

In the user plane, the statistical properties of the generated traffic need to match those of the captured data – provided that the same subscriber activity patterns (same traffic mix) are used. Until these requirements are matching, the model parameters need refinement through further traffic analysis and fine-tuning of the parameters.

VIII. Deployment for EPC load testing

Network testing in an active way is carried out by simulating nodes that (i) send control-plane protocol messages to the Network Under Test (NUT) (ii) keep track of the status of dialogs, transactions and contexts, and (iii) handle user-plane traffic sent to/received from the NUT.

Our methodology can be used for testing various kinds of core networks, however its main strengths can be demonstrated best with the LTE Enhanced Packet Core, shown in Figure 4.

A. Nodes simulated and nodes under test

In this section we demonstrate the testing methodology through a practical example. Consider that the traffic generator simulates several eNodeBs, each handling hundreds of thousands of subscribers, who are attaching to the network, entering and leaving the simulated eNodeB areas, and generate user plane traffic. The Network Under Test is the LTE EPC. In this example the traffic generator pushes the real Mobility Management Entity (MME), Serving Gateway (SGW) and Packet Data Network Gateway (PDN-GW) to their limits. In order ot reach this aim, the traffic generator simulates the following node-types:

- eNodeB,
- Home Subscriber Server (HSS),
- Policy and Charging Rules Function (PCRF),
- external data server.

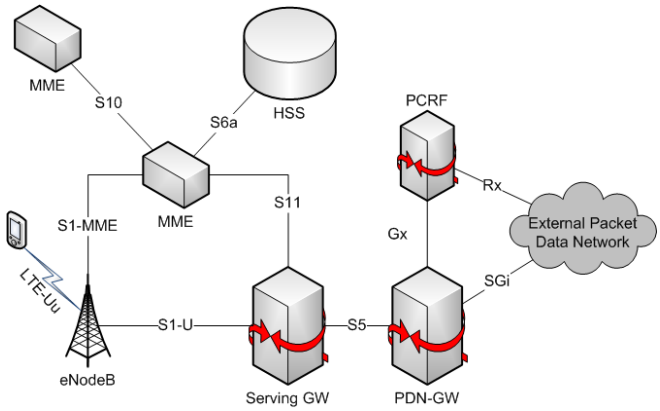

Fig. 4. LTE Architecture: main elements and interfaces

The main element to simulate is *eNodeB*: the traffic generator mimics as if subscribers were accessing the network through various eNodeBs, allowing the simulation of inter-eNodeB and inter-MME mobility as well.

The user plane traffic initiated at the eNodeBs reaches the NUT at the SGW, and leaves it via the PDN-GW towards the external packet data network to get served. In the EPC testing architecture the endpoints of this traffic must also be simulated in order to mimic the behavior of the *external data server* in a controlled manner. Both uplink and downlink traffic should show realistic patterns while deploying the models.

B. Interfaces and procedures of the traffic generator

In order to execute successful load tests, various procedures have to be conducted simultaneously. In this section we briefly list the procedures required for these tests – and leave out others (such as radio-related, or signaling security related procedures) that are out of interest from the core network testing point of view.

The *S1-MME* interface lays between the eNodeBs and the MMEs. Its procedures include (i) initial setup and configuration of eNodeB and MME communication, (ii) subscriber attach and detach, (iii) subscriber mobility management (including location updating and SGSN selection to 2G or 3G handovers as well), (iv) bearer management (related to subscriber QoS profiles and other environmental variables).

The *S1-U* interface is defined between the eNodeBs and the SGWs. Synthetic user plane traffic is injected in here:

request-type packets leave the eNodeBs, and response-type packets arrive from the direction of the external data servers through the SGW. The generated traffic patterns are the actual deployment of the user plane traffic models and their mixture, as defined by the composite model.

The *S6a* interface carries the communication between the HSS and the MMEs. Since the HSS is a simulated node in this traffic generation architecture, *S6a* procedures that are awakened by the load test must be implemented. These include procedures on (i) mobility management, (ii) subscriber-specific QoS control, (iii) session establishment support, (iv) authentication, and (v) authorization.

The PCRF is connected to the PDN-GW through the *Gx* interface. Since the PCRF is also a simulated entity in this setup, it must handle procedures related to charging and to policy control appropriately.

C. Considerations on test execution

Once the traffic generator entities are functionally able to communicate with the NUT, the load testing methodology can be put into practice. There are three, significantly different phases of these load test executions: (i) subscriber attach phase, (ii) load test with the deployment of the composite model, (iii) subscriber detach phase. While the first and last phases are self-explanatory (attaching and detaching the required number of subscribers, taking the attach and detach models into account), the middle phase is very complex.

During the middle phase, user plane traffic is generated towards the SGW and the PDN-GW from the eNodeBs and the simulated external data server, respectively. This user traffic is generated through the patterns of the composite model: various portions of the subscriber pool generate portions of traffic types, as shown in Figure 3.

IX. Conclusion

The general methodology for testing mobile core networks based on models from real-life data consists of the following procedures: *observation*, *analysis*, *model creation*, *implementation* and finally *verification* and *deployment*. The first four procedures lead towards building a composite model of control plane and user plane traffic, and result in a traffic generator. During verification and model refinement, the created composite model is compared to the network behavior to assert that the models fit to the reality. The last procedure is the actual deployment of the system: setting up the traffic generator in the network under test and identifying performance characteristics through various tests.

The presented composite traffic model is built as a superposition of control plane models (including user behavioral models, protocol state transitions, message sequences and a pool of message templates), and user plane traffic models (including application traffic models, and models for traffic mixtures).

The application of the methodology is demonstrated through an LTE EPC example, where the Network Under Test consists of the SGW, PDN-GW and the MME. In this use case, the traffic generator consists of simulated HSS, PCRF, external data server functions, as well as several eNodeBs. This environment simulates the behavior of subscribers who are attaching to the network, moving within it and generate user plane traffic.

The work leading to these results has received funding from the European Union's Seventh Frame work Programme (FP7/2007-2013) under grant agreement No. 317762.

References

[1] P. Olaszi, "Complex Load Testing of Mobile PS and CS Core," EuroNOG 2012, September 2012. [Online]. Available: http://www.data.proidea.org.pl/euronog/2edycja/materials/Peter_Olaszi-Complex_Load_Testing_of_Mobile_PS_and_CS_Core.pdf

[2] Polaris, "LTE EPC Load Tester," Product Description, August 2013. [Online]. Available: http://www.polarisnetworks.net/support-load-tester.html

[3] Shenick, "Testing LTE/4G - Evolved Packet Core with TeraVM," Shenick Solution Brief, June 2013. [Online]. Available: http://www.shenick.com/media/files/LTE_EPC_testing_with_TeraVMv061213.pdf

[4] IXIA, "IxLoad Access," Data Sheet, May 2013. [Online]. Available: http://www.ixiacom.com/pdfs/datasheets/ixload_lte_access.pdf

[5] ng4T, "NG40-EPC-S1," Data Sheet, 2013. [Online]. Available: http://www.ng4t.com/Datasheets/datasheet_NG40-EPC-S1.pdf

[6] MobileMetrics, "Torrent 6100 LTS," Product Description, 2012. [Online]. Available: http://www.mobilemetrics.net/lte-test.htm

[7] Aricent, "Total Testing for LTE," Product Description, 2013. [Online]. Available: http://www.aricent.com/pdf/Aricent_Solution_Brief_LTE_Testing.pdf

[8] EXFO, "Nethawk EAST EPC for Evolved Packet-Core Testing," Product Description, 2011.

[9] NS-3, "ns-3 Discrete-event Network Simulator," 2013. [Online]. Available: \url{http://www.nsnam.org/}

[10] M. Olsson, S. Sultana, S. Rommer, L. Frid, and C. Mulligan, *SAE and the Evolved Packet Core*. Oxford, UK: Academic Press, 2009.

[11] B. Chandrasekaran, "Survey of Network Traffic Models," Part of Raj Jain's Computer Systems Analysis Lectures, 2006. [Online]. Available: http://www1.cse.wustl.edu/~jain/cse567-06/ftp/traffic_models3.pdf

[12] Cisco, "Traffic Analysis for Voice over IP," White paper. [Online]. Available: http://www.cisco.com/en/US/docs/ios/solutions_docs/voip_solutions/TA_ISD.pdf

[13] IDATE, "Mobile traffic forecasts 2010–2020 report," UMTS Forum, Report 44, Jan. 2011. [Online]. Available: http://www.umts-forum.org/component/option,com_docman/task,doc_download/gid,2537/Itemid,213/

[14] S. Aalto and P. Lassila, "Impact of size-based scheduling on flow level performance in wireless downlink data channels," in *Proceedings of the 20th International Teletraffic Conference*, 2007.

[15] M. F. Arlitt and C. L. Williamson, "Internet Web Servers: Workload Characterization and Performance Implications," *IEEE/ACM Transactions on Networking*, vol. 5, no. 5, pp. 631–645, October 1997.

[16] A. Williams, M. Arlitt, C. Williamson, and K. Barker, *Web Workload Characterization: Ten Years Later*. Springer Science, 2005.

[17] K. Park, G. Kim, and M. Crovella, "On the Relationship Between File Sizes, Transport Protocols, and Self-Similar Network Traffic," in *Proceedings of ICNP '96*. IEEE Computer Society, 1996.

[18] M. E. Crovella and A. Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 835–846, December 1997.

[19] J. Shaikh, M. Fiedler, P. Arlos, and D. Collange, "Modeling and Analysis of Web Usage and Experience Based on Link-level Measurements," in *Proceedings of ITC '12*, 2012.

[20] S. Bregni and L. Jmoda, "Accurate Estimation of the Hurst Parameter of Long-Range Dependent Traffic Using Modified Allan and Hadamard Variances," *IEEE Transactions on Communications*, vol. 56, no. 11, pp. 1900–1906, November 2008.

[21] S. Andreev, A. Anisimov, Y. Koucheryavy, and A. Turlikov, "Practical Traffic Generation Model for Wireless Networks," in *4th ERCIM eMobility Workshop*, Luleå, Sweden, May 2010.

[22] G. Terdik and T. Gyires, "Levy Flights and Fractal Modeling of Internet Traffic," *IEEE/ACM Trans. Netw.*, vol. 17, no. 1, pp. 120–129, Feb. 2009.