Secure and Efficient Validation of Data Traffic Flows in Fixed and Mobile Networks

Julius Mueller
Technical University Berlin
Berlin, Germany
Julius.Mueller@tu-berlin.de

Yahya Al-Hazmi
Technical University Berlin
Berlin, Germany
yahya.al-hazmi@tu-berlin.de

Mohammad Fal Sadikin
Technical University Berlin
Berlin, Germany
mohammad.falsadikin@tu-berlin.de

Dragos Vingarzan
Fraunhofer FOKUS Institute
Berlin, Germany
Dragos.Vingarzan@fokus.fraunhofer.de

Thomas Magedanz
Fraunhofer FOKUS Institute
Berlin, Germany
Thomas.Magedanz@fokus.fraunhofer.de

ABSTRACT

Services residing within the application layer in today's fixed and mobile Next-Generation-Networks (NGN) environments are abstracting from the underlying network layer and therefore assuming pure IP connectivity without taking in-network services as Quality-of-Service (QoS) guarantees, routing decisions, etc. into consideration. Real-time services (security or multimedia) require reliable end-to-end (E2E) transport of packet data flows starting at the device going over fixed and mobile access and core networks into autonomous networks towards a specific service.

End-to-end monitoring provides QoS information on the full data path, but analyzing all or most parts of one packet data flow through fragmented heterogeneous network domains is more complex and requires distributed monitoring techniques in each network compartment. This paper presents an efficient packet tracking algorithm using distributed hash based sampling technique. First the concept of packet tracking is presented followed by an overview of a prototype implementation. The paper concludes with a validation scenario of packet tracking applied on the 3GPP Evolved Packet Core exemplarily.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Measurement, Performance, Verification

PM2HW2N'12, October 21–22, 2012, Paphos, Cyprus.

Keywords

Evolved Packet Core; Packet Tracking; SLA Validation; Mesh Security; Monitoring

1. INTRODUCTION

The Internet consists of multiple heterogeneous networks such as access- and core-networks as well as manifold autonomous systems and backbones. Nearly each system is controlled independently either by its network operator or independently in case of ad-hoc or mesh networks. Distinct and isolated network domains are connected over Border-Gateway-Controller (BGC) [18] which interconnect networks of distinct operator domains and ensure Service Level Agreements (SLA) in between them. SLA's are used to define a certain performance or guaranteed delivery time of packets exchanged between pairwise two network domains.

Services residing within the application layer in today's fixed and mobile network environments are abstracting from the underlying network layer and therefore assuming pure IP connectivity without taking in-network services as Quality-of-Service (QoS) guarantees, routing decisions, security and mobility into consideration. These services assuming best-effort packet transport without demanding specific QoS, routing schemes (shortest, cheapest or most reliable path, etc.).

Nowadays the Internet plays an important role in critical domains such as finance, health, security, entertainment and military where real-time services with guaranteed resources become necessary. In comparison to best effort data transport, real-time services require reliable transport of packet data flows starting at the device going through the access and core networks into autonomous networks towards a service.

End-to-End monitoring provides QoS information on the full data path, but analyzing all or most parts of one packet data flow through fragmented heterogeneous network domains is more complex and requires distributed monitoring techniques. In order to optimize the access and core network packet data transport, E2E monitoring does only provide an overall performance indicator, but doesn't expose network delay information insides of per hop behavior through the network. In particular these measurements are required to optimize individual network parts through the identification

of loops, significant delays or packet loss due to overload situations.

This paper presents an efficient packet tracking algorithm using distributed hash based sampling technique applyed on a mobile broadband core network. Distributed probes measure packet data flows in a single network compartment, a network domain or multiple independent operator networks in parallel by reporting measurements towards a centralized aggregation point. In our case, one probe is running on each network element and exposes measurements for further studies through the network provider.

In general, packet tracking enables the following key functionalities for network operator and service provider:

- Total End-to-End as well as individual Network-to-Network SLA Validation (telecommunication access and core networks)
- Loop detection in mesh and ad-hoc networks to improve effective packet routing within a network
- Per Hop real-time link quality measurements and security constraints validations
- Flexible measurement granularity adjustment from single packet up to flows
- Handover delay measurements between networks and influences on the data path

The paper is structured as followed. The validation approach is presented first in section 2 contains an introduction into the software tool Packet Tracking and OpenEPC, followed by the description of the validation scenario test environment. Another section presents implementation details of the prototypical demonstrator realized by Technical University of Berlin and Fraunhofer FOKUS. Finally the validation section applies the concept of packet tracking on the 3GPP Evolved Packet Core exemplary. Packet flow measurement related work is presented in section 5. The paper concludes with chapter 6 outlining the future work with packet tracking and its extensions.

2. VALIDATION APPROACH

The strong application demands on high quality support from the network (e.g. low delays, packet loss) leads to the need for monitoring the fulfillment of this demand. This means that there is a need for understanding the properties and the behavior and the network itself. This section represents an efficient approach for monitoring the network traffic flows. The packet tracking as a passive monitoring approach is used for this purpose. The multi-hop packet tracking allows tracking packets on the network and providing various types of packet related information.

Thus, it allows an End-to-End monitoring that provides QoS information along the entire data path. The OpenEPC toolkit is used in this work as an applicable environment for the mobile networks as outlined in Figure 1. It connects heterogeneous accesses networks such as 3GPP LTE, UMTS and GPRS and non-3GPP IEEE WiFi and WiMAX, offering a platform for research in the area of mobile networks.

The following sections describe the evolved packet core in detail, which reflects the basic design of the OpenEPC platform, and the packet tracking monitoring tool.

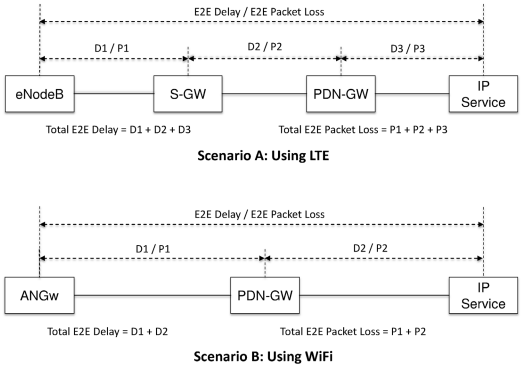

Figure 1: Packet Tracking on Evolved Packet Core

2.1 Evolved Packet Core

3GPP standardizes with the All-IP Evolved Packet Core a counterpart to the radio access network technology Long-Term-Evolution (LTE) and its evolution LTE-Advanced (LTE-A). The EPC provides network mobility between heterogeneous 3GPP [4] and non-3GPP [3] access network technologies, provides QoS through Policy and Charging Control [2], IP connectivity and security [1].

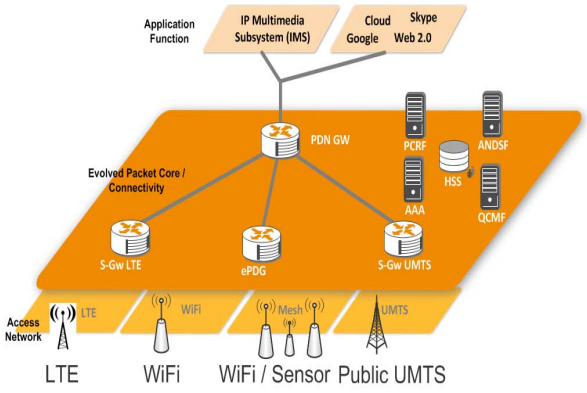

Figure 2: OpenEPC Layered Architecture

The OpenEPC toolkit developed by Fraunhofer FOKUS and TU Berlin which adopts large parts of the 3GPP EPC standards and represents an all-IP architecture permitting the research and the development of different wireless applications into a real wireless environment. For this matter, OpenEPC includes the mechanisms of connecting end-user devices to different access networks and supporting communication interfaces to the various service platforms, offering the most suitable connection for the various services in terms of access network selection and of resource adaptation to the current network conditions [8]. By its modular structure depicted in Figure 2, the OpenEPC toolkit enables the fast deployment of test network architectures in various combinations coming to meet the requests from the research community.

2.2 Packet Tracking

Today we face the fact that Internet strives for increasingly sophisticated traffic measurement technique for efficient managing and providing QoS guarantee, references for traffic engineering, security validation, as well as for research and development purpose. Since the emerging of traffic engineering various measurement tools and techniques have been introduced reaching from active measurements like ICMP ping and traceroute to passive measurements like pcap and flow measurement.

However, most of such tools and techniques introduce various disadvantaged and limitations reaching from the active technique like significant processing overhead and lower accuracy issue affected by injecting additional probe traffic into the network to the passive technique like data flood generated by large amount of measurement traffic. In this subchapter, we introduce our solution in order to tackle such various disadvantages and limitations by providing passive measurement tool named Packet Tracking [21] with efficient hash-based packet selection and agile transport method for exporting flow information and selected packet observation data called IPFIX [7] protocol. Indeed, we first discuss our motivation to provide a specific measurement tool followed by the concept of Packet Tracking and its implementation as well.

2.2.1 Motivation

Measurement techniques can be classified into two groups, active and passive measurement. Both techniques have specific advantages and disadvantages. For instance, active measurement requires injection of additional traffic into the network in order to measure the conditions of the network. Such technique is efficient as there is no need to have fully access to all the network resources and no measurement infrastructure requires as well. Such technique is less accurate as the additional traffic injected to the network changes the network property and increase overhead affecting the increase of delay and processing time. While passive technique does not require additional traffic, just observe the existing traffic by deploying the existing probe installed into network devices to enable such observation. Basically passive technique is well recognized as high precision measurements as such technique does not change the network properties as there is no injection of artificial traffic probe into the network as well as more efficient in term of reducing the overhead problem. However, this technique introduces tremendous problem in analyzing the network characteristics especially the large amount of data traffic coming from the observed network make it inefficient and very difficult to observe the network.

Many filtering and sampling techniques for passive measurements have been introduced in order to tackle such large problem with amount of data traffic. In general filtering and sampling techniques are conducted by two approaches, which are: First, flow monitoring approach, it observes the network based on the flow characteristics which are usually divided into five tuples (e.g. Source IP address, Destination IP address, Source Port, Destination Port, and Protocol). This flow technique introduces various challenges examples here are the need of additional maintenance and update for the flow entries when the packet comes in as well as required large storage and longer processing affected by the longer flow cache. Second, packet monitoring approach, it samples packets based on packet characteristics or variety of packet statistics. Such technique significantly reduces the overhead and calculation effort as well as provides better accuracy by selecting only subset of packets at every measurement point to estimate the real traffic characteristic. Such technique also introduces specific limitation in the way of the packet is sampled. Example here is the possibility of eradicating small flow and losing such flow records.

Packet Tracking provides a possible solution to enhance passive measurement limitations through two mechanisms: First, by deploying hash-based packet sampling which enables to selection of subset of packets to be measured in order to reduce the amount of data from observed traffic and significantly improve the accuracy of network characteristic estimation. Second, by deploying IPFIX for efficient, flexible, reliable and secure transport process in order to keep the overhead low and satisfy the measurement needs as well.

Packet Tracking can be used for multiple purposes. Among these:

- One Way Delay (OWD) measurement: provides a delay between two synchronized measurement points. Packet Tracking enables more efficient and accurate OWD measurement as a passive technique, which does not introduce additional traffic, and also no changing in the network property as well.

- Traceback system: is a critical security mechanism in global Internet services to against serious problem like Distributed Denial of Services (DDoS) attacks. Such mechanism enables to trace back the source of malicious traffic and stop the attack from the source point, thus the entire of service deployment can be protected. Packet Tracking provides the needs of packet traceback by tracing hop by hop the origin of the malicious packet. By deploying hash-based packet selection, Packet Tracking efficiently trace back the subset of malicious packet in order to keep the overhead low as well as provide configurable traceback system based on the risk probability in particular application domain.

2.2.2 Hash-based Packet Sampling

The limitation of packet sampling is not suitable for multipoint measurement as there is no mechanism to coordinate the selected packet in order to consistently select the sampling and precisely estimate the network characteristic. Hash-based packet sampling technique comes up with tangible solution which enables to coordinate the selected subset of packets by generating Packet ID and timestamp. Packet ID generation is an adjacent field to hash-based selection in multiple observation points. Each packet passing an observation point is assigned a packet ID and timestamp that are exported to a multipoint collector. A hash function H maps parts of the packet content C, that are invariant between observation points, onto a hash range R. The packet is selected if H(C) is an element of S, which is a subset of R called the hash selection range [23]. To this end, the consistently selected subset of packets can be traced and correlated from multiple observation points. Furthermore, OWD is calculated based on timestamp difference generated at every observed node. Figure 3 illustrates the correlation of packets at different observation points based on packet ID.

By deploying Hash-based Packet Selection [23], Packet

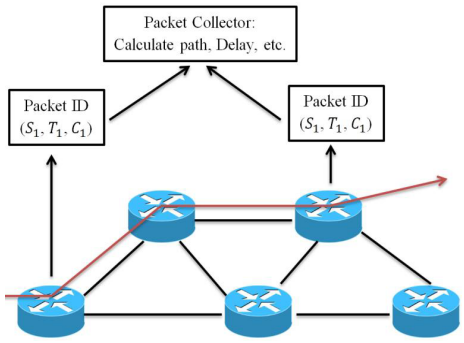

Figure 3: Packet ID Generation. S = sequence, T = arrival time, C= content (header + payload)

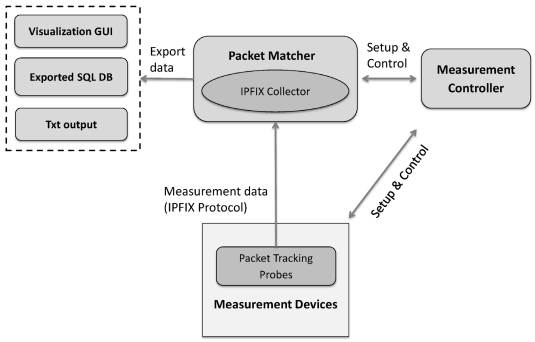

Figure 4: Packet Tracking Overview

Tracking can efficiently select a subset of packets, generate packet ID and timestamp, in order to coordinate the sampling in large scale system. Thus the network condition can be precisely estimated as well. Such innovation enables to overcome the problem with most of sampling techniques which could probably eliminate the packet in multi-point measurements. Packet Tracking can accurately measure the traffic characteristic like OWD by consistent selecting and time stamping the subset of packets in such multi-point measurements.

2.2.3 *Packet Tracking Architecture*

The architecture Packet Tracking consists of four main components as illustrated in Figure 4:

Packet Tracking Probe.
is lightweight measurement daemon provided by impd4e deployed in the network at every observed node. It supports Hash-based Packet Selection for consistent selection the subset of sampling. The impd4e captures packet by using libpcap, uses libipfix as an implementation for IPFIX protocol for reporting measurement results, and libev for its internal event handling.

Packet Matcher.
is an IPFIX collector taking measurement data from impd4e probes and delivers packet path data as an output (e.g. to netview, log files, aggregator, etc.). Packet Matcher uses ipfix4java, a pure IPFIX java implementation for IPFIX collectors.

Visualization GUI (Netview).
receives and visualizes measurement data by providing a graphical output of packet tracking paths to facilitate analysis of processed data. Furthermore, it also has several extensions like aggregator, which takes packet delay data and computes (e.g. minimal, maximal, and average delays) and stores pre-aggregated delay results in local database.

Measurement Controller.
is to coordinate setup and control of measurement parameters beween the probes running on the measurement devices and the Packet Matcher.

Packet Tracking is flexible measurement tool which can be deployed in any network domain by installing probe (impd4e) at every observed measurement node. In this case the matcher which is typically installed in user's local machine must be publicly reachable for the installed measurements probe at every node. Figure 4 shows a high level overview on the interaction between the packet tracking components.

As depicted in Figure 4, Packet Tracking provides multiple options for exporting measurement data that can be then deployed for various purposes. For instance, the visualization GUI that shows per hop delay and path information. Exported SQL database for querying delay results is also an example of output option. Measurement data can also be exported in text files.

3. DEMONSTRATOR SETUP

This paper focuses on the validation of data flows in fixed and mobile networks and for this purpose the EPC is used. The validation study is carried out by monitoring the important QoS parameters, in particular, one-way delay and packet loss. For a demonstration, the OpenEPC platform is used as a prototype implementation of the EPC. An end-user is able to connect to any of the available networks provided by the OpenEPC and is downloading data from an Internet server. Packet tracking monitoring probes are deployed in multiple components along the path between the end-user and the Internet server. In particular, probes are deployed on all OpenEPC components that are show Fig.1. Hop-by-hop QoS monitoring information is collected from these probes.

Two setup scenarios are discussed in this paper. They differ than each other on the network technology used by the end-user as an access network among the available ones provided by the OpenEPC. In the first scenario the LTE is used while the WiFi is used in the second scenario. Figure 1 gives an overview on the individual OpenEPC components between the end-user and the IP service in both scenarios. Furthermore, it shows where multiple delay and packet loss measurements can be conducted. For having an efficient and effective validation, measurements are taken in two phases: end-to-end measurements, and one-hope measurements between the individual nodes along the entire path.

The environment setup consists of four Eee PC boxes representing the EPC components, one Eee PC notebook as an eNodeB in the case of LTE scenario, and one Eee PC notebook as a client. All components run Ubuntu Linux, with

dual core Intel Atom @ 1.60GHz N270 processor, 100MB/s network cards, and 2GB memory. A DLINK fast Ethernet switch Adapter USB 2.0 DUB-E100 provides additional Ethernet interfaces to the Eee PC boxes. A netgear 8 port 100/1000Mbps switch GS608v3 interconnects all scenario components.

It is a critical mechanism to synchronize sampling and perform high precision measurement. Therefore, Network Time Protocol (NTP) is deployed at each EPC component to realize time synchronization.

4. EVALUATION AND CONCLUSION

This section presents the results of the experimentations that have been conducted in the two scenarios involving LTE and WiFi as outlined in section 3. We present the measurements in figures, analyze and discuss them within this section.

OWD measurements have been conducted while analyzing TCP traffic on each component within the EPC architecture between the eNodeB and the IP service. In particular a file has been downloaded from a server (IP Service) through the EPC towards a UE/device. The setup and the involved network elements are in an isolated network and are therefore have network resources exclusively.

We are focusing in this evaluation only on the core network part of the Evolved Packet System (EPS) and not on the radio network part (LTE), because the latter one utilizes the shared medium radio and negatively effects the measurement due to packet loss and additional delays.

Two measurement scenarios on downloading a TCP data stream are evaluated in the scope of this paper; each of them has its origin at the IP service residing in the operator network, going through the evolved packet core through the related EPC components and finally terminating in access network specific radio controller (ANGw or eNodeB).

The measurements were captured between the IP services and the specific radio access network controller, both involving components on the path and excluding the terminal/UE from the measurements, which would be connected over a wireless link.

In detail we have the following chain of components involved in data traffic routing between the service and the UE:

- LTE: IP Service, PDNGW, SWG, eNodeB
- WiFi/802.11: IP Service, PDNGW, ANGw

In both scenarios, the End-End delay (within the core network) and delays between the individual EPC components are measured. The motivation behind this has twofold objective:

First, it's aimed to investigate whether the packet tracking probes are slowing down the routing of the packets through gateways, on which the probes are deployed.

Second, for improvement purposes of the functionality of the EPC, to figure out which of the individual EPC components causes the highest delay in the End-to-End packet routing. The End-to-End OWD is performed by deploying packet tracking probes only on the end-components of the EPC depending on the used access technology. Probes are deployed in the two discussed scenarios on the IP Service component, and on the eNodeB in the case of LTE, while on the ANGw in the case of WiFi.

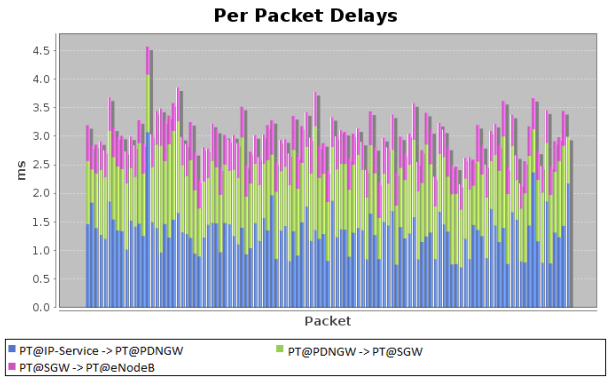

Figure 5: Per Components Breakdown End-to-End Delay between IP-Service and eNodeB

In the case of using LTE scenario, the End-to-End delay is shown in Figure 5, where the x-axis represents the number of packets that are monitored (around 300 packets in the conducted experiments), while the y-axis indicates the delay in milliseconds. The average End-to-End delay is around 2.74 milliseconds. Delays between the individual EPC components are also measured and illustrated in Figure 6.

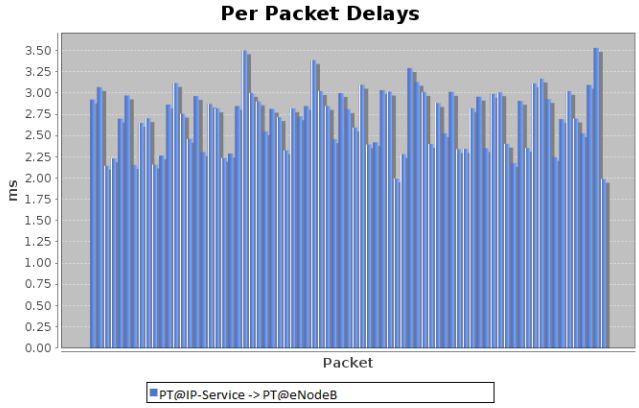

Figure 6: Total End-to-End Delay between IP-Service and eNodeB

The blue graph (at the bottom) represents the delay between the IP service component and the PDNGW with an average value of 1.8 milliseconds. The green graph (at the middle) indicates the delay between the PDNGW and the SGW with an average value of 1.23 milliseconds, while the delay between the SGW and the eNodeB is represented by the pink graph (at the top) with an average of 0.54 milliseconds.

The difference between the average End-to-End delay (2.74 milliseconds) and the sum of the individual one-hop average delays (around 2.86 milliseconds) is 0.12 milliseconds. It indicates that the probes do not significantly disturb or change the network property as it is well known in passive measurement tool characteristic.

In case of using WiFi 802.11, the End-to-End delay is shown in Figure 7, with an average value of around 2.22 milliseconds. Delays between the individual components are also measured and to be seen in Figure 8. The blue graph (at the bottom) represents the delay between the IP service

component and the PDNGW with an average value of 1.46 milliseconds. The green graph (at the top) indicates the delay between the PDNGW and the ANGw with an average value of 0.9 milliseconds.

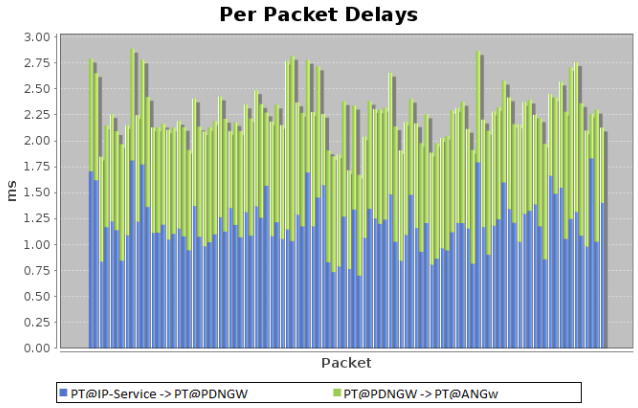

Figure 7: Per Components Breakdown End-to-End Delay between IP-Service and ANGw

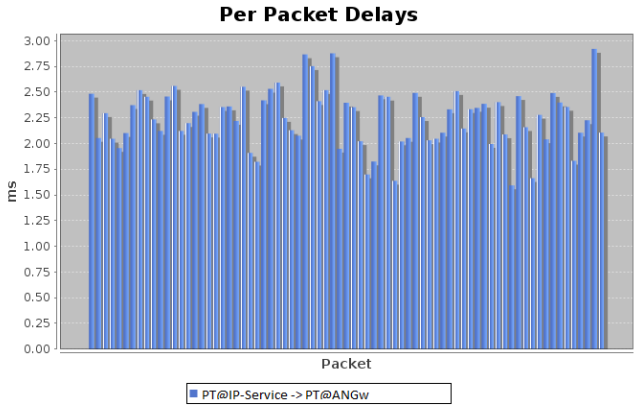

Figure 8: Total End-to-End Delay between IP-Service and ANGw

The difference between the average End-to-End delay (2.22 milliseconds) and the sum of the individual one-hop average delays (around 2.36 milliseconds) is 0.14 milliseconds.

The differences between the two paths for LTE and WiFi regarding the delay are caused by the different encapsulation and packet forwarding strategy. While IP packets for the LTE technology are encapsulated in 3GPP specified GRPS-Tunneling-Protocol user plane (GTP-U) [5], IP packets for WiFi technology are encapsulated again in IP (IP-in-IP) in order to enable layer 3 routing within the EPC.

Since non-3GPP networks such as WiFi may be secured with Authentication, Authorization and Accounting optionally, additional IPsec [12] tunneling may influence the network performance negatively. The presented scenario does not include Ipsec, which is left for future work.

5. RELATED WORK

The high growth in the number of applications, devices and users in the Internet and telecommunication sectors has led to a very complex traffic and dynamically changeable usage pattern. As a matter of fact, traffic monitoring and measurements are becoming more and more important within the context of future networking since management requirements will be increased. Many research activities and tools in the area of traffic monitoring and measurements have been introduced to obtain various metrics (e.g. OWD, Round Trip Time (RTT), packet loss, etc.).

HADES (HADES Active Delay Evaluation System) [10] is an active monitoring tool for providing QoS measurements, such as one-way delay, IP packet delay variation and packet loss. For multicast traffic measurement, authors in [17] used dbeacon which is a multicast beacon written in C++ with the main purpose of monitoring and collecting QoS related statistics such as loss, delay and jitter between beacons. It supports both IPv4 and IPv6 multicast, collecting information using both any-source-multicast and source-specific-multicast. Dbeacon was used in [17] to gather multicast traffic loss, delay and jitters between some nodes that are running IPTV service delivery. The software platform D-ITG [6] enables traffic generation and actively measure the traffic in order to perform several packet level metrics like one-way-delay, RTT, packet loss, jitter, and throughput. SmokePing [11] performs and collects latency, latency distribution, and packet loss measurements by deploying RRD tool for maintaining a long term data-store and displaying such measurement metrics. UDPmon [22] is a UDP throughput measurement tool which enables to perform packet loss and RTT measurements. Furthermore, there are many other works that performed measurements in order to investigate IP performance metrics [20] [14] [13] [9] [19].

However, most of these tools and works incorporate active measurement techniques with particular issue like the possible slowing down the routing or changing network property affected by injecting artificial traffic into the network. In this paper, the packet tracking as a passive measurement tool is used. It uses an efficient and flexible protocol for transporting measurement data, called IPFIX [7]. Packet Tracking has specific sampling mechanism called hash based packet selection. Sampling mechanism is the selection of a subset of packets at multiple observation points. It allows a synchronized selection at all observation points. Thus, the large amount of network measurement data can be reduced in order to effectively observe the traffic property at each observation point.

6. OUTLOOK AND FUTURE WORK

The Multi-Hop-Packet Tracking is a flexible measurement tool facilitating efficient and secure inspection of detailed hop-by-hop metrics (e.g. delay, packet loss, service chains, QoS, etc.) by deploying Hash-based sampling and IPFIX as efficient and flexible export of measurement results. It is used in this paper for validating the data flows in fixed and mobile networks through monitoring End-to-End as well as per hop QoS measurements, in particular one-way-delay.

As a future work Packet Tracking can be further developed in order to improve its capabilities. For instance, here is to deploy flow based sampling or filtering for further deployment in various monitoring and measurement applications (e.g. Deep Packet Inspection, traffic classification, TCP connection tracking, flow counting, traffic anonymisation, etc.). Packet Tracking currently enables packet based measurement with packet sampling technique that helps on reduc-

ing the huge calculations and processing effort. However, flow based measurement with flow based sampling will be the future work.

Indeed, flow based measurement like Deep Packet Inspection is well recognized as the most effective way to inspect and detect malicious attacks by separating good and bad packets in the network. However such technique introduces tremendous disadvantages reaching from the huge effort in the network like significantly increase overhead, highly memory and processing power demands, to costly requirement like a lot of resource consumptions. Furthermore the immense number of data traffic coming from every measurement point generates such a big calculation effort and requires large storage and the longer processing affecting the more and more matter of such networking effort and operational cost as well.

The Multi-Hop Packet Tracking is a promising tool to tackle the above challenges by deploying flow selection techniques with hash-based sampling approach which only analyses a subset of flows for each observation node. Thus, such overhead and costly challenges can be significantly solved effectively.

7. ACKNOWLEDGMENTS

This work was undertaken in the context of the EU FP7 FIRE project OpenLab [15]. Parts of this work were undertaken within the scope of the G-LAB DEEP project [16] funded by the German Federal Ministry of Education and Research as part of the G-Lab project. The authors would like to thank Christian Henke for his work with packet tracking.

8. REFERENCES

[1] 3GPP. Evolved Packet System (EPS); 3GPP EPS AAA interfaces. TS 29.273, 3rd Generation Partnership Project (3GPP), Sept. 2008.

[2] 3GPP. Policy and charging control architecture. TS 23.203, 3rd Generation Partnership Project (3GPP), June 2010.

[3] 3GPP. Architecture enhancements for non-3GPP accesses. TS 23.402, 3rd Generation Partnership Project (3GPP), Mar. 2012.

[4] 3GPP. General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access. TS 23.401, 3rd Generation Partnership Project (3GPP), Mar. 2012.

[5] 3GPP. General Packet Radio Service (GPRS); GPRS Tunnelling Protocol (GTP) across the Gn and Gp interface. TS 29.060, 3rd Generation Partnership Project (3GPP), Mar. 2012.

[6] A. Botta, A. Dainotti, and A. Pescapè. Multi-protocol and multi-platform traffic generation and measurement. *INFOCOM 2007 DEMO Session*, 2007.

[7] B. Claise, S. Bryant, G. Sadasivan, S. Leinen, T. Dietz, and B. Trammell. Specification of the ip flow information export (ipfix) protocol for the exchange of ip traffic flow information (rfc 5101). Technical report, Technical report, The Internet Engineering Task Force (IETF)(January 2008), 2008.

[8] M. Corici, F. Gouveia, T. Magedanz, and D. Vingarzan. Openepc: A technical infrastructure for early prototyping of ngmn testbeds. In *Testbeds and Research Infrastructures. Development of Networks and Communities*, volume 46, pages 166–175. 2011.

[9] A. Hanemann, J. Boote, E. Boyd, J. Durand, L. Kudarimoti, R. Łapacz, D. Swany, S. Trocha, and J. Zurawski. Perfsonar: A service oriented architecture for multi-domain network monitoring. *Service-Oriented Computing-ICSOC 2005*, pages 241–254, 2005.

[10] P. Holleczek, R. Karch, R. Kleineisel, S. Kraft, J. Reinwand, and V. Venus. Statistical characteristics of active ip one way delay measurements. In *Networking and Services, 2006. ICNS'06. International conference on*, pages 1–1. IEEE, 2006.

[11] N. Jeliazkova, L. Iliev, and V. Jeliazkov. Uperfsonarui-a standalone graphical user interface for querying perfsonar services. In *Modern Computing, 2006. JVA'06. IEEE John Vincent Atanasoff 2006 International Symposium on*, pages 77–81. IEEE, 2006.

[12] S. Kent and K. Seo. Security Architecture for the Internet Protocol. RFC 4301, Internet Engineering Task Force (IETF), 2005.

[13] E. Magana, D. Morato, M. Izal, J. Aracil, F. Naranjo, F. Astiz, U. Alonso, I. Csabai, P. Haga, G. Simon, et al. The european traffic observatory measurement infrastructure (etomic). In *IP Operations and Management, 2004. Proceedings IEEE Workshop on*, pages 165–169. IEEE, 2004.

[14] M. Mellia, R. Lo Cigno, and F. Neri. Measuring ip and tcp behavior on edge nodes with tstat. *Computer Networks*, 47(1):1–21, 2005.

[15] ICT Project. Openloab. Website, 2012. Available online at http://www.ict-openlab.eu/, last visited on Mai 4th 2012.

[16] B. F. Project. G-lab deep. Website, 2012. Available online at www.g-lab-deep.de, last visited on Mai 4th 2012.

[17] A. Qadir, M. Arefin, and H. Sandström. Reliable iptv service delivery using pim-ssm routing. *Journal of Scientific Research*, 1(3):495–507, 2009.

[18] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP-4). Rfc, Internet Engineering Task Force (IETF), 2006.

[19] G. Shen, Y. Wang, Y. Xiong, B. Zhao, and Z. Zhang. Hptp: Relieving the tension between isps and p2p. In *Proc of IPTPS*, 2007.

[20] B. Tierney, J. Metzger, J. Boote, E. Boyd, A. Brown, R. Carlson, M. Zekauskas, J. Zurawski, M. Swany, and M. Grigoriev. perfsonar: Instantiating a global network measurement framework. *SOSP Wksp. Real Overlays and Distrib. Sys*, 2009.

[21] TUB and FOKUS. Packet tracking. Website, 2012. Available online at www.fokus.fraunhofer.de/go/track, last visited on Mai 8th 2012.

[22] C. Valliyammai and S. Selvi. Relational network monitoring system for grid performance optimization. In *Advanced Computing and Communications, 2008. ADCOM 2008. 16th International Conference on*, pages 170–173. IEEE, 2008.

[23] T. Zseby, M. Molina, N. Duffield, S. Niccolini, and F. Raspall. Sampling and filtering techniques for ip packet selection. *draft-ietf-psamp-sample-tech-07 (work in progress)*, 2005.