Assessment of IxLoad in an MPG Environment

Yue Peng / Zhiqiang Tang

Abstract

University Name
The Department of Information Technology and Media (ITM)

Author: Yue Peng / Zhiqiang Tang
E-mail address: kluvcc@gmail.com, tang-zhiqiang@hotmail.com

Study programme: Computer Engineering, 120

Examiner: Tingting.Zhang@miun.se, per.osterstrom@ericsson.com
Tutors: Rohit Guliani, Ericsson, rohit.guliani@ericsson.com
Scope: 29079 words inclusive of appendices
Date: 2012-09-07

Assessment of IxLoad in an MPG Environment

Yue Peng / Zhiqiang Tang
Abstract

Long Term Evolution (LTE) is the latest mobile network technology published by the Third Generation Partnership Project (3GPP). It might become a dominant technology for the next generation and it is attracting a great deal of attention from the top global corporations.

IxLoad is a real-world traffic emulator, designed by the test solution provider giant Ixia. Mobile Packet Gateway (MPG) has been developed by Ericsson and is supporting the Global System for Mobile Communication (GSM), Wideband Code Division Multiple Access (WCDMA), LTE traffic. In this thesis, MPG is utilized to assess the capacity and LTE functionality of IxLoad. Capacity estimation will verify the maximum simulated users that can be supported by IxLoad and will test the maximum throughput IxLoad can achieve with a particular number of simulated users, under conditions involving a particular application scenario such as browsing HTTP. In addition to Session Management some other features such as Track Area Update and Handover, Busy Hour Functionality, Deep Packet Inspection, Multiple Access Point Names (APNs) and Dynamic Quality of Service Enforcement are also covered in the functionality assessment.

Moreover, this thesis gives a brief introduction to Evolved Packet System (EPS), Evolved Packet Core (EPC), and to the functionality of MPG in addition to the role of MPG in EPS. Meanwhile the newest features of IxLoad are also presented in this document. Finally, as the outcome of this thesis, several suggestions are proposed in relation to improvements for IxLoad and MPG.

Keywords: MPG, SGW, PGW, IxLoad
Acknowledgements

Firstly we would like to thank all of the personnel at GGSN-MPG division of Gothenburg Ericsson, especially to our supervisor Rohit Guliani, examiner Per Österström, Fredrik Larsson, Azadeh Hosseinzad Amirkhizi, Johan Köpman and Erik Mannergren.

In addition, we appreciate all the support from Ixia, especially that provided by Daniel Musat, Radu Stefan and Jan Adersson.

We also wish to thank all our friends in Sweden.

Last but not least, we would like to thank our wonderful families, for all their enduring support and always believing in us.

Yue Peng / Zhiqiang Tang
Gothenburg Sweden, 2011
## Table of Contents

Abstract ............................................................................................................. ii

Acknowledgements ....................................................................................... iii

Table of Contents ........................................................................................... iv

Terminology ................................................................................................... vii

1 Introduction.......................................................................................... 12
   1.1 Background and Problem Motivation .................................... 12
   1.2 Overall Aim ........................................................................ 12
   1.3 Scope ............................................................................... 13
   1.4 Concrete and Verifiable Goals ........................................... 13
   1.5 Outline ............................................................................. 13
   1.6 Contributions ..................................................................... 14

2 Evolved Packet Core Network Background ................................... 15
   2.1 UMTS Packet Core Network ............................................ 15
   2.2 LTE Packet Core Network ............................................. 16
   2.3 Protocol ........................................................................... 21
      2.3.1 GTP ........................................................................ 21
      2.3.2 Diameter ................................................................... 25

3 Mobile Packet Gateway ..................................................................... 30
   3.1 Session Management ................................................................ 30
      3.1.1 ECM State Change .................................................. 30
      3.1.2 Default Bearer Activation and Deactivation ............... 31
      3.1.3 Dedicated Bearer Activation .................................... 32
      3.1.4 Tracking Area Update ............................................. 34
      3.1.5 Handover ................................................................... 35
   3.2 Quality of Service ................................................................ 38
      3.2.1 PCEF Function ......................................................... 40
      3.2.2 QoS Handling ......................................................... 40
   3.3 Deep Packet Inspection ....................................................... 41

4 IxLoad ..................................................................................................... 43
   4.1 IxLoad Overview .................................................................. 43
# Table of Contents

4.2  IxLoad eGTP Stack ................................................................. 47  
4.3  Advanced Subscriber Model .................................................. 50  
4.4  Dynamic QoS Enforcement ..................................................... 53  
4.5  Multiple APNs ....................................................................... 55  

5  Test Environment ..................................................................... 57  
5.1  Back to Back ......................................................................... 57  
5.2  MPG Test Environment .......................................................... 57  

6  Test Scenarios .......................................................................... 59  
6.1  Capacity Evaluation ............................................................... 59  
6.1.1  GTP-C Transaction Rate ................................................... 59  
6.1.2  Max user with Different L7 Applications ........................... 62  
6.1.3  IxLoad 1G Port Throughput Test ....................................... 64  
6.1.4  IxLoad 10G Port Throughput Test ...................................... 70  
6.2  Session Management ............................................................. 74  
6.2.1  ECM State Change .......................................................... 74  
6.2.2  Tracking Area Update ....................................................... 77  
6.2.3  Downlink Data Notification ................................................. 80  
6.2.4  X2-Based Handover ......................................................... 83  
6.2.5  Network Initiated Dedicated Bearer ..................................... 86  
6.3  Other Scenarios ..................................................................... 88  
6.3.1  Busy Hour Functionality .................................................... 88  
6.3.2  Deep Package Inspection .................................................. 91  
6.3.3  Dynamic QoS Enforcement ............................................... 94  
6.3.4  Multiple APNs ................................................................. 98  

7  Feedback .................................................................................. 101  
7.1  Feedback to Ixia ................................................................. 101  
7.1.1  S1-Based Handover with Indirect Tunneling ....................... 101  
7.1.2  IxLoad Report Generator ................................................... 102  
7.1.3  Three-Way and Four-Way Tear Down ............................... 103  
7.1.4  Limitation of IPv6 ........................................................... 104  
7.1.5  Future Suggestions of Improving the IxLoad ....................... 105  
7.2  Feedback to Ericsson (Ericsson Confidential) ......................... 106  
7.2.1  Traffic Shaping ............................................................... 106  
7.2.2  Policy Control ............................................................... 106  
7.2.3  Session Management ..................................................... 106  

8  Conclusions ............................................................................. 107  
8.1  Conclusions to IxLoad Assessment ......................................... 107  
8.2  Future Work ......................................................................... 108
Assessment of IxLoad in an MPG Environment
Yue Peng / Zhiqiang Tang

Table of Contents

References......................................................................................................110
Appendix A: Open Issues of IxLoad ........................................................114
Appendix B: Official TR of MPG ...............................................................116
# Terminology

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>3GPP</td>
<td>3rd Generation Partnership Project</td>
</tr>
<tr>
<td>3G</td>
<td>3rd Generation</td>
</tr>
<tr>
<td>4G</td>
<td>4th Generation</td>
</tr>
<tr>
<td>AAA</td>
<td>Authentication, Authorization, and Accounting</td>
</tr>
<tr>
<td>AMBR</td>
<td>Aggregate Maximum Bit Rate</td>
</tr>
<tr>
<td>ARP</td>
<td>Allocation and Retention Priority</td>
</tr>
<tr>
<td>APN</td>
<td>Access Point Name</td>
</tr>
<tr>
<td>AVP</td>
<td>Attribute Value Pairs</td>
</tr>
<tr>
<td>BHCA</td>
<td>Busy Hour Call Attempt</td>
</tr>
<tr>
<td>CCA</td>
<td>Credit-Control-Answer</td>
</tr>
<tr>
<td>CCR</td>
<td>Credit-Control-Request</td>
</tr>
<tr>
<td>CDR</td>
<td>Charging Data Record</td>
</tr>
<tr>
<td>CS</td>
<td>Circuit Switched</td>
</tr>
<tr>
<td>DBP</td>
<td>Diameter Based Protocol</td>
</tr>
<tr>
<td>DCCA</td>
<td>Diameter Credit Control Application</td>
</tr>
<tr>
<td>DDN</td>
<td>Downlink Data Notification</td>
</tr>
<tr>
<td>DL</td>
<td>Downlink</td>
</tr>
<tr>
<td>DPI</td>
<td>Deep Packet Inspection</td>
</tr>
<tr>
<td>DUT</td>
<td>Device Under Test</td>
</tr>
<tr>
<td>ECM</td>
<td>EPS Connection Management</td>
</tr>
</tbody>
</table>
### Terminology

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDGE</td>
<td>Enhanced Data Rates for GSM Evolution</td>
</tr>
<tr>
<td>eNodeB</td>
<td>Evolved Base Station</td>
</tr>
<tr>
<td>EPC</td>
<td>Evolved Packet Core</td>
</tr>
<tr>
<td>EPS</td>
<td>Evolved Packet System</td>
</tr>
<tr>
<td>E-UTRA</td>
<td>Evolved Universal Terrestrial Radio Access</td>
</tr>
<tr>
<td>eGTP</td>
<td>Evolved GPRS Tunneling Protocol</td>
</tr>
<tr>
<td>GBR</td>
<td>Guaranteed Bit Rate</td>
</tr>
<tr>
<td>GGSN</td>
<td>Gateway GPRS Support Node</td>
</tr>
<tr>
<td>GSA</td>
<td>Global mobile Suppliers Association</td>
</tr>
<tr>
<td>GSM</td>
<td>Global System for Mobile Communication</td>
</tr>
<tr>
<td>GTP</td>
<td>GPRS Tunneling Protocol</td>
</tr>
<tr>
<td>GW</td>
<td>Gateway</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphic User Interface</td>
</tr>
<tr>
<td>H-PCRF</td>
<td>Home PCRF</td>
</tr>
<tr>
<td>HLR</td>
<td>Home Location Register</td>
</tr>
<tr>
<td>HSPA</td>
<td>High Speed Packet Access</td>
</tr>
<tr>
<td>HSS</td>
<td>Home Subscriber Server</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>IMS</td>
<td>IP Multimedia Subsystem</td>
</tr>
<tr>
<td>IMSI</td>
<td>International Mobile Subscriber Identity</td>
</tr>
<tr>
<td>IP-CAN</td>
<td>IP Connectivity Access Network</td>
</tr>
<tr>
<td>IPSec</td>
<td>IP Security</td>
</tr>
<tr>
<td>LTE</td>
<td>Long Term Evolution</td>
</tr>
</tbody>
</table>
### Terminology

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC</td>
<td>Medium Access Control</td>
</tr>
<tr>
<td>MBR</td>
<td>Maximum Bit Rate</td>
</tr>
<tr>
<td>MBRD</td>
<td>Maximum Bit Rate for Downlink</td>
</tr>
<tr>
<td>MBRU</td>
<td>Maximum Bit Rate for Uplink</td>
</tr>
<tr>
<td>MCC</td>
<td>Mobile Country Code</td>
</tr>
<tr>
<td>MIMO</td>
<td>Multiple Input Multiple Output</td>
</tr>
<tr>
<td>MME</td>
<td>Mobility Management Entity</td>
</tr>
<tr>
<td>MNC</td>
<td>Mobile Network Code</td>
</tr>
<tr>
<td>MPG</td>
<td>Mobility Packet Gateway</td>
</tr>
<tr>
<td>MSISDN</td>
<td>Mobile Subscriber ISDN Number</td>
</tr>
<tr>
<td>MSS</td>
<td>Maximum Segment Size</td>
</tr>
<tr>
<td>MTU</td>
<td>Maximum Transmitted Unit</td>
</tr>
<tr>
<td>NAS</td>
<td>Non Access Stratum</td>
</tr>
<tr>
<td>OCS</td>
<td>Online Charge System</td>
</tr>
<tr>
<td>OFDM</td>
<td>Orthogonal Frequency-Division Multiplexing</td>
</tr>
<tr>
<td>OFDMA</td>
<td>Orthogonal Frequency-Division Multiple Access</td>
</tr>
<tr>
<td>OFCS</td>
<td>Offline Charge System</td>
</tr>
<tr>
<td>PCC</td>
<td>Policy Charging Control</td>
</tr>
<tr>
<td>PCEF</td>
<td>Policy and Charging Enforcement Function</td>
</tr>
<tr>
<td>PCRF</td>
<td>Policy and Charging Rules Function</td>
</tr>
<tr>
<td>PDN</td>
<td>Packet Data Network</td>
</tr>
<tr>
<td>PDU</td>
<td>Protocol Data Unit</td>
</tr>
<tr>
<td>PGW</td>
<td>Packet Data Network Gateway</td>
</tr>
</tbody>
</table>
Terminology

PMIP        Proxy Mobile IP
PS             Packet Switched
RAA        Re-Authorization Answer
RADIUS  Remote Authentication Dial In User Service
RAR       Re-Authorization Request
RAT          Radio Access Technologies
RFC        Request for Command
RNC          Radio Network Controller
RRC          Radio Resource Control
QCI           QoS Class Identifier
QoS           Quality of Service
SACC       Service Aware Charging and Control
SDF-ID     Service Data Flow ID
SGW         Serving Gateway
SGSN       Serving GPRS Support Node
TA            Tracking Area
TAU           Tracking Area Update
TFT          Traffic Flow Template
TLS          Transport Layer Security
ToS          Type of Service
T-PDU     Transport Protocol Data Unit
TEID          Tunnel Endpoint Identifier
UDP        User Datagram Protocol
### Terminology

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>UE</td>
<td>User Equipment</td>
</tr>
<tr>
<td>UL</td>
<td>Uplink</td>
</tr>
<tr>
<td>UMTS</td>
<td>Universal Mobile Telecommunications System</td>
</tr>
<tr>
<td>UTRAN</td>
<td>Universal Terrestrial Radio Access Network</td>
</tr>
<tr>
<td>V-PCRF</td>
<td>Visited PCRF</td>
</tr>
<tr>
<td>WCDMA</td>
<td>Wideband Code Division Multiple Access</td>
</tr>
<tr>
<td>WiMAX</td>
<td>Worldwide Interoperability for Microwave Access</td>
</tr>
</tbody>
</table>
1 Introduction

1.1 Background and Problem Motivation

More than 2 billion subscribers are using 3GPP Second Generation technology GSM and the Third Generation technology Universal Mobile Telecommunication System (UMTS) for mobile communication and it suggests that 3GPP has been a resounding success. LTE is the latest mobile network technology published by 3GPP. According to the report from Global mobile Suppliers Association (GSA), 110 operators in 48 countries will invest in LTE networks and at least 45 LTE networks are expected to be in the service by the end of 2012 [1].

As a world-leading provider of telecommunication equipment, Ericsson has committed itself to inventing products and services in relation to LTE. With the equipments supplied by Ericsson, the Swedish-Finnish operator TeliaSonera launches the world’s first commercial LTE network in Stockholm. MPG is a mature product of Ericsson, combining Serving Gateway in addition to PDN Gateway functionality, and is also simultaneously supporting GSM, WCDMA and LTE traffic. Ixia is a global-leading provider of IP performance test systems, offering several products and solutions for LTE network elements test. IxLoad is one of these products and is able to emulate LTE traffic. Ericsson provides the MPG test environment and the thesis workers to Ixia in order to assess the performance of IxLoad.

1.2 Overall Aim

This master thesis aims to assess the capacity and LTE functionality of IxLoad in the Ericsson MPG environment and to find a feasible solution towards improving the IxLoad. A set of test cases will be presented to prove whether IxLoad has offered a satisfactory quality in the LTE testing. All the test cases are designed based on the capacity features and LTE functionalities. In addition to results of each test case, several suggestions for improving Ixia IxLoad and Ericsson MPG will be proposed as the outcome of this thesis.
1.3 Scope
This thesis focuses on the verification of IxLoad performance in Ericsson MPG environment. In this report, the background of LTE networks and the main features of MPG related to the thesis will be stated initially. Following this, IxLoad will be introduced specifically with the LTE features. The most important part of this thesis is a list of test cases that prove the performance of IxLoad. The outcome of this thesis will be the conclusion of the test results and the feedback given to both Ixia and Ericsson.

1.4 Concrete and Verifiable Goals
This thesis focuses on assessing the capacity and some central functionalities of IxLoad on a LTE traffic simulation. The target of capacity assessment is to compare the real performance of IxLoad in the MPG environment with experimental performance from the Ixia lab environment [2]. The functionality assessment aims to assess the LTE features offered by IxLoad.

The capacity assessment covers the maximum number of simulated users that IxLoad can support and the maximum throughput that can be achieved by IxLoad, under certain conditions.

In addition to Session Management such as Track Area Update and Handover, some other features including Busy Hour Functionality, Deep Packet Inspection, Multiple APNs in addition to the Dynamic Quality of Service Enforcement will also be verified in the functional assessment.

Finally, some constructive suggestions are desired by Ixia for improving IxLoad. Furthermore, the test cases developed during the thesis can be used by Ericsson to improve and verify the quality of MPG.

1.5 Outline
Chapter 2 introduces the background knowledge of the UMTS and LTE packet core network, as well as the particular protocols used in the LTE core network, such as eGTP and Diameter.

Chapter 3 states several functionalities of the MPG that are related to the thesis work. This part focuses on the session management, service charging and Quality of Service (QoS) treatment.
Chapter 4 is a brief outline of the features of IxLoad. In this chapter, eGTP features are particularly highlighted, in comparison of those of the previous thesis [3].

Chapter 5 shows two test environments used in the thesis: the back to back test environment that is only associated with the Ixia Chassis and MPG environment, which Ericsson MPG is involved in.

Chapter 6 presents all the test cases used to evaluate the IxLoad capacity and features in the MPG or back to back test environment. Each test case is illustrated in the form of purpose, expected results, configuration as well as verification and results.

Chapter 7 aims to provide some feedback to both Ixia and Ericsson. For Ixia, some improvement suggestions for the IxLoad are given based on the issues which arose. Several suggestions for the MPG are also stated as the feedback to Ericsson.

Chapter 8 concludes this thesis work and offers some proposals for future work.

1.6 Contributions
Yue Peng and Tang Zhiqiang cooperated together during the entire thesis project. All the test cases are simulated together and each of the authors has been responsible for 50% of the thesis report in every chapter.
2 Related Work

2.1 Evolved Packet Core Network Background

In recent years, data consumption services, such as mobile TV, location based service and online game have been growing at a rapid rate. The global mobile data traffic has increased by an approximate factor of 2.6 from 2009 to 2010 [4]. Cisco Visual Networking Index forecasts that mobile data traffic will increase by a factor of 39 between 2009 and 2014 as data consumption will double during each year. In 2014, 1.5 billion machines will be connected to a mobile network and more than 5.6 billion subscribers will use data service [5]. Mobile data service appears to be overtaking the position of voice service in the operator developing strategies, which also drives the evolution of the networks.

2.2 UMTS Packet Core Network

The Third Generation (3G) of mobile network is represented in the GSM-domain by UMTS, which is specified by 3GPP. The most common form of UMTS uses Wideband Code Division Multiple Access (WCDMA) as its underlying air interface. WCDMA specifies the 5M Hz wide channel, which is significantly more than GSM.

Figure 2-1 Architecture of UMTS [4]
Figure 2-1 shows the overall architecture of the UMTS system, consisting of the radio access network and core network. The core network is essentially divided into the Circuit Switched (CS) domain and the Packet Switched (PS) domain. The PS core network is responsible for setting up, routing, controlling and terminating the PS data calls. Packet switched elements are Serving GPRS Support Node (SGSN), Gateway GPRS Support Node (GGSN) and the Packet Data Network (PDN), as shown in the shaded portion of Figure 2-1.

An SGSN manages to deliver the data packets from one Base Station to another within its geographical service area. It handles the communication with User Equipment (UE) and the establishing of the connection between UE and the PDN.

GGSN is the main component in the PS core network, which is interworking between the internal GPRS network and external packet switched network. GGSN is responsible for IP address assignment and is the default router for the connected UE. The GGSN also performs authentication and charging functions [7]. GGSN forwards the downlink IP packets from PDN to SGSN through the Gn interface and exchanges the uplink IP packets from SGSN to PDN via the Gi interface.

### 2.3 LTE Packet Core Network

As a pre-4th Generation (4G) technology, LTE constructs a new network system, called EPS. EPS consists of EPC and an evolved access network, offering a totally new experience to the subscribers with broadband access and many applications.
Figure 2-2 illustrates the EPC Network architecture, nodes and interfaces. The main function of EPC is to provide a packet-switched service as well as service and bearer level management such as QoS. Figure 2-2 shows that EPC is not only supporting LTE and non-3GPP access networks, but also backward compatible with 2G GSM and 3G WCDMA radio network. This thesis will concentrate on the basic EPS architecture of LTE, as shown in Figure 2-3. It consists of Evolved Base Stations (eNodeB), Mobility Management Entity (MME), Serving Gateway (SGW), Packet Data Network Gateway (PGW), and Policy and Charging Rules Function (PCRF).
**eNodeB**

Evolved Universal Terrestrial Radio Access Network (E-UTRAN) consists of eNodeBs, which provide the radio interface for UEs and are responsible for the complete radio resource management in LTE. eNodeB holds the air interface stack and networking stack. In LTE, eNodeB integrates some of the features of Radio Network Controller (RNC) in 3G network [9].

- According to the 3GPP standard, when UE starts up, eNodeB will manage the radio resource for UE at the Uplink/Downlink (UL/DL) bearer level, i.e., the bearer rate enforcement, radio admission control and the uplink and downlink transmission allocation.
- When a packet reaches eNodeB, it will perform the header compression and user plane ciphering, i.e., compress the IP header, encrypt the data stream, add the GTP header to the payload and then send it to the SGW.
- eNodeB is also responsible for the MME selection, based on the information provided by UE.

eNodeBs are allowed to interconnect with each other via the X2 interface, which is used for the handover.
MME
The MME is the crucial control node in the LTE network. It is responsible for all the control plane functions related to subscribers and session management. More specifically, MME has the following main functions [10].

- Handle the location update and subscribers’ roaming.
- Execute track area list management and paging
- In charge of subscribers’ authentication as well as Non-Access-Stratum (NAS) signaling security.
- Take care of PGW and SGW selection.
- Deal with inter core network signaling for mobility between 3GPP access network.
- Responsible for bearer management, including dedicated bearer establishment.

Since the MME carries out all the functions related to subscribers and session management, each eNodeB should be connected to at least one MME via the S1 interface and MME must store the subscription information in addition to the location information for each subscriber registered in this MME. To be able to obtain the subscription information, MME is connected to the Home Subscriber Server (HSS), which contains the subscription details for each mobile user who is authorized by the operator.

Serving Gateway
SGW is the termination point of the user plane interface towards eNodeB. Each eNodeB should connect to at least one SGW and one SGW can serve several eNodeBs simultaneously in terms of load capacity. The main functions of SGW are as follows.

- Packet routing and forwarding
  When there is no NAS connection between the subscriber and access network, the subscriber is in EPS Connection Management (ECM) – IDLE state. For the subscribers in idle state, once the SGW receives the user’s data from PGW, the SGW firstly buffers the downlink packets and then initiates the network triggered service request procedure. For the subscribers in the ECM-CONNECTED state, it can be described by means of two different scenarios. In the downlink, the SGW receives the IP packets from PGW and routes them to a particular eNodeB which covers the subscriber that these packets belong to.
In the uplink, the SGW forwards the IP packets received from eNodeB to a particular PGW determined by MME or the subscription information of the sender.

- Mobility anchoring for inter-3GPP network
  SGW also acts as an anchor for the inter-eNodeB handover and for inter-3GPP (GSM, WCDMA, LTE) mobility.

- Charging
  SGW supports the generation of the billing record per UE or bearer, and hence it resolves any inter-operator charging problems that may arise in the case where a roaming subscriber has to access SGW of visited mobile network.

**PDN Gateway**

PGW provides the connectivity between the internal EPS and the external PDNs and terminates the SGi interface towards the PDNs. A UE is able to connect to more than one PGW if it requires to gain access to more than one PDN.

The functions of PGW are included for both GPRS Tunnelling Protocol (GTP) based and the Proxy Mobile IP (PMIP) based S5/S8.

- PGW supports the packet filtering on a per user basis, for example, it supports the deep packet inspection which enables advanced network and user services management.
- PGW is responsible for the UE IP address allocation and the transport level packet marking in both uplink and downlink. It also performs the service level gating control and rate enforcement through rate policing and shaping [8].
- Referring to the bearer QoS, PGW communicates with PCRF via the Gx interface to determine the bearer properties, such as the UL/DL rate enforcement based on the APN Aggregate Maximum Bit Rate (AMBR).

**Policy and Charging Rules Function**

PCRF is the policy and charging control element, which is responsible for both the network based policy control handling including the service data detection, gating control and QoS control and flow-based charging control towards the Policy and Charging Enforcement Function (PCEF). In the LTE core network, PCRF acts as an intelligent device based on the
ongoing traffic or existing configuration that triggers events towards PGW.

In a non-roaming scenario, only one single PCRF is sufficient with one IP Connectivity Access Network (IP-CAN) session of UE.

In a roaming scenario with a local breakout of traffic, there might be two PCRFs associated with one UE’s IP-CAN session, the Home PCRF (H-PCRF) and Visited PCRF (V-PCRF) [9].

2.4 Protocol

2.4.1 GTP

GTP is used for constructing the GTP tunnel between two nodes in order to separate traffic flow. The original version of GTP is version 0, standardized by European Telecommunications Standards Institute (ETSI). The prime target of this protocol is to cater to the specific requirements of GSM. The later version of GTP has been improved by 3GPP in 3G UMTS. Meanwhile version 0 has quickly been replaced by version 1, because of the security problem. At the end of 2008, GTP version 2 was published by 3GPP. In the new release of GTP, the GTP control protocol has been upgraded from GTPv1-C to GTPv2-C to enhance the bearer handling.

GTP consists of two main components: GTP-C and GTP-U. GTP-C is used for signaling on the control plane and GTP-U is used to transmit user data on the user plane [10]. More specifically, three versions of GTP-C and two versions of GTP-U have been developed, as shown in Figure 2-4. In addition, GTP prime (GTP’) also belongs to GTP. GTP’ has the same message structure as GTP-C and GTP-U, but the function of GTP’ is to carry charging data to the billing centre.
GTP-C and GTP-U

To specify the source and destination of the GTP tunnel, each endpoint of the GTP tunnel is identified with Tunnel endpoint Identifier (TEID), IP address and a UDP port number. The value of TEID is assigned by the tunnel endpoint at the receiving side of the GTP tunnel. Tunnel endpoints exchange TEID with GTP-C messages. The TEID value is a consequence of the negotiation with a control plane protocol GTP-C. In addition to tunnel management (set up, modify, maintain, delete), GTP-C also conducts location management. During the handover, GTP-C exchanges the moving users’ information between the two nodes. Figure 2-5 shows the protocol stack of GTPv2-C.
GTP-U is used for user data packets transmission via GTP tunnels. Specifically, a GTP-U tunnel conveys user original packet (Transport Protocol Data Unit (T-PDU)) and a signaling message from the GTP-U sending endpoints (ingress GTP tunnel) to the receiving endpoints (egress GTP tunnel). At the ingress GTP tunnel, T-PDU is encapsulated with a GTP header, User Datagram Protocol (UDP) and IP header. If the size of the resulting IP packet is larger than the Maximum Transmitted Unit (MTU), then the fragmentation should be performed according to the Internet Engineering Task Force (IETF) Request For Command (RFC) 791 for an outer layer of IPv4 or IETF RFC 2460 for an outer layer of IPv6. The TEID obtained from GTP-C is tagged in the head of GTP header so that GTP-U can identify which tunnel a particular T-PDU belongs to. As a result, fragmented packets can be multiplexed and demultiplexed by GTP-U between two GTP-U tunnel endpoints, so as to share expensive resources. At the egress GTP tunnel, received fragmented packets are reassembled. Then these reassembled packets are passed to IP/UDP/GTP-U. Hence T-PDU is recovered. The GTPv1-U protocol stack is illustrated in Figure 2-6.
3GPP outlines the Evolved GPRS Tunnelling Protocol (eGTP) with two separate specifications: GTPv1-U and GTPv2-C [13]. The following sections will introduce more details of GTPv1-U and GTPv2-C from the perspective of the interface based on 3GPP.

3GPP 29.274 [11] defines that the GTPv2-C should be used across the seven EPC signaling interfaces: S3, S4, S5, S8, S10, S11 and S16. This master thesis will concentrate on S5-C and S11-C, which are the most important interfaces of the control plan in MPG.

S5 is the IP interface between SGW and PGW. S5-C provides the bearer management between the two gateways. There are two protocol options for S5. The former is a GTP based variant. The release 8 of 3GPP applies GTPv2-C to the control plane of S5 and GTPv1-U to the user plane of S5. The latter is a PMIP based variant. PMIP-C and PMIP-U are used for the control plane and the user plane of S5 respectively. The interface S11-C takes charge of carrying the control signaling between MME and SGW. In contrast to that of the S5, the GTPv2-C is the only protocol supported by the interface S11-C in the release 8 of 3GPP.
3GPP 29.281 [14] defines that the GTPv1-U is used on the S1-U, X2, S4, S5, S8 and S12 interfaces of the EPS. To evaluate the performance of MPG in the case of heavy traffic, this thesis will provide a brief introduction of S1-U and focus on X2-U and S5-U interfaces.

The S1-U is responsible for conveying the user traffic between eNodeB and SGW. If the negotiation between eNodeB, MME and SGW via S1-MME and S11 is successful, then S1-U adopts the negotiated parameters such as TEID to build up GTP tunnel for the user data transmission. The X2-U is in charge of transporting the user data packets between the eNodeBs. For example, the user plane of eNodeBs A and B connect via X2-U. When the subscriber moves from eNodeB A to B, the interface X2-U is used for a short duration to forward the buffered packets from eNodeB A to B. The interface S5-U takes charge of the user plane data tunnel creation between SGW and PGW. Once the two gateways reach an agreement via S5-C, S5-U establishes a GTP tunnel as a bridge connecting SGW and PGW. The GTP protocol stack of the interfaces X2, S1, S5, S11 is presented in Figure 2-7.

### 2.4.2 Diameter

Diameter is a protocol used for Authentication, Authorization, and Accounting (AAA). The protocol name Diameter comes from Remote Authentication Dial In User Service (RADIUS) in the sense that the radius is half of the diameter. With the rapid growth of mobile Internet, traditional AAA protocols such as RADIUS lack sufficient capability to handle more and more complex network access scenarios. Diameter has been developed in order to satisfy these requirements. It not only supports more network access scenarios with better security, but also provides improved failure handling and thus offers a more reliable message delivery and in addition it has better extensibility. The core of the Diameter is the Diameter Based Protocol. The additional extension of Diameter
is called application. Several applications, such as, Diameter Credit Control Application, Diameter Mobile IPv4 Application, Diameter Network Access Server Application, have been extended based on Diameter Based Protocol.

**Diameter Based Protocol**

Diameter Based Protocol (DBP) is promoted by IETF and is defined in RFC3588 which specifies the minimum requirement for the implementation of Diameter [15]. One advantage of DBP is that it is still sufficiently feasible for it to provide embedded system implementation, while it performs more complex functionality than RAIDs.

DBP communication is on the basis of a transport level connection. A reliable transmission is achieved by utilizing either TCP or SCTP to convey DBP messages. As the transport connection of the two Diameter peers is established successfully, they can exchange basic information to discover the other’s identities and capability. It then becomes possible for each of them to assume the other’s role. In addition to performing basic access control at the edge of network, the Diameter client starts to forward authentication information and service specific authorization information to the Diameter server via DBP. Diameter server is in charge of answering authorization, authentication and accounting requests. Moreover, Diameter client keeps inquiring the state of transport path between client and server periodically. Once the transport failure occurs, it can be detected quickly and Diameter client launches the failure handling. In the case that Diameter client fails to reach Diameter server due to transport failure, Diameter client saves all the pending requests in a queue and performs a failover procedure. If an alternative Diameter server is available, these pending messages will be sent this server.

Each Diameter message consists of a Diameter header and a variable number of Attribute Value Pairs (AVP). The header of the Diameter message contains a unique command code to indicate the intention of the message. Currently, fourteen commands are defined in DBP. Each of the Diameter request commands has a corresponding answer command. For example, Device-Watchdog-Answer is the answer command for the Device-Watchdog-Request. All the actual data is encapsulated in the form of AVP. Each AVP uses a unique AVP code to identify itself. At present, there are fifty AVPs defined in DBP. In DBP, it is possible to group the AVP values of a similar type, and thus the data field can be made up of several AVPs. In this way, the data field is expanded. Simi-
lar to RADIUS, some commands and AVPs are allowed to be specified by the vendors on the basis of DBP.

**Diameter Credit Control Application**

Diameter Credit Control Application (DCCA) is the extension of DBP and standardized in RFC4006 [16]. DCCA specifies the communication criterion between credit control clients such as the Gateway and credit control servers that consist of a back-end account system and a balance system, and thereby provides a framework to carry out real-time credit control for different end user services, such as download services and message services.

Two credit authorization models, credit authorization with direct debiting and credit authorization with money reservation, are supported by RFC4006. For both models, the service requested by the end user should be forwarded to the credit control server from the credit control client, before the credit control client initiates the service. The credit control server must check whether the user account of the requesting service is valid and the requested service is subscribed. The credit control client does not provide any service to the users, until the credit authorization has been approved by the credit control server. If the user is overdue, the credit control server commands the credit control client to deny any chargeable service by this user.

In the credit authorization with the direct debiting model, as the credit control server receives a credit authorization request from the client side, it deducts a predefined amount of money from the user’s account and replies with a successful credit authorization answer. Then the credit control client starts to offer the service to the end user. This is a single transaction process and the session state will no longer be maintained after this one-time event. In the credit authorization with the money reservation model, the credit control server not only reserves a suitable amount of money but also allocates certain resources corresponding to this reserved money to the user. In contrast to the first model, the second model is a session based credit control. The credit control client and server must retain the credit control sessions as long as the service is not terminated. When the granted resources are about to be consumed, but the service is still continuing, the credit control must report the used quota and apply for new resources.
Two significant command messages, Credit-Control-Request (CCR) and Credit-Control-Answer (CCA), are brought forward in RFC4006 to fulfill two models. CCR is sent from the credit control client to the credit control server in order to request authorization for a service. CCA is the answer to the CCR and is sent by the credit control server to inform the credit control client of the results for the corresponding authorization request. For the first time credit authorization in the credit authorization with money reservation model, the credit control client configures a CC-Request-Type as initial request and it the meanwhile specifies other parameters of the CCA message appropriately, and then sends the CCR message to the server side. The credit control server writes the allocated resource into the Granted-Service-Unit of CCA message and sends it back to the client side. Before the allocated resources are exhausted, the credit control client can apply the extra resources by setting the CC-Request-Type to update the request and to report the consumed quota with Used-Service-Unit in the CCR message. The new resources are indicated in the Granted-Service-Unit of CCA message which has been replied by the credit control server, if the user account has sufficient credits. In addition to a traditional balance check, both service price enquiry and refund are supported by RFC4006. For a service price enquiry, the credit control client sets the Requested-Action AVP to price enquiry and fills in the Service-Identifier AVP with the requested service event information in the CCR message. Once the credit control server receives CCR, it estimates the cost of the service and inserts the cost into Cost-Information AVP of CCA. The credit control client acknowledges the price from CCA.

**Gx protocol**

![Gx interface](image)

In LTE system, PGW is connected to PCRF via an Gx interface. The intention of introducing the Gx interface into the LTE network is to implement the Policy and Charging Control (PCC) functionality. The PCC functionality can be divided into PCC rules handling and event handling for PCC. The PCC rules handling is responsible for the PCC rules installation, modification and removal. Event handling is in charge of informing PCRF that a certain event PCRF has subscribed to in PGW,
is occurring. Thus PCRF is capable of changing PCC rules, when the certain event happens.

The Gx protocol is defined as a vendor specific Diameter application, used for Gx interface communication. 3GPP TS29.210 [18] clarifies that the Gx protocol should be based on the DBP specified in the RFC 3588 and the DCCA specified in the RFC 4006. According to the Diameter protocol, PCRF deals with the PCC rule requests, acting as the Diameter server. PGW plays a role of the Diameter client, who requests the PCC rules.
3 Mobile Packet Gateway

MPG is a packet core network element for the infrastructure construction of the LTE network. It has been developed by Ericsson, following the 3GPP standards. MPG combines Serving Gateway and PDN Gateway functionality [19]. GSM, WCDMA, LTE traffic can be supported simultaneously [20]. Furthermore, MPG is able to perform packet inspection and service classification. Operators can choose the traffic shaping mechanism, in the light of the network load condition that MPG detects.

3.1 Session Management

3.1.1 ECM State Change

In LTE networks, UE only has two EPS Connection Management (ECM) states. UE stays either in ECM-IDLE state or in ECM-CONNECTED state. When UE is in the ECM-IDLE state, there is no NAS signaling connection between UE and PDN. On the contrary, UE in the ECM-CONNECTED state should retain its signaling communication with MME. When UE reaches the timer how long UE can be in the state of CONNECTED without any data activity, E-UTRAN will withdraw radio resources and the Core network will release the connection relating to UE. The ECM state of UE is changed to IDLE from CONNECTED. If UE intends to obtain data service again, it should firstly be shifted to CONNECTED state first. Once UE establishes signaling connection with MME, it will enter ECM-CONNECTED state.

There are four scenarios in ECM state change, as shown in Figure 3-1. The first two scenarios are that UE or network such as eNodeB initiates resource release between UE and Core network. The ECM state of UE is changed to IDLE from CONNECTED. The other two scenarios are a UE...
triggered service request and a network triggered service request under the condition that UE is in the state of ECM-IDLE. Both triggers lead to switching UE state from ECM-IDLE to ECM-CONNECTED. The four scenarios are supported by MPG.

### 3.1.2 Default Bearer Activation and Deactivation

In the EPS bearer activation, the default bearer is the first bearer activated when a UE connects to the PDN and it remains established during the lifetime of the PDN connection. The EPS bearer provides the UE with a continuous IP connectivity. In most cases, the default bearer will be associated with the default QoS information and will be used for the IP traffic without any specific QoS treatment requirement.

![Figure 3-2 Activation, modification, deactivation of a default bearer](image)

The Figure 3-2 is a basic procedure of activation, modification and deactivation of a default bearer. All the procedures are based on the 3GPP standard [9].
UE initiates the activation procedure by transmitting the Attach Request to eNodeB. eNodeB forwards the Attach Request message to MME connected to that eNodeB. MME sends the Create Session Request message to SGW, which contains the UE identification information, APN selection, the QoS parameters and the tunnel endpoint ID. As long as SGW receives the request, it will create a new entry in its EPS Bearer table and forwards the Create Session Request message to PGW. PGW will allocate all the resources to initiate the requested EPS bearer with routing the user plane Protocol Data Units (PDU) between the SGW and PDN and reply with a Create Session Response to SGW. SGW returns the response to MME and MME immediately sends an Attach Accept message to eNodeB. eNodeB will then forward the RRC connection Reconfiguration message to the UE.

In a modified form, eNodeB sends the Attach Complete message to the MME and it then transmits the Modify Bearer Request message to SGW, which includes the TEID and the address of the eNodeB used for the downlink traffic on the S1-U interface. In the handover scenarios, SGW forwards the request message to PGW and the PGW acknowledges SGW by sending an accepted Modify Bearer Response. As soon as MME receives the response, SGW is able to send its buffered downlink packets.

In the deactivation procedure, UE sends a Detach Request message to inform MME to terminate the PDN connection. MME will send the Delete Session Request per PDN connection to SGW, making the active EPS bearer in SGW regarding the particular UE to be deactivated by MME. SGW releases all the related EPS bearer information and meanwhile, it sends a Delete Session Request to PGW, which replies with a Delete Session Response after the release of all the allocated resources. Finally, SGW sends back a Delete Session Response to MME until it receives a response from PGW.

3.1.3 Dedicated Bearer Activation

In addition to the default bearers, dedicated bearers also can be used in the PDN connection. A dedicated bearer is always linked to a default bearer and represents the additional bearer resources between UE and PDN. Usually the dedicated EPS bearer context activation procedure is initiated by the network, but it may be requested by the UE with the UE requested bearer resource allocation procedure or the UE requested bearer resource modification procedure [21]. The procedure can be
initiated together with the default bearer. It also can be initiated at a later stage, as long as the default bearer remains in an active state.

In the dedicated bearer activation procedure, a PCRF that can be deployed with the dynamic PCC rule is required. A dynamic PCC rule contains a set of information with regards to the service data flow, rating group and single service data flow gating. Within the dynamic charging rule, all the data would be assigned in a real time analysis. Figure 3-3 shows the network initiated dedicated bearer activation procedure. The S11 and Gx procedure will be stated in detail.

PCRF initiates the dedicated bearer activation by sending a CCA message or a Re-Authorization Request (RAR) message to the PGW. The message includes the policy decisions, including the charging information, the QCI information, default bearer QoS, APN-AMBR etc.

PGW sends a Re-Authorization Answer (RAA) to PCRF and sends a Create Bearer Request message which assigns those QoS parameters to SGW via the S5/S8 interface and SGW will forward the Create Bearer Request message to MME. When MME receives the request, it will select the EPS Bearer identity to assign to the UE and signals the Bearer Setup Request to eNodeB. eNodeB will then construct an RRC Connection
Reconfiguration message to UE. After the negotiation and reconfiguration between eNodeB and UE, MME acknowledges the bearer activation to SGW with the Create Bearer Response. Finally SGW replies to the response from PGW to complete the activation procedure of the dedicated bearer.

### 3.1.4 Tracking Area Update

The core network must always know the location of every UE registered in this network in case these UEs are called by other UEs. To be able to keep track of the locations of UEs, LTE core network requires UEs under certain conditions to implement Tracking Area Update (TAU) procedure in order to report their location information, which is similar in GSM/WCDMA network.

LTE network provider divides its covered areas into one or several Tracking Areas (TAs). Each TA is constituted with a set of contiguous eNodeBs, as shown in Figure 3-4. The size of TA is attained by analyzing the traffic model, taking network signaling overheads and radio paging load into account, in order to achieve a tradeoff between signaling load and paging load. For example, a TA of smaller size can relieve the pressure for all eNodeBs within the TA on handling heavy traffic for paging, but it produces high signaling overhead due to location update procedure. The concept of the tracking area list is brought into the LTE system to lighten the location update signaling overhead. After successful registration, every UE owns a tracking area list allocated by the core network. The implication is that a UE is allowed to belong to several TAs simultaneously. As long as UEs are moving within TAs allocated in the tracking area lists, this does not cause any signaling overhead for TAU. Usually TAU occurs in the three scenarios stated below.

- UEs register the LTE core network for the first time.
- UEs pass a border to a TA, which is not in the tracking area list.
- UEs updates the LTE core network about its location, when the location update timer expires.

Three scenarios are presented in Figure 3-4. When the UE1 locating in TA1 is conducting the first time registration, it triggers TAU to inform the core network of its current position. Then UE1 receives a tracking area list which consists of TA1 and TA2. As long as UE1 stays in TA1 and TA2, TAU is only triggered by the location update timer. By receiving the TAU messages periodically from UE1, the core network is aware that UE1 is reachable in its network. As UE1 moves to TA3, UE1 receives
TA information that contains TA3 instead of TA2 from eNodeB. After comparing the received TA with the allocated TAs in the tracking area list, it notices that TA3 is not in the current tracking area list, which leads to the TAU procedure in order to inform the core network that UE1 is now in the new TA.

![Image of Tracking Area Update](image)

**Figure 3-4 Tracking area update [8]**

After UE triggers TAU, MME will make a decision as to whether the SGW reallocation is a necessary TAU procedure, so there are two scenarios defined in 3GPP standard [12]: The TAU procedure with and without the SGW change. Ericsson MPG supports both cases.

### 3.1.5 Handover

The significant revolution from a landline telephone to a mobile phone is that users can retain online services while moving wherever they want. Handover is a key technique to make this become a reality. Handover relates to the process that transfers an ongoing voice call or data service from one radio channel to another. For WCDMA network, soft handover is employed in the UTRAN architecture, because of CDMA based radio interface. LTE adopts the Orthogonal Frequency Division Multiple Access (OFDMA) based radio interface [22]. It not only avoids expensive radio resource costs due to the soft handover but also improves the spectral efficiency.

In the LTE system, the handover can be classified as an intra-EUTRAN handover and inter Radio Access Technologies (RAT) handover. Inter RAT handover refers to mobility between LTE and 3GPP earlier tech-
nologies, such as GSM and WCDMA. According to the 3GPP standard [12], the intra-EUTRAN handover consists of

- Inter eNodeB handover without MME relocation and without SGW relocation
- Inter eNodeB handover without MME relocation and with SGW relocation
- Inter eNodeB handover with MME relocation and without SGW relocation
- Inter eNodeB handover with MME relocation and with SGW relocation.

Ericsson MPG supports all of the above handover procedures, but IxLoad currently only supports inter eNodeB handover without MME relocation and without SGW relocation, so the thesis work will focus on this scenario.

From the perspective of the interface, the inter eNodeB handover without MME relocation and without SGW relocation can be divided into an X2-based handover and an S1-based handover. An X2-based is shown in Figure 3-5 and the UE1 performs handover from eNodeB1 to eNodeB2 via the direct interface X2 between two eNodeBs. Figure 3-5 also illustrates that UE2 is handed over from eNodeB3 to eNodeB4 using the
interface S1 and eNodeB3 and eNodeB4 shares the same MME2 and SGW1. This is an S1-based handover without MME relocation and without SGW relocation.

The X2-based handover can be divided into three phases: handover preparation, handover execution and handover completion. In the preparation phase, the channel measurements including the received signal strength and carrier to noise ratio are periodically transmitted from UEs to the serving eNodeB via an uplink with the Radio Resource Control (RRC) signals. The source eNodeB makes decisions as to whether it is necessary to initiate a handover operation. Once the source eNodeB determines the handover, the necessary information for the handover preparation is sent to the target eNodeB, which then starts to prepare the handover and replies with an acknowledge message, which contains the information for the direct forwarding tunnels.

During the execution phase, the downlink and the uplink of the UE’s data are forwarded from the source eNodeB to the target eNodeB with direct forwarding tunnels via the X2 interface. The signaling procedure of the handover completion phase is shown in Figure 3-6, where the solid line represents the control plane messages and the dotted line stands for the user plane messages. During the phase of the handover completion, the downlink data is still forwarded by the source eNodeB,
but the uplink data starts to be uploaded via the target eNodeB. To inform SGW that UE has changed cell, the target eNodeB sends a Path Switch Request to MME, which then sends a Modify Bearer Request message to SGW. The Modify Bearer Request message contains the target eNodeB address and TEID for the downlink user plane. SGW replies MME with a Modify Bearer Response message. Then SGW begins to deliver the downlink packets to the target eNodeB with the newly received address and TEID. Thus the uplink and the downlink of the handover user are successfully switched to the target eNodeB. End Marker is transmitted to the Source eNodeB to inform the source eNodeB that the downlink path has already switched to the target eNodeB. The source eNodeB also sends the End Marker to the target eNodeB. It means that there will be no user data forwarded from the source eNodeB. In a relatively short time the direct forwarding tunnels between two eNodeBs are released, after the target eNodeB receives the Path Switch Acknowledgement. It indicates that the handover is over.

For the S1-based handover without MME relocation and without SGW relocation, the simulation results show that IxLoad is not compatible with MPG. Ixia engineer has already defined it as a new feature, which will be developed later. It can therefore not be verified in this thesis, but will form part of some future work.

### 3.2 Quality of Service

Mobile communication operators intend to provide multiple services, including high bandwidth services, such as file sharing, delay sensitive services, such as voice call, and best effort services like browsing a website across their networks. These services which are provided for all the subscribers are sharing the numbered radio and core network resources. It is essential for operators to make sure that each service running through these sharing resources offers, as a minimum, an acceptable user experience. Meanwhile the operators would like to provide different treatment for the identical service to the end users of different subscription, forming a revenue perspective. As a consequence, QoS makes it possible for the operators to be able to differentiate services and subscribers, while controlling the performance experienced by a certain service or subscriber group.

3GPP standards specify QoS requirements for the traffic between UE and PGW in LTE networks. The traffic between UE and PGW is transported via logical transport channels. Each logical transport channel is
built up by one EPS bearer. Each EPS bearer is related to some QoS parameters, which states the properties of the transport channel and the implication of this is that all the traffic through the same transport channel has identical QoS treatment.

<table>
<thead>
<tr>
<th>GCI</th>
<th>Resource Type</th>
<th>Priority</th>
<th>Packet Delay Budget (NOTE 1)</th>
<th>Packet Error Loss Rate (NOTE 2)</th>
<th>Example Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GBR</td>
<td>2</td>
<td>100 ms</td>
<td>10^{-5}</td>
<td>Conversational Voice</td>
</tr>
<tr>
<td>2</td>
<td>GBR</td>
<td>4</td>
<td>150 ms</td>
<td>10^{-5}</td>
<td>Conversational Video (Live Streaming)</td>
</tr>
<tr>
<td>3</td>
<td>GBR</td>
<td>3</td>
<td>50 ms</td>
<td>10^{-5}</td>
<td>Real Time Gaming</td>
</tr>
<tr>
<td>4</td>
<td>Non-GBR</td>
<td>5</td>
<td>300 ms</td>
<td>10^{-5}</td>
<td>Non-Conversational Video (Buffered Streaming)</td>
</tr>
<tr>
<td>5</td>
<td>Non-GBR</td>
<td>6</td>
<td>300 ms</td>
<td>10^{-5}</td>
<td>Video (Buffered Streaming)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TCP-based (e.g., e-mail, chat, ftp)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>File sharing, progressive video, etc.</td>
</tr>
<tr>
<td>7</td>
<td>Non-GBR</td>
<td>7</td>
<td>100 ms</td>
<td>10^{-5}</td>
<td>Video (Live Streaming)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Interactive Gaming</td>
</tr>
<tr>
<td>8</td>
<td>Non-GBR</td>
<td>8</td>
<td>300 ms</td>
<td>10^{-5}</td>
<td>Video (Buffered Streaming)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TCP-based (e.g., e-mail, chat, ftp)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>File sharing, progressive video, etc.</td>
</tr>
<tr>
<td>9</td>
<td>Non-GBR</td>
<td>9</td>
<td>300 ms</td>
<td>10^{-5}</td>
<td>Video (Live Streaming)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Interactive Gaming</td>
</tr>
</tbody>
</table>

Table 3-1 Standardized QCI characteristics [24]

Each EPS bearer is associated with two QoS parameters: QoS Class Identifier (QCI) and Allocation and Retention Priority (ARP). QCI not only points out real-time traffic or non-real-time traffic that the bearer carries as shown in Table 3-1, but also indicates how to treat IP packets transmitted by the bearer on the user plane. The control plane treatment of the bearer is determined by ARP.

Based on whether a certain amount of bandwidth is reserved for a bearer, the bearers can be divided into Guaranteed Bit Rate (GBR) bearers and non-Guaranteed Bit Rate (non-GBR) bearers. With the exception of QCI and ARP, GBR bearers have two other QoS parameters: GBR and Maximum Bit Rate (MBR). 3GPP release 8 defines that MBR should be set equal to GBR. As traffic delivered to MBR on a GBR bearer is exceeded, extra packets may be thrown away. In addition to QCI and ARP, non-GBR bearers have AMBR parameters for both downlink and uplink, including UE-AMBR and APN-AMBR, in order to limit the total traffic via non-GBR bearers. UE-AMBR defines the maximum total bit rate for all the non-GBR bearers for a single user. APN-AMBR specifies the maximum total bit rate provided by non-GBR bearers for a single user across a particular APN.
3.2.1 PCEF Function

As introduced in Section 2.2, PCRF is the policy and charging control element. It is responsible for making policy and charging control decisions. The Policy and Charging Enforcement Function (PCEF) is carried out by MPG. The communication between MPG and PCRF is using the Diameter protocol of Section 2.3.2 via the Gx interface. PCRF and MPG provide mutual support for each other. On one hand, MPG requires PCRF to provide the information relating to the policy decisions, such as MBR, APN-AMBR. These decisions will be implemented in MPG and translated into a GTPv2-C message Create Session Response or an Update Bearer Request to inform MME. On the other hand, PCRF also relies on the traffic information of UEs in order to make proper decisions. MPG monitors the UE traffic and sends feedback measurements to PCRF.

In addition, MPG reports the resources usage of UEs to Offline Charging System (OFCS) and Online Charging System (OCS). According to reports, OCS implements credit management for pre-paid UEs. The credit of post-paid UEs is managed in OFCS. The Charging Data Records (CDRs) is generated based on the charging events of reports and is delivered to the billing center.

3.2.2 QoS Handling

MPG attains QCI, ARP from policy decisions. If QCI is smaller than 5, it indicates that UE is applying for a GBR bearer. As long as resources are available, MPG allocates GBR bearer to UE, while reserving a fixed amount of bandwidth for UE. On the contrary, MPG assigns a non-GBR bearer to UE with the QCI that is larger or equal to 5. UE with a non-GBR bearer may suffer packet loss, since there is not guarantee as to how much the traffic non-GBR bearer can afford. ARP contains the information how important the resource request is. In the case where there are insufficient resources, MPG decides whether the resource request should be rejected in accordance to ARP. Depending on the bearer type (GBR or non-GBR), PCRF instructs MPG about the value of MBR or AMBR respectively. It may result in a downgrading or upgrading of the requested bearer QoS. MBR and AMBR are the upper bound for a GBR bearer and a non-GBR bearer. Usually MPG is in charge of removing the extra packets beyond the upper bound of the downlink. Bandwidth limitation of the uplink is taken care of by eNodeB.
3.3 **Deep Packet Inspection**

One key functionality of MPG is in supporting Deep Packet Inspection (DPI) and service classification to the user plane traffic. Packet Inspection is a networking technology that Internet Service Providers use to monitor subscribers’ data traffic, mediate its speed and improve the network security [25]. It is carried out, when the packet passes an inspection point. DPI enables the observer not only to collect statistics, or search for viruses and spam, but also to search for controversial content or illegally downloaded, copyright-protected material [26].

Basically there are three levels of packet inspection, the header inspection, deep inspection and heuristic inspection. Header inspection, also called Shallow Packet Inspection, inspects IP packets only up to Layer 4, mostly based on the L3/L4 information. For instance, the header information such as the source and destination of IP addresses and the port numbers and the transport protocol are analyzed. Because of these five parameters, it is also known as the 5-tuple inspection. Figure 3-7 specifies how packet inspection operates on the headers in the different layers.

![Figure 3-7 Different levels of packet inspection [27]](image)

Deep inspection is an extension of header inspection, which examines not only the headers but also the packet payload, and takes advantage of the protocol based analysis (Layer 7), which is performed in parallel with the header inspection, in order to minimize any impact on the performance [27].
In the payload filtering, one possible way to analyze packets is to match the strings contained in the Layer 7 header by using a certain operator or a combination of operators. String match involves the search for a sequence of textual characters or numeric values within the contents of the packet. For example, Figure 3-8 gives an IP packet of HTTP GET page. In order to inspect this packet and place it into the correct classification for the user tracking or charging, it could be performed by matching the string “kazaa” in the User-Agent field. In addition to this contain case, more operators could be defined and combined, like is, not is, contains, or not contains, starts with etc. With different application protocols, MPG has defined multiple operations to inspect all the packets.

However, DPI is not sufficient for all applications. Heuristic inspection, shown in Figure 3-7, can be used to classify the traffic over proprietary protocols and P2P encryption. Heuristics inspection analyzes the packet information based on the IP flow characteristics in addition to the packet content, e.g. flow rate, bursts, connections, etc. It supports most of the P2P protocols, the Skype traffic and some other VoIP traffic etc.

The service classification function is to configure a set of classification rules that identify the individual services of the UE payload to match the packet inspection results. For each flow, MPG assigns a unique Service Data Flow ID (SDF-ID). In the packet inspection process, an incoming packet flow would be inspected with either a header inspection or a heuristics inspection, and then the specific flow will be classified by the defined rule and assigned an SDF-ID. In case the packet inspection (all the three types) is not successful, the flow will be classified to the default group configured as a default SDF-ID.
4 IxLoad

IxLoad is a full-featured layer4-7 test tool that provides real-world traffic emulation test of voice, video, and data network and components. It is experienced in the stateful protocol emulation and supplies multiple simulations of network services, such as the subscribers, clients and servers.

This chapter will give a general overview of IxLoad. Some new features of IxLoad also will be presented in this chapter, such as eGTP stack, network command over eGTP, QoS enforcement and the multiple APNs. As it is a continuous thesis, some more information of IxLoad can be referred to in [23].

4.1 IxLoad Overview

IxLoad is a scalable tool for testing the converged multiple services, application delivery platforms as well as security devices and systems. It emulates data, voice, and video subscribers and associated protocols to ensure quality of experience (QoE) [29]. It supports various Internet protocols such as HTTP, P2P, FTP, SMTP, POP3 and voice protocols such as SIP, MGCP, PSTN, SGMP and video protocols including RTSP, IGMP, Silverlight and so on. IxLoad simulates the client and server, with the device under test (DUT) in between. Ixia chassis are populated with hot-swappable test interface cards and each test port is equipped with an independent processor and substantial memory. Ixia chassis is able to transmit and receive both the control and data plane traffic throughout DUT. In this thesis, DUT is MPG.

IxLoad server programs consisting of IxOS and IxExplorer are running on the chassis. IxOS is responsible for the startup and management of the Ixia chassis, while IxExplorer provides the resource management for the hardware, as well as filtering and capturing functions [30].

IxLoad client program can be installed in any windows computer. It provides a user friendly graphic user interface (GUI) to configure each test case and connects to the IxLoad chassis with the Ethernet port via IP. Figure 4-1 shows the general view of IxLoad client GUI. There are four main windows, Test Configuration, Statistics, Data Miner and Analyzer.
In the Test Configuration window, user can configure network stacks and detailed network parameters both on originate and terminate. Each application is able to be added to layer 7 in order to simulate pure or mixed traffic with user defined profiles. Greater detail of LTE traffic configuration will be mentioned in Chapter 6. IxLoad has offered a set of test objectives, such as subscriber, simulated users, connection per second, concurrent connection, throughput, etc. A user could choose one test objective depending on the test scenario. In addition, the Timeline for the test case must be specified. It includes the ramp time, sustain time and ramp down time. Figure 4-2 shows an example of 8000 simulated users. The ramp up phase takes 8 seconds at a speed of 100 sessions per second and ramp down takes 10 seconds at a speed of 80 sessions per second. The test case will take a total of 38 seconds in which 20 seconds is the sustain time.
Port assignment is the most important step to finalize the configuration of a test case. The right ports for the client and server must be assigned by connecting to the Ixia chassis, which will go through with the emulated control plane and data traffic.

In order to run the test, IxLoad provides several buttons, shown as in the red rectangle in Figure 4-1. Apply configuration and release configuration buttons are responsible for loading and releasing a test case. Two stop buttons are settled in IxLoad in order to handle different stop scenarios. Stop button will stop the running test by force, while the stop with ramp down button will stop the test step by step with the ramp down value configured in the Timeline.

In the statistics window, the realtime statistics during the test are shown as tables, plots or charts. It is a visible way to observe how the test case is running in real time. The statistics are classified in groups by layers in the network model, such as the layer 2-3 group, the eGTP group and different the application groups, presented in Figure 4-3.
In the analyzer window, the control and data plane packets are captured in case the analyzer is enabled before the test case is running. IxLoad analyzes the capture by displaying the ladder diagram, which indicates the interaction between source and destination, shown in Figure 4-4. IxLoad Analyzer is useful in order to trace some network issues during the test.
4.2 IxLoad eGTP Stack

IxLoad supports the eGTP protocol, which makes it possible to test both the control plane session and the user data traffic in the EPC network. IxLoad provides an eGTP-S1/S11 eNB/MME plug-in, which is the emulator of UE, eNodeB and MME and it provides the S1 and S11 interfaces in the LTE network.

IxLoad implements the eGTP functionalities with a series of parameters configurations in the form of a GUI. Normally, the eGTP-S1/S11 eNB/MME plug-in is created over the IP stack and the following Figure 4-5 shows the stack configuration view of IxLoad release 5.30.

![Figure 4-5 eGTP stack view](image)

In the User Equipment stack, shown in figure 4-6, the real world activities can be assigned with simple configuration. Take the UE basic information as an example; the International Mobile Subscriber Identity (IMSI) number and the Mobile Subscriber ISDN Number (MSISDN) number are required in the configuration to identify a specific user. In additional, the mobility parameters are embedded in the UE level. The mobility checkbox in GUI must be enabled in case of a handover activity without network command. In addition, the mobile path and mobility interval should be specified in a handover case.
To create an eGTP session, another major configuration is the APN stack. Apart from the PGW IP address, QCI value should be configured and the Maximum Bit Rate for Downlink and Uplink (MBRD and MBRU) that GBR bearers are expected to be provided are also need to be set. In addition, for the non-GBR bearer, the APN level QoS in a similar manner to that for the AMBR for downlink and uplink which are mandatory for the user plane traffic. The Time line options in the APN setting indicates the lifetime of a bearer, which is only available in the Net Traffic model. It can be used in the busy hour functionalities to limit each bearer’s lifetime.

IxLoad emulates the S1-MME interface by allocating a parent MME in the eNodeB stack, as illustrated in Figure 4-7. In the eNodeB connectivity, the options should be considered in different handover scenarios. If checked, the two eNodeBs will have a direct forwarding path with the X2 interface.
IxLoad also simulates the S11 interface by configuring the SGW IP address and count in the MME stack. In the MME stack in Figure 4-8, other parameters need to be considered are the T3 and N3 timers. For instance, the create session T3 indicates the number of seconds to wait for the Create Session Response message. The create session N3 is the maximum number of retransmissions that will be permitted for a Create Session Request message [23]. Both the T3 and N3 values are supposed to match the configuration in MPG; otherwise it may cause duplicate messages.

IxLoad supplies a group of eGTP statistic viewers, shown in Figure 4-9. The basic session and bearer information during the runtime of the test case can be found in the eGTP statistics. However, the majority of the statistics relate to the control plane counters. The user could also check the real time status of each connection by the drilldown feature with either per PDN or per Session. In the session management statistics window, all the S11 control plane massages are counted, including all
the retry messages. In the latest version of IxLoad, it supports the eGTP control plane tunnel activation and deactivation rate, the user plane data rates, as well as the packet rate of GTP-U. In the real world, the packet rate (packet per second) would be of greater concern than the bit rate (bit per second).

Furthermore, eGTP statistics windows output the detailed information of both the handover management and the Tracking Area Updates. All the statistics data is recorded to the CSV logs as the test results.

4.3 **Advanced Subscriber Model**

As is known, IxLoad provides Net Traffic and Subscriber models in the client side. Net Traffic model supports all available protocols and test objective types, but the network parameters are the same for all Net Traffics. A subscriber is a special type of Net Traffic that simulates patterns in relation to residential customers who receive voice, video and data service over a single physical connection [23]. It only supports some of the protocols and the available test objective is Subscriber, but it is more flexible to use some commands in order to configure the traffic model than in Net Traffic model. Take Figure 4-10 as an example, the subscriber will trigger each event sequentially following the arrows. The subscriber firstly opens a web browser and then downloads a file via FTP. the subscriber opens a mail page to check the mails. After a while, the subscriber browsers another web page and downloads files via BitTorrent. The subscriber model makes it possible to simulate the real world user traffic patterns.
IxLoad develops more additional functions for the advanced subscriber model in the latest IxLoad release 5.30EB than the previous versions, such as the network commands for the eGTP stack.

Within the eGTP control plane panel of the advance subscriber model, there are a list of network commands for the eGTP stack. The user is able to self define different commands based on different test scenarios. IxLoad handles all the commands sequentially in loops. All the eGTP commands are shown on the left side of Figure 4-11. These commands will be utilized in the Chapter 6.
Create Session/Delete Session pair of messages is the basic eGTP session creation and deletion procedure. In the session setup, Create Session command will construct a Create Session Request message for signaling and a Modify Bearer Request for the downlink packets transmission. In the session teardown, Delete Session command generates the Delete Session Request in MME. These two commands are the fundamentals of an eGTP scenario. Figure 4-11 also presents a simple HTTP example running over an eGTP stack with the network commands. UE initiates an eGTP session by sending the Create Session command to MPG. After the creation, UE will trigger an HTTP get page event and wait for the HTTP server response. The duration of waiting is defined in the Think command. At the end, UE deletes the session by sending the Delete Session command. All the activities are processed sequentially. The user is able to define even more complicated scenarios according to the real network.

Bearer modification command is used in modifying the APN level AMBR. Modify bearer command will be sent on the S11 interface by MME to the SGW and on the S5/S8 interface by the SGW to the PGW as part of the HSS initiated Subscribed QoS Modification procedure. The command contains the updated AMBR values both for downlink and uplink.

Enter/Exit Idle State commands pair triggers the subscriber to enter/exit the ECM-IDLE state. A Release Access Bearer Request will be sent on the S11 interface by the MME to the SGW as part of the S1 release procedure, while a Release Access Bearer Response will be sent on the S11 interface by the SGW. This pair of commands could be useful in the Tracking Area Update and Downlink Data Notification (DDN) scenarios.
Handover/TAU command is able to initiate the handover or TAU procedure depending on the UE state. If the UE is in the ECM-CONNECTED state, a handover will be performed. On the other hand, if the UE is in the ECM-IDLE state, a TAU procedure will be triggered. For the different types of handover events, the command will provide different eGTP control plane messages.

Wait for paging command transits the state of UE from the ECM-IDLE state to the ECM-CONNECTED state and initiates the execution of the network triggered service request procedure, after receiving a waking up message (Downlink Data Notification Request) [23].

4.4 Dynamic QoS Enforcement

Dynamic QoS enforcement based on eGTP is supported by IxLoad. It includes QoS enforcement and dynamic QoS implicitly. With dynamic QoS, IxLoad is capable of adjusting the QoS parameters of UE according to received the Update Bearer Request message. With QoS enforcement, IxLoad is able to negotiate with MPG about the QoS configuration and the extract negotiated results from Create Dedicated Bearer Request, Create Bearer Response. Once the QoS enforcement is activated, IxLoad launches the QoS negotiation with MPG through the whole session lifetime, until simulation is ended.
Figure 4-12 depicts the QoS setting window, which consists of a QoS to Type of Service (ToS) mapping and Dynamic QoS Enforcement configuration. eNodeB and MPG execute QoS mechanism based on the bearer level, which is beyond the transport level. However, the QoS control of transmission between eNodeB and MPG works on the transport level [31]. In the simulation, eNodeB is emulated by IxLoad on the client side, so IxLoad needs to transform QoS parameters of packets from the bearer level QCI to the transport level ToS, before it sends the packets out. In order to make the simulation more realistic, IxLoad offers a QoS to ToS mapping table and allows the user to set the QoS translation pattern. Dynamic QoS enforcement aims at the bearer level and when it is enabled, IxLoad implements the QoS enforcement for the uplink with GBR and APN-AMBR. The downlink of the QoS enforcement is taken care of by MPG. It affects all the users simulated by IxLoad in accordance with the characteristics of the bearers allocated to them. The users of the GBR bearers could experience packets loss, if traffic delivered on the GBR bearer exceeds MBR. The users owning non-GBR bearers will lose packets as soon as the throughput of the non-GBR bearers passes a particular APN which is beyond APN-AMBR. So far, IxLoad has not yet supported UE-AMBR. APN-AMBR is the only QoS parameter that can be used for restraining non-GBR bearer traffic. On the one hand, QoS enforcement ensures that the network is able to differentiate services as well as subscribers, while controlling overall throughput. On the other hand, it results in increased TCP transmission and UDP packet loss.
4.5 Multiple APNs

APN is the name of access point for data connection in GSM/UMTS/EPC network. APN is composed of Network Identifier and Operator Identifier [32]. Network Identifier points at external packet data network which GGSN/PDN Gateway is connected to. Operator Identifier is an optional parameter for APN, and it consists of Mobile Country Code (MCC) and a Mobile Network Code (MNC). Every operator is identified with MCC and MNC. For example, the APN of China Unicom is “Uniwap.mnc01.mcc460.gprs”. “Uniwap” is Network Identifier and “mnc01.mcc460.gprs” is Operator Identifier. 460 is the MCC of China and 01 is MNC of China Unicom. In practice, subscribers only need to configure Network Identifier such as “Uniwap” as APN in the mobile devices, since core network is able to obtain an Operator Identifier from IMSI.

![Figure 4-13 UE with multiple PDN connections [8]](image)

Usually operators would like to define multiple APNs, serving for different PDN connectivity services. Each APN is corresponding to a PDN. A particular desired service is obtained by accessing a particular PDN. For instance, one end user signs a contract with the operator, which promises to provide a PDN connection service to Internet. As a result, the operator provides the end user with the information of an APN associated to Internet PDN. During the session setup, MME takes responsibility for selecting PGW for data connection, referring to APN provided by end user or subscription information of APN from HLR. PGW is in charge of establishing a PDN connection between the end user and Internet PDN. Then this user can access all the available services on the Internet including chatting with MSN. The end user is also
allowed to access multiple PDNs at the same time. Once the end user subscribes to the PDN connectivity services of Internet, IP Multimedia Subsystem (IMS) as well as corporate network with the operator, the user can work everywhere at any time. As long as a mobile device can receive radio signals, the end user is able to not only arrange a video conference provided by the operator via IMS with customers but also use a browser website to search for the newest business information on the Internet and read company internal documents by accessing a corporate network outside company simultaneously, as shown in Figure 4-13.

The new feature Multiple APNs is introduced in the release 5.30 of IxLoad to simulate above scenarios. A single subscriber is allowed to own more than one APN, which is similar to the case for a real network. IxLoad provides a setting window to append a number of APNs if required. In commercial networks, operators choose proper selection model to indicate the origin of APN according to the security needs. To make session setup procedure closer to the real world, three selection models corresponding to 3GPP TS 29.060 are offered by IxLoad, as shown in Figure 4-14. If the selection model is “MS provided APN, subscription not verified”, the APN used for identifying PGW is provided by mobile device and Home Location Register (HLR) does not verify it. The second selection model is “Network provided APN, subscription not verified”. The network specifies a default APN in the case where UE does not provide it. It also does not require verification from HLR. When the selection model “MS or network provided APN, subscribed verified” is chosen, HLR should verify the user’s subscription associated to APN provided by mobile device or network [33].

![Figure 4-14 Multiple APNs setting and selection modes](image-url)
5 Test Environment

5.1 Back to Back
IxLoad not only supports emulating data, voice and associated protocols but can also simulate network elements. In the back to back test, two ports of IxLoad are interconnected. One port is emulating subscribers, eNodeB as well as MME and the other port is simulating SGW, PGW and PDN. Both IxLoad ports exchange control plane signals and data traffic via GTP-C and GTP-U protocols, shown in Figure 5-1.

The reason why two ports communicate with GTP protocol is because of the stack structure of EPC. As mentioned in Section 2.3.1, MME is connected to SGW via S11 interface in the control plane and eNodeBs are linked with SGW via S1-U interface in the user plane. For EPC, GTPv2-C and GTPv1-U are used to encapsulate control signals of S11 interface and data traffic of S1-U interface respectively. The emulation of IxLoad strictly follows 3GPP standards. Moreover, the traffic between MME/eNodeBs and SGW can be collected by capturing two ports by means of the IxLoad analyzer, which provides another way of looking into the status of network elements, in addition to statistics.

![Diagram of IxLoad back to back test environment](image-url)

5.2 MPG Test Environment
In MPG test environment, MPG which is a combination of SGW and PGW performs as the Device under Test. IxLoad simulates the client with eGTP stack including the MME, eNodeB and UE functionalities by the S1/S11 interfaces and simulates the packet data network as server by the SGi interface, as shown in Figure 5-2.
In the physical connection, both MPG and Ixia chassis are connected to the same HP switch in different ports. VLAN tagging is enabled in both Gi interface and S1/S11 interfaces. The Ixia chassis is able to connect to the switch with a 1G physical port, also, it can connect to the switch with a 10G port which is aggregated by 12 1G physical ports. The 10G aggregated port is used in some performance tests. Furthermore, a mirror is setup to monitor the control plane and user plane traffic on the S1/S11 and Gi interfaces. It is capable of using the Wireshark or Tcpdump tool to capture the packets that are helpful in tracking some issues.
6 Test Scenarios

In this chapter different test scenarios in MPG will be presented.

6.1 Capacity Evaluation

6.1.1 GTP-C Transaction Rate

Purpose:

IxLoad is professional not only in the stateful traffic simulation, but also in the signaling test. With eGTP stack, IxLoad can manage up to 500 GTP-C transactions per second in a 1G port and 6000 GTP-C transactions per second in an aggregated 10G port. For this purpose, this test case is designed to verify the actual performance of IxLoad in GTPv2-C transaction handling.

Expected Result:

In a 1G port test, session activation rate should reach 500 tunnels per second in ramp up stage and session deactivation rate should reach 500 tunnels per second.

In a 10G aggregated port test, the session activation and deactivation rate are supposed to reach 12 times the result for 1G port.

Configuration:

Since this is a signaling capacity case, it is better to provide a best configuration for the eGTP session activation and deactivation. Hence, one node controller and several session controllers are setup in MPG. The create/delete session T3 and N3 need to be configured in MPG. An APN without any Service Aware Charging and Control (SACC) is used in this case to avoid any other CPU usage. The limitation for the user allocation in MPG should be configured to more than 100K in the APN level.

In IxLoad 5.30.200.10 EB, only the client network is sufficient for setting up the S11 signaling. On the application layer, set the commands as simple as possible, because there will be no user plane traffic during the test. In this test case, a HTTP GET command with 1 concurrent connection is configured, as shown in Figure 6-1.
The test objective is simulated user and the user count should be set to the maximum number for one port (100K simulated users) to make the ramp up time line longer. In the global setting that has been shown in Figure 6-2, both the initiation rate and the release rate are set to 500 procedures per second. Meanwhile, the ramp up value in timeline table should be configured as 500. With 100K users, the expected ramp up time is about 200 seconds.

It is important to set the same create/delete session T3 and N3 values in IxLoad and MPG, because the control plane traffic will doubtless be busy in the ramp up stage. The consistent T3 and N3 values will reduce the possibility of a retransmission issue.

Verification and Results:

When the test case is running, it is better to monitor the CPU load of each session controller in MPG. If the CPU load is up to 100%, MPG would be unable to handle more session activation events and there would be session failures.
From the IxLoad eGTP statistics, all the bearers are created successfully after about 3 minutes and 20 seconds, during this period, the bearer activation rate in the Rate Control window is sustained at about 500 tunnels per second and goes down to 0 when the ramp up is finished. After a while, click the blue button to stop the test case with a ramp down value, the deactivation rate will be active in turn and it sustains 500 tunnels per second. For the ramp down timeline, the key point is to make use of the stop with ramp down button, which sends the delete session requests to MPG one by one instead of merely releasing all the sessions. The powerful button is available in the 5.30EB build or later.

From MPG counters, the active bearer counter is increasing rapidly with around 500 bearers per second at the ramp up stage. During the sustain time, the active bearer number is 100K without any failures. At the ramp down stage, the active bearer counter decreases with the rate at about 500 bearers per second and the deactivated bearer counter is increasing, reaching 100K by the end of this test.

From the results in Figure 6-4, both activation rate and deactivation rate can achieve 500 tunnels per second. However, the activation rate does not continue to be as stable as was the case at the beginning when the active session number was high, because the continuous high load will cause high CPU usage in MPG which will have an impact on the activation rate to a certain degree.

As there is a limited configuration in the current MPG that has been used for verification, it is not possible to verify 10G port objective in this thesis.
6.1.2 Max user with Different L7 Applications

Purpose:

IxLoad supports most of the L7 protocols over IP stack, but for eGTP stack and currently IxLoad supports the FTP, HTTP, POP3, SMTP, IMAP and Video on Demand (VOD). For each port and load module, IxLoad has a limited memory to initiate all the users. This series of test cases are for verifying the maximum users that are supported both with the 1G port and the 10G aggregated port in the MPG environment.

Expected Result:

From Ixia website [2], the maximum numbers of UEs in each application that IxLoad supports are listed in Figure 6-1.

<table>
<thead>
<tr>
<th>Test Case</th>
<th>1G Port</th>
<th>10G Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max UEs with Default Bearers – ftp</td>
<td>53.5 K</td>
<td>642 K</td>
</tr>
<tr>
<td>Max UEs with Default Bearers – http</td>
<td>100 K</td>
<td>1.2 M</td>
</tr>
<tr>
<td>Max UEs with Default Bearers – Video on Demand (VOD)</td>
<td>2 K</td>
<td>24 K</td>
</tr>
<tr>
<td>Max UEs with Default Bearers – POP3</td>
<td>86.5 K</td>
<td>1038 K</td>
</tr>
<tr>
<td>Max UEs with Default Bearers – SMTP</td>
<td>90 K</td>
<td>1080 K</td>
</tr>
<tr>
<td>Max UEs with Default Bearers – IMAP</td>
<td>74 K</td>
<td>888 K</td>
</tr>
</tbody>
</table>

Table 6-1 Expected results in the maximum user

Configuration:

In MPG, two session controllers are enabled to ensure that there is sufficient capability for signaling handling. A simple APN without any charging or control services is running in this test case to avoid any limitation.

In IxLoad 5.10.152.64 EB, all the applications are running over the eGTP stack. The objective for this case is to test whether IxLoad can reach the maximum user amount in the MPG node, therefore the configuration of the eGTP tunnel is the simplest one and is similar to using only one MME and one eNodeB without any mobility enabled. In the TCP stack, all the values such as buffer size are set as default. In the 10G case, the ports must be aggregated in the software level before being assigning to the client and server Net Traffic.
The configuration of each application is set to the basic default value to achieve the goal of maximum users:

**FTP:** one get command is set in the client side and the dummy file is set to default 4096b which is equal to the transmit/receive buffer size.

**HTTP:** HTTP version 1.1 is selected in this case, and each user has 1 concurrent TCP connection. The page size in the get command is 1b.

**POP3:** a simple mail get command is configured.

**SMTP:** one send command with simple mail message is configured.

**IMAP:** the get mail command is added to this application with user name “user[00-]” and password “password[00-]”.

**Video on Demand:** video client is set in this test case and a “play media” command is added to make it a simple VOD model to achieve the test objective.

Figure 6-5 shows the Layer 7 commands that are used in this test case.

![Figure 6-5 Different application commands in the maximum user case](image)

**Verification and Results:**

The same procedure is processed to verify whether IxLoad is able to achieve the maximum users in each case. FTP is taken as an example. It is simulated in a 10G aggregated port and each port will be able to accommodate 53.5K users. To verify the results, it is necessary to check both GTP-C and GTP-U statistics.

With the ramp up value of 120, IxLoad initiates 120 eGTP sessions per second and loads to 12 ports equally. The statistic of eGTP - General contains the information about the number of sessions that have been initiated and created successfully. If any failure occurs, the counter for sessions failed will be increased accordingly. The results of eGTP General window show that there are 53.5K eGTP sessions created on each port. The amount of users achieves the expected maximum value of
642K. The FTP client Objective window shows that the signaling setup of simulated users has been completed, as shown in Figure 6-6. On the other hand, from the MPG counters, there are 642K bearers activated in total with no failures.

There are 12 test cases simulated in total and each case achieves the maximum simulated users, as Table 6-1 shows.

### 6.1.3 IxLoad 1G Port Throughput Test

**Purpose:**

The throughput and latency are the key performance indicators for wideband services. One of advantages of IxLoad is to simulate the behaviour of large number of users for several kinds of applications and generates heavy throughput. This test case is to check the highest throughput a 1G port of IxLoad can achieve for a certain application under the condition involving the maximum number of users in Ericsson MPG.

**Expected Results:**

The throughput objective of 1G port is listed in Table 6-2, which is published in the Ixia website.
Configuration:

For each test case in Table 6-2, the configuration of procedure is almost the same, so only FTP applications are picked to demonstrate how this work is performed. In order to present greater detail of IxLoad 1G port performance, the identical test case is run in the back to back and MPG environments.

First throughput test is implemented in the back to back environment of Section 6.1. Then the throughput test is conducted on the IxLoad-MPG environment of Section 6.2. In the back to back test, UE, eNodeB, MME are emulated by IxLoad client and SGW, PGW and application server are simulated by IxLoad server. For the IxLoad-MPG test environment, SGW and PGW are replaced by MPG. The traffic between IxLoad client and IxLoad server is passed through MPG.

Two independent ASM 1000XMV 12 load modules of IxLoad chassis are used in the throughput test. Each ASM 1000XMV 12 has twelve 1G ports and one 10G port. In order to avoid a physical connection shift, IxLoad is connected to physically switch with 10G port. In this test case, traffic is generated from a 1G port and then forwarded to the 10G port. However, it also can be tested by connecting 1G port to switch directly. If so, the cable must be changed in the next test case.

There are two key parameters impacting upon the throughput performance. The former one is the file size and the latter one is the buffer size. The physical memory space of IxLoad port is fixed. With the increasing user numbers, the buffer size for each user is decreased. As a result, the file size should be scaled according to the buffer size. How to distribute constant memory to each user and how to choose file size can be considered as a mathematic model. The objective value is the maximum throughput. The constraint values are memory and user number. This
model can be solved with convex optimization theory, but it is beyond the scope of this thesis work. Thus the recommended value provided by Ixia is preferred in this thesis.

Figure 6-7 TCP buffer size configure

Figure 6-8 File size of FTP application configuration

Figure 6-7 plots TCP parameters configuration screen. The TCP buffer size is divided into a receive buffer size and a transmit buffer size. Both values can be set respectively as shown in Figure 6-4, but an aspect which ought to be highlighted is that the buffer size of the client side and the server side should be coincident. The file size configuration is shown in Figure 6-8. The configuration interface is quite close to real surfing, and it is not only possible to pick the application server but also to choose the object/file (Different file has different size). These test cases are implemented with the parameters shown below.

- FTP with 25K users
  The weight of each file downloaded by users is 1048576 bytes. The behaviour of the users is to download files, so the buffer size for receive direction should be larger than that of transmit direction in order to obtain a larger TCP window size and thus achieve a higher throughput. The suggested values of TCP transmit buffer and receive buffer for the client side are 1024 bytes and 16384 bytes. The server side adopts the opposite approach. In other words, server uses TCP transmit buffer of 16384 bytes and TCP receive buffer of 1024 bytes.

- FTP with 50K users
The smaller size of file is applied to this scenario, which is 65536 bytes. The buffer size is decreased due to a doubling of the amount of users. The TCP receive buffer and transmit buffer on the client side are 4096 bytes and 1024 bytes. On the server side, the TCP receive buffer and TCP transmit buffer are 1024 bytes and 4096 bytes.

**Verification and Results:**

The same procedures are used to verify the results for FTP applications in two test environments, so only FTP application with 50K users in the back to back test environment is used to explain how it is verified. There will be a greater concentration will be on the results analysis.

At the beginning of the simulation, the simulated user number is increasing smoothly according to the ramp up value as in a real network. It is impossible to load 50K users simultaneously, because of hardware limitations. As long as the simulation is stable, the total user number should be equal to the value predefined. As Figure 6-9 has shown, the amount of users is fixed at 50K which is consistent with the number of active subscribers in MPG statistics.
Besides user number, the throughput is the most important value. It is evaluated based on the throughput statistics of IxLoad. The reason will be explained at the end of this section. IxLoad statistics provide two views in relation to the throughput results. The first view is on the application layer, illustrated in FTP Client – Throughput Objective of figure 6-10. The other is on the L2-3 layer, displayed in L2-3 Throughput Status of Figure 6-10. Comparing both views, it is clear that the throughput of L2-3 layer is larger than that of the application layer, since IP header and TCP/UDP header are taken into account in the L2-3 layer. IEEE802.3 formulates that the Maximum Transmission Unit (MTU) is 1500 Bytes. Considering that the header of TCP/UDP occupies 20/8 Bytes plus 20 Bytes IP header, the maximum size per MTU for application layer is only 1460 or 1472 Bytes.

The resource scheduling algorithm should also be investigated. Some scheduling algorithms such as the classic maximum rate, assign most of resources to the users who can bring more throughput and this results in hungry users. In other words, some users have excellent data services and others suffer poor services. As an advanced packets test tool, the resource scheduling algorithm is too confidential to be published. However, its fairness can be tested. Figure 6-10 lists the downlink and uplink
throughput of individual users, which proves that every user enjoys a similar download and upload data service. So it is a fair resource allocation scheduling implemented in IxLoad.

Figure 6-11 Throughput statistics of individual users

Figure 6-12 Comparison of throughput in back to back and MPG environments

Figure 6-12 illustrates that the same test case in the MPG test environment achieves a higher throughput because IxLoad server bears more overhead during the back to back simulation. In the back to back test, IxLoad server is not only required to simulate FTP server but also emulates the behaviour of SGW and PGW. As a consequence of decreased performance of IxLoad server, the lower throughput is generated to IxLoad client. For both test environments, the throughput is decreased with the increasing number of users. As the number of users rises, more CPU load is consumed for session management. Moreover, it causes the traffic to be more complex. IxLoad and MPG have to spend more CPU load on handling such traffic.
The data of Table 6-3 is collected from the statistics of IxLoad, after verifying the accuracy of the throughput statistics of IxLoad with MPG statistics. IxLoad statistics provide fantastic charts such as that displayed in Figure 6-10 to describe the real-time throughput and is the reason why this thesis chooses the statistics of IxLoad as data source for throughput assessment. To confirm that IxLoad is trustworthy, the throughput statistics of IxLoad are compared with the real throughput, which is estimated by MPG counters according to the expression that the throughput via MPG is equal to the accumulated incoming throughput counter of MPG between two time slots divided by the time interval shown in the timer counter. The comparison result shows that the throughput statistics of IxLoad is reliable. Therefore 1G port and 10G port throughput analysis are on the basis of the throughput statistics of IxLoad.

Table 6-3 shows the throughput test results of 1G port for six test cases. With the exception of the Application Replay with 25K UEs, all the test cases achieve the objective value. The reason why uplink throughput of HTTP is slightly lower than 83 Mbps is due to the configuration of the page size. More discussion will be continued in the following 10G port throughput test.

### 6.1.4 IxLoad 10G Port Throughput Test

**Purpose:**

The throughput performance of 1G port has already assessed in the last test case. This test case combines 12 1G ports to work together and hence generates up to 10G traffic. Two intentions are behind this test. The
capacity of IxLoad to generate the maximum throughput can be verified. The second target is to estimate the influence of TCP acceleration feature of IxLoad on producing huge traffic.

Expected Results:

The expected results in the 10G throughput test are shown in Table 6-4, which are verified in back to back environment from Ixia.

<table>
<thead>
<tr>
<th>10G Maximum throughput Test Case</th>
<th>Expected Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>P2P - 300K UEs with Default Bearers</td>
<td>6Gbps</td>
</tr>
<tr>
<td>P2P - 600K UEs with Default Bearers</td>
<td>3.9Gbps</td>
</tr>
<tr>
<td>HTTP - 300K UEs with Default Bearers</td>
<td>1 X 9.6Gbps (UL X DL)</td>
</tr>
<tr>
<td>HTTP - 600K UEs with Default Bearers</td>
<td>1 X 9.6Gbps</td>
</tr>
<tr>
<td>HTTP - 1200K UEs with Default Bearers</td>
<td>1 X 9.6Gbps</td>
</tr>
<tr>
<td>Application Reply - 300K UEs with Default Bearers</td>
<td>9.6 X 9.6Gbps</td>
</tr>
<tr>
<td>FTP - 300K UEs with Default Bearers</td>
<td>7.44G</td>
</tr>
<tr>
<td>FTP - 600K UEs with Default Bearers</td>
<td>3.2G</td>
</tr>
</tbody>
</table>

Table 6-4 Performance table in 10G port [2]

Configuration:

The configuration of 10G port throughput test is almost the same as that of 1G port throughput test. Port assignments are required to be changed. 12 ports are necessary for IxLoad client and IxLoad server respectively. The 10G aggregation mode should be checked, as shown in Figure 6-13. In the 10G aggregation mode, 10G traffic is generated with CPU of twelve 1G ports and outputted to a 10G port. Corresponding to 12 ports, the reserved IP address number for eNodeB, MME or application server must be adjusted to 12.
Test Scenarios

Yue Peng / Zhiqiang Tang

2012-09-07

Assessment of IxLoad in an MPG Environment

Figure 6-13 Port assignments for 10G aggregation mode

- HTTP with 300K users
  With TCP transmit buffer of 4096 and receive buffer of 40960, users download a webpage of 16K from HTTP servers, whose TCP transmit buffer is 40960 bytes and TCP receive buffer is 4096 bytes.

- HTTP with 600K users
  TCP transmit buffer of HTTP servers is 40960 bytes and the TCP receive buffer of HTTP server is 4096 bytes. For client side, the TCP transmit buffer is 4096 bytes and the TCP receive buffer is 40960 bytes. Page size is still 16K.

- HTTP with 1200K users
  The size of HTTP page is decreased to 4K. All the TCP buffers are set to 4096 bytes.

- P2P with 300K users
  1000 KB is chosen as the size of download file. P2P application utilizes BitTorrent protocol for file transmission. The size of the client TCP receive buffer and the size of the server TCP transmit buffer are equal to 16384 bytes. The size of client TCP transmit buffer and the size of the server TCP receive buffer size are 1024 bytes.

- P2P with 600K users
  The size of download files is 1000 KB. Files are transmitted via BitTorrent protocol. The TCP receive buffer and TCP transmit buffer of client are set to 8192 bytes and 1024 bytes respectively. Conversely the TCP receive buffer and the TCP transmit buffer of the server are configured to 1024 bytes and 8192 bytes.

- Application Replay with 300K users
  The buffer size is set exactly the same as that for HTTP with 300K users. The capture file is provided by Ixia.
Assessment of IxLoad in an MPG Environment
Yue Peng / Zhiqiang Tang

Test Scenarios
2012-09-07

- FTP with 300K users
  The buffer configuration is the same as FTP with 25K users in 1G port throughput test.

- FTP with 600K users
  The buffer configuration is the same as FTP with 50K users in 1G port throughput test.

Verification and Results:

Table 6-5 Maximum throughput test results of 10G port

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Expected Throughput</th>
<th>Test Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>P2P - 300K UEs with Default Bearers</td>
<td>6Gbps</td>
<td>6.45Gbps</td>
</tr>
<tr>
<td>P2P - 600K UEs with Default Bearers</td>
<td>3.9Gbps</td>
<td>4.33Gbps</td>
</tr>
<tr>
<td>HTTP - 300K UEs with Default Bearers</td>
<td>1 X 9.6Gbps</td>
<td>0.49 X 9.85Gbps</td>
</tr>
<tr>
<td>HTTP - 600K UEs with Default Bearers</td>
<td>1 X 9.6Gbps</td>
<td>0.50 X 9.85Gbps</td>
</tr>
<tr>
<td>HTTP - 1200K UEs with Default Bearers</td>
<td>1 X 9.6Gbps</td>
<td>0.58 X 9.85Gbps</td>
</tr>
<tr>
<td>Application Reply - 300K UEs with Default Bearers</td>
<td>9.6 X 9.6Gbps</td>
<td>8.85 X 2.84Gbps</td>
</tr>
</tbody>
</table>

Table 6-5 shows the result of P2P and HTTP maximum throughput test via a 10G port. The objective value is the capacity officially promised by Ixia. The result is based on recoding the actual figures in simulation. In P2P simulation, the throughput is about 6.45G for 300K users and 4.33G for 600K users. Both exceed the objective values. In addition, IxLoad also achieves impressive results in the HTTP maximum throughput test. In the case of 300K, 600K or 1200K users, the downlink throughput is around 9.85G. It is beyond objective value of 9.6G, but the actual uplink throughput is lower than the objective value in the three cases. The deviation can be explained because of the size of download file recommended by Ixia is still not optimal. The results of HTTP maximum throughput would be better, if the simulations used a smaller webpage size. For a smaller size of webpage, the downlink throughput is decreased, but the uplink throughput can be increased due to more requests in the same time interval. In the perfect configuration, the objective value is achievable.

In a similar fashion to that for the P2P, the configuration for HTTP throughput test uses a similar file size as well as the buffer size for 300K and 600K simulated users. However, it still achieves its objective value for downlink in both cases. It is because of TCP acceleration feature of
IxLoad. At present, TCP acceleration only supports HTTP application and Application replay. What can TCP acceleration do? Briefly, TCP acceleration can carry out much higher throughput on an internet connection than is possible by a standard TCP by breaking a long TCP control loop into several control loops. This feature is shown in the IxLoad in the form of a checkbox. The user can decide whether or not to check this. The same HTTP throughput test with and without TCP acceleration have been compared. It appears that the throughput of the identical configuration is enhanced by up to 2Gbps through the use of the TCP acceleration feature.

The Application Replay could not achieve a high throughput. 10G port throughput is derived from 12 1G ports by means of aggregation. The consequence of the Application Replay not achieving throughput goal in the 1G throughput test, means that it is impossible for IxLoad to achieve the 10G port throughput objective value. Although 1G port throughput test results for FTP applications surpasses the objective values, the 10G port throughput test of FTP applications can only achieve 4.7G with 300K UEs in the release 5.10.152.64 EB and it becomes worse in the latest version because of a variety of technical issues. These may be overcome in the next release.

### 6.2 Session Management

#### 6.2.1 ECM State Change

**Purpose:**

ECM state change mechanism introduced in Section 3.1.1 benefits both UE and network. The network is able to re-assign the resources that belonged to idle UE to the connected UE, so as to improve the efficiency of resource usage. UE can save the power used for maintaining the network connection and thus extend the service life of battery. This is what operators would like to see in their networks. It is important for real-world traffic simulator to be able to handle this type of scenario, so this test case is constructed to see if IxLoad is capable of carrying out this task or not. In the test, IxLoad client emulates the behaviour of UE, eNodeB as well as MME during ECM state change of UE, while cooperating with MPG and IxLoad server.

**Configuration:**
There is no special setting in MPG. The IxLoad client is configured to subscriber model, since this model has the command to change UE state. The IxLoad server emulates a FTP server. The session of the test case is constructed as shown in Figure 6-14. At the beginning, UE initiates an attach procedure with a Create Session command so that UE can download a file from the FTP server. After 20 seconds, which is defined in the first Think command, UE enters IDLE state using “Enter Idle State” command. UE will remain in IDLE state for 10 seconds as set in the Think command. Then UE is switched to CONNECTED state by the Exit Idle State command and starts to upload file to the FTP server. The size of the downloaded and uploaded files are so small that the time spent on these two actions can be ignored, compared to the time configured for Think, meaning that the duration is starting from Create Session to Delete Session, called one loop, is around 30 seconds. In the subscriber model, the loop button is disabled and hence IxLoad only repeats the loop, until the time configured in Timeline is exhausted.

Expected Results:

When session is created, UE is in the state of ECM-CONNECTED. After 20 seconds of Think, UE is initiated to enter ECM-IDLE by sending a Release Access Bearer Request to SGW. SGW replies with an acknowledge message and the state of UE becomes ECM-IDLE in IxLoad. After another 10 seconds of Think, UE re-enters ECM-CONNECTED state and requests to upload a file to the FTP server. Accordingly, the state of UE in IxLoad should be shown as ECM-CONNECTED. After upload activ-
ity, UE quits services. The time for upload and download can be ignored, because of small file size. Therefore the whole procedure should take approximately 30 seconds.

**Verification and Results:**

To verify this test case, it is necessary to look at GTPv2 session messages. There are two ways to verify session messages. One involves capturing by Analyzer. This is a capture tool similar to Wireshark and which is integrated into IxLoad. Figure 6-15 shows one loop of session messages for the test case captured by analyzer with port number 2123 as the capture filter. The duration of the session during one loop is 30 seconds, as expected. The session starts with four GTPv2 messages of the attach procedure. In the 20th second, UE enters IDLE state and MME sends Release Access Bearers Request to MPG to inform it that eNodeB is going to delete all the information relating to UE and drawback radio resources. MPG deletes the user plane belonging to UE and replies with a Release Access Bearers Response, which contains the flag that the request has been accepted. It indicates that IxLoad plays the role of UE, eNodeB and MME successfully during the ECM state change from Connected to Idle. According to the test case configuration, UE shifts to enter CONNECTED state 10 seconds later and MME asks MPG to again build up the PDN connection for UE. The capture shows that Modify Bearer Request is initiated properly by MME and answered by MPG using Modify Bearer Response. It hints that the emulated MME realizes that the state of UE is changed and request MPG to allocate the resources again for UE.

<table>
<thead>
<tr>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10.115.17.15</td>
<td>10.119.15.177</td>
<td>GTPv2</td>
<td>Create session request</td>
</tr>
<tr>
<td>0</td>
<td>10.115.15.177</td>
<td>10.119.15.17</td>
<td>GTPv2</td>
<td>Create Session Response</td>
</tr>
<tr>
<td>0</td>
<td>10.115.17.15</td>
<td>10.119.15.137</td>
<td>GTPv2</td>
<td>Modify Bearer Request</td>
</tr>
<tr>
<td>0</td>
<td>10.115.15.137</td>
<td>10.119.15.15</td>
<td>GTPv2</td>
<td>Modify Bearer Response</td>
</tr>
<tr>
<td>20</td>
<td>10.115.17.15</td>
<td>10.119.15.137</td>
<td>GTPv2</td>
<td>Release Access Bearers Request</td>
</tr>
<tr>
<td>20</td>
<td>10.115.15.137</td>
<td>10.119.15.17</td>
<td>GTPv2</td>
<td>Release Access Bearers Response</td>
</tr>
<tr>
<td>30</td>
<td>10.115.17.15</td>
<td>10.119.15.137</td>
<td>GTPv2</td>
<td>Modify Bearer Request</td>
</tr>
<tr>
<td>30</td>
<td>10.115.15.137</td>
<td>10.119.15.15</td>
<td>GTPv2</td>
<td>Modify Bearer Response</td>
</tr>
<tr>
<td>30</td>
<td>10.115.17.15</td>
<td>10.119.15.137</td>
<td>GTPv2</td>
<td>Delete Session Request</td>
</tr>
<tr>
<td>30</td>
<td>10.115.15.137</td>
<td>10.119.15.17</td>
<td>GTPv2</td>
<td>Delete Session Response</td>
</tr>
</tbody>
</table>

*Figure 6-15 GTPv2 capture of ECM state shift*

The other way to verify session is based on the statistics of IxLoad. The simulation time configured in Timeline is 150 seconds. According to the configuration, each loop takes 30 seconds. Thus 5 ECM state shifts are supposed to be seen in the statistics. EGTP-Other Messages window in Figure 6-13 illustrates that 5 Release Access Bearer Request messages are sent and 5 correct Release Access Bearer Responses are received by IxLoad. Considering that 5 times attach results in 5 modify bearer mes-
Assessment of IxLoad in an MPG Environment

Yue Peng / Zhiqiang Tang

Test Scenarios

2012-09-07

Messages entering CONNECTED state causes 5 times another 5 modify bearer messages and thus a total of 10 modify bearer messages should be seen in the statistics. This matches the counter for Modify Bearer Request Tx and Modify Bearer Response Rx in EGTP-Session Management Messages window in Figure 6-16. In order to ensure that IxLoad session management statistics is providing the true information, the statistics relating to these messages are double checked on the MPG. There were no session failures during the test. The modify bearer messages counters of MPG are increased by 10, after this test case.

Figure 6-16 Statistics of ECM state shift

6.2.2 Tracking Area Update

Purpose:

TAU is one of the most important mobility scenarios in an LTE network. TAU is initiated by UE to report current location in the state of ECM-IDLE. IxLoad has offered a set of network commands to trigger TAU activity. MPG supports TAU with MME relocation without SGW relocation. Hence it is possible to use MPG to verify if this common mobility scenario is supported by IxLoad.

Configuration:

In IxLoad, the test case is created in subscriber mode. As shown in Figure 6-17, there are two eNodeBs (eNB-R1 and eNB-R2) and each eNodeB is connected to one MME respectively. The X2 interface between the two eNodeBs is enabled. In the UE configuration, the mobility option is supposed to be checked and the mobile path is from eNB-R1 to eNB-R2, but the mobility interval could be ignored in the subscriber network command case.
In the subscriber mode, the eGTP network commands are powerful to setup the TAU scenario. Once IxLoad client sends the “create session” commands and creates one session, one FTP “get” command will be implemented. After 10 seconds of the session creation, the UE will enter into the ECM-IDLE state. In another 5 seconds, a “TAU” command will be initiated by IxLoad. 5 seconds after the TAU activity, the UE will switch to the ECM-CONNECTED state and continue the user plane traffic with an FTP “put” command. All the eGTP and Layer 7 commands are shown in Figure 6-18.

Expected Results:

At the beginning of the test, UE is connected to eNB-R1 and downloads a file is downloaded from FTP server. In 10 seconds, UE enters ECM-IDLE state. 5 seconds later, UE moves to eNB-R2 and by implementing a TAU procedure, UE should be able to register successfully in eNB-R2 in the state of ECM-IDLE. After 5 seconds, UE quits ECM-IDLE state and starts to use FTP service again to upload a file via eNB-R2.
The capture by analyzer, including control plane messages as presented in Figure 6-19 and user plane messages, is used to state the whole process of this scenario. The session begins with a PDN connection creation procedure. Soon the FTP stream for downloading file via eNB-R1 can be seen in the user plane capture. After 10 seconds, UE enters ECM-IDLE state. Figure 6-19 shows the parent MME of eNB-R1 and sends a Release Access Bearer Request to MPG which replies with an accepted Release Access Bearer Response message that indicates that MPG retrieves all the resources allocated to UE. When UE moves to eNB-R2, it is observed in Figure 6-19 that the parent MME of eNB-R2 triggers a TAU procedure with Modify Bearer Request, which contains its IP address and TEID. MPG updates all the information of UE and returns Modify Bearer Response to the new MME. 5 seconds later, UE re-enters ECM-CONNECTED state. MME simulator of IxLoad uses Modify Bearer Request to apply for new resources from MPG. After the request is accepted by MPG, the user plane capture of analyzer shows that there is an upload FTP stream from UE to the FTP server via eNB-R2, corresponding to the expected results.

![Figure 6-19 GTPv2 messages in TAU](image)

The simulation results can also be verified by the statistics of IxLoad. Figure 6-20 shows that the first eNodeB that UE is registered in is eNB-R1 and then UE is connected to eNB-R2 after TAU procedure. Furthermore, the session management message counters of IxLoad match the counters in MPG. The window of Tracking Area Update statistics of Figure 6-20 illustrates that the number of successful TAU activities is 3 without any failures, which is consistent with the S11 counter in MPG.
6.2.3 Downlink Data Notification

Purpose:

Section 6.4.4 discusses the advantage associated with the ECM state. IxLoad supports emulation of UE in ECM-IDLE state and pre-defines a timer so as to generate traffic from PDN. As MPG receives incoming data for UE in IDLE state, it informs MME to wake up UE. This test case makes use of the downlink data notification feature of MPG in order to verify the reaction of IxLoad to network a triggered service request procedure.

Configuration:

MPG can perform downlink data notification correctly without any special configuration. Both IxLoad client and server act as two stateless peers. UDP is chosen to exchange data for both peers, since UDP is a flexible protocol for a time slot based test, which makes it possible to control, not only the size of the packet and their frequency, but also its duration. Figure 6-21 depicts the network command configuration of this test case in the IxLoad client. As usual, the session setup for the default bearer is first, followed by a Generate Trigger command. The trigger generated by IxLoad client leads to activation of Wait for Trigger in IxLoad server side, illustrated in Figure 6-22. As a result, IxLoad server begins to count up time and it will push traffic in the form of UDP stream to IxLoad client in 40 seconds. After transmitting the trigger, IxLoad client starts to generate a UDP stream to IxLoad server. The duration of UDP stream is set to 20 seconds. Then UE emulated by IxLoad client enters ECM-IDLE state. UE will remain in the IDLE state, until it is paged by eNodeB. The Think command behind “Wait For Paging” retains UE online for up to 20 seconds, which is equal to the duration of UDP stream transmitted by IxLoad server so that the UE can receive all the data at the end of the session.
Assessment of IxLoad in an MPG Environment

Yue Peng / Zhiqiang Tang

Test Scenarios

2012-09-07

Figure 6-21 Network command configuration of downlink data notification for IxLoad client

Figure 6-22 Network command configuration of downlink data notification for IxLoad server

Expected Results:

In the first 20 seconds, UE uploads UDP stream via MPG. Then UE enters ECM-IDLE state. 20 seconds later, the UE data comes to MPG, but UE is still in the ECM-IDLE state. MPG wakes up UE which then enters ECM-CONNECTED state and downloads data via MPG. The download process takes around 20 seconds.

Verification and Results:

For fully verifying the downlink data notification functionality of IxLoad, it needs to be confirmed by the control plane with capture and the user plane with statistics. Figure 6-23 shows that UE enters ECM-IDLE state in the 20th second and MPG releases all the UE resources. 20 seconds later, MPG launches downlink data notification due to incoming UE traffic. MME accepts request and replies with a downlink data notification acknowledgement to MPG. Following this MME requests
Assessment of IxLoad in an MPG Environment

Yue Peng / Zhiqiang Tang

Test Scenarios 2012-09-07

resources for UE. MPG reallocates bearers to UE. As the tunnel between eNodeB and MPG is built up, MPG forwards downlink data to UE. It is noted that IxLoad performs correctly on the control plane of the downlink data notification. The UE’s ECM state change can be verified with the statistics of IxLoad. Figure 6-24 shows that the UE state is changing through time. After UE finishes the attach procedure, the state for the UE is ECM-CONNECTED. When UE enter ECM-IDLE state, it becomes Idle. As downlink data arrives, the ECM state of UE is changed from IDLE to CONNECTED. In addition to its state, an EGTP per PDN connection statistic window provides other information of UE, such as IMSI, IP and APN.

<table>
<thead>
<tr>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10.119.15.17</td>
<td>10.119.15.17</td>
<td>GTPv2</td>
<td>Create Session Request</td>
</tr>
<tr>
<td>0</td>
<td>10.119.15.17</td>
<td>10.119.15.17</td>
<td>GTPv2</td>
<td>Create Session Response</td>
</tr>
<tr>
<td>0</td>
<td>10.119.15.17</td>
<td>10.119.15.17</td>
<td>GTPv2</td>
<td>Modify Bearer Request</td>
</tr>
<tr>
<td>0</td>
<td>10.119.15.17</td>
<td>10.119.15.17</td>
<td>GTPv2</td>
<td>Modify Bearer Response</td>
</tr>
<tr>
<td>0</td>
<td>10.119.15.17</td>
<td>10.119.15.17</td>
<td>GTPv2</td>
<td>Release Access Bearers Request</td>
</tr>
<tr>
<td>0</td>
<td>10.119.15.17</td>
<td>10.119.15.17</td>
<td>GTPv2</td>
<td>Release Access Bearers Response</td>
</tr>
<tr>
<td>41</td>
<td>10.119.15.17</td>
<td>10.119.15.17</td>
<td>GTPv2</td>
<td>Downlink Data Notification</td>
</tr>
<tr>
<td>41</td>
<td>10.119.15.17</td>
<td>10.119.15.17</td>
<td>GTPv2</td>
<td>Downlink Data Notification Acknowledgement</td>
</tr>
<tr>
<td>41</td>
<td>10.119.15.17</td>
<td>10.119.15.17</td>
<td>GTPv2</td>
<td>Modify Bearer Request</td>
</tr>
<tr>
<td>41</td>
<td>10.119.15.17</td>
<td>10.119.15.17</td>
<td>GTPv2</td>
<td>Modify Bearer Response</td>
</tr>
<tr>
<td>60</td>
<td>10.119.15.17</td>
<td>10.119.15.17</td>
<td>GTPv2</td>
<td>Echo Request</td>
</tr>
<tr>
<td>60</td>
<td>10.119.15.17</td>
<td>10.119.15.17</td>
<td>GTPv2</td>
<td>Echo Response</td>
</tr>
<tr>
<td>60</td>
<td>10.119.15.17</td>
<td>10.119.15.17</td>
<td>GTPv2</td>
<td>Delete Session Request</td>
</tr>
<tr>
<td>60</td>
<td>10.119.15.17</td>
<td>10.119.15.17</td>
<td>GTPv2</td>
<td>Delete Session Response</td>
</tr>
</tbody>
</table>

Figure 6-23 GTPv2 capture of downlink data notification

Figure 6-24 UE state in different time

The reason why user plane is particularly verified is that IxLoad may drop packets in large numbers when it is waking up UE. There is a short period during which MPG receives UE’s packets and when user plane connection between UE and MPG does not exist. The statistics of IxLoad are used to confirm whether large packets are dropped in this case. To be able to easily distinguish the source of UDP streams, the throughput generated by IxLoad server is doubled during the configuration. Apparently the first curve is the throughput of uplink and the second curve is the throughput corresponding to downlink data in Figure 6-25. The pink
line represents the throughput of peer in IxLoad client side and blue line stands for the throughput of peer in IxLoad server side. During the throughput ramp up and ramp down, the pink and blue lines almost overlap. When the throughput is stable, both lines are smooth. This implies that there is no severe packet loss in the scenario of a downlink data notification. Figure 6-25 shows that the pink line is slightly higher than the blue one, because the throughput of IxLoad client contains both a TCP and GTP headers in addition to the throughput generated by IxLoad server.

![Figure 6-25 Throughput plot of downlink data notification](image)

### 6.2.4 X2-Based Handover

**Purpose:**

As mentioned in Section 4.3.3, at the present time X2-based handover without MME relocation and without SGW relocation is the only scenario supported by IxLoad in MPG test environment. It is also one of the most important scenarios operators focus on. Low latency and lossless forwarding of X2-based handover can increase the user experience, while X2-based handover reduces the overhead of SGW by transferring data forwarding role for UE handovers from SGW to eNodeB. Considering that the current hardware configuration of eNodeB has enough capacity to handle data forwarding due to a handover, operators would like to make use of X2-based handover as much as possible so as to minimize the cost for purchasing extra SGWs. In this test case, MPG plays not only the role of PGW for PDN connection but also SGW for anchoring traffic to target eNodeB. Based on this, it should be assessed whether IxLoad, as a simulation tool, is able to emulate the behaviour of UE, eNodeBs and MME properly, when cooperating with MPG in the
Assessment of IxLoad in an MPG Environment

Yue Peng / Zhiqiang Tang

2012-09-07

Test Scenarios

X2-based handover without MME relocation and without SGW relocation scenario.

**Test Scenarios**

**2012-09-07**

X2-based handover without MME relocation and without SGW relocation scenario.

**Configuration:**

IxLoad client simulates a UE, two eNodeBs as well as one MME. HTTP server is emulated by IxLoad server. MPG is connected between the IxLoad client and IxLoad server. UE is doing round-trip between two eNodeBs, while browsing the website of HTTP server. The part of mobility settings of IxLoad client is shown in Figure 6-27. In this configuration, the mobility of UE is enabled and the route of the round-trip is specified in Mobile Path. Mobility Interval sets the total time used for round-trip. Combining these three mobility parameters, a simple test scenario is conducted. UE is moving from eNodeB1 to eNodeB2 in the first 30 seconds. Later, UE takes 30 seconds to move back to eNodeB1 from eNodeB2. During the entire scenario, UE always uses HTTP service.

Expected Results:

In the first 30 seconds, UE is moving from eNodeB1 to eNodeB2. Later, UE is moving back to eNodeB1 from eNodeB2 during 30 seconds. In other words, two handover events occur in 60 seconds. For each handover event, IxLoad follows the procedure of Section 3.1.5 to upload.
uplink data by either source eNodeB or target eNodeB at the different stages of handover process. Meanwhile MPG keeps pointing downlink traffic to the eNodeB where UE is located correctly, according to the received mobility information of UE from MME. Within 60 seconds, no HTTP failure should have occurred which is caused by handover.

**Verification and Results:**

For handover verification, the evaluation should not only concentrate on whether handover procedure is successful, but also focus on whether IxLoad passes traffic to MPG in a rational manner. EGTP-Handover shows that two “X2-based no MME change no SGW change” handover events are initiated and both handovers are successful. It matches the two events in which UE moves from eNodeB1 to eNodeB2 and returns from eNodeB2 to eNodeB1. The GTPv2 messages of handover without any failure are also visible in the switch capture. In the user plane, the throughput of UE must be passed to MPG smoothly, except handover period. The throughput wave of UE has a small concave area during the handover in the HTTP Client-Throughput Objective of Figure 6-28. However, handover should not affect user experience. This can be verified in Figure 6-28. There is no HTTP failure in HTTP Client. It appears that there is no reduction in the user experience because of X2-based handover.

Furthermore, Figure 6-29 proves that the target eNodeB replaces the source eNodeB to upload UE’s data to MPG, once the handover enters completion period. 10.115.18.10 is the address for the source eNodeB and 10.115.18.11 is the address for the target eNodeB. All the packets of UE’s traffic after the handover control plane messages are uploaded from the target eNodeB and is in accordance with 3GPP standard. The conclusion is that IxLoad is capable of emulating control plane and user plane of UE, eNodeBs as well as MME during X2-based handover in the case of no MME relocation and no SGW relocation.
6.2.5 Network Initiated Dedicated Bearer

Purpose:

The idea of dedicated bearer makes IP traffic control more flexible. When UE requires a specific QoS treatment, additional bearers (dedicated bearers) are created between UE and PDN, using Traffic Flow Template (TFT). These bearers can be deactivated, as they are no longer required. Thus released resources could be allocated to other UEs, which require a better QoS treatment. It improves the efficiency of resource usage. In version 5.10.165.3EB, IxLoad starts to support dedicated bearer. This test scenario is to verify the activation of a dedicated bearer.

Configuration:

First of all, dedicated bearer function is supposed to be enabled in the APN level configuration of MPG.

Secondly, as IxLoad has not as yet implemented an emulated PCRF, an Ericsson internal tool is used to simulate PCRF. In the dedicated bearer activation procedure, PCRF sends the RAR or CCA message to PGW in
case Policy and Charging Control is deployed. In the RAR message, the rating group, QoS information including QCI and MBR, and the flow information related to the PDN must be defined. Additionally, the dedicated bearer trigger frequency, that is an interval, is specified in the procedure.

This test case is running with IxLoad 5.10.165.3EB. It is an HTTP application over the eGTP stack with one eNodeB and MME and only one user is simulated. An APN with a PCC deployed is selected in the APN setting, and the PDN IP address must be correctly configured in order to correspond. There are three important options in the eGTP configuration in the application settings, shown in Figure 6-30. If “run activity on default bearer only” is checked, IxLoad will consider the default bearer QoS values and the activity will always run on the default bearer. If “UE initiated service request for this activity” is checked, the emulated UE will initiate a service request by sending a Bearer Resource Command with the TFT operation set to “Create New” and the flow control setting can be user defined in the meantime. If “Default bearer fallback” is checked, it will fallback to the default bearer, in case in which no dedicated bearer is created. Since it is a network initiated dedicated bearer activation, only the last option checked is sufficient.

![Figure 6-30 Configurations to enable dedicated bearer](image)

**Expected Results:**

A default bearer is firstly activated when it is running. 15 seconds later, PCRF will initiate the RAR message to create a dedicated bearer. From the counter in MPG, there is one active default bearer and one active dedicated bearer. IxLoad statistics shows the bearer status. The control
Plane messages should show the correct procedure as stated in Figure 3-2.

**Verification and Results:**

When running, the first default bearer is created successfully. After the trigger interval, that is 15 second in this case, PCRF sends the RAR message to PGW to initiate the dedicated bearer activation procedure. From the capture of IxLoad analyzer shown in Figure 6-31, PGW sends the Create Bearer Request including all the mandatory information to the MME and will quickly receive the accepted response. From the perspective of MPG statistics, there are two active EPS bearers and one active dedicated EPS bearer in the APN statistics.

![Figure 6-31 GTPv2 messages of dedicated bearer activation](image)

For IxLoad statistics there are two sessions that are created successfully in the all tab of EGTP - General statistics of Figure 6-32, in which one session is the expected dedicated bearer, as shown in the dedicated bearers tab of EGTP - General statistics of Figure 6-32.

![Figure 6-32 Dedicated bearer counter in IxLoad](image)

### 6.3 Other Scenarios

#### 6.3.1 Busy Hour Functionality

*Purpose:*
Busy Hour Call Attempts (BHCA) is the number of session attempts at the busiest hour of the day (peak hour), used to evaluate the capacity of a mobile network. The higher BHCA, the higher stress will be on the network. This test case is constructed to verify the ability of IxLoad to handle BHCA. Greater emphasis will be on the control plane and the higher throughput is not target.

**Configuration:**

A high load test scenario with a large number of users is used. Two session controllers in MPG are used and a no SACC APN is configured for PDN connection.

In IxLoad, the test case is set in net traffic mode, and only one eNodeB as well as one MME is created in the eGTP stack without any mobility events. The maximum number of users that IxLoad supports for HTTP application in a 1G port is 100K. For this test case, continuous creation and deletion brings additional CPU and memory costs, which affects the number of simulated users. Hence only 90K users are selected as the objective value in order to verify the BHCA functionality of IxLoad.

The ramp up value is set to 150 sessions per second which means that every second IxLoad initiates 150 sessions for 150 users. The lifetime of the user which is 600 seconds is obtained by the number of simulated users 90K divided by the ramp up value of 150 sessions per second. It is configured in figure 6-33, as the duration for Think. Each user is online for 10 minutes. When the lifetime expires, the user will be offline and it will then trigger session termination. Because there is the same lifetime for all users, the 150 sessions created within the same second will be deleted simultaneously in the 600 seconds. 600 seconds lifetime ensures that the first time for a session termination occurs simultaneously with the number of active users reaching the test objective. As the active user number reaches the test objective, the sessions of the first 150 users are terminated. Meanwhile another 150 new users are created by the IxLoad request for PDN connection in order to retain the 90K simulated users. At a later stage, every second 150 sessions are deleted, while 150 sessions are created, which is typical busy hour behaviour. It lasts, until simulation ends.
In the application command configuration, one “Think” is placed to slow down the speed of generation of the traffic. As is shown in Figure 6-34, HTTP 1.1 is chosen and each user is allowed to have 1 concurrent TCP connection at most. The smallest page size is selected to ensure that the user plane traffic will not impact upon the session setup during the busy hours.

To assess the stability of IxLoad through a long period of busy hours, the timeline is supposed to be sufficiently long, so that the sustain time of time-line is up to 3 hours.

**Expected Results:**

In this case, 90K bearers for 90K users will be activated in the ramp up stage. In the sustain stage, the bearers of users whose lifetime expires will be deactivated and a corresponding number of bearers for new users will be activated. The total active bearer number should remain at 90K for 3 hours.

**Verification and Results:**

Supposing that related sessions are succeeded, each user emulated by IxLoad only owns a bearer in the case of no dedicated bearer. Thus the...
verification process is ongoing from the perspective of bearers. During the ramp up time, IxLoad creates the bearers in accordance with the ramp up value of 150 bearers per second. Once the number of bearers achieves the objective value, the session termination is initiated. The bearers for users whose lifetime has expired are deleted at the speed of 150 bearers per second and an almost equal number of new bearers are created. The slight variation is because some sessions are not created successfully due to extra request messages from IxLoad. The issue relating to extra request messages has been previously reported to Ixia and the corresponding patch will be delivered shortly. Although the final number of active bearers cannot reach the objective value of 90K, it is quite close to 90K, as presented in Figure 6-35.

The amount of active subscribers in MPG statistics fluctuates around 90K. The session controllers maintain a high CPU load due to the large number of creating and deleting session events.

![Active Bearers in BHCA](image)

**Figure 6-35 Active bearer in BHCA**

### 6.3.2 Deep Package Inspection

**Purpose:**

Packet inspection and service classification is the core feature of Service Aware Charging and Control solution of MPG. With DPI, all the ongoing traffic flows will be filtered and classified based on a given rule. In addition, the service classification and the rating group are enabled to carry out the service charging function. A multiple applications scenario will be simulated and the three types of packet inspection will be verified in this test case.

**Configuration:**
In MPG, a specific APN is configured, which has the services including the set of deep packet inspection, header inspection and heuristics inspection in addition to the service classification and rating group. The SDF-ID table is configured in MPG to verify the classification. Table 6-5 displays some of the SDF-IDs that are related to this test.

<table>
<thead>
<tr>
<th>SDF-ID</th>
<th>Flow Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>20000</td>
<td>Default</td>
</tr>
<tr>
<td>52150</td>
<td>TCP header</td>
</tr>
<tr>
<td>52110</td>
<td>HTTP contain &quot;http1&quot;</td>
</tr>
<tr>
<td>52250</td>
<td>FTP</td>
</tr>
<tr>
<td>52750</td>
<td>POP3</td>
</tr>
<tr>
<td>53001</td>
<td>BitTorrent</td>
</tr>
</tbody>
</table>

Table 6-5 SDF-ID table in DPI test

In IxLoad, multiple applications including HTTP, FTP, BitTorrent, POP3 are simulated. Net Traffic mode is selected in this test case. Each application has one client and one server. Figure 6-36 shows the application command configuration of each application.

In addition, it is necessary to emphasize the importance of configuring the correct APN. “ixia-sacc” is the APN that has been configured for the packet inspection and service classification set in MPG. Figure 6-37 illustrates that the exact same APN is configured in IxLoad. It is also better to set high values for APN-AMBR, especially for downlink, since there might be heavy throughput in the case of one single user, which can avoid possible packet loss due to the constraints of AMBR.

Expected Results:
IxLoad is able to simulate an APN with service charging and control function. With multiple applications DPI test, the key aspect which must be checked in MPG, is the service classification. The following bullets are the expected SDF-ID mapping.

- HTTP, FTP and POP3 in DPI: all the handshake packets are classified in the header group and all the data packets are placed with the specific SDF-ID according to the payload rule, such as containing http1 in the page name, or using POP3 protocol.

- HTTP, FTP and POP3 in header inspection: all the packets are classified by filtering the TCP header to the header group.

- BitTorrent in Heuristics inspection: some setup packets are assigned with a default SDF-ID and all others are classified in the BitTorrent group.

**Verification and Results:**

With the multiple applications, in the first case, HTTP, FTP and POP3 packets are handled by DPI and the different traffic flows go to the correct classification, shown as the service data flow identity configured (SDF-ID) in MPG.

In the second case, HTTP, FTP and POP3 packets are handled by header inspection and each traffic flow is classified to the specific SDF-ID. For instance, in HTTP, the handshake packets (including SYN, ACK, FIN messages) are passed to 52150 and all the data packets are passed to the 52110 in DPI, while all the packets go to 52150 group in the header inspection case, shown in Table 6-6.

For the BitTorrent, one of the P2P protocol is suitable for use with the heuristics inspection. After inspection of MPG, the majority of the P2P packets are classified as SDF-ID 53001 and some packets are classified as SDF-ID 20000, listed in Table 6-6. The reason why some packets appear in the default group is that the setup and teardown messages of each P2P flow will not match any policy in either the heuristic inspection or the header inspection and will be considered as a default flow type.
For a HTTP user, DPI has to consume more CPU load of MPG than header inspection, which is caused by a more complex packet inspection process. In other words, DPI could result in a lower throughput than header inspection with the same CPU load in MPG. This is determined by running a simulation and the simulation statistics for IxLoad are shown in Figure 6-38. The accuracy of the throughput statistics of IxLoad has already been verified in Section 6.1.3, so this figure is reliable. Section 6.1.3 also highlights the effect of buffer size and file size on throughput under the condition of no packet inspection. In a similar manner to that for the throughput test without packet inspection, the throughput of three types of packet inspection depends on the configuration of buffer size and file size.

<table>
<thead>
<tr>
<th>Application</th>
<th>Deep Packet Inspection</th>
<th>Header Inspection</th>
<th>Heuristics Inspection</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP</td>
<td>52110, 52150</td>
<td>52150</td>
<td>N/A</td>
</tr>
<tr>
<td>FTP</td>
<td>52250, 52150</td>
<td>52150</td>
<td>N/A</td>
</tr>
<tr>
<td>POP3</td>
<td>52750, 52150</td>
<td>52150</td>
<td>N/A</td>
</tr>
<tr>
<td>BitTorrent</td>
<td>N/A</td>
<td>N/A</td>
<td>20000, 53001</td>
</tr>
</tbody>
</table>

Table 6-6 Service classification results with different application

6.3.3 Dynamic QoS Enforcement

Purpose:

The importance of QoS for operators is discussed in Section 4.3. The QoS mechanism is implemented in IxLoad with Dynamic QoS Enforcement feature. Section 4.3 briefly introduced this feature of IxLoad. The pur-
pose of this test case is to verify whether, in practice, it is possible for the IxLoad to dynamically negotiate QoS parameters with MPG and how well IxLoad performs QoS enforcement.

**Configuration:**

The network topological diagram is presented in Figure 6-39. PCRF is involved in the test to trigger QoS profile change of UE. A particular configuration about PCRF, such as the IP address, has been integrated into the MPG which will send QoS profile of UE to PCRF, when UE requests resources from MPG, in order to check whether the requested QoS profile and that which was predefined do in fact match. Several policies are configured in the PCRF. In the purpose of the test, one policy is to upgrade UE’s uplink APN-AMBR to 20 Mbps. The policy will work in one minute, from when UE starts to use resources.

This test case consists of two phases. For both phases, FTP application running on UE always attempts to push the 20 Mbps FTP stream in the uplink. In the first phase, UE initiates one session with uplink APN-AMBR at 10 Mbps and QCI 5. The reason why QCI is chosen to be 5 is that APN-AMBR only works on non-GBR bearers and UE will be allocated non-GBR bearers in the case of QCI 5. As the session is successfully created, UE generates traffic according to the negotiated QoS profile. The second phase is one minute later. PCRF sends RAR to inform MPG that UL-APN-AMBR has been updated to 20 Mbps. The QoS enforcement of IxLoad is assessed in the first phase. The intension of the second phase is to verify the reaction of IxLoad to dynamic QoS messages.

**Expected Results:**

Referring to [9], the signaling procedure based on the above configuration is shown in Figure 6-40.
Assessment of IxLoad in an MPG Environment

Yue Peng / Zhiqiang Tang

Test Scenarios

It illustrates that MME initiates the attach procedure by sending a Create Session Request to MPG. The Create Session Request contains QoS profile of UE. MPG forwards QoS profile of UE to PCRF with CCR message via Diameter protocol. Because QCI 5 and UL-APN-AMBR 10 Mbps do not violate any policy setting in PCRF, the two values as well as other QoS parameters are agreed by PCRF and acknowledged to MPG with CCA message. With a Create Session Response message, MPG forwards the negotiated QoS profile back to MME. The Attach Accept message is sent from MME to eNodeB, which includes QoS information of UE. eNodeB assigns radio resources to UE according to QoS information provided by MME. It implicitly points out that the maximum throughput of UE is limited to 10Mbps, due to the constraints of radio resources. In other words, QoS enforcement should occur in the IxLoad client side. As soon as the 1 minute timer has been triggered, RAR that carries updated UL-APN-AMBR is sent to MPG, as shown in Figure 6-40. By receiving Update Bearer Request from MPG, MME
notices that UL-AMBR has changed to 20 Mbps and commands eNodeB to update radio resource allocation. Hence the UE maximum bandwidth for uplink is adjusted to 20 Mbps.

In summary, the throughput of uplink should be controlled at below 10 Mbps within the first one minute. Later, the maximum bandwidth for uplink is changed to 20 Mbps due to RAR message of PCRF. More traffic is allowed to be generated from UE profiting from double bandwidth.

Verification and Results:

The above expected results can be confirmed by means of the statistics of IxLoad. The FTP throughput statistic of IxLoad reflects how satisfactorily the IxLoad implements Dynamic QoS Enforcement. Figure 6-41 illustrates that the throughput of UE fluctuates at around 10 Mbps within the first minute. Considering that the throughput of application is always attempting to push towards 20 Mbps, it follows that the QoS enforcement works. Because there are no packets dropped in the MPG counters, the only possibility is that QoS enforcement is executed in the IxLoad client side. Because the signaling messages between UE and MME are IxLoad internal messages, it is impossible to directly check whether the content of control message Attach Accept as well as RRC Connection Reconfiguration are correct. However, it is feasible to conclude how IxLoad executes QoS enforcement. The TCP failure counter of IxLoad shows that there is no TCP failure caused by packet loss. It means that eNodeB does not throw away any packet and the throughput is controlled by adjusting radio resources.

In one minute, EGTP-Bearer Management Messages of Figure 6-41 shows that IxLoad MME simulator receives an Update Bearer Request from MPG and answers it properly. Then the throughput of UE increases from 10 Mbps to 20 Mbps. It proves that IxLoad is able to handle dynamic QoS events on the control plane and the user plane. It is also visible that the throughput becomes stable in Figure 6-41, compared to that for the previous minute. It is because the bandwidth assigned to UE is no longer smaller than traffic generated by application.
6.3.4 Multiple APNs

Purpose:

Section 4.3 provides a brief introduction about the practical application of APN and presents the functionality of multiple APNs as a new feature of IxLoad. This test case will verify whether multiple APNs feature works as well as stated in Section 4.4.

Configuration:

The basic idea is that a UE simulated by IxLoad client attempts to simultaneously two PDNs via MPG. Two PDNs are emulated by IxLoad server. One PDN provides FTP download service and other offers web browsing service. Two access points corresponding to two PDNs are configured in the IxLoad client, as shown in Figure 6-42. The former is ixia-nosacc, which has 100Mbps limitation for uplink and downlink. The other is ixia-sacc whose throughput is limited to 200Mbps. Both APNs
are associated with the same MPG by means of the third selection model. The corresponding information for both APNs and PDNs is also integrated in MPG.

Figure 6-43 presents the network command configuration for Multiple APNs test. At the beginning, UE initiates a Create Session command and requests a connection to FTP server so as to download a file of 4096 bytes. After 10 seconds, which is set with the first Think command, the requirement of the UE is to connect to HTTP server in order to browse a 1 byte webpage by using the second Create Session command. 10 seconds later, the PDN connection to FTP server is released. After 10 seconds, UE releases the PDN connection to the web server. Both 10 seconds intervals are controlled by Think command.

**Expected Results:**

One UE is able to create more than one bearer; for instance, two bearers in this case, are connected to two different APNs, ixia-nosacc and ixia-sacc APNs. From the MPG counter, two EPS bearers are activated. From the IxLoad statistics viewers, the same UE has connected two APNs and obtained different traffic flows (HTTP and FTP).

**Verification and Results:**

The simulation time set in the timeline is 60 seconds. In the light of above configuration, each loop initiates two sessions in order to request two PDNs connection. If only three Thinks are taken into account, one loop takes 30 seconds. Four sessions should be initiated in 60 seconds, which matches the EGTP-General window of Figure 6-44. There are 4 sessions initiated and created successfully. EGTP-per PDN connection of
Figure 6-44 shows that two PDN connections belong to the identical UE, whose IMSI number is 226041000000001. The IP addresses that the MPG assigned for the two PDN connection are 16.0.0.3/11 and 16.32.0.3/11. This is consistent with APN configuration of MPG. The local IP pool 16.0.0.0/11 and 16.32.0.0/11 are allocated for APN ixia-nosacc and APN ixia-sacc respectively. This is reflected in the MPG statistics, which shows that one active subscriber is using two PDN connections in order to access the FTP server and HTTP server.

In addition to the control plane, the user plane must also be confirmed. In a loop, UE downloads one file from FTP server and one file from the HTTP server. There are a total of two loops. Figure 6-45 presents two curves in the FTP client – throughput objective and HTTP client – throughput objective respectively. The throughput objective of IxLoad provides a convenient manner in which to view the average throughput of UE. The low average throughput of both views is due to small downloaded files. Accordingly the throughput counters of MPG for two APN connections are increasing, because of the behaviour of the subscribers.
7 Feedback

7.1 Feedback to Ixia

7.1.1 S1-Based Handover with Indirect Tunneling

Except X2-based handover mentioned in Section 3.1.5, S1 based handover is another most important handover scenario in an LTE network. IxLoad has already implemented the handover functionality both in Net Traffic mode and advanced subscriber mode. However, when testing the S1-based handover in MPG, it shows that Ixia does not fully support all the cases of S1-based handover stated in the 3GPP standard [9].

A simple S1-based handover with no MME relocation and no SGW relocation case is taken as an example. Two eNodeBs are connected to the same MME without X2 connection between each other. After the session is created successfully, the handover event will be triggered after a predefined time interval. From IxLoad’s perspective, the emulated MME sends a Create Indirect Data Forwarding Tunnel Request message to SGW via S11 interface, but it receives a Create Indirect Data Forwarding Tunnel Response massage with a failure cause value 106, as shown in Figure 7-1. Then IxLoad skips the rest of handover procedures and declares that handover is unsuccessful.

![Figure 7-1 Error message in the S1-based handover in MPG](image)

According to the 3GPP standard [34], failure cause value 106 indicates “Data forwarding not supported”. It indicates that the other side does not support Create Indirect Data Forwarding Tunnel message. However, IxLoad should not declare handover failure immediately, because
Create Indirect Data Forwarding Tunnel is an optional message specified in 3GPP standard [9], as shown in Figure 7-2.

Figure 7-2 Signaling during S1-based handover without MME and SGW relocation

In the expected way, IxLoad could continue the rest handover procedure with the Modify Bearer Request/Response messages in the case that the other side does not support the optional messages.

It is recommended that IxLoad is also compatible with the other side, which may not support optional messages of handover procedure. For example, IxLoad could provide an option to enable or disable the optional messages. This feature has been suggested as a feature request to Ixia.

7.1.2 IxLoad Report Generator

IxLoad supplies a set of statistics for different layers and collects all the data in the csv logs under the background, when running a test case. Based on the amount of logs, it is possible for IxLoad to produce a complete report that states the detailed simulation results for analysis of the test case. However, even in the latest version, only the statistics of layer 7 applications are included in the report, which is not sufficient to outline the entire test results. For instance, most of the test cases in this thesis work are based on the eGTP stack. It is recommended that the statistic report could contain the eGTP group and the L2/L3 group.
Moreover, in the graphical timelines per activity chapter of the IxLoad report, there is a table to describe the latency of certain activities with the Minimum/Maximum/Average values. For most of the activities, the maximum value is far beyond the average value. Figure 7-3 is HTTP connection latency statistics recorded in report.

![Figure 7-3 HTTP connection Latencies in the IxLoad report](image)

In the table of Figure 7-3, the maximum HTTP connect latency is up to 2458 μs and the average latency is about 248 μs. The average value is very close to the minimum value, but there is an extremely awful peak latency which is around ten times as the average value. In this case, the peak value more looks like an unexpected jitter in the network. From the perspective of analyzing network latency, it is better to show the distribution of the latency of packet, instead of maximum latency and minimum latency during the running time. IxLoad could split the timeline between minimum latency and maximum latency into several intervals of equal size and count the delayed packet number for each interval. This feature could be placed in the process of collecting statistics. However, this proposal has its limitation that user cannot control the latency. The extension of above proposal is to take packet latency of practical network into account during the simulation. IxLoad could provide an acceptable latency distribution by default and it is allowed to be configured by user for different purposes. Based on the configuration of latency distribution, IxLoad generates traffic. This feature will make user be able to control network latency, while emulating the real world traffic.

### 7.1.3 Three-Way and Four-Way Tear Down
IxLoad has a termination option in TCP configuration for HTTP application, which determines the method that server uses to close the TCP connections. The default setting of termination option uses three-way tear down to terminate connection with the messages FIN ACK, FIN ACK and ACK. If four-way tear down is chosen, IxLoad attempts to
close the connection by the using messages FIN ACK, ACK, FIN ACK and ACK. Figure 7-4 shows the handshake procedure of the two methods for TCP termination in HTTP 1.0.

![Figure 7-4 three-way and four-way teardown in TCP](image)

In the back to back test environment, with the same configuration, three-way handshake results in higher throughput than four-way handshake, because two more efficient modules stackAdapter and koneLib are loaded during the simulation in the case of three-way handshake. Both modules can push very high throughput. However, in the single user throughput test in MPG environment, especially with packet inspection, the opposite result that four-way tear down shows much higher throughput is obtained. It is hard to explain why IxLoad has entirely different throughput performance under two test environments. As it is an ongoing issue, hopefully soon some feedback can be received from Ixia.

It is recommended to add some comments in the user guide to clarify that the selection of three-way handshake or four-way handshake option in configuration not only introduces different way to close TCP connection, but also brings in different throughput performance.

### 7.1.4 Limitation of IPv6

In the process of assessing eGTP stack of IxLoad, this thesis witnesses some limitations of IxLoad in the IPv6 traffic.

Currently IxLoad is not supporting to configure the IPv6 parameters for the RADIUS messages. IxLoad does not support the Multimedia Messaging Service (MMS) traffic with MM1 protocol either, under the condition that MM1 is the interface between UE and an MMSC.

In the latest release 5.30EB of IxLoad, it declares that the emulated eNodeB and MME stack is working over IPv6. Because of the time
Assessment of IxLoad in an MPG Environment

Yue Peng / Zhiqiang Tang

7.1.5 Future Suggestions of Improving the IxLoad

**eGTP Impairment**

The real network condition is not always as ideal as expected, especially LTE network. LTE is open for both trusted access and non-trusted access. There is a higher potential risk that some devices or software programs used in LTE network are not compliant to specification strictly and thus generate exception events that LTE network may not be able to handle. It is important for IxLoad to be able to simulate both main flows and exceptional flows.

In IxLoad, the impairment plug-in is used to manipulate network flow by injecting delay, drop, reorder packets, fragmentation or errors. IxLoad has developed impairment plug-in inside the GTP stack, but the eGTP stack is not supported. GTP stack is used for emulating the traffic of GSM and UMTS. LTE’s traffic is based on eGTP stack. Compared with GSM and UMTS, LTE suffers more interference from UEs. Hence this impairment plug-in is more essential for eGTP stack. With this plug-in, IxLoad can not only simulate the ideal LTE traffic but also emulate the real LTE traffic.

**eGTP Wizard**

In the verification of the eGTP templates of IxLoad, most of eGTP rxf files have a set of the configurations in common, such as the network element IP address and VLAN, UE and APN configurations. At present, when creating a new eGTP based rxf, user needs to add the emulated eGTP stack over IP first, and then modify the all the default values that are provided by Ixia to the values that match the real test environment. The user must also click, modify and change the panels a lot. All the actions do not seem convenient. Considering that IxLoad has already developed some Net Traffic wizards in the GUI, it is recommended that IxLoad also provides an eGTP wizard for generating a basic eGTP configuration file.

**IxLoad GUI**

In the current version of IxLoad, with any mis-operation, the user has to rollback to the previous configuration manually. Sometimes it will cause
a lot of extra work. It is needed that IxLoad introduces an undo and redo operation.

7.2 Feedback to Ericsson (Ericsson Confidential)
Due to the Ericsson confidential rule, this section will not be shown in the official report.
8 Conclusions

8.1 Conclusions to IxLoad Assessment

This master thesis aims to assess the capacity and LTE functionality of IxLoad in the MPG test environment. A brief introduction to the EPS, EPC, the functionality of MPG in addition to the role of MPG in EPS is given to assist the overall understanding of the readers in relation to the LTE network and the Ericsson MPG. In a similar manner IxLoad overview is presented to spread the common sense of IxLoad.

Using the above knowledge based on the goals made in Section 1.4, several test cases have been designed to verify the capacity and the functionality of IxLoad. The capacity estimation verifies IxLoad can actually support the maximum number of simulated users as promised by Ixia, in the MPG with a particular application. The maximum throughput that IxLoad can achieve with the particular number of simulated users is also tested and the simulation results basically match the official performance published by Ixia.

During the process of the functionality assessment, this thesis checks the Busy Hour Functionality, Deep Packet Inspection, Multiple APNs as well as the Dynamic Quality of Service Enforcement, in addition to the Session Management such as the Track Area Update and Handover. There was a pass achieved in all the test cases relating to the functionality assessment. With the expectation of the FTP and Application Reply, all the tests involving the capacity estimation have achieved the figures promised by Ixia promised. The issues that have been discovered during the test have been forwarded to Ixia. Some patches corresponding to the issues have already been released by Ixia.
This thesis proves that IxLoad is a powerful real-world traffic emulator which is as good as declared by Ixia in its website. However, it is still not yet the perfect article. Some suggestions are proposed in the thesis, for the majority of which involve extending the functionality of IxLoad and improving its usage. With the suggested improvements, IxLoad would be able to simulate many more common scenarios such as eGTP impairment and it would be easier to operate it by using the eGTP Wizard. From release 5.10 to release 5.30, there has been a qualitative change in IxLoad and it is to be hoped that the next release will still provide a tremendous visual impact.

In summary, this thesis work completes all the concrete and verifiable goals of Section 1.4 and achieves the overall aim of Section 1.2.

8.2 Future Work

2G GSM and 3G UMTS technologies have been adopted by the major operators in the most countries. More than 2 billion subscribers have already proved the success of both technologies. As a reliable successor of UMTS, LTE is facing more challenges than was the case with its predecessor. For example, it has to handle the IP transition from IPv4 to IPv6 smoothly and to seamlessly enable mobility between LTE and other technologies such as WCDMA. It is interesting to emulate these scenarios using IxLoad to check whether all the network element simulators perform in a correct manner. That could be the future work.

IPv4 is the fourth revision of the Internet Protocol, which provides about 4.3 billion IP addresses for Internet connection [35]. Each device must pose at least one IP address in order to connect to the Internet. The rapid development of the Internet industry means that more and more people would like the ability to access Internet and thus there has been an increase in devices such as IPTV which are required in order to make this connection. These requirements mean that there is an acceleration in the likelihood that the IPv4 addresses will run out. The last block of IPv4 address was handed out in Feb 2011. It means that IPv4 address will be exhausted officially. The IPv6 will inevitably replace the IPv4 but it is still in its infancy and thus the two will have to coexist for a period of time. In LTE, there are three types of PDN connections: IPv4, IPv6 and the dual stack IPv4/IPv6 which are supported by EPS. For the first two types, PGW assigns either an IPv4 or IPv6 address to UE for PDN connection. In the dual stack model, an IPv4 address and an IPv6 address are allocated to UE during the PDN connection setup. Hence, UE can
use an IPv4 address to communicate with other devices of the same PDN, which only support IPv4 address. The IPv6 address has been developed for exchanging data with devices which are using an IPv6 address in PDN. Because most of the old Internet devices are still using an IPv4 and the new Internet devices start to adopt the IPv6 in order to avoid an address bottleneck of for IPv4, the dual stack IPv4/IPv6 will be a common solution for the Internet connection, until the IPv4 is completely removed from the Internet. All the test cases for this thesis are based on the single stack model. In the future, more time should be spent on testing network element behaviour in the dual stack mode.

The US research organization Maravedis reports that the top 25 operators have committed to LTE and the investment for the LTE infrastructure will amount to 14 billion dollars by 2015 [36]. Because of resource limitation, it is impractical to cover LTE services to all the regions within a short time period. Operators prefer to firstly launch LTE services in the urban area before the suburbs. For subscribers, it is unacceptable, if all the services are lost as soon as they move to the suburbs. Operators also want to reuse all the radio access resources of the previous network. A compromise solution thus appears; by using mobility technologies, subscribers are allowed to move seamlessly between LTE and other radio access networks. Intra LTE mobility was introduced in Section 4.3.3.

In addition, there is also Inter RAT mobility and Inter Technology mobility. Inter RAT mobility refers to mobility between LTE and the earlier 3GPP technologies, such as GSM and WCDMA. The mobility between LTE and non 3GPP technologies such as Worldwide Interoperability for Microwave Access (WiMAX) is Inter Technology mobility. Compared to the Inter Technology mobility, the Inter RAT mobility is more mature, since these technologies are all standardized by 3GPP. This thesis has had both time and test environment limitations in order to assess the mobility between LTE and the earlier 3GPP technologies with IxLoad.

Moreover, S1-based handover has not been verified in this thesis, as mentioned in Section 3.1.5. S1-based handover of intra LTE mobility and inter Technology mobility could also form part of future work.
References


References


[29] Ixia, “IxLoad: Overview”，

[30] Ixia, “IxExplorer”，


[36] Roger Field, "Research indicates 226 million LTE users by 2015", 
## Appendix A: Open Issues of IxLoad

Detailed issue statements and status are referred in the document: Issue_List_last_update_0329.xls

<table>
<thead>
<tr>
<th>Issue No.</th>
<th>Issue Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
<td>BUG592817: inconsistent capture filter will cause port crash.</td>
</tr>
<tr>
<td>0002</td>
<td>S1-based MME relocation handover has failures in back to back.</td>
</tr>
<tr>
<td>0003</td>
<td>BUG595511: “Modify on the fly” not works properly.</td>
</tr>
<tr>
<td>0004</td>
<td>In Max UE POP3 case, there is an error “Missing recovery IE” at the beginning of running.</td>
</tr>
<tr>
<td>0005</td>
<td>IxLoad does not support Ericsson S1-based handover without indirect data forwarding. It is a feature request to Ixia.</td>
</tr>
<tr>
<td>0006</td>
<td>BUG598610: IxLoad failed to clean all the sessions in the HTTP throughput test. The eGTP counters were not correct either.</td>
</tr>
<tr>
<td>0007</td>
<td>BUG????: port crash in HTTP 1.2M UE case when the user number reaches high.</td>
</tr>
<tr>
<td>0009</td>
<td>BUG????: IxLoad sends redundant Create Session Request to MPG in the throughput test, which causes session failures.</td>
</tr>
<tr>
<td>0010</td>
<td>BUG602996: GTP-u packets drop in S1-based no MME relocation and no SGW relocation back to back test.</td>
</tr>
<tr>
<td>0011</td>
<td>BUG598771: stat manager error when clicking on the stats windows during the test running.</td>
</tr>
<tr>
<td>0012</td>
<td>Error “Process ‘kse_classic’ has exited prematurely on node (1,1,12)” occurred in the test case which has multiple applications.</td>
</tr>
<tr>
<td>0013</td>
<td>“Failed to retrieve or write statistics on the stats plug-in” error in the X2 handover case in MPG.</td>
</tr>
<tr>
<td>0014</td>
<td>BUG603443: in S1-based with MME relocation and SGW relocation handover case, IxLoad assigned the users to different eNodeB at the beginning of the simulation.</td>
</tr>
<tr>
<td>0015</td>
<td>Packets are dropped in MPG with the policy rule, and IxLoad does not have the correct counter for the policy value.</td>
</tr>
<tr>
<td>Issue No.</td>
<td>Issue Description</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------</td>
</tr>
<tr>
<td>0001</td>
<td>BUG592817: inconsistent capture filter will cause port crash.</td>
</tr>
<tr>
<td>0002</td>
<td>S1-based MME relocation handover has failures in back to back.</td>
</tr>
<tr>
<td>0017</td>
<td>No GTP-c transaction rate statistics in IxLoad.</td>
</tr>
<tr>
<td>0019</td>
<td>IxLoad cannot handle X2-based handover with 32 eNodeBs. BUG598610: pending sessions in the X2-based handover in MPG with 32 eNodeBs.</td>
</tr>
<tr>
<td>0021</td>
<td>BUG566221: in handover case, IxLoad forwards the uplink data to the incorrect eNodeB.</td>
</tr>
<tr>
<td>0022</td>
<td>An unknown error with the GUI problem, when refreshing the chassis ports.</td>
</tr>
<tr>
<td>0023</td>
<td>BUG????: captured packet number is mismatch with received packet number in Analyzer.</td>
</tr>
<tr>
<td>0024</td>
<td>TCP acceleration is not efficient in a single user test case.</td>
</tr>
<tr>
<td>0025</td>
<td>L2/L3 stats group does not work. A server reboot is needed to recover this issue.</td>
</tr>
<tr>
<td>0026</td>
<td>Ixia chassis card 2 disappeared when running a throughput test. Chassis reboot is needed to recover.</td>
</tr>
<tr>
<td>0027</td>
<td>eGTP stats group only show up when the test case is configured.</td>
</tr>
<tr>
<td>0029</td>
<td>In the advanced subscriber mode, when it goes to the 8th session, IxLoad has a port crash problem besides the session failure.</td>
</tr>
<tr>
<td>0030</td>
<td>The new version of IxLoad has an issue in open old rxf file with the APN configuration.</td>
</tr>
<tr>
<td>0031</td>
<td>The 4-way and 3-way teardown has some impact in the DPI test.</td>
</tr>
<tr>
<td>0032</td>
<td>IxLoad 5.30.200.10 has GUI crash issue in generating report.</td>
</tr>
</tbody>
</table>
Appendix B: Official TR of MPG

(Ericsson Confidential)