MAPS™ UMTS for IuPS Interfaces Emulator

(UMTS IuPS Emulation over IP)
MAPS™ UMTS for IuPS Interfaces

**MAPS™ UMTS-IuPS Emulator**
- Generates up to 20,000 Subscribers
- Up to 2000 Simultaneous Calls

**MAPS™ UMTS-IuPS Emulator**
- PacketLoad Hardware
  - (w/ 4 x 10G cards; w/ 4 x 1G cards)
  - Generates up to 4 or 40 Gbits/sec
  - Stateful TCP/HTTP Traffic

**3G/3.5G**

**Femtocell**
- HnB Gateway
- IuPS

**Node B**
- User
- Uu

**RNC**
- MSC
- SGSN

**HnB**
- IuH

**Interface**
- Circuit Switched Domain
- Packet Switched Domain
Highlights

- Simulates RNC, and SGSN entities.
- Generates and supports all Mobility Management, Session Management, RANAP and DTAP messages.
- User controlled access to SCTP, M3UA, and SCMG (SCCP Management), RANAP, and DTAP messages.
- Ready scripts for Routing Area Updating, GPRS Attach, Paging, and Handover (Relocation) procedures for quick testing.
- Supports Authentication, TMSI Reallocation, Encryption, and other optional procedures.
- Supports mobile traffic simulation with additional licenses.
- Provides fault insertion, and erroneous call flows testing capability.
- Impairments can be applied to messages to simulate error conditions.
- Setup a virtual real-time network simulating 3G-UMTS network element using ‘MAPS™ 3G Wireless Lab Suite’.
- Supports PS network web browsing data traffic simulation along with generation of real-world traffic in the lab.
- Supports M3UA termination type as a signaling gateway process (SGP), an application server process (ASP), or an IP server process (IPSP).
- Supports Packet Traffic, Mobile Traffic, Gateway Traffic, External Gateway simulation with additional licenses.

GL Communications Inc.
## UMTS IuPS Protocol Stack

### Supported Protocols and Specification Used

<table>
<thead>
<tr>
<th>Supported Protocols</th>
<th>Specification Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>IuPS Interface</td>
<td></td>
</tr>
<tr>
<td>SCTP</td>
<td>RFC 4960</td>
</tr>
<tr>
<td>SCCP</td>
<td>Q.713, CCITT (ITU-T) Blue Book</td>
</tr>
<tr>
<td>M3UA</td>
<td>RFC 3332</td>
</tr>
<tr>
<td>RANAP</td>
<td>3GPP TS 25.413 V9.1.0</td>
</tr>
<tr>
<td>GMM / SM</td>
<td>GPP TS 24.008 V5.16.0 (2006-06)</td>
</tr>
<tr>
<td>SMS</td>
<td>3GPP TS 03.40 V7.5.0 &amp; 3GPP TS 04.11 V7.1.0 GSM 03.38 version 7.2.0 Release 1998</td>
</tr>
</tbody>
</table>

### Control Plane

- GMM | SM | SMS
- RANAP
- SCCP
- M3UA
- SCTP
- IP
- MAC

### User Plane

- IuUP
- GTP-U
- UDP
- IP
- MAC

UMTS IuPS over IP
UMTS IuPS Paging Call Procedure

Authentication, Ciphering, Security Mode Procedure
- Security Mode Command
- Security Mode Complete
- Service Accept
- Paging Service Request
- CC Connection Confirm

RAB Assignment Procedure
- Activate PDP Context Accept
- Deactivate PDP Context Request
- Deactivate PDP Context Accept
- Detach Request
- Detach Accept
- Iu-ReleaseCommand
- Iu-ReleaseComplete
- RLSD Released
- RLC Release Complete

PS Domain PSTN, ISDN
UMTS IuPS Relocation Call Flow

Authentication, Ciphering, Security Mode Procedure
- Attach Request
- Attach Accept
- Attach Complete
- Activate PDP Context Request
- Activate PDP Context Accept

RAB Assignment Procedure
- Relocation Request
- Relocation Command
- Deactivate PDP Context Request
- Deactivate PDP Context Accept
- Detach Request
- Detach Accept
- Iu-ReleaseCommand
- Iu-ReleaseComplete

PS Domain (PSTN, ISDN)

SGSN

RNC

IuPS

IuB

User
IuPS Over IP Testbed Setup
IuPS over IP Profile Editor
IuPS over IP Incoming Call Handle Configuration
IuPS over IP Call Generation

Loading Scripts and Profiles

Message Sequence

Decode Message

Active Calls  Call Status  Call Events
IuPS over IP Call Reception

Message Sequence

Decode Message

Call Results
IuPS over IP Events and Traffic Log

Events Log

Traffic Log
Load Generation

- Stability/Stress and Performance testing using Load Generation
- Different types of Load patterns to distribute load
- User can load multiple patterns for selected script
- User configurable Test Duration, CPS, Maximum and Minimum Call Rate etc.

Fixed

Ramp

Uniform

Normal

Saw-tooth

Step
IuPS over IP Bulk Call Generation
IuPS over IP Call and Message Statistics

Call Statistics

Message Statistics

GL Communications Inc.
Schedule Test to Run Automatically
Customizations - Call Flow (Scripts)

- Scripts are written in our proprietary *.gls scripting language. They represent generic state machines intended to provide protocol/signaling logic for a call and establish bearer traffic.

- Each instance of a script corresponds to a single transaction/call, i.e., if you place 500 calls in parallel you will actually have 500 script instances running at once. If you place 500 calls in series the same script will execute and terminate 500 times.

- It is possible to create your own scripts, but almost never necessary! We attempt to provide all necessary scripts out of the box.
Customizations - Protocol Messages

When the script actually sends a message it does so by loading a hdl file template from disk ("AttachRequest.hdl" in the right hand screenshot).

These message templates provide the actual structure of the message, the script simply populates it with values contained in its variables.

These messages are customizable by the user, header fields can be altered and removed. Binary-based messages are edited in our provided message editor.
Packet Statistics and Reports

TCP/IP
- SYN, SYN_ACK
- ACK, FIN, RST
- HTTP POST / RESPONSE
- TCP/IP Checksum Errors

PCAP Replay
- Packets Sent and Received

UDP
- Packets Sent and Received

URL
- Connections Established
- FW Addresses Not Blocked
- URL HTTP Wrong Response RX
MAPS™ API Architecture

- API wraps our proprietary scripting language in standard languages familiar to the user:
  - Python
  - Java
  - VB Scripts
  - TCL

- Clients and Servers support a “Many-to-Many” relationship, making it very easy for users to develop complex test cases involving multiple signaling protocols.

GL Communications Inc.
The API is broken into High and Low level function calls / scripts.

For High Level scripts, all the fine-grained protocol control happen in the script running on the MAPS server, hidden from the API user.

Low Level scripts put the API user in complete control of the protocol stack. This makes Low Level scripts more flexible and powerful, but also correspondingly more complex.
CLI/API Support
THANK YOU