Signaling and Traffic Simulation using MAPS
Message Automation and Protocol Simulation (MAPS™)

MA - Message Automation
+
PS - Protocol Simulation
• MAPS stands for Message Automation and Protocol Simulation
• It is a generic framework for the generation of telecommunications protocol messages and transmission of bearer traffic
• MAPS™ is built on a proprietary scripting language developed by GL Communications
• All MAPS™ products come with out-of-the-box scripts that act as fully functional state machines for the relevant protocol
About MAPS™ (Contd.)

- **Scripts**: Scripts act as the state-machine, or engine for a given call. The logic of what messages to send when is all contained in a script.

- **Messages**: MAPS has an inventory of generic Message Templates (ex: Invite.txt) which it loads from the hard drive when transmitting an actual message. Messages are completely customizable.

- **Profiles**: Scripts and Messages are kept as generic as possible. Specific information (ex: Contact = 12345@sip.carrier.com) about a call is sourced from .xml profiles.
Basic Requirements for Emulation

• **Message Templates**
  - The message templates are nothing but structure of message stored in particular file format. e.g.: SS7 Protocol suite message template will have “. HDL” format.

• **A ‘Script’**
  - To send and receive these messages between two nodes and take appropriate actions for a particular message.

• **An ‘Import’ mechanism**
  - A mechanism for reading the contents of the message template, and replacing the Key Identifier with the value given by the user (or some other means) at the run time. This process of inserting the user values into the message template before sending is called “Import”.
Basic Requirements for Emulation (Contd.)

- **An ‘Export’ mechanism**
  - A mechanism to extracting Key Identifier values from the received response and store for the future use (in the same call scenario) is called “Export” (This exported value can also be imported to message template in future)

- **A ‘Profile’ file**
  - Once the Key Identifiers are identified for all the message templates in a call scenario, required values are configured for these Key Identifiers in a file called Profile
Supported Protocols / Interfaces

**LTE (X2, S1, eGTP, SGs, SLs)**
- IMS, Diameter

**5G (N1N2, N4, N8, N10, N11, N12, N13, N14, N17, N20, N21, N22, 29, S1)**

**SIP, Megaco, MGCP, SIP-1, SIGTRAN (ISUP, ISDN)**

**4G Access (E-RAN)**
- Maps™ HD
- 4G devices

**VoIP (Packet Network)**
- SIP, RTP, H.323, T.38

**VoIP (Packet Network)**
- VoIP devices

**3G Access (UTRAN)**
- 3G access
- Maps™ HD
- UMTS (IuH, IuCS, IuPS, GnP) GPRS

**2G Access (GERAN)**
- 2G access
- CAS, ISDN, MLPPP, SS7, IUP, MAP, CAMEL, INAP, GSM (A, Abis)

**Signaling, Traffic Generation and Analysis**
- Maps™ 1G – Software-based, or Hardware-based
- Maps™ HD 40G, 10G, 1G – Rackmount Platforms
Common Protocol Emulation Framework

LTE Emulation

SS7 Emulation

SIP Emulation
Working Principle

Message Automation and Protocol Simulation (MAPS)

- **Message Templates**
  - Scripts: Create Call Sequence
  - Profiles: Assign values to variables
  - Event Profiles: Load Event Profile

- **Call Control**
  - CALL GENERATION
    - (Load Scripts, Profiles)
  - CALL RECEPTION

- **IP/TDM Network**

  **DUT**
Customize Test Scenarios using Scripts

- Unlimited access in creating test scenarios
- Build valid or invalid and conformance test cases
- A simple, easy to learn but very powerful scripting language
- Can be an Extremely simple scripts to test a particular scenario. But Flexible enough to simulate a complete protocol state machine
- A GUI based 'Script Editor' helps to build scripts even before syntax and semantics of the scripting language is familiar
Send "Initial Address" "InitialAddressImport";

Recv "Address Complete" "AddressCompleteExport";

Recv "Answer" "AnswerImport";

TxRx:tx_TDM file: filename = "Vijay.pcm";

Send "Release" "ReleaseImport";

Recv "Release Complete" "ReleaseCompleteExport";
Customize Protocol Messages

Message Editor
Call Generation

Active Calls
Call Status
Call Events

Loading Scripts and Profiles

Message Sequence

Decode Message
Fine Control over Call Behavior
User Events

Control moves to “Hold” section, after selecting the “Hold” User Event:

```
"Hold":
    CallHoldInitiated = 1;
    [ISUPScriptId] goto "Hold";
    resume;
```

Control moves to “Retrieve” section, after selecting the “Retrieve” User Event:

```
"Retrieve":
    CallHoldInitiated = 0;
    [ISUPScriptId] goto "Retrieve";
    resume;
```

```
"Suspend":
    SuspendInitiated = 1;
    [ISUPScriptId] goto "Suspend Call";
    resume;
```
Script Contents

"PlaceCall"(opc,dpc,cic):
    StartChildScript (ISUPScriptId,"ISUP","ISUP.qls",LoadedProfileName);
    ActiveUserEvent: Add:"Terminate Call","Initiate Reset","Clear Call";
    Status = "ISUP Call Initiated";
    ISUPState = "ISUP CALL INITIATED";
    (ISUPScriptId) goto"ISUPMakeCall":cic,opc,dpc,SLS,NetInd,ConnectionID,StreamID,
    return;

"OnISUPCallInitiated"(opc,dpc,cic):
    ReportEvent (InitialAddress = "Initial Address");
    resume;

"OnISUPCallProgressReceived":
    ReportEvent (AddressComplete = "Address Complete");
    resume;

"OnISUPCallConnected":
    Result="Pass";
    ReportEvent (Answer = "Answer");
    Status = "ISUP Call Connected";
    ISUPState = "ISUP CALL CONNECTED";
    if (StopAll=1)
        goto "Terminate Call":Cause;
    endif
    ActiveUserEvent:Add:"Hold","Suspend";
    ActiveUserEvent:Remove:"Accept Call","Reject Call";
    if (CallDuration != 0)
        starttimer CallDurationTimer CallDuration msec;
    else
        goto "Terminate Call":Cause;
    endif
    return;
### Script Flow

<table>
<thead>
<tr>
<th>Script ID</th>
<th>Script Name</th>
<th>Subscript Name</th>
<th>Line No</th>
<th>Script Flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Isup_Call.gis</td>
<td></td>
<td>68</td>
<td>goto &quot;PlaceCall&quot;;opc,dpc,cic;</td>
</tr>
<tr>
<td>*</td>
<td>Isup_Call.gis</td>
<td></td>
<td>94</td>
<td>&quot;PlaceCall&quot;(opc,dpc,cic);</td>
</tr>
<tr>
<td>*</td>
<td>Isup_Call.gis</td>
<td></td>
<td>95</td>
<td>StartChildScript (ISUScriptId,&quot;ISUP&quot;,&quot;ISUP.gis&quot;,loadedProfileName);</td>
</tr>
<tr>
<td>ISUN</td>
<td>ISUP.gis</td>
<td></td>
<td>5</td>
<td>&quot;Init&quot;:</td>
</tr>
<tr>
<td>ISUN</td>
<td>ISUP.gis</td>
<td></td>
<td>7</td>
<td>ISUSState = &quot;IDLE&quot;;</td>
</tr>
<tr>
<td>ISUN</td>
<td>ISUP.gis</td>
<td></td>
<td>9</td>
<td>ISUResult = &quot;Unknown&quot;;</td>
</tr>
<tr>
<td>ISUN</td>
<td>ISUP.gis</td>
<td></td>
<td>9</td>
<td>SetScriptVariable(ParentScriptId,ISUResult = ISUResult);</td>
</tr>
<tr>
<td>ISUN</td>
<td>ISUP.gis</td>
<td></td>
<td>10</td>
<td>ParentScriptId = &quot;ISUP&quot;;</td>
</tr>
<tr>
<td>ISUN</td>
<td>ISUP.gis</td>
<td></td>
<td>11</td>
<td>Cause = 16;</td>
</tr>
<tr>
<td>ISUN</td>
<td>ISUP.gis</td>
<td></td>
<td>12</td>
<td>CUTFExpected = 0;</td>
</tr>
<tr>
<td>ISUN</td>
<td>ISUP.gis</td>
<td></td>
<td>13</td>
<td>AddressCompleteSent=0;</td>
</tr>
<tr>
<td>ISUN</td>
<td>ISUP.gis</td>
<td></td>
<td>14</td>
<td>KeyIdentifier: opc , dpc , cic ;</td>
</tr>
<tr>
<td>ISUN</td>
<td>ISUP.gis</td>
<td></td>
<td>15</td>
<td>ReleaseInitiated = 0;</td>
</tr>
<tr>
<td>ISUN</td>
<td>ISUP.gis</td>
<td></td>
<td>16</td>
<td>ReleaseReceived = 0;</td>
</tr>
<tr>
<td>ISUN</td>
<td>ISUP.gis</td>
<td></td>
<td>17</td>
<td>CallActive = 0;</td>
</tr>
<tr>
<td>ISUN</td>
<td>ISUP.gis</td>
<td></td>
<td>18</td>
<td>MsgHandler : &quot;ISUPMsgHandler&quot;;</td>
</tr>
<tr>
<td>ISUN</td>
<td>ISUP.gis</td>
<td></td>
<td>19</td>
<td>ReleaseGunTimeStarted=0;</td>
</tr>
<tr>
<td>ISUN</td>
<td>ISUP.gis</td>
<td></td>
<td>21</td>
<td>Wait;</td>
</tr>
<tr>
<td>*</td>
<td>Isup_Call.gis</td>
<td></td>
<td>96</td>
<td>ActiveUserEvent:Add:&quot;Terminate Call&quot;,&quot;Initiate Reset&quot;,&quot;Clear Call&quot;;</td>
</tr>
<tr>
<td>*</td>
<td>Isup_Call.gis</td>
<td></td>
<td>97</td>
<td>Status = &quot;ISUP Call Initiated&quot;;</td>
</tr>
<tr>
<td>*</td>
<td>Isup_Call.gis</td>
<td></td>
<td>98</td>
<td>ISUSState = &quot;ISUP CALL INITIATED&quot;;</td>
</tr>
<tr>
<td>*</td>
<td>Isup_Call.gis</td>
<td></td>
<td>99</td>
<td>ISUScriptId goto&quot;ISUPMakeCall&quot;(ocic,opc,dpc,sls,NetInd,ConnectionId,StreamID, ...</td>
</tr>
<tr>
<td>ISUN</td>
<td>ISUP.gis</td>
<td></td>
<td>32</td>
<td>&quot;ISUPMakeCall&quot;(ocic,opc,dpc,sls,NetInd,ConnectionId,StreamID, CallingNumber, CalledNum...</td>
</tr>
<tr>
<td>ISUN</td>
<td>ISUP.gis</td>
<td></td>
<td>33</td>
<td>send &quot;InitialAddress&quot; &quot;InitialAddressImport&quot; &quot;StreamID&quot; = StreamID &quot;ConnectionID&quot;...</td>
</tr>
<tr>
<td>ISUN</td>
<td>ISUP.gis</td>
<td></td>
<td>34</td>
<td>if [ContinuityIndicator=0]</td>
</tr>
<tr>
<td>ISUN</td>
<td>ISUP.gis</td>
<td></td>
<td>36</td>
<td>endif;</td>
</tr>
<tr>
<td>ISUN</td>
<td>ISUP.gis</td>
<td></td>
<td>37</td>
<td>ISUSState=&quot;CALL INITIATED&quot; ;</td>
</tr>
<tr>
<td>ISUN</td>
<td>ISUP.gis</td>
<td></td>
<td>38</td>
<td>Status = &quot;Call Initiated&quot; ;</td>
</tr>
<tr>
<td>ISUN</td>
<td>ISUP.gis</td>
<td></td>
<td>39</td>
<td>EventLog (&quot;Call Initiated&quot;);</td>
</tr>
<tr>
<td>ISUN</td>
<td>ISUP.gis</td>
<td></td>
<td>40</td>
<td>starttimer T7,7TimeOut;</td>
</tr>
<tr>
<td>ISUN</td>
<td>ISUP.gis</td>
<td></td>
<td>41</td>
<td>(ParentScriptId) goto &quot;OnISUPCallInitiated&quot;;opc,dpc,cic;</td>
</tr>
</tbody>
</table>
# Incoming Call Handler

![Incoming Call Handler Configuration](image)

<table>
<thead>
<tr>
<th>Message Name</th>
<th>Script Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signalling Link Test Message</td>
<td>SLTM.gls</td>
</tr>
<tr>
<td>Initial Address</td>
<td>Isup_Call.gls...</td>
</tr>
<tr>
<td>Release</td>
<td>Rx_CIC_Management.gls</td>
</tr>
<tr>
<td>Reset Circuit</td>
<td>Rx_CIC_Management.gls</td>
</tr>
<tr>
<td>Continuity Check Request</td>
<td>Rx_CIC_Management.gls</td>
</tr>
<tr>
<td>Blocking</td>
<td>Rx_CIC_Management.gls</td>
</tr>
<tr>
<td>Unblocking</td>
<td>Rx_CIC_Management.gls</td>
</tr>
<tr>
<td>Circuit Group Reset</td>
<td>Rx_CIC_Management.gls</td>
</tr>
<tr>
<td>Circuit Group Blocking</td>
<td>Rx_CIC_Management.gls</td>
</tr>
<tr>
<td>Circuit Group Unblocking</td>
<td>Rx_CIC_Management.gls</td>
</tr>
<tr>
<td>Release Complete</td>
<td>Rx_CIC_Management.gls</td>
</tr>
</tbody>
</table>

**Scripts**
- Isup_Call.gls
- Isup_Call - Reject.gls
- Isup_Call - Forward.gls
- Isup_Call - Conference.gls
Incoming Call Handler

<table>
<thead>
<tr>
<th>IAM</th>
<th>AnswerCall.gls</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACM</td>
<td></td>
</tr>
<tr>
<td>ANM</td>
<td></td>
</tr>
</tbody>
</table>

MAPSTM searching for appropriate script to load against the received msg

I AM CIC=1

Answer call Script is bind to the recv msg with CIC=1

I AM CIC=1

Another IAM msg is received with CIC=1

I AM CIC=1

2Another IAM msg is received with CIC=

Answer Call Script ID 1 CIC=1

Answer Call Script ID 2 CIC=2

Recv msg is bind to the same script

Binds to the new script with ID=2

Script Loaded
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.
Success Call Ratio Statistics

MAPS™ Features

Call Graph

Call Stats
Message Statistics

- Message Stats provides a running tabular log of all messages transmitted, retransmitted and received during the session.
- Provides an easy way to monitor the reception of error responses during load generation.

<table>
<thead>
<tr>
<th>Message Type</th>
<th>Tx Count</th>
<th>Rx Count</th>
<th>Retransmit Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALERTING</td>
<td>240</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CALL PROCEEDING</td>
<td>240</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CONNECT</td>
<td>240</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CONNECT ACKNOWLEDGE</td>
<td>0</td>
<td>240</td>
<td>0</td>
</tr>
<tr>
<td>SETUP</td>
<td>0</td>
<td>240</td>
<td>0</td>
</tr>
<tr>
<td>DISCONNECT</td>
<td>0</td>
<td>210</td>
<td>0</td>
</tr>
<tr>
<td>RELEASE</td>
<td>210</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RELEASE COMPLETE</td>
<td>0</td>
<td>210</td>
<td>0</td>
</tr>
</tbody>
</table>
User Defined Graphs and Statistics
Traffic Simulation

Simulate User Equipment
- HD Voice
- FAX

Interfaces

- All Protocols
  - 5G, N102, N1A, N8, N10, N13, N12, N13, N12...
  - LTE, S1, eGTP, X2, SGs, Diameter
  - GPRS, UMTS ISCS, ISUP, ISUP, GnGp
  - SIP, SIP, Magaco, MGCP, SIGTRAN (SS7, ISDN)
  - CAS, ISDN, SS7, IUP, MEPPP, GSM, Aini, BICC, MAP, CAP, INAP

Hundreds of thousands of simultaneous calls

Load, Stress, Performance
- 5G, 4G, 3G, 2G Networks
- GTP Traffic
- RTP Media
- IP Networks
- TDM, PSTN Networks
- Voice, Digits, Tones, Fax, Video, SMS, Data, Packet, GTP

MAPS™ HD
High Density (HD) Call Generator
Supported RTP Codecs

- **PCMU/PCMA**: 64kbps, 8000Hz, VAD
- **G.722/G.722.1**: 24/32/64kbps, 16000Hz, No VAD
- **G.729/G.729B**: 8kbps, 8000Hz, VAD
- **GSM 6.10 FR**: 13.2kbps, 8000Hz, No VAD
- **GSM EFR**: 12.2kbps, 8000Hz Yes VAD
- **GSM**: 5.6kbps, 8000Hz, Yes VAD
- **G.726**: 16/24/32/40kbps, 8000Hz, Yes VAD
- **AMR**: 4.75/5.15/5.9/6.7/7.4/7.95/10.2/12.2kbps, 8000Hz, Yes VAD *(OPTIONAL LICENSE)*
- **AMR WB**: 4.75/5.15/5.9/6.7/7.4/7.95/10.2/12.2kbps, 16000Hz, Yes VAD *(OPTIONAL LICENSE)*
- **EVRC**: 1/8, 1/2, 1 rate, 8000Hz, No VAD *(OPTIONAL LICENSE)*
- **EVRC_B**: 1/8, 1/2, 1 rate, 8000Hz, Yes VAD *(OPTIONAL LICENSE)*
- **EVRC_C**: 1/8, 1/2, 1 rate, 16000Hz, Yes VAD *(OPTIONAL LICENSE)*
- **SMV**: Modes 0,1,2 and 3, 8000Hz, No VAD *(OPTIONAL LICENSE)*
- **ILBC**: 15.2/13.33kbps, 8000Hz, No VAD
- **SPEEX**: 8kbps, 8000Hz, Yes VAD
- **SPEEX WB**: 11.2kbps, 16000Hz, Yes VAD
TDM Traffic Simulation

Tx
- Pre recorded PCM files
- DTMF, MF, MFR2B and MFR2F Digits
- User Defined Tones
- FAX
- AAL2

Rx
- PCM files
- DTMF, MF, MFR2B and MFR2F Digits
- User Defined Tones
- FAX
- AAL2
RTP Traffic Simulation

IP Network

MAPSTM

Voice Channels

PSTN/IP Phones

Tx
- Pre recorded GLW files
- DTMF, MF Digits
- User Defined Tones
- Insert Voice
- FAX T.30

Rx
- GLW files
- DTMF, MF Digits
- User Defined Tones
- FAX T.30
Multi Interface Simulation

SS7 Point Code → SSP

SS7

Signaling Gateway → MAPS™ MGC

SIGTRAN

IP: 192.168.1.X

SIP

IP Phone → Media Gateway

MEGACO

RTP

Media Gateway → MAPS Multi Interface

TDM
Multiple Transport Support

Dual T1 E1 Express (PCIe) Board  Quad / Octal T1 E1 PCIe Card  Rackmount Quad T1 E1 Analyzer

16-Port T1 E1 Breakout-Box

tProbe™ - Portable USB based T1 E1 VF FXO FXS and Serial Datacom Analyzer
IP variants of MAPS™ can be run on any modern Windows server.

A typical i7 platform will be able to handle ~2000 concurrent RTP sessions through a conventional server-grade NIC.

We also offer an HD (High Density) appliance which can deliver up to 20,000 concurrent RTP sessions per Unit of rack space.
High Density (HD) RTP Traffic Simulation

- Rackmount network appliance with 4x1GigE NIC
- Transport over UDP and TCP, IPv4 and IPv6, and TLS for secure transport
- Easily achieve up to 20,000 endpoints per appliance (5000 per port)
- Up to 350 calls per second (with RTP traffic)
- Scales to around 100,000 to 200,000 endpoints with use of Master Controller for single point of control
- Manage 10+ MAPS™ systems with single point of control from master controller
Introduction to MAPS™ Configurations

- Testbed Setup
- Global Configuration
- Profiles
Local and Global Variables

Test Bed

Global Profiles

Global Variables

Profile 1

Profile n

List of Instructions

Local Variables

List of Instructions

Local Variables
Testbed Configuration

[Image of a software interface showing testbed configuration details]
## Global Configuration

![Global Configuration - Global Profile]

<table>
<thead>
<tr>
<th>Config</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call Parameters</td>
<td></td>
</tr>
<tr>
<td>Call Answer Time in msec</td>
<td>1000</td>
</tr>
<tr>
<td>Call Duration in msec</td>
<td>30000</td>
</tr>
<tr>
<td>Inter Call Duration in msec</td>
<td>1000</td>
</tr>
<tr>
<td>Randomization Parameters</td>
<td></td>
</tr>
<tr>
<td>Enable Randomization in msec</td>
<td>Disable</td>
</tr>
<tr>
<td>Minimum CallDuration in msec</td>
<td>60000</td>
</tr>
<tr>
<td>Maximum CallDuration in msec</td>
<td>1000</td>
</tr>
<tr>
<td>Minimum AnswerCallDuration in msec</td>
<td>1000</td>
</tr>
<tr>
<td>Maximum AnswerCallDuration in msec</td>
<td>30000</td>
</tr>
<tr>
<td>Minimum InterCallDuration in msec</td>
<td>3000</td>
</tr>
<tr>
<td>Maximum InterCallDuration in msec</td>
<td>20000</td>
</tr>
<tr>
<td>Enable Location Update</td>
<td>Enable</td>
</tr>
<tr>
<td>Call Control Failure Parameters</td>
<td></td>
</tr>
<tr>
<td>Enable Call Control Failure Parameters</td>
<td>False</td>
</tr>
<tr>
<td>Call Control Failure Cause</td>
<td>User alerting, no answer</td>
</tr>
<tr>
<td>IMIA Parameters</td>
<td></td>
</tr>
<tr>
<td>Dynamic Registration</td>
<td>Not Required</td>
</tr>
<tr>
<td>Destination Audit</td>
<td>Not Required</td>
</tr>
<tr>
<td>IMIA Specific Timers</td>
<td></td>
</tr>
<tr>
<td>TaskTimer in msec</td>
<td>2000</td>
</tr>
<tr>
<td>ThrustTimer in msec</td>
<td>10000</td>
</tr>
<tr>
<td>DAILDTimer in msec</td>
<td>5000</td>
</tr>
<tr>
<td>SCMG Timers</td>
<td></td>
</tr>
<tr>
<td>Tmitinfo in msec</td>
<td>30000</td>
</tr>
<tr>
<td>iuCS Protocol Specific Timers</td>
<td></td>
</tr>
<tr>
<td>Tbis in msec</td>
<td>10000</td>
</tr>
<tr>
<td>TJO3 in msec</td>
<td>30000</td>
</tr>
<tr>
<td>TM in msec</td>
<td>10000</td>
</tr>
</tbody>
</table>
Sample Profile
MAPS™ Scripts can be written in different ways as we have flexible commands such as \textbf{Go to}, \textbf{IF Else IF}, \textbf{Timers}, \textbf{Actions}, \textbf{User Events} etc.

\begin{itemize}
  \item \textbf{Two Types of Scripting}
    \begin{itemize}
      \item Simple, Non-Event driven
      \item Event Driven
    \end{itemize}
  \item \textbf{Non-Event driven:} Defines flow sequentially without monitoring any events. These can be small and simple scripts using send and receive actions
  \item \textbf{Event Driven:} Defines flow on basis of user selected events. Using Event Driven scripting one can achieve Protocol State Machines as per protocol specifications
\end{itemize}
//Script Description 

//Initialization Section
--------
--------

//Action section
Send “MessageName” “ImportFile Name”; 
Recv “Message” “ExportFile Name”; 
Result = “Pass”; 
State = “…………..”; 
Status = “……………..”; 
Exit;
Script Events

• **Message Handler:** On Receipt of any Message Event control move to defined section in script

• **Traffic Event:** On detection of any traffic actions, Control move to detected Traffic Event Section like “Digits Detected”, Tone Detected”, etc.

• **Timer:** On Expiry of Timer, control moves to respective Timer Expiry section

• **User Events:**
  - Within scripts: Goto “Label”
  - User Intervention: User Event
  - Intervention from another Script: Apply Event to another script
### Structure of Event Driven Script

//Script Description .................

//Initialization Section
- Initialize Variables
- Initialize Message Handler

Message Handler Section
- "Message 1":
- "Message 2":

"Timer Expiry Section":

"Traffic Handler Section":

"User Events Section":

---

![Diagram of script structure](image-url)
Below call flow scenario using MAPS Script

Send “Initial Address” “InitialAddressImport”;
Recv “Address Complete” “AddressCompleteExport”;
Recv “Answer” “AnswerImport”;
_TxRx:tx _TDM file: filename = “Vijay.pcm”;
Send “Release” “ReleaseImport”;
Recv “Release Complete” “ReleaseCompleteExport”;

Initial Address Message (IAM)
Address Complete Message (ACM)
Answer Message (ANM)
Voice
Release (REL)
Release Complete (RLC)
Sample Script
Understanding Send and Receive Messages
Basic Receive/Decode Command

Decode Message

List of Protocol Fields in a Message

- Protocol
- Message Type
- Mandatory Fixed Parameters
- Mandatory Var Parameters
  - Calling Party Number
  - Calling Plan Indicator
  - Internal Network Number Indicator
  - Called Address Signal
  - Calendar Address Signal
- Optional Var Parameters

Extracted Variables

- Called Address Signal: 5551234
- Called Address Signal: 5551111

Protocol Engine

DECODER

54 68 47 66 54
0A 00 02 32 43
06 05 03 83 15
10 10 A0 68 10

Incoming Message

EXPORT

Called Address Signal
Calling Address Signal
Send Command With Impairment

Called Address Signal = 8978675400
Calling Address Signal = 9987095827

In Message Editor, 8th byte with Offset 7 represents the Message Type Field.

Protocol Engine

DECODER

05 12 50 02 32 01 00
01 02 01 0A 00 02
08 06 83 10 55 15 11
01 0A 06 81 11 55 15

Message Template

DECODER

IMPORT

DECODER

ENCODER

05 12 50 02 32 01 00
02 20 01 0A 00 02
08 06 83 10 55 15 11
01 0A 06 81 11 55 15

Message Type

Optional Var Parame

Protocol Fields

Called Address Signal

Calling Address Signal

Message Type

Optional Var Parame

Protocol Fields

Message Template

IMPAIR

Create MsgTypeImpair.dat file
IMPAIR REP 1 OFFS 7 AND 00

Message Sent with Impairment

DUT
CLI/APIs for Remote Control and Test Automation
Python Client

```
INVITE sip:0001@192.168.1.36; SIP/2.0
Via: SIP/2.0/UDP 192.168.1.36:5060;branch=z9hG4bK_178932388-5260-12832
Max-Forwards: 70
Allow: INVITE,BYE,CANCEL,ACK,INFO,OPTIONS,SUBSCRIBE,NOTIFY, REFER,REGISTER
From: C001 <sip:0001@192.168.1.36>;tag=TagFromTLS_178932388-5377-12832
To: C001 <sip:0001@192.168.1.36>
Call-ID: GL-MAPS_4_178932388-5260-12832@192.168.1.36
Sip:1 INVITE
Contact: 0001 <sip:0001@192.168.1.36>
Supported: 100rel
Content-Type: application/sdp
Content-Length: 594
```

```
54
```
Java Client
Remote MAPS™ Server

- Multi-node and multi-interface simulation from a single GUI
- Suitable for testing any core network, access network, and inter-operability functions
- Single Licensing Server controlling server and client licenses (no. of users)
- Unlimited number of remote client user can be defined at the server
- Admin privileges to control Testbed and access to configuration files for each remote client user
- Remote Client users has privileges to perform all other functions - call emulation, edit scripts/profiles, and view statistics
- Option to license multiple clients either at Remote client systems (MAPS™ Remote Client to control one or more MAPS™ Server - PKS111) or at the MAPS™ Server systems (MAPS™ Server with Multi-user capability - PKS113)
- Simultaneous traffic generation/reception at 100% on all servers

Remote MAPS™ (PKS111 and PKS113)

Client Systems

- MAPS Remote Client
- MAPS Remote Client
- MAPS Remote Client

Server Systems

- MAPS Core
- MAPS Core
- MAPS Core

Multiple Remote MAPS™ Clients connected to single MAPS™ server system

Single Remote MAPS™ Client connected to Multiple MAPS™ server system
Send Reports to Database

- MAPSTM generated reports can be sent to Database using built in commands
- This helps to monitor and analyze test Remotely
Thank you