MAPS™ Skinny Protocol Emulator

Skinny Call Control Protocol (SCCP) Emulation
Skinny Call Control Protocol (SCCP), also referred to as "Skinny", is a Cisco Systems proprietary signaling and control protocol used to communicate between IP devices and Cisco Unified Communications Manager for call establishment, teardown, and control in VoIP environments.
## Main Features

### Signaling
- Simulates Skinny Call Control Protocol (SCCP) IP Phones (Skinny Client).
- User-configurable SCCP signaling links.
- Supports call hold and call resume.
- Each SCCP message template facilitates customization of the protocol fields and access to the various protocol fields from the scripts.
- Scripted call generation and call reception.

### Signaling Procedures
- Registration
- Call control (setup, teardown, and statistics)

### Traffic
- Supports RTP media (digits, voice file, tones, IVR, FAX) over TDM lines.

### Other Features
- Automation, Remote access, and Schedulers to run tests 24/7.
- Supported on Windows® 7, and higher version operating systems.
- Supports 64-bit version to enhance signaling performance***

### Applications
- End-to-End testing of Cisco Call Manager and IP Phones
MAPS™ RTP HD (PKS109) is a specialized 1U rack mounted network appliance designed to easily achieve up to 20,000 endpoints per appliance, it also supports generation of high volume of calls with traffic for load testing network.
MAPS™ Skinny HD Features

- Transport over UDP and TCP, IPv4 and IPv6, and TLS for secure transport
- Unique endpoint emulation using IP address, MAC address, and VLAN tagging
- 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
- Simulate various traffic conditions to measure the performance and capability of an IP network element
- Easy-to-use MAPS™ scripting interface (UI) for test scripts to be created
- User can modify any signalling messages, message parameters, and/or information elements.
- Packet level fault insertion can be performed by impairing any bit / byte of a message.
- Real-time monitoring and reporting of registration and call statistics
- Statistics can be viewed on any pair of endpoints
Skinny Call Flow

A typical call flow between IP Phone and Cisco Call Manager

- **Registration**
  IP Phone registers its IP, type, & name with the CCM, and provides its "Capabilities" (voice/video codec supported) to CCM.

- **Call Control (setup, teardown, and statistics)**
  Phone periodically sends "KeepAlive" messages to the CCM.
  Offhook (place call) - CCM instructs with the lamp on/off, through the prompt, key settings, and the dialtone messages.
  Onhook (end call) – CCM instructs the phone to stop transmitting, close the channels, set the call status to disconnect, and send the default user prompt.

- **Media (audio) Stream Control**
  Media Transmission includes Conference ID, Pass through Party ID, Remote IP & Port Address, Packet, Payload Capability, Max Frames per Packet details.
MAPS™ Skinny can be configured as Softphone (Skinny Client) to generate and to receive calls to/from Cisco Call Manager (DUT).
### Skinny Testbed Setup

#### User Interface Details

![Skinny Testbed Setup Interface](image-url)

- **Config Section**:
  - **Skinny Configuration**:
    - **TCP Transaction Type**: Client
    - **Phone IP Address**: 192.168.1.36
    - **CCM IP Address**: 192.168.20.75
    - **Destination Port**: 2000

- **End User Configuration**:
  - **SkinnyProfiles.xml**

- **Profile Editor**:
  - **DefaultProfile**
  - **Enter Char**: [Field]
  - **SkinnyProfiles.xml**: [Field]

- **Buttons**:
  - **Start**
  - **Edit**

---

**GL Communications Inc.**

*Slide Number 8*
Skinny Auto and User Defined Traffic Profiles

Auto_Traffic Profile

User Defined Traffic Profile
Skinny RTP Core


GL Software RTP Application Is Licensed
GL Software RTP Application With Video Simulation Support Is Not Licensed
GL Software RTP Application With Voice Quality Support Is licenced
GL Software RTP Application With AMR-NB Codec Is licenced
GL Software RTP Application With AMR-WB Codec Is licenced
GL Software RTP Application With EURC Codec Is licenced
GL Software RTP Application With EURCB Codec Is licenced
GL Software RTP Application With EURC-C Codec Is licenced

RTP is attached to IP Address : 192.168.12.118
WAITING FOR SIP CONNECTION...on 192.168.12.118 port.. 30102.
Connected to SIP Core.
Skinny Call Generation

Inbound Call - Line Dir Number

Active Calls  Call Status  Call Events

Loading Scripts and Profiles

Message Sequence

Decode Message

GL Communications Inc.
Skinny Call Generation

Running Outbound and InBound Call Scripts

Loading Scripts and Profiles

Message Sequence

Decode Message

GL Communications Inc.
## Skinny Outbound and Inbound Call Event Log

### Events Table

<table>
<thead>
<tr>
<th>Date/Time</th>
<th>Captured Events</th>
<th>Call Trace Id</th>
<th>Script Name</th>
<th>Script Id</th>
</tr>
</thead>
<tbody>
<tr>
<td>2016-4-22 12:13:10.000</td>
<td>192.168.1.36 Media职Phone Address = 192.168.1.36</td>
<td>539</td>
<td>Phone_OutBoundCall.gls</td>
<td>CGProScript1_29.1723388535-5517.4724</td>
</tr>
<tr>
<td>2016-4-22 12:13:10.000</td>
<td>Media职Port = 1026</td>
<td>539</td>
<td>Phone_OutBoundCall.gls</td>
<td>CGProScript1_29.1723388535-5517.4724</td>
</tr>
<tr>
<td>2016-4-22 12:13:10.000</td>
<td>192.168.1.36 Media職Phone Address = 192.168.1.36</td>
<td>539</td>
<td>Phone_InBoundCall.gls</td>
<td>CGProScript1_29.1723388535-5517.4724</td>
</tr>
<tr>
<td>2016-4-22 12:13:10.000</td>
<td>Media職Port = 1026</td>
<td>539</td>
<td>Phone_InBoundCall.gls</td>
<td>CGProScript1_29.1723388535-5517.4724</td>
</tr>
<tr>
<td>2016-4-22 12:13:10.000</td>
<td>192.168.1.36 Media職Phone Address = 192.168.1.36</td>
<td>539</td>
<td>Phone_InBoundCall.gls</td>
<td>CGProScript1_29.1723388535-5517.4724</td>
</tr>
<tr>
<td>2016-4-22 12:13:10.000</td>
<td>Media職Port = 1026</td>
<td>539</td>
<td>Phone_InBoundCall.gls</td>
<td>CGProScript1_29.1723388535-5517.4724</td>
</tr>
<tr>
<td>2016-4-22 12:13:10.000</td>
<td>192.168.1.36 Media職Phone Address = 192.168.1.36</td>
<td>539</td>
<td>Phone_InBoundCall.gls</td>
<td>CGProScript1_29.1723388535-5517.4724</td>
</tr>
<tr>
<td>2016-4-22 12:13:10.000</td>
<td>Media職Port = 1026</td>
<td>539</td>
<td>Phone_InBoundCall.gls</td>
<td>CGProScript1_29.1723388535-5517.4724</td>
</tr>
<tr>
<td>2016-4-22 12:13:10.000</td>
<td>192.168.1.36 Media職Phone Address = 192.168.1.36</td>
<td>539</td>
<td>Phone_InBoundCall.gls</td>
<td>CGProScript1_29.1723388535-5517.4724</td>
</tr>
<tr>
<td>2016-4-22 12:13:10.000</td>
<td>Media職Port = 1026</td>
<td>539</td>
<td>Phone_InBoundCall.gls</td>
<td>CGProScript1_29.1723388535-5517.4724</td>
</tr>
<tr>
<td>2016-4-22 12:13:10.000</td>
<td>192.168.1.36 Media職Phone Address = 192.168.1.36</td>
<td>539</td>
<td>Phone_InBoundCall.gls</td>
<td>CGProScript1_29.1723388535-5517.4724</td>
</tr>
<tr>
<td>2016-4-22 12:13:10.000</td>
<td>Media職Port = 1026</td>
<td>539</td>
<td>Phone_InBoundCall.gls</td>
<td>CGProScript1_29.1723388535-5517.4724</td>
</tr>
<tr>
<td>2016-4-22 12:13:10.000</td>
<td>192.168.1.36 Media職Phone Address = 192.168.1.36</td>
<td>539</td>
<td>Phone_InBoundCall.gls</td>
<td>CGProScript1_29.1723388535-5517.4724</td>
</tr>
<tr>
<td>2016-4-22 12:13:10.000</td>
<td>Media職Port = 1026</td>
<td>539</td>
<td>Phone_InBoundCall.gls</td>
<td>CGProScript1_29.1723388535-5517.4724</td>
</tr>
</tbody>
</table>

- **Save Events**: Clear, Capture Events to File
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**

![Fixed Load Pattern](image)

**Ramp**

![Ramp Load Pattern](image)

**Uniform**

![Uniform Load Pattern](image)

**Normal**

![Normal Load Pattern](image)

**Saw-tooth**

![Saw-tooth Load Pattern](image)

**Step**

![Step Load Pattern](image)

---

**GL Communications Inc.**
Skinny Bulk Call Generation

<table>
<thead>
<tr>
<th>Sr No</th>
<th>Script Name</th>
<th>Profile</th>
<th>Call Info</th>
<th>Script Execution</th>
<th>Status</th>
<th>Events</th>
<th>Event</th>
<th>Result</th>
<th>Total Iterations</th>
<th>Completed Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Phone_InBoundCall</td>
<td>Profile0002</td>
<td>Start</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Phone_InBoundCall</td>
<td>Profile0003</td>
<td>Start</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Phone_InBoundCall</td>
<td>Profile0004</td>
<td>Start</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Phone_InBoundCall</td>
<td>Profile0005</td>
<td>Start</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Phone_InBoundCall</td>
<td>Profile0006</td>
<td>Start</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Phone_InBoundCall</td>
<td>Profile0007</td>
<td>Start</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Phone_InBoundCall</td>
<td>Profile0008</td>
<td>Start</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Phone_InBoundCall</td>
<td>Profile0009</td>
<td>Start</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Phone_InBoundCall</td>
<td>Profile0010</td>
<td>Start</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Phone_InBoundCall</td>
<td>Profile0011</td>
<td>Start</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

// The below function is used to create the tcp connection with Call Manager

```
CallDurationTimes = "Idle``;";
```
Skinny Global Configuration

<table>
<thead>
<tr>
<th>Config</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Configuration</td>
<td></td>
</tr>
<tr>
<td>Call Parameters</td>
<td></td>
</tr>
<tr>
<td>Call Duration in msec</td>
<td>30000</td>
</tr>
<tr>
<td>Call Answer Time in msec</td>
<td>100</td>
</tr>
<tr>
<td>Auto Parameters</td>
<td></td>
</tr>
<tr>
<td>Auto Answer</td>
<td>1</td>
</tr>
<tr>
<td>Media Port</td>
<td>1024</td>
</tr>
<tr>
<td>Session Creation Timer</td>
<td>5000</td>
</tr>
<tr>
<td>Calculate Rtp Statistics</td>
<td>Enable</td>
</tr>
</tbody>
</table>

[Image of the Global Configuration window with settings]
Skinny Call Ratio Statistics

Call Graph

Call Stats

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 ("SoftKeyEventMessage.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.
Call Stats provide a running tabular log of system level stats, tracked stats include: Total Calls, Active Calls, Completed Calls, Passed Calls, Failed Calls, Instantaneous Calls/Sec
• 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.
• 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.
Thank You