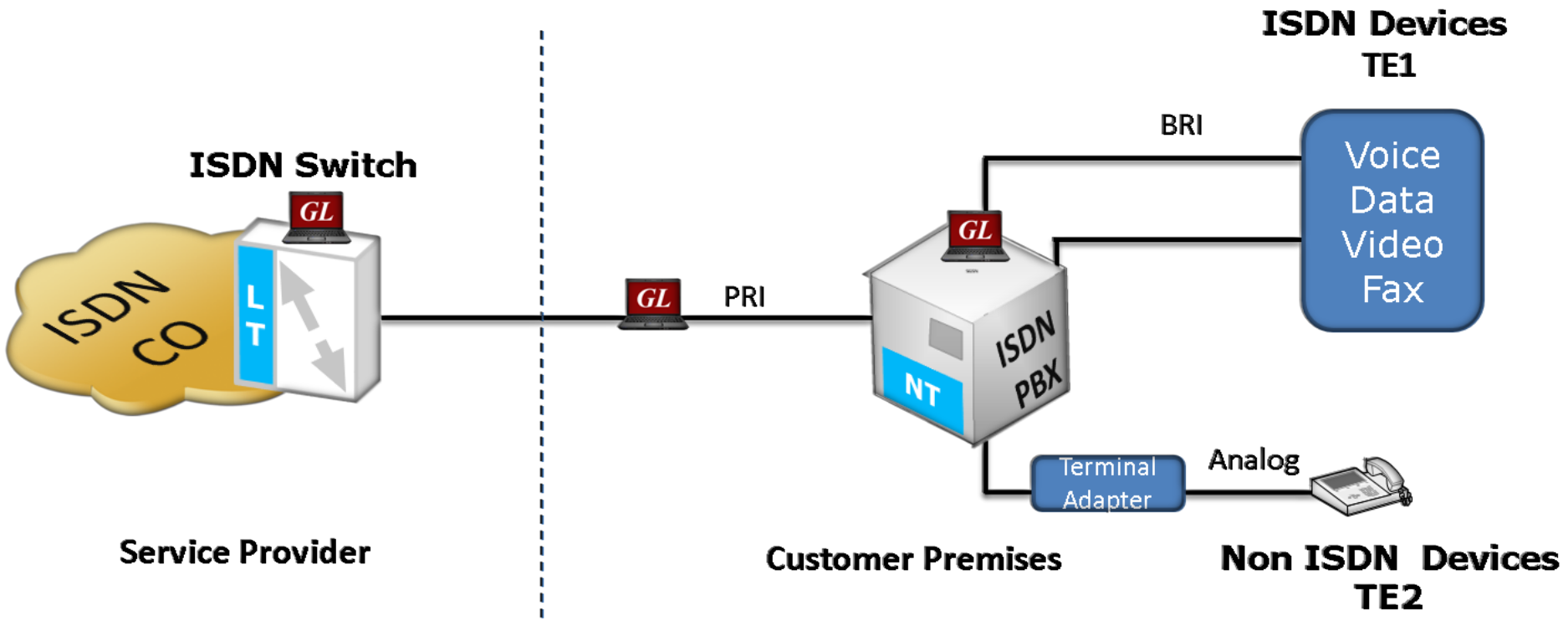
MAPS™ ISDN Protocol Emulator

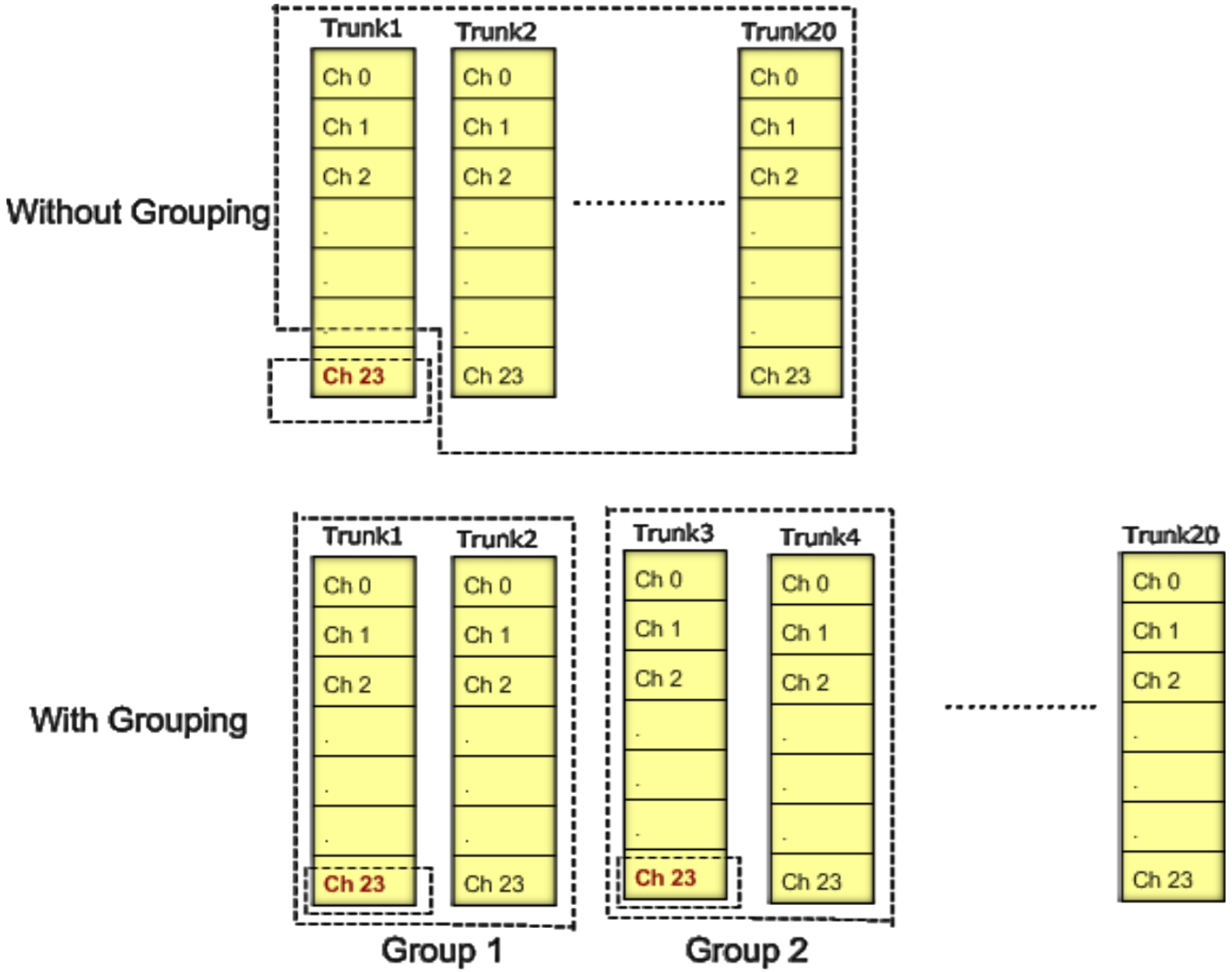
GL Communications Inc.
818 West Diamond Avenue - Third Floor, Gaithersburg, MD 20878
Phone: (301) 670-4784 Fax: (301) 670-9187 Email: info@gl.com
Website: https://www.gl.com

Network Architecture

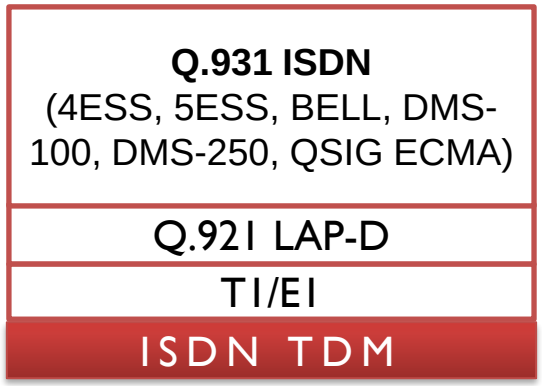
ISDN Analysis and Simulation
Simulate and Monitor Elements in ISDN Network over TDM



NFAS Grouping



Supported Standards and Protocols

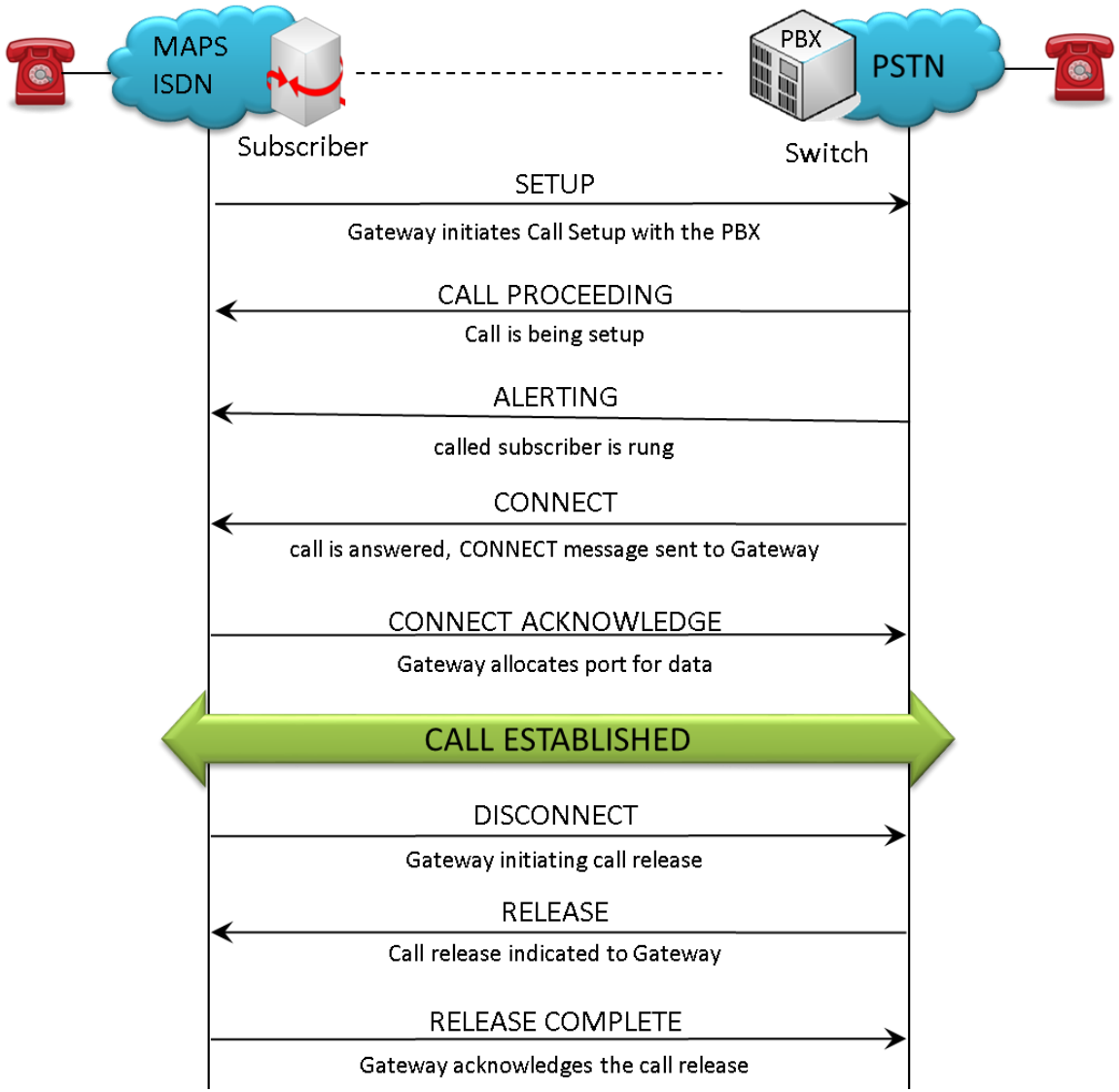


Supported Protocols	Standard / Specification Used
Q.921 (LAPD)	ITU-T Q.921
SR-4994	National ISDN PRI Standard
Q.931	ITU-T Q.931 / Q.932(Facility IE) / Q.955.3 (MLPP Procedures)
4ESS	ISDN PRI (TR-41449)
5ESS	ISDN PRI (Lucent Tech - 5ESS 2000)
BELL	ISDN PRI (Bell Core SR-NWT-002343)
DMS-100	Nortel's Switch DMS 100 NIS-A2111-1
DMS-250	Nortel's Switch DMS 250 NIS-A2111-4
QSIG ECMA	Standard ECMA-143 4th Edition - December 2001

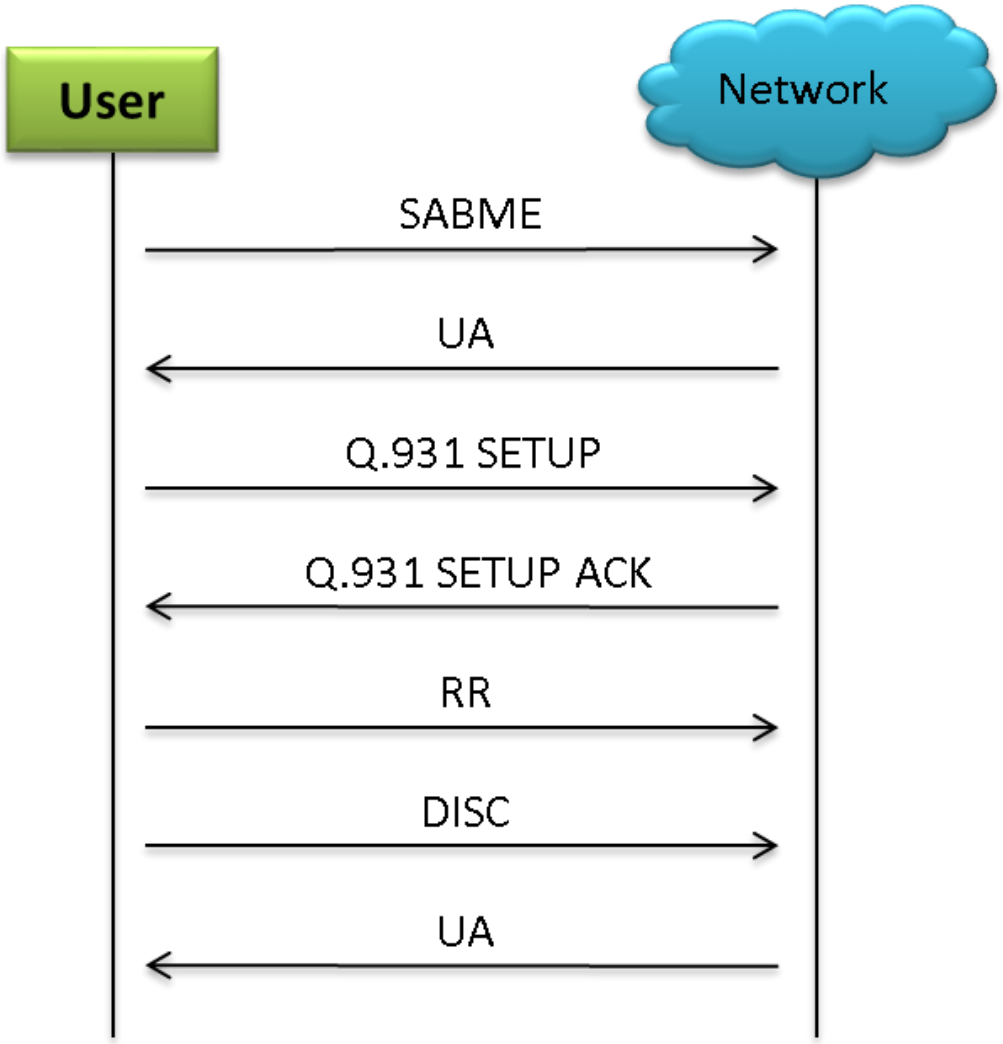
Main Features

- ISDN simulation over TDM (T1/E1)
- Multiple T1/E1 line interfaces supported
- Access to all ISDN Message Parameters such as Call Reference Value, Called Number, Calling Number, Release Cause, and more
- Access to LAPD SABME, UA, RR, DISC signaling messages at layer 2
- Switch and Subscriber Emulation
- Provides various release cause codes such as rejected, no user response, user busy, congested, and so on to troubleshoot the problems in ISDN
- Overlap sending of ISDN messages
- Supports NFAS testing for T1 only

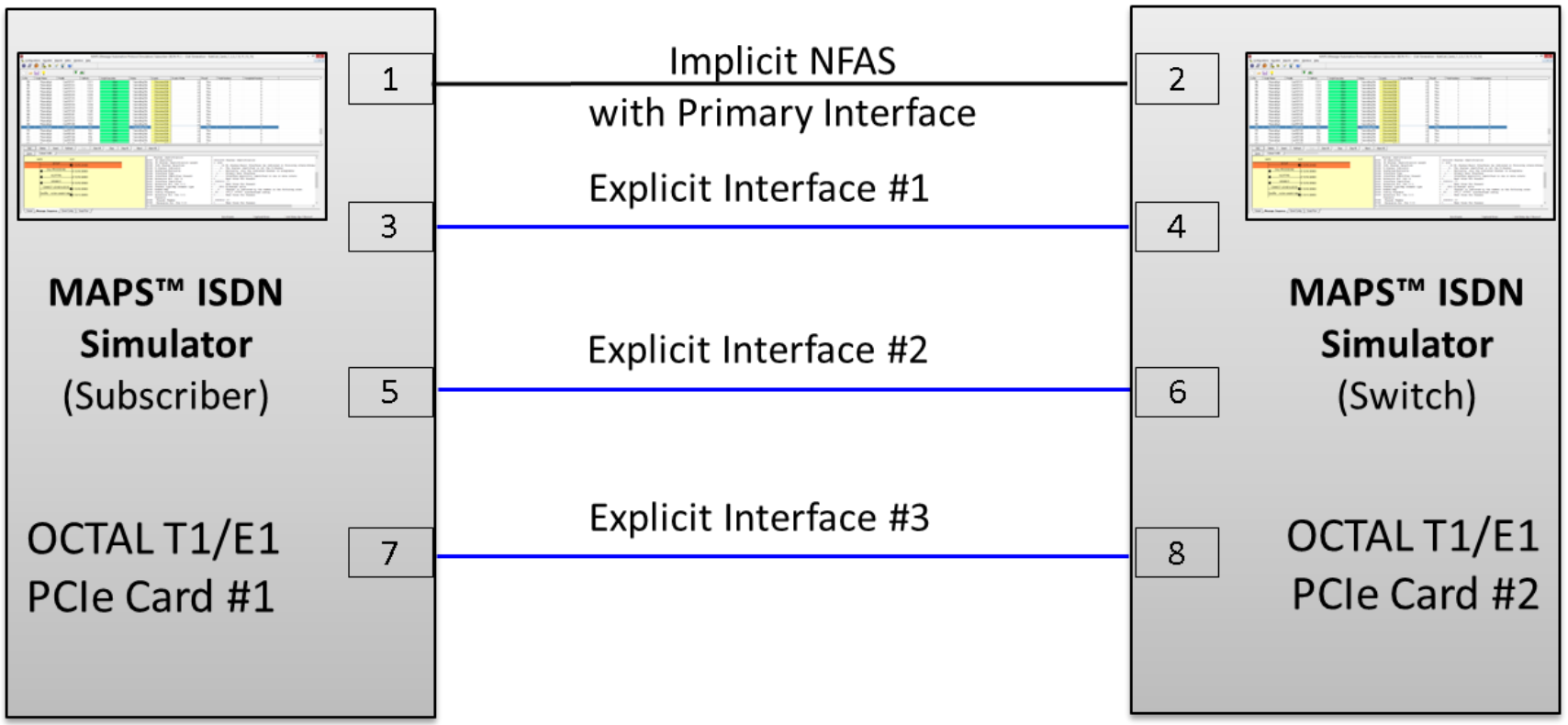
Typical ISDN Call Flow



Typical LAPD Call Flow

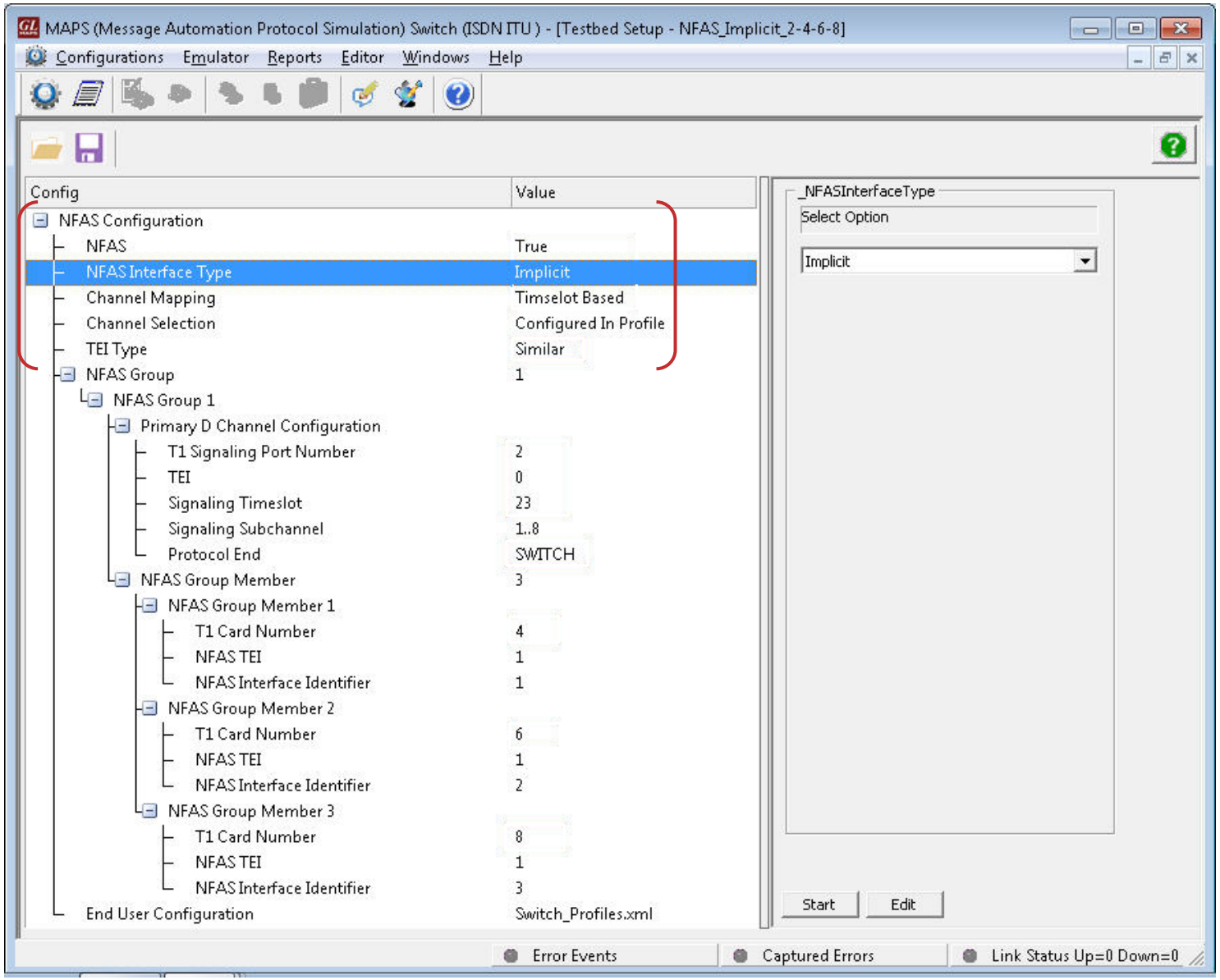


ISDN Calls with NFAS Option



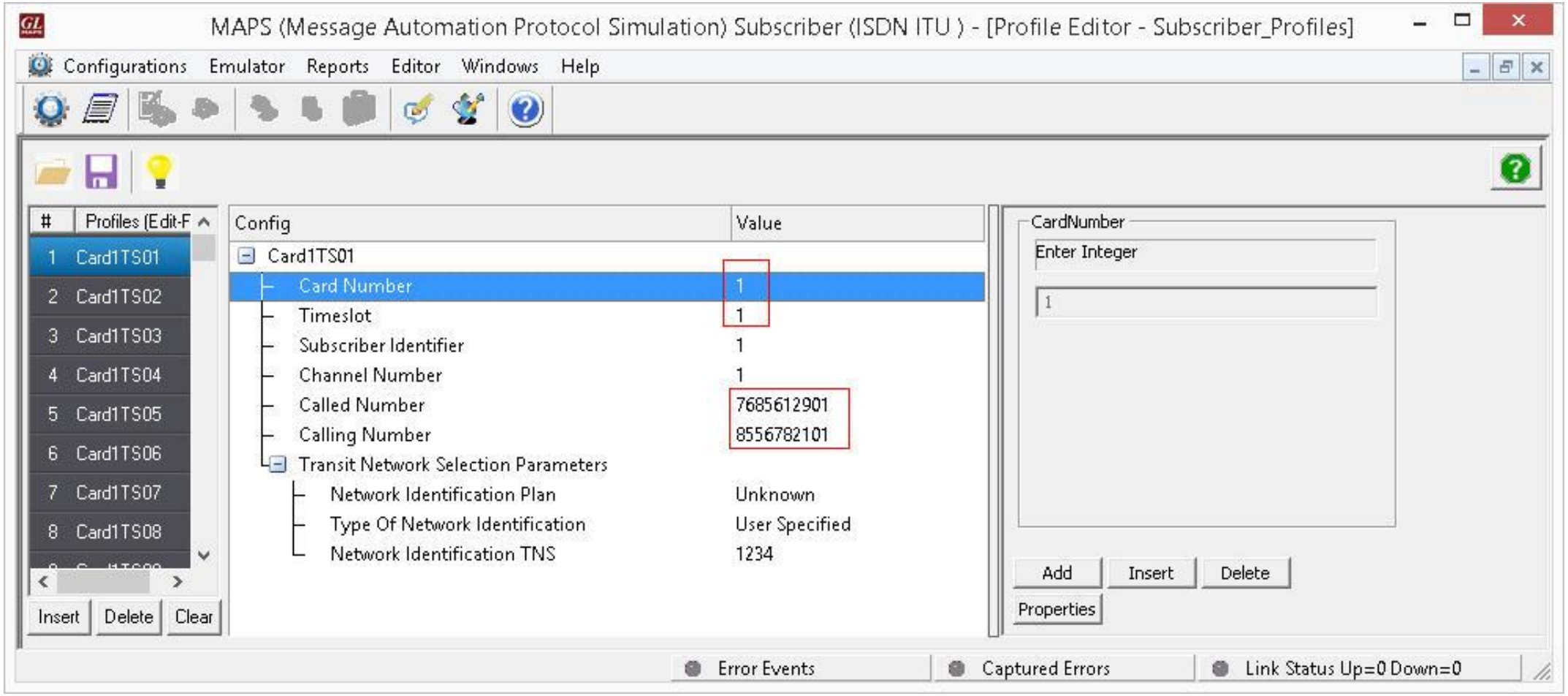
GL Communications

ISDN TestBed Setup Configuration



GL Communications

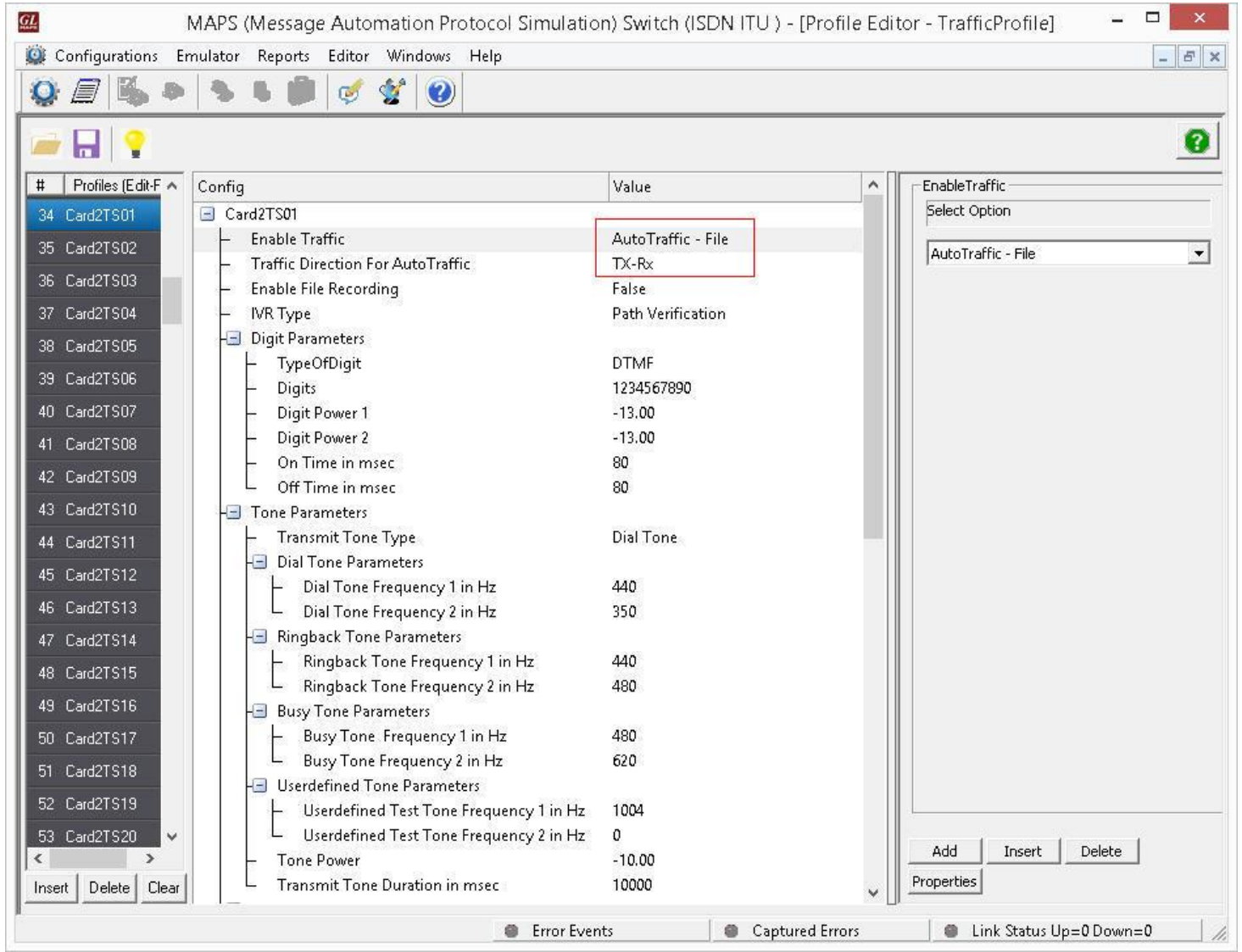
ISDN Subscriber Profile Configuration



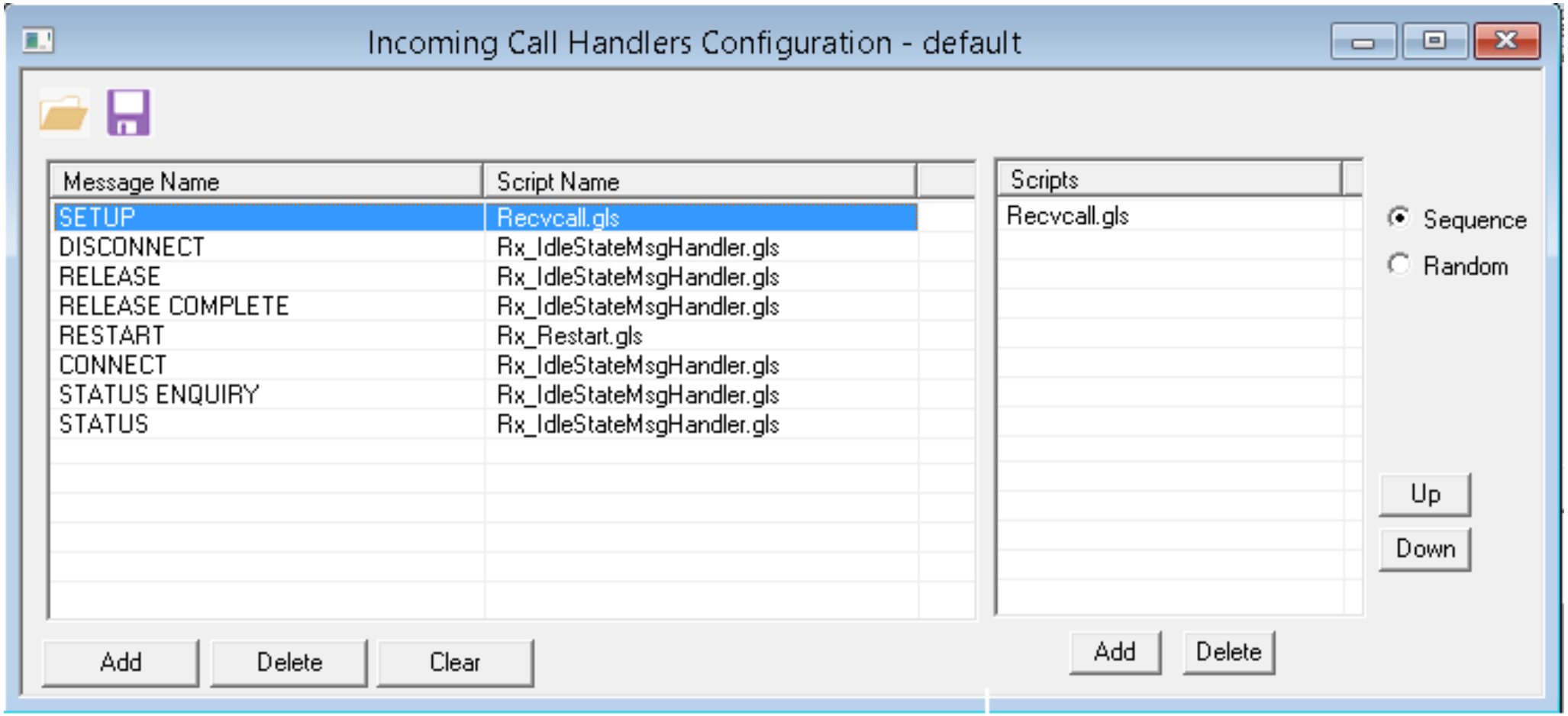
Analog and TDM Traffic Simulation

- Transmission, detection of TDM and Analog traffic - digits, voice files, single tones, dual tones, Dynamic VF, FAX, and IVR over established calls
- The volume of calls can vary from one to hundreds of calls depending on the T1 E1 or Analog platform of choice
- All variations of Fax traffic supported over 2 wire analog and T1 E1, such as page size, resolution, min and max data rate, and codec type – including high speed fax such as V.34

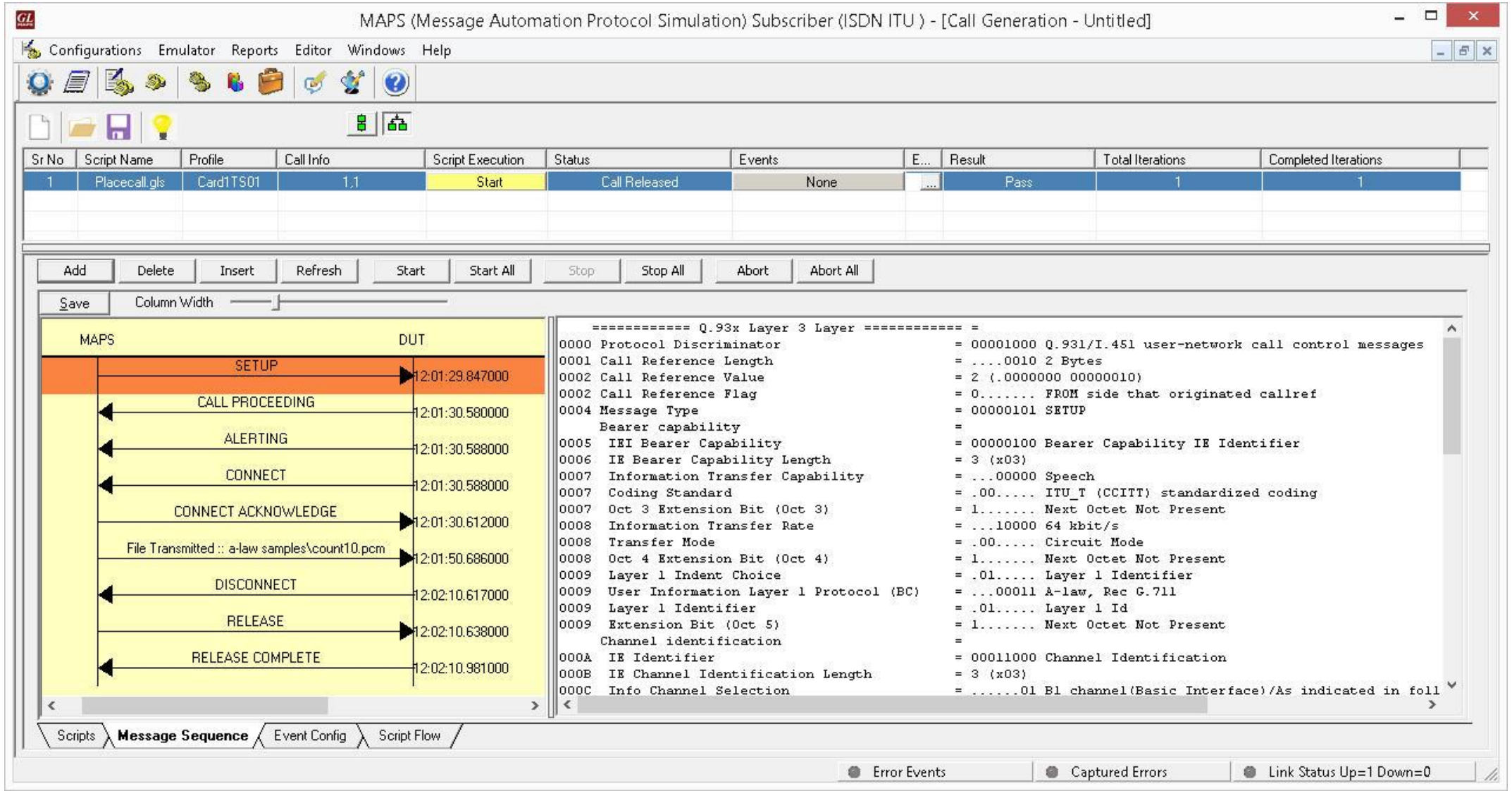
ISDN Traffic Profile Configuration



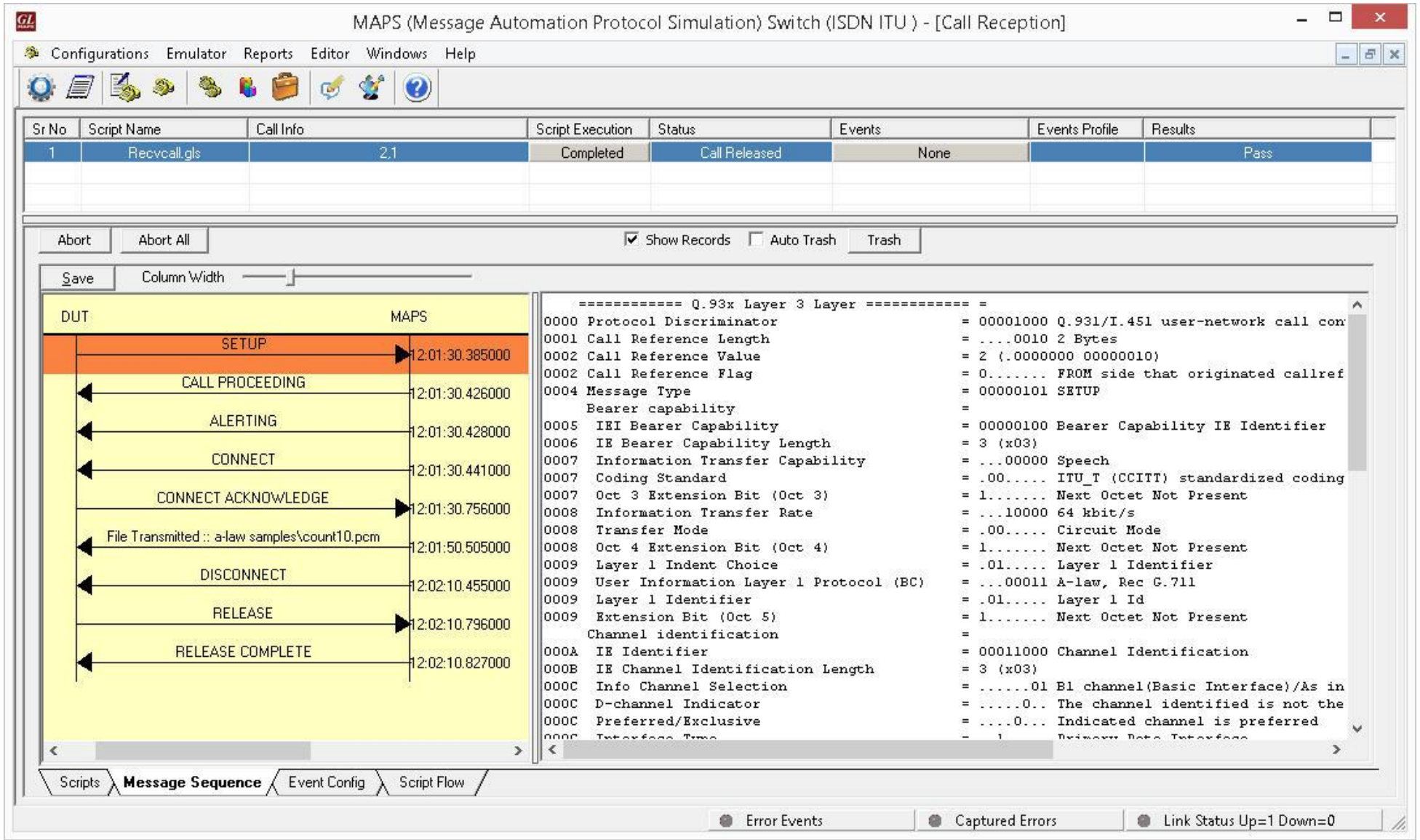
ISDN Incoming Call Handler Configuration



ISDN Call Generation

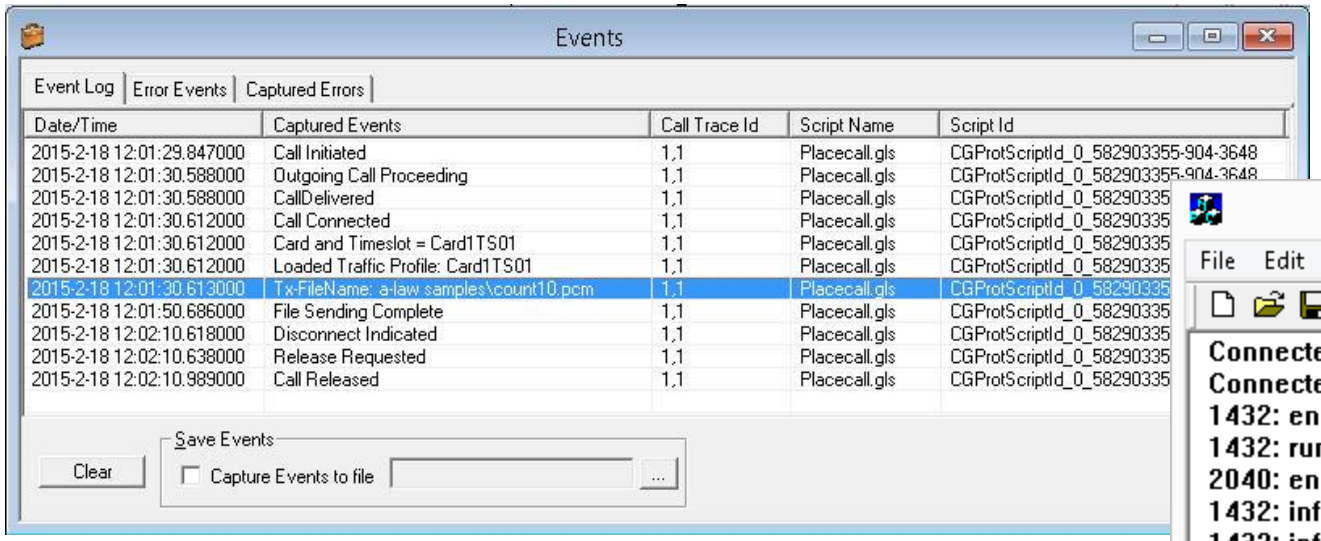


ISDN Call Reception

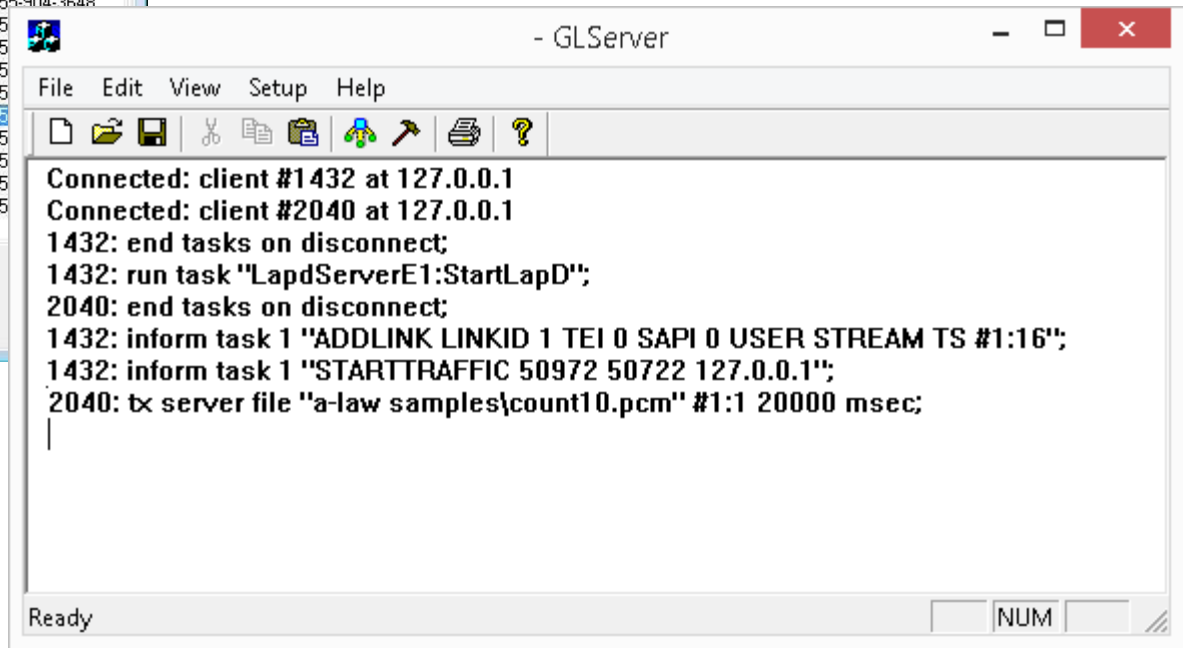


ISDN Events and Server Traffic Log

Events Log



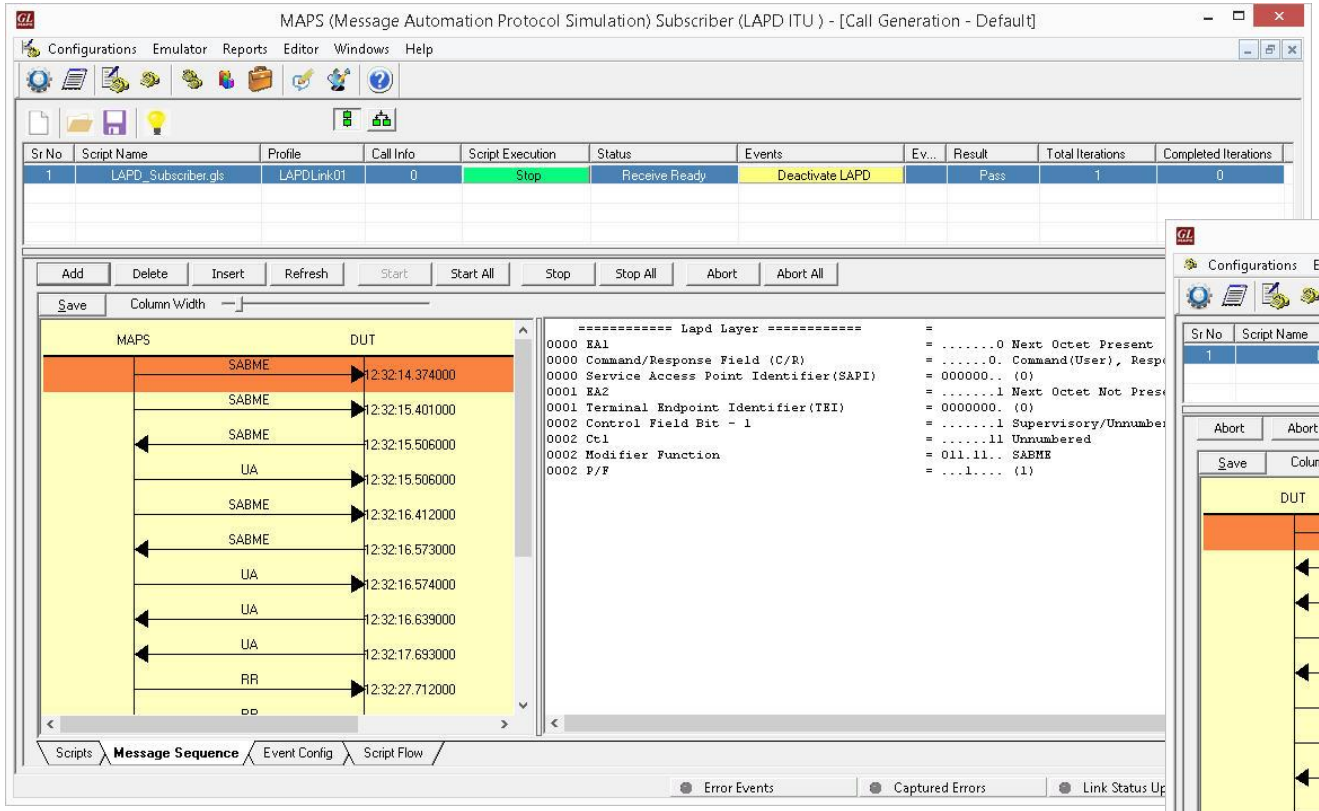
WCS Server Traffic Log



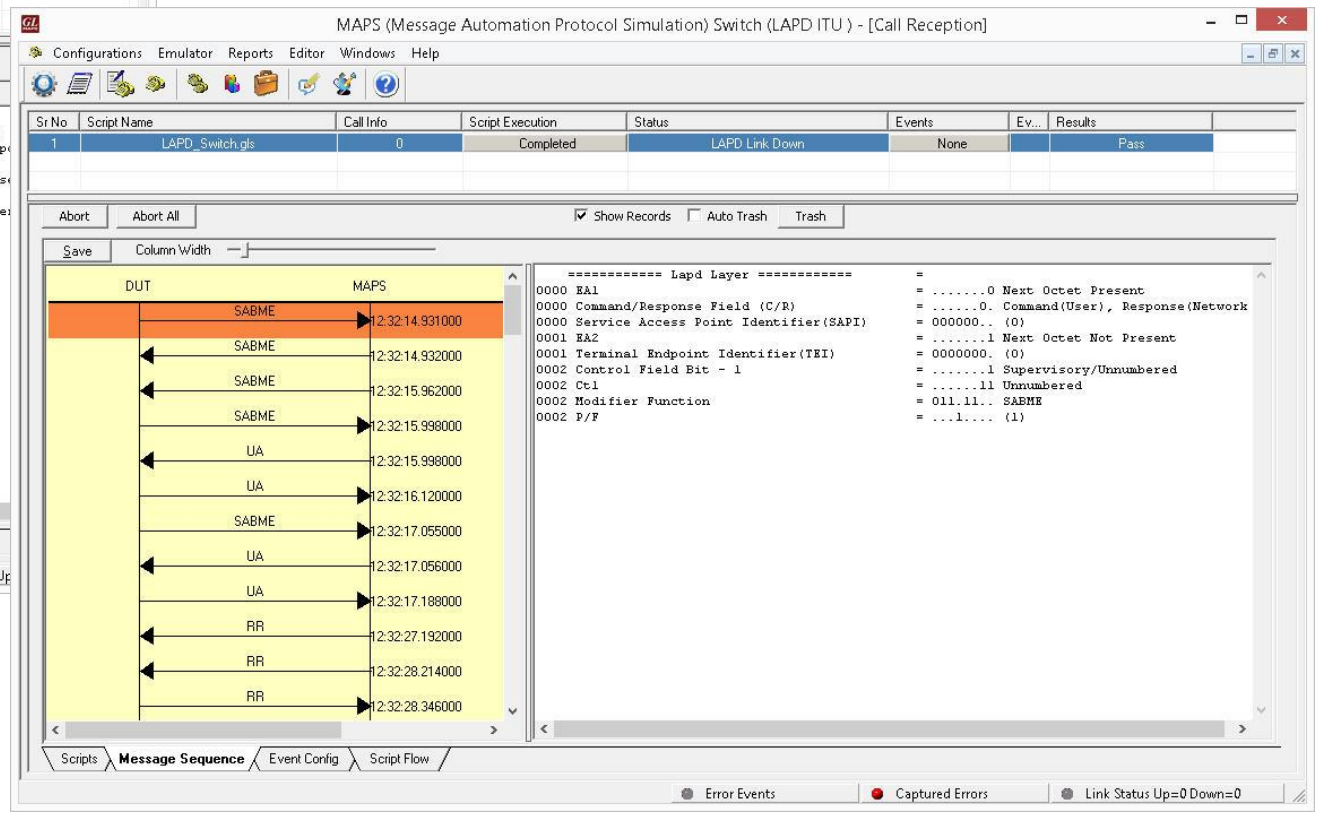
GL Communications

LAPD Call Simulation

Call Generation



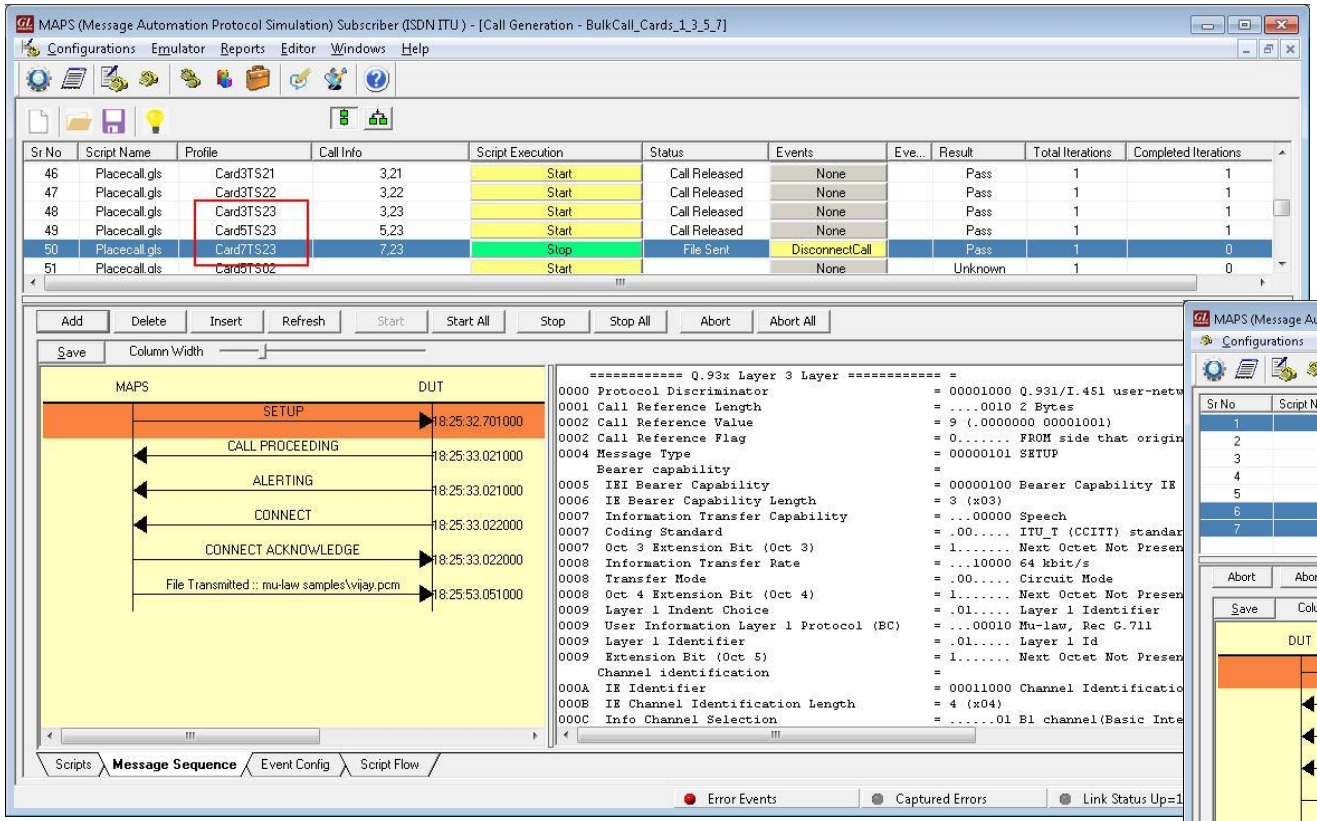
Call Reception



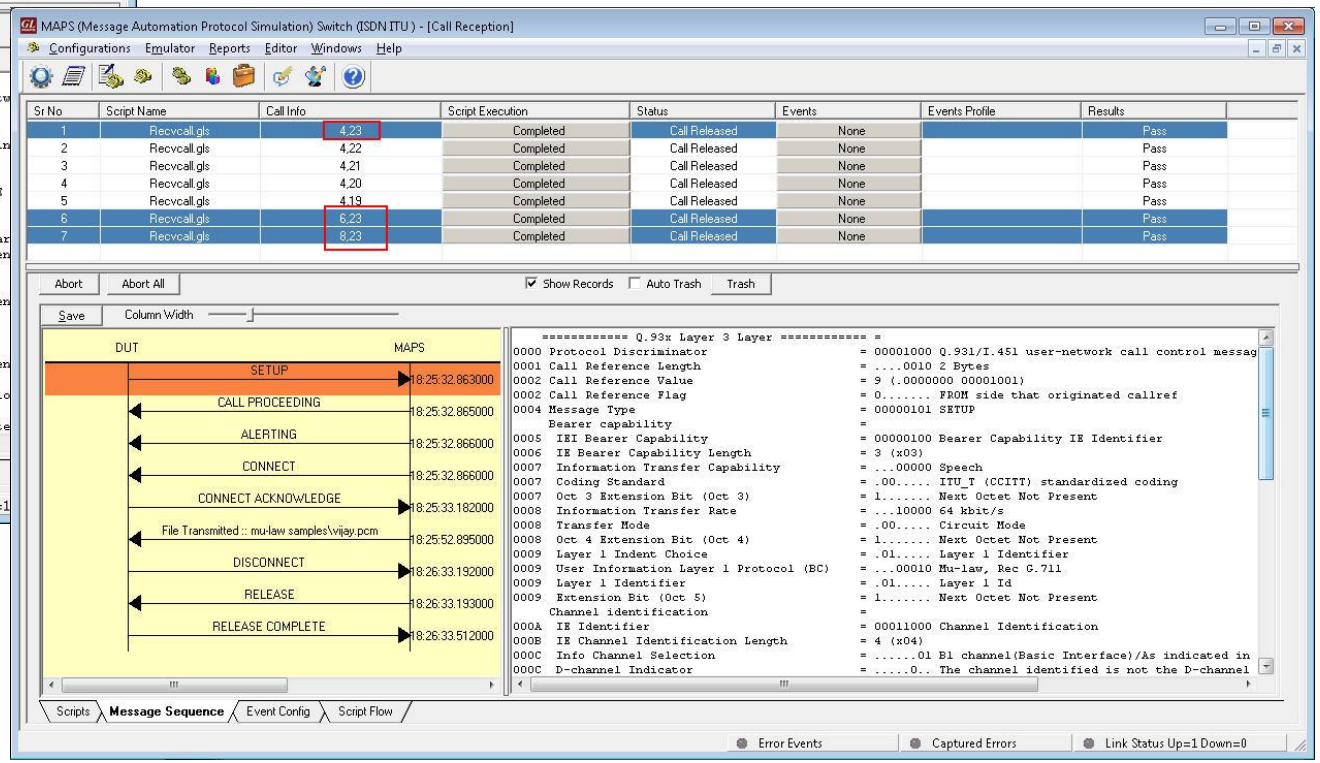
GL Communications

ISDN NFAS Call Simulation

Call Generation



Call Reception

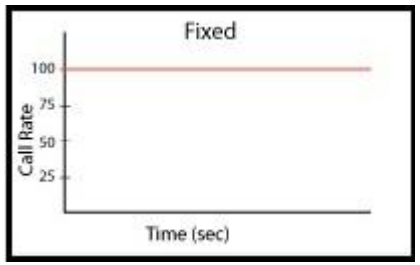


GL Communications

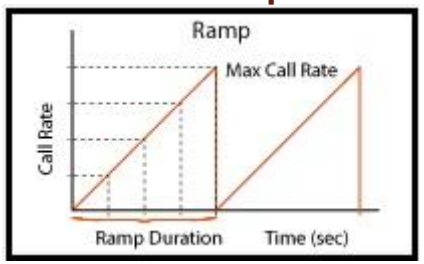
Load Generation

- 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.

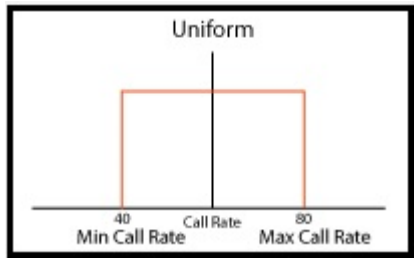
Fixed



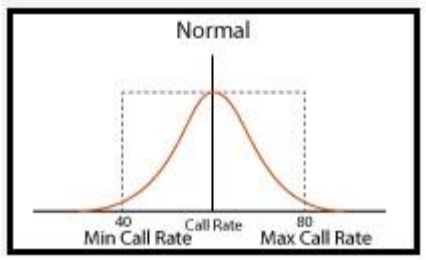
Ramp



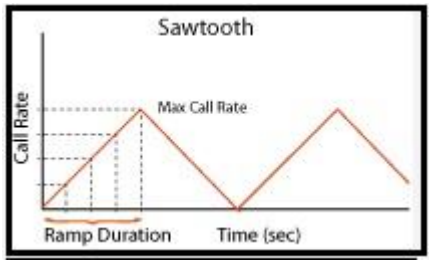
Uniform



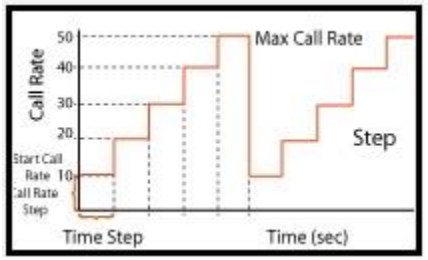
Normal

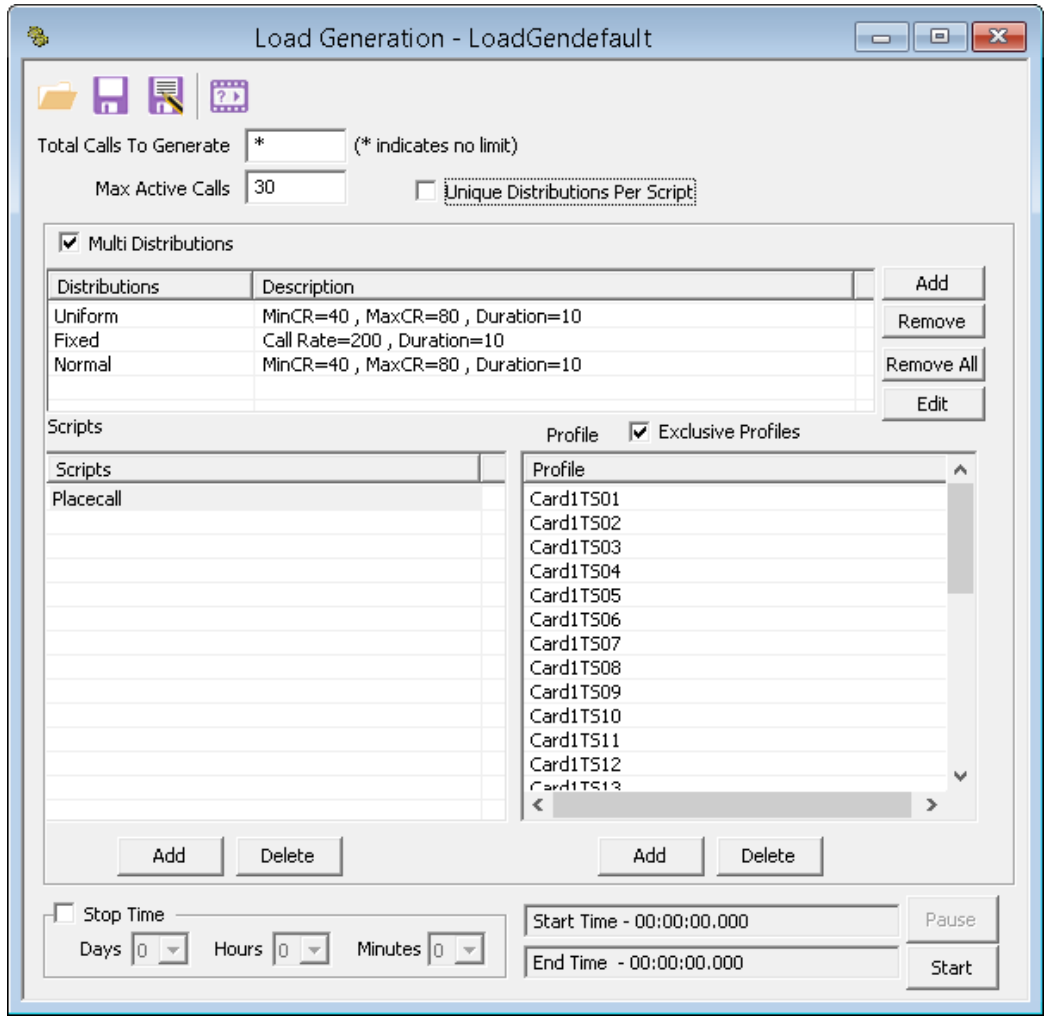


Saw-tooth



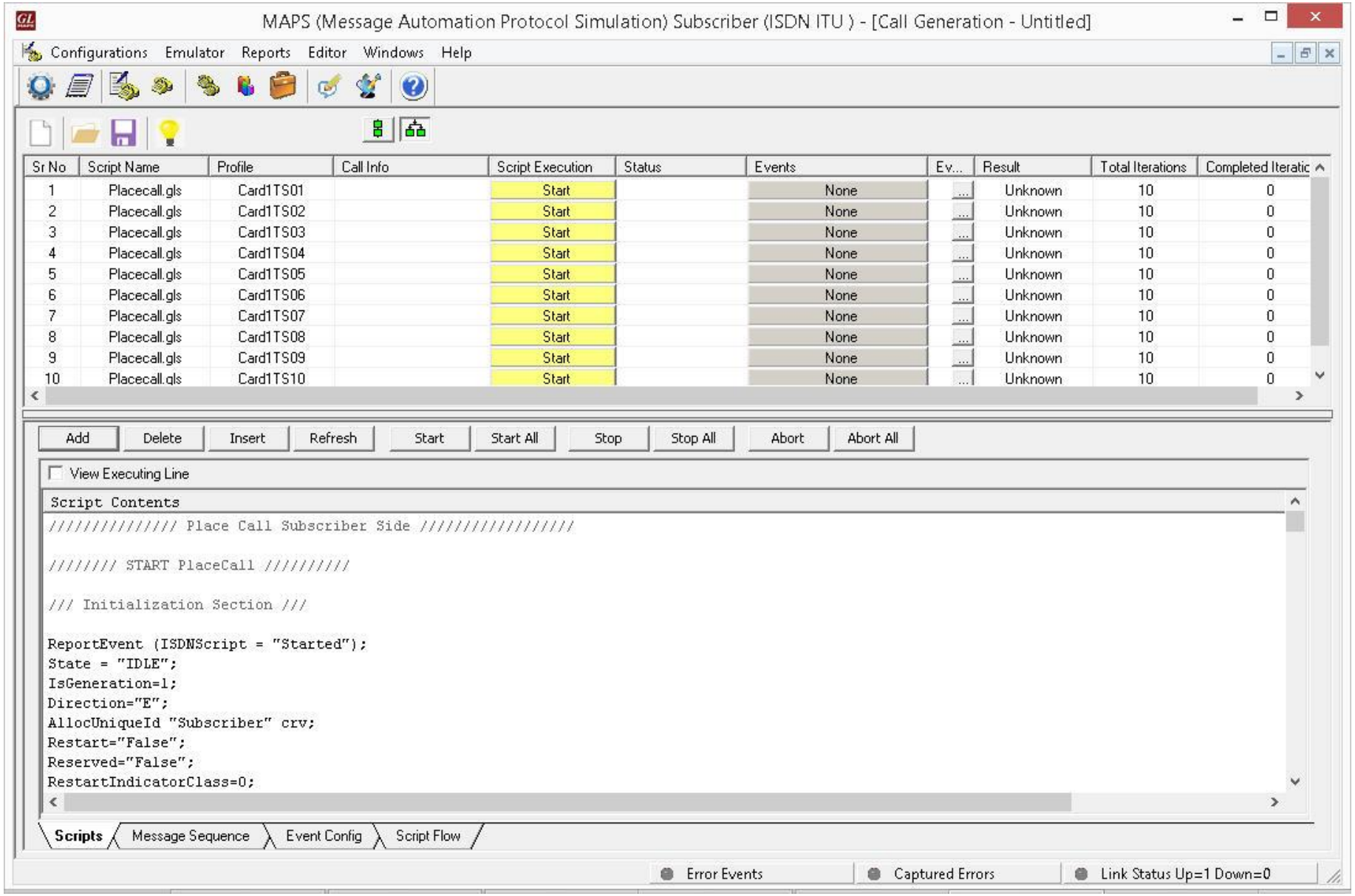
Step





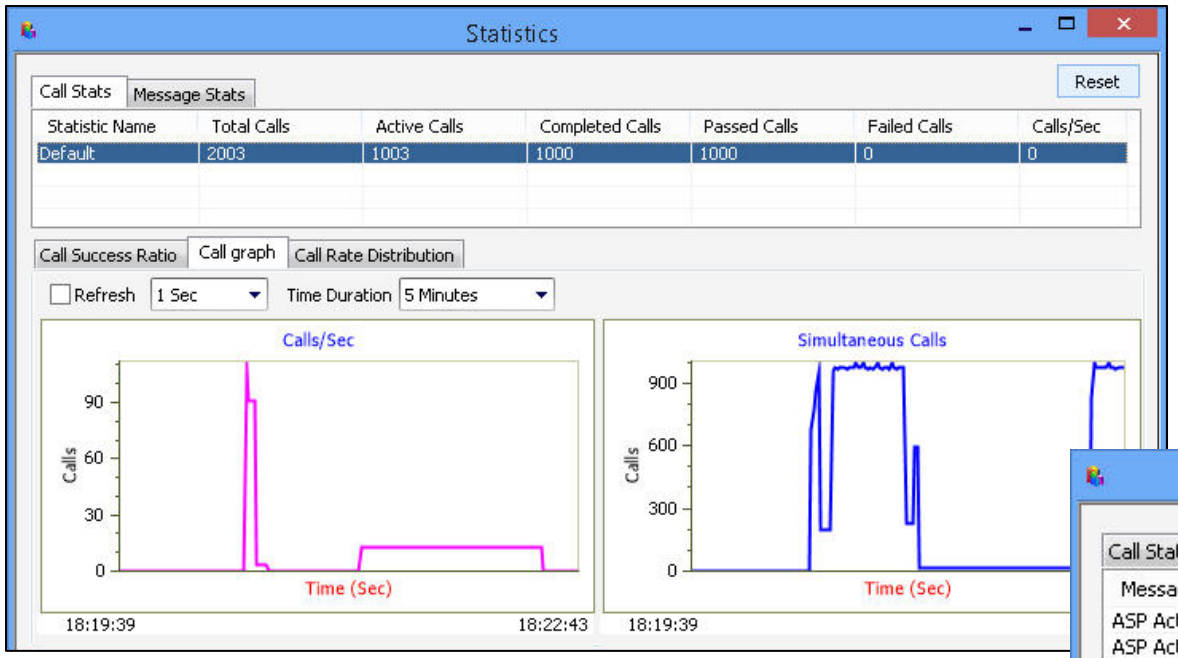
GL Communications

Bulk Call Generation

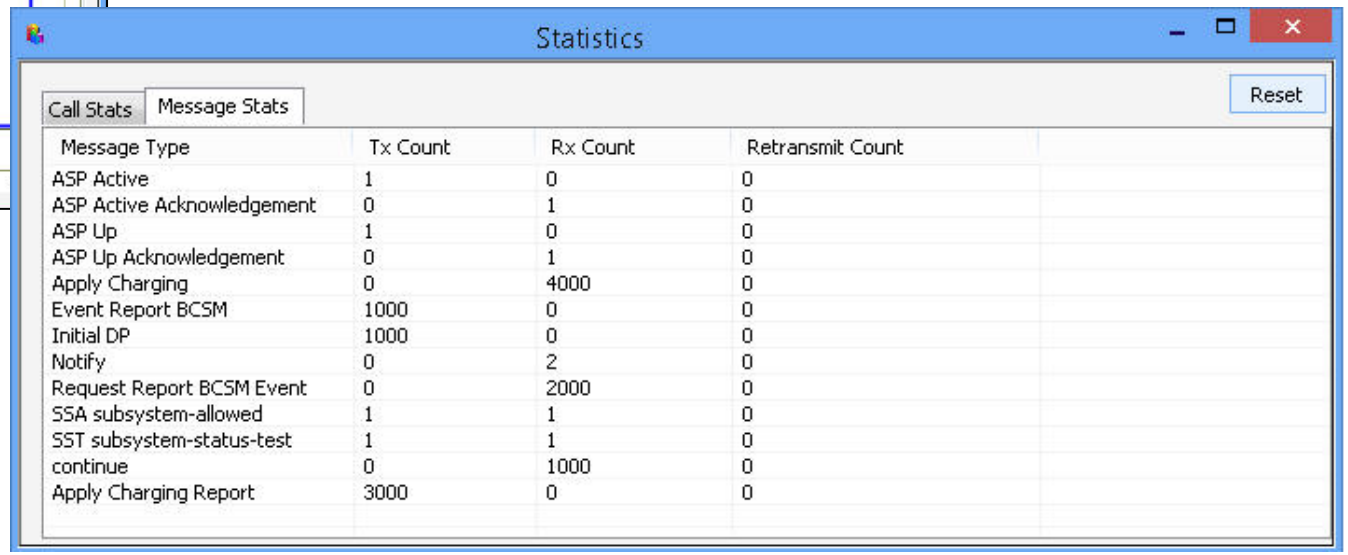


Bulk Call Statistics and Graph

Call Stats and Graph



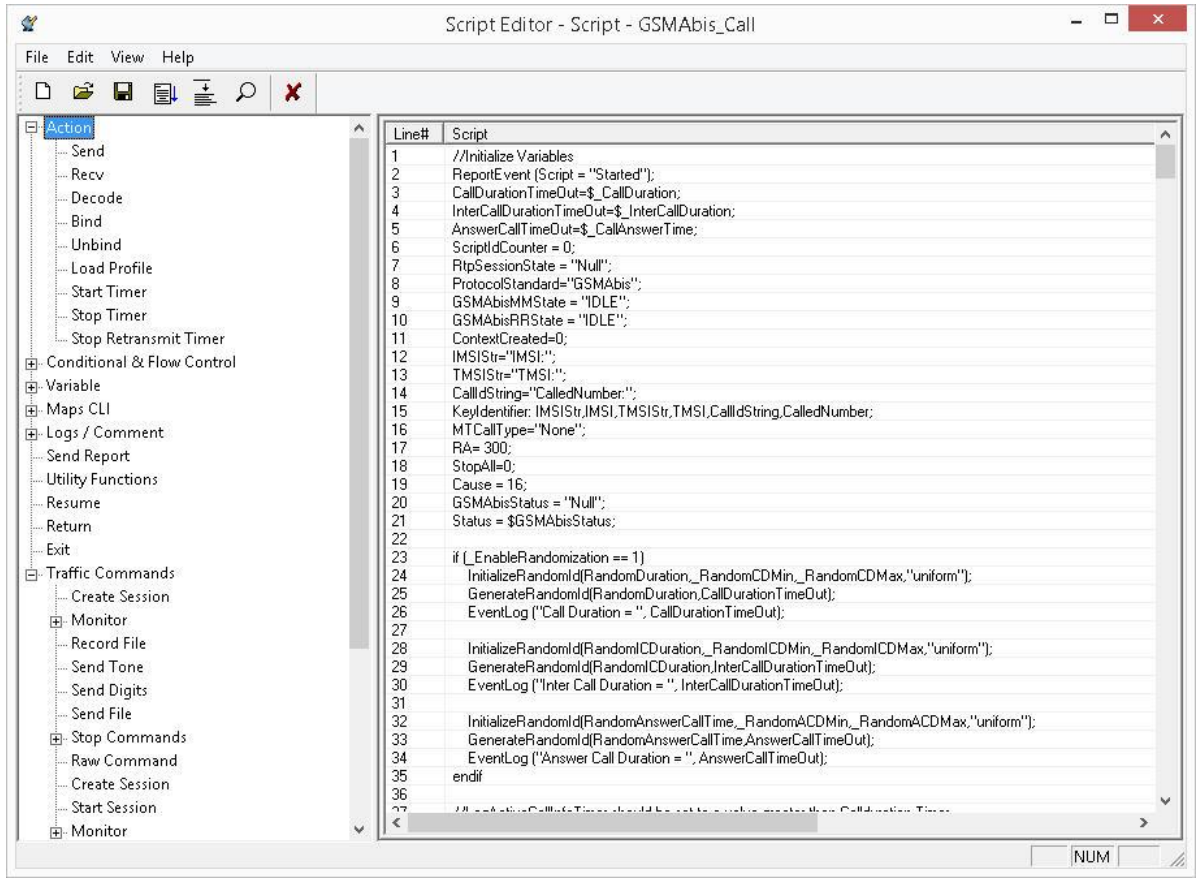
Message Stats



Message Type	Tx Count	Rx Count	Retransmit Count
ASP Active	1	0	0
ASP Active Acknowledgement	0	1	0
ASP Up	1	0	0
ASP Up Acknowledgement	0	1	0
Apply Charging	0	4000	0
Event Report BCSM	1000	0	0
Initial DP	1000	0	0
Notify	0	2	0
Request Report BCSM Event	0	2000	0
SSA subsystem-allowed	1	1	0
SST subsystem-status-test	1	1	0
continue	0	1000	0
Apply Charging Report	3000	0	0

Customizations - Call Flow (Scripts)

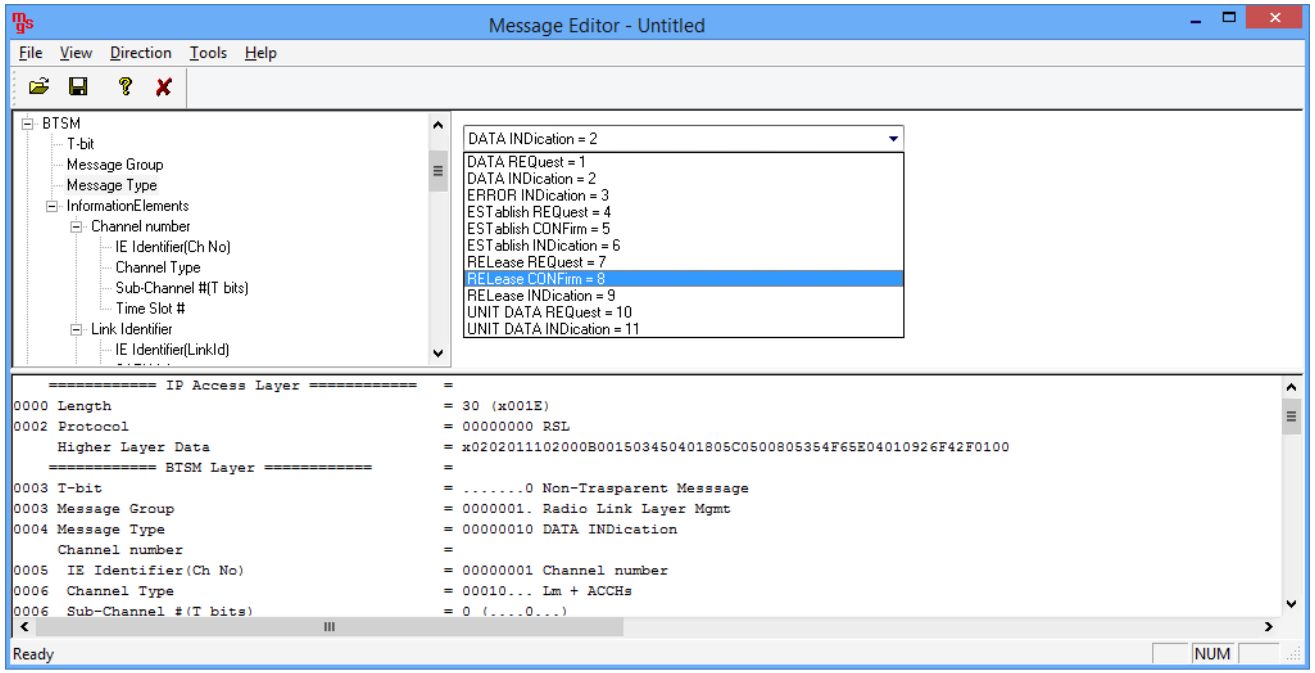
- Scripts are written in our proprietary *.gls scripting language. They represent generic state machines intended provide protocol/signaling logic for a call and establish bearer traffic
- Each instance of a script corresponds to a single transaction/call, i.e., if you place 500 calls in parallel you 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



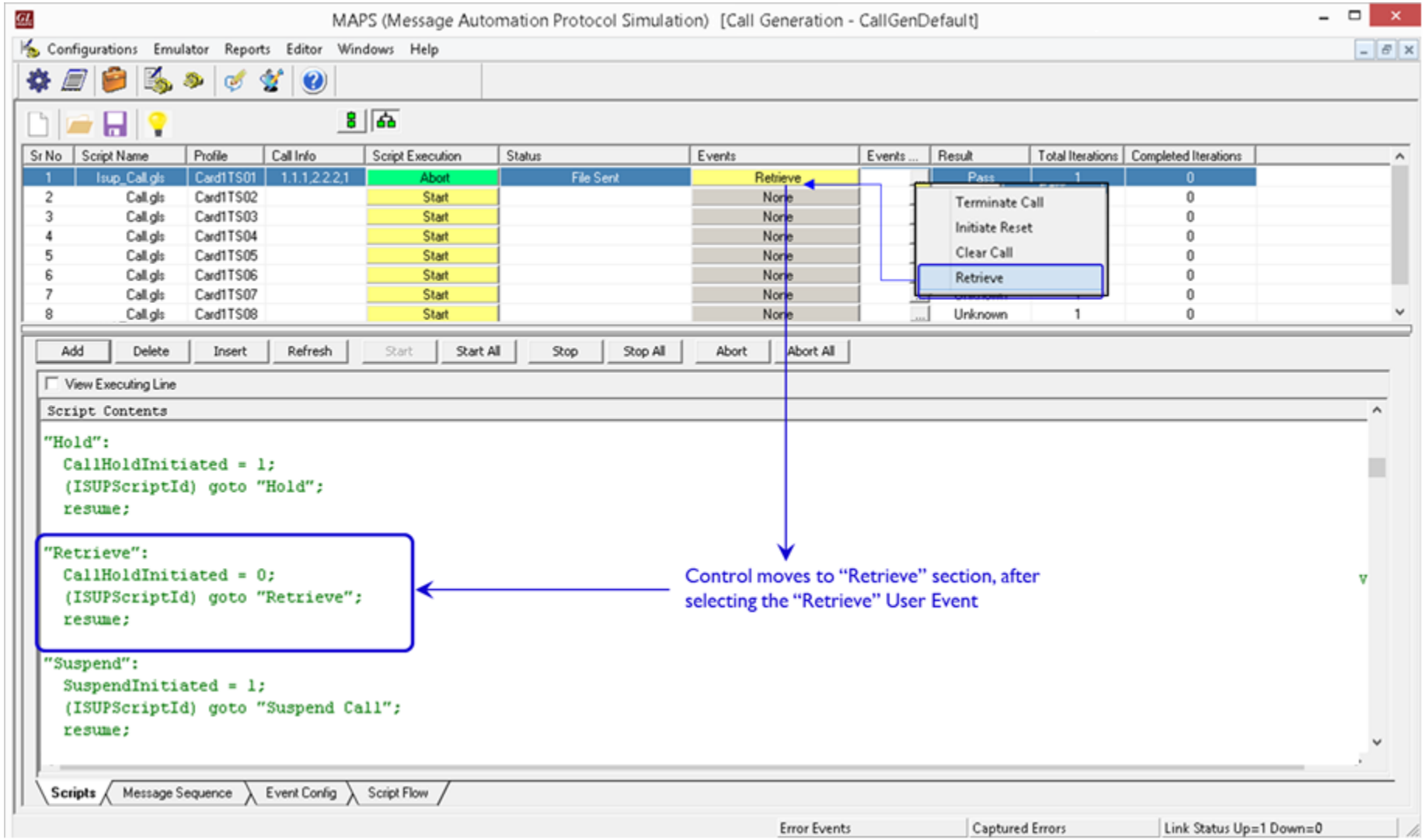
GL Communications

Customizations - Protocol Messages

- When the script actually sends a message it does so by loading a hdl file template from disk (“SGsAP-PAGING-REQUEST.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



Customizations - User Events



Customizations - Statistics and Reports

MOS, R-Factor

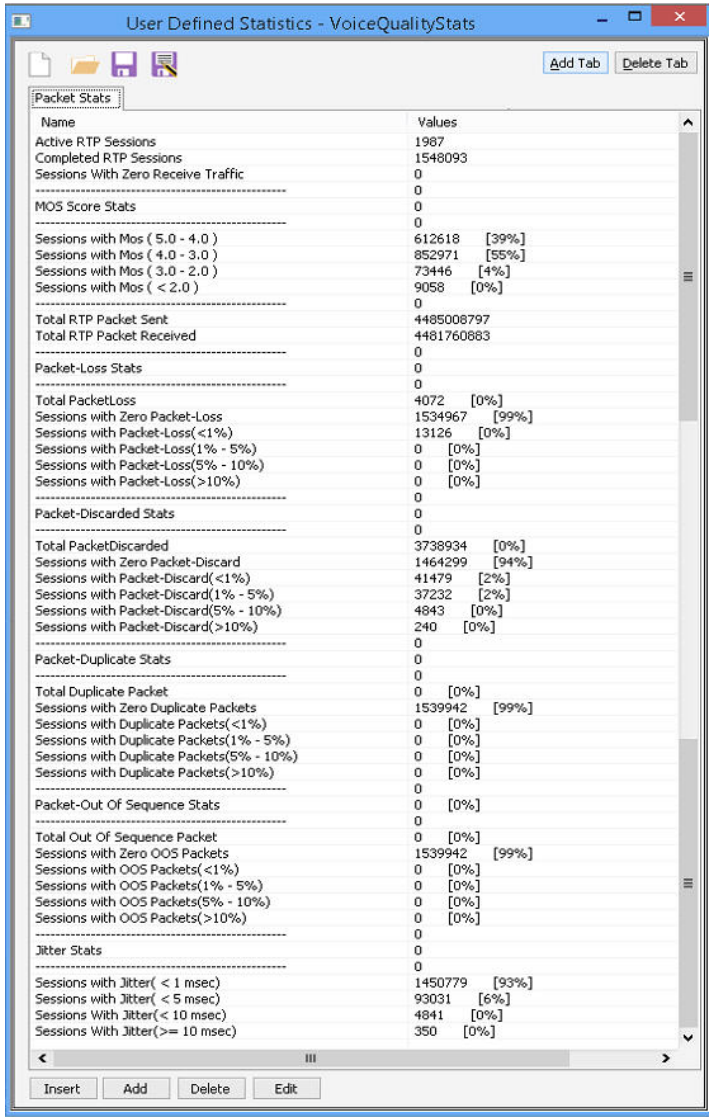
Packet Loss

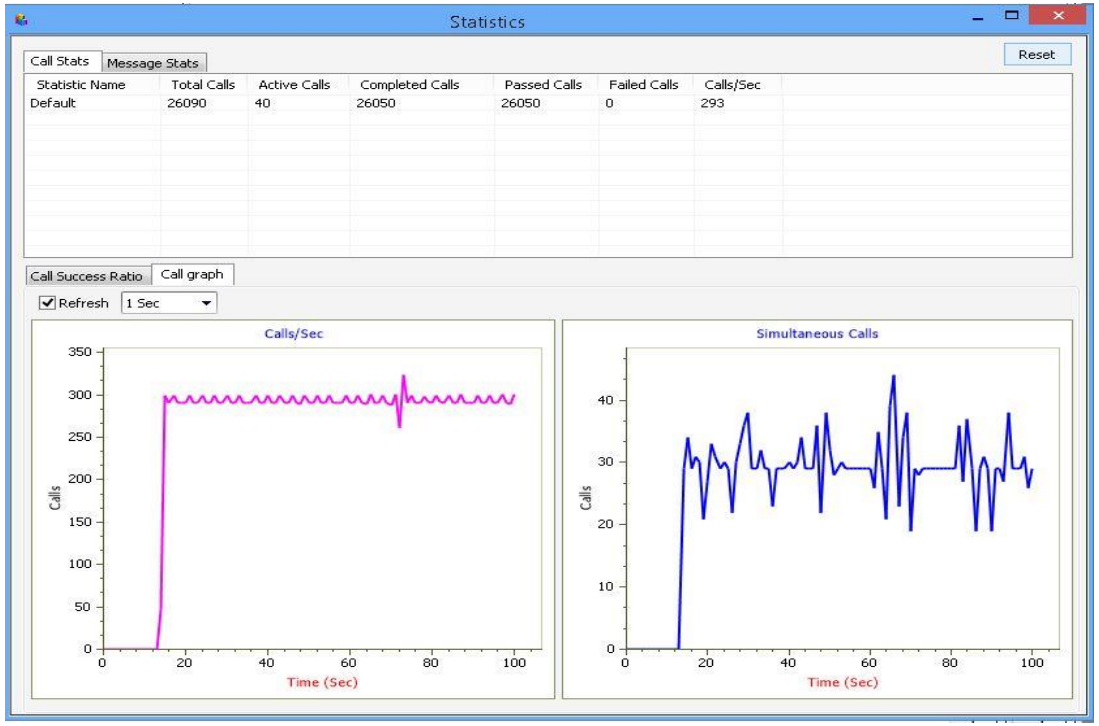
Packets Discarded

Duplicate Packets

Out-Of-Sequence Packets

Jitter Statistics

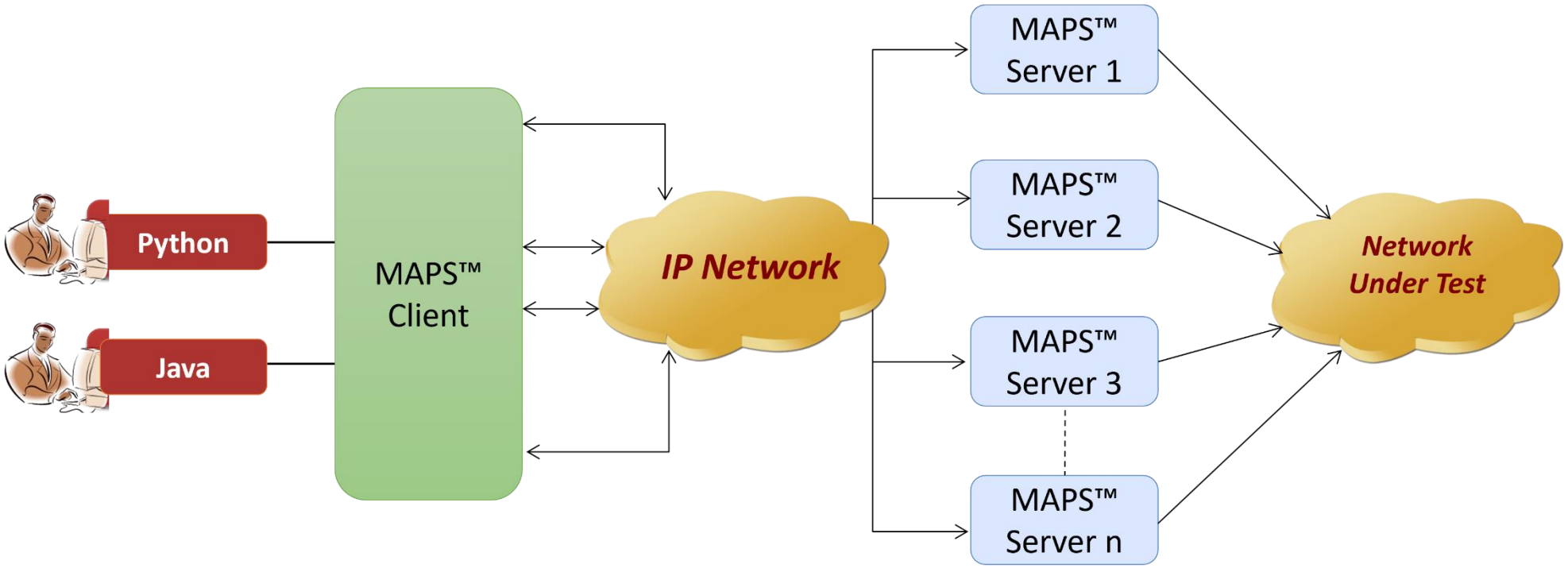




Call Stats provide a running tabular log of system level stats, tracked stats include: Total Calls, Active Calls, Completed Calls, Passed Calls, Failed Calls, Instantaneous Calls/Sec

GL Communications

MAPS™ API Architecture

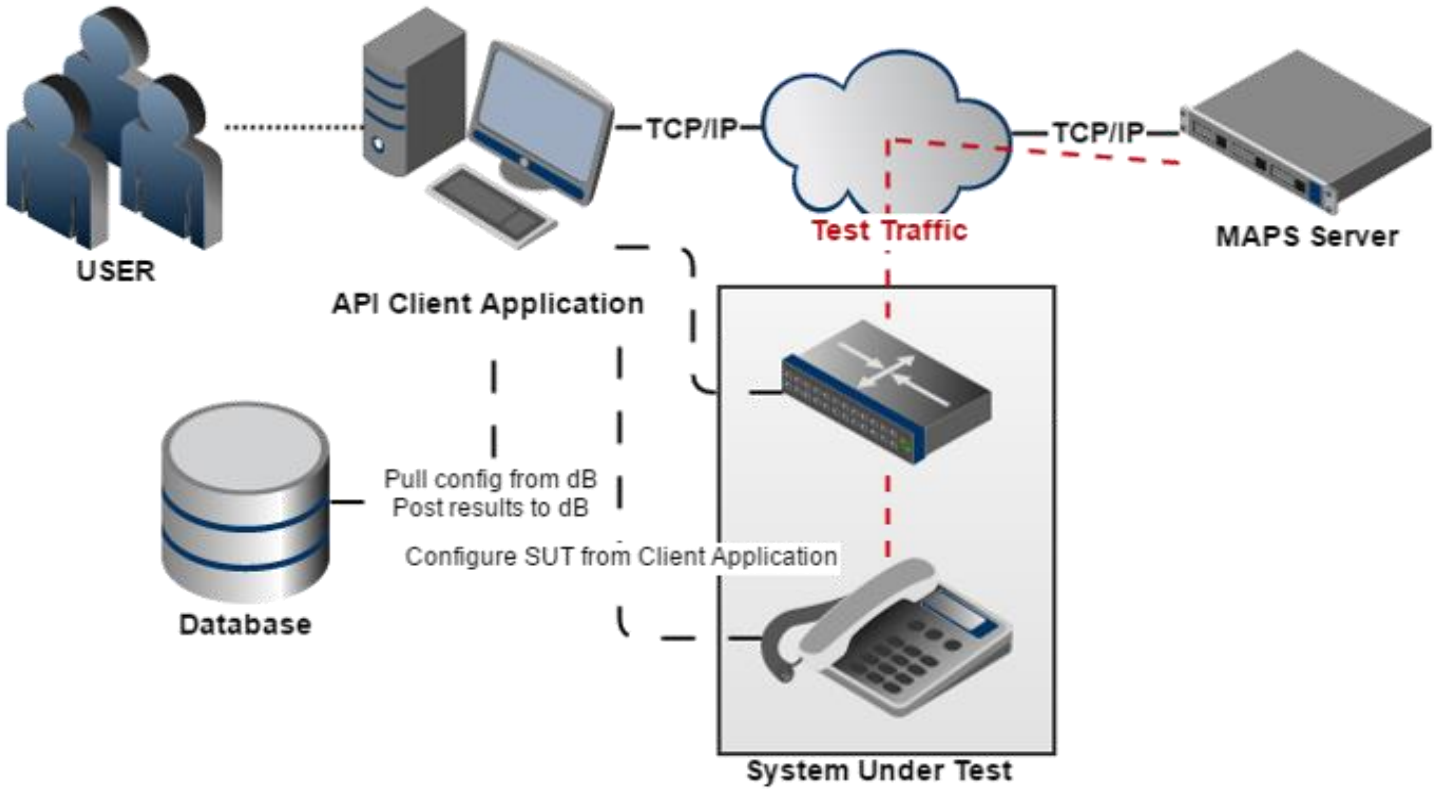


- API wraps our proprietary scripting language in standard languages familiar to the user:
  - Python
  - Java
- Clients and Servers support a “Many-to-Many” relationship, making it very easy for users to develop complex test cases involving multiple signaling protocols

API Architecture

System Integration

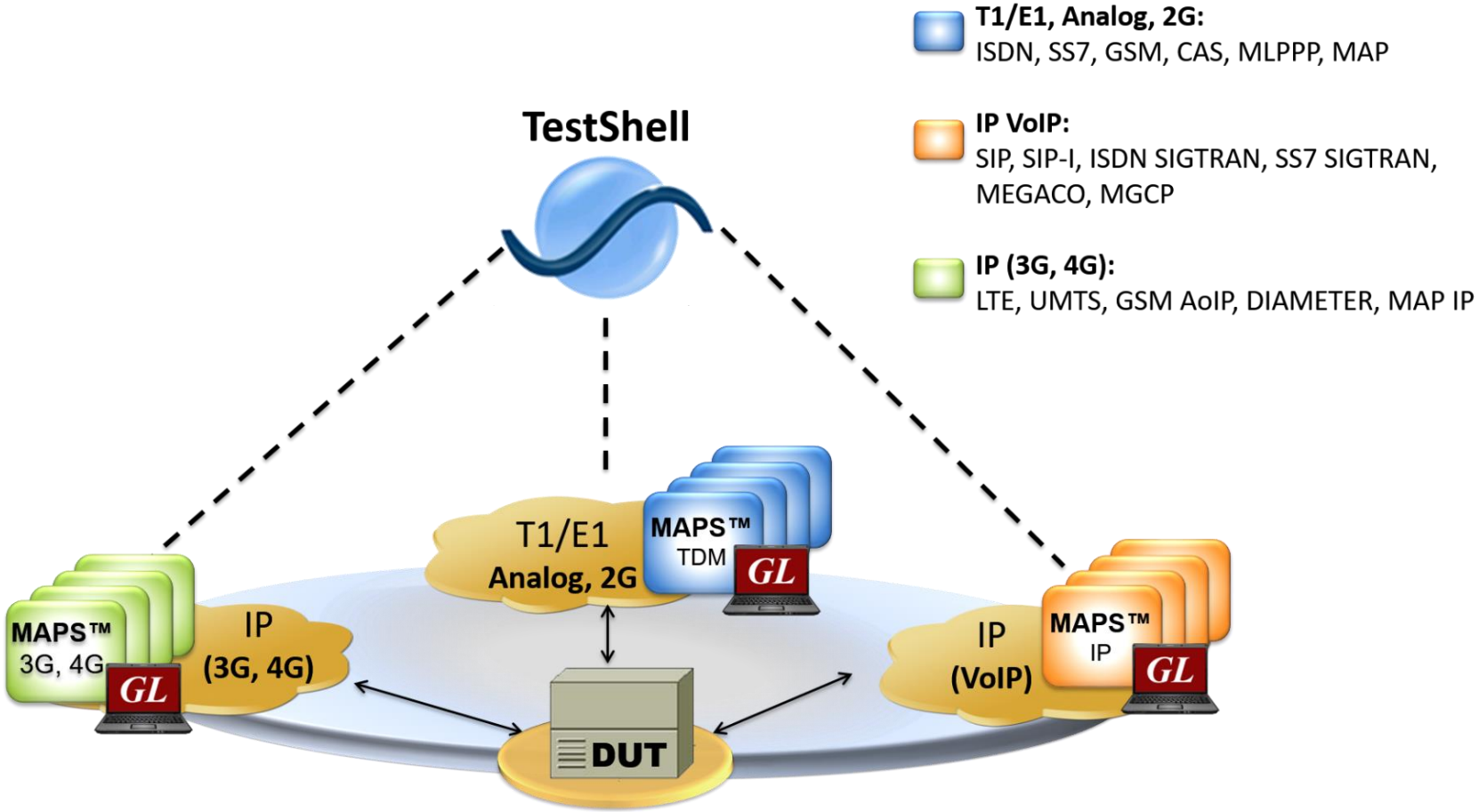
- The same Client Application used to control MAPS™ can be, and very often is, used to control other elements of the System Under Test



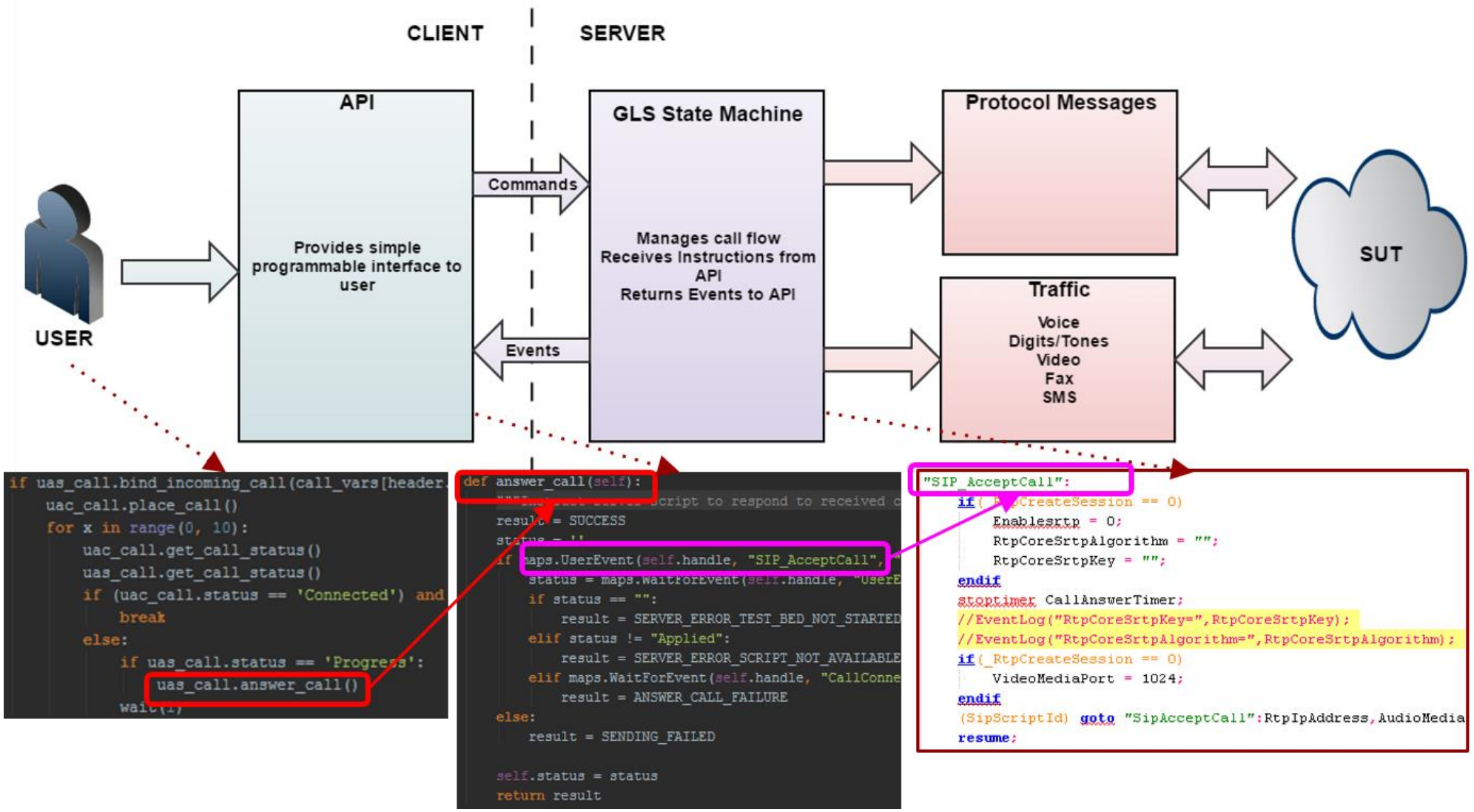
API Architecture

System Integration

- Client Application can be as simple as executing a script from an IDE or it can be integrated into a full-fledged automation test suite like QualiSystems TestShell or HP UFT



API Architecture



APIs High Level vs Low Level

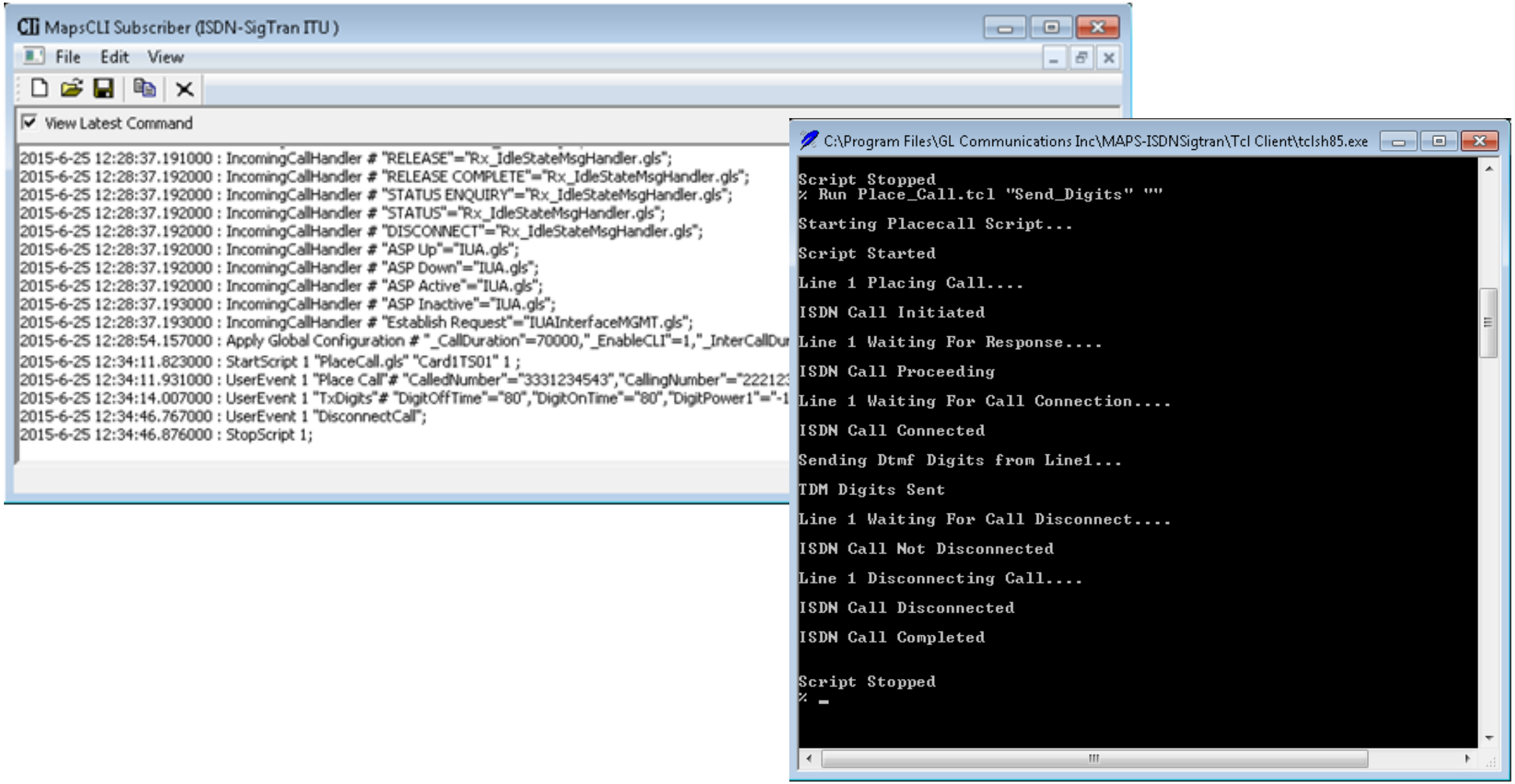
- The API is broken into High and Low level function calls / scripts
- For High Level scripts, all the fine-grained protocol control happen in the script running on the MAPS server, hidden from the API user
- Low Level scripts put the API user in complete control of the protocol stack. This makes Low Level scripts more flexible and powerful, but also correspondingly more complex

```
my_call = local_server.start_call_script("HIGH", "PLACE_CALL")
if my_call.handle != 0:
    my_call.set_local_variable("Contact", "(s)", local_contact)
    my_call.set_local_variable("AddressOfRecord", "(s)", local_aor)
    my_call.set_local_variable("To", "(s)", remote_uri)
    my_call.place_call()
```

```
if local_server.status == "STARTED":
    my_call = local_server.start_call_script("LOW", "PLACE_CALL")
    if my_call.handle != 0:
        my_call.set_local_variable("Contact", "(s)", local_contact)
        my_call.set_local_variable("AddressOfRecord", "(s)", local_aor)
        my_call.set_local_variable("To", "(s)", remote_uri)

        if my_call.rtp_action.create_session(rtp_address, rtp_port) == SUCCESS:
            my_call.send_message("Invite", "InviteImport")
            recvd_msg = my_call.receive_message(timeout)
            if recvd_msg == "100 TRYING" or recvd_msg == "180 RINGING":
```

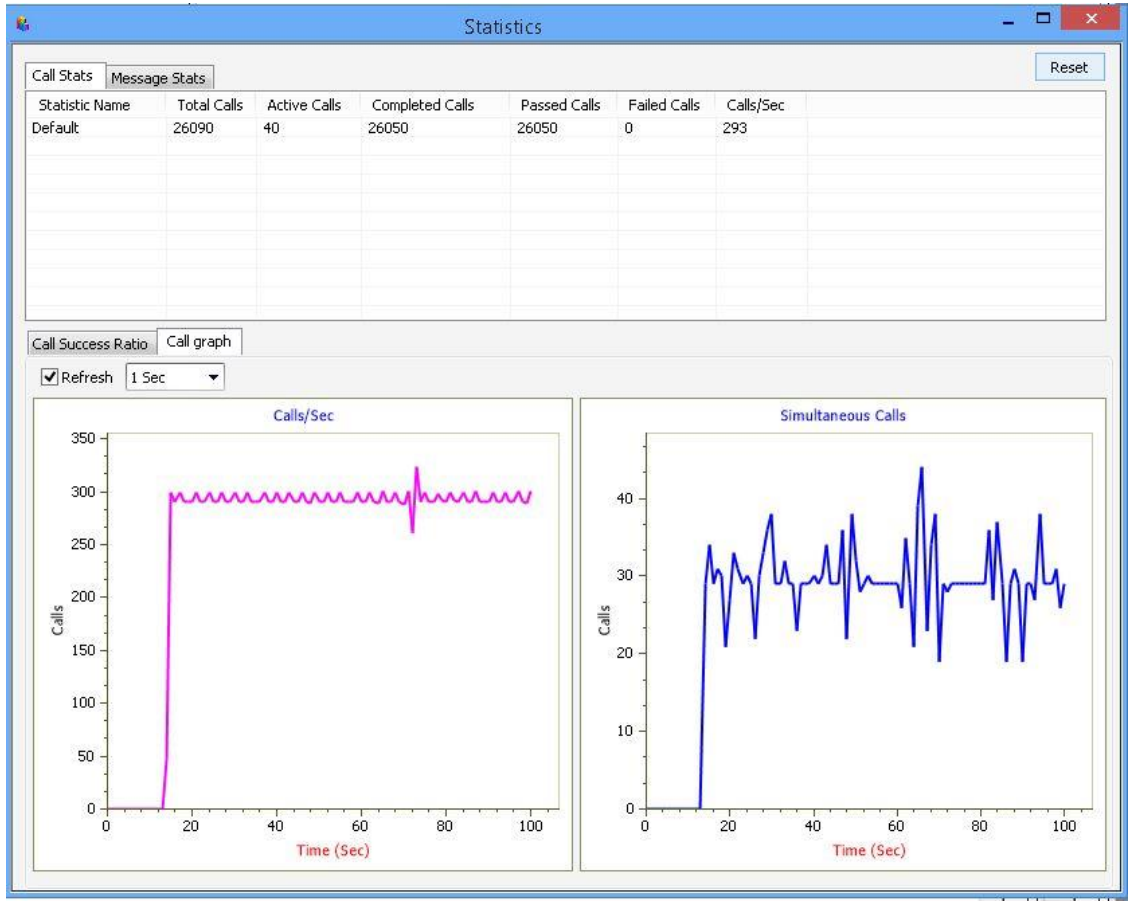
CLI Support



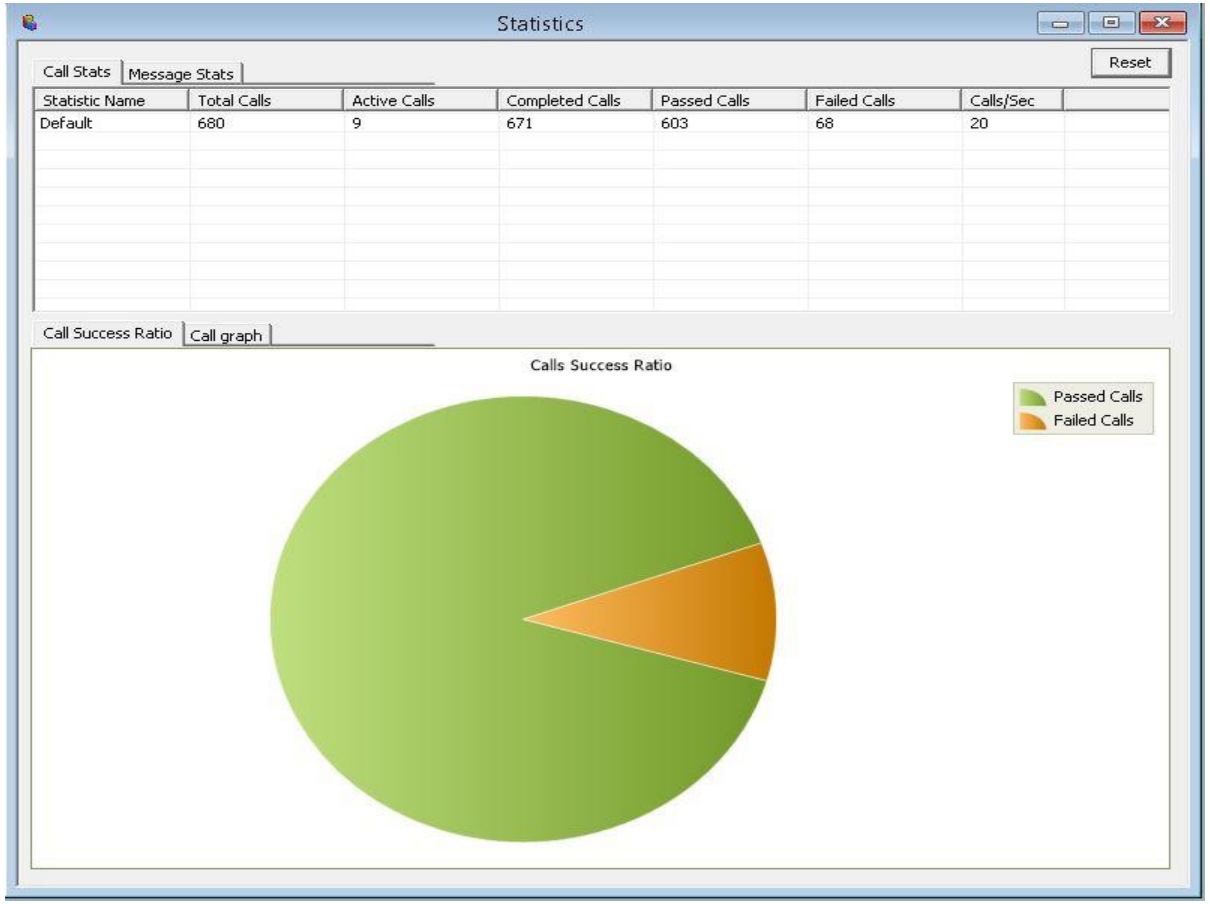
GL Communications

Success Call Ratio Statistics

Call Graph



Call Stats



GL Communications

Thank You