

---

---

# Windows Client Server

---

---



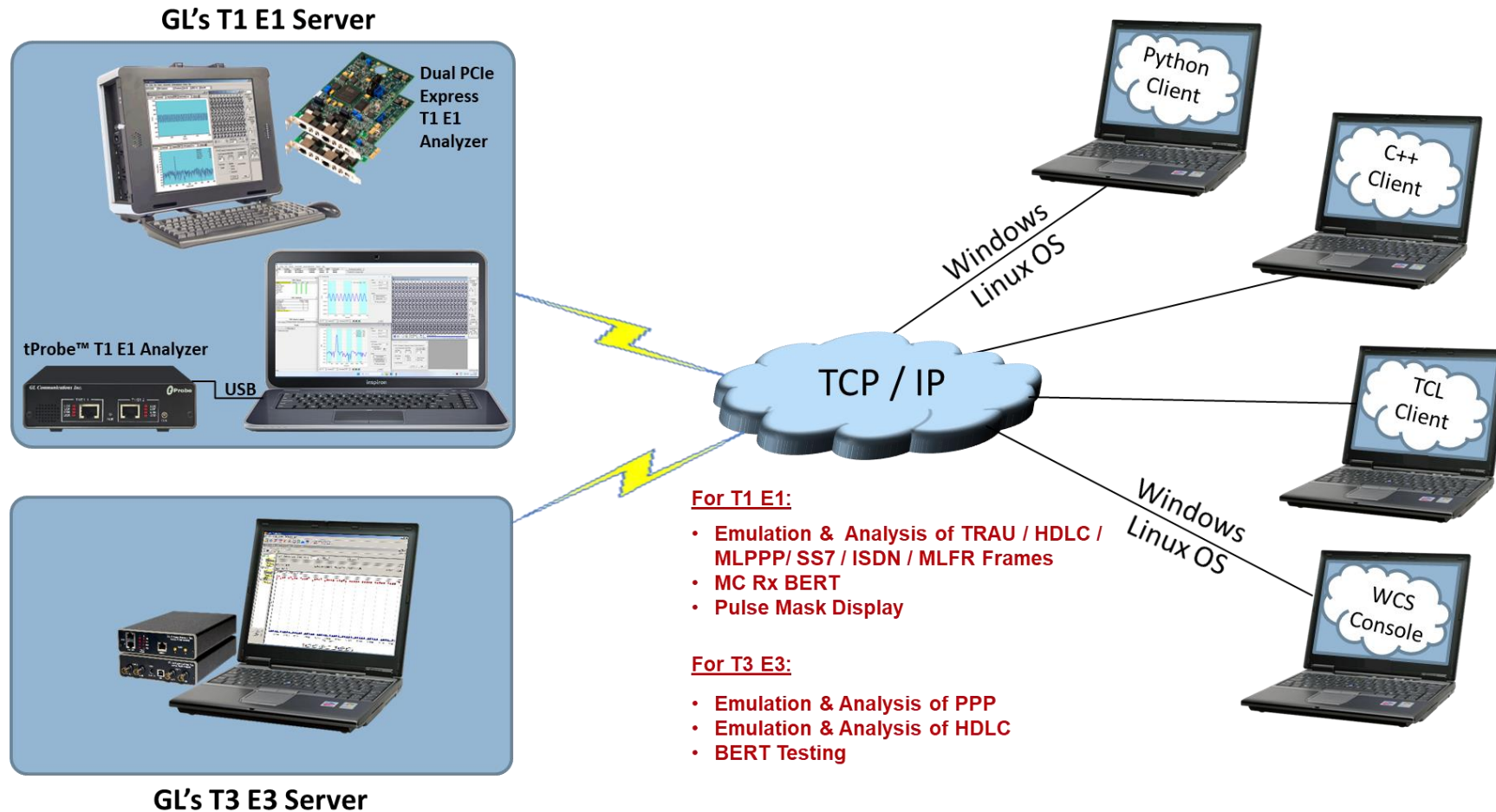
818 West Diamond Avenue - Third Floor, Gaithersburg, MD 20878  
Phone: (301) 670-4784 Fax: (301) 670-9187 Email: [info@gl.com](mailto:info@gl.com)  
Website: <https://www.gl.com>

# Index

- Window Client Server (WCS) General Features
- Tasks Control
- Windows/Linux Client Console
- DLL Based Tasks
  - Place and Answer Calls
  - Precision Delay Measurement
  - Transmit and Receive Files
  - Transmission and Detection of Digits
  - Client-Server w/ CAS Simulator
  - SS1 Emulation and Analysis
  - ISDN Emulation
  - Jitter measurement and Pulse Mask Testing
  - DSP Operations and Dynamic DSP Capability
  - HDLC Emulation and Analysis and
  - HDLC Record / Playback
  - File based HDLC Record / Playback and Remote Record / Playback Module
  - PPP, MLPPP, and Multi-Channel Emulation and Analysis
- DLL Based Tasks
  - File based TRAU Record / Playback Module
  - Multi-channel TRAU Emulation and Analysis Module
  - File based HDLC Record / Playback over SA bits Module
  - MAPS™ - Script Based Protocol Simulation
  - Multi-link Frame Relay Emulation Module
  - File based HDLC Record / Playback over FDL Module
  - Multi-Channel Rx BERT Module
  - Client-Server w/ Traffic Classifier
  - SS7 Decode Agent and ISDN Decode Agent

# Windows Client Server (WCS)

- GL's Windows Client/Server software allows the user of T1 E1 analyzers, the capability of remote operation, automation, and multi-site connectivity



# Applications

- Intrusive / Non-intrusive T1 E1 testing
- Monitoring multiple site locations from a single client
- Shared use of T1 E1 test equipment from multiple client locations
- Automated factory testing on production lines
- Simultaneous testing of high capacity T1 E1 systems through a single Client
- Integration of T1 E1 testing into more complex testing systems
- Collection of call records from remote locations based on signaling (SS7, CAS, ISDN)

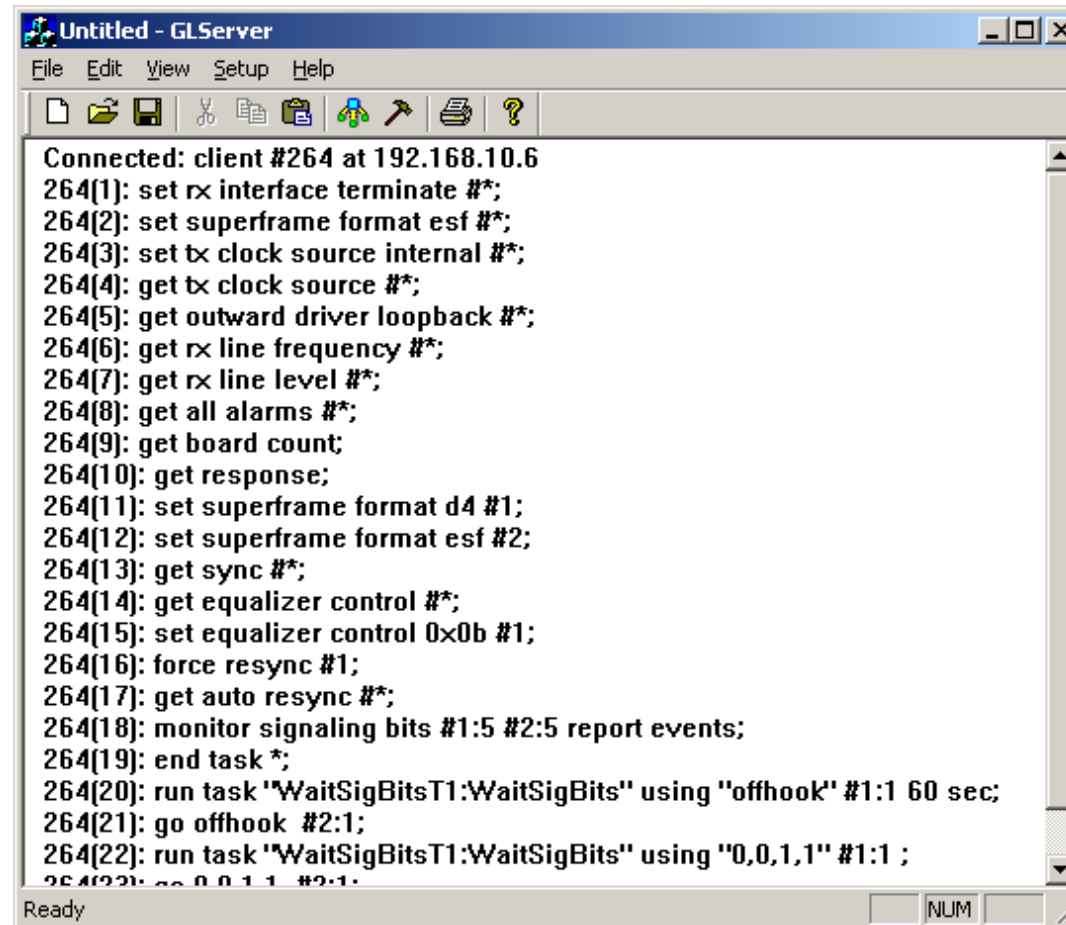
# Features

- Easy control of T1 E1 servers through software clients via TCP/IP sockets
- Supported on Windows®/Linux® operating systems
- Server software can run multiple tasks simultaneously
- Simple modifiable scripts may be developed to perform simple to complex testing
- Perform G.168 EC compliance tests, protocol analysis (HDLC, ISDN, SS7, FDL, MLPPP)
- Perform BERT on selected timeslots involving multiple paths simultaneously
- Monitor, report, and record alarms at various sites every two seconds or as they occur
- Detect and report DTMF/MF/MFC-R2 digits on channels as they occur
- Remote Protocol Analyzers (SS7, ISDN, GR303, V5, HDLC, and Frame relay) can be integrated with Windows Client Server to remotely analyze protocols

# Windows Client / Server Software

## T1 E1 Server

- The log display area is read-only, and normally shows a record of transactions of various types
- Commands and tasks from the client are logged



```
Untitled - GLServer
File Edit View Setup Help
[Icons]
Connected: client #264 at 192.168.10.6
264(1): set rx interface terminate #*;
264(2): set superframe format esf #*;
264(3): set tx clock source internal #*;
264(4): get tx clock source #*;
264(5): get outward driver loopback #*;
264(6): get rx line frequency #*;
264(7): get rx line level #*;
264(8): get all alarms #*;
264(9): get board count;
264(10): get response;
264(11): set superframe format d4 #1;
264(12): set superframe format esf #2;
264(13): get sync #*;
264(14): get equalizer control #*;
264(15): set equalizer control 0x0b #1;
264(16): force resync #1;
264(17): get auto resync #*;
264(18): monitor signaling bits #1:5 #2:5 report events;
264(19): end task *;
264(20): run task "WaitSigBitsT1:WaitSigBits" using "offhook" #1:1 60 sec;
264(21): go offhook #2:1;
264(22): run task "WaitSigBitsT1:WaitSigBits" using "0,0,1,1" #1:1 ;
264(23): go 0,0,1,1 #2:1;
Ready NUM
```

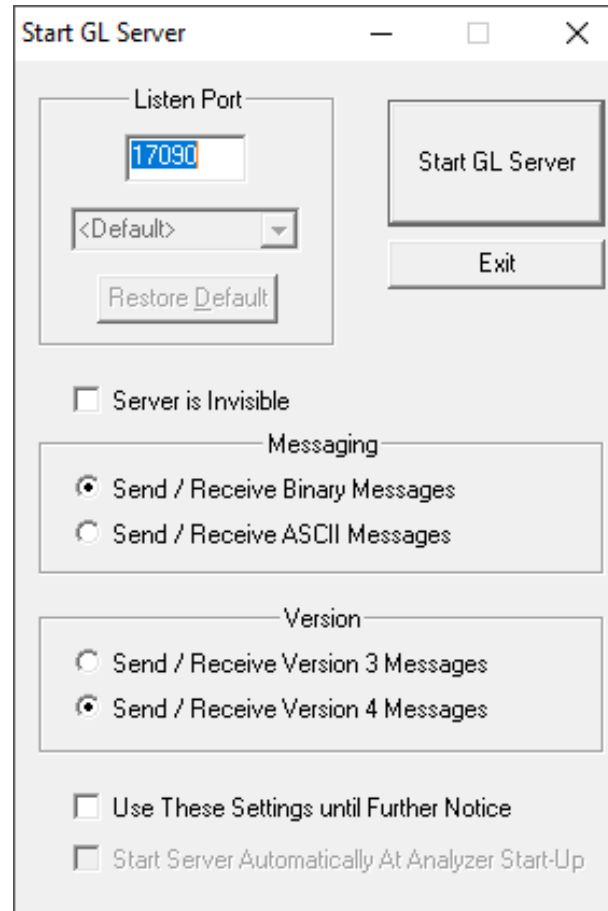
# Windows/Linux Client Console

- Windows/Linux Client (WLC) is a Command Line Interface (CLI) application that issues commands to T1 E1 WCS server and display replies into Console/PowerShell/Terminal Windows. WLC works in Windows® and Linux® versions. However, through SSH or another remote access terminal it can be used on any operating system. WLC is a portable Windows/Linux WCS client communication library compatible with WCS server

```
upArrow - prev cmd; downArrow - next cmd; F7 - recent command list; exit - to disconnect and quit;
F:\src\GLClient\WcsCons\x64\Release>wcscons
Type '?' for help.
conn 192.168.10.78 17090
OK
$monitor all alarms #1;
Task1>>start=0x2481991b
$monitor all alarms #2;
Task2>>start=0x2481ba82
query task 2
>OK
Task2>>#2.los=false, #2.los_count=0, #2.ais=false, #2.ais_count=0, #2.sync=false, #2.sync_count=0, #2.nloop=false, #2.nloop_count=0, #2.rbl=false, #2.rbl_count=0, #2.ferr=false, #2.ferr_count=0, #2.ryel=false, #2.ryel_count=0, #2.bpv=false, #2.bpv_count=0, #2.esovr=false, #2.esovr_count=0, #2.esunf=false, #2.esunf_count=0
query task 1
>OK
Task1>>#1.los=false, #1.los_count=0, #1.ais=false, #1.ais_count=0, #1.sync=false, #1.sync_count=0, #1.nloop=false, #1.nloop_count=0, #1.rbl=false, #1.rbl_count=0, #1.ferr=false, #1.ferr_count=0, #1.ryel=false, #1.ryel_count=0, #1.bpv=false, #1.bpv_count=0, #1.esovr=false, #1.esovr_count=0, #1.esunf=false, #1.esunf_count=0
get multiframe format *
>Unexpected input '*' at offset 23
get multiframe format #*
>#1.mf_fmt=193e; #2.mf_fmt=193e; #3.mf_fmt=193e; #4.mf_fmt=193e
disconn
OK
```

# Launch the Server

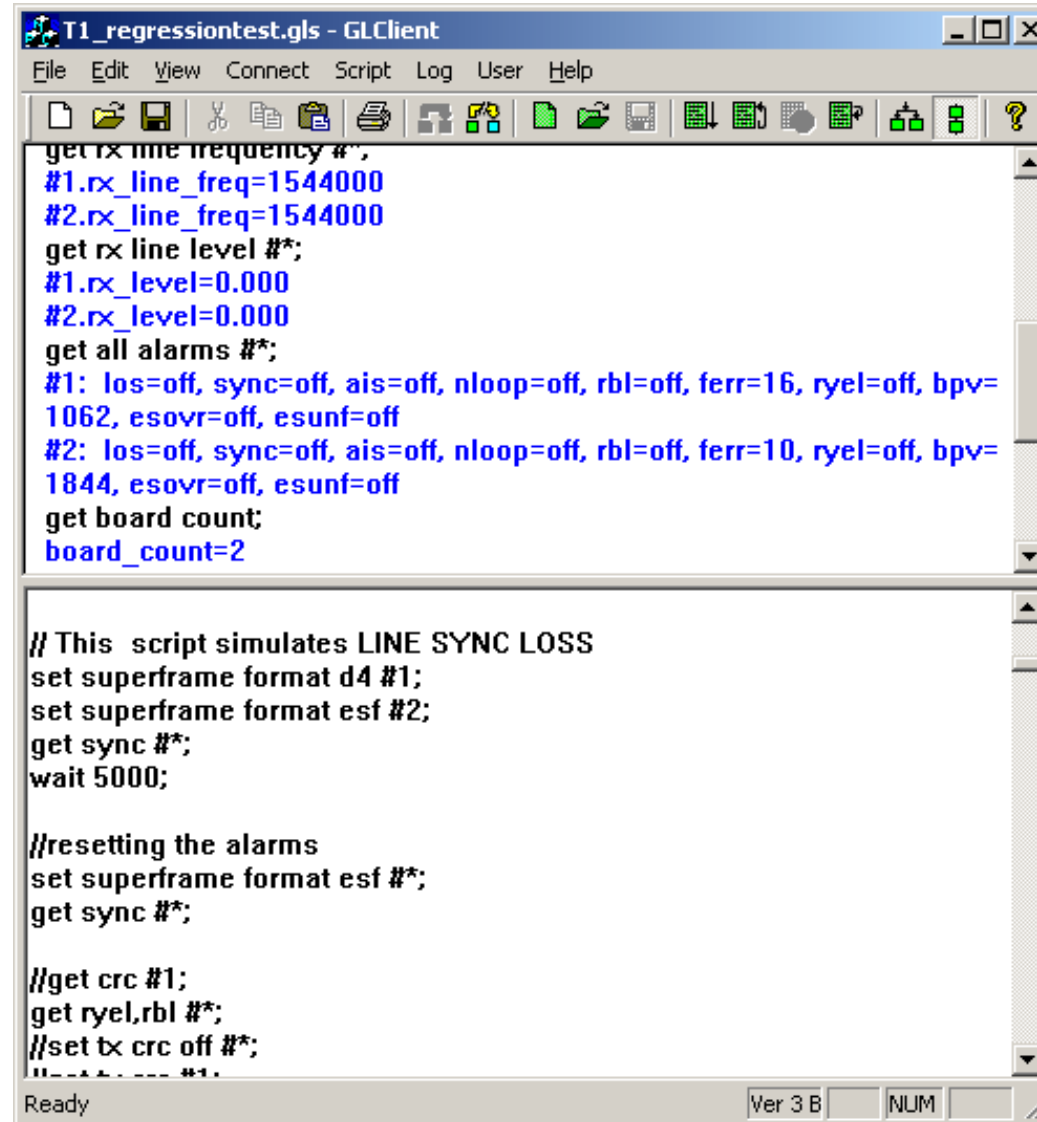
- Launchpad specifies which TCP/IP port should be used to listen for incoming connection requests from clients, as well as the messaging options (ASCII or binary, version 3 or 4)
- The version 4 messaging strategy improves the reliability of message reception at the other end of the communications link





# T1 E1 Client

- In the lower workspace area, the client users key in commands or load in commands from previously saved files
- The upper log area displays the script and the server responses



The screenshot shows the 'T1\_regressiontest.gls - GLClient' window. The interface includes a menu bar (File, Edit, View, Connect, Script, Log, User, Help) and a toolbar with various icons. The main workspace is divided into two sections. The upper section displays the results of executed commands, showing line frequencies, levels, and alarm statuses for two channels. The lower section contains a script for simulating a LINE SYNC LOSS, including commands for setting superframe formats, waiting, and resetting alarms.

```
get rx line frequency #*;
#1.rx_line_freq=1544000
#2.rx_line_freq=1544000
get rx line level #*;
#1.rx_level=0.000
#2.rx_level=0.000
get all alarms #*;
#1: los=off, sync=off, ais=off, nloop=off, rbl=off, ferr=16, ryel=off, bpv=
1062, esovr=off, esunf=off
#2: los=off, sync=off, ais=off, nloop=off, rbl=off, ferr=10, ryel=off, bpv=
1844, esovr=off, esunf=off
get board count;
board_count=2

// This script simulates LINE SYNC LOSS
set superframe format d4 #1;
set superframe format esf #2;
get sync #*;
wait 5000;

//resetting the alarms
set superframe format esf #*;
get sync #*;

//get crc #1;
get ryel,rbl #*;
//set tx crc off #*;
//set tx crc off #1;
```

# WCS Basic Commands Provisioning T1 E1 Cards

- Card Type and Count
- Idle Code
- Signaling Bits
- Audio Operations
- Loopback Settings
- Multi-frame (“Superframe”) Format
- Line Coding
- Transmit Unframed All 1s
- Signaling Modes
- Cyclic Redundancy Check (T1 E1)
- Receiver Interface
- Transmit Clock Source
- Transmit / Receive Equalizer Control
- Jitter Attenuation
- Network Loopback Detection
- International, National, and Extra Bits
- Frame and Multiframe Synchronization
- Drop / Insert
- Alarms
- Power

# Tasks Control

- File Recording and Playback
- Bit Error Rate Test
- Error Insertion
- Alarm Monitoring
- Power Monitoring
- Signaling Bits Monitoring
- Tone and Digits Transmission
- Digit Detection
- Advanced Tone Generation and Detection
- Awaiting Specified Signaling Bits Patterns

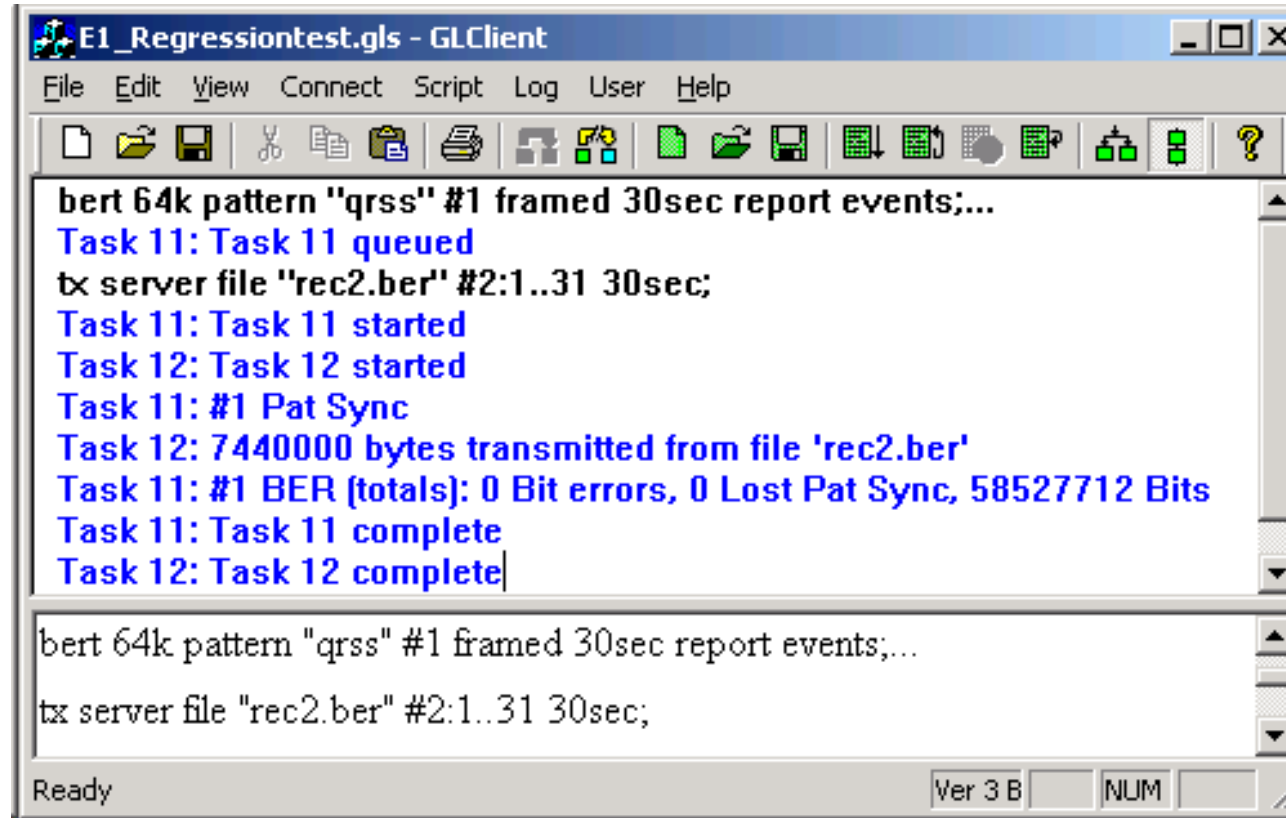
# WCS Modules (DLL Based Commands)

- Place and Answer Calls
- Precision Delay Measurement
- Transmit and Receive Files (Module license #-XX610)
- Transmission and Detection of Digits (Module license #- XX620)
- Client-Server w/ CAS Simulator (Module license #- XX625)
- SS1 Emulation and Analysis (Module license # - XX626)
- ISDN Emulation (Module license #- XX629)
- Pulse Mask Testing
- DSP Operations (Module license #- XX630)
- Dynamic DSP Capability (Module license # XX631)
- Multi-Channel HDLC Emulation and Analysis and File based High Throughput HDLC Record / Playback (Module license #- XX634)
- PPP, MLPPP, and Multi-Channel Emulation and Analysis (Module license #- XX635, 36)

# WCS Modules (DLL Based Commands Contd.)

- File based HDLC Record / Playback and Remote Record / Playback Module (Module license #- XX640, 41)
- File based TRAU Record / Playback Module (Module license #- XX645)
- Multi-channel TRAU Emulation and Analysis Module (Module license #- XX646)
- File based HDLC Record / Playback over SA bits Module (Module license #- XX650)
- MAPS-ISDN (Module License# - XX648)
- MAPS-SS7 (Module License # - XX649)
- Multi-link Frame Relay Emulation Module (Module license #- XX655)
- File based HDLC Record / Playback over FDL Module (Module license #- XX660)
- Multi-Channel Rx BERT Module (Module license #- XX670)
- Client-Server w/ Traffic Classifier (Module license #- XX680)
- SS7 Decode Agent (Module license #- XX690)
- ISDN Decode Agent (Module license #- XX691)
- MAPS-GSM A Interface Emulator (Module License#- XX692)
- MAPS-GSM Abis Interface Emulator (Module License#- XX693)

# Transmit and Receive Files (Module license #-XX610)



The screenshot shows a window titled "E1\_Regressiontest.gls - GLClient" with a menu bar (File, Edit, View, Connect, Script, Log, User, Help) and a toolbar. The main area contains a log of commands and status messages:

```
bert 64k pattern "qrss" #1 framed 30sec report events;...  
Task 11: Task 11 queued  
tx server file "rec2.ber" #2:1..31 30sec;  
Task 11: Task 11 started  
Task 12: Task 12 started  
Task 11: #1 Pat Sync  
Task 12: 7440000 bytes transmitted from file 'rec2.ber'  
Task 11: #1 BER (totals): 0 Bit errors, 0 Lost Pat Sync, 58527712 Bits  
Task 11: Task 11 complete  
Task 12: Task 12 complete|
```

Below this, a second log entry is partially visible:

```
bert 64k pattern "qrss" #1 framed 30sec report events;...  
tx server file "rec2.ber" #2:1..31 30sec;
```

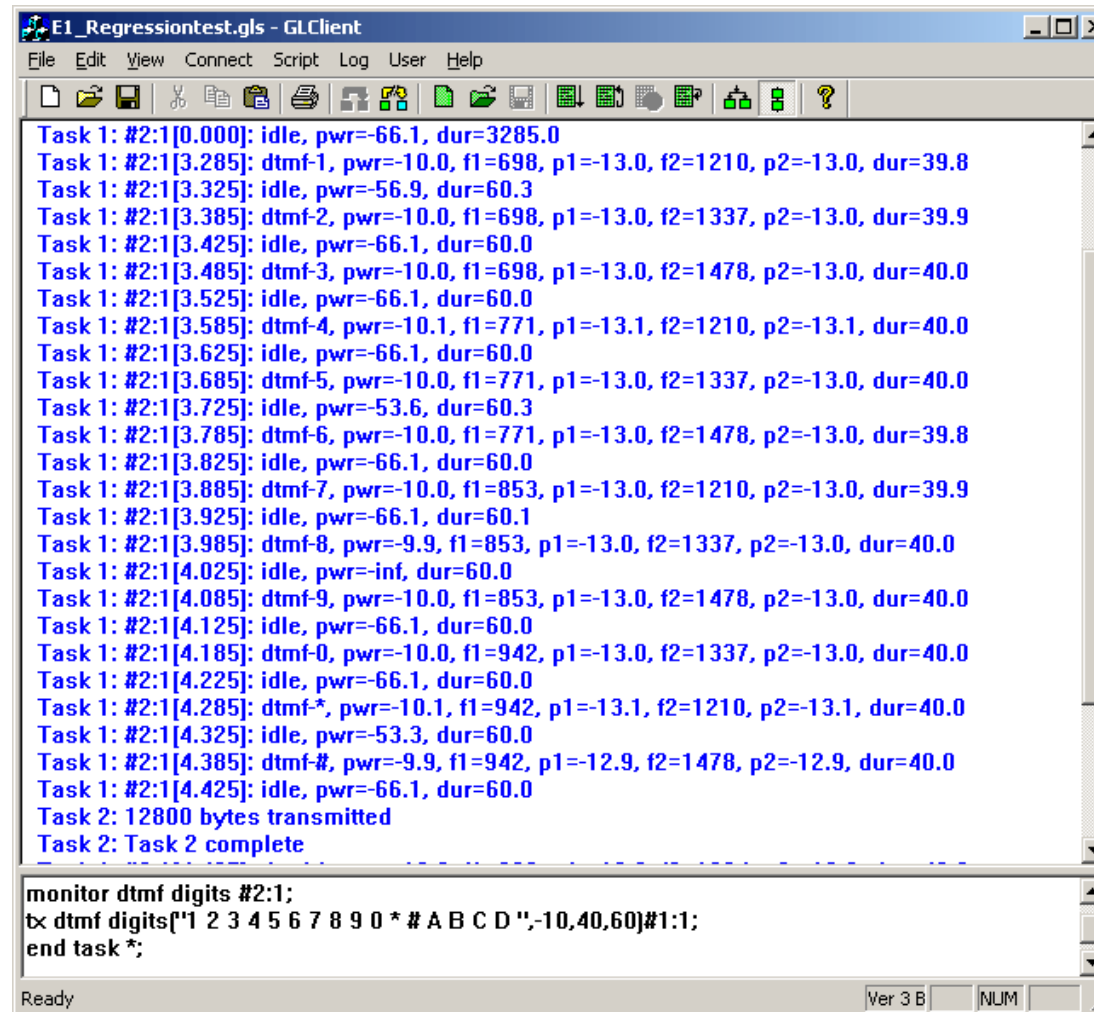
The status bar at the bottom shows "Ready", "Ver 3 B", and "NUM".

- Transmit only / Receive only on all or selected timeslots with one single command

# Transmission and Detection of Digits (Module license #- XX620)

- Detects and reports DTMF/MF/MFC-R2 digits on channels as they occur

## Sample script for Transmitting and Monitoring DTMF Digits



```
E1_Regressiontest.gls - GLClient
File Edit View Connect Script Log User Help
Task 1: #2:1[0.000]: idle, pwr=-66.1, dur=3285.0
Task 1: #2:1[3.285]: dtmf-1, pwr=-10.0, f1=698, p1=-13.0, f2=1210, p2=-13.0, dur=39.8
Task 1: #2:1[3.325]: idle, pwr=-56.9, dur=60.3
Task 1: #2:1[3.385]: dtmf-2, pwr=-10.0, f1=698, p1=-13.0, f2=1337, p2=-13.0, dur=39.9
Task 1: #2:1[3.425]: idle, pwr=-66.1, dur=60.0
Task 1: #2:1[3.485]: dtmf-3, pwr=-10.0, f1=698, p1=-13.0, f2=1478, p2=-13.0, dur=40.0
Task 1: #2:1[3.525]: idle, pwr=-66.1, dur=60.0
Task 1: #2:1[3.585]: dtmf-4, pwr=-10.1, f1=771, p1=-13.1, f2=1210, p2=-13.1, dur=40.0
Task 1: #2:1[3.625]: idle, pwr=-66.1, dur=60.0
Task 1: #2:1[3.685]: dtmf-5, pwr=-10.0, f1=771, p1=-13.0, f2=1337, p2=-13.0, dur=40.0
Task 1: #2:1[3.725]: idle, pwr=-53.6, dur=60.3
Task 1: #2:1[3.785]: dtmf-6, pwr=-10.0, f1=771, p1=-13.0, f2=1478, p2=-13.0, dur=39.8
Task 1: #2:1[3.825]: idle, pwr=-66.1, dur=60.0
Task 1: #2:1[3.885]: dtmf-7, pwr=-10.0, f1=853, p1=-13.0, f2=1210, p2=-13.0, dur=39.9
Task 1: #2:1[3.925]: idle, pwr=-66.1, dur=60.1
Task 1: #2:1[3.985]: dtmf-8, pwr=-9.9, f1=853, p1=-13.0, f2=1337, p2=-13.0, dur=40.0
Task 1: #2:1[4.025]: idle, pwr=-inf, dur=60.0
Task 1: #2:1[4.085]: dtmf-9, pwr=-10.0, f1=853, p1=-13.0, f2=1478, p2=-13.0, dur=40.0
Task 1: #2:1[4.125]: idle, pwr=-66.1, dur=60.0
Task 1: #2:1[4.185]: dtmf-0, pwr=-10.0, f1=942, p1=-13.0, f2=1337, p2=-13.0, dur=40.0
Task 1: #2:1[4.225]: idle, pwr=-66.1, dur=60.0
Task 1: #2:1[4.285]: dtmf-*, pwr=-10.1, f1=942, p1=-13.1, f2=1210, p2=-13.1, dur=40.0
Task 1: #2:1[4.325]: idle, pwr=-53.3, dur=60.0
Task 1: #2:1[4.385]: dtmf-#, pwr=-9.9, f1=942, p1=-12.9, f2=1478, p2=-12.9, dur=40.0
Task 1: #2:1[4.425]: idle, pwr=-66.1, dur=60.0
Task 2: 12800 bytes transmitted
Task 2: Task 2 complete

monitor dtmf digits #2:1;
tx dtmf digits["1 2 3 4 5 6 7 8 9 0 * # A B C D",-10,40,60]#1:1;
end task *;
```

# CAS Simulator (Module license #- XX625)

Timestamp	Setup Time	TS	Trunk	Send Signaling	Receive Signaling
14:47:21		5	E1:0	CALL_RELEASED	
14:47:21		5	E1:1	0,1,0,1	
14:47:21		5	E1:1	CALL_RELEASED	
14:47:21		5	E1:0		1001
14:47:21		5	E1:1		0101
14:47:21		5	E1:0		0101
14:47:37	0.000	5	E1:0	0,0,0,0	
14:47:37		5	E1:1		0000
14:47:37	0.000	5	E1:1	SEIZURE_DETECTED	
14:47:37	0.001	5	E1:1	INCOMING_CALL	
14:47:40	3.006	5	E1:0	0,0,0,0	
14:47:41	4.507	5	E1:0	0,1,0,1	
14:47:42	4.425	5	E1:1		0101
14:47:44	7.528	5	E1:0	0,0,0,0	

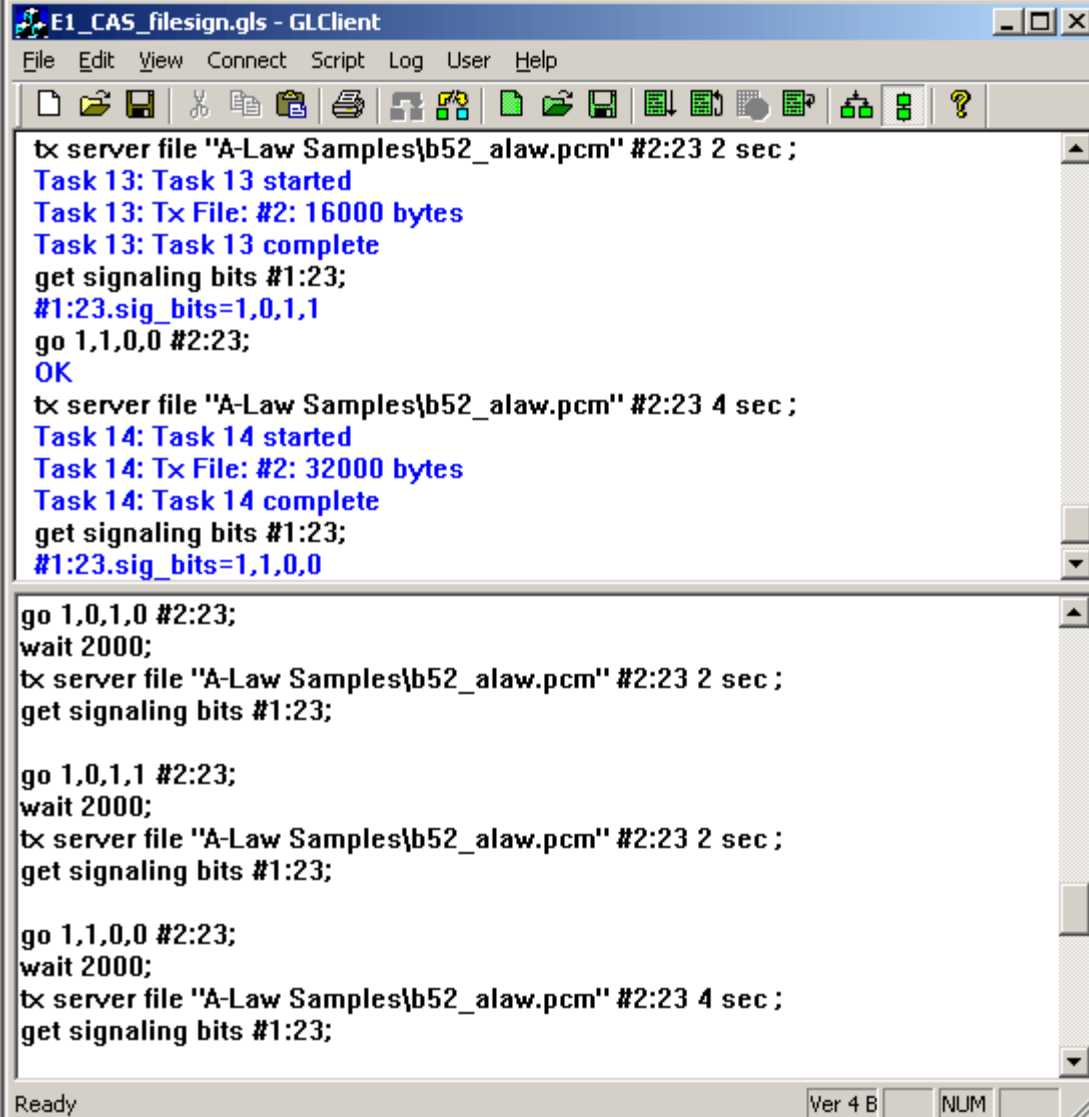
```
1 //Initial Signaling Definitions
2 State=INIT
3 Register Outbound; P=0001,PR=1001;
4 Start Signaling Detector; ABCD=0n,0n,0n,0n;
5 Start Detector; tone=na.mkd; dtmf=dtmf.mkd; qual=qual40.mkd;
6 Change Signal=1001;
7 Send Call Event=CALL_RELEASED;
8 End State
9
10 //Outbound initial state,
11 State=OUTBOUND_INITIATED
12 //Call Connected state when receives A=0100 signaling
13 IF Signal=0101; THEN
14 Send Call Event=CALL_CONNECTED;
15 END IF
16 //Call disconnected before answering call when receives 1101 signaling
17 IF Signal=1100; THEN
18 Change Signal=1001;
19 Send Call Event=CALL_RELEASED;
20 END IF
21 End State
22
23 //Outbound Connected State, when receives AR=1101 signaling
24 //Call disconnected after answering the call
25 State=CONNECTED_OUTBOUND
26 IF... Statements
27 Change Signal=1001;
28 Send Call Event=CALL_RELEASED;
29 END IF
30 End State
31
32
33
```

- CAS simulation using GUI based client-side application
- Simulates and analyzes any user-defined CAS protocols by providing signaling bit transitions and forward / backward frequency tones / digits



# CAS Simulator

- CAS simulation using client-server command line application
- Create user-defined CAS scripts with a script editor
- Supported Protocols:
  - E1 MFC-R2 (All variants, fully/semi compelled)
  - T1 Winkstart ( R1 wink)
  - T1 Loopstart
  - T1 Groundstart
  - E1 European Digital CAS (EUC)
  - User-defined CAS Protocol



```
tx server file "A-Law Samples\b52_alaw.pcm" #2:23 2 sec ;
Task 13: Task 13 started
Task 13: Tx File: #2: 16000 bytes
Task 13: Task 13 complete
get signaling bits #1:23;
#1:23.sig_bits=1,0,1,1
go 1,1,0,0 #2:23;
OK
tx server file "A-Law Samples\b52_alaw.pcm" #2:23 4 sec ;
Task 14: Task 14 started
Task 14: Tx File: #2: 32000 bytes
Task 14: Task 14 complete
get signaling bits #1:23;
#1:23.sig_bits=1,1,0,0

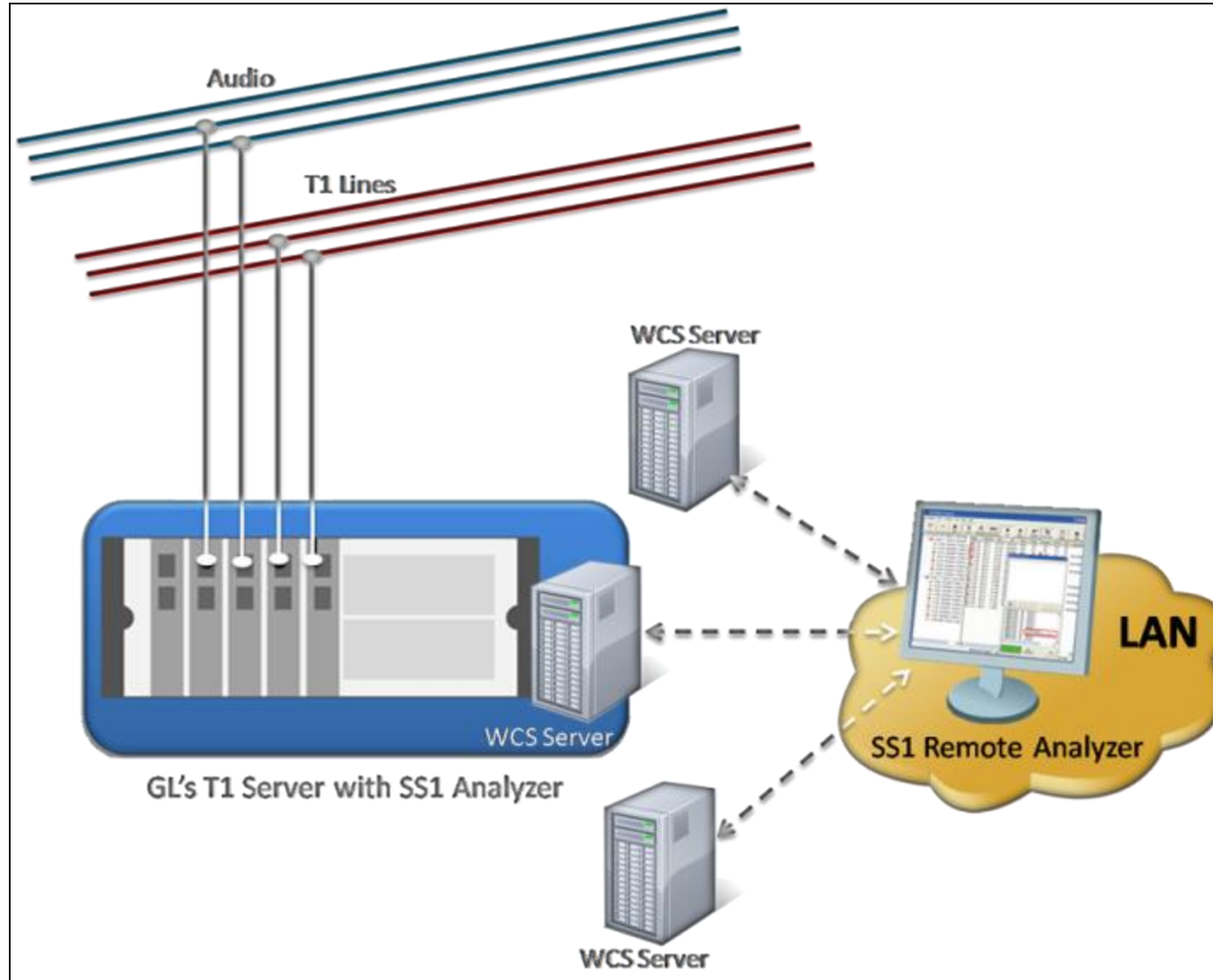
go 1,0,1,0 #2:23;
wait 2000;
tx server file "A-Law Samples\b52_alaw.pcm" #2:23 2 sec ;
get signaling bits #1:23;

go 1,0,1,1 #2:23;
wait 2000;
tx server file "A-Law Samples\b52_alaw.pcm" #2:23 2 sec ;
get signaling bits #1:23;

go 1,1,0,0 #2:23;
wait 2000;
tx server file "A-Law Samples\b52_alaw.pcm" #2:23 4 sec ;
get signaling bits #1:23;
```

Ready Ver 4 B NUM

# Real-time/ Remote SS1 Emulator and Analyzer (Module License # - XX626)



# Highlights

- Supports Remote, Real-time and File-based analysis using client-server based SS1 Analyzer
- Analyzer can capture either TDM or VF audio signals
- Analyzer can analyze either 2-digit or 3-digit dial codes
- Analyzer displays received dial codes, including the characteristics of the underlying tones
- Generate and introduce SS1 Dial Codes on Transmit Channels using SS1 Dialer
- Dual monitoring capability allows multiple instances of SS1 analyzer to simultaneously tap E and W direction traffic
- Operate the SS1 Analyzer either remotely from the data acquisition site, or on the local PC
- Spectral Graph feature presents a captured dial code as a graphical waveform

# SS1 Analyzer (Module License # - XX626)

SS1 Analyzer - #2:0

Configure View Connect Run Results Help

Setup Load Save Connect Disconnect View WCS Run Stop Real-Time From File Clear Help

File Edit View Setup Help

Connected: client #34  
 348: get app license 1  
 348: stop tasks on dis  
 348: get port type #\*;  
 348: get port count;  
 348: monitor tones('M  
 348: rx server file "C:  
 Ready

18:49:51 '23' TS=#2:0 dur=950

- +0.000 '2' TS=#2:0 dur=425
- +0.000 'SS1/mark' TS=#2:0 dur=100
- +0.100 'SS1/space' TS=#2:0 dur=42
- +0.141 'SS1/mark' TS=#2:0 dur=59
- +0.201 'SS1/space' TS=#2:0 dur=224
- +0.425 '3' TS=#2:0 dur=525
- +0.425 'SS1/mark' TS=#2:0 dur=100
- +0.525 'SS1/space' TS=#2:0 dur=41
- +0.566 'SS1/mark' TS=#2:0 dur=59
- +0.625 'SS1/space' TS=#2:0 dur=41
- +0.666 'SS1/mark' TS=#2:0 dur=59
- +0.726 'SS1/space' TS=#2:0 dur=224

18:49:47 '45' TS=#2:0 dur=1350

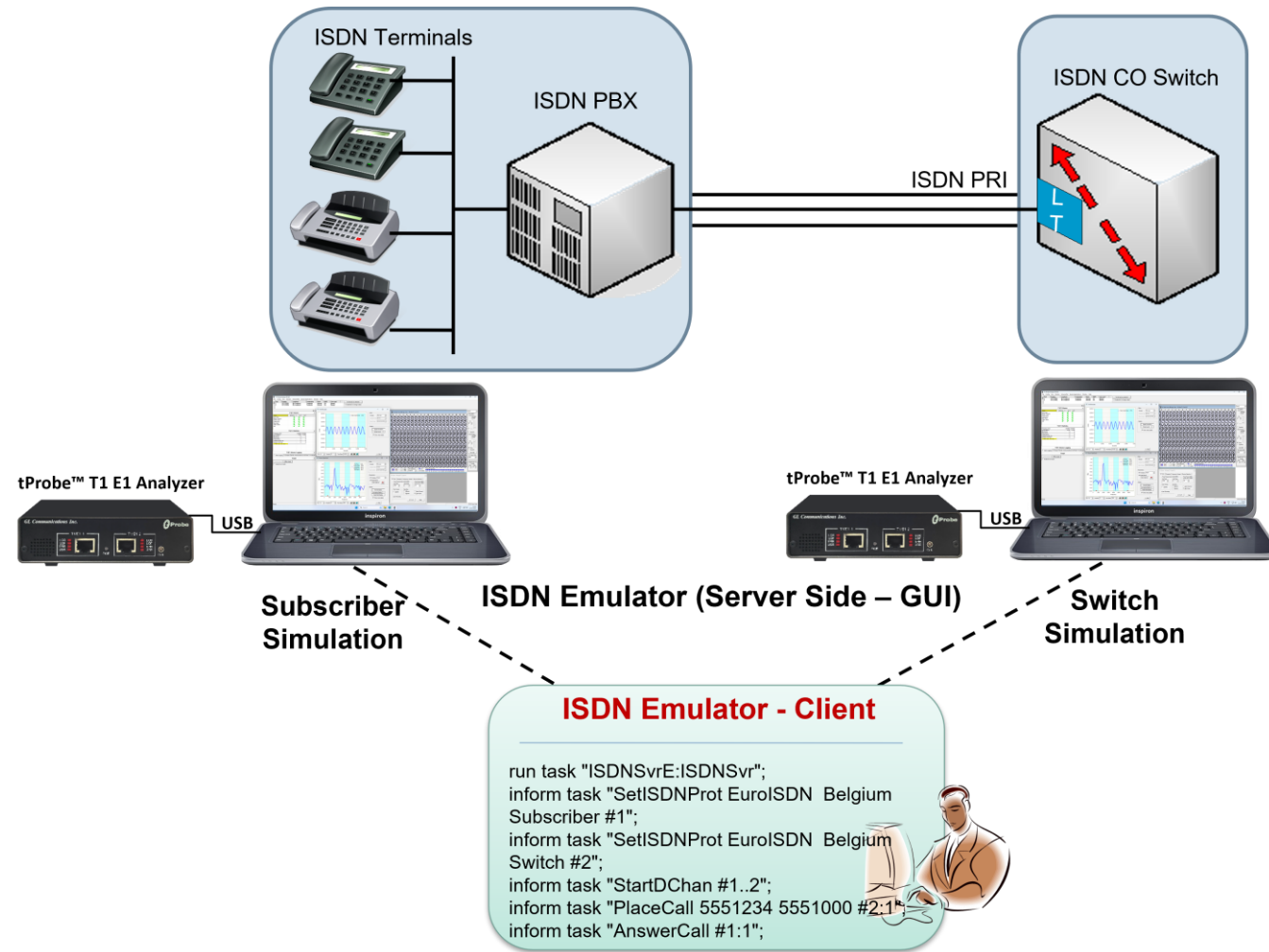
Seq#	Channel	Time	ID	Freq	Power	Duration	Tot Pwr	S/N
2-1-01	#2:0	18:49:51	SS1/mark	2601	-8.0	100	-8.0	35
2-1-02	#2:0	+0.100	SS1/space	2401	-7.9	42	-7.9	21
2-1-03	#2:0	+0.141	SS1/mark	2601	-8.0	59	-8.0	40
2-1-04	#2:0	+0.201	SS1/space	2401	-8.0	224	-8.0	27
2-2-01	#2:0	+0.425	SS1/mark	2601	-8.0	100	-8.0	30

Spectral Display

Parameter	Low Reject	Low Accept	Standard	High Accept	High Reject
Initial Mark (2600 Hz) duration (ms)	95	95	100	105	160
Nominal Mark (2600 Hz) duration (ms)	30	55	58	65	120
Nominal Space (2400 Hz) duration (ms)	20	35	42	45	90
Final Space (2400 Hz) duration (ms)	150	200	225	(no limit)	(no limit)
Privacy Set duration (ms)	130	390	400	410	(no limit)
Privacy Release duration (ms)	610	995	1000	1005	(no limit)
Mark (2600 Hz) frequency (Hz)	2563	2597	2600	2603	2637
Space (2400 Hz) frequency (Hz)	2366	2392	2400	2408	2434
Signal power range (dBm)	-24	-10	-8	-6	3

Ready

# ISDN Emulation (Module license # - XX629)



# Sample script for Placing and Answering ISDN calls

- Place and Answer ISDN Calls
- Monitor all link state and call state

The screenshot displays two windows. The left window, titled 'Untitled - GLClient', contains a script for managing ISDN calls. The right window, titled 'ISDN Protocol Analysis Q.93x', shows a table of captured frames and a detailed view of a specific frame.

**GLClient Script:**

```
Task 1: TS#2:28,CallState=PROCEEDING
Task 1: TS#2:28,CallState=ALERTING
Task 1: TS#2:29,CallState=PROCEEDING
Task 1: TS#2:29,CallState=ALERTING
Task 1: TS#2:30,CallState=PROCEEDING
Task 1: TS#2:30,CallState=ALERTING
Task 1: TS#2:31,CallState=PROCEEDING
Task 1: TS#2:31,CallState=ALERTING
inform task "AnswerCall #1:1..31";
Task 1 informed
Task 1: TS#1:1,CallState=CONNECTED
Task 1: TS#1:2,CallState=CONNECTED
Task 1: TS#1:3,CallState=CONNECTED
Task 1: TS#1:4,CallState=CONNECTED
Task 1: TS#1:5,CallState=CONNECTED
Task 1: TS#1:6,CallState=CONNECTED

run task "ISDNsvrE:ISDNsvr";
inform task "SetISDNProt EuroISDN Belgium Switch #1";
inform task "SetISDNProt EuroISDN Belgium Subscriber #2";
inform task "StartDChan #1..2";
inform task "PlaceCall 5551234 5551000 #2:1..31";
inform task "AnswerCall #1:1..31";
inform task "DisconnectCall CAUSE_NORMAL_CLEAR #1:1..2";
inform task "StopDChan #1..2";
```

**ISDN Protocol Analysis Q.93x Table:**

Frame#	TIME (Relative)	Len	E...	C/R	SAPI	TEI	CTL	P/F	N(S)	N(R)	F...	CRV	Message Type
177	00:00:47.382125	6		Response(User), Comma...	0	0	Supervisory	0		49	RR		
178	00:00:47.482250	15		Response(User), Comma...	0	0	Information	0	49	30		25	ALERTING
179	00:00:47.484250	16		Response(User), Comma...	0	0	Information	0	50	30		26	CALL PROCEEDING
180	00:00:47.504375	15		Response(User), Comma...	0	0	Information	0	51	30		26	ALERTING
181	00:00:47.506375	16		Response(User), Comma...	0	0	Information	0	52	30		27	CALL PROCEEDING
182	00:00:47.508500	15		Response(User), Comma...	0	0	Information	0	53	30		27	ALERTING
183	00:00:47.510500	16		Response(User), Comma...	0	0	Information	0	54	30		28	CALL PROCEEDING
184	00:00:47.512500	15		Response(User), Comma...	0	0	Information	0	55	30		28	ALERTING

**ISDN Protocol Analysis Q.93x Details:**

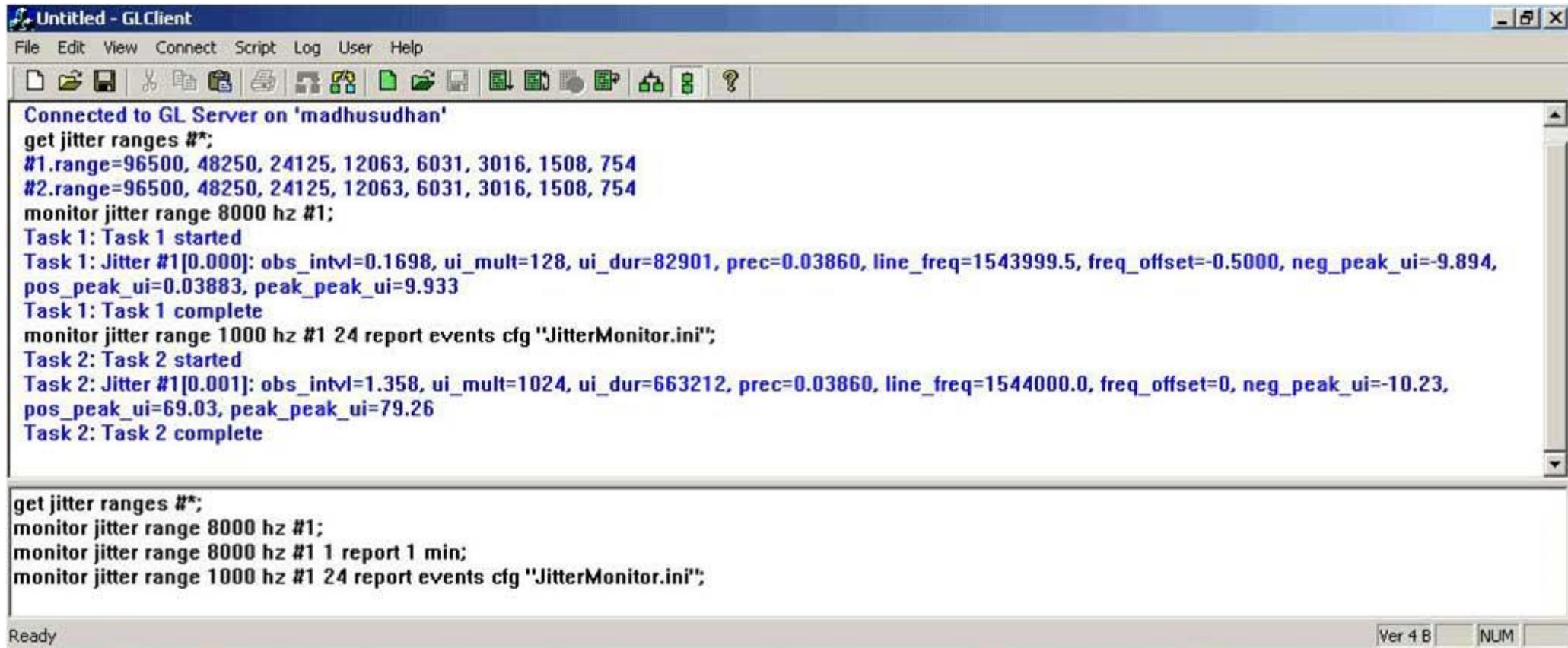
```
Card1 TimeSlot=16 Frame=177 at 00:00:47.382125 OK Len=6
HDLC Frame Data + FCS
----- LAPD Layer -----
C/R = .....1. Response(User), Command(Network)
SAPI = 000000.. (0)
TEI = 000000.. (0)
Ctl = .....01 Supervisory
Supervisory Function = ....00.. RR
P/F = .....0 (0)
N(R) = 0110001. (49)
```

**Hex Dump of the Frame Data:**

```
+-----+-----+-----+-----+-----+-----+
02 01 01 62 B8 C6                                     b,Æ
```



# WCS Jitter Measurement



The screenshot shows a window titled "Untitled - GLClient" with a menu bar (File, Edit, View, Connect, Script, Log, User, Help) and a toolbar. The main text area contains the following commands and output:

```
Connected to GL Server on 'madhusudhan'  
get jitter ranges #*;  
#1.range=96500, 48250, 24125, 12063, 6031, 3016, 1508, 754  
#2.range=96500, 48250, 24125, 12063, 6031, 3016, 1508, 754  
monitor jitter range 8000 hz #1;  
Task 1: Task 1 started  
Task 1: Jitter #1[0.000]: obs_intvl=0.1698, ui_mult=128, ui_dur=82901, prec=0.03860, line_freq=1543999.5, freq_offset=-0.5000, neg_peak_ui=-9.894,  
pos_peak_ui=0.03883, peak_peak_ui=9.933  
Task 1: Task 1 complete  
monitor jitter range 1000 hz #1 24 report events cfg "JitterMonitor.ini";  
Task 2: Task 2 started  
Task 2: Jitter #1[0.001]: obs_intvl=1.358, ui_mult=1024, ui_dur=663212, prec=0.03860, line_freq=1544000.0, freq_offset=0, neg_peak_ui=-10.23,  
pos_peak_ui=69.03, peak_peak_ui=79.26  
Task 2: Task 2 complete
```

Below this, a separate section shows the following commands:

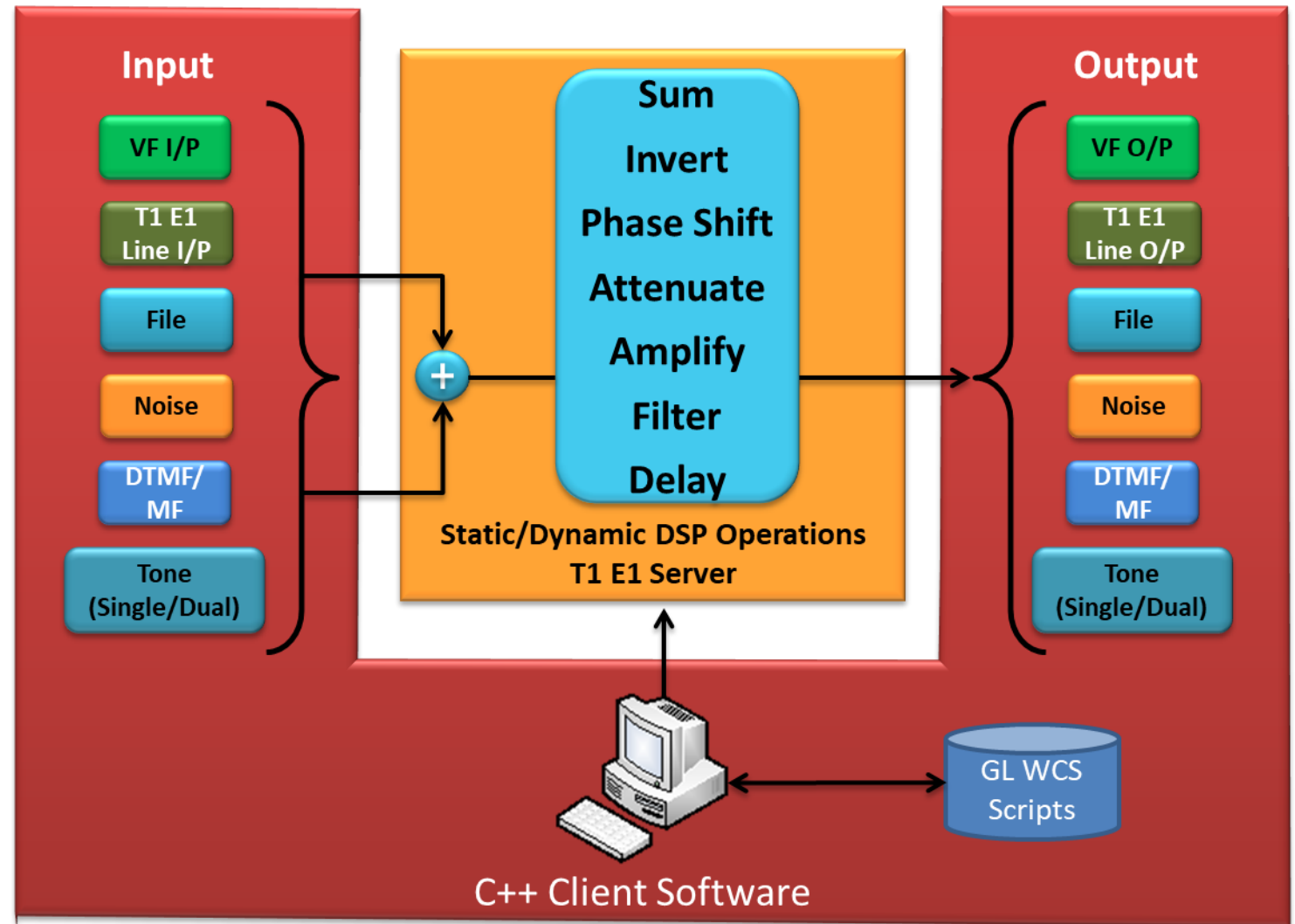
```
get jitter ranges #*;  
monitor jitter range 8000 hz #1;  
monitor jitter range 8000 hz #1 1 report 1 min;  
monitor jitter range 1000 hz #1 24 report events cfg "JitterMonitor.ini";
```

The status bar at the bottom shows "Ready" on the left and "Ver 4 B" and "NUM" on the right.

- Monitor Jitter ranges and perform Jitter measurement through Windows Client-Server commands

# Digital Signal Processing (DSP) (Module license #- XX630)

- Static Operators - provides the ability to specify a sequence of digital signal processing steps to be performed on incoming and/or outgoing timeslots
- Dynamic Operators - perform dynamic or time-varying operations via schedules, which specifies a sequence of digital signal processing steps to be performed at specified time offset for each operator on incoming and/or outgoing signals





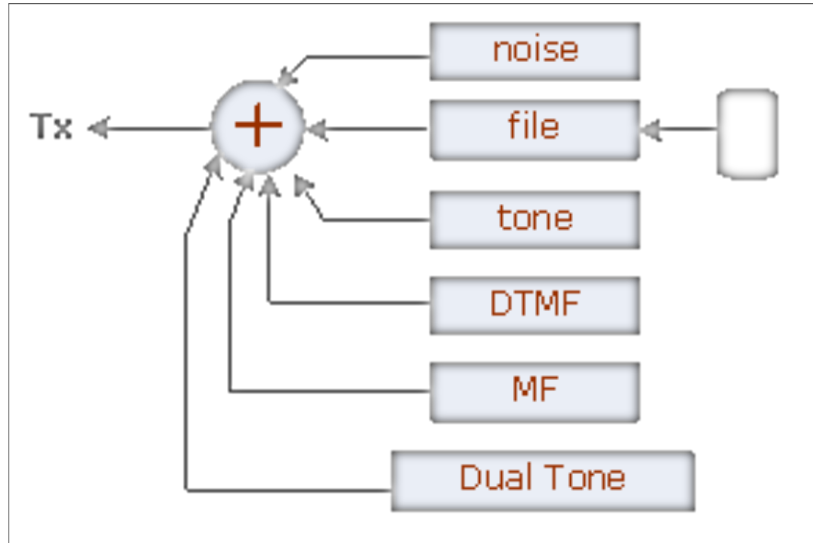
# Static DSP Operations

- Following functions can be performed using Static DSP operators:
  - Sum
  - Invert
  - Filter
  - Delay
  - Amplify
  - Attenuate
  - Bxor, Bor, Brev, Bnot, Band
  - Infile, Outfile
  - White noise, Tone, Dual Tone, Phase Shift, DTMF Digits, MF Digits, MFCR2 Digits
  - Power monitor, Signaling Bits monitor, Const, Bytes, and many others

# Static DSP Operations

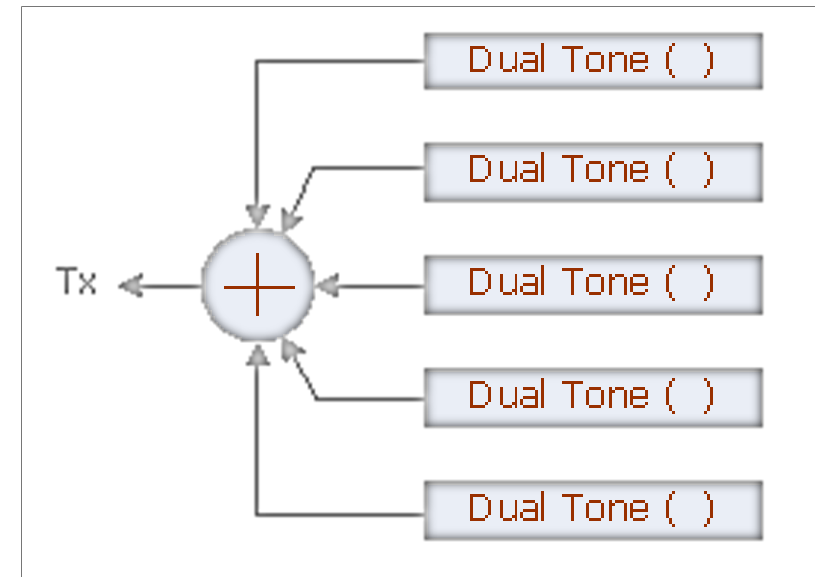
- Basic Static Operations for Echo Paths Simulation
- Sum digitally synthesized sources
- Sum multiple Dual tone generators
- Sum signal with delayed and attenuated version of itself
- Parallel echo paths summed with digitally synthesized tone / noise / file
- Sum signal with inverted version of itself
- Static Operations using C++ Client
- Transmit filtered tones and white noise
- Adding speech and noise to the receive data
- Adding noise and phase shift tone to the speech data
- Testing Arithmetical Functions on Incoming Bit Stream
- Double talk simulation for echo canceller testing

# Static DSP Operations - Examples

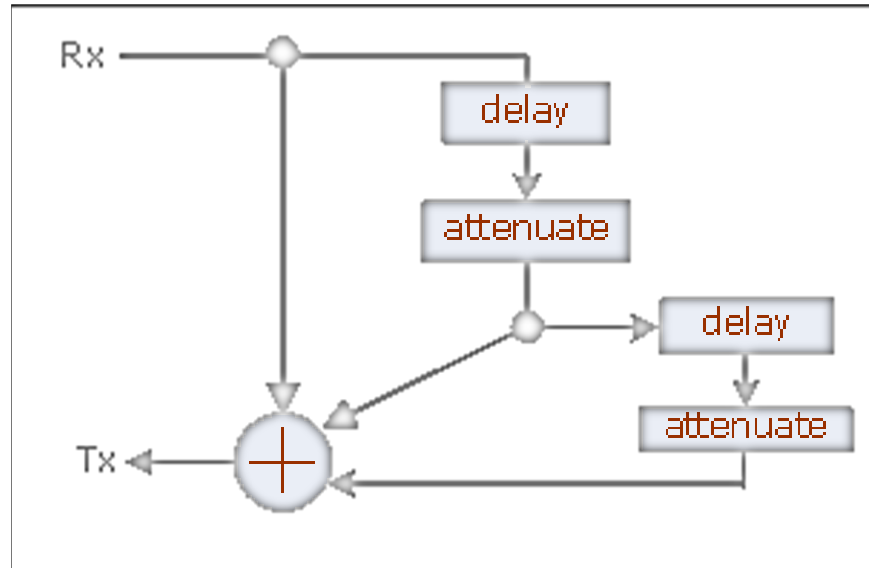


Digitally synthesized generators of tone, noise, DTMF digits, MF digits, and dual tone are summed and transmitted into timeslot

Multiple dual tone generators with possibly different parameters are summed and transmitted into timeslot

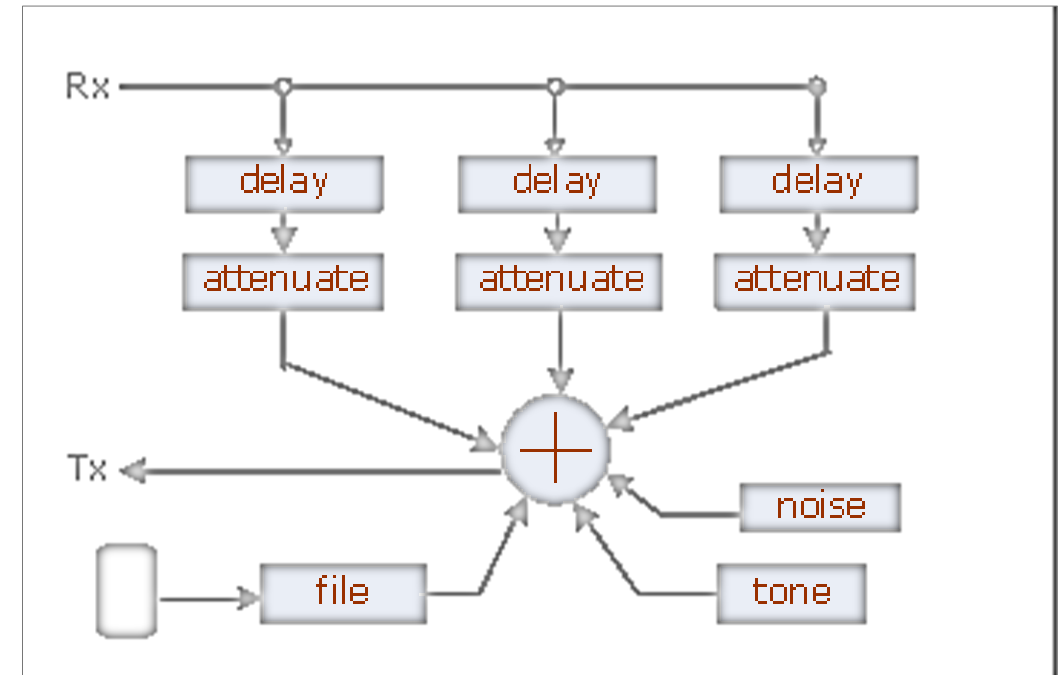


# Static DSP Operations – Examples (Contd.)

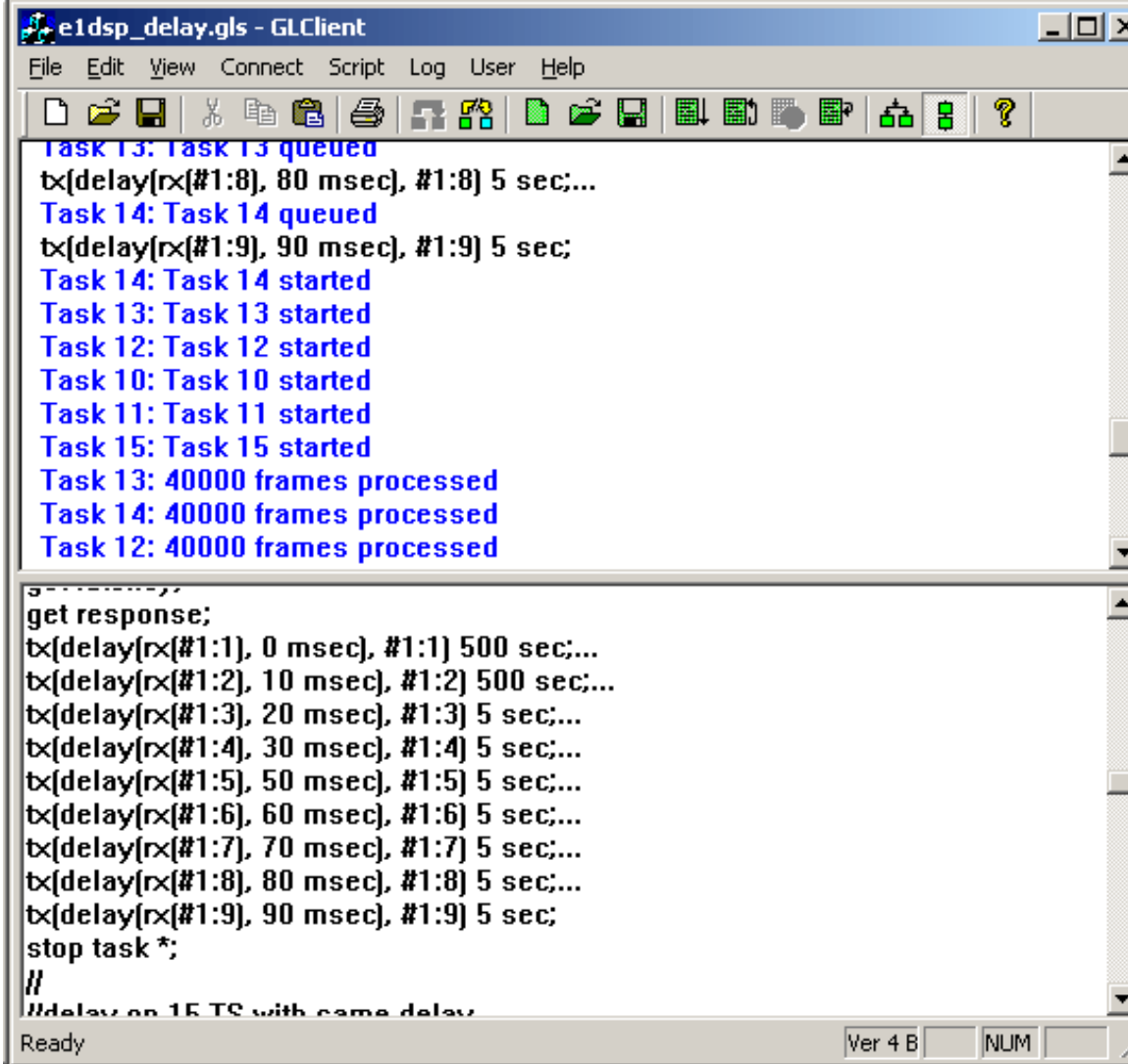


Three parallel echo paths are summed with a digitally synthesized tone and noise and a PCM file, a more complex structure for echo path modelling

Receive timeslot is summed with delayed and attenuated versions of itself and transmitted back



# Static DSP Operations – Examples (Contd.)



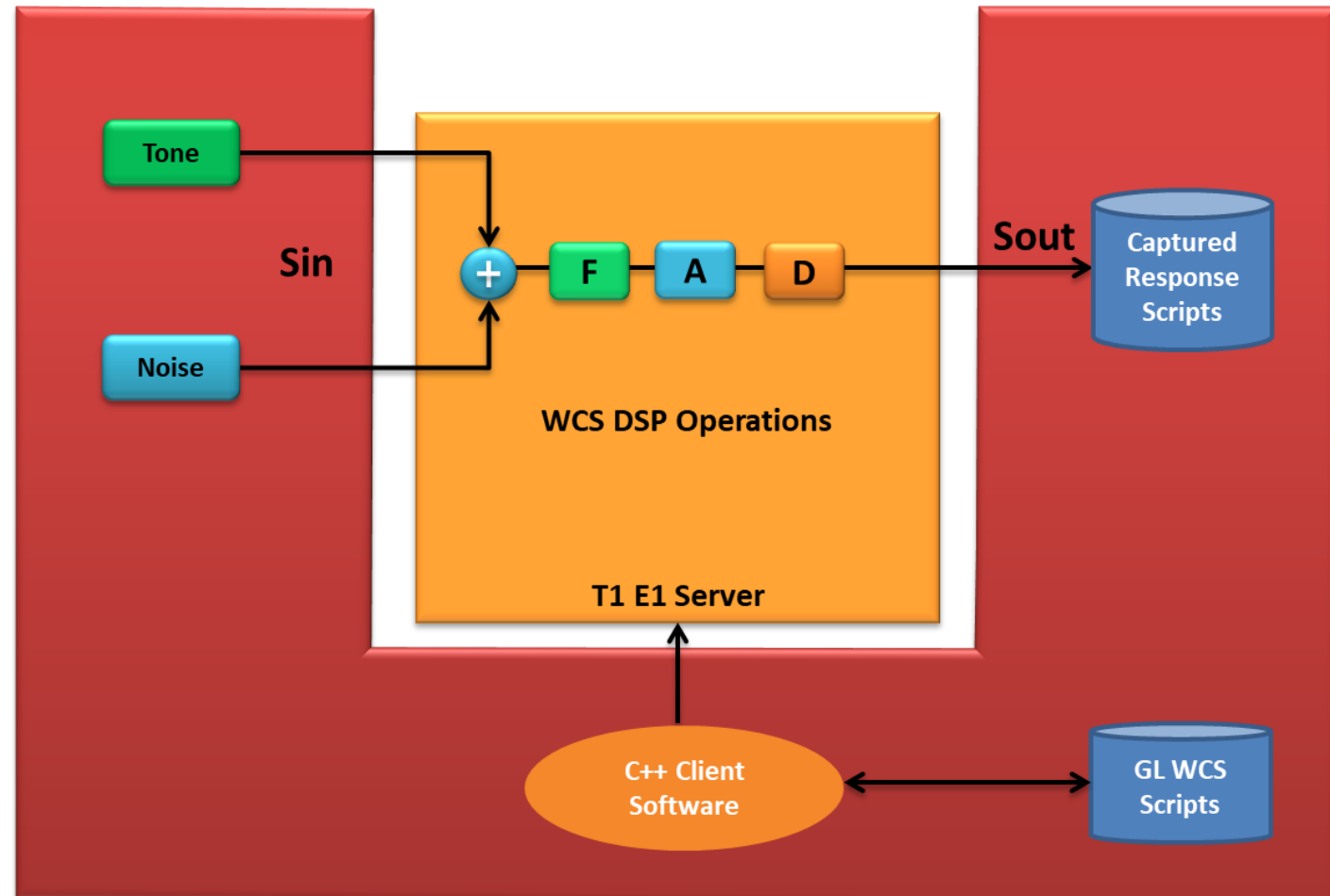
The screenshot shows a window titled "e1dsp\_delay.gls - GLClient" with a menu bar (File, Edit, View, Connect, Script, Log, User, Help) and a toolbar. The main area contains two text boxes with blue text logs. The top box shows task 13 and 14 being queued, started, and then processing 40,000 frames. The bottom box shows a "get response;" command followed by a series of task definitions with increasing delays (0 to 90 msec) and a "stop task \*;" command. The status bar at the bottom indicates "Ready" and "Ver 4 B NUM".

```
Task 13: Task 13 queued
tx(delay(rx(#1:8), 80 msec), #1:8) 5 sec;...
Task 14: Task 14 queued
tx(delay(rx(#1:9), 90 msec), #1:9) 5 sec;
Task 14: Task 14 started
Task 13: Task 13 started
Task 12: Task 12 started
Task 10: Task 10 started
Task 11: Task 11 started
Task 15: Task 15 started
Task 13: 40000 frames processed
Task 14: 40000 frames processed
Task 12: 40000 frames processed

get response;
tx(delay(rx(#1:1), 0 msec), #1:1) 500 sec;...
tx(delay(rx(#1:2), 10 msec), #1:2) 500 sec;...
tx(delay(rx(#1:3), 20 msec), #1:3) 5 sec;...
tx(delay(rx(#1:4), 30 msec), #1:4) 5 sec;...
tx(delay(rx(#1:5), 50 msec), #1:5) 5 sec;...
tx(delay(rx(#1:6), 60 msec), #1:6) 5 sec;...
tx(delay(rx(#1:7), 70 msec), #1:7) 5 sec;...
tx(delay(rx(#1:8), 80 msec), #1:8) 5 sec;...
tx(delay(rx(#1:9), 90 msec), #1:9) 5 sec;
stop task *;
//
//delay on 15 TS with same delay
```

# Dynamic Digital Signal Processing (DSP) (Module license #- XX631)

- Scripted DSP commands provide the ability to specify a sequence of digital signal processing steps to be performed on incoming and/or outgoing timeslots
  - The operations can be made dynamic or time-varying via schedules
  - Schedules are categorized into Time, Operators, Transition, and Value



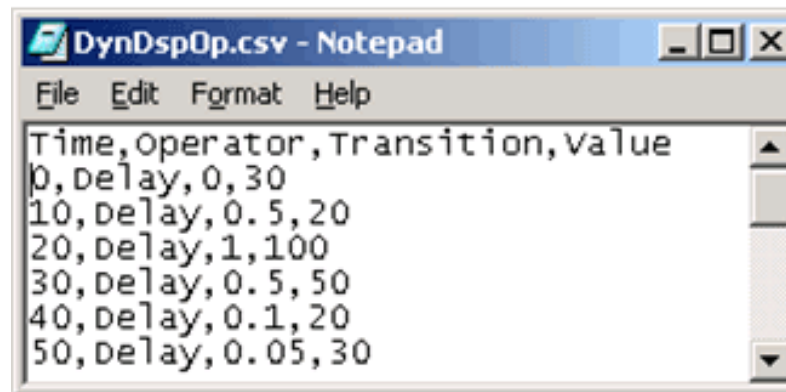
# Dynamic DSP Operations

- Offline Dynamic DSP Operations
  - Amplify ("AmplifyDspOp" - dynamic amplification)
  - Attenuate ("AttenDspOp " - dynamic attenuation)
  - Delay ("DelayDspOp" - dynamic delay)
  - Filter ("FiltDspOp" - dynamic filter models)
- Real-time Dynamic DSP Operations
  - Delay / Attenuate ("AttenDspOp " and "DelayDspOp")
  - Filter ("FiltDspOp")

# Example - Offline Delay Testing ("DelayDspOp")

## DelayDspOp WCS Test Script

```
set latency 4;  
set response 6;  
  
// (2) Delay  
outfile(delay(tone(1000 hz, -10 dbm), 10 msec), "WinClientServer\DynDspOp\  
DynDelayDspOp.ala") 60 sec cfg "WinClientServer\DynDspOp\DynDspOp.ini";
```



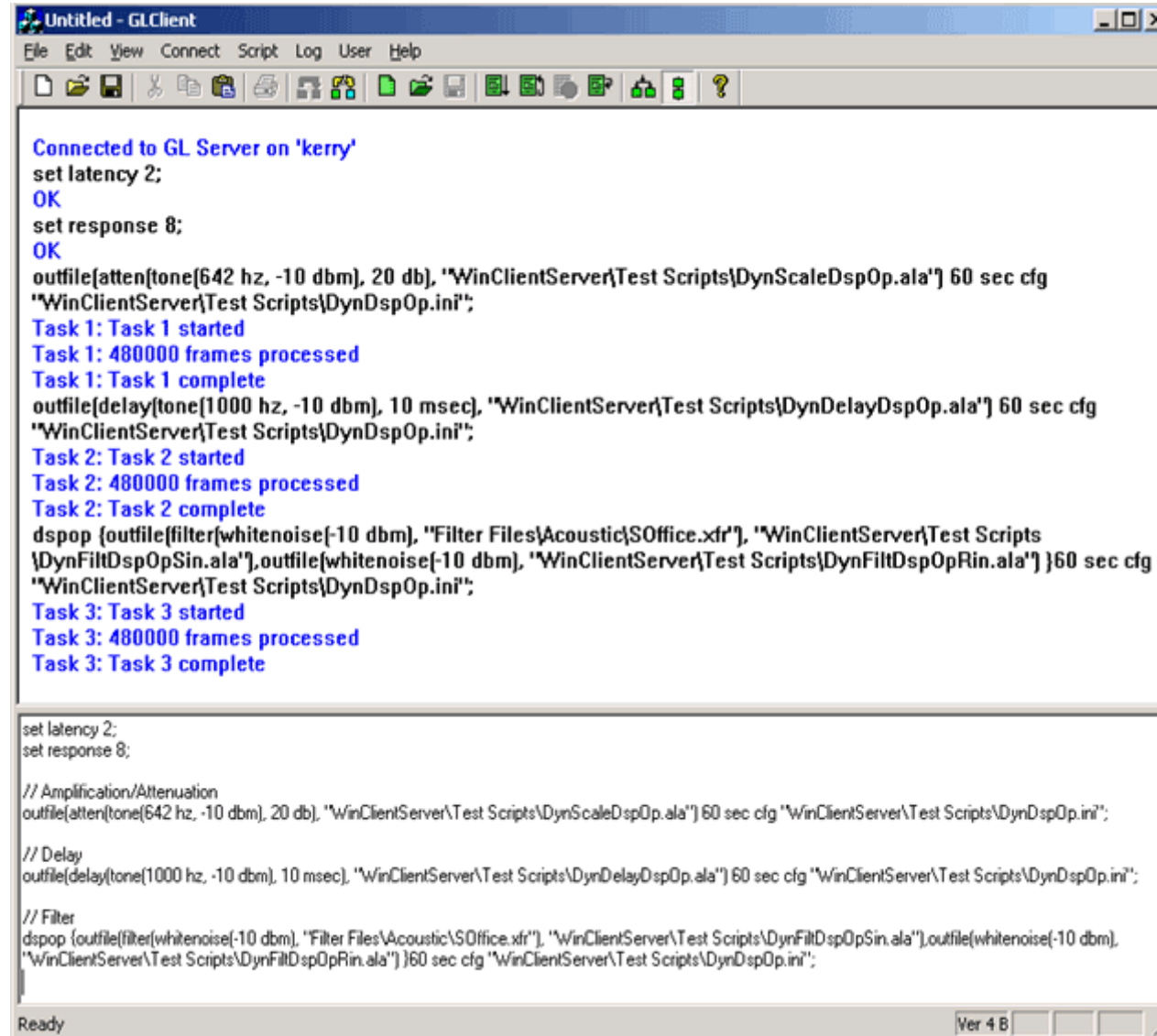
Time	Operator	Transition	Value
0	Delay	0	30
10	Delay	0.5	20
20	Delay	1	100
30	Delay	0.5	50
40	Delay	0.1	20
50	Delay	0.05	30

## Applicable Schedule

- Input tone is delayed as per the specified Time, Transition, and Values defined in the Schedule \*.csv file



# Dynamic Digital Signal Processing (DSP)



The screenshot shows a window titled "Untitled - GLClient" with a menu bar (File, Edit, View, Connect, Script, Log, User, Help) and a toolbar. The main area contains a terminal window with the following text:

```
Connected to GL Server on 'kerry'  
set latency 2;  
OK  
set response 8;  
OK  
outfile[atten(tone[642 hz, -10 dbm], 20 db), "WinClientServer\Test Scripts\DynScaleDspOp.ala"] 60 sec cfg  
"WinClientServer\Test Scripts\DynDspOp.ini";  
Task 1: Task 1 started  
Task 1: 480000 frames processed  
Task 1: Task 1 complete  
outfile[delay(tone[1000 hz, -10 dbm], 10 msec), "WinClientServer\Test Scripts\DynDelayDspOp.ala"] 60 sec cfg  
"WinClientServer\Test Scripts\DynDspOp.ini";  
Task 2: Task 2 started  
Task 2: 480000 frames processed  
Task 2: Task 2 complete  
dspop {outfile[filter(whitenoise[-10 dbm], "Filter Files\Acoustic\SOffice.xfr"), "WinClientServer\Test Scripts  
\DynFiltDspOpSin.ala"],outfile[whitenoise[-10 dbm], "WinClientServer\Test Scripts\DynFiltDspOpRin.ala"]} 60 sec cfg  
"WinClientServer\Test Scripts\DynDspOp.ini";  
Task 3: Task 3 started  
Task 3: 480000 frames processed  
Task 3: Task 3 complete
```

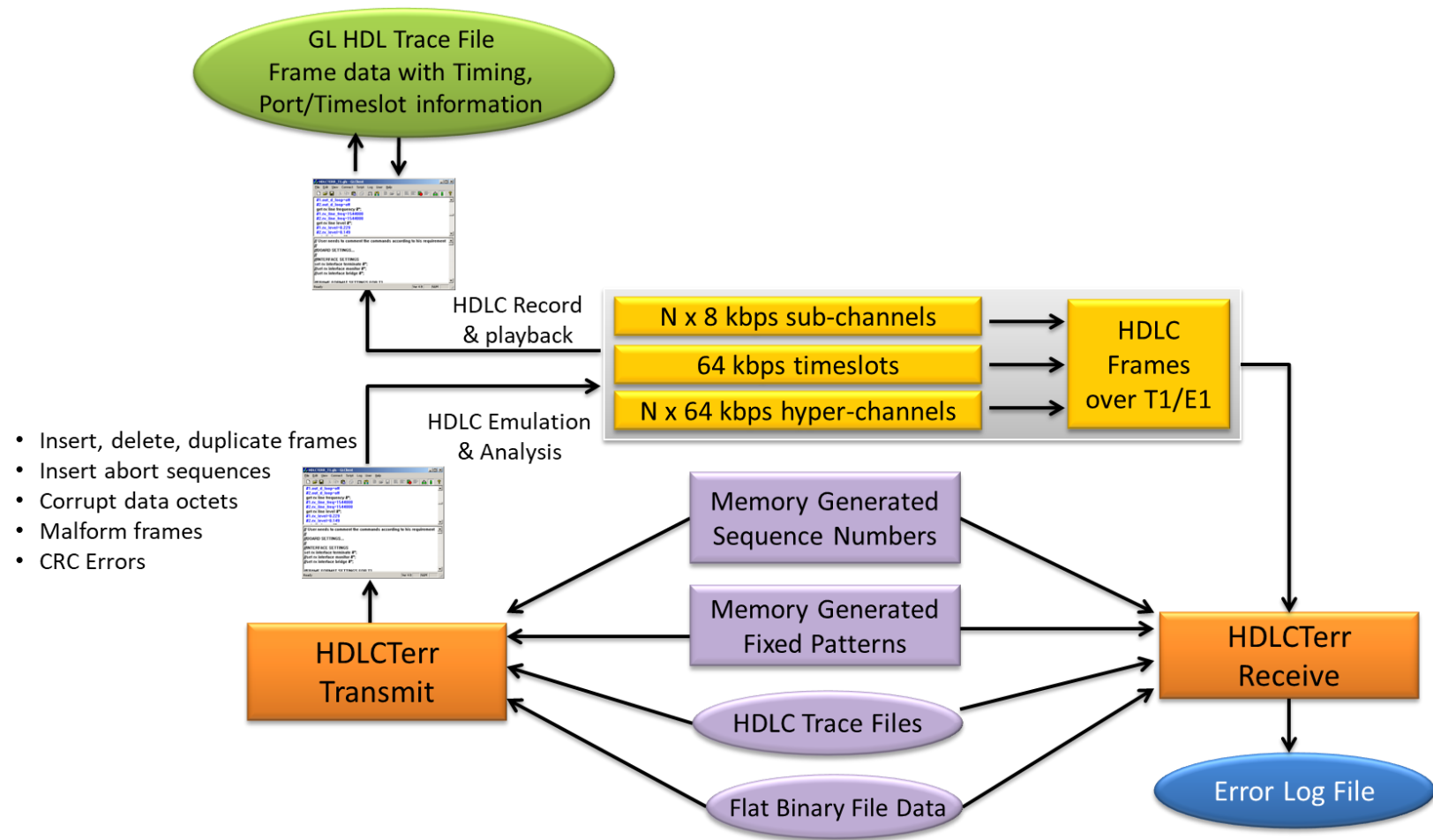
---

```
set latency 2;  
set response 8;  
  
// Amplification/Attenuation  
outfile[atten(tone[642 hz, -10 dbm], 20 db), "WinClientServer\Test Scripts\DynScaleDspOp.ala"] 60 sec cfg "WinClientServer\Test Scripts\DynDspOp.ini";  
  
// Delay  
outfile[delay(tone[1000 hz, -10 dbm], 10 msec), "WinClientServer\Test Scripts\DynDelayDspOp.ala"] 60 sec cfg "WinClientServer\Test Scripts\DynDspOp.ini";  
  
// Filter  
dspop {outfile[filter(whitenoise[-10 dbm], "Filter Files\Acoustic\SOffice.xfr"), "WinClientServer\Test Scripts\DynFiltDspOpSin.ala"],outfile[whitenoise[-10 dbm],  
"WinClientServer\Test Scripts\DynFiltDspOpRin.ala"]} 60 sec cfg "WinClientServer\Test Scripts\DynDspOp.ini";
```

Ready Ver 4 B

# HDLC Emulation and Analysis (Module License #- XX634)

- Offers high throughput file-based HDLC record and playback (support for various bandwidth over multiple links and option to speed up / slow down the transmission)
- Performs multi-channel HDLC emulation and analysis



# Overview

- The HDLCTerr module performs multi-channel HDLC emulation and analysis
- It permits frame error testing and transmission of memory generated sequences of fixed or variable length HDLC frames, GL \*.HDL Trace file frames, and various bandwidth streams
- The HDLCHpio module performs file-based HDLC record and playback actions
- It permits receive / transmit of HDLC streams of various bandwidth (hyper channel, timeslot, and multiple sub-channel streams per timeslot)

# Key Features

- Port, timeslot, subchannel translation
- Time preservation, speeding up and slowing down during playback
- Advanced performance support for multiprocessor computers
- Flexible transmit options
- Flexible receive options
- Extensive documentation
- Real-time counters
- Supports transmission and reception on Non contiguous timeslots
- Supports Octet and Bit Sync

# Impairments

- Various impairments can be introduced before frames are transmitted. Global impairments (effective for all the HDL streams) can be specified as well as impairments can be introduced per stream basis before frame transmission
- One can specify a limited number of impairments, set continuous impairment in each frame, or apply impairment to each Nth frame leaving some frames intact
- The following types of Impairments can affect an entire HDL frame:
  - Frame deletion
  - Frame insertion
  - Frame duplication

# Impairments (Contd.)

Impairments can also modify some octets in a frame at a certain offset and these include:

- Inserting octets
- Deleting octets
- Bitwise ANDing octets
- Bitwise Oring octets
- Bitwise XORing octets

In addition the following frame structure impairments can be introduced:

- CRC (FCS) errors
- Frame errors (non-integral number of octets between flags)
- Abort sequences

# Tx/Rx Performance of HDLC

- Allows transmission/reception of \*.HDL files located on the server and on client

## Sample Script for HDLC Capture / Playback

The screenshot displays two windows. The left window, titled 'Untitled - GLClient', shows a script with the following content:

```
run task "HdlcFuncE1:TxServerFile" using " 'hdlc_isdn\dcos
FLAGS 100" #1:1..3;
Task 1: Task 1 started
run task "HdlcFuncE1:RxClientFile" using "'c:\test.hdl' 1000
100000" #1:1..3;
Task 2: Task 2 started
Task 1: Task 1 terminated

run task "HdlcFuncE1:TxServerFile" using " 'hdlc_isdn\
700 FLAGS 100" #1:1..3;
run task "HdlcFuncE1:RxClientFile" using "'c:\test.hdl' 1
100000" #1:1..3; |
```

The right window, titled 'HDLC Protocol Analysis LAPD', shows a table of captured frames:

Dev	TS...	Su...	Frame#	TIME (Relative)	Len	Error	C/R	SAPI	TEI	CTL	P/F	N(S)	N(R)	FUNC
✓ 2	1-3		692	00:00:03.222458	11		Co...	0	0	Inform...	0	96	75	
✓ 2	1-3		693	00:00:03.227083	11		Co...	0	0	Inform...	0	97	75	
✓ 2	1-3		694	00:00:03.231708	6		Co...	0	0	Super...	0		97	RR
✓ 2	1-3		695	00:00:03.236125	6		Co...	0	0	Super...	0		98	RR
✓ 2	1-3		696	00:00:03.240583	11		Res...	0	0	Inform...	0	75	98	
✓ 2	1-3		697	00:00:03.245208	6		Res...	0	0	Super...	0		76	RR
✓ 2	1-3		698	00:00:03.249625	11		Co...	0	0	Inform...	0	98	76	
✓ 2	1-3		699	00:00:03.254250	11		Co...	0	0	Inform...	0	99	76	

Below the table, the software displays details for frame 692:

```
Card2 TimeSlots=1-3 Frame=692 at 00:00:03.222458 OK Len=11
HDLC Frame Data + FCS
===== LAPD Layer =====
C/R = .....0. Command(User), Response(Netw
SAPI = 000000.. (0)
TEI = 0000000. (0)
Ctl = .....0 Information
N(S) = 1100000. (96)
P = .....0 (0)
N(R) = .....0 (75)
```

At the bottom, a hex dump of the frame data is shown:

```
Hex Dump of the Frame Data
+-----+-----+-----+-----+-----+-----+-----+-----+
00 01 C0 96 08 02 AF 02 01 D4 2A
```

The status bar at the bottom indicates: 'Running, Utilization 0.22%' and 'Captured 700 frames'.

# Sample Script for Multi-Channel File-based HDLC Record and Playback Actions

```

HDLCHPIO_E1.gls - GLClient
File Edit View Connect Script Log User Help

Connected to GL Server on 'poornima'
run task "HdlchpioE1:TxRx" #1..2:1;
Task 1: Task 1 started
inform task * "Rx 'C:\rev\Ts.hdl' 500000 CONT TS ";
OK
inform task * "Tx 'C:\tx\10sec_1flags.hdl' EOF TS FLAGS 1";
OK
inform task * "start";
OK

//To check transmission and reception operation on single T
run task "HdlchpioE1:TxRx" #1..2:1;
inform task * "Rx 'C:\rev\Ts.hdl' 500000 CONT TS ";
inform task * "Tx 'C:\tx\10sec_1flags.hdl' EOF TS FLAGS 1";
inform task * "start";
query task 1;
stop task*;

Ready
    
```

**HDLC Protocol Analysis LAPD**

File View Capture Statistics Database Configure Help

0 GoTo

Dev	TSlot	Sub...	Frame#	TIME (Relative)	Len	Error	C/R	SAPI	TEI	CTL	P/F	N(S)	N(R)	FUNC
✓ 2	1		0	00:00:00.000000	166		Com...	29	30	Supervis...	1		81	RNR
✓ 1	1		1	00:00:00.000250	166		Com...	29	30	Supervis...	1		81	RNR
✓ 2	1		2	00:00:00.021125	166		Com...	47	116	Informati...	0	126	19	
✓ 1	1		3	00:00:00.021375	166		Com...	47	116	Informati...	0	126	19	
✓ 2	1		4	00:00:00.042375	167		Com...	63	37	Informati...	1	64	54	
✓ 1	1		5	00:00:00.042625	167		Com...	63	37	Informati...	1	64	54	
✓ 2	1		6	00:00:00.063750	167		Resp...	35	106	Unnumb...	0			Reserved
✓ 1	1		7	00:00:00.064000	167		Resp...	35	106	Unnumb...	0			Reserved

Card2 TimeSlot=1 Frame=0 at 00:00:00.000000 OK Len=166

HDLC Frame Data + FCS

```

----- LAPD Layer -----
C/R           = .....0. Command(User), Response(Network)
SAPI          = 011101.. (29)
TEI           = 0011110. (30)
Ctl           = .....01 Supervisory
Supervisory Function = ....01.. RNR
P/F           = .....1 (1)
N(R)         = 1010001. (81)
    
```

Hex Dump of the Frame Data

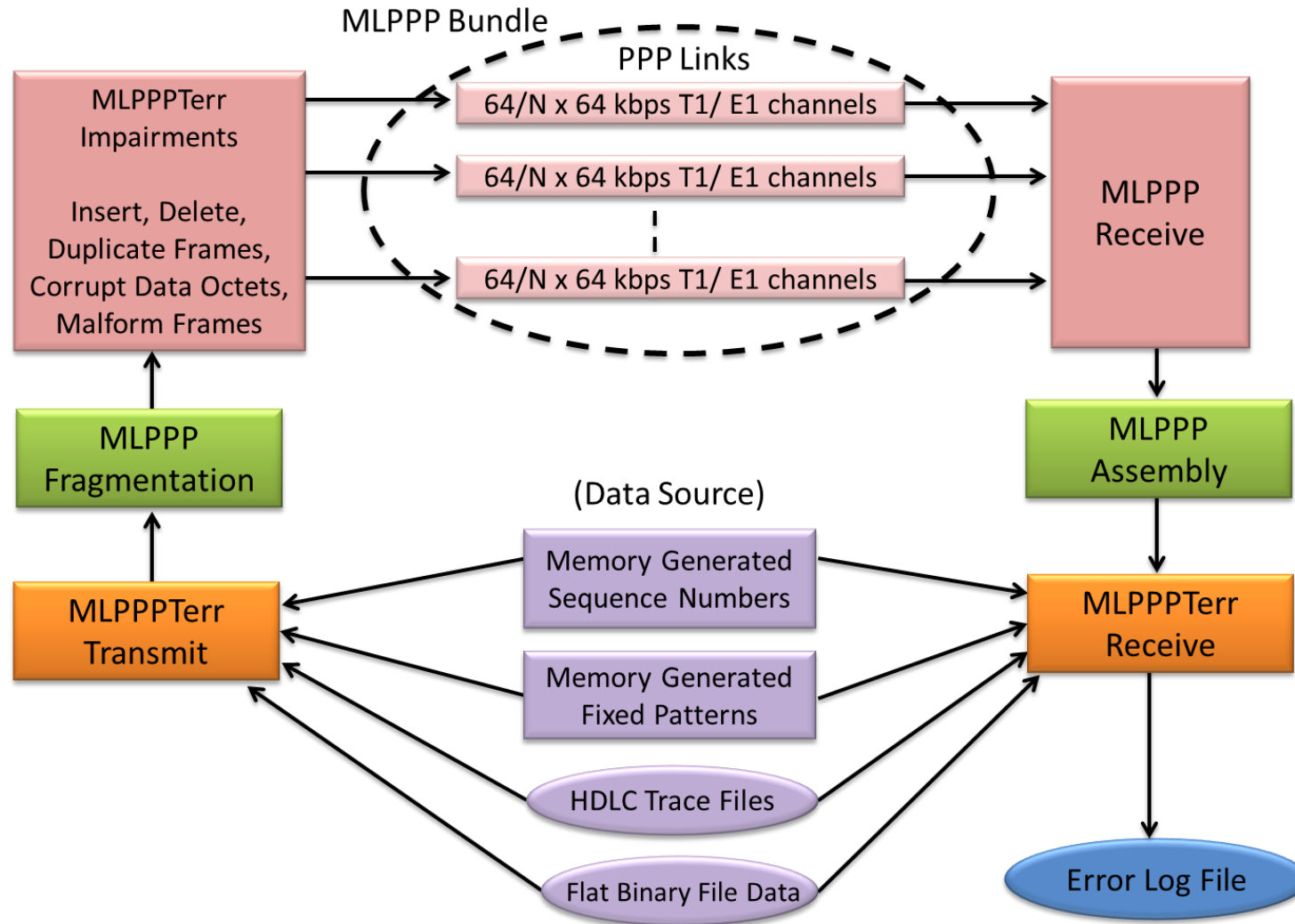
```

+-----+-----+-----+-----+-----+-----+-----+-----+
74 3D 55 A3 24 D3 E7 36 D8 B2 92 9D AB 60 54 2E   t=Uÿ$0ç60²' |«<'T.
AD 7B E7 10 7A FA F1 9D 12 B5 7B 0B E4 28 A4 4C   -{ç zúñ | µ{ ä(ML
2C 73 96 32 7F A6 BA D7 A4 3C 28 3C A6 E5 62 30   ,s|2||²x«(<|âb0
27 89 41 74 F7 89 2F 1C E6 48 97 38 D6 EA 9A B4   '|At=|/ æH|80è|'
    
```

Running, Utilization 0.23%      C:\Temp.Hdl      Captured 1280 frames



# MLPPP Emulation and Analysis (Module License #- XX635, XX636, XX637)



# Sample Script for Emulation and Analysis of MLPPP Frames

```
Untitled - GLClient
File Edit View Connect Script Log User Help
run task "MLPPPTerrE1:Tx";
Task 1: Task 1 started
inform task 1 "TX: FRAMES 1000 SEQNUM MSB4 FIXLEN 2048";
OK
inform task 1 "TS #1:1..31 MRU 256";
OK
run task "MLPPPTerrE1:Rx";
Task 2: Task 2 started
inform task 2 "RX: FRAMES 1000 SEQNUM MSB4 LOG 'D:\log\frame1.log' RECORD 'D:\log\frame1.hdl' FIXLEN 2048";
OK
inform task 2 "TS #2:1..31 MRU 256";

run task "MLPPPTerrE1:Tx";
inform task 1 "TX: FRAMES 1000 SEQNUM MSB4 FIXLEN 2048";
inform task 1 "TS #1:1..31 MRU 256";
//inform task 1 "HC #1:1..15 MRU 256";
//inform task 1 "HC #1:16..31 MRU 256";
//inform task 1 "SC #1:1..31:1..7 MRU 256";

run task "MLPPPTerrE1:Rx";
inform task 2 "RX: FRAMES 1000 SEQNUM MSB4 LOG 'D:\log\frame1.log' RECORD 'D:\log\frame1.hdl' FIXLEN 2048";
inform task 2 "TS #2:1..31 MRU 256";
//inform task 2 "HC #2:1..15 MRU 256";

Ready Ver 3 B CAP NUM
```

PPP Protocol Analysis PPP

Dev	TS...	Su...	Frame#	TIME (Relative)	Len	Error	Layer3Protocol	LCP C...	Seq No	Class	IPCP C...	Layer3Pr
✓ 2	5		27	00:00:00.033875	266		ML PPP		4259	0		Padding
✓ 2	3		28	00:00:00.034875	266		ML PPP		4260	0		Padding
✓ 2	20		29	00:00:00.037500	266		ML PPP		4261	0		Padding
✓ 2	25		30	00:00:00.041375	266		ML PPP		4262	0		Padding
✓ 2	26		31	00:00:00.042250	266		ML PPP		4263	0		Padding
✓ 2	15		32	00:00:00.043250	266		ML PPP		4264	0		Padding
✓ 2	18		33	00:00:00.045750	12		ML PPP		4265	0		Padding

Card2 TimeSlot=5 Frame=27 at 00:00:00.033875 OK Len=266

HDLC Frame Data + FCS

```
===== PPP Link Layer =====
Address          = 11111111 (255)
Ctl              = 00000011 (3)
Protocol         = 00000000 00111101 ML PPP
===== ML PPP Layer =====
Beginning Fragment = 0..... No
Ending Fragment   = .0..... No
Mlppp Class       = ..0000.. (0)
Sequence Number(Long) = 4259 (x0010A3)
```

Hex Dump of the Frame Data

```
+-----+-----+-----+-----+-----+-----+-----+-----+
FF 03 00 3D 00 00 10 A3 01 D9 00 00 01 D9 00 00  ȳ = ε ũ ũ
01 D9 00 00 01 D9 00 00 01 D9 00 00 01 D9 00 00  ũ ũ ũ ũ
01 D9 00 00 01 D9 00 00 01 D9 00 00 01 D9 00 00  ũ ũ ũ ũ
01 D9 00 00 01 D9 00 00 01 D9 00 00 01 D9 00 00  ũ ũ ũ ũ
```

Running. Utilization 2.57% C:\Temp.Hdl Captured 4768 frames

- Simulates MC-MLPPP and PPP protocols over T1 E1 links
- Offers GUI as well as command line interfaces

# Tx/Rx Performance of TRAU

## (Module license #- XX645)

Sample script for Capture/Playback of TRAU traffic

```

E1_Trau wcs scripts.gls - GLClient
File Edit View Connect Script Log User Help

Connected to GL Server on 'harsha'
run task "TrauFuncE1:TxFile" using "'Trau\AMR_Cisco.HDL' EOF 16K SC:C0 "#1:1;
Task 3: Task 3 started
Task 3: Task 3 terminated

//Transmit trau frames from AMR_Cisco.HDL over port-1 TS-1, SC 1-2 till end of file
run task "TrauFuncE1:TxFile" using "'Trau\AMR_Cisco.HDL' EOF 16K SC:C0 "#1:1;

//Receive trau frames to Rx.hdl file in port-2 TS-1,SC 1-2
run task "TrauFuncE1:RxFile" using "'Trau\Rx.hdl' 4000000 100000 16K SC:C0
UPLINK" #2:1;

Ready
Ver 3 B CAP NUM
  
```

Dev	TS...	Su...	Frame#	TIME (Difference)	Len	Error	TRAU F...	TRAU F...	Fra...	Spe...	Tim...	CRC	RIF	AM...
✓ 2	1	1-2	192	00:00:00.020000	40		Uplink (...)	Adaptiv...	Valid	Go...	No...	Vali...	Indi...	Cod...
✓ 2	1	1-2	193	00:00:00.020000	40		Uplink (...)	Adaptiv...	Valid	Go...	No...	Vali...	Indi...	Cod...
✓ 2	1	1-2	194	00:00:00.020000	40		Uplink (...)	Adaptiv...	Valid	Go...	No...	Vali...	Indi...	Cod...
✓ 2	1	1-2	195	00:00:00.020000	40		Uplink (...)	Adaptiv...	Valid	Go...	No...	Vali...	Indi...	Cod...
✓ 2	1	1-2	196	00:00:00.020000	40		Uplink (...)	Adaptiv...	Valid	Go...	No...	Vali...	Indi...	Cod...
✓ 2	1	1-2	197	00:00:00.020000	40		Uplink (...)	Adaptiv...	Valid	Go...	No...	Vali...	Indi...	Cod...
✓ 2	1	1-2	198	00:00:00.020000	40		Uplink (...)	Adaptiv...	Valid	Go...	No...	Vali...	Indi...	Cod...
✓ 2	1	1-2	199	00:00:00.020000	40		Uplink (...)	Adaptiv...	Valid	Go...	No...	Vali...	Indi...	Cod...

Card2 TimeSlot=1 SubChannels=1-2 Frame=192 at 11:29:18.067250 OK Len=40

HDL Frame Data + FCS

```

===== TRAU Layer =====
Frame Sync                = Valid Frame Sync (00000000000000011111111111111111)
Frame Direction           = Uplink (User)
Frame Type (Full Rate, 16kbps, C1-C5) = .00110... Adaptive Multi-Rate Narrow Band Codec (AMR-1)
Time Alignment (C6-C11) for TAC_AMR   = .....00 0000.... No change in frame timing
Req or Ind Flag-RIF (C12) for Uplink   = .....0... Indication (Codec Mode)
Configuration Protocol (C14-C16)       = .....00 .0..... BTS does not support TFO or TFO is
Message No (C17-C18)                 = .....00.... BTS does not support TFO or TFO is disabled
  
```

Hex Dump of the Frame Data

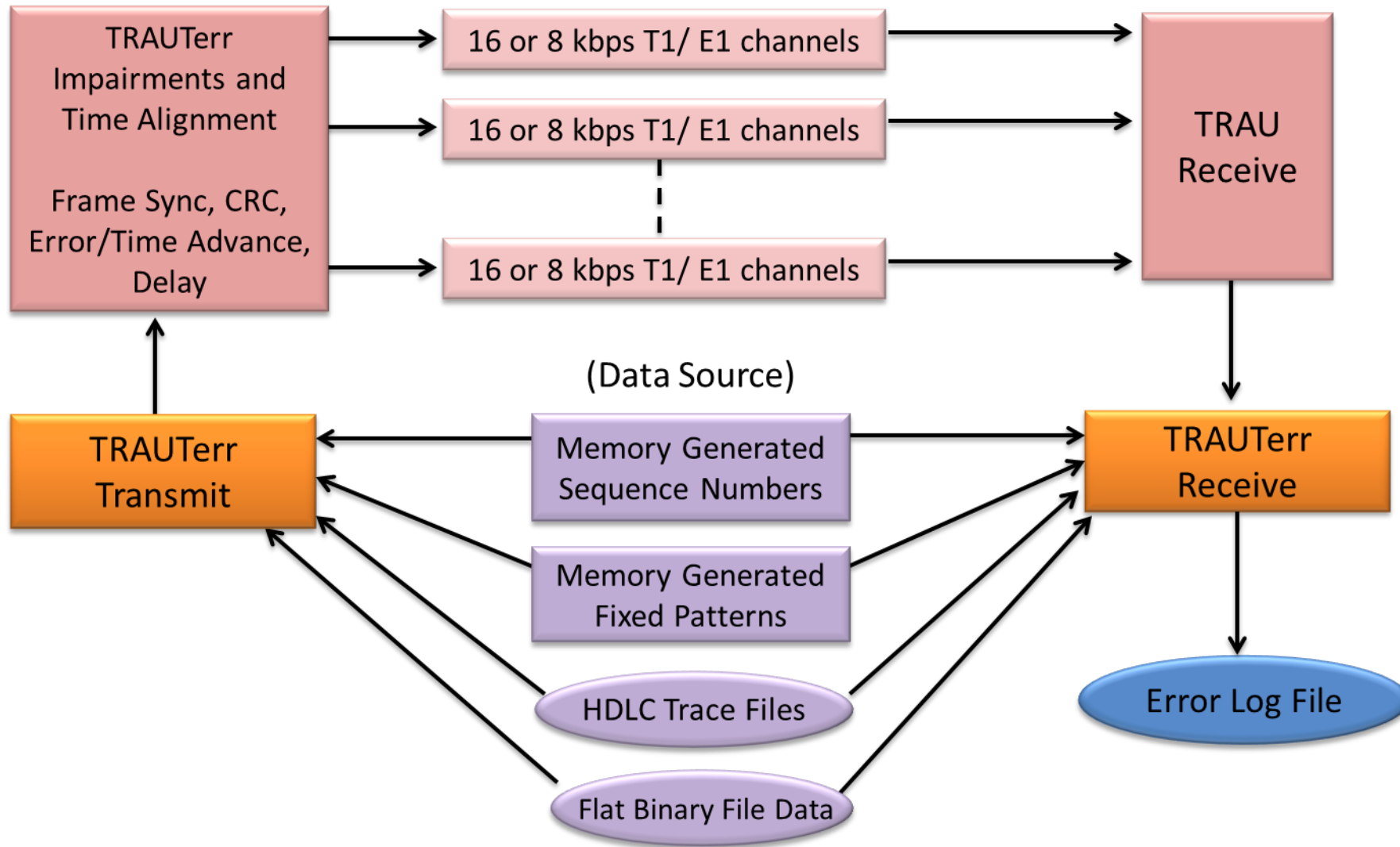
```

00 00 98 00 83 BF FF FF FF F5 B8 78 97 E9 C1 FF      | 00000000000000011111111111111111
F4 54 C4 39 EE 22 98 B0 91 85 92 46 F2 0A 9C 51      | 00000000000000000000000000000000
BF FF FF FF FF FF FF FF                          | 00000000000000000000000000000000
  
```

Running. Utilization 0.15% C:\Temp.HDL Captured 200 frames

- Receives and transmits TRAU frames in .HDL file format

# TRAU Emulation and Analysis (Module license #- XX646)





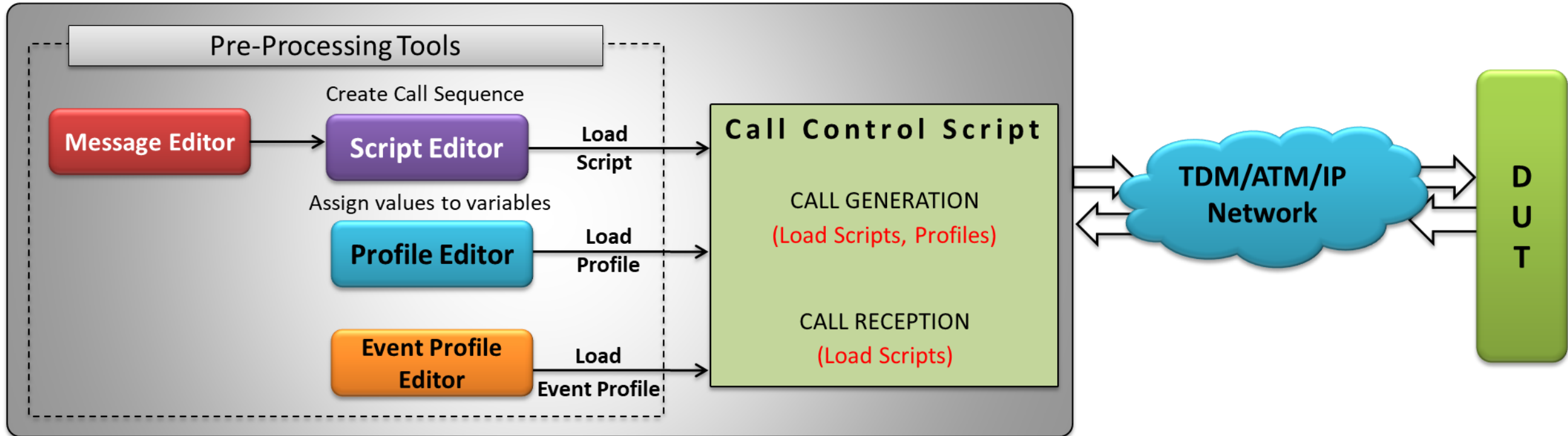
# MAPS™ – Script Based Protocol Simulation

## ISDN, ISUP, MAP, MLPPP, CAS, and GSM Protocols

- Uses client-server technique and provides a scripted protocol simulation and conformance testing platform
- Supports simulation of a variety of protocols such as SS7, ISDN, GSM, CAS, and MLPPP over T1 E1 network
- This message automation tool covers solutions for both protocol simulation and protocol analysis
- Includes various test plans and test cases to support the testing of a required real-time scenario
- Provides the unlimited ability to edit messages and control scenarios (message sequences) "Message sequences" are generated through scripts

# MAPS™ Working Principle

## Message Automation and Protocol Simulation (MAPS™)



# Call Generation and Reception

**Call Generation - Untitled**

Sr No	Script Name	Profile	Call Info	Script Execution	Status	Events	Events Profile	Result	Total Iterations	Completed Iterations
1	PlaceCall.gls	TS1.xm	1	Start	Call Released	None		Pass	1	1

Buttons: Add, Delete, Insert, Start, Abort, Refresh, Start All, Abort All

MAPS      DUT

```

      SETUP → 17:49:08.625000
      CALL PROCEEDING ← 17:49:09.468000
      ALERTING ← 17:49:09.484000
      CONNECT ← 17:49:19.515000
      CONNECT ACKNOWLEDGE → 17:49:19.640000
      DISCONNECT ← 17:50:20.140000
          
```

```

===== Q.93x Layer
Protocol Discriminator
Call Reference Length
Call Reference Value
Call Reference Flag
Message Type
Bearer capability
IEI Bearer Capability
IE Bearer Capability L
Information Transfer C
Coding Standard
Oct 3 Extension Bit (0
Information Transfer P
Transfer Mode
Oct 4 Extension Bit (0
User Information Layer
          
```

Scripts | Message Sequence | Event Config

**Call Reception**

Sr No	Script Name	Call Info	Script Execution	Status	Events	Events Profile	Results
1	AnswerCall.gls		Abort	Call Active	Disconnect		Pass

Buttons: Abort, Auto Trash, Trash

MAPS      DUT

```

      SETUP ← 17:53:43.359000
      CALL PROCEEDING → 17:53:43.3
      ALERTING → 17:53:43.3
      CONNECT → 17:53:53.3
      CONNECT ACKNOWLEDGE ← 17:53:53.6
          
```

```

===== Q.93x Layer 3 Layer =====
Protocol Discriminator
Call Reference Length
Call Reference Value
Call Reference Flag
          
```

Scripts | Message Sequence | Event Config

**GLServer**

```

File Edit View Setup Help
[Icons]
Connected: client #756 at 127.0.0.1
Connected: client #760 at 192.168.1.63
756: run task "IsdnServerT1:Sim";
756: inform task 1 "START TS #1:23 SUBSCRIBER";
756: inform task 1 "STARTTRAFFIC 17000 14000 192.168.1.63";
Connected: client #1208 at 127.0.0.1
Connected: client #1224 at 192.168.1.63
1208: run task "IsdnServerT1:Sim";
1208: inform task 2 "START TS #2:23 SWITCH";
1208: inform task 2 "STARTTRAFFIC 9000 8000 192.168.1.63";
Ready
          
```



# Multi-Link FrameRelay Emulation (Module license # - XX655)

- WCS Multi-Link Frame Relay is a CLI based client application
- Activate/deactivate the individual bundle links in the MFR bundle
- Create/delete the virtual channels on the links
- Sends MFR frames with or without impairments
- Receives MFR frames
- Generates and receives traffic using source and sink types
  - Sequence numbers
  - Hex string frame
  - Binary flat files
  - HDL trace files (GL's proprietary file format)
- Various impairments can be applied on each individual FR links and virtual channels

# Multi-Link Frame Relay Emulation (Module license # - XX655)

Sample script for Transmission and Reception of MFR Frames

## Multi-Link Frame Relay Emulation (Module license # - XX655)

```
FrameRelay_E1.gls - GLClient
File Edit View Connect Script Log User Help
[Icons]
OK
inform task 3 "CREATE VC HC #1:1..31 DLCI 1 FRAG FORMAT END TO END FRAGSIZE 256";
OK
inform task 3 "Tx: HC #1:1..31 DLCI 1 CONT FIXLEN 1500 SEQNUM MSB4";
OK
inform task 3 "START TX HC #1:1..31 DLCI 1";
OK
query task 3;
Task 3:
Simulation=Frame Relay, Total FR Links=1, Active FR Links=1, Selected Link=1:1..31, Link Status=Active
===== HDLC Stats =====, Tx Octets=9159516, Tx Frames=35502, Rx Octets=0, Rx Frames=0, Tx Over
Runs=0, Rx Over/Under Runs=0, CRC Error Count=0,
===== Virtual Channel Stats =====, Number of VC's on FR Link: '1:1..31'=1,
VC 1, DLCI=1, Tx Frames=5917, Tx Frags=35502, Rx Frames=0, Rx Frags=0, Lost Frags=0, Received c
Matched count=0, Modified count=0, Inserted count=0, Deleted count=0
OK
//I here should be fragmentation with B=1,E=U for first fragment,
//B=0,E=0 for in between fragments and B=0, E=1 for last fragment.

run task "MFREmulatorE1:TxRx";
inform task 1 "SIMULATION FR";
inform task 1 "HC #1:1..31 FLAGS 100";
//inform task 1 "TS #1:1..31 FLAGS 100";
//inform task 1 "SC #1:1..31:1..8 FLAGS 100";
inform task 1 "ACTIVATE HC #1:1..31";
//inform task 1 "ACTIVATE TS #1:1..31";
//inform task 1 "ACTIVATE SC #1:1..31:1..8";
inform task 1 "CREATE VC HC #1:1..31 DLCI 1 FRAG FORMAT END TO END FRAGSIZE 500";
//inform task 1 "CREATE VC TS #1:1..31 DLCI 1 FRAG FORMAT END TO END FRAGSIZE 500";
//inform task 1 "CREATE VC SC #1:1..31:1..8 DLCI 1 FRAG FORMAT END TO END FRAGSIZE 500";
inform task 1 "Tx: HC #1:1..31 DLCI 1 FRAMES 10 FIXLEN 1500 SEQNUM MSB4";
//inform task 1 "Tx: TS #1:1..31 DLCI 1 FRAMES 10 FIXLEN 1500 SEQNUM MSB4";
//inform task 1 "Tx: SC #1:1..31:1..8 DLCI 1 FRAMES 10 FIXLEN 1500 SEQNUM MSB4";

Ready Ver 4 B
```

Frame Relay Protocol Analysis LAPP

File View Capture Statistics Database Call Detail Records Configure Help

Dev	TS...	Su...	Frame#	TIME (Relative)	Len	DLCI	DE	BECN	FECN	CTL	NLPID	Sequenc...
✓ 2	1-31		0	-00:00:00.004403	264	1	0	0	0	Unnu...	FRF.12 Fragme...	0
✓ 2	1-31		1	-00:00:00.002935	264	1	0	0	0	Unnu...	FRF.12 Fragme...	1
✓ 2	1-31		2	-00:00:00.001467	264	1	0	0	0	Unnu...	FRF.12 Fragme...	2
✓ 2	1-31		3	00:00:00.000000	264	1	0	0	0	Unnu...	FRF.12 Fragme...	3
✓ 2	1-31		4	00:00:00.001467	264	1	0	0	0	Unnu...	FRF.12 Fragme...	4
✓ 2	1-31		5	00:00:00.002935	228	1	0	0	0	Unnu...	FRF.12 Fragme...	5
✓ 2	1-31		6	00:00:00.004258	264	1	0	0	0	Unnu...	FRF.12 Fragme...	6

Card2 TimeSlots=1-31 Frame=0 at -00:00:00.004403 OK Len=264

HDLC Frame Data + FCS

===== LAPP Layer =====

EA = .....0 (0)  
C/R = .....0. Command(User), Response(Network)  
DLCI = 1 (000000... 0001....)  
EA = .....1 (1)  
DE = .....0. (0)  
BECN = .....0.. (0)  
FECN = .....0... (0)

Hex Dump of the Frame Data

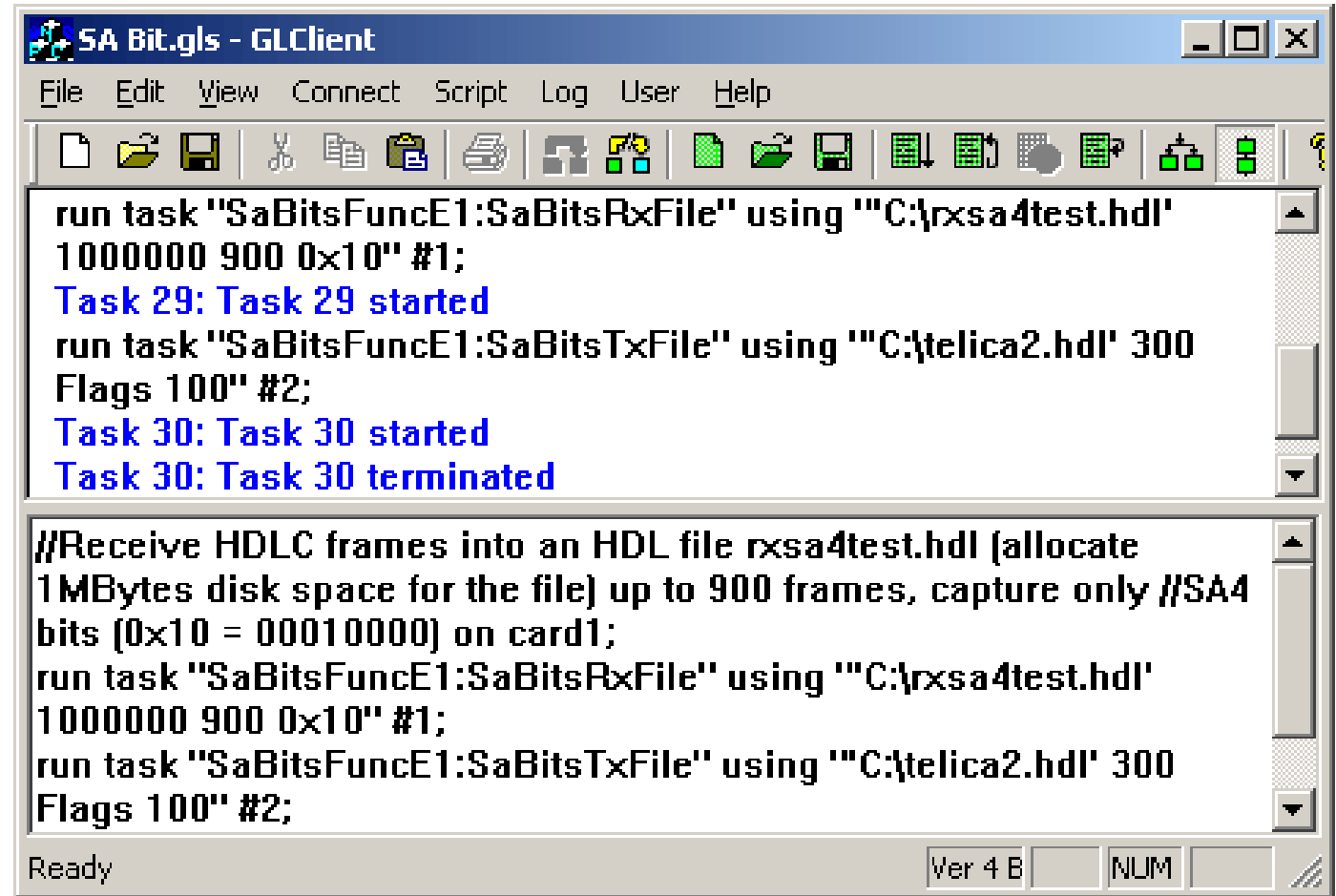
```
+-----+-----+-----+-----+
00 11 03 B1 80 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Running. Utilization 21.39% C:\Temp.Hdl Captured 64186 frames

# File based Record/Playback over FDL (Module license #- XX660)

Sample script for Transmission and Reception of SA HDLC frames from an HDL file

- Receives / Transmits SA HDLC frames in HDL files



The screenshot shows the 'SA Bit.gls - GLClient' application window. The menu bar includes File, Edit, View, Connect, Script, Log, User, and Help. The toolbar contains various icons for file operations and system functions. The main text area displays a script with the following content:

```
run task "SaBitsFuncE1:SaBitsRxFile" using "'C:\rxsa4test.hdl'  
1000000 900 0x10" #1;  
Task 29: Task 29 started  
run task "SaBitsFuncE1:SaBitsTxFile" using "'C:\telica2.hdl' 300  
Flags 100" #2;  
Task 30: Task 30 started  
Task 30: Task 30 terminated
```

Below the script, there is a comment and a second set of commands:

```
//Receive HDLC frames into an HDL file rxsa4test.hdl [allocate  
1MBytes disk space for the file) up to 900 frames, capture only //SA4  
bits (0x10 = 00010000) on card1;  
run task "SaBitsFuncE1:SaBitsRxFile" using "'C:\rxsa4test.hdl'  
1000000 900 0x10" #1;  
run task "SaBitsFuncE1:SaBitsTxFile" using "'C:\telica2.hdl' 300  
Flags 100" #2;
```

The status bar at the bottom indicates 'Ready' and 'Ver 4 B NUM'.

# File based Record/Playback over FDL (Module license #- XX660)

Sample script for transmission and reception of HDLC frames and signals over FDL

```
FDL.gls - GLClient
File Edit View Connect Script Log User Help
run task "FdIFuncT1:FdITxFile" using "'fdl
\mix2.fdl' CONT FLAGS 100" #1 ;
Task 2: Task 2 started
run task "FdIFuncT1:FdITxFile" using "'fdl
\mix2.fdl' CONT FLAGS 100" #1 ;
Ready Ver 3 B
```

FDL Analysis

Msg #	Time	Length	Error	Address	Ctrl	From	Type	Information
0	0:00:05.966000	75		x8108	x08 AT&T	CI	1-hour Enhanced Performanc...	x25084155-67C20001-00020003-00040005-3...
1	0:00:06.316000	37		x8108	x08 AT&T	CI	1-hour Performance Data	x25084155-40C20001-00020003-33000000-0...
2	0:00:06.590000	8		xC308	x08 AT&T	NI	Maintenance Request	x22414940
3	0:00:06.806000	225		x8108	x08 AT&T	CI	24-hour Enhanced Performan...	x25084155-6CC20001-00020003-00040005-3...
4	0:00:07.456000	8		x8108	x08 AT&T	CI	Maintenance Request	x22414964
5	0:00:07.672000	213		x8108	x08 AT&T	CI	24-hour Performance Data	x25084155-64C20001-00020003-33000000-0...
6	0:00:08.298000	8		xC308	x08 AT&T	NI	Maintenance Request	x22414901
7	0:00:08.514000	10		x8108	x08 AT&T	CI	Confirmation	x22084155-01C2
8	0:00:08.734000	10		x8108	x08 AT&T	CI	Confirmation	x22084155-4200
9	0:00:08.954000	8		xC308	x08 AT&T	NI	Maintenance Request	x22414902
10	0:00:09.170000	21		x8108	x08 AT&T	CI	Enhanced Configuration Data	x25084155-E7C20201-00020003-00040000-00...
11	0:00:09.414000	12		x8108	x08 AT&T	CI	Expand ESE Response	x25084155-42C21224

Card2 Frame=0 at 0:00:05.966000 OK Len=75  
LAPB Information  
Address = x81  
Control = x08  
Message-oriented  
RESPONSE 1-Hour Enhanced Performance Data Message  
CMD = x25  
Status = x41 Unknown or invalid request  
Originating Unit = U (x41)  
Target Unit = g (x55)  
PROJECT = 100 / Card Enhanced 1-hour Performance Data

Running, Utilization 15.343% C:\Temp\Fdl Captured 362 frames Errors 1 CRC, 0 Frame

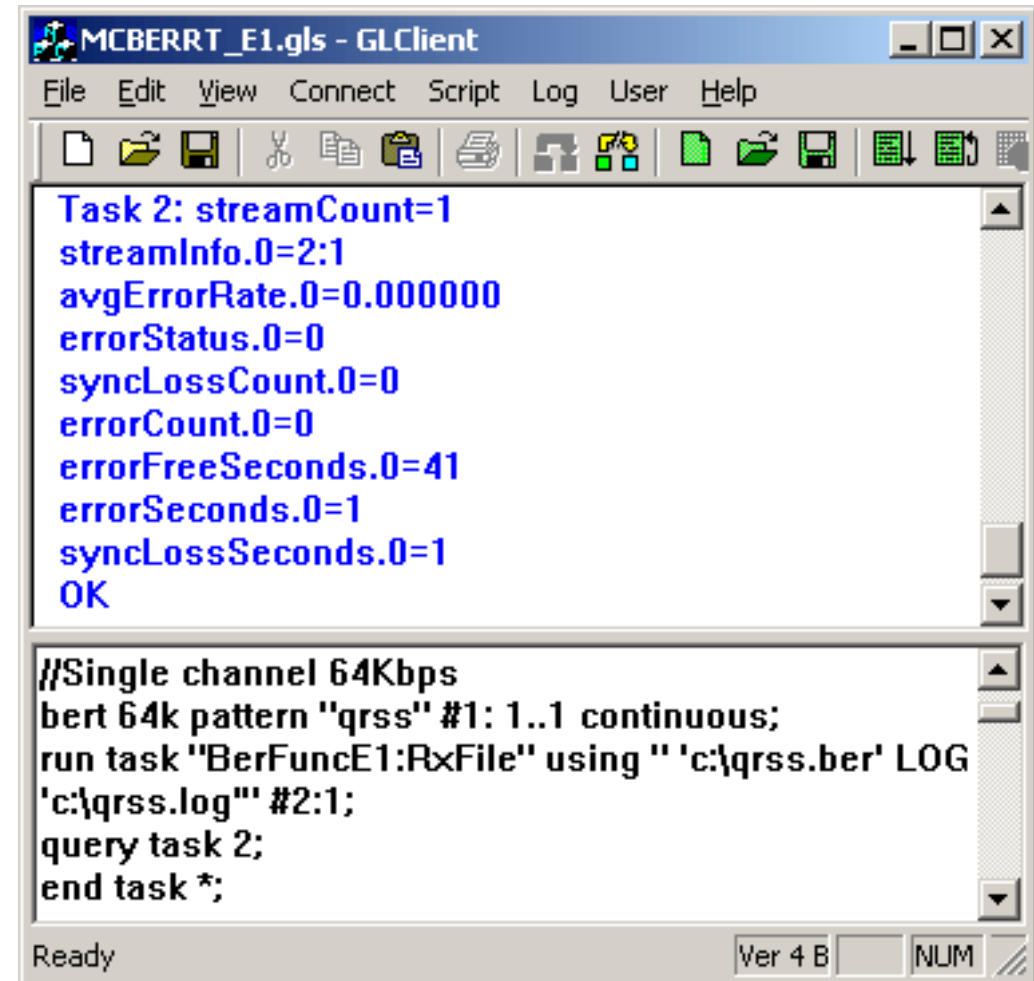
- Receive and transmit HDLC frames and signals in \*.hdl file format over facility data link (FDL)

# Multi-Channel Rx BER Testing

(Module license #- XX670)

Sample script for transmission and reception of 'QRSS' frame patter at 64 kbps on single channel

- Allows comparing data streams to a pattern file
- Streams may be captured on multiple cards, in a hyper channel Nx64 kbps, multiple channels 64 kbps or multiple subchannels Nx8 kbps



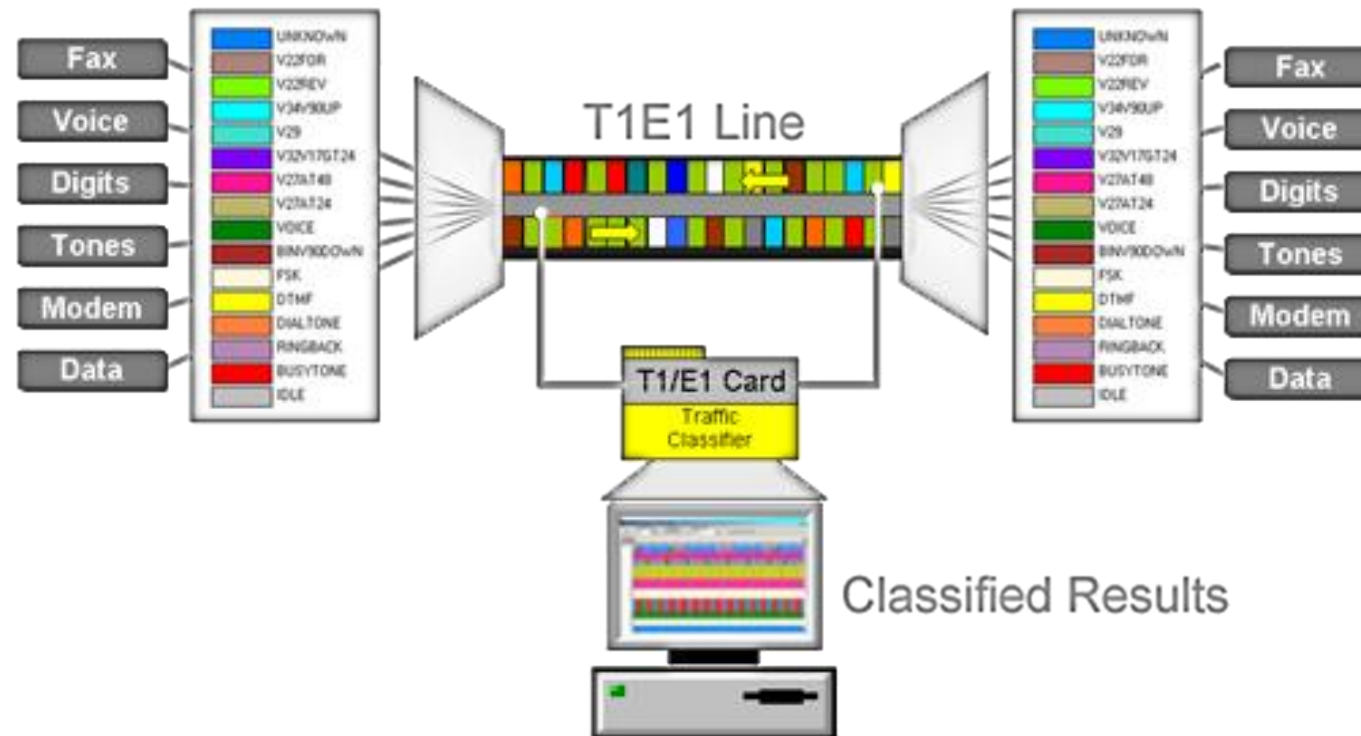
The screenshot shows a software window titled "MCBERRT\_E1.gls - GLClient" with a menu bar (File, Edit, View, Connect, Script, Log, User, Help) and a toolbar. The main area is divided into two sections. The top section displays test results for "Task 2" in blue text: streamCount=1, streamInfo.0=2:1, avgErrorRate.0=0.000000, errorStatus.0=0, syncLossCount.0=0, errorCount.0=0, errorFreeSeconds.0=41, errorSeconds.0=1, and syncLossSeconds.0=1. Below this is an "OK" button. The bottom section contains a script in black text: //Single channel 64Kbps, bert 64k pattern "qrss" #1: 1..1 continuous;, run task "BerFuncE1:RxFile" using "'c:\qrss.ber' LOG 'c:\qrss.log'" #2:1;, query task 2;, and end task \*;. The status bar at the bottom shows "Ready", "Ver 4 B", and "NUM".

```
Task 2: streamCount=1
streamInfo.0=2:1
avgErrorRate.0=0.000000
errorStatus.0=0
syncLossCount.0=0
errorCount.0=0
errorFreeSeconds.0=41
errorSeconds.0=1
syncLossSeconds.0=1
OK

//Single channel 64Kbps
bert 64k pattern "qrss" #1: 1..1 continuous;
run task "BerFuncE1:RxFile" using "'c:\qrss.ber' LOG
'c:\qrss.log'" #2:1;
query task 2;
end task *;
```

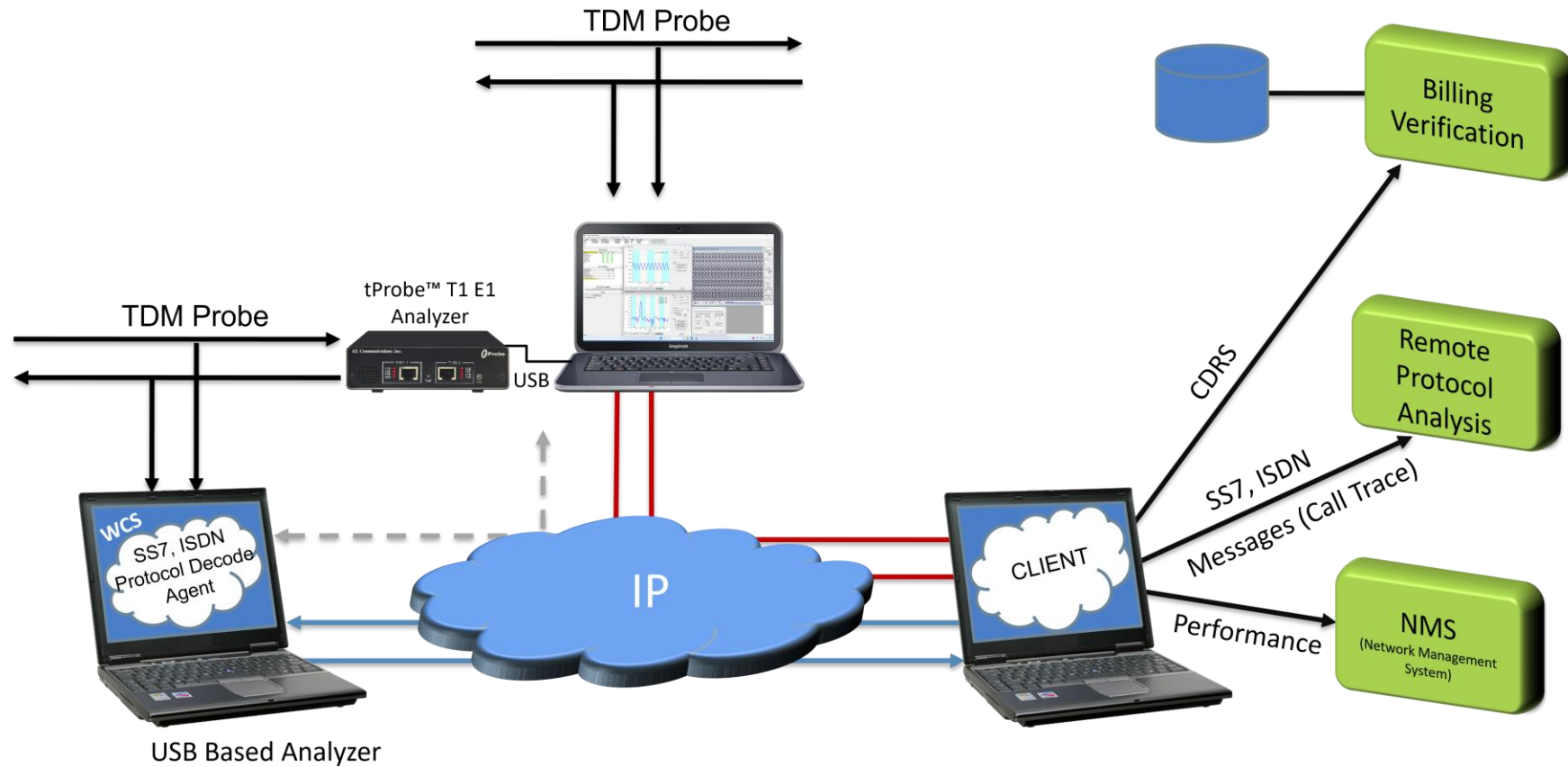
# Traffic Classifier (Module license #- XX680)

- GUI based client application
- Analyze and classifies various traffic on a T1 / E1 line such as voice, fax, data, tones (dial tone, ring-back tone, busy tone etc) as well as identify dialing digits and other events
- Uses various identifying schemes depending on the country of your selection and classifies the traffic, based on the dialing tones stipulated for that country





# SS7 and ISDN Decode Agents (Console Based Clients)



- Remote performance monitoring - SS7/ISDN messages with call trace for Remote Protocol Analysis



# SS7 and ISDN Decode Agents (Console Based Clients)

- Allows applying filters based on protocol layers and fields to limit the amount of data sent to client
- Allows capturing on specified timeslots, Nx64 hyper-channels, or sub-channels
- Builds CDRs, and streams over TCP/IP to remote site
- Allows the following types of information to be sent to the remote client
  - Frame header information (HDR)
  - Frame raw data (DATA)
  - Frame protocol decode field names and values (FIELDS)
  - CDR field names and values (CDRS) in \*.csv format

# SS7 Decode Agent (Module license # - XX690)

## Sample Script of SS7 INI File

```
ss7 - Notepad
File Edit Format View Help
[WCSPROTAN]
Module=wcsPaSS7E1:wcsPaSS7E1
IpAddr=127.0.0.1
IpPort=17090
ProtocolStandard="SS7 ANSI Standard"
LayerFilter.0="MTP2"
LayerFilter.1="MTP3"
LayerFilter.2="ISUP"
Fieldfilter.0="'Calling Address Signal'"
;==== Capture Streams====
HC.0=#1:1..31
HC.1=#2:1..31
;TS.0=#1:1
;TS.1=#2:1
;SC.0=#1:15:1..7
;SC.1=#2:20:1..7
SEND=HDR DATA FIELDS CDRS
;SEND=FIELDS CDRS
```

```
C:\WINDOWS\system32\cmd.exe - ConsFldCdrToCsv SS7.ini
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\>cd WCS
C:\WCS>ConsFldCdrToCsv SS7.ini
Connecting...
Starting task WcsPaSS7E1:WcsPaSS7E1 ...
>>inform task 1 "SET PROT DEF NAM 'SS7 ANSI Standard';>>
>>inform task 1 "PROT LAYER FILT DEF ADD 'MTP2';>>
>>inform task 1 "PROT LAYER FILT DEF ADD 'MTP3';>>
>>inform task 1 "PROT LAYER FILT DEF ADD 'ISUP';>>
>>inform task 1 "PROT FIELD FILTER DEFAULT ADD 'Calling Address Signal';>>
>>inform task 1 "CAPT HC #1:1..31";>>
>>inform task 1 "CAPT HC #2:1..31";>>
>>inform task 1 "SEND FRAME FIELDS CDRS";>>
>>inform task 1 "START";>>
```

- Remote performance monitoring - SS7/ISDN messages with call trace for Remote Protocol Analysis

**Thank you**