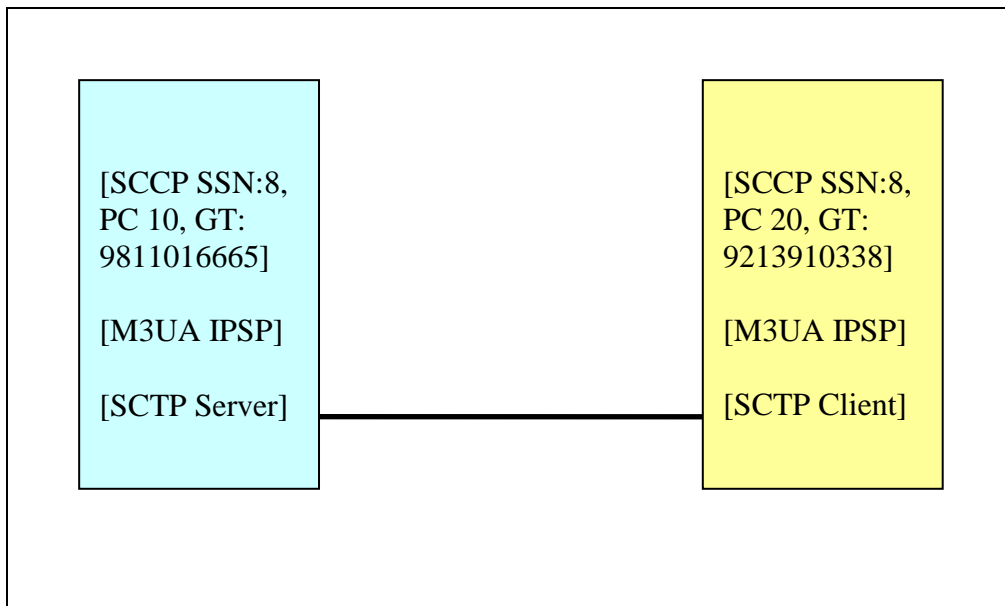


M3UA & CL-SCCP Demonstration



Pre-Requisites

- 1) LK-SCTP
- 2) Sure Speed CL-SCCP
- 3) Sure Speed M3UA
- 4) Wireshark [Please use wireshark to capture packets between Client and Server entity to verify the demonstrations. **Please note that wireshark when configured in ANSI mode does not decode ITU SCCP messages and vice versa, so you may see some malformed SCCP messages in wireshark, but this is due to limitation of wireshark.**]

Building the Demo Application

To build the demo application, change to directory “./sccp/demo” and give command “make”. This will prepare the executable “./smdemo”. The source code that may be used to understand how applications using SCCP and M3UA layers may be built are present in source files “./sccp/demo/sccp.c” and “./sccp/demo/m3ua.c” respectively.

M3UA demo applications are available separately also. They illustrate all the M3UA functionalities in a detailed manner. Please refer M3UA Programmer’s Manual for details about M3UA demo applications.

Beginning the Demonstrations

To begin the demonstration, we first need to start the server entity as following:

```
./smdemo <server-ip-address>:<server-port> <client-ip-address>:<client-port> s <ansi|itu>
```

Once Server has begun successfully, we will now begin the client entity a following:

```
./smdemo <client-ip-address>:<client-port> <server-ip-address>:<server-port> c <ansi|itu>
```

The last parameters in above commands specify the SCCP [ITU or ANSI] standard for which demonstration is to be seen. Please note that demo application configures local subsystems and remote subsystems for both ANSI and ITU SCCP standards. So, the initial exchange of SST [Subsystem Test] and SSA [Subsystem Allowed] messages happen as per both ANSI and ITU SCCP standards. After which demonstration is done only for the specific SCCP standard based on the Command Line Input parameter.

As soon as both Server and Client entities are active, they would undergo M3UA ASP/IPSP activation procedure. Once M3UA IPSP/ASP are active, SCCP subsystems would get RESUME indications, based on which they will exchange SPMG-SST and SPMG-SSA messages and become available for message processing.

Demonstration 1) N-STATE-REQ [ALLOWED->PROHIBITED]

Demonstration 1 is performed at Client entity where Management invokes N-STATE-REQ primitive to make SSN=8 PROHIBITED. This causes sending of SPMG-SSP message from Client entity to Server Entity. This would also cause triggering of Subsystem Test Procedure at the Server entity. An SPMG-SST message would be sent by Server entity to Client entity after 30 seconds of receipt of SPMG-SSP message. The timer of 30 seconds is a configurable timer.

Demonstration 2) N-STATE-REQ [PROHIBITED->ALLOWED]

Demonstration 2 is performed at client entity after 45 seconds of demonstration 1. In this case Management invokes N-STATE-REQ primitive to make SSN=8 allowed. This causes triggering of SPMG-SSA message from Client entity to Server entity.

Demonstration 3) UDT with Route-GT

Demonstration 3 would cause sending of a UDT message based on GTT routing from Client Entity to Server Entity. The GTT would take place on Server Entity and message would be routed to the local subsystem with SSN=8 at Server Entity. Please note that SCCP Data is filled with all 0 bytes.

Demonstration 4) UDT with Route-SSN

Demonstration 4 would cause sending of a UDT message based on SSN routing from Client Entity to Server Entity. The message would be routed to the local subsystem with SSN=8 at Server Entity. Please note that SCCP Data is filled with all 0 bytes.

Demonstration 5) XUDT with Route-GT

Demonstration 5 would cause sending of a XUDT message based on GTT routing from Client Entity to Server Entity. The GTT would take place on Server Entity and message would be routed to the local subsystem with SSN=8 at Server Entity. Please note that SCCP Data is filled with all 0 bytes.

Demonstration 6) XUDT with Route-SSN

Demonstration 6 would cause sending of a XUDT message based on SSN routing from Client Entity to Server Entity. The message would be routed to the local subsystem with SSN=8 at Server Entity. Please note that SCCP Data is filled with all 0 bytes.

Demonstration 7) Segmented XUDT [240 Bytes] with Route-SSN

Demonstration 7 would cause sending of two segmented XUDT messages based on SSN routing from Client Entity to Server Entity. The messages would be reassembled at server entity and then routed to the local subsystem with SSN=8 at Server Entity. Please note that SCCP Data is filled with all 0 bytes.

Demonstration 8) Segmented XUDT [240 Bytes] with Route-GT

Demonstration 8 would cause sending of two XUDT messages based on GTT routing from Client Entity to Server Entity. The GTT would take place on Server Entity and message would be routed to the local subsystem with SSN=8 at Server Entity. The message passed to subsystem would be a reassembled message. Please note that SCCP Data is filled with all 0 bytes.

Demonstration 9) UDTS Due to failed GTT

Demonstration 9 would cause sending of a UDT message based on GTT routing from Client Entity to Server Entity. As the called party GT is incorrect, Server Entity would send back UDTS to the Client entity. On receiving UDTS, Client entity would give N-NOTICE-IND to the local subsystem with SSN=8. Please note that SCCP Data is filled with all 0 bytes.

Demonstration 10) XUDTS Due to failed GTT

Demonstration 10 would cause sending of a XUDT message based on GTT routing from Client Entity to Server Entity. As the called party GT is incorrect, Server Entity would send back UDTS to the Client entity. On receiving UDTS, Client entity would give N-NOTICE-IND to the local subsystem with SSN=8. Please note that SCCP Data is filled with all 0 bytes.