ISUP

Table of Contents

(ISDN User Part or ISUP) is part of which is used to establish telephone calls and manage call switches(`exchanges`). Exchanges are connected via E1 or T1 trunks. Each trunk is divided by means of TDM into time slots. Each time slot is distinguished as circuit. Circuits (identified by code) are used as medium to transmit voice data between user equipment (or exchanges if more than one is involved).

```
allows not only to setup a call, but to exchange information about exchange state and
its resources(circuits).
```

ℹ Restcomm is based on `Q.76X` series of documents.

ISUP Configuration

Restcomm stack is configured with simple properties. Currently following properties are supported:

Table 1. ISUP Configuration options

Name	Default value	Value range	Description
ni	None, must be provided	0-3	Sets value of network indicator that should be used by stack.
localspc	None, must be provided	0 - (2^14)-1	Sets local signaling point code. It will be used as OPC for outgoing signaling units.
t1	4s	4s - 15s	Sets T1 value. Started when REL is sent. See A.1/Q.764
t5	5 min.	5min - 15 min	Sets T5 value. Started when initial REL is sent. See A.1/Q.764
t7	20s	20s -30s	Sets T7 value. (Re)Started when Address Message is sent. See A.1/Q.764
t2	15s	15s - 60s	Sets T12 value. Started when BLO is sent. See A.1/Q.764
t13	5min	5min - 15min	Sets T13 value. Started when initial BLO is sent. See A.1/Q.764

Name	Default value	Value range	Description
t14	5s	15s - 60s	Sets T14 value. Started when UBL is sent. See A.1/Q.764
t15	5min	5min - 15min	Sets T15 value. Started when initial UBL is sent. See A.1/Q.764
t16	5s	15s - 60s	Sets T16 value. Started when RSC is sent. See A.1/Q.764
t17	5min	5min - 15min	Sets T17 value. Started when initial RSC is sent. See A.1/Q.764
t18	5s	15s - 60s	Sets T18 value. Started when CGB is sent. See A.1/Q.764
t19	5min	5min - 15min	Sets T19 value. Started when initial CGB is sent. See A.1/Q.764
t20	5s	15s - 60s	Sets T20 value. Started when CGU is sent. See A.1/Q.764
t21	5min	5min - 15min	Sets T21 value. Started when initial CGU is sent. See A.1/Q.764
t22	5s	15s - 60s	Sets T22 value. Started when GRS is sent. See A.1/Q.764
t23	5min	5min - 15min	Sets T23 value. Started when initial GRS is sent. See A.1/Q.764
t28	10s	10s	Sets T28 value. Started when CQM is sent. See A.1/Q.764
t33	12s	12s - 15s	Sets T33 value. Started when INR is sent. See A.1/Q.764

Note that before start user must provide two interfaces to stack:

Mtp3UserPart

implementation of transport layer which should be used by stack

CircuitManager

circuit manager implementation. This interface stores information on mapping between `CIC`(Circuit Identification Code) and `DPC`(Destination Point Code) used as destination for outgoing messages.

ISUP Usage

The `org.restcomm.protocols.ss7.isup.ISUPStack` interface defines the methods required to represent ISUP Protocol Stack. ISUPStack exposes `org.restcomm.protocols.ss7.isup.ISUPProvider`. This interface defines the methods that will be used by any registered ISUP User application implementing the `org.restcomm.protocols.ss7.isup.ISUPListener` to listen ISUP events(messages and timeouts).

ISUP Example

Below is simple example of stack usage:

```
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import java.util.Properties;

import org.restcomm.protocols.ss7.isup.ISUPEvent;
import org.restcomm.protocols.ss7.isup.ISUPListener;
import org.restcomm.protocols.ss7.isup.ISUPProvider;
import org.restcomm.protocols.ss7.isup.ISUPStack;
import org.restcomm.protocols.ss7.isup.ISUPTimeoutEvent;
import org.restcomm.protocols.ss7.isup.ParameterException;
import org.restcomm.protocols.ss7.isup.impl.ISUPStackImpl;
import org.restcomm.protocols.ss7.isup.message.ISUPMessage;

import org.restcomm.ss7.linkset.oam.Layer4;
import org.restcomm.ss7.linkset.oam.Linkset;

public class ISUPTest implements ISUPListener
{

    protected ISUPStack stack;
    protected ISUPProvider provider;

    protected Linkset isupLinkSet;


    public void setUp() throws Exception {

        this.isupLinkSet = ....; //same linksets as in SS7Service
        this.stack = new ISUPStackImpl();
        this.stack.configure(getSpecificConfig());
        this.provider = this.stack.getIsupProvider();
        this.provider.addListener(this);
        Mtp3UserPart userPart = // create with proper factory, dahdii, dialogi, m3ua
        this.stack.setMtp3UserPart(userPart);
        CircuitManagerImpl circuitManager = new CircuitManagerImpl();
        circuitManager.addCircuit(1, 431613); // CIC - 1, DPC for it - 431613


        this.stack.setCircuitManager(circuitManager);
        this.stack.start();

    }


    public void onEvent(ISUPEvent event) {
        ISUPMessage msg = event.getMessage();
        switch(msg.getCircuitIdentificationCode().getCIC())
        {
            case AddressCompleteMessage._COMMAND_CODE:
            //only complete
            break;
            case ConnectedMessage._COMMAND_CODE:
            case AnswerMessage._COMMAND_CODE:
            //we are good to go
            ConnectedNumber cn = (ConnectedNumber)msg.getParameter(ConnectedNumber
._PARAMETER_CODE);
            //do something
            break;
            case ReleaseMessage._COMMAND_CODE:
            //remote end does not want to talk
            RealeaseCompleteMessage rlc = provider.getMessageFactory().createRLC();
            rlc.setCircuitIdentificationCode(msg.getCircuitIdentificationCode());
            rlc.setCauseIndicators(   ((ReleaseComplete)msg).getCauseIndicators());
            provider.sendMessage(rlc);

        }

    }

    public void onTimeout(ISUPTimeoutEvent event) {
        switch(event.getTimerId())
        {
            case ISUPTimeoutEvent.T1:
                //do something
                break;
            case ISUPTimeoutEvent.T7:
                //do even more
                break;
        }

    }

}
```