Example

Table of Contents

This chapter tours around the core constructs of Restcomm SCTP Library with simple examples to let you get started quickly. You will be able to write a client and a server on top of Restcomm SCTP Library right away when you are at the end of this chapter.

Before Getting Started

The minimum requirements to run the examples which are introduced in this chapter are only two; the latest version of Restcomm SCTP Library and JDK 1.7 or above with SCTP support. At time of writing this guide linux kernel has native support for SCTP (lksctp) and also Solaris includes SCTP.

Initiating Management

The primitive step in uisng Restcomm SCTP Library is to create instance of `Management` class and set appropriate parameters.

```
    private static final String SERVER_NAME = "testserver";
    private static final String SERVER_HOST = "127.0.0.1";
    private static final int SERVER_PORT = 2345;

    private static final String SERVER_ASSOCIATION_NAME = "serverAsscoiation";
    private static final String CLIENT_ASSOCIATION_NAME = "clientAsscoiation";

    private static final String CLIENT_HOST = "127.0.0.1";
    private static final int CLIENT_PORT = 2346;
    ...
    ....
    ....

    Management management = new NettySctpManagementImpl("SCTPTest"); //1. See NOTE
below
    management.setConnectDelay(10000);// Try connecting every 10 secs //2. See NOTE
below
    management.start();
```

Footnotes from comments on the source code above below.

1. Create new instance of NettySctpManagementImpl and setting the management name. The management will search for SCTPTest_SCTP.xml file to load the previously configured Serever's or Association's. If file is not found it will create one.
2. connectDelay is only useful for Associations acting as client side trying to connect to peer. The value specified is time in milliseconds the unedrlying socket will try to connect to peer if the existing connection is broken or even if its fresh connection attempt.

Adding Server and server Association

Once the Managment is setup, application can create `Server` and add server `Association`

```
    Server server = this.management.addServer(SERVER_NAME, SERVER_HOST, SERVER_PORT);
    Association serverAssociation = this.management.addServerAssociation(CLIENT_HOST,
CLIENT_PORT, SERVER_NAME, SERVER_ASSOCIATION_NAME); //1. See NOTE below

    serverAssociation.setAssociationListener(new ServerAssociationListener()); //2.
See NOTE below

    this.management.startAssociation(SERVER_ASSOCIATION_NAME);
    this.management.startServer(SERVER_NAME);
```

Footnotes from comments on the source code above.

1. Add the server and server association. Multiple server's can be added to each management and each server can have multiple server association's
2. The instance of AssociationListener should be registered with newly created Associaton before starting it. There is no dependency on order of starting server and server association.

Below is example of class implementing `AssociationListener`

```
class ServerAssociationListener implements AssociationListener {

    private final byte[] SERVER_MESSAGE = "Server says Hi".getBytes();
    private boolean serverAssocUp;
    private boolean serverAssocDown;
    private byte[] serverMessage;

    /*
     * (non-Javadoc)
     *
     * @see
     * org.mobicents.protocols.sctp.AssociationListener#onCommunicationUp
     * (org.mobicents.protocols.sctp.Association)
     */
    @Override
    public void onCommunicationUp(Association association, int maxInboundStreams, int
maxOutboundStreams) {
        System.out.println(this + " onCommunicationUp");

        serverAssocUp = true;

        try {
            ByteBufAllocator byteBufAllocator = association.getByteBufAllocator();
            ByteBuf byteBuf;
            if (byteBufAllocator != null) {
                byteBuf = byteBufAllocator.buffer();
            } else {
                byteBuf = Unpooled.buffer();
            }
            byteBuf.writeBytes(SERVER_MESSAGE);
            PayloadData payloadData = new PayloadData(SERVER_MESSAGE.length, byteBuf,
true, false, 3, 1);
            association.send(payloadData);
        } catch (Exception e) {
        }
    }

    /*
     * (non-Javadoc)
     *
     * @see
     * org.mobicents.protocols.sctp.AssociationListener#onCommunicationShutdown
     * (org.mobicents.protocols.sctp.Association)
     */
    @Override
    public void onCommunicationShutdown(Association association) {
        System.out.println(this + " onCommunicationShutdown");
        serverAssocDown = true;
    }

    /*
     * (non-Javadoc)
     *
     * @see
     * org.mobicents.protocols.sctp.AssociationListener#onCommunicationLost
     * (org.mobicents.protocols.sctp.Association)
     */
    @Override
    public void onCommunicationLost(Association association) {
        System.out.println(this + " onCommunicationLost");
    }
    /*
     * (non-Javadoc)
     *
     * @see
     * org.mobicents.protocols.sctp.AssociationListener#onCommunicationRestart
     * (org.mobicents.protocols.sctp.Association)
     */
    @Override
    public void onCommunicationRestart(Association association) {
        System.out.println(this + " onCommunicationRestart");
    }

    /*
     * (non-Javadoc)
     *
     * @see
     * org.mobicents.protocols.sctp.AssociationListener#onPayload(org.mobicents
     * .protocols.sctp.Association,
     * org.mobicents.protocols.sctp.PayloadData)
     */
    @Override
    public void onPayload(Association association, PayloadData payloadData) {
        System.out.println(this + " onPayload");

        ByteBuf byteBuf = payloadData.getByteBuf();
        serverMessage = new byte[byteBuf.readableBytes()];
        byteBuf.getBytes(0, serverMessage);
        ReferenceCountUtil.release(byteBuf);

        System.out.println(this + "received " + new String(serverMessage));
    }

    @Override
    public void inValidStreamId(PayloadData payloadData) {
    }
}
```

Adding Association

Once the Managment is setup, application can create client side `Association`.

```
        Association clientAssociation = this.management.addAssociation(CLIENT_HOST,
CLIENT_PORT, SERVER_HOST, SERVER_PORT, CLIENT_ASSOCIATION_NAME);
        clientAssociation.setAssociationListener(new ClientAssociationListenerImpl());
        this.management.startAssociation(CLIENT_ASSOCIATION_NAME);
```

Uasge for anonymous Associations

You may work not with a list of preconfigured associations but accept any incoming connections. For this you need:

- configure Server and set its `acceptAnonymousConnections` option to true
- configure no association
- implement ServerListener interface and register it to SCTP Management
- implement ServerListener.onNewRemoteConnection() method like:

```
    public void onNewRemoteConnection(Server server, Association association) {
        if (<if I want to reject this incoming connection>) {
            association.rejectAnonymousAssociation();
        } else {
            association.acceptAnonymousAssociation(new ServerAssociationListener(ad));
        }
    }
```