

Appendix B

OMNET AUTHENTICATION CODE

```
#include "tvws.h"

Define_Module(tvwsd);
void tvwsd::initialize()
{
    if (strcmp("server", getName()) == 0)
    {
        ev << "server,sending initial message" << endl;
        msg = new cMessage("good day ServerMsg");
        // send(msg, "out");
        //bubble("call master!");

        //send(msg, "out");
    }
    if (strcmp("master", getName()) == 0)
    {
        ev<<"master,sending initial message" << endl;
        msg = new cMessage("Hello world");
        // send(msg, "out");
        //msg = new cMessage("masterMsg");

        //send(msg, "out");
    }

    scheduleAt(0.0, msg);

} // end initialize()

void tvwsd::handleMessage(cMessage *msg)
{
    ev << "Received message " << msg->getName()<< "" << endl;
    //send(msg, "out");
    servermessage(msg);
}

void tvwsd::mastermessage(cMessage *msg)
{
    randM = uniform(0,4);
    if (strcmp("master", getName()) == 0)
```

```

if (recmsgtype=0)
{
    {
        ev << "master, sending initial message" << endl;
if (msgtype = 10)
    {
        //bubble("ARRIVED, Hello in master!");
        msgtype ++;
        msg = new cMessage(" Sending RegReqMsg" );
        ev <<"msg type " << msgtype;
        ev <<" msg name " << msg << endl;

        }// end messagetype = 10

if (msgtype = 12)
    {
        ev <<"msg type " << msgtype ;

        // bubble("ARRIVED, MasterID req!");
        if (masterID = randS)
            {
                msg = new cMessage("ServerID Req" );
                ev <<"  msg name " << msg << endl;

                masterValue = mymastervalue[randS];
                serverID = randM;
                serverValue = myservervalue[randM];
            }
            msgtype ++;
        }// end messagetype = 12

if (msgtype = 14)
    {
        ev <<"msg type " << msgtype ;

        // bubble("ARRIVED, MasterID req!");
        if (serverValue = myservervalue[randM])
            {
                finsh ="pass";

                ev <<"  msg name " << msg << endl;

            }
            msgtype ++;
        }// end messagetype = 14

```

```

    if (msgtype = 16)
    {
        msgtype++;
        masterValue = mymastervalue[randS];
        msg = new cMessage("user list Req" );
    }

} // end recieve message type
} // end (strcmp("master", getName()) == 0)

send(msg, "out");

}

void tvwsd::servermessage(cMessage *msg)
{ randS = uniform(0,4);
  char *Mans;
  if (strcmp("server", getName()) == 0)
  {
      msgtype == 10;
      recmsgtype =0;

      // bubble("ARRIVED, in server!");
      ev <<"sending initial message 10 " ;
      msg = new cMessage("serverMsg Saying Hello ");
      if (msgtype = 11)
      {
          ev <<" msg type " << msgtype ;
          msg = new cMessage("MasterID Req");
          ev <<" msg name " << msg << endl;
          bubble("ARRIVED, REGREQ message!");

          //masterID = M_ID[s];
          masterID =randS;
          Mans = mymastervalue[randS];
          msgtype++;
      }

      if (msgtype = 13)
      { ev <<"msg type " << msgtype ;
        bubble("recieve Master value!");
        if (masterValue = Mans)
        {
            if (serverID = 3)

```

```

        {
            //serverValue = "server1";
            serverValue = myservervalue[randM];
            bubble(myservervalue[randM]);
            msg = new cMessage("serverID resp");
            ev <<" msg name " << msg << endl;
        }
    }
    msgtype++;
} // end if (msgtype = 13)

if (msgtype = 15)
{
    ev <<"msg type " << msgtype ;

    if (finsh = "pass" )
    {
        msg = new cMessage("Ack");
        ev <<" msg name " << msg << endl;
        bubble("Master Authentication success !");

        // endSimulation();
    }
    msgtype ++;
}

if ((msgtype = 17) and (masterValue = Mans))
{
    mymasterUlist[0]="U1";
    mymasterUlist[1]="U2";

    msg = new cMessage("User List");
    // endSimulation();
}
msgtype ++;
//}

//send(msg, "out");
}
mastermessage(msg);
}

```

