**Name: Ahmad Amjad Mughal**

**Reg: 121672**

**Class: BSCS-6C**

# Lab 03: Understanding the Concept of Load Balancer inside Data Centers

**Package.ned**

**package** loadbalancesimulations.simulations;

**simple** Server

{

**gates**:

**input** in[];

**output** out[];

}

**network** LoadBalancingTopology

{

**@display**("abc=131,92");

**submodules**:

A: Server{

**@display**("p=152.6453,10.65");

}

B: Server{

**@display**("p=101.462004,65.36005");

}

C: Server{

**@display**("p=82.99,27.30001");

}

D: Server{

**@display**("p=41.76,10.65");

}

**connections**:

A.out++ **-->** { delay = 100ms; } **-->** B.in++;

D.out++ **-->** { delay = 100ms; } **-->** B.in++;

C.out++ **-->** { delay = 100ms; } **-->** B.in++;

B.out++ **-->** { delay = 100ms; } **-->** C.in++;

B.out++ **-->** { delay = 100ms; } **-->** D.in++;

B.out++ **-->** { delay = 100ms; } **-->** A.in++;

}

**Server.cc**

#include "Server.h"

#include <string.h>

#include <omnetpp.h>

Define\_Module(Server);

int d=1;

using namespace omnetpp;

/\* Implementation of defined functions initialize() and handleMessage() in a sequential fashion.\*/

void Server::initialize()

{

/\*If Message gets received from trafficGenerator then instance of Message is created and is delivered as well. \*/

if (strcmp("A" , getName())==0)

{

cMessage \*msg= new cMessage("class");

send(msg, "out[0]");

}

}

//If message gets received from LoadBalancer and checking bit is 1 then move this message to Sink Server C else Sink Server D receives this message.

void Server:: handleMessage(cMessage \*msg) {

// TODO Auto-generated destructor stub

if (strcmp("B", getName())==0){

if (d==1){

send(msg, "out", 4);

d=0;

}

else

send(msg, "out", 5);

}

else

send(msg, "out", 3);

}

**Server.h:**

#ifndef SERVER\_H\_

#define SERVER\_H\_

#include <omnetpp.h>

using namespace omnetpp;

class Node: public cSimpleModule {

protected:

virtual void initialize();

virtual void handleMessage(cMessage \*msg);

};

#endif /\* SERVER\_H\_ \*/

**omnetpp.ini:**

[General]

network= LoadBalancingTopology

