

Lab Report

Course Title: Computer Networks Laboratory
Course Code: CSE-3634

Autumn-2021

Lab No: 5

Name of Labwork: Improving the random routing of Lab-4 by excluding the incoming gate for forwarding.

Student's ID : C183047
Name : Muhammad Abdullah Tuhin
Date of
Performance : 23 february 2022
Date of
Submission : 9 April 2022

| |
|---------|
| Marks : |
|---------|

1. **Introduction:** In this lab I develop a simulation that Turning into real Network of 10 routers and two hosts. The message is “frame47”.

2. **Description:** One of the nodes will generate message, here PC47_2 will generate message. That message will be passed to router_47[9], then it will forward the message to router_47[8] and router_47[7] in sometime one of these router will deliver it to router_47[5] or router_47[8]. After router_47[5] getting the message it will continue to deliver to their nearby router_47[6] , router_47[3] , router_47[4] . If router_47[6] get the message then it will again forward to router_47[5] or router_47[8] & then message could possibly back to these two router . Then router_47[3] and router_47[4] is now forward the message to the next router_47[1] . Now router_47[1] will deliver the message to router_47[2] or router_47[0] .After all these by forwarding message from one to one router somehow message will come to router_47[0] & then it will send the message to PC47_1 after these long period of time. But actually it will be less than a minute.

4. Module:

frame_47.msg

```
message frame47
{
    int hopcount = 0;
}
```

router47.ned

```
@license(LGPL);
simple router47
{
    parameters:
        @display("i=block/routing");
    gates:
```

```

        inout gate[];
    }

network lab5_47_network
{
    @display("bgb=825.28796,595.128");
    types:
        channel Channel extends ned.DelayChannel
        {
            delay = 100ms;
        }
    submodules:
        rt47[10]: router47 {

        }
        pc47_1: pc47 {

        }
        pc47_2: pc47 {

        }
    connections:
        rt47[0].gate++ <--> Channel <--> rt47[1].gate++;
        rt47[0].gate++ <--> Channel <--> rt47[2].gate++;
        rt47[1].gate++ <--> Channel <--> rt47[3].gate++;
        rt47[1].gate++ <--> Channel <--> rt47[3].gate++;
        rt47[4].gate++ <--> Channel <--> rt47[8].gate++;
        rt47[4].gate++ <--> Channel <--> rt47[9].gate++;
        rt47[3].gate++ <--> Channel <--> rt47[4].gate++;
        rt47[2].gate++ <--> Channel <--> rt47[7].gate++;
        rt47[8].gate++ <--> Channel <--> rt47[9].gate++;
        rt47[1].gate++ <--> Channel <--> rt47[5].gate++;
        rt47[5].gate++ <--> Channel <--> rt47[6].gate++;
        rt47[6].gate++ <--> Channel <--> rt47[9].gate++;
        rt47[7].gate++ <--> Channel <--> rt47[8].gate++;
        rt47[7].gate++ <--> Channel <--> rt47[4].gate++;
        pc47_1.interface <--> Channel <--> rt47[0].gate++;
        pc47_2.interface <--> Channel <--> rt47[5].gate++;
    }
}

```

pc47.ned

simple PC47_Lab5

```

{

    parameters:

```

```

        bool sendMsgOnInit = default(false);

        bool recvMsgAtDest = default(false);

        gates:

            inout interface;

    }

```

router47.cc

```

#include <stdio.h>
#include <string.h>
#include <omnetpp.h>

#include "frame_47_m.h"

using namespace omnetpp;

class router47 : public cSimpleModule
{
    protected:

        virtual void forwardMessage(Frame47 *msg);

        virtual void initialize() override;

        virtual void handleMessage(cMessage *msg) override;
};

Define_Module(router47);

void router47::initialize()
{
}

void router47::handleMessage(cMessage *msg)
{
    Frame47 *ttmsg = check_and_cast<Frame47 *>(msg);

    forwardMessage(ttmsg);
}

void router47::forwardMessage(Frame47 *msg)
{

```

```

// Increment hop count.
msg->setHopCount(msg->getHopCount() + 1);

// Same routing as before: random gate.
int n = gateSize("gate");
int k = intuniform(0, n - 1);

cGate *arrivalGate = msg->getArrivalGate();

if (arrivalGate != NULL)
{
    int arrivalGateIndex = arrivalGate->getIndex();

    if (n >= 2)
    {
        while (arrivalGateIndex == k)
        {
            k = intuniform(0,n-1);
        }
    }
}

EV << "Forwarding message " << msg << " on gate[" << k << "]\n";

send(msg, "gate$o", k);
}

```

pc47.cc

```

#include <stdio.h>

#include <string.h>
#include <omnetpp.h>

using namespace omnetpp;

#include "frame_47_m.h"

class pc47 : public cSimpleModule
{
private:
    bool sender,receiver;

protected:

    virtual Frame47 *generateMessage();

```

```

    virtual void forwardMessage(Frame47 *msg);

    virtual void initialize() override;

    virtual void handleMessage(cMessage *msg) override;
};

Define_Module(pc47);

void pc47::initialize()
{
    sender=par("sendMsgOnInit");

    receiver=par("recvMsgAtDest");

    if (sender == true) {
        Frame47 *msg = generateMessage();

        send(msg, "interface$o");
    }
}

void pc47::handleMessage(cMessage *msg)
{
    Frame47 *ttmsg = check_and_cast<Frame47 *>(msg);

    if (receiver == true) {
        // Message arrived.

        EV << "Message " << ttmsg << " arrived after " << ttmsg->getHopcount() << "
hops.\n";

        bubble("ARRIVED, Deleting Frame!");

        delete ttmsg;
    }
    else {
        forwardMessage(ttmsg); // We need to forward the message.
    }
}

Frame47 *pc47::generateMessage()
{
    char msgname[20];

```

```

    sprintf(msgname, "FRAME47");

    // Create message object and set source and destination field.

    Frame47 *msg = new Frame47(msgname);

    return msg;
}

void pc47::forwardMessage(Frame47 *msg)
{
    // Increment hop count.

    msg->setHopcount(msg->getHopcount()+1);

    EV << "Forwarding back message " << msg << " on its interface "<<"\n";

    send(msg, "interface$o");
}

```

omnetpp4.ini

```

[General]

[Config lab5_47_network]

seed-0-mt=352569

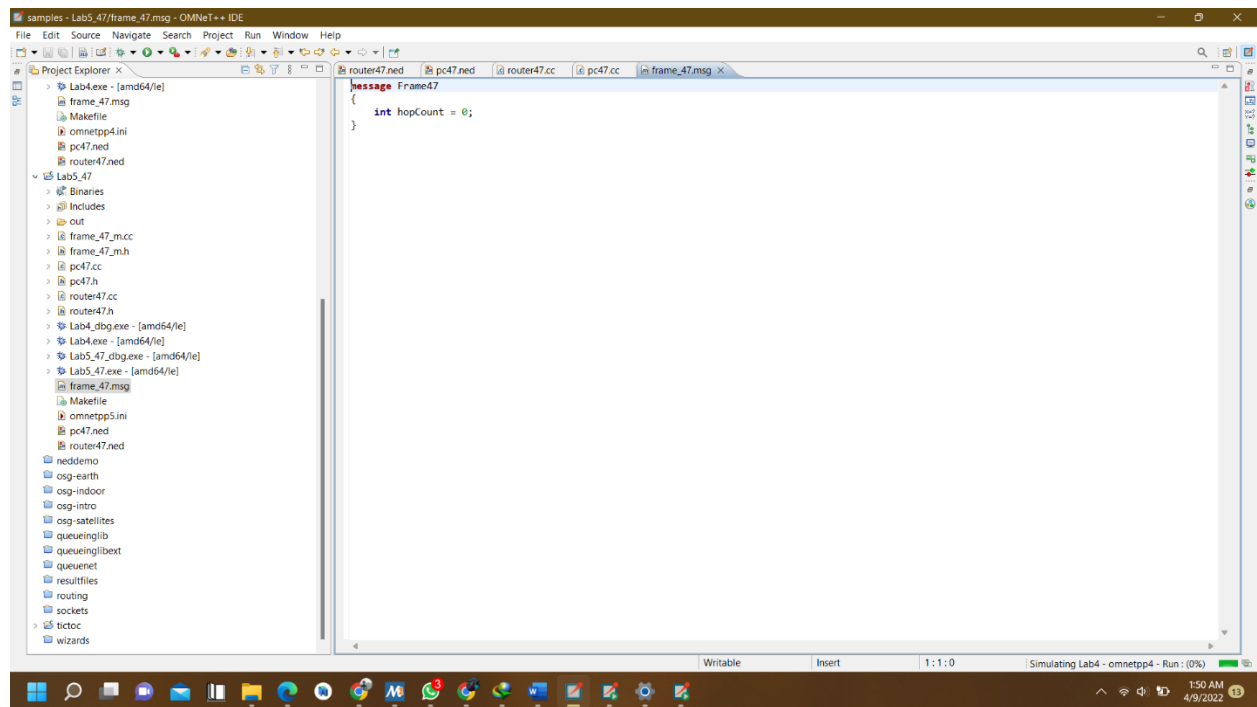
network = lab5_47_network

**.pc47_2.sendMsgOnInit = true

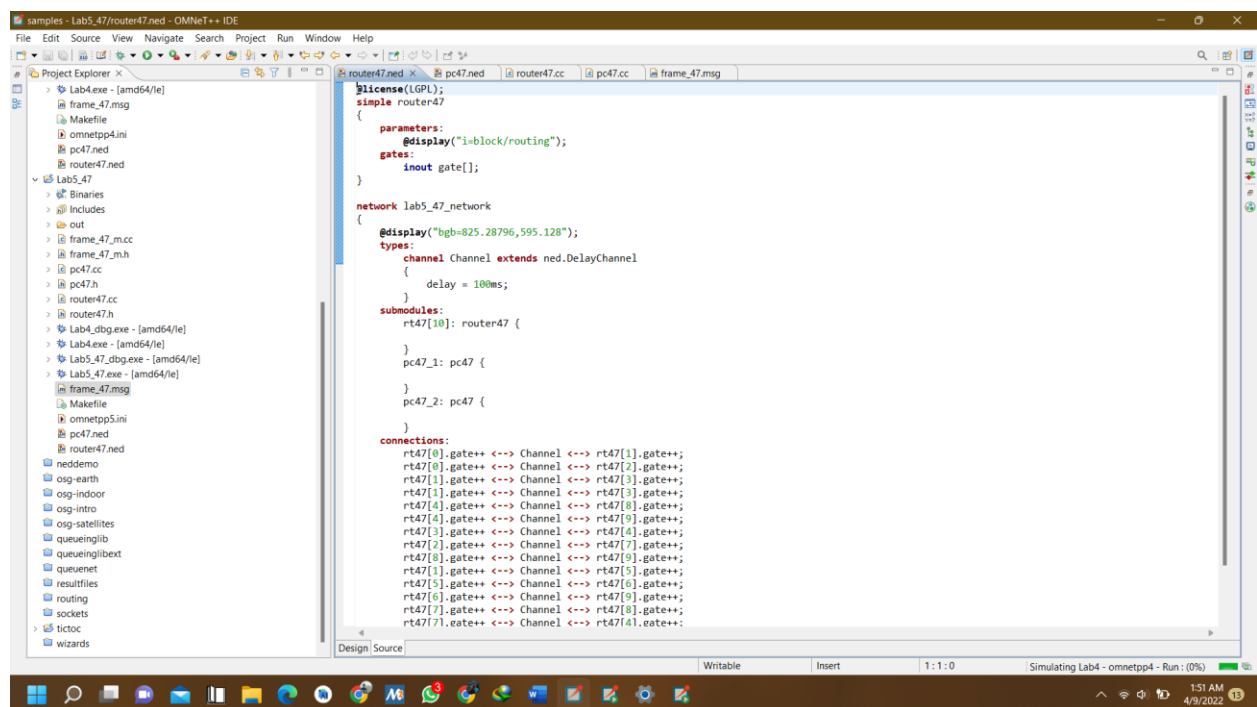
**.pc47_1.recvMsgAtDest = true

```

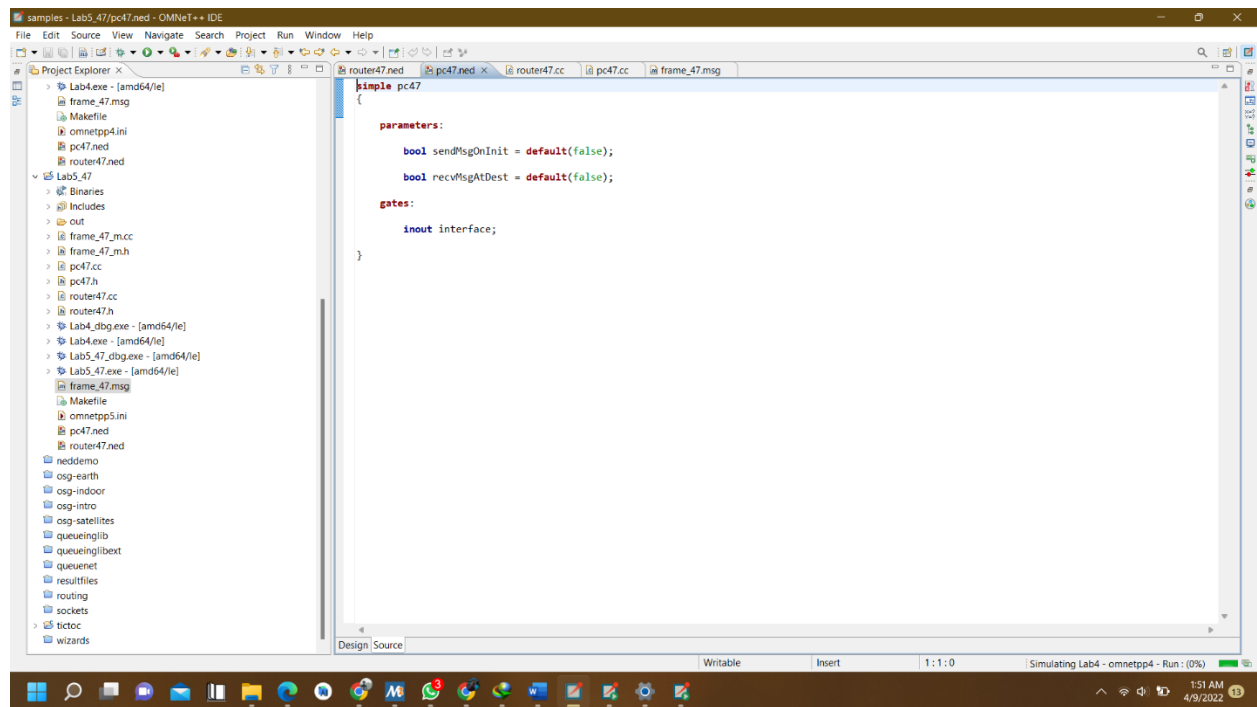
4.Frame Message file :



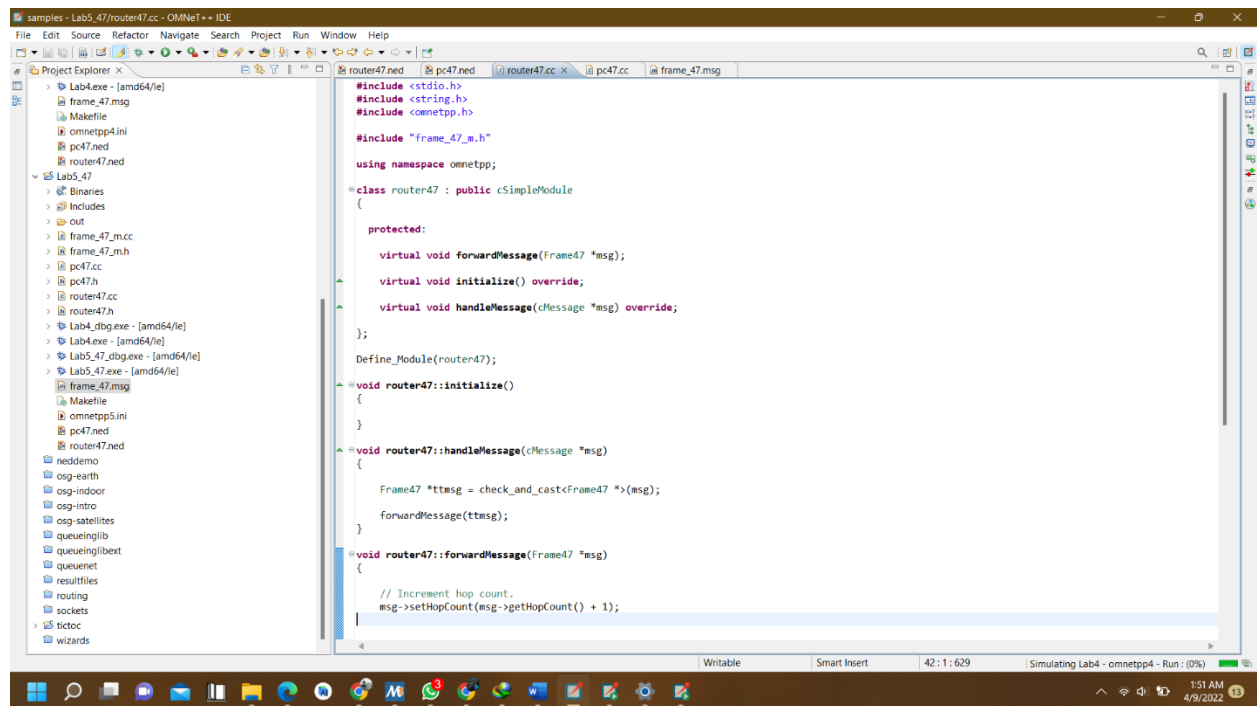
5. Router NED file:



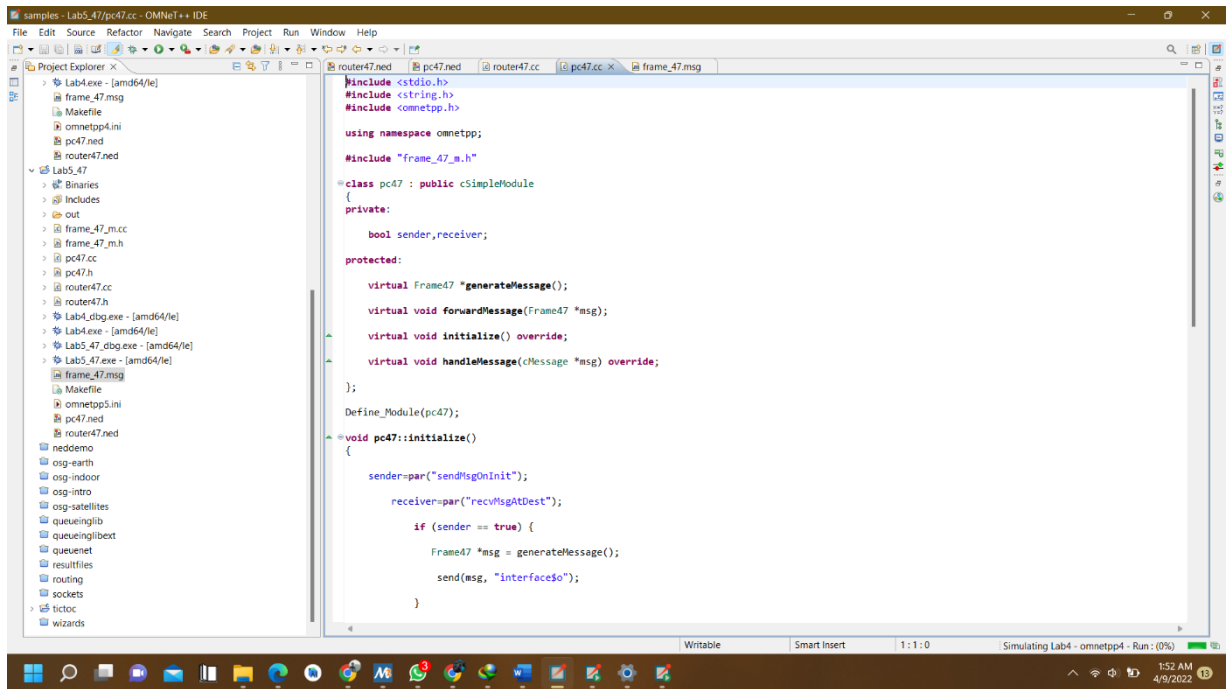
6. PC NED file :



7. Router CC file :

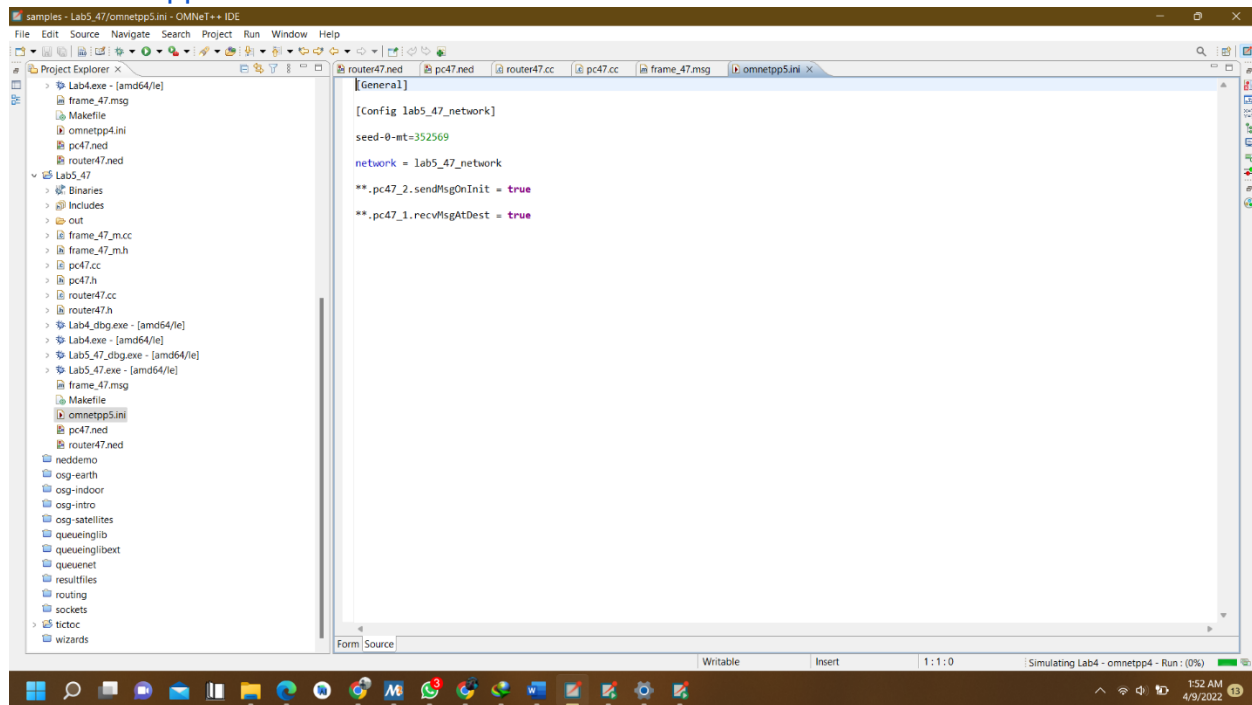


8. PC CC file :

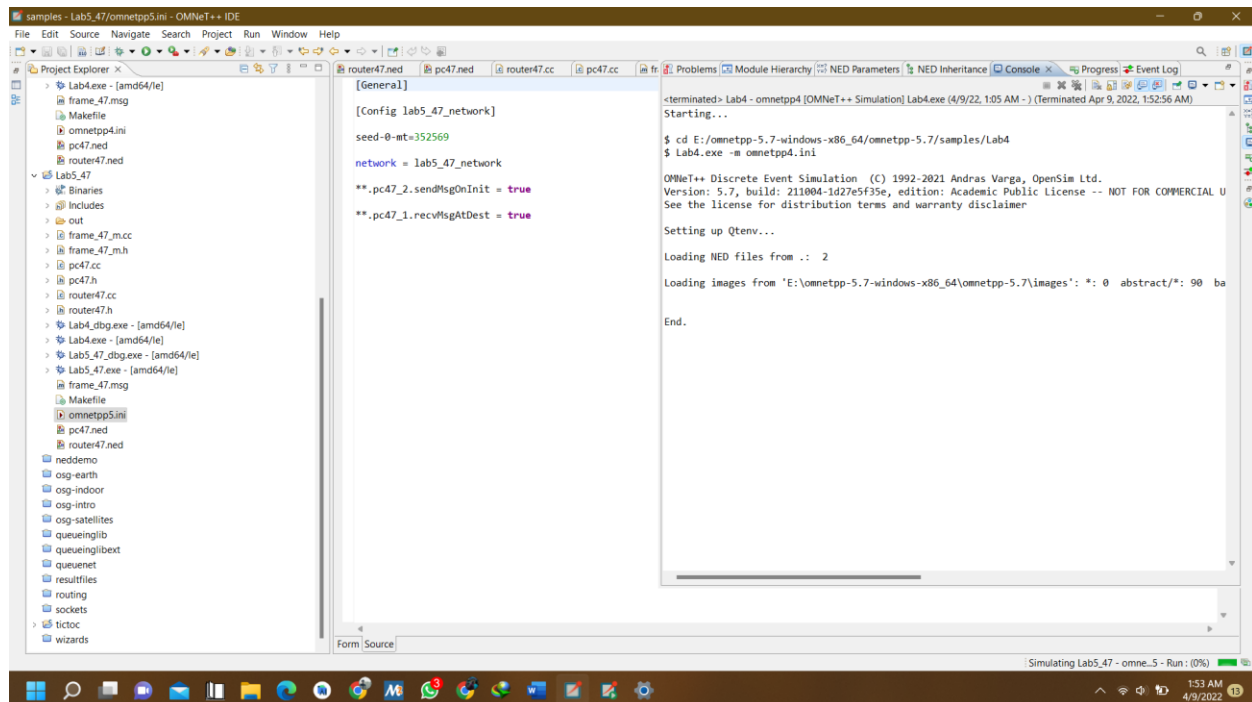


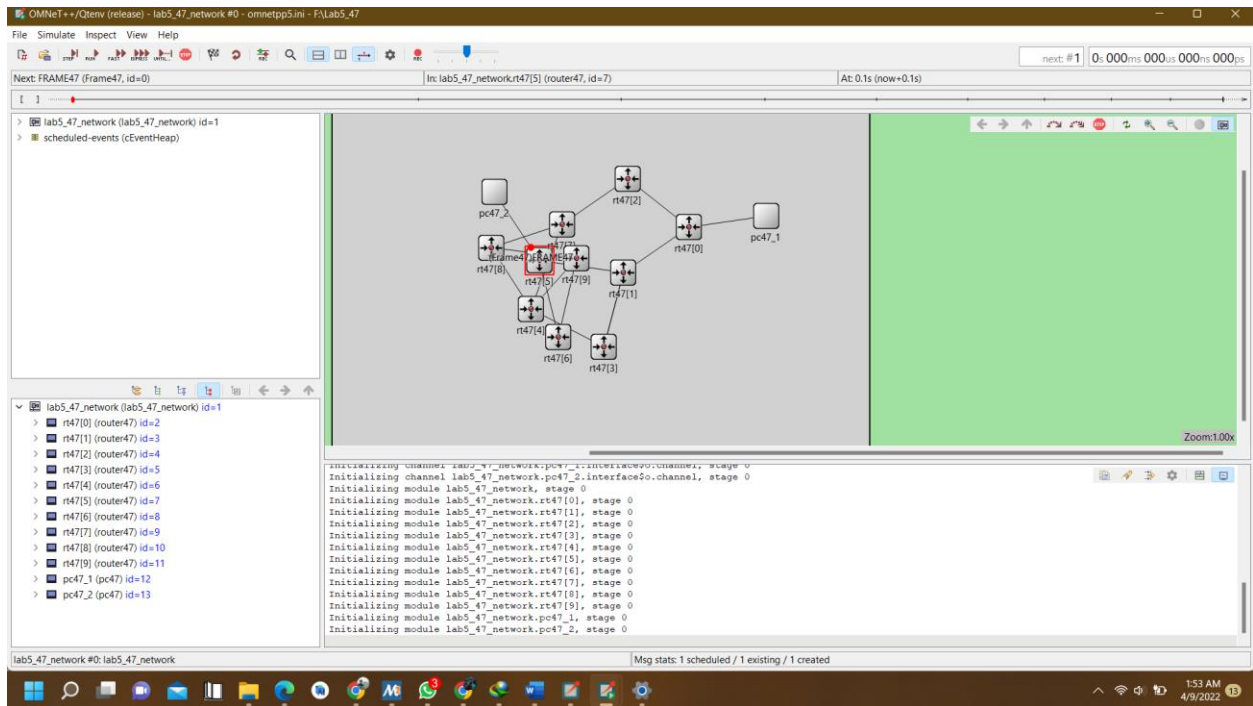
9.

10. Omnetpp INI file :

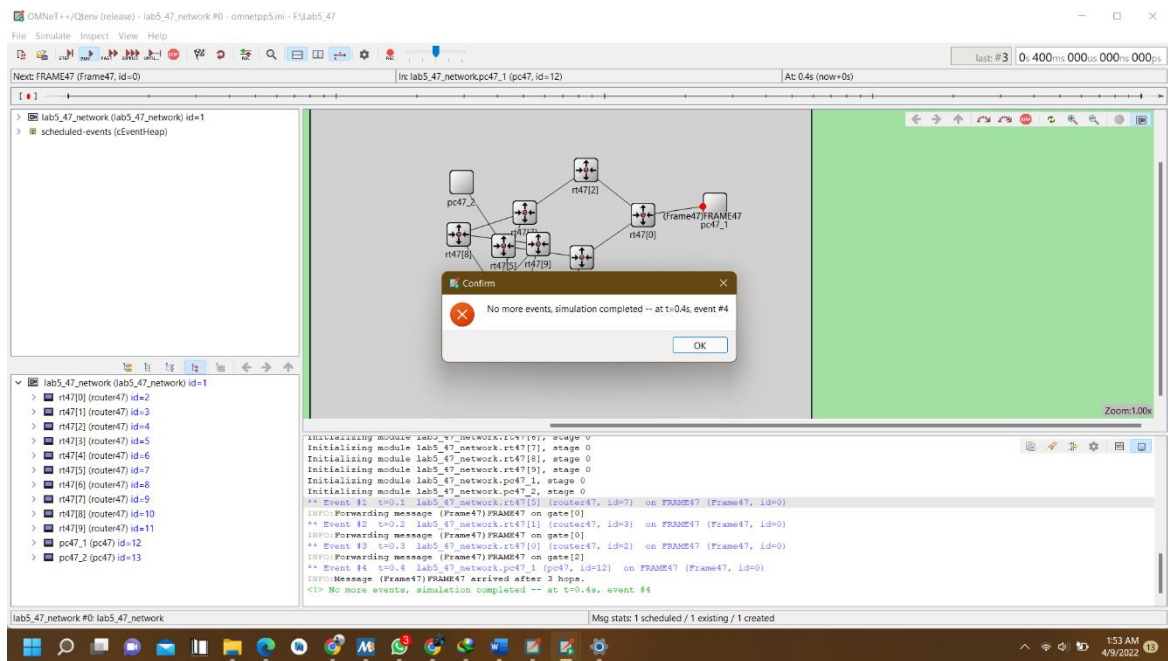


11. Build and Simulation:





12. Result Analysis



12 . Conclusion: Here from result analysis u can see that my project is successfully completed, for more confirmation we can see the build from manual given up there. In result analysis we can see the message is forwarding from PC47_2 to router_47[9] and following router_47[9] to router_47[1] and finally the will be reached at 4 event later at PC47_1 & there is no bug, no error .

And finally lab5 is successfully complete.

