Universiti Malaysia Sabah

Facultyof Computingand Informatics

KP00303

NETWORK SIMULATION

SEM 1-2023/2024

ASSIGNMENT 2

NAME	**ID**
SELVA GANAPATHY A/L MUTHU KUMAR	BI20110217
YOGANATHAN SHUNMUGAM	BI20160312
PAGITHEREN A/L UMABATHY	BI20160316
NESHADRAJ A/L SASIDHARAN	BI20160350

KP00303 Network Simulation

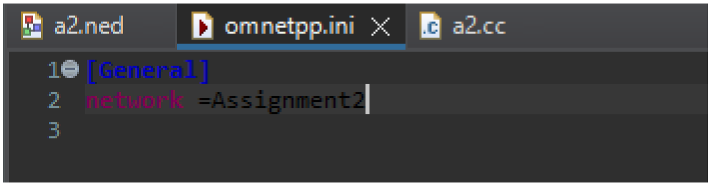
Faculty of Computing and Informatics, Universiti Malaysia Sabah
Semester 1, 2023/24
Lecturer: Shaliza Hayati A. Wahab

Assignment 2 (100 marks)

This assignment is a group of three persons. Due date: 11 January 2024, 5:00 pm. Submit all the answers and simulations codes in class website and the printout to my office (Room 65, Level 2, Block A).

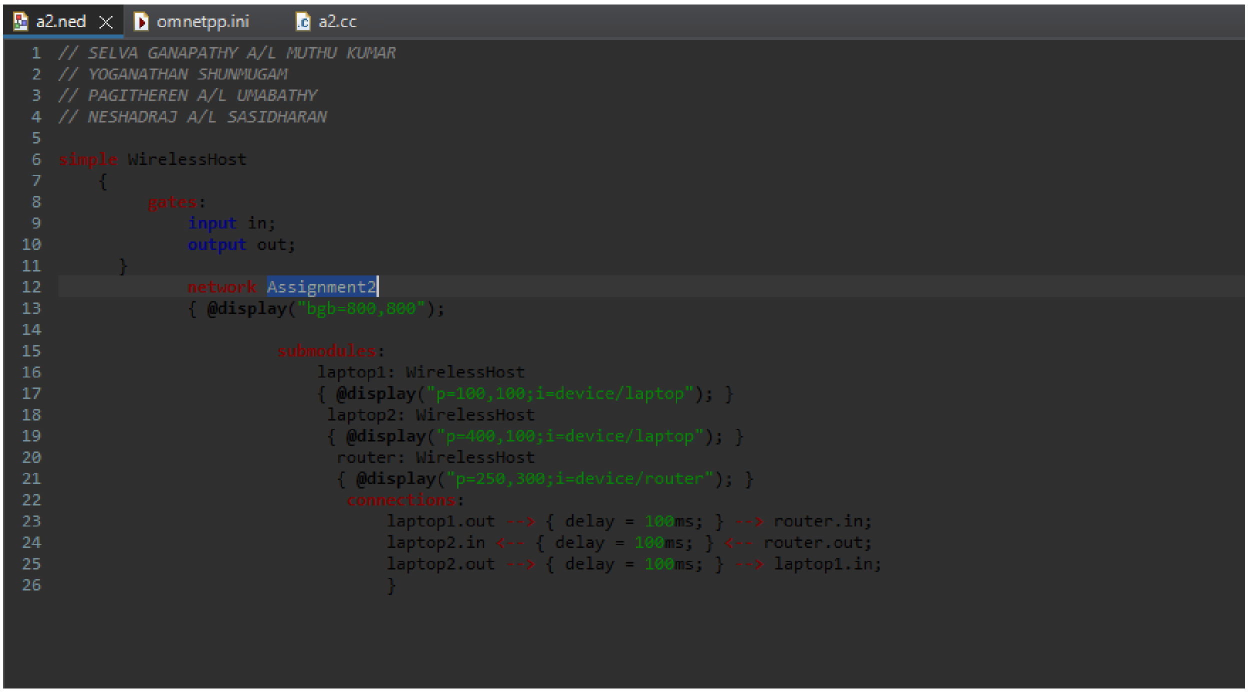
1. Create OMNeT++ simulation source codes (.ini, .ned and .cc) to test a wireless network with one wireless router and two laptops. Put your names and matric numbers in the coding.

omnetpp.ini:



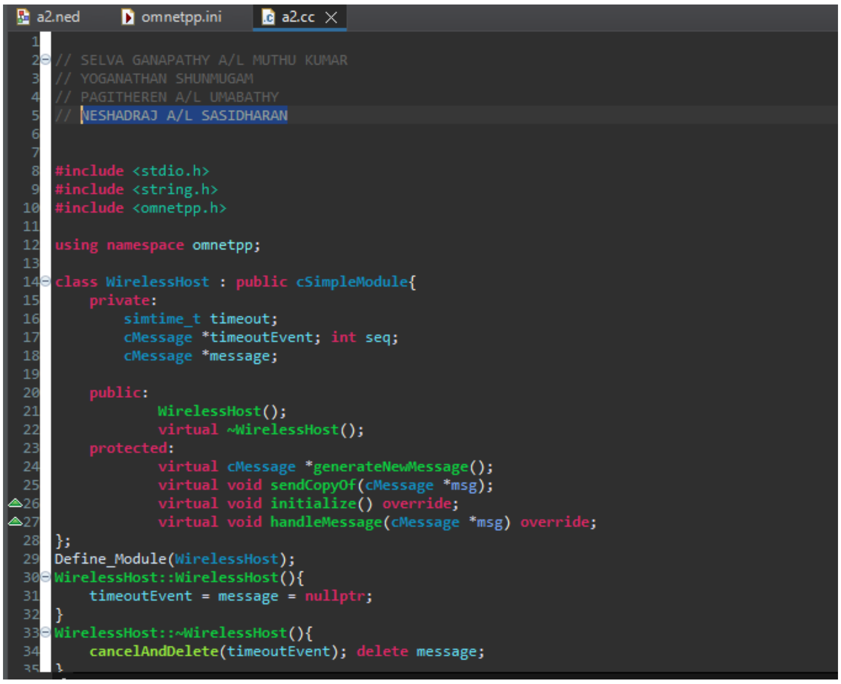
```
[General]
network =Assignment2

```

a2.ned:

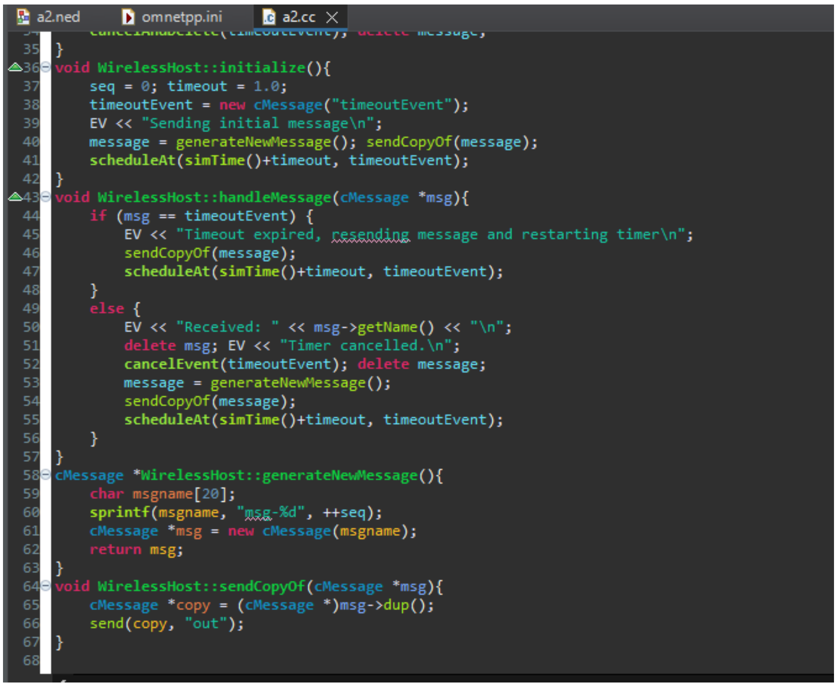


```
// SELVA GANAPATHY A/L MUTHU KUMAR
// YOGANATHAN SHUNMUGAM
// PAGITHEREN A/L UMABATHY
// NESHADRAJ A/L SASIDHARAN

simple WirelessHost
    {
         gates:
             input in;
             output out;
      }
             network Assignment2
             { @display("bgb=800,800");

                      submodules:
                          laptop1: WirelessHost
                          { @display("p=100,100;i=device/laptop"); }
                           laptop2: WirelessHost
                           { @display("p=400,100;i=device/laptop"); }
                            router: WirelessHost
                            { @display("p=250,300;i=device/router"); }
                             connections:
                                 laptop1.out --> { delay = 100ms; } --> router.in;
                                 laptop2.in <-- { delay = 100ms; } <-- router.out;
                                 laptop2.out --> { delay = 100ms; } --> laptop1.in;
                                 }
```

a2.cc:

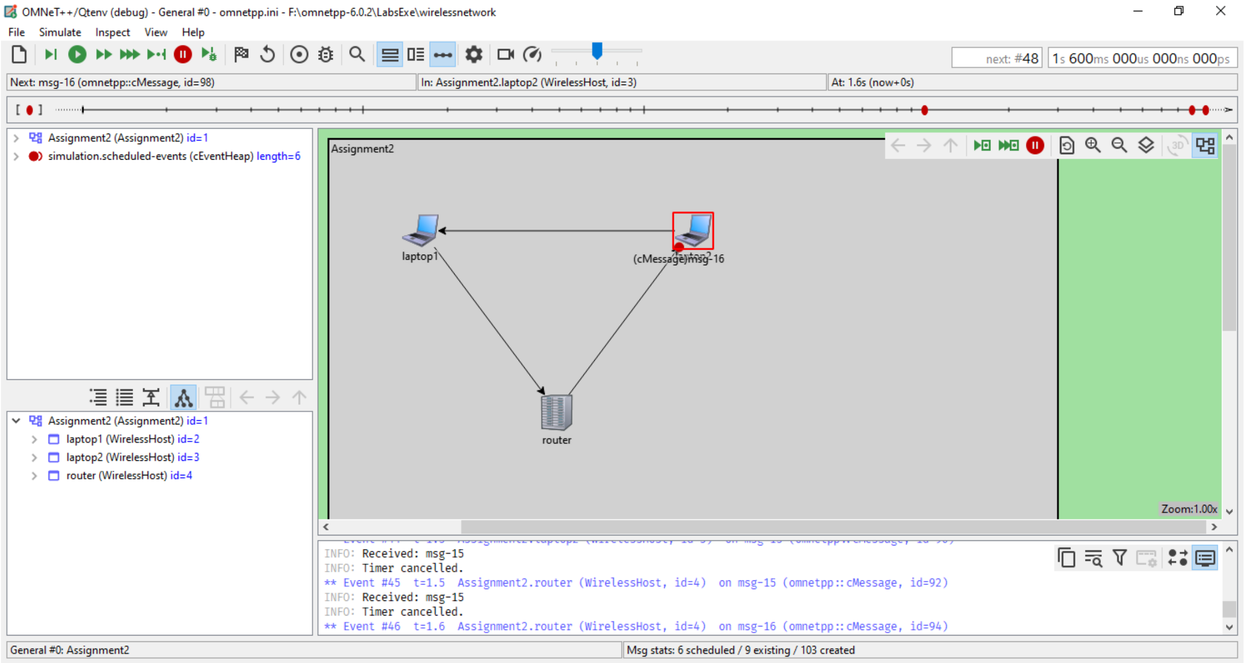


```

// SELVA GANAPATHY A/L MUTHU KUMAR
// YOGANATHAN SHUNMUGAM
// PAGITHEREN A/L UMABATHY
// NESHADRAJ A/L SASIDHARAN


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

using namespace omnetpp;

class WirelessHost : public cSimpleModule{
    private:
        simtime_t timeout;
        cMessage *timeoutEvent; int seq;
        cMessage *message;

    public:
            WirelessHost();
            virtual ~WirelessHost();
    protected:
            virtual cMessage *generateNewMessage();
            virtual void sendCopyOf(cMessage *msg);
            virtual void initialize() override;
            virtual void handleMessage(cMessage *msg) override;
};
Define_Module(WirelessHost);
WirelessHost::WirelessHost(){
    timeoutEvent = message = nullptr;
}
WirelessHost::~WirelessHost(){
    cancelAndDelete(timeoutEvent); delete message;
```



```
}
void WirelessHost::initialize(){
    seq = 0; timeout = 1.0;
    timeoutEvent = new cMessage("timeoutEvent");
    EV << "Sending initial message\n";
    message = generateNewMessage(); sendCopyOf(message);
    scheduleAt(simTime()+timeout, timeoutEvent);
}
void WirelessHost::handleMessage(cMessage *msg){
    if (msg == timeoutEvent) {
        EV << "Timeout expired, resending message and restarting timer\n";
        sendCopyOf(message);
        scheduleAt(simTime()+timeout, timeoutEvent);
    }
    else {
        EV << "Received: " << msg->getName() << "\n";
        delete msg; EV << "Timer cancelled.\n";
        cancelEvent(timeoutEvent); delete message;
        message = generateNewMessage();
        sendCopyOf(message);
        scheduleAt(simTime()+timeout, timeoutEvent);
    }
}
cMessage *WirelessHost::generateNewMessage(){
    char msgname[20];
    sprintf(msgname, "msg-%d", ++seq);
    cMessage *msg = new cMessage(msgname);
    return msg;
}
void WirelessHost::sendCopyOf(cMessage *msg){
    cMessage *copy = (cMessage *)msg->dup();
    send(copy, "out");
}

```

Simulation:



OMNeT++/Qtenv (debug) - General #0 - omnetpp.ini - F:\omnetpp-6.0.2\LabsExe\wirelessnetwork

Next: msg-16 (omnetpp::cMessage, id=98)

In: Assignment2.laptop2 (WirelessHost, id=3)

At: 1.6s (now+0s)

Assignment2 (Assignment2) id=1

simulation.scheduled-events (cEventHeap) length=6

- Assignment2 (Assignment2) id=1
  - laptop1 (WirelessHost) id=2
  - laptop2 (WirelessHost) id=3
  - router (WirelessHost) id=4

```
INFO: Received: msg-15
INFO: Timer cancelled.
** Event #45  t=1.5  Assignment2.router (WirelessHost, id=4)  on msg-15 (omnetpp::cMessage, id=92)
INFO: Received: msg-15
INFO: Timer cancelled.
** Event #46  t=1.6  Assignment2.router (WirelessHost, id=4)  on msg-16 (omnetpp::cMessage, id=94)
```

General #0: Assignment2

Msg stats: 6 scheduled / 9 existing / 103 created

Source code:

omnetpp.ini:

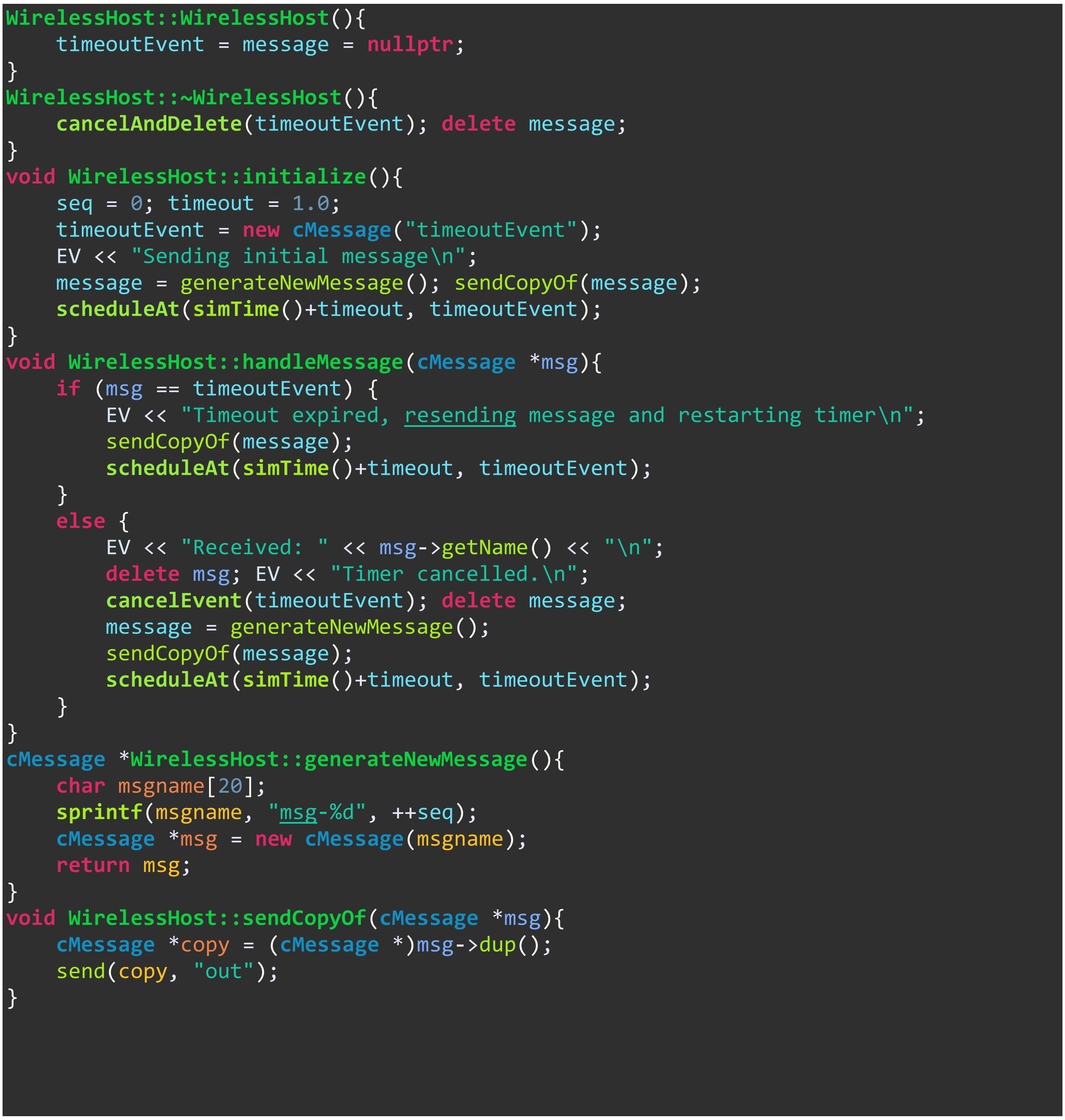
```
[General]
network = Assignment2
```

a2.ned:

```
// SELVA GANAPATHY A/L MUTHU KUMAR
// YOGANATHAN SHUNMUGAM
// PAGITHEREN A/L UMABATHY
// NESHADRAJ A/L SASIDHARAN

simple WirelessHost
    {
         gates:
             input in;
             output out;
      }
             network Assignment2
             { @display("bgb=800,800");

                      submodules:
                          laptop1: WirelessHost
                          { @display("p=100,100;i=device/laptop"); }
                           laptop2: WirelessHost
                           { @display("p=400,100;i=device/laptop"); }
                            router: WirelessHost
                            { @display("p=250,300;i=device/router"); }
                             connections:
                                 laptop1.out --> { delay = 100ms; } --> router.in;
                                 laptop2.in <-- { delay = 100ms; } <-- router.out;
                                 laptop2.out --> { delay = 100ms; } --> laptop1.in;
                                 }
```

a2.cc:

```
// SELVA GANAPATHY A/L MUTHU KUMAR
// YOGANATHAN SHUNMUGAM
// PAGITHEREN A/L UMABATHY
// NESHADRAJ A/L SASIDHARAN


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

using namespace omnetpp;

class WirelessHost : public cSimpleModule{
    private:
        simtime_t timeout;
        cMessage *timeoutEvent; int seq;
        cMessage *message;

    public:
            WirelessHost();
            virtual ~WirelessHost();
    protected:
            virtual cMessage *generateNewMessage();
            virtual void sendCopyOf(cMessage *msg);
            virtual void initialize() override;
            virtual void handleMessage(cMessage *msg) override;
};
Define_Module(WirelessHost);
```



```
WirelessHost::WirelessHost(){
    timeoutEvent = message = nullptr;
}
WirelessHost::~WirelessHost(){
    cancelAndDelete(timeoutEvent); delete message;
}
void WirelessHost::initialize(){
    seq = 0; timeout = 1.0;
    timeoutEvent = new cMessage("timeoutEvent");
    EV << "Sending initial message\n";
    message = generateNewMessage(); sendCopyOf(message);
    scheduleAt(simTime()+timeout, timeoutEvent);
}
void WirelessHost::handleMessage(cMessage *msg){
    if (msg == timeoutEvent) {
        EV << "Timeout expired, resending message and restarting timer\n";
        sendCopyOf(message);
        scheduleAt(simTime()+timeout, timeoutEvent);
    }
    else {
        EV << "Received: " << msg->getName() << "\n";
        delete msg; EV << "Timer cancelled.\n";
        cancelEvent(timeoutEvent); delete message;
        message = generateNewMessage();
        sendCopyOf(message);
        scheduleAt(simTime()+timeout, timeoutEvent);
    }
}
cMessage *WirelessHost::generateNewMessage(){
    char msgname[20];
    sprintf(msgname, "msg-%d", ++seq);
    cMessage *msg = new cMessage(msgname);
    return msg;
}
void WirelessHost::sendCopyOf(cMessage *msg){
    cMessage *copy = (cMessage *)msg->dup();
    send(copy, "out");
}
```

2. Consider the following single-server queuing system from time = 0 to time = 25 sec. Arrivals and service times are as follows:

- Customer 1 arrives at t = 1 second and requires 5 seconds of service time
- Customer 2 arrives at t = 1 second and requires 2 seconds of service time
- Customer 3 arrives at t = 2 seconds and requires 3 seconds of service time
- Customer 4 arrives at t = 12 seconds and requires 6 seconds of service time

Calculate the system throughput (X), total busy time (B), mean service time (Ts), utilization (U), mean system time (delay in system) (W), and mean number in the system (L).

a) System throughput (X) $= \frac{\text{Number of customer (completed job)}}{\text{Total time}}$

$= \frac{4}{25}$

$= 0.16$

b) Total busy time (B) $=$ Total service time for all job

$= 5 + 2 + 3 + 6$

$= 16$

c) Mean service time (Ts) $= \frac{\text{Total service time } (B)}{\text{Number customer (complete job)}}$

$= \frac{16}{4}$

$= 4$

d) Utilization (U) $= \frac{\text{Total service time } (B)}{\text{Total time}}$

$= \frac{16}{25}$

$= 0.64$

e) Mean system time (delay in system) (W)

$$= \frac{\text{Total time spend in the system (service time + waiting time)}}{\text{Number of customer (completed job)}}$$

$$= \frac{[(1+0+5)+(1+5+2)+(2+7+3)+(12+13+6)]}{4}$$

$= 14.25$

f) Mean number in the system (L) $=$ Utilization (U) x Mean system time (W)

$= 0.64 \text{ x } 14.25$

$= 9.12$

GitHub link:

GitHub Repo