



NP HW2 GUIDE

jdchen

11.28.2012

Outline

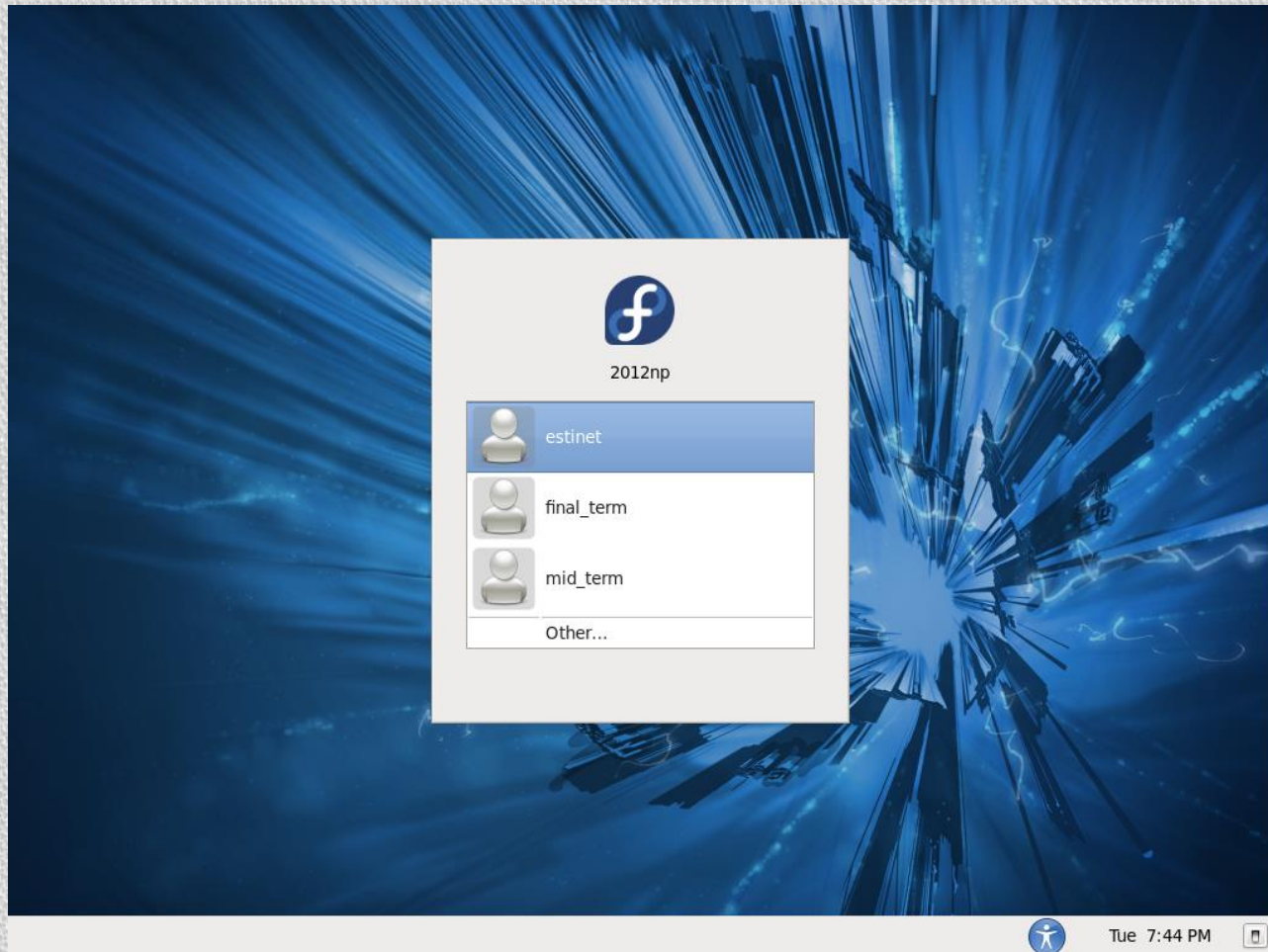
- Login and Environment Setup
- EstiNet Network Simulator
- Simulation Topology Setup
- Run Simulation
- Verification
- Other Important issue
- HW2 Grading Policy

Before we start...

- It is suggested that you should make a new directory whenever you login a VM
- Use that directory as your working directory.
- When you're finished, use USB stick/ Network to backup your code
- Delete the working directory when you leave

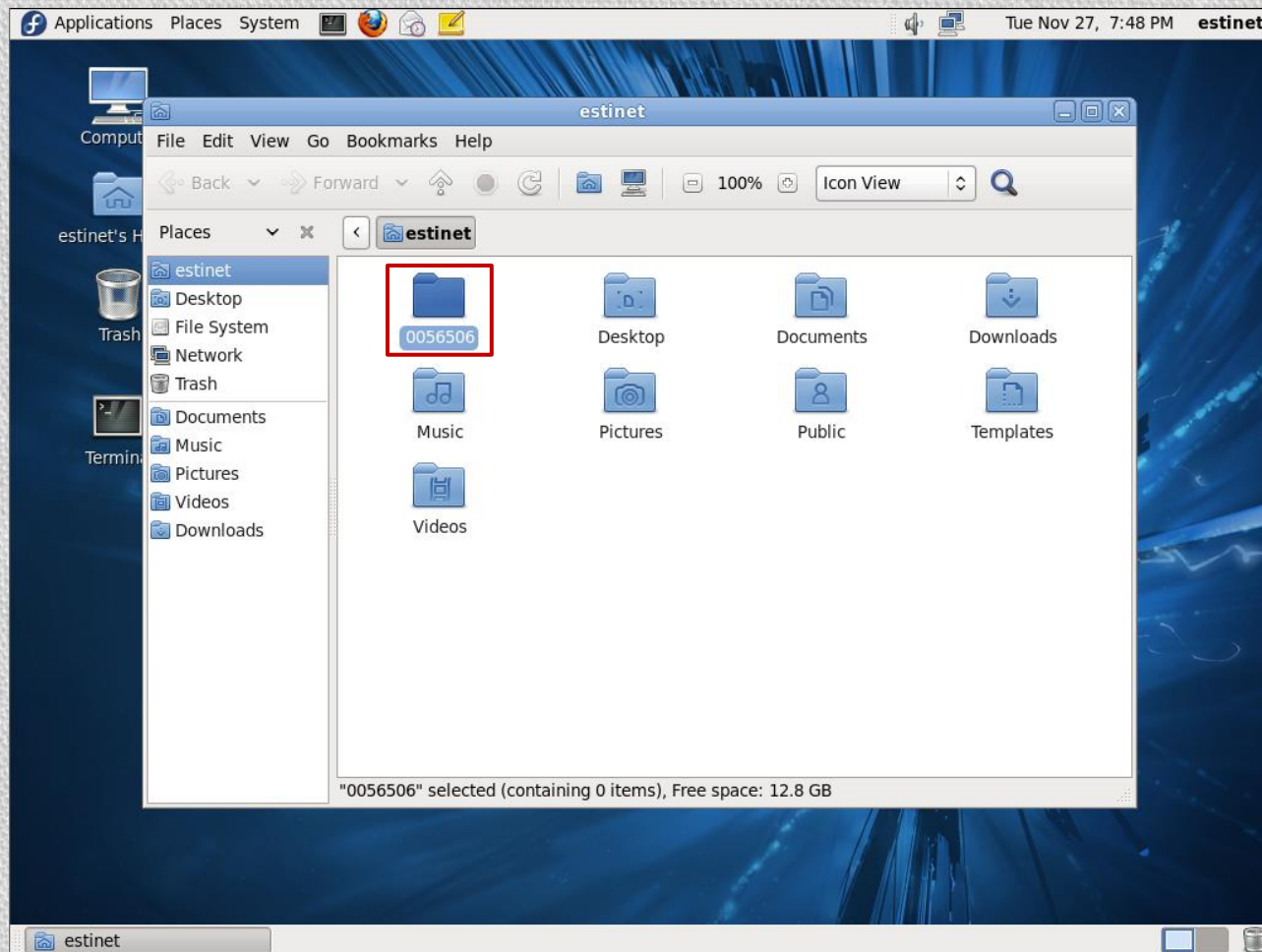
Login

- Account: estinet
- Password: estinet



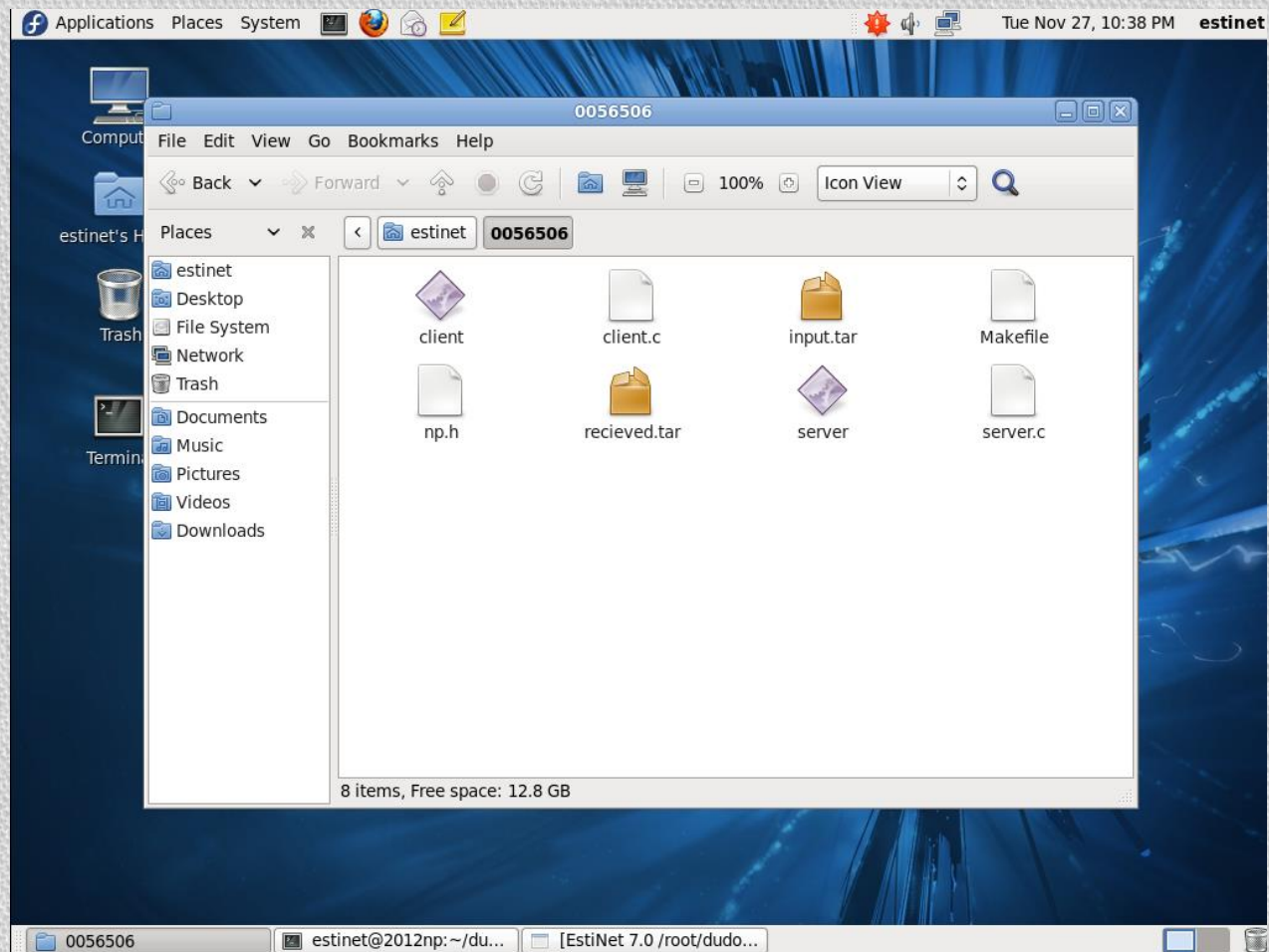
Create your own directory

- You should create your own working directory under /home/estinet



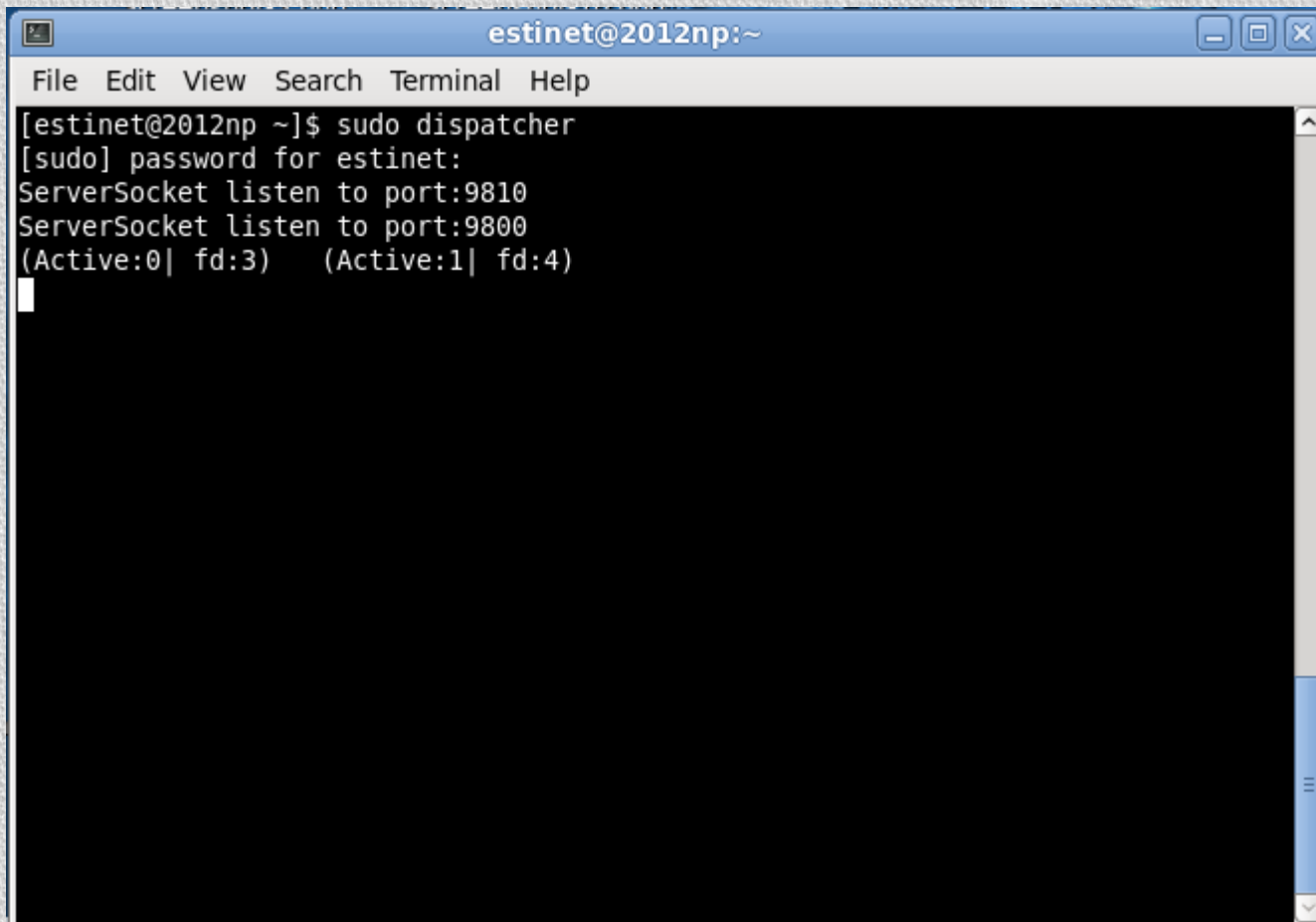
Use your own directory as working env.

- And put all your code in the corresponding directory



Start dispatcher

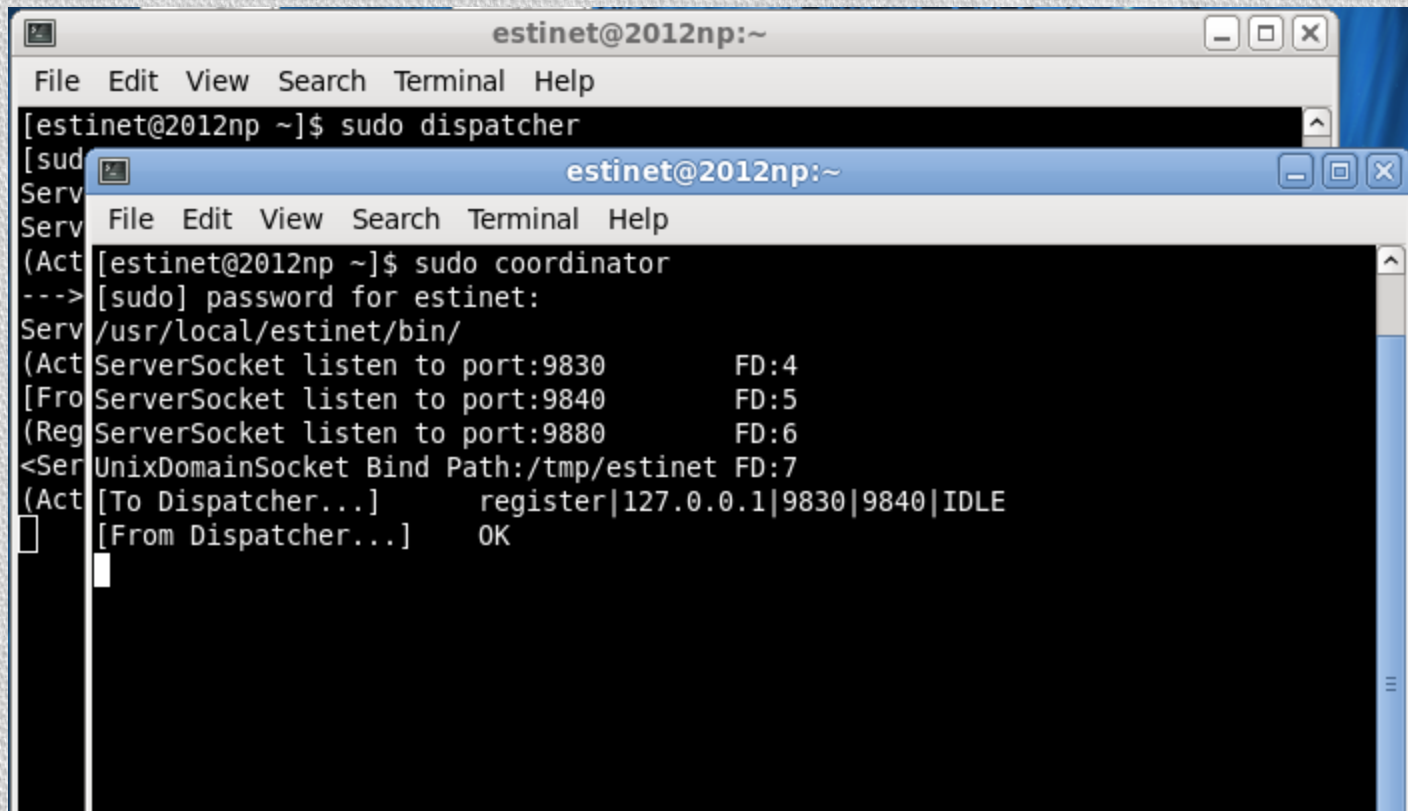
- dispatcher is one of the components of EstiNet
- `sudo dispatcher`

A terminal window titled 'estinet@2012np:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the command 'sudo dispatcher' being executed. It prompts for a password, which is entered. The output shows 'ServerSocket listen to port:9810' and 'ServerSocket listen to port:9800'. The prompt then changes to '(Active:0| fd:3) (Active:1| fd:4)' and a cursor is visible on the next line.

```
estinet@2012np:~  
File Edit View Search Terminal Help  
[estinet@2012np ~]$ sudo dispatcher  
[sudo] password for estinet:  
ServerSocket listen to port:9810  
ServerSocket listen to port:9800  
(Active:0| fd:3) (Active:1| fd:4)  
█
```


Start coordinator

- coordinator is one of the component of EstiNet
- sudo coordinator

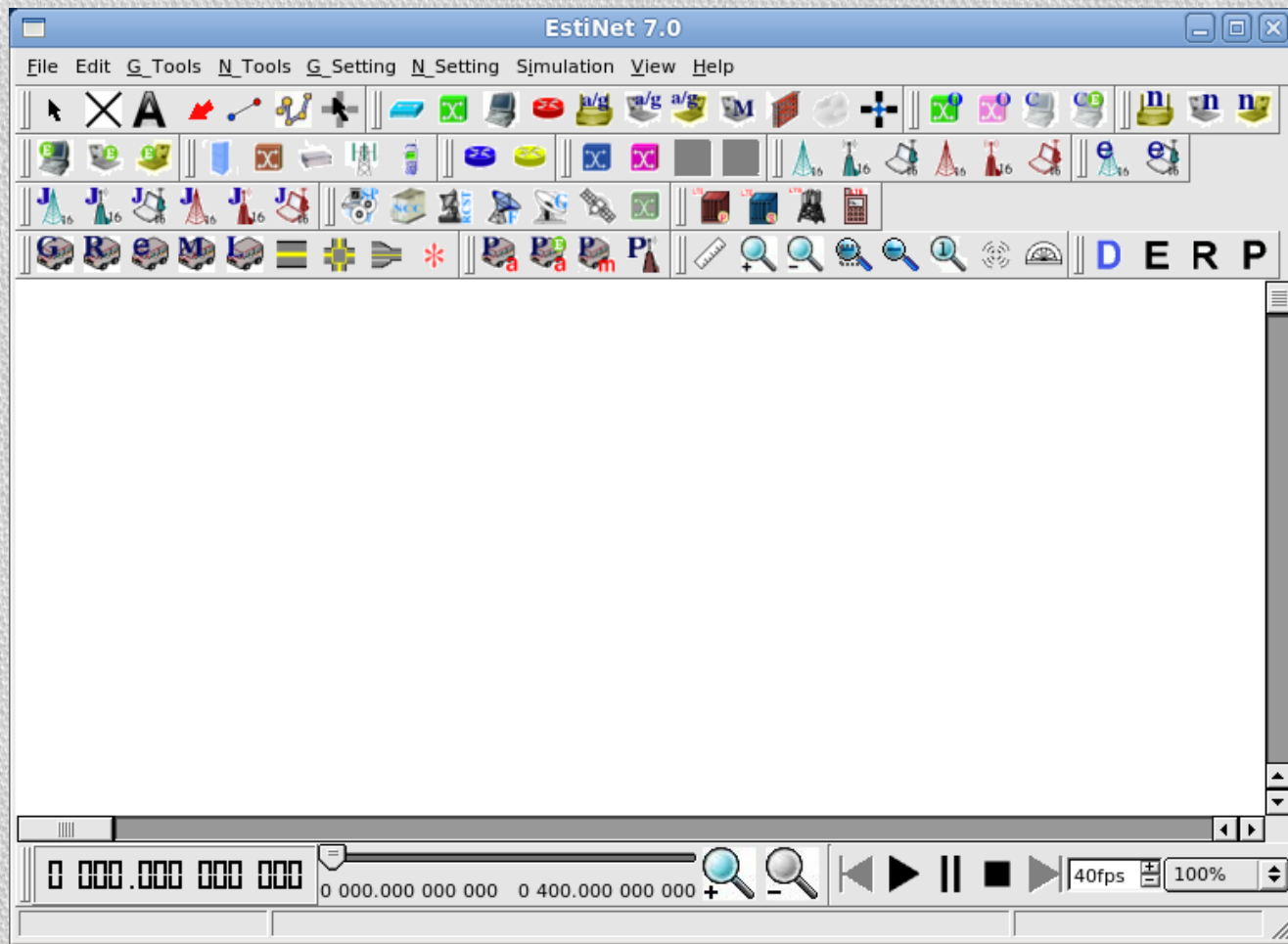


The image shows two overlapping terminal windows. The background window is titled 'estinet@2012np:~' and shows the command 'sudo dispatcher' being entered. The foreground window, also titled 'estinet@2012np:~', shows the command 'sudo coordinator' being entered, followed by a password prompt. The output of the 'sudo coordinator' command is displayed, showing the coordinator's configuration and its registration with the dispatcher.

```
estinet@2012np:~  
File Edit View Search Terminal Help  
[estinet@2012np ~]$ sudo dispatcher  
[sudo]  
Serv  
Serv  
File Edit View Search Terminal Help  
(Act [estinet@2012np ~]$ sudo coordinator  
---> [sudo] password for estinet:  
Serv /usr/local/estinet/bin/  
(Act ServerSocket listen to port:9830      FD:4  
[From ServerSocket listen to port:9840      FD:5  
(Reg ServerSocket listen to port:9880      FD:6  
<Ser UnixDomainSocket Bind Path:/tmp/estinet FD:7  
(Act [To Dispatcher...]    register|127.0.0.1|9830|9840|IDLE  
[From Dispatcher...]    OK
```

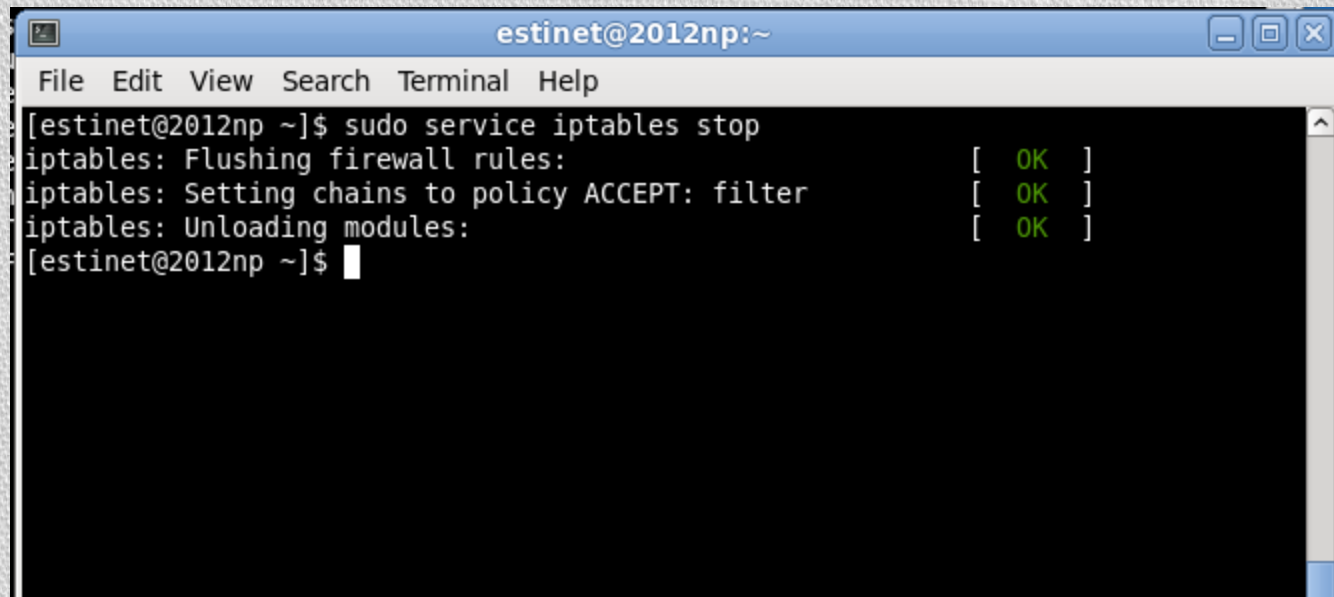

Start estinetgui

- estinetgui is where you can assign your network situation
- `sudo estinetgui`



Shutdown iptables service

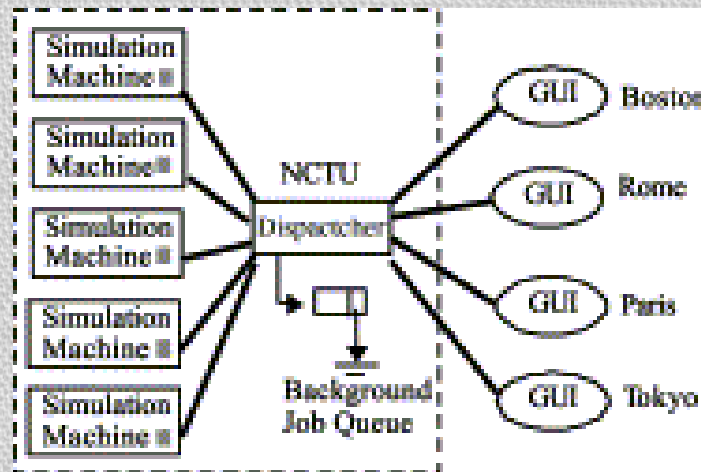
- Iptables is a default firewall in fedora system. As EstiNet produces real world network flows, iptables will treat these flows as malicious packets and drop.
- `sudo service iptables stop`



```
estinet@2012np:~  
File Edit View Search Terminal Help  
[estinet@2012np ~]$ sudo service iptables stop  
iptables: Flushing firewall rules:      [ OK ]  
iptables: Setting chains to policy ACCEPT: filter [ OK ]  
iptables: Unloading modules:            [ OK ]  
[estinet@2012np ~]$
```


EstiNet

- estinetgui
 - An user interface for network topology setup
- coordinator
 - connect to dispatcher and bring up simulation engine
- dispatcher
 - As there can be more than one simulation engine to be used, the dispatcher manages the simulation engine communication and GUI requests.



Simulation Topology Setup

- There are 4 modes in estinetgui

|| D E R P

- Draw, Edit, Run, Play back

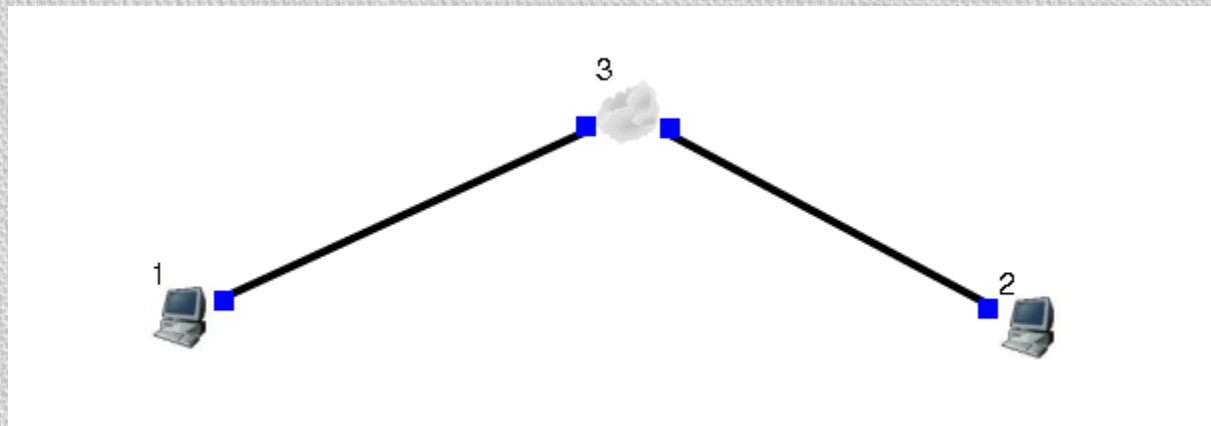
- Use draw mode to draw network topology



- We'll use ethernet for this homework



- Use 2 host and 1 WAN to create a topology like below



Simulation Topology Setup



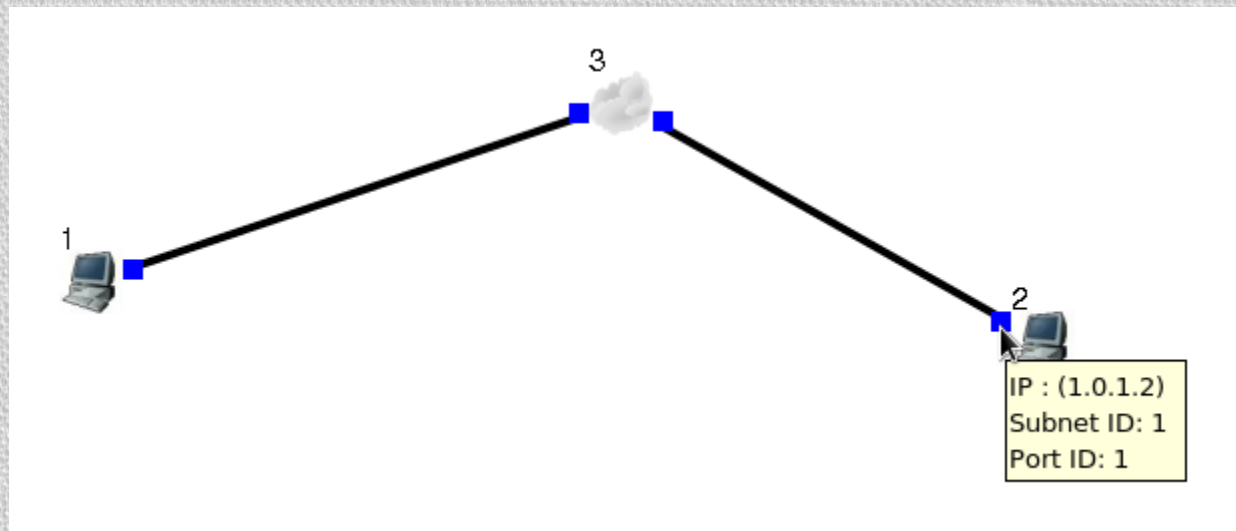
- Switch to E mode
 - This mode can set parameters of the topology we just designed.
- We will need to setup 5 different parameters:
 - Setup the program which will be run on the topology
 - Link delay and link bandwidth
 - Packet loss rate
 - Log the throughput of each node
 - Random number seed

Setup parameters

- We don't need to change any of our code to run simulation in EstiNet
- Copy your executable program to `/usr/local/estinet/tools`
 - But in this homework, you only need to paste your program in `/tmp/np/`
 - `cp server /tmp/np/`
 - `cp client /tmp/np/`
- Specify the program name of your own, and it will work like a charm

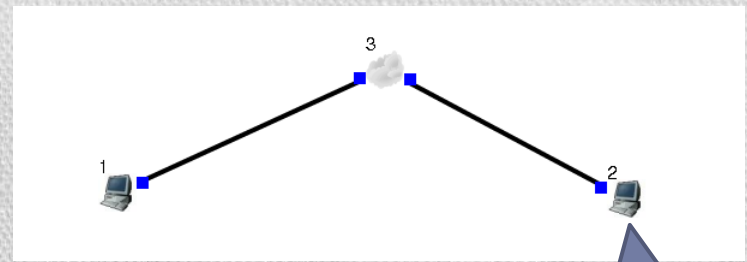
Setup parameters

- Hold your mouse on the square beside the host for a moment, you will get the IP of the corresponding host.



Setup parameters

- Double click the host, you will get the window below
- Click Add button



Double
click

host

Host ID Name

Application | Down time | Mobile IP | Emulation

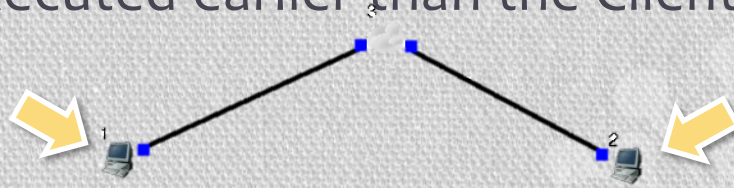
Start time(s)	Stop time(s)	Command	Input f
---------------	--------------	---------	---------

Add
Modify
Add RTP
Modify RTP
Delete
App. Usage

Command console Node editor OK Cancel

Setup parameters

- You should fill the name of your program on both sides.
- Server should be executed earlier than the Client.



Traffic

Start time (sec) Stop time (sec)

Command

Input file name

Please note that the starting time for a server program should be set to a time that is earlier than the starting time set for a client program. Otherwise, the client cannot connect to the server immediately and its TCP connection may unnecessarily timeout. For example, if you let rtcp (server) start at time 0 second, it is better to let stcp (client) start at time 0.1 second.

Traffic

Start time (sec) Stop time (sec)

Command

Input file name

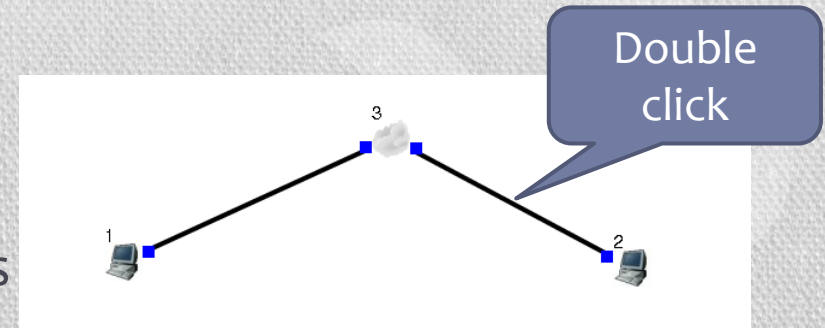
Please note that the starting time for a server program should be set to a time that is earlier than the starting time set for a client program. Otherwise, the client cannot connect to the server immediately and its TCP connection may unnecessarily timeout. For example, if you let rtcp (server) start at time 0 second, it is better to let stcp (client) start at time 0.1 second.

Setup parameters

- We need to specify the network link delay, bandwidth.
- The default link delay is 10 us, bandwidth is 10 Mbps
- Topology we need in this homework are
 - Link delay: 10000 us (10 ms)
 - Link bandwidth: 100 Mbps

Setup parameters

- Double click the link between nodes
- Delay: 10000 us (10 ms)
- Bandwidth: 100 Mbps
- Click C.T.A.L (copy to all links)



LINK

From WAN3 to Host2

Delay : (us)

Bandwidth: (Mbps)

BER :

Down time

Start (s)	End (s)	
-----------	---------	--

From Host2 to WAN3

Delay: (us)

Bandwidth: (Mbps)

BER :

Down time

Start (s)	End (s)	
-----------	---------	--

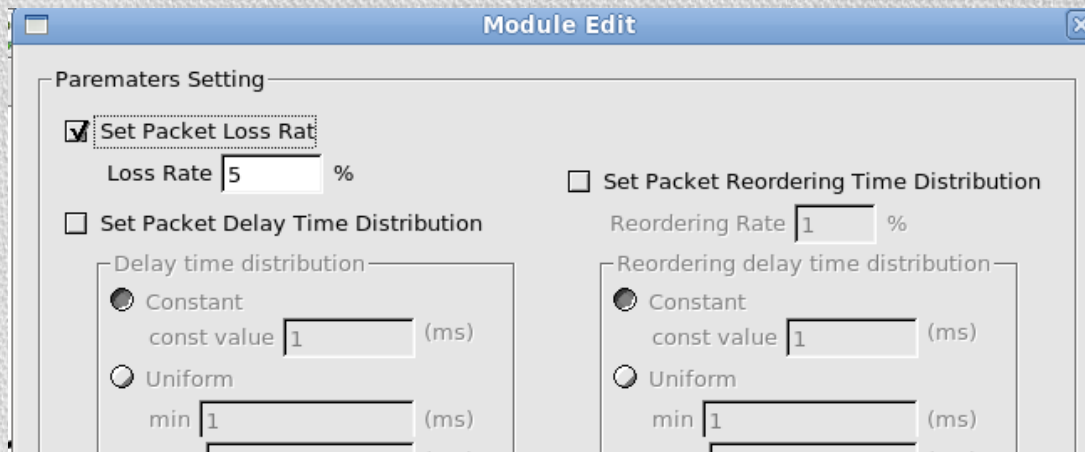
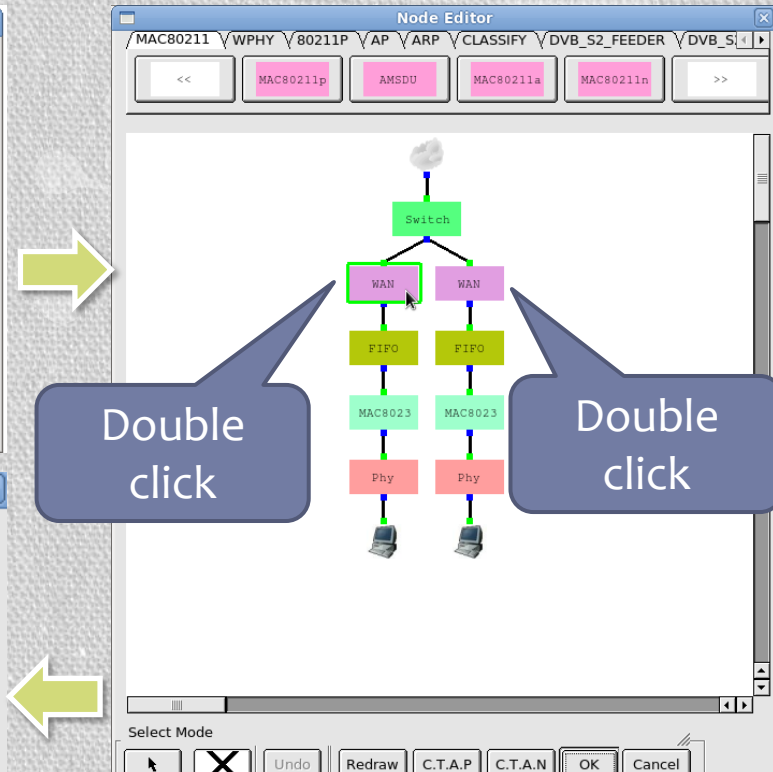
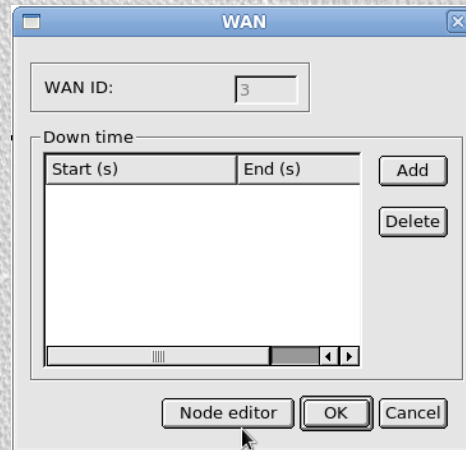
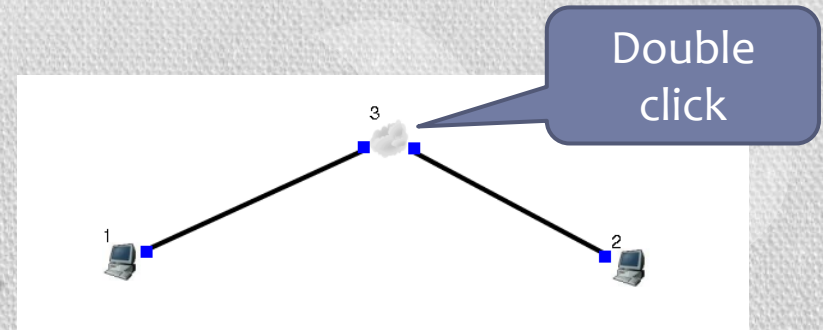
Copy to All Links

Setup parameters

- We will need to setup a packet lost network simulation
- We use cloud(WAN) to archive that need.

Setup parameters

- Double click the cloud (WAN)
- Click “Node editor”
- Double click “WAN”
- Set Packet Loss Rate
 - You can try different values from 1 to 10%
 - **Set on both side**

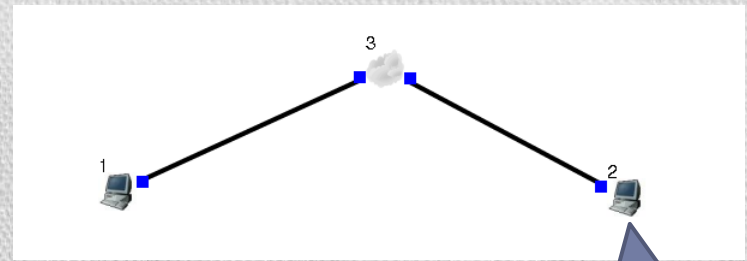


Setup parameters

- We will need to understand how many network resource does your program consume.
- EstiNet has the option to log your throughput during execute time.

Setup parameters

- Double click the host, you will get the window below
- Click “Node editor” button below



Double
click

host

Host ID Name

Application | Down time | Mobile IP | Emulation

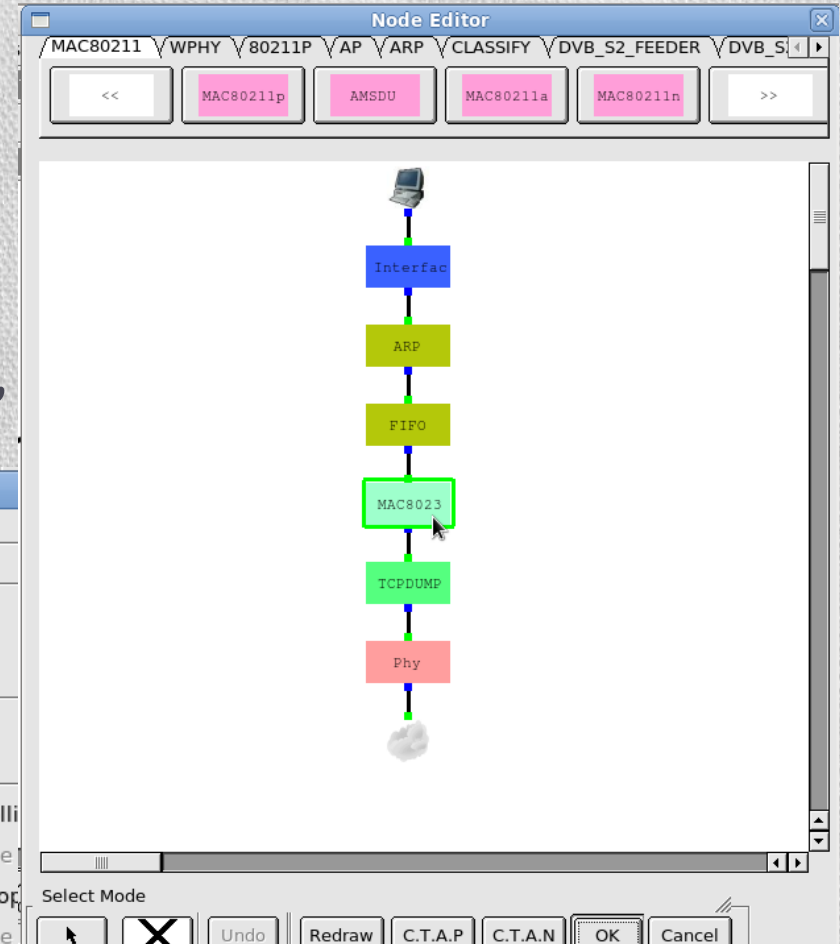
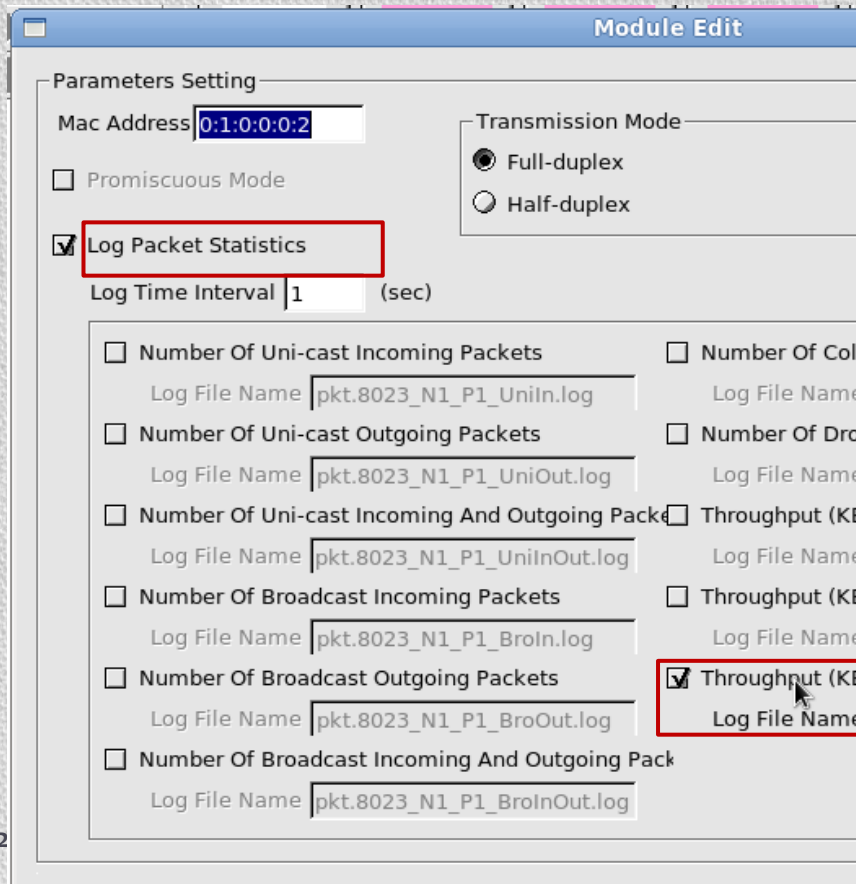
Start time(s)	Stop time(s)	Command	Input f
---------------	--------------	---------	---------

Add
Modify
Add RTP
Modify RTP
Delete
App. Usage

Command console Node editor OK Cancel

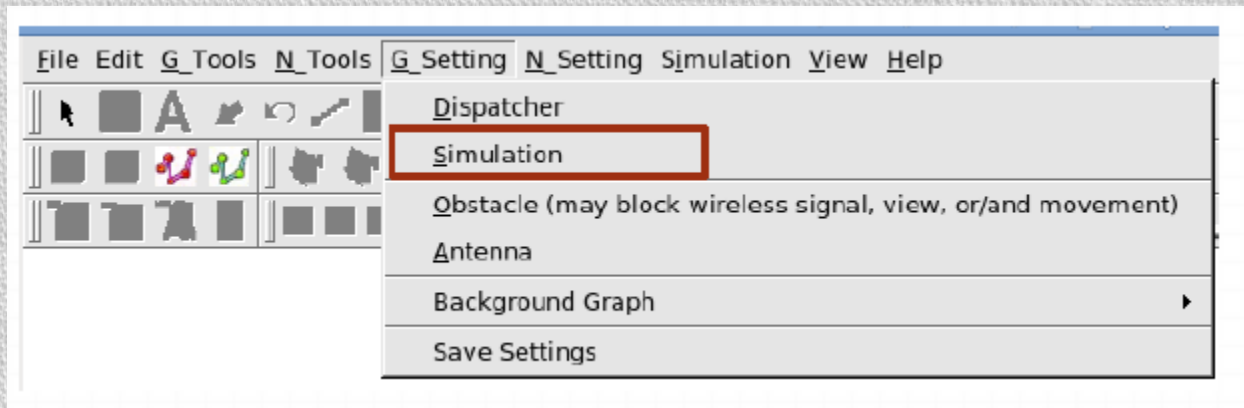
Setup parameters

- Double click “MAC 8023”
- “Log packet statistics”
 - “Throughput (KB/sec) of in/out Packet”



Setup parameters

- You have to know three options and parameters in this homework
 - **Simulation time, random seed, and simulation speed**
- Find them in G_setting -> simulation



Setup parameters

- Simulation time
 - The time in the simulation network.
- You don't need to change it until your program need more time to finish it work.

Simulation

Simulation | Speed | Real Time | GUI | GDB Debug | System

Simulation time: 400 sec(s)

Max_X: 3000 meter(s)

Max_Y: 3000 meter(s)

Max_Z: 0 meter(s)

☒ Generate packet animation log file (*.ptr)

- ☒ Fixed Internet
- ☒ Wireless LAN
- ☒ GPRS network
- ☒ Optical network
- ☒ WiMax network
- ☒ MobileWiMax network
- ☒ MobileRelayWiMax network (transparent mode)
- ☒ MobileRelayWiMax network (non-transparent mode)
- ☒ LTE network

Random Number Seed : 0

(Using the default value 0 means that the seed will be automatically generated.)

OK Cancel

Setup parameters

- Random number seed
 - If you want your simulation case is repeatable, you should set it to a non 0 number
 - If the number is non 0, the result of a case will be always the same on each run.
 - If the number is 0, the result of a case will be different after each run.

Simulation

Simulation | Speed | Real Time | GUI | GDB Debug | System

Simulation time: 400 sec(s)

Max_X: 3000 meter(s)

Max_Y: 3000 meter(s)

Max_Z: 0 meter(s)

☒ Generate packet animation log file (*.ptr)

- ☒ Fixed Internet
- ☒ Wireless LAN
- ☒ GPRS network
- ☒ Optical network
- ☒ WiMax network
- ☒ MobileWiMax network
- ☒ MobileRelayWiMax network (transparent mode)
- ☒ MobileRelayWiMax network (non-transparent mode)
- ☒ LTE network

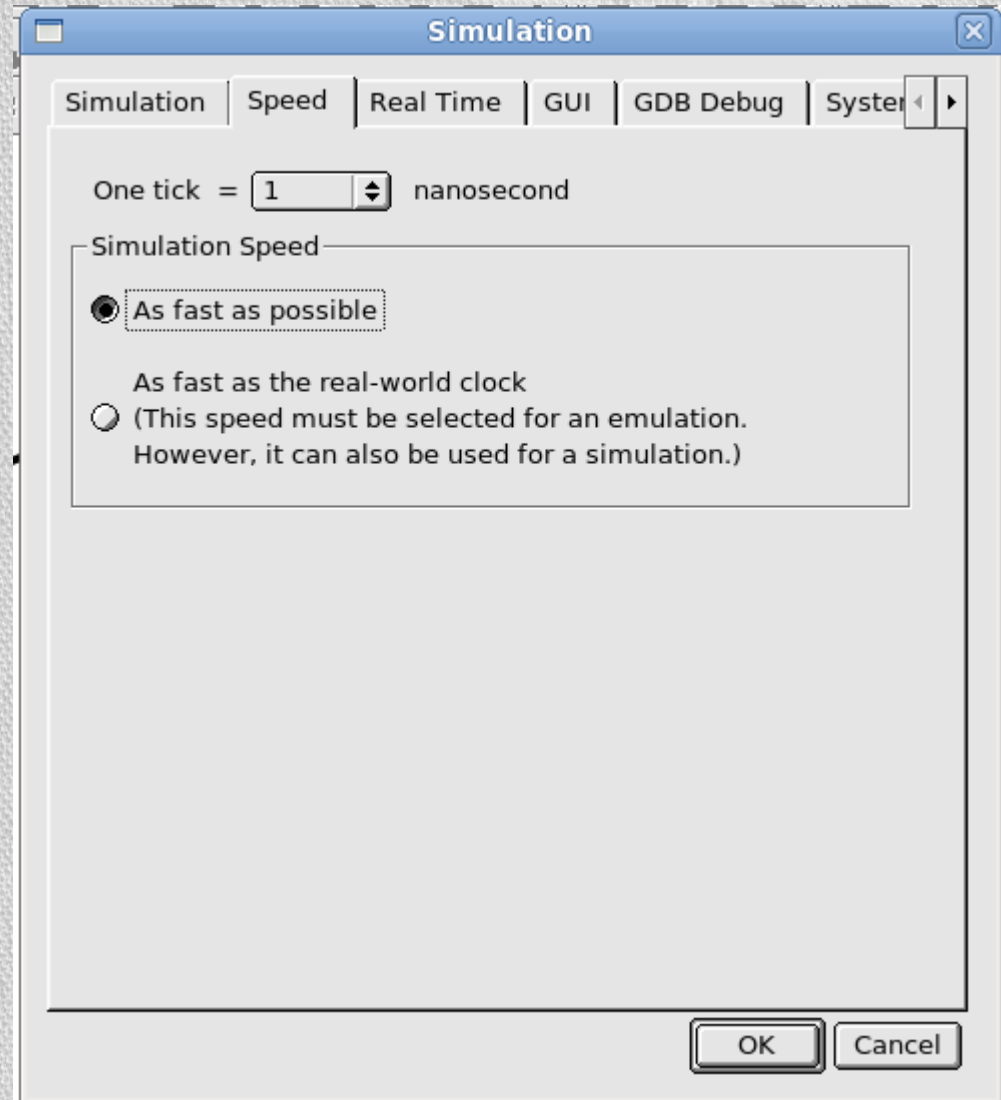
Random Number Seed : 0

(Using the default value 0 means that the seed will be automatically generated.)

OK Cancel

Setup parameters

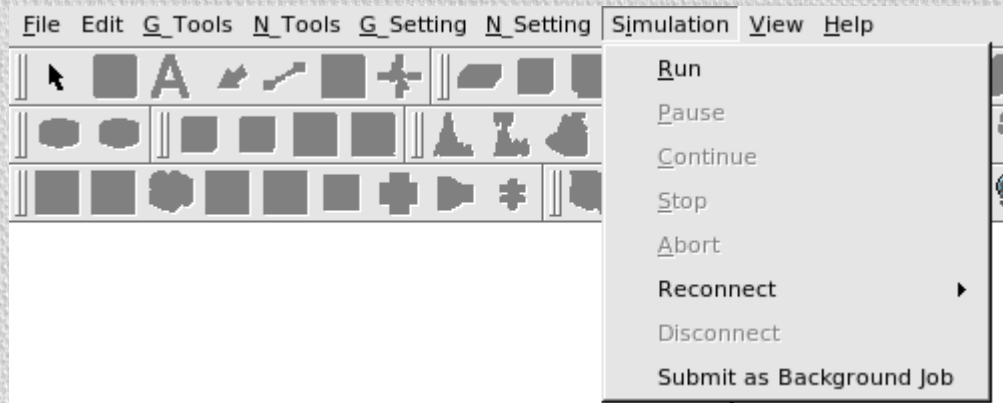
- Simulation speed:
 - If you want to use command console, you have to use “As fast as the real-word clock” mode.
 - Otherwise, “As fast as possible”, mode can save your time.



Run simulation



- Use “R” mode
- Simulation -> Run



- If you want to stop before simulation is done
- Simulation -> Stop

Log time

- You should use gettimeofday to examine how long is the transfer time.
- Use gettimeofday before and after the receiving data loop.
 - #include <sys/time.h>

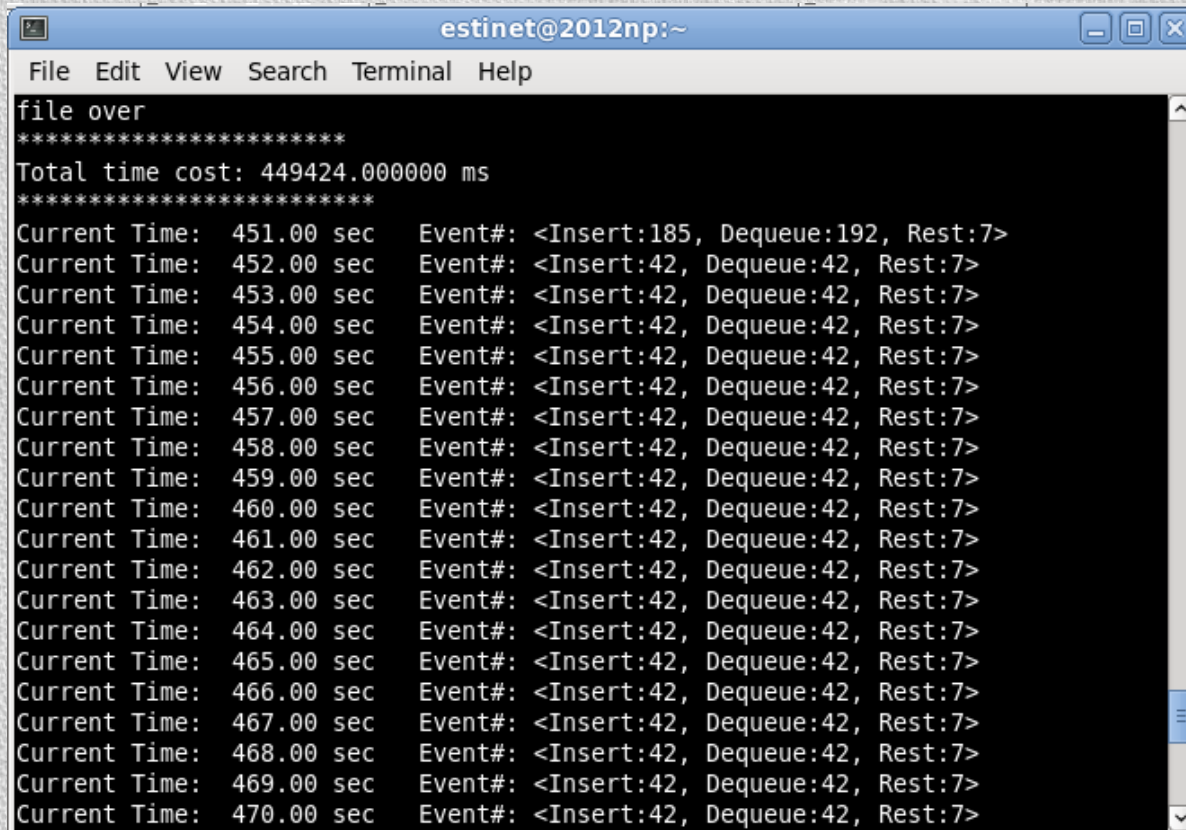
```
struct timeval timeintv_a, timeintv_b;  
// get start time  
if(gettimeofday(&timeintv_a, NULL)!=0){  
    printf("gettimeofday failed\n");  
    exit(1);  
}
```

• Receiving Loop{}

```
//get end time  
if(gettimeofday(&timeintv_b, NULL)!=0){  
    printf("gettimeofday failed\n");  
    exit(1);  
}  
timecost = (timeintv_b.tv_sec - timeintv_a.tv_sec) * 1000;  
timecost += (timeintv_b.tv_usec - timeintv_a.tv_usec) / 1000;  
  
printf("*****\nTotal time cost: %f ms\n*****\n", timecost);
```


Log time

- Your program output will be shown on the coordinator window.
- You should check coordinator window to understand your time consumed.



```
estinet@2012np:~
File Edit View Search Terminal Help
file over
*****
Total time cost: 449424.000000 ms
*****
Current Time: 451.00 sec Event#: <Insert:185, Dequeue:192, Rest:7>
Current Time: 452.00 sec Event#: <Insert:42, Dequeue:42, Rest:7>
Current Time: 453.00 sec Event#: <Insert:42, Dequeue:42, Rest:7>
Current Time: 454.00 sec Event#: <Insert:42, Dequeue:42, Rest:7>
Current Time: 455.00 sec Event#: <Insert:42, Dequeue:42, Rest:7>
Current Time: 456.00 sec Event#: <Insert:42, Dequeue:42, Rest:7>
Current Time: 457.00 sec Event#: <Insert:42, Dequeue:42, Rest:7>
Current Time: 458.00 sec Event#: <Insert:42, Dequeue:42, Rest:7>
Current Time: 459.00 sec Event#: <Insert:42, Dequeue:42, Rest:7>
Current Time: 460.00 sec Event#: <Insert:42, Dequeue:42, Rest:7>
Current Time: 461.00 sec Event#: <Insert:42, Dequeue:42, Rest:7>
Current Time: 462.00 sec Event#: <Insert:42, Dequeue:42, Rest:7>
Current Time: 463.00 sec Event#: <Insert:42, Dequeue:42, Rest:7>
Current Time: 464.00 sec Event#: <Insert:42, Dequeue:42, Rest:7>
Current Time: 465.00 sec Event#: <Insert:42, Dequeue:42, Rest:7>
Current Time: 466.00 sec Event#: <Insert:42, Dequeue:42, Rest:7>
Current Time: 467.00 sec Event#: <Insert:42, Dequeue:42, Rest:7>
Current Time: 468.00 sec Event#: <Insert:42, Dequeue:42, Rest:7>
Current Time: 469.00 sec Event#: <Insert:42, Dequeue:42, Rest:7>
Current Time: 470.00 sec Event#: <Insert:42, Dequeue:42, Rest:7>
```


HW requirements

- Execution format
 - ./server [port]
 - ./client [ip] [port]
- Your program should read the input file place in /tmp/input.tar
 - That is, function should be set as `fp = fopen("/tmp/input.tar", "rb");`
 - If you're developing, use your own file.
- The received file should be placed in your own folder
 - Name as "received.tar"
- **Write a simple report about your design(in 400 words)**

HW requirements

- Homework parameters
 - Delay between 100 ms
 - Drop rate between 5% (for both data packet and ack packet)
 - Test file size will be approximately 20 MB
 - All students will be having the same random number seed.
- Grading policy
 - We will run the simulation three times, each using different timeout method you implemented. Each program should produced the correct result.
 - We will test only your correctness, time/ transfer size efficiency will not be concern.
- Due day 12/14(Wed.)