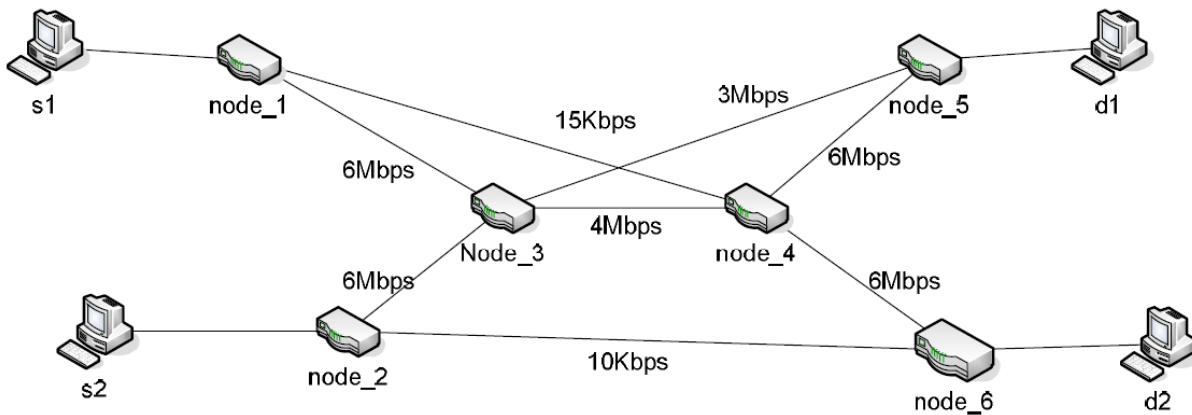


I. Đề bài và yêu cầu

Cho mạng thông tin với cấu hình như Hình 1 dưới đây. Nút s1 phát ra luồng gói gửi tới đầu cuối d1, trong khi nút s2 tạo ra hai luồng gói gửi tới nút d1 và tới nút d2. Cả 3 luồng thông tin nói trên đều sử dụng giao thức truyền UDP. Kích thước gói do s1 và s2 phát ra tương ứng là 1300 byte và 1000 byte; số lượng gói phát ra trên mỗi luồng tuân theo phân bố Poisson, tốc độ lần lượt là 1000 gói/s, 1500 gói/s và 2000 gói/s.



Hình 1

1. Dụng kịch bản mô phỏng hệ thống thông tin nói trên với thời gian mô phỏng dài hơn 5 phút.
2. Thu thập số liệu mô phỏng và vẽ đồ thị thời gian trễ tại mỗi bộ định tuyến.
3. Trong một phiên chạy mô phỏng, tốc độ phát sinh gói của mỗi luồng tăng đi 100 gói/s sau mỗi khoảng thời gian 30s, thống kê và vẽ đồ thị tỷ lệ mất gói tại các bộ định tuyến trong mỗi khoảng thời gian đó.
4. Thay đổi tham số tốc độ phát sinh và kích thước gói theo chiều biến thiên ngược nhau, vẽ đồ thị biểu diễn độ trễ truyền thông của các luồng. Nhận xét kết quả thu được.

II. Phân tích yêu cầu

1. Phân tích đề bài

Bài toán được đặt ra có một số số liệu đầu và hoặc tham số không nêu trong đề bài, vì vậy để thực hiện được cần tùy chọn một số giá trị tham số như sau: các máy tính (node s1 và s2) nối với mạng bằng liên kết Ethernet 100Mb. Trễ truyền lan trên tất cả các liên kết là 10ns. Chiều dài mặc định của tất cả các hàng đợi là 50.

Bài toán yêu cầu xây dựng mạng thông tin với 3 luồng dữ liệu UDP từ 2 node nguồn đến 2 đích thông qua 6 node mạng (router). Yêu cầu cụ thể phải thực hiện trên phần mềm NS2: Dựng được kịch bản mô phỏng hệ thống, vẽ đồ thị thời gian trễ tại mỗi router, thay đổi tốc độ phát sinh gói, vẽ đồ thị tỷ lệ mất gói tại các router, vẽ đồ thị biểu diễn độ trễ truyền thông.

Để thực hiện được các yêu cầu đó thì các thành viên trong nhóm cần phải:

- Cài đặt, tìm hiểu và sử dụng tốt hệ điều hành ubuntu, Bộ công cụ mô phỏng ns2
- Tìm hiểu về định tuyến
- Thiết lập được số lượng gói phát sinh theo phân bố Poisson
- Tìm hiểu về các giao thức truyền thông UDP
- Tìm hiểu về công cụ xử lý số liệu và vẽ đồ thị trên ns2

2. Phương hướng thực hiện yêu cầu trên phần mềm NS2

a. Tạo phân bố Poisson

Trong Ns2 mặc định khi không có sẵn traffic Poisson để có thể gắn vào agent UDP để mô phỏng lưu lượng mạng. Tuy nhiên Ns2 lại có sẵn traffic Exponential – thay đổi theo phân bố mũ. Do đó, dựa vào mối quan hệ giữa phân bố Poisson và phân bố Exponential, ta có thể xây dựng tiến trình Poisson qua một đối tượng ExponentialOn/Off với các thành phần tham số:

. packet_size_ dung lượng cố định của gói dữ liệu

. burst_time_ thời gian “bật” khởi tạo

. idle_time_ thời gian “tắt” khởi tạo

. rate_ tốc độ của gói

Để tạo ra tiến trình Poisson ta cho tham số burst_time_ về 0 và idle_time bằng $\frac{1}{\lambda}$ của tiến trình

Poisson.

b. Tính thời gian trễ tại mỗi bộ định tuyến

Thời gian trễ tại mỗi bộ định tuyến được tính từ lúc node đó bắt đầu nhận gói tin từ node trước đến khi node đó gửi gói tin ấy đi sang node khác. Để tính thời gian này ta sẽ tạo một file traceAll.tr để lưu vết tất cả các sự kiện xảy ra. Sau đó sẽ xây dựng một hàm lưu trong file .awk để trích xuất dữ liệu thời gian gói tin đến và đi ở các node từ trace-file .tr.

c. Tăng gói tin

Để tăng tốc độ gói tin sau mỗi khoảng thời gian 30 s ta dùng 1 thủ tục và trong thủ tục này sẽ có câu lệnh gọi lại chính nó sau mỗi 30s. Như vậy ta chỉ cần gọi thủ tục này 1 lần là được. Về phương pháp ta thực hiện như sau.:

- Nguồn s1 phát các gói tin với tốc độ là
 $1300(\text{byte}/\text{gói}) \times 1000 (\text{gói}/\text{s}) = 1.3 \text{ MBps}$
- Nguồn s2 phát ra 2 luồng với các gói tin tốc độ là 1.5MBps và 2MBps
- Cứ sau 30s nguồn s1 và s2 tăng tốc độ gói là 100 gói/s nên ta có lượng tăng sau 30s là:

$$s1: 100 * 1300 = 130000 \text{ bytes}$$

$$s2: 100 * 1000 = 100000 \text{ bytes (2 luồng)}$$

- Khi thay đổi số gói (λ) sẽ dẫn đến idle_time_ và rate_ thay đổi. Cách tính idle_time_ như sau:

$$\frac{1}{\text{idle_time_}} = \lambda, \text{ vậy sau khi giảm } \lambda \text{ thì}$$

$$\text{idle_time_} = \frac{1}{\lambda + 100}$$

- Rate được tính như sau:
 S1: rate_ = rate_ + 130000 (bytes)
 S2: rate_ = rate_ + 100000 (bytes)

d. Tính tỉ lệ mất gói tại mỗi bộ định tuyến :

Tính tỉ lệ mất gói tại một bộ định tuyến bằng cách lấy số gói bị drop chia cho tổng số gói truyền là số gói bị drop và số gói sent. Trong khoảng thời gian 30s thì lại reset số gói drop và số sent về 0. Ta cũng sẽ tạo một hàm được lưu trong file awk để trích xuất các sự kiện gói đến và gói bị rớt từ file trace để tính toán ra tỉ lệ mất gói

e. Tính độ trễ truyền thông của các luồng.

Trễ truyền thông của các luồng được tính bằng thời gian từ lúc gửi gói đi từ node nguồn đến khi nhận được gói đó ở node đích. Tương tự như trên, ta cũng sẽ tạo một hàm lưu trong file awk để trích xuất dữ liệu từ file trace, hàm sẽ lấy ra thời điểm phát, thời điểm đến đích hoặc bị hủy của tất cả các gói tin của tất cả các luồng để tính ra độ trễ. Các gói và các luồng sẽ được phân biệt bằng pktid và fid.

III.Lập kế hoạch và phân công công việc

a. Lập kế hoạch

Bài tập lớn được thực hiện trong 8 tuần , từ tuần 7 đến tuần 15 của thực học, bảo vệ bài tập lớn tuần 16.:

- Tuần 1: Cài đặt hệ điều hành Ubuntu và NS2, Các thành viên trong nhóm liên hệ với nhau, tổ chức họp nhóm bầu nhóm trưởng , tìm hiểu yêu cầu đề bài, phân công công việc cho các thành viên trong nhóm
- Tuần 2: Tìm hiểu giao thức truyền thông mạng UDP , phát gói tin theo phân bố Poission, Thực hiện mô phỏng 1 số hệ thống mạng thông tin đơn giản trên NS2.

- Tuần 3: Code mô hình mạng thông tin của bài số 1, chạy thử và kiểm tra,
- Tuần 4: Gắn các nguồn dữ liệu vào các node và truyền như yêu cầu đề bài.
- Tuần 5,6 : Tìm hiểu về trễ đường truyền, trễ tại các bộ định tuyến và tỷ lệ mất gói tại các bộ định tuyến.
- Tuần 7,8: Kiểm tra lại code lần cuối và viết báo cáo.

b. Phân công việc

STT	Họ và Tên	Công việc
1	Đặng Đình Tuấn	<p>Tìm hiểu công cụ xgraph và AWK.</p> <p>Tính toán trễ tại mỗi bộ định tuyến và vẽ đồ thị.</p>
2	Hồ Anh Tuấn Lê Văn Tuấn	<p>Tăng tốc độ phát sinh gói của mỗi luồng đi 100 gói/s sau mỗi khoảng thời gian 30s, thống kê và vẽ đồ thị tỉ lệ mất gói tại mỗi bộ định tuyến trong mỗi khoảng thời gian đó.</p>
3	Hồ Anh Tuấn Lê Văn Tuấn Vũ Việt Vương	<p>Xây dựng tiến trình Poisson</p> <p>Thay đổi tốc độ phát sinh và kích thước gói theo chiều biến thiên ngược nhau. (sau mỗi 30s thay đổi 100 gói/s và 100byte).</p> <p>Tính toán trễ truyền thông các luồng.</p> <p>Nhận xét kết quả thu được</p>

I. Quá trình thực hiện

1. Tìm hiểu và chuẩn bị các kiến thức lý thuyết liên quan

a. Các phương thức định tuyến

NS-2 thực thi ba chính sách định tuyến:

- static routing (định tuyến tĩnh): Là định tuyến được dùng mặc định trong ns2. Định tuyến sử dụng thuật toán tính toán đường đi chính xác một lần tại lúc khởi động. Khi cấu trúc mạng thay đổi sẽ phải cập nhật lại, do đó chỉ phù hợp với hệ thống đơn giản, có kết nối đơn và đường truyền dữ liệu được xác định trước.

- session routing (định tuyến động) : các nút mạng tự động tìm ra đường đi tối ưu đến các nút khác rồi lập ra bảng định tuyến. Bảng định tuyến sẽ được cập nhật thường xuyên khi có thay đổi . Trong bài tập này ta sẽ sử dụng phương pháp định tuyến này
- DV routing: định tuyến véc tơ khoảng cách

Dùng phương thức rproto trong lớp Simulator để xác định chính sách định tuyến được dùng.

\$ns rproto <type>

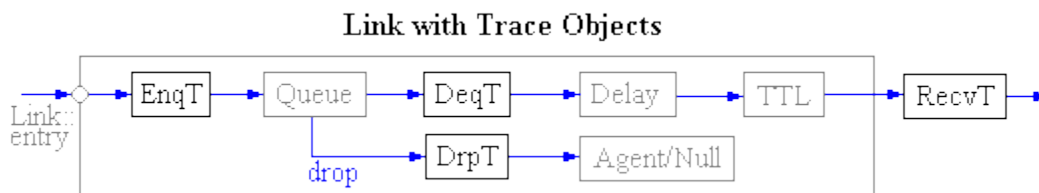
b. Tìm hiểu về các giao thức truyền thông UDP

Giao thức UDP (User Datagram Protocol) hay còn gọi là giao thức gói người dùng là một trong hai giao thức cốt lõi của giao thức TCP/IP, cho phép người dùng có thể gửi những dữ liệu ngắn được gọi là datagram từ máy tính này đến máy tính khác. UDP không đảm bảo sự tin cậy và thứ tự truyền nhận, các gói dữ liệu có thể đến không đúng thứ tự hoặc bị mất mà không có thông báo. Tuy nhiên, UDP nhanh và hiệu quả hơn đối với những mục tiêu như kích thước nhỏ và yêu cầu khẩn trương về thời gian, giao thức này hữu dụng đối với việc trả lời các truy vấn với số lượng lớn người yêu cầu. Những ứng dụng phổ biến sử dụng UDP như DNS (Domain Name System), ứng dụng streaming media, Voice over IP, Trivial File Transfer Protocol (TFTP), và game trực tuyến.

c. Tracing

Trong NS, các hoạt động mạng được trace (lưu dấu) trong các simplex link. Nếu mô phỏng được chỉ trực tiếp đến các hoạt động trace (bằng lệnh *\$ns trace-all file* hay *\$ns namtrace-all file*) thì các link sẽ có các đối tượng trace được chèn thêm vào. User cũng có thể tạo đối tượng trace với loại xác định để lưu vết mạng từ nguồn đến đích bằng lệnh *create-trace {type file src dst}*.

Khi từng đối tượng được chèn thêm đối tượng trace (EnqT, DeqT, DrpT, RecvT) nhận packet, nó sẽ ghi vào file trace xác định mà không tiêu tốn thời gian mô phỏng. Và truyền packet đến đối tượng mạng kế tiếp.



File trace chuẩn trong NS-2 và định dạng của file này như sau:

event	time	from node	to node	pkt type	pkt size	flags	fid	src addr	dst addr	seq num	pkt id
-------	------	-----------	---------	----------	----------	-------	-----	----------	----------	---------	--------

```

r : receive (at to_node)
+ : enqueue (at queue)          src_addr : node.port (3.0)
- : dequeue (at queue)         dst_addr : node.port (0.0)
d : drop      (at queue)
    
```

- event: thao tác được thực hiện trong mô phỏng
- time: thời gian xuất hiện sự kiện
- from node: node 1 là node được trace
- to node: node 2 là node được trace
- pkt type: loại packet
- pkt size: kích cỡ packet
- flags: cờ
- fid: mã luồng
- src addr: địa chỉ node nguồn
- dst addr: địa chỉ node đích
- seq num: số trình tự
- pkt id: mã packet duy nhất

d. Xgraph

Xgrap là một công cụ vẽ đồ thị được cung cấp bởi ns. Xgraph cho phép chúng ta tạo ra các file postscript , ảnh , và 1 số định dạng khác bằng cách ấn vào “Hdcpy”. câu lệnh gọi Xgraph có thể được khai báo trong kịch bản TCL do đó có thể xuất ra ngay đồ thị khi kết thúc mô phỏng .

Đầu vào của Xgraph là 1 hoặc nhiều file chứa mỗi cặp giá trị x-y trên 1 dòng (mỗi dòng sẽ chứa tọa độ của 1 điểm trên đồ thị) . Ví dụ lệnh : Xgraph f1 f2 sẽ vẽ ra trên cùng 1 đồ thị của file f1,f2.

Một số lựa chọn khi sử dụng Xgraph :

- Title : -t “ten_do_thi”
- Kích thước c : -geometry xsize x ysize
- Tiêu đề cho các trục : -x “xtitle” -y “ytittle”
- Màu của chữ và lưới i – v
- Màu nền – bg

2. Thực hiện chương trình

a) Tạo phân bố Poisson

Chúng em tạo phân phối Poisson theo phương pháp đã trình bày trong phần phân tích yêu cầu. Ví dụ

```
# Tao phan phoi Possion cho udp1
```

Tạo một đối tượng Exponential

```
set Poisson1 [new Application/Traffic/Exponential]
$Poisson1 set packetSize_ 1000
```

Cho tham số burst_time_ về 0 và idle_time bằng $\frac{1}{\lambda}$ của tiến trình Poisson. Theo bài ra

có lamda cho luồng UDP1 là 1000 => idle_time_ = 0.001

```
$Poisson1 set burst_time_ 0
$Poisson1 set idle_time_ 0.001
$Poisson1 set rate_ 5Mb
```

```
$Poisson1 attach-agent $udp1
```

b) Tính thời gian trễ tại mỗi bộ định tuyến

Ta tạo file awk với nội dung hàm như sau.:

```
# Node 1
# TimePktIn: thời điểm gọi tin bắt đầu đi vào hàng đợi
# TimePktOut: thời điểm gọi tin đi ra
# Delay: độ trễ truyền thông của gọi tin
BEGIN {TimePktIn[50000];TimePktOut[50000];Delay[50000];}
{
    if (($1 == "r") && ($4 == "2")) TimePktIn[$12] = $2;

    if (($1 == "-") && ($3 == "2"))
    {
        TimePktOut[$12] = $2;
        Delay[$12] = TimePktOut[$12]-TimePktIn[$12];
        print $12" "Delay[$12];
    }
}
END {}
```

Hàm trên sẽ dùng 2 lệnh if để kiểm tra các sự kiện thêm và xuất một gói ở node 1. Đánh dấu các mốc thời gian xảy ra các sự kiện đó rồi trừ cho nhau để tính ra trễ

c) Tăng gói tin

Thủ tục tăng gói tin được khai báo như sau:

```
proc PackageIncrease {} {
    global ns Poisson1 Poisson2 Poisson3
    set now [$ns now]
    set time 30
    set idle1 [$Poisson1 set idle_time_]
    set idle2 [$Poisson2 set idle_time_]
    set idle3 [$Poisson3 set idle_time_]
    set rate1 [$Poisson1 set rate_]
    set rate2 [$Poisson2 set rate_]
}
```



```

set rate3 [$Poisson3 set rate_]
$Poisson1 set idle_time_ [expr 1.0/[expr 1.0/[$idle1 + 100]]]
$Poisson2 set idle_time_ [expr 1.0/[expr 1.0/[$idle2 +100]]]
$Poisson3 set idle_time_ [expr 1.0/[expr 1.0/[$idle3 +100]]]
$Poisson1 set rate_ [expr $rate1 + 130000]
$Poisson2 set rate_ [expr $rate2 + 100000]
$Poisson3 set rate_ [expr $rate3 + 100000]
$ns at [expr $now + $time] "PackageIncrease"
}

```

Hàm này sẽ lấy đầu vào là các `idle_time_` và `rate_` của các phân phối poisson (thực chất là các đối tượng ExponentialOn/Off) để tính toán lại giá trị mới sau khi tốc độ phát sinh tăng lên và gán trở lại cho các đầu vào đó

d) Tính tỉ lệ mất gói tại mỗi bộ định tuyến

Tạo file awk chứa hàm sau:

```

#Tỉ lệ mất gói tại node1
BEGIN {packetDrop = 0;packetReceive = 0; ThoiGianXet =0;}
{
    if ($2 >= ThoiGianXet)
    {
        if (packetDrop + packetReceive) print ThoiGianXet"
"100*packetDrop/(packetReceive + packetDrop);
        else print ThoiGianXet" "0;
        packetDrop =0;
        packetReceive =0;
        ThoiGianXet=ThoiGianXet +30;
    }
    if (($3=="2") && ($1=="+"))
        packetReceive++;

    if (($3=="2") && ($1=="d"))
        packetDrop++;
}
END {}

```

Hàm trên dùng hai biến `packetDrop` và `packetReceive` để lưu số gói mất và số gói được lưu vào trong hàng đợi. Biến `ThoiGianXet` dùng để chỉ các khoảng thời gian bội số của 30s. Nghĩa là cứ sau 30s ta lại tính độ trễ một lần trong cả khoảng thời gian 30s đó, sau đó lại cho `packetDrop` và `packetReceive` về 0 để tính độ trễ cho 30s tiếp theo. Dữ liệu đầu vào cũng được trích xuất từ file trace thông qua các lệnh if

e) Tính độ trễ truyền thông của các luồng.

```
BEGIN {TimeReceive[50000];TimeSend[50000];Delay[50000]}
{
action = $1;
time = $2;
from = $3;
to = $4;
type = $5;
pktsize = $6;
flow_id = $8;
src = $9;
dst = $10;
seq_no = $11;
packet_id = $12;

if ((flow_id=="1"))
{
    if ((action == "+")&&(from=="0"))
    {
        TimeSend[packet_id] = time;
    }
    if ((action == "r") && (to == "8"))
    {
        TimeReceive[packet_id] = time;
        Delay[packet_id] = TimeReceive[packet_id]-
TimeSend[packet_id];
        print time " "Delay[packet_id] >> "DelayUDP1.tr";
    }
    if (action == "d")
    {
        Delay[packet_id] = 0;
        print time " "Delay[packet_id] >> "DelayUDP1.tr";
    }
}

if ((flow_id=="2"))
{
    if ((action == "+") && (from=="1"))
    {
        TimeSend[packet_id] = time;
    }
    if ((action == "r") && (to == "9"))
    {
        TimeReceive[packet_id] = time;
        Delay[packet_id] = TimeReceive[packet_id]-
TimeSend[packet_id];
    }
}
```

```
        print time " "Delay[packet_id]>> "DelayUDP2.tr";
    }
    if (action == "d")
    {
        Delay[packet_id] = 0;
        print time " "Delay[packet_id]>> "DelayUDP2.tr";
    }
}
if ((flow_id=="3"))
{
    if ((action == "+")&&(from=="1"))
    {
        TimeSend[packet_id] = time;
    }
    if ((action == "r") && (to == "8"))
    {
        TimeReceive[packet_id] = time;
        Delay[packet_id] = TimeReceive[packet_id]-
TimeSend[packet_id];
        print time " "Delay[packet_id]>> "DelayUDP3.tr";
    }
    if (action == "d")
    {
        Delay[packet_id] = 0;
        print time " "Delay[packet_id]>> "DelayUDP3.tr";
    }
}

}

END {}
```

Trong hàm trên, lệnh if đầu tiên sẽ kiểm tra mã luồng của sự kiện trong file trace đang xét tới để tìm ra các packet nào của luồng nào. Các lệnh if bên trong sẽ kiểm tra sự kiện gửi packet ở nút nguồn để tìm ra thời điểm gửi, kiểm tra sự kiện nhận packet ở nút đích để kiểm tra thời điểm nhận. Tính hiệu thời gian của 2 thời điểm trên ta sẽ có được trễ truyền thông của gói đang xét.

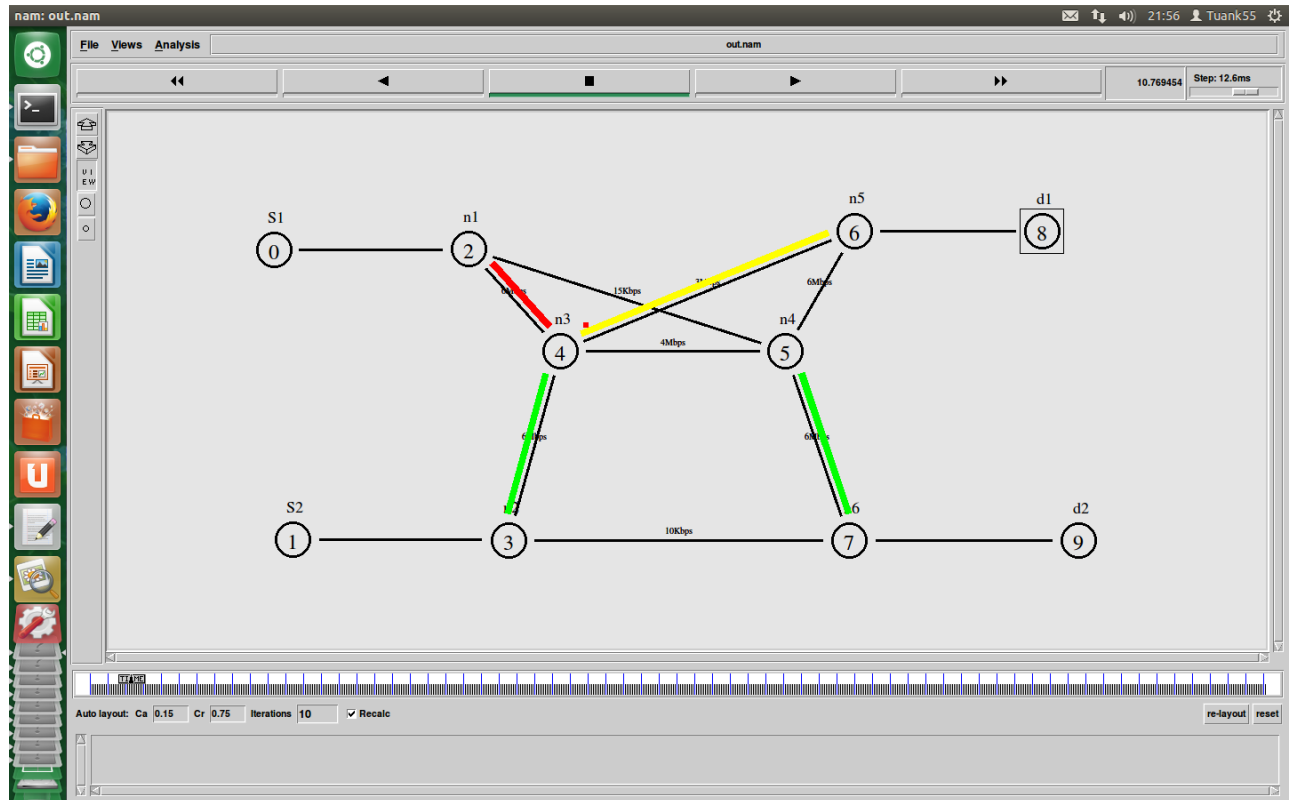
II. Thực hiện mô phỏng

1. Kịch bản mô phỏng:

Thời điểm(s)	Mục đích
1.0	Luồng 1,2,3 truyền dữ liệu
2.0	Luồng 2 truyền dữ liệu
3.0	Luồng 3 truyền dữ liệu
30.0	Bắt đầu tăng tốc độ gói 100 gói/s sau chu kỳ 30s (Với yêu cầu 4 thì ko cho hàm này thực thi)
300.0	Luồng 1 kết thúc truyền dữ liệu
300.0	Luồng 2 kết thúc truyền dữ liệu
300.0	Luồng 3 kết thúc truyền dữ liệu
305.0	Đóng các file trace và data

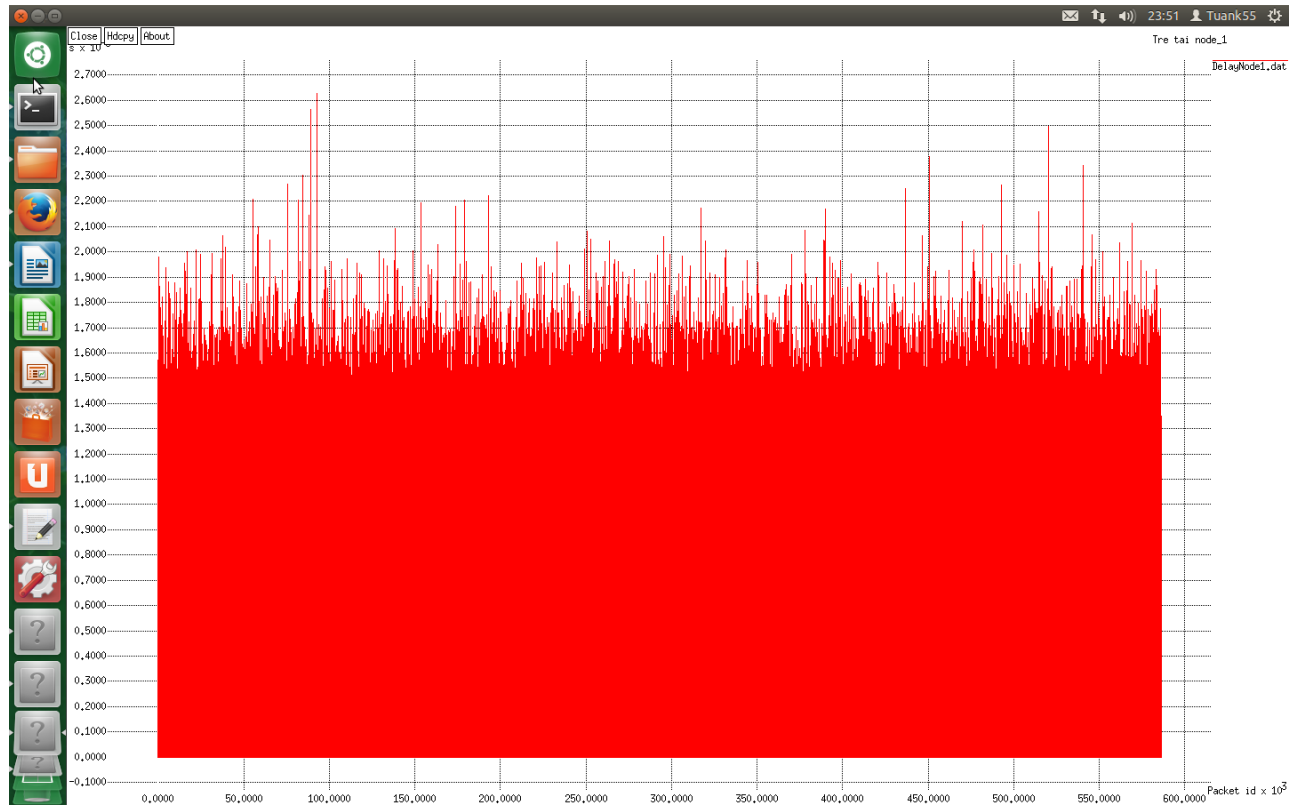
2 Kết quả mô phỏng

a. Xây dựng mô hình mạng

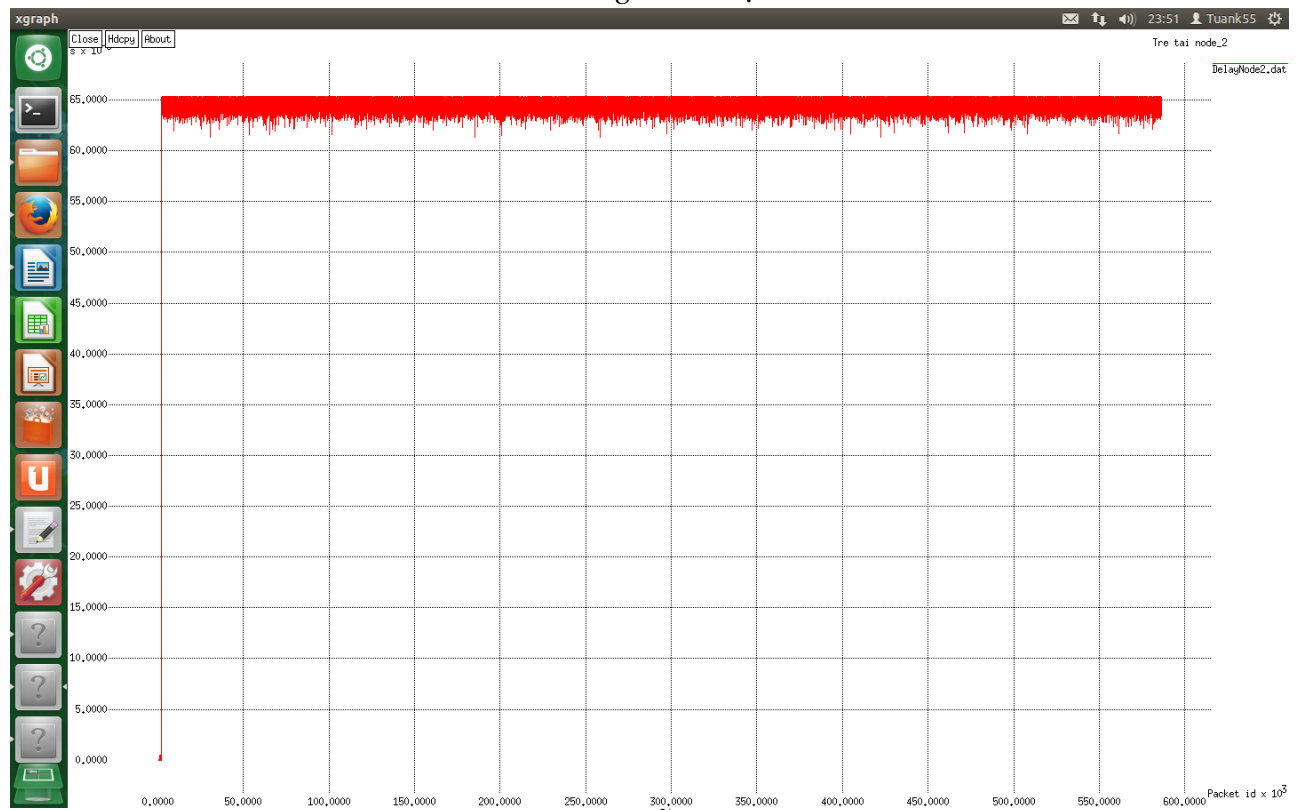


Trong file mã nguồn chúng em đã sử dụng định tuyến động, gán giá cho các liên kết tỉ lệ nghịch với băng thông để các gói tin đi theo những liên kết có băng thông rộng nhất, tránh mất gói. Kết quả cho thấy khi mô phỏng, các luồng đã truyền tin theo đúng như tính toán.
 Luồng UDP 1 đi từ s1 qua n1 n3 n5 về d1
 Luồng UDP 2 đi từ s2 qua n2 n3 n4 và n6 về d2
 Luồng UDP 3 đi từ s2 qua n2 n3 n5 về d1

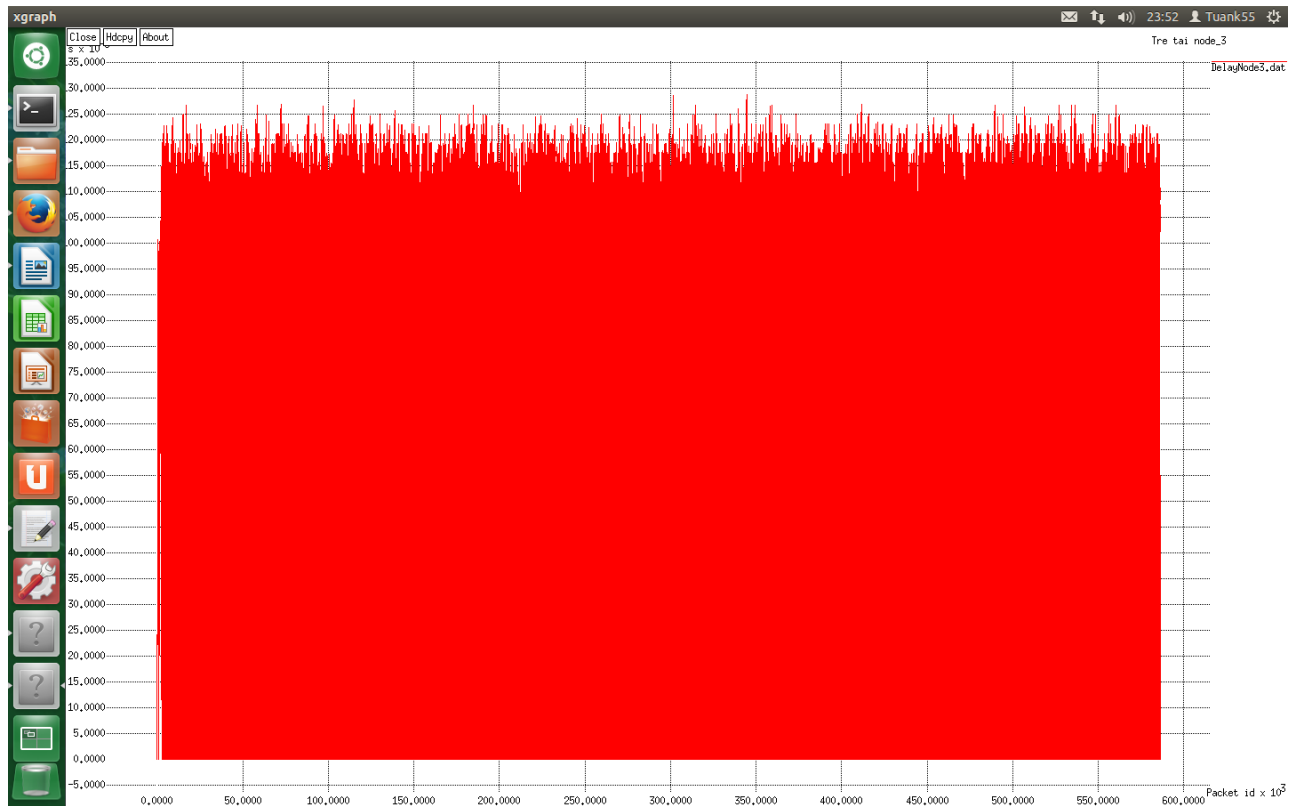
b. Tính trễ tại các bộ định tuyến



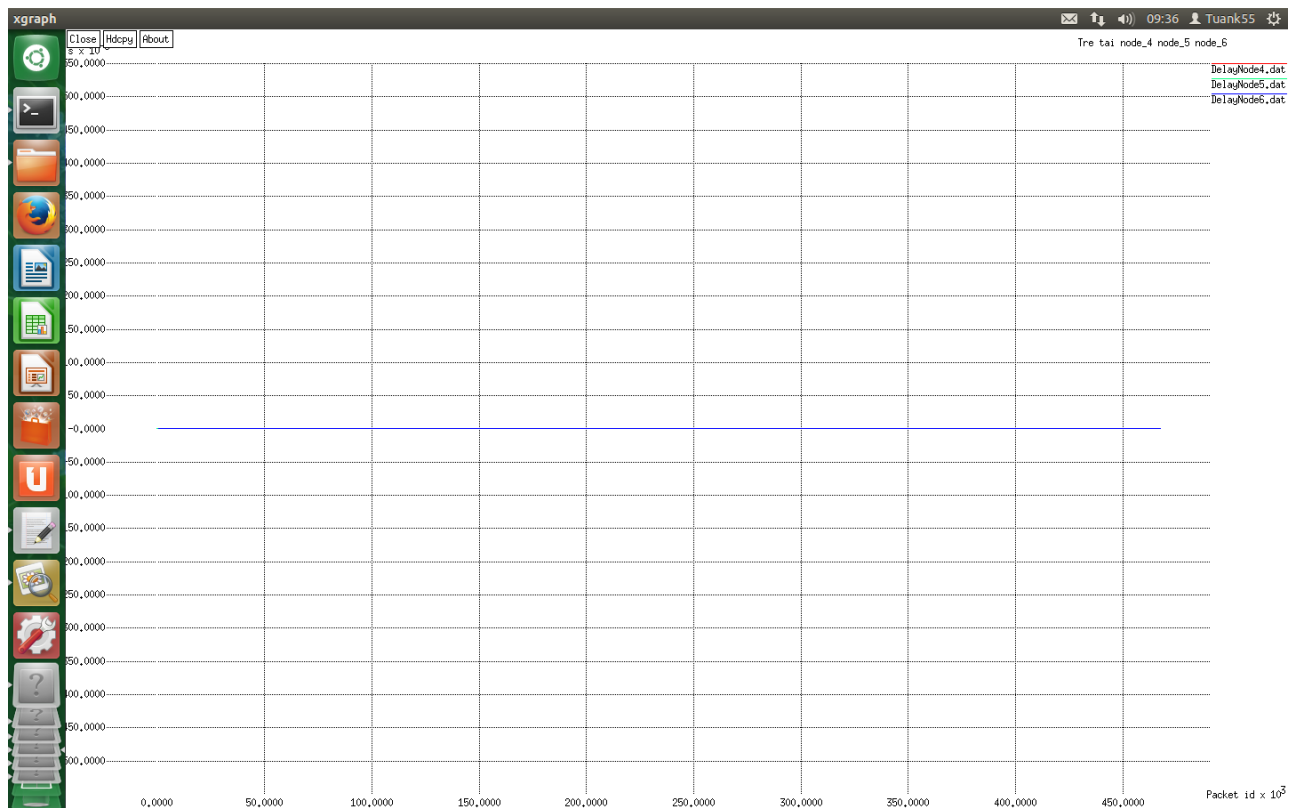
Thời gian trễ tại node 1



Thời gian trễ tại node 2

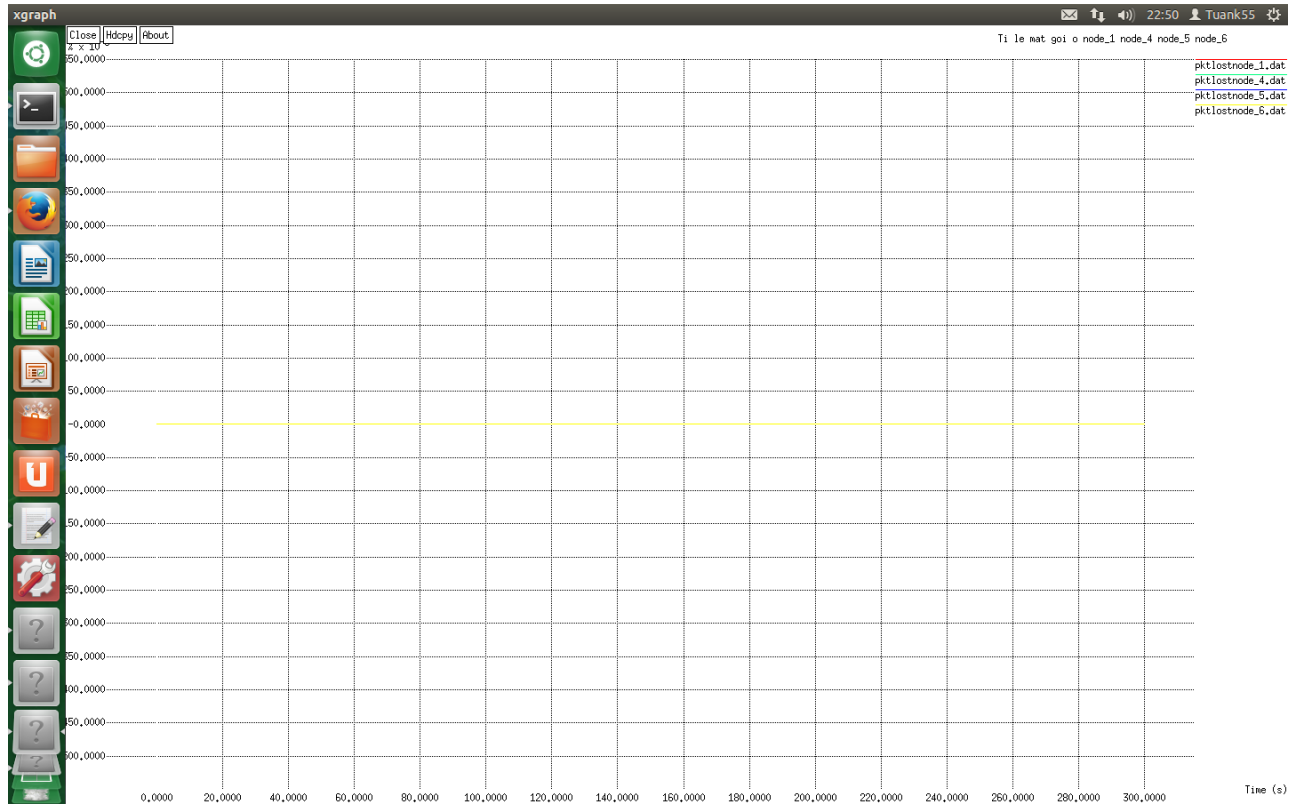


Thời gian trễ tại node 3

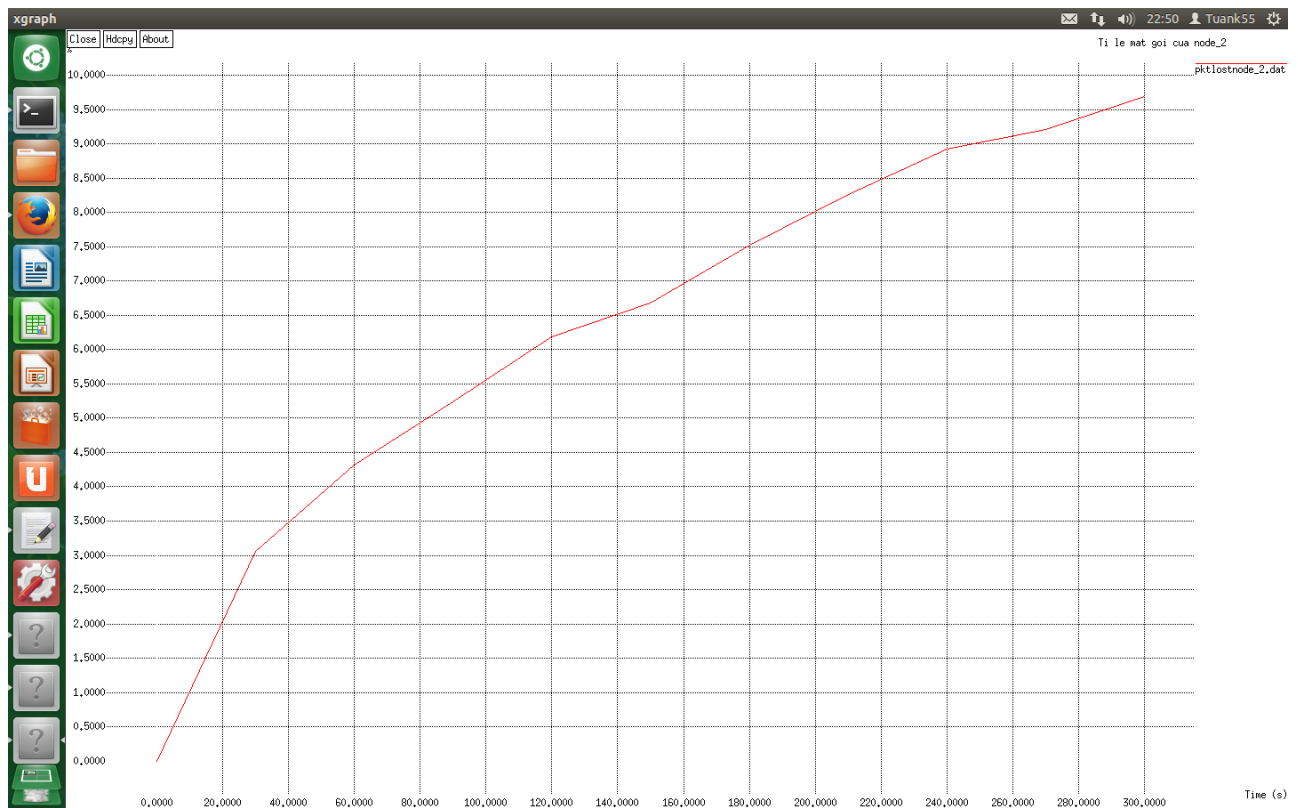


Thời gian trễ tại node 4 5 6

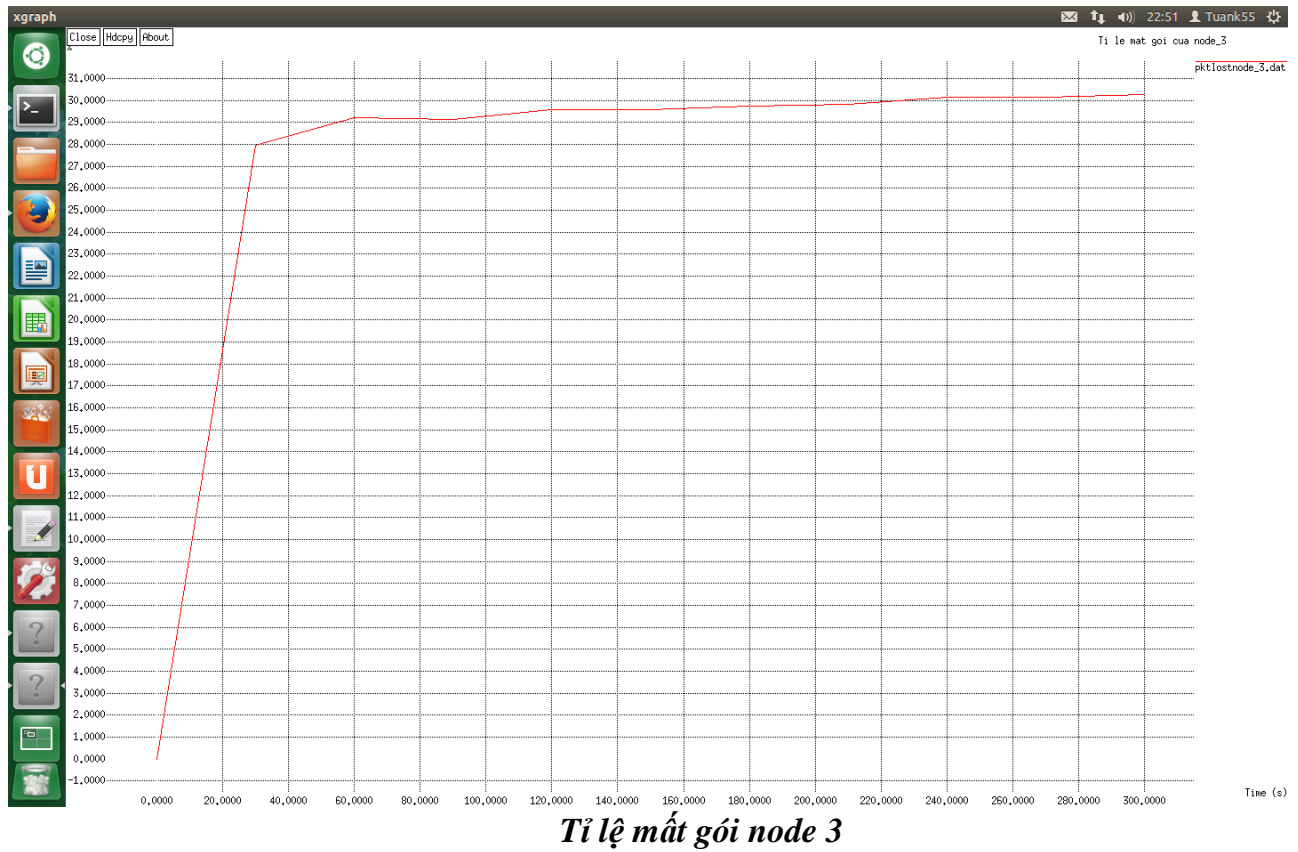
a. Tính tỉ lệ mất gói ở các node khi tăng tốc độ gửi



Tỉ lệ mất gói ở các node 1 456

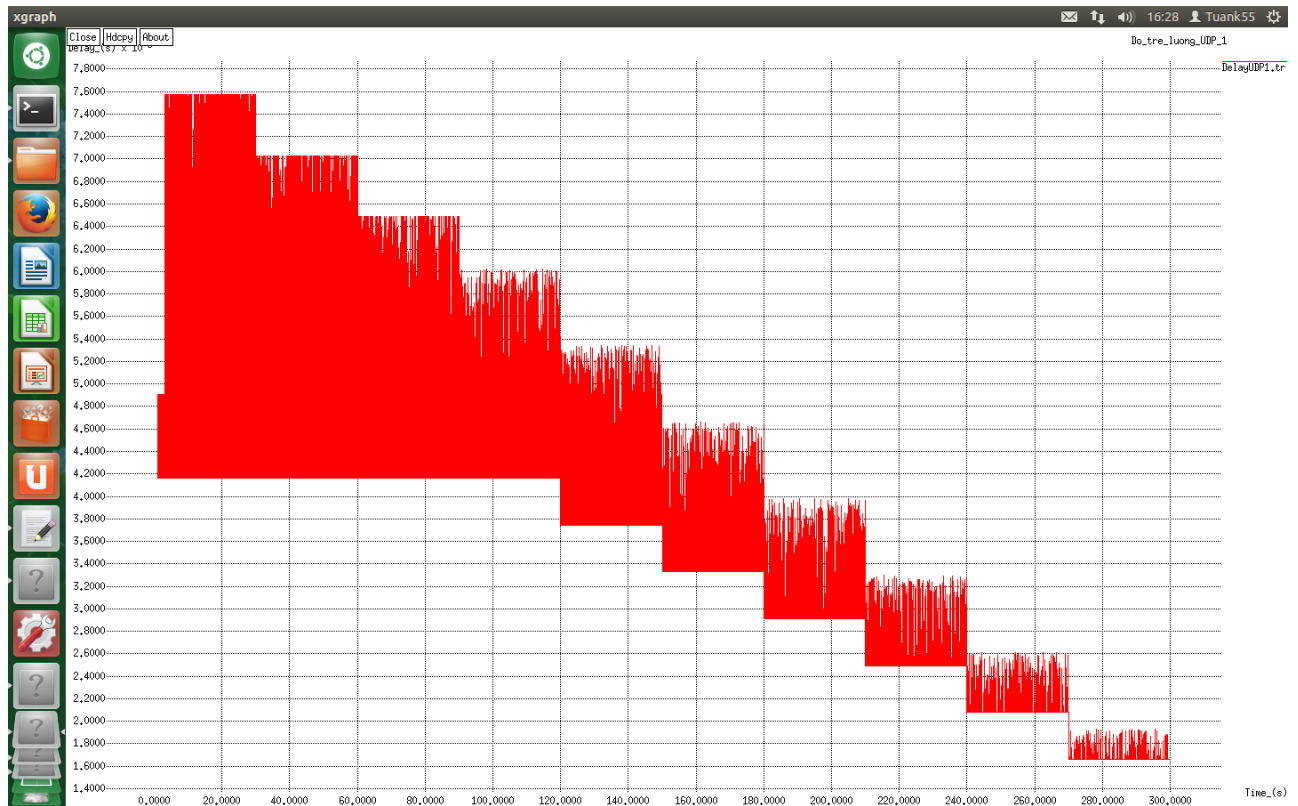


Tỉ lệ mất gói node 2

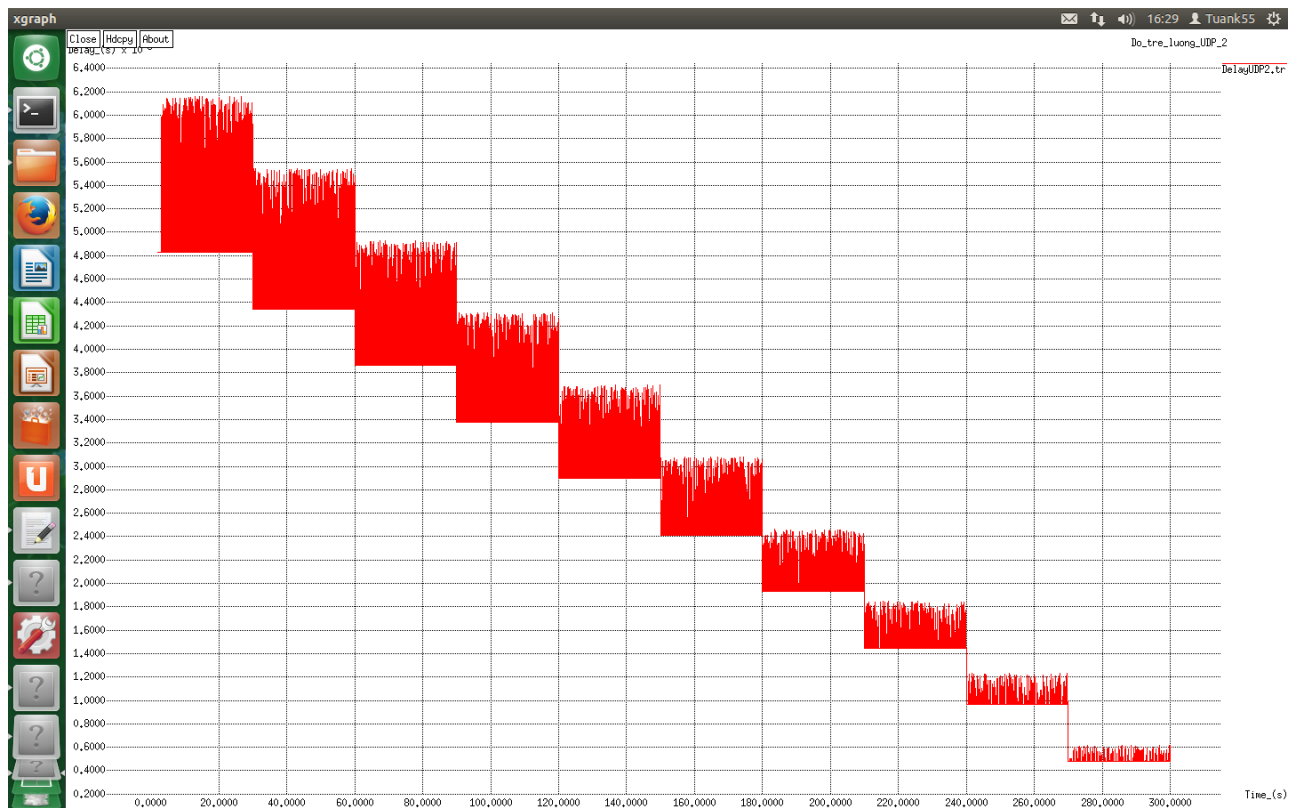


b. Tính trở truyền thông các luồng

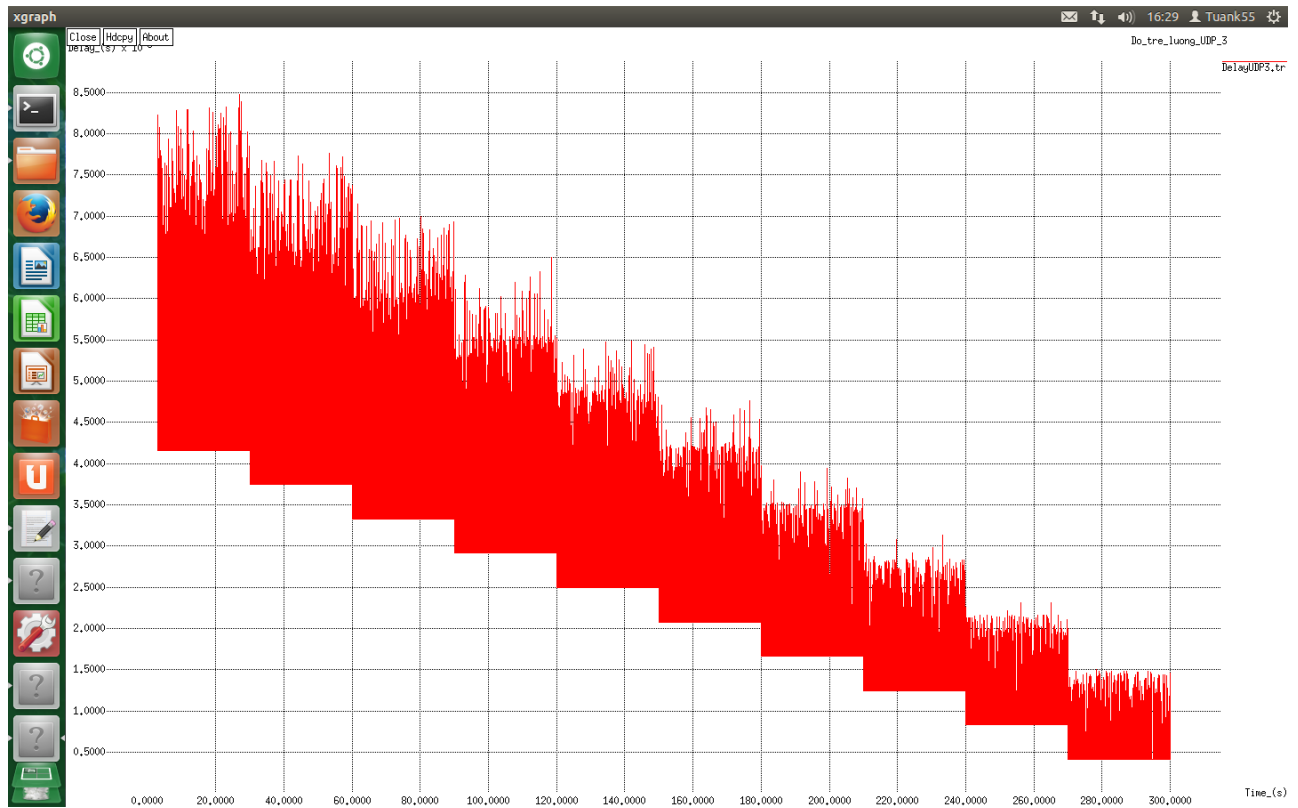
Trường hợp 1: Tăng tốc độ phát sinh thêm 100 gói/s và giảm kích thước gói 100byte sau mỗi khoảng 30s



Trễ truyền thông luồng 1

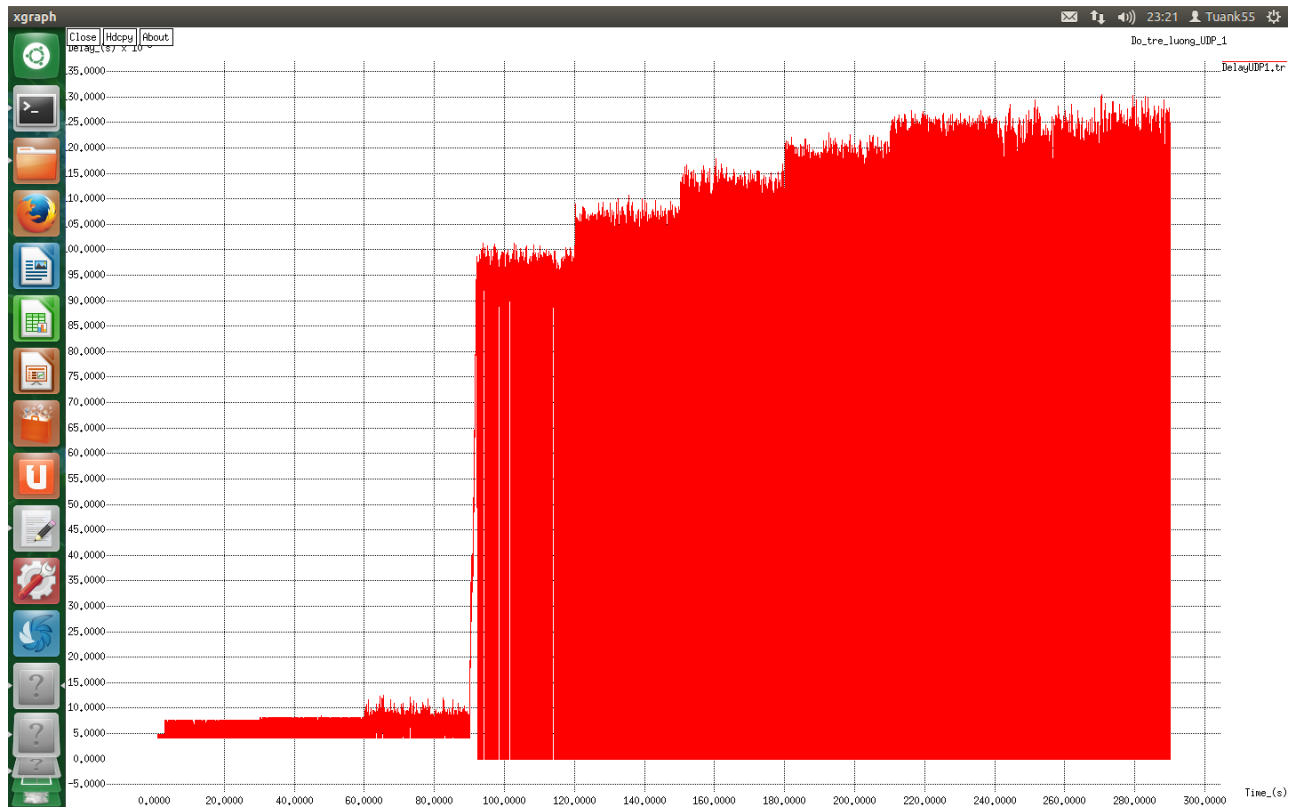


Trễ truyền thông luồng 2

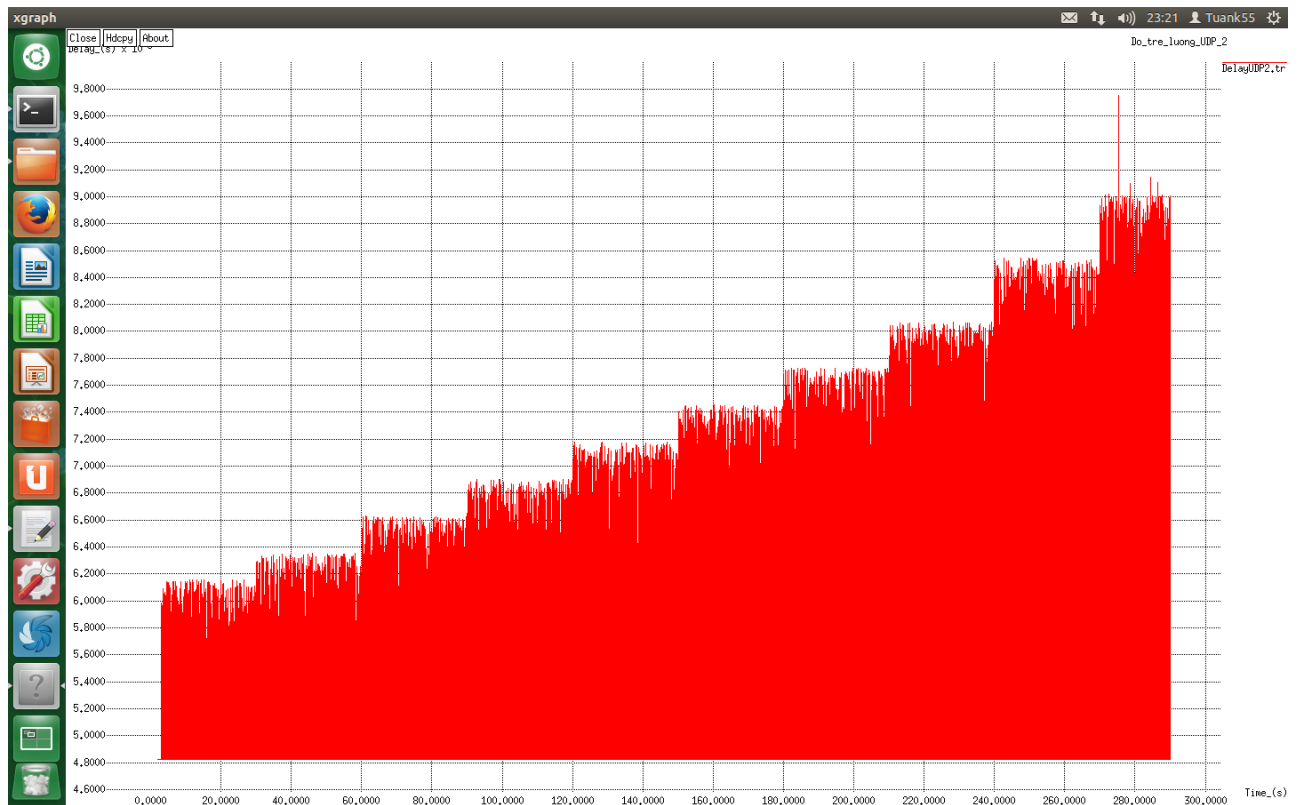


Trễ truyền thông luồng 3

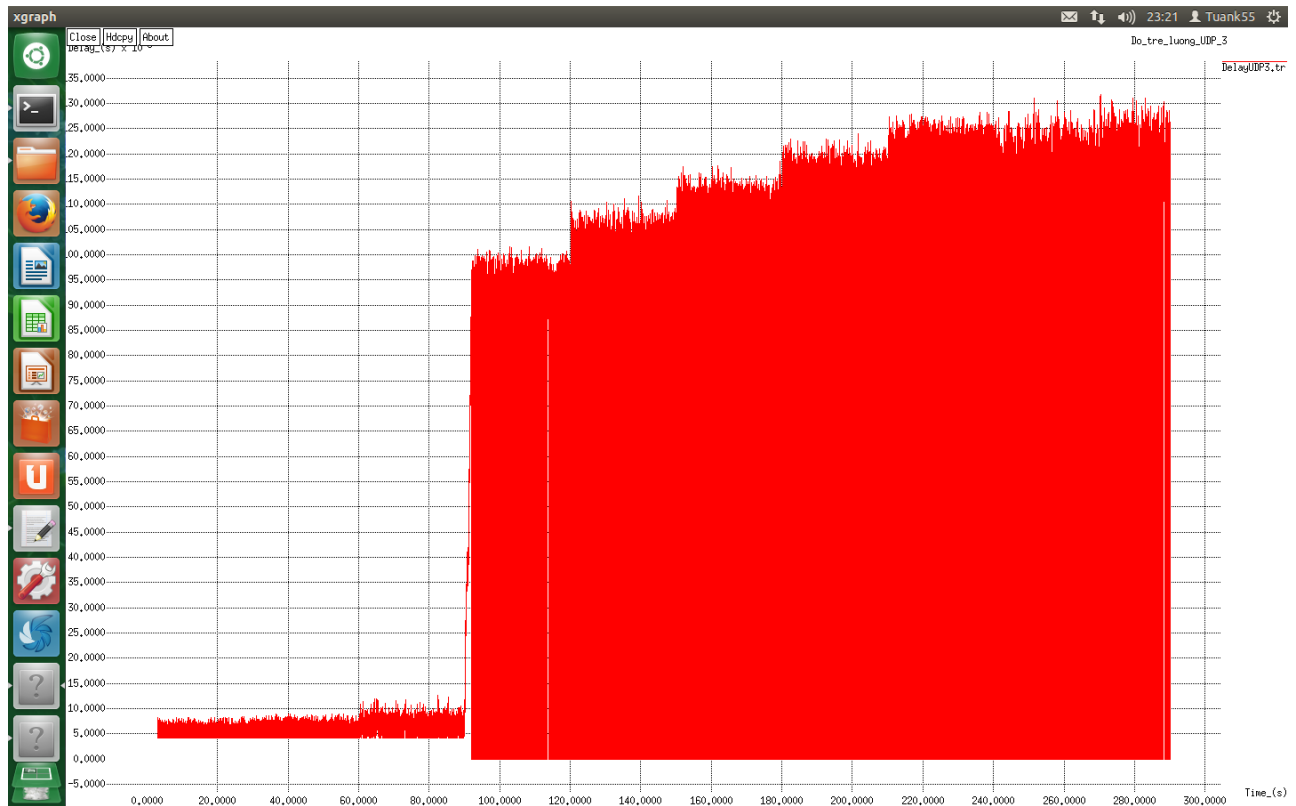
Trường hợp 2: Giảm tốc độ phát sinh thêm 100 gói/s và tăng kích thước gói 100byte sau mỗi khoảng 30s



Trễ luồng UDP 1



Trễ luồng UDP 2



Trễ luồng UDP 3

Nhận xét: Khi ta tăng tốc độ phát sinh và giảm kích thước gói sau mỗi 30s thì trễ truyền thông của cả 3 luồng UDP đều giảm xuống rõ rệt. Còn khi ta giảm tốc độ phát sinh và tăng kích thước gói sau mỗi 30s thì trễ truyền thông của cả 3 luồng UDP đều tăng lên. Như vậy, kích thước gói tin có ảnh hưởng mạnh hơn tới trễ truyền thông trong giao thức UDP.

KẾT LUẬN

Trong quá trình học tập từ trước tới giờ, chúng em đã được làm quen và sử dụng một số công cụ mô phỏng, cũng hiểu được vai trò và ý nghĩa của việc mô phỏng đối với thiết kế trong thực tế. Đây là lần đầu tiên chúng em được tiếp xúc và làm việc với công cụ Ns2, thực sự thì vẫn chỉ ở mức biết sử dụng chứ chưa thể nói là sử dụng thành thạo được. Tuy nhiên, thông qua việc thực hiện bài tập lớn lần này trên công cụ ns2, chúng em đã hiểu thêm và hình dung được nhiều vấn đề liên quan đến mạng, định tuyến, truyền thông trong mạng... Từ đó sẽ hiểu hơn, thật chặt chẽ cố hơn về những kiến thức lý thuyết ở trên lớp mà thầy đã dạy. Mặc dù đã cố gắng hết sức nhưng chắc chắn bài làm của chúng em còn nhiều thiếu sót, mong Thầy góp ý cho chúng em những nhận xét quý báu để bài tập lớn này được hoàn thiện hơn.

Lời cuối, nhóm chúng em xin kính chúc Thầy sức khỏe, công tác tốt, và gặt hái được nhiều thành công hơn nữa trong đào tạo cũng như nghiên cứu!

Nhóm chúng em xin chân thành cảm ơn!

Tài liệu tham khảo

1. Slide bài giảng Cơ sở mạng thông tin của Thầy Phạm Văn Tiến
2. Tài liệu tham khảo NS2 (Ns2_manual, Introduction to Network Simulator NS2 – Teerawat Issariyakul)
3. Tài liệu hướng dẫn NS2 <http://www.isi.edu/nsnam/ns/tutorial/>

