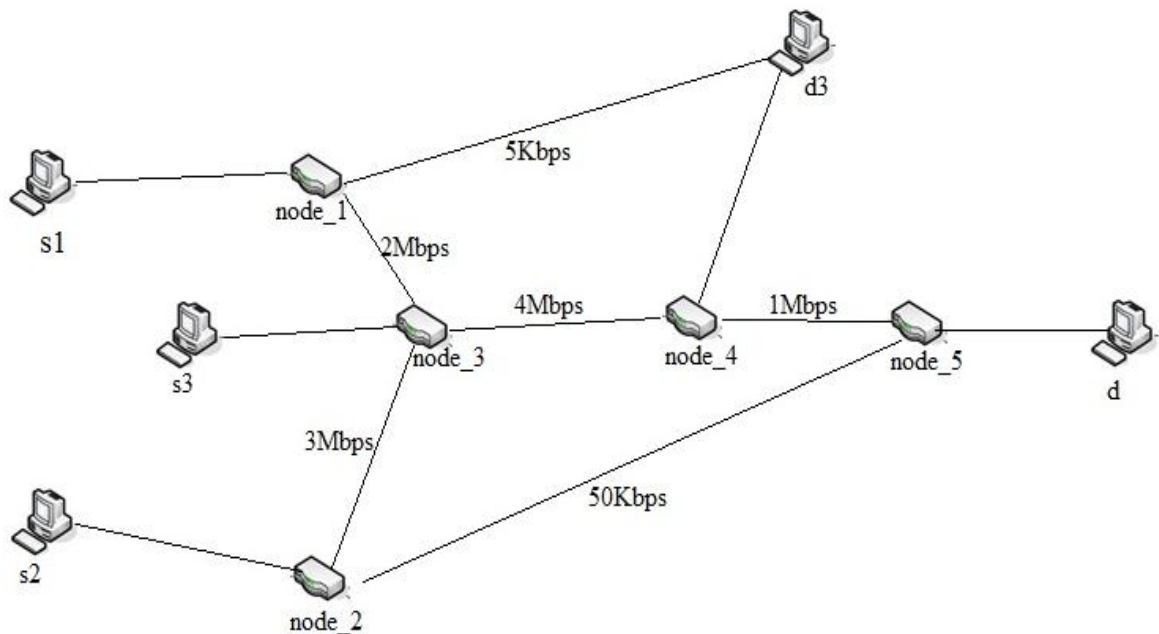


## I. Đề bài và yêu cầu

Cho mạng thông tin có cấu hình như hình vẽ. Nút s1 và s2 tương ứng tạo ra các nguồn lưu lượng tcp1 và tcp2 gửi gói tin đến nút d dựa trên giao thức truyền TCP. Các gói được phát sinh gói tại các nguồn này một cách đều đặn với tốc độ tương ứng là 500 gói/s và 600 gói/s, chiều dài gói là 1300 byte. Nút s3 có nguồn lưu lượng UDP phát ra các gói gửi tới nút d3 với chiều dài là 1024 byte, số lượng gói tuân theo phân bố Poisson, tốc độ 400 gói/s. Các hàng đợi gắn với các liên kết sử dụng cơ chế SFQ.



1. Viết kịch bản và chạy mô phỏng hệ thống thông tin nói trên trong khoảng thời gian không ngắn hơn 4 phút.
2. Thu thập và vẽ đồ thị các số liệu khả thông và tỉ lệ mất gói của từng luồng tại từng nút, kích thước hàng đợi trên các liên kết node\_2 – node\_5, node\_4 – node\_5 và node\_3 – node\_4.
3. Thu thập số liệu và vẽ đồ thị biểu diễn số lượng gói đến d3 bị trễ nhiều hơn 1s.
4. Thay đổi kích thước gói, tốc độ phát sinh gói của luồng UDP rồi làm lại các thao tác trong mục 2 và mục 3. 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 vào 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, s2, s3) nối với mạng bằng liên kết Ethernet. Trễ truyền lan trên tất cả các liên kết là 10ms. Chiều dài mặc định của tất cả các hàng đợi là 50.

Bài toán yêu cầu xây dựng mạng thông tin với 2 luồng dữ liệu TCP từ 2 node nguồn đến 1 đích thông qua 5 node mạng (router) và 1 luồng dữ liệu UDP từ 1 node nguồn đến 1 node đích thông qua 2 node mạng (router). Yêu cầu cụ thể phải thực hiện trên phần mềm NS2:

- Dựng được kịch bản mô phỏng hệ thống trong khoảng thời gian không dưới 4 phút.
- Thu thập và vẽ đồ thị và các số liệu khả thông và tỉ lệ mất gói của từng luồng tại từng nút, kích thước hàng đợi trên các liên kết.
- Thu thập số liệu và vẽ đồ thị biểu diễn số lượng gói đến d3 bị trễ nhiều hơn 1s.
- Thay đổi kích thước gói, tốc độ phát sinh gói của luồng UDP. Nhận xét kết quả thu được.

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 (hoặc các phần mềm giả lập), bộ công cụ mô phỏng NS2.
- Tìm hiểu về khả thông, tỉ lệ mất gói, hàng đợi.
- 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.
- Tìm hiểu các giao thức truyền thông TCP.

### 2. Phương hướng thực hiện yêu cầu trên phần mềm mô phỏng 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:

*. packetize\_ 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 cho 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 30s 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

$$1300(\text{byte/gói}) \times 500(\text{gói/s}) = 0.65 \text{ Mbps}$$

- Nguồn s2 phát ra 1 luồng với gói tin tốc độ là

$$1300(\text{byte/gói}) \times 600(\text{gói/s}) = 0.78 \text{ Mbps}$$

- Nguồn s3 phát ra 1 luồng với gói tin tốc độ là

$$1024(\text{byte/gói}) \times 400(\text{gói/s}) = 0.41 \text{ Mbps}$$

Khi thay đổi số gói ( $\lambda$ ) 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:

$$S1: \text{rate}_- = \text{rate}_- + 130000 \text{ (bytes)}$$

$$S2: \text{rate}_- = \text{rate}_- + 100000 \text{ (bytes)}$$

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

Tính tỉ lệ mất gói tại một bộ tính 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

#### 1. Lập kế hoạch

Bài tập lớn được thực hiện trong 8 tuần, từ tuần 3 đến tuần 11 của thực học, bảo vệ bài tập lớn tuần 12:

- Tuần 1: Cài đặt hệ điều hành Ubuntu (hoặc các phần mềm giả lập) 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 15, 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 số liệu khả thông, tỉ lệ mất gói, kích thước hàng đợi, trễ đường truyền.
- Tuần 7,8: Kiểm tra lại code lần cuối và viết báo cáo.

#### 2. Phân công công việc

Họ và tên	Công việc
1. Trần Đình Đạt	Tìm hiểu công cụ xgraph và AWK. Tính toán trễ đường truyền UDP và vẽ đồ thị.

<p>2. Phạm Trọng Hội</p> <p>3. Lê Thành Nam</p>	<p>Thay đổi kích thước gói, tốc độ phát sinh gói của luồng UDP rồi thu thập và vẽ đồ thị.</p> <p>Nhận xét kết quả.</p>
<p>4. Nguyễn Văn Tuyền</p> <p>5. Huỳnh Tấn Vương</p>	<p>Thu thập và vẽ đồ thị các số liệu khả thông và tỉ lệ mất gói của từng luồng tại từng nút, kích thước hàng đợi trên các liên kết.</p>

## IV. 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 rtpproto trong lớp Simulator để xác định chính sách định tuyến được dùng.

\$ns rtpproto <type>

#### b. Tìm hiểu về các giao thức truyền thông TCP

TCP (Transport Control Protocol) là một giao thức truyền tải chịu trách nhiệm truyền khoảng 90% lưu lượng dữ liệu trên Internet. Chính vì vậy, nắm vững và hiểu rõ TCP đóng vai trò vô cùng quan trọng để định cỡ lưu lượng trong Internet. Mặc dù TCP đã được triển khai rộng khắp, nó vẫn phát triển không ngừng. IETF (Internet Engineering Task Force) là tổ chức tiêu chuẩn hóa chính liên quan tới giao thức TCP. Không giống như các tổ chức chuẩn hóa khác (ITU hoặc diễn đàn ATM) tất cả các tiêu chuẩn của IETF miễn phí và có sẵn trên mạng. Một số đoạn mã NS mô tả phân tích TCP thông qua mô phỏng.

#### c. Tìm hiểu về các giao thức truyền thông UDP

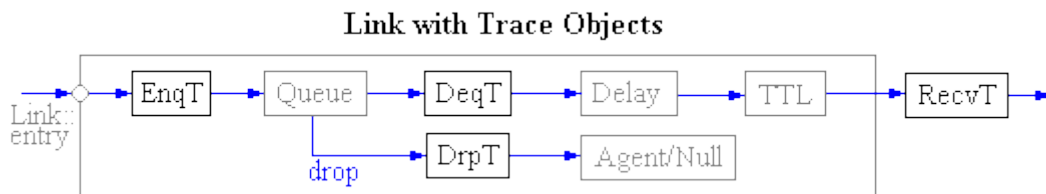
Giao thức UDP (User Datagram Protocol) hay còn gọi là giao thức gói người dùng là một trong hai giao thức cốt lõi của giao thức TCP/IP, cho phép người dùng có thể gửi những dữ liệu ngắn được gọi là datagram từ máy tính này đến máy tính khác. UDP không đảm bảo sự tin cậy và thứ tự truyền nhận, các gói 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 khắt khe 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.

#### d. 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 NS2 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

### e. 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 “Hdcp”. 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 : -geometry xsize x ysize
- Tiêu đề cho các trục : -x “xtitle” –y “ytitle”
- Màu của chữ và lưới – v
- Màu nền – bg

## 1. Thực hiện chương trình

Chương trình được thực hiện qua 39 nội dung như sau:

### 1. Tạo đối tượng mô phỏng

```
set ns [new Simulator]
```

### 2. Phương thức định tuyến

```
$ns rtproto Session
```

### 3. Gắn màu cho 3 luồng dữ liệu

```
$ns color 1 Blue
```

```
$ns color 2 Red
```

```
$ns color 3 Green
```

### 4. Set tốc độ

```
set lambda1 400.0
```

### 5. Set kích thước gói

```
set pksize 1024.0
```



**6. Thời gian gửi gói**

set ArrivalTime1 [new RandomVariable/Exponential]

\$ArrivalTime1 set avg\_ [expr 1/\$lambda1]

**7. Tạo file Trace lưu tất cả thông tin về mạng đang xét**

set tracefile [open TraceAll.tr w]

\$ns trace-all \$tracefile

**8. Tạo file mô phỏng**

set nf [open out.nam w]

\$ns namtrace-all \$nf

**9. Tạo ra 3 file chứa thông tin bảng thông**

set f1 [open khathong1.tr w]

set f2 [open khathong2.tr w]

set f3 [open khathong3.tr w]

**10. Nguồn s1**

set s1 [\$ns node]

\$s1 color Red

\$s1 shape box

**11. Nguồn s2**

set s2 [\$ns node]

\$s2 color Red

\$s2 shape box

**12. Nguồn s3**

set s3 [\$ns node]

\$s3 color Red

\$s3 shape box

**13. Đích d**

set d [\$ns node]

\$d color Black

\$d shape hexagon

#### 14. Định d3

set d3 [\$ns node]

\$d3 color Black

\$d3 shape hexagon

#### 15. Tạo các Router mạng Ri

set n1 [\$ns node]

\$n1 color blue

set n2 [\$ns node]

\$n2 color blue

set n3 [\$ns node]

\$n3 color blue

set n4 [\$ns node]

\$n4 color blue

set n5 [\$ns node]

\$n5 color blue

#### 16. Thiết lập liên kết giữa các node mạng

\$ns duplex-link \$s1 \$n1 500Mb 10ms SFQ

\$ns duplex-link \$s2 \$n2 600Mb 10ms SFQ

\$ns duplex-link \$s3 \$n3 400Mb 10ms SFQ

\$ns duplex-link \$n1 \$n3 2Mb 10ms SFQ

\$ns duplex-link \$n2 \$n3 3Mb 10ms SFQ

\$ns duplex-link \$n1 \$d3 5kb 10ms SFQ

\$ns duplex-link \$n3 \$n4 4Mb 10ms SFQ

\$ns duplex-link \$n2 \$n5 5kb 10ms SFQ

\$ns duplex-link \$n4 \$n5 1Mb 10ms SFQ

\$ns duplex-link \$n4 \$d3 50Mb 10ms SFQ

\$ns duplex-link \$n5 \$d 1Mb 10ms SFQ

#### 17. Giới hạn kích thước hàng đợi giữa các liên kết

\$ns queue-limit \$s1 \$n1 50

\$ns queue-limit \$s2 \$n2 50

\$ns queue-limit \$s3 \$n3 50

\$ns queue-limit \$n1 \$n3 50

\$ns queue-limit \$n2 \$n3 50

\$ns queue-limit \$n1 \$n3 50

\$ns queue-limit \$n3 \$n4 50

\$ns queue-limit \$n2 \$n5 50

\$ns queue-limit \$n4 \$n5 50

\$ns queue-limit \$n4 \$d3 50

\$ns queue-limit \$n5 \$d 50

### 18. Sắp xếp topo mạng (dùng cho hiển thị NAM)

\$ns duplex-link-op \$s3 \$n3 orient right

\$ns duplex-link-op \$n3 \$n4 orient right

\$ns duplex-link-op \$n4 \$n5 orient right

\$ns duplex-link-op \$n5 \$d orient right

\$ns duplex-link-op \$n3 \$n1 orient up

\$ns duplex-link-op \$n1 \$s1 orient left

\$ns duplex-link-op \$n4 \$d3 orient up

\$ns duplex-link-op \$n1 \$d3 orient right

\$ns duplex-link-op \$n3 \$n2 orient down

\$ns duplex-link-op \$n2 \$n5 orient right

\$ns duplex-link-op \$n2 \$s2 orient left

### 19. Lập lịch các sự kiện mô phỏng

\$ns at 0.0 "\$s1 label \"S1\""

\$ns at 0.0 "\$s2 label \"S2\""

\$ns at 0.0 "\$s3 label \"S3\""

\$ns at 0.0 "\$n1 label \"n1\""

\$ns at 0.0 "\$n2 label \"n2\""

\$ns at 0.0 "\$n3 label \"n3\""

\$ns at 0.0 "\$n4 label \"n4\""

\$ns at 0.0 "\$n5 label \"n5\""

\$ns at 0.0 "\$d label \"D\""

\$ns at 0.0 "\$d3 label \"D3\""

## 20. Đặt kích thước băng thông cho các liên kết

\$ns duplex-link-op \$n1 \$n3 label "2Mb"

\$ns duplex-link-op \$n2 \$n3 label "3Mb"

\$ns duplex-link-op \$n3 \$n4 label "4Mb"

\$ns duplex-link-op \$n4 \$n5 label "1Mb"

\$ns duplex-link-op \$n2 \$n5 label "50Kb"

\$ns duplex-link-op \$n1 \$d3 label "5Kb"

\$ns duplex-link-op \$n5 \$d label "1Mb"

## 21. Định nghĩa thủ tục “finish”

```
proc finish { } {
```

```
    global ns nf cwnd f1 f2 f3 tracefile
```

```
    $ns flush-trace
```

## 22. Đóng file NAM

```
        close $nf
```

```
        close $tracefile
```

```
        close $f1
```

```
        close $f2
```

```
        close $f3
```

## 23. Lưu thông tin mất gói tại node \_x của luồng vào file node\_x.dat

```
        exec awk -f node_1.awk TraceAll.tr >node_1.dat
```

```
        exec awk -f node_2.awk TraceAll.tr >node_2.dat
```

```
        exec awk -f node_3.awk TraceAll.tr >node_3.dat
```

```
        exec awk -f node_4.awk TraceAll.tr >node_4.dat
```

```
        exec awk -f node_5.awk TraceAll.tr >node_5.dat
```

```
        exec awk -f queuesize2_5.awk TraceAll.tr > queuesize2_5.tr
```

```

exec awk -f queuesize4_5.awk TraceAll.tr > queuesize4_5.tr
exec awk -f queuesize3_4.awk TraceAll.tr > queuesize3_4.tr
exec nam out.nam &
exec awk -f tre.awk TraceAll.tr > treduongtruyend3.tr
exec xgraph treduongtruyen.tr -geometry 1000x500 &
exec xgraph queuesize2_5.tr -bg white -geometry 800x400 -t " Queue
Length 2_5 " -x " Time (sec) " -y " Packet " &
exec xgraph queuesize4_5.tr -bg white -geometry 800x400 -t " Queue
Length 4_5 " -x " Time (sec) " -y " Packet " &
exec xgraph queuesize3_4.tr -bg white -geometry 800x400 -t " Queue
Length 3_4 " -x " Time (sec) " -y " Packet " &
exec xgraph khathong1.tr khathong2.tr khathong3.tr -bg white -y "Bandwith
(Mbps)" -x "Time (s)" -t "Do thi kha thong" -geometry 800x400 &
exec xgraph node_1.dat node_2.dat node_3.dat node_4.dat node_5.dat -bg white
-t "Ti le mat goi o cac nut " -x "Time (s)" -y "%" -geometry 800x400 &
exit 0
}
proc record {} {

```

```

    global sink1 sink2 sink3 f1 f2 f3
    set ns [Simulator instance]
    set time 0.5

```

#### 24. Số byte nhận được từ mỗi luồng

```

set bw1 [$sink1 set bytes_]
set bw2 [$sink2 set bytes_]
set bw3 [$sink3 set bytes_]

```

#### 25. Cài đặt thời gian hiện tại

```

set now [$ns now]

```

#### 26. Tính toán băng thông và ghi nó vào file

```

puts $f1 "$now [expr $bw1/$time*8/1000000]"
puts $f2 "$now [expr $bw2/$time*8/1000000]"
puts $f3 "$now [expr $bw3/$time*8/1000000]"

```

## 27. Reset giá trị sink

```
$sink1 set bytes_ 0
$sink2 set bytes_ 0
    $sink3 set bytes_ 0
    set now [$ns now]
    $ns at [expr $now+$time] "record"
}
```

## 28. Tạo giá cho các liên kết theo bảng thông

```
$ns cost $n1 $n3 2
$ns cost $n2 $n3 1.33
$ns cost $n1 $d3 800
$ns cost $n3 $n4 1
$ns cost $n2 $n5 80
$ns cost $n4 $n5 4
$ns cost $n4 $d3 800
$ns cost $n3 $n1 2
$ns cost $n3 $n2 1.33
$ns cost $d3 $n1 800
$ns cost $n4 $n3 1
$ns cost $n5 $n2 80
$ns cost $n5 $n4 4
$ns cost $d3 $n4 800
```

## 29. Giám sát hàng đợi

```
$ns duplex-link-op $n1 $n3 queuePos 0.5
$ns duplex-link-op $n2 $n3 queuePos 0.5
$ns duplex-link-op $n3 $n4 queuePos 0.5
$ns duplex-link-op $n4 $n5 queuePos 0.5
```

## 30. Nguồn TCP1

```
set tcp1 [new Agent/TCP]
```

\$ns attach-agent \$s1 \$tcp1

### 31. Đích d-sink1

set sink1 [new Agent/TCPSink/DelAck]

\$ns attach-agent \$d \$sink1

\$ns connect \$tcp1 \$sink1

\$tcp1 set fid\_ 1

\$tcp1 set packetSize\_ 1300

\$tcp1 set rate\_ 5.2Mb

### 32. Tạo file fpt1

set ftp1 [new Application/FTP]

\$ftp1 attach-agent \$tcp1

\$ftp1 set Type\_ FTP

### 33. Nguồn TCP2

set tcp2 [new Agent/TCP]

\$ns attach-agent \$s2 \$tcp2

### 34. Đích d-sink2

set sink2 [new Agent/TCPSink/DelAck]

\$ns attach-agent \$d \$sink2

\$ns connect \$tcp2 \$sink2

\$tcp2 set fid\_ 2

\$tcp2 set packetSize\_ 1300

\$tcp2 set rate\_ 6.24Mb

### 35. Tạo file fpt2

set ftp2 [new Application/FTP]

\$ftp2 attach-agent \$tcp2

\$ftp2 set Type\_ FTP

### 36. Tạo nguồn UDP và tạo kết nối

set udp1 [new Agent/UDP]

\$udp1 set class\_ 3

```
$ns attach-agent $s3 $udp1
set sink3 [new Agent/LossMonitor]
$ns attach-agent $d3 $sink3
$ns connect $udp1 $sink3
$udp1 set packet_size_ 1024
```

### 37. Gửi gói

```
proc sendpacket1 {} {
    global ns udp1 ArrivalTime1 psize
    set time [$ns now]
    $ns at [expr $time + [$ArrivalTime1 value]] "sendpacket1"
    $udp1 send $psize
}
```

```
puts "\n          BAI TAP LON MANG THONG TIN - BAI 15 – D3D \n"
puts "    please wait..."
```

### 38. Thiết lập thời gian mô phỏng

```
$ns at 0.0 "record"
$ns at 0.2 "$ftp1 start"
$ns at 0.2 "sendpacket1"
$ns at 1.1 "$ftp2 start"
$ns at 235.0 "$ftp2 stop"
$ns at 235.5 "$ftp1 stop"
$ns at 240.0 " finish "
```

### 39. Chạy

```
$ns run
```

## V. Thực hiện mô phỏng

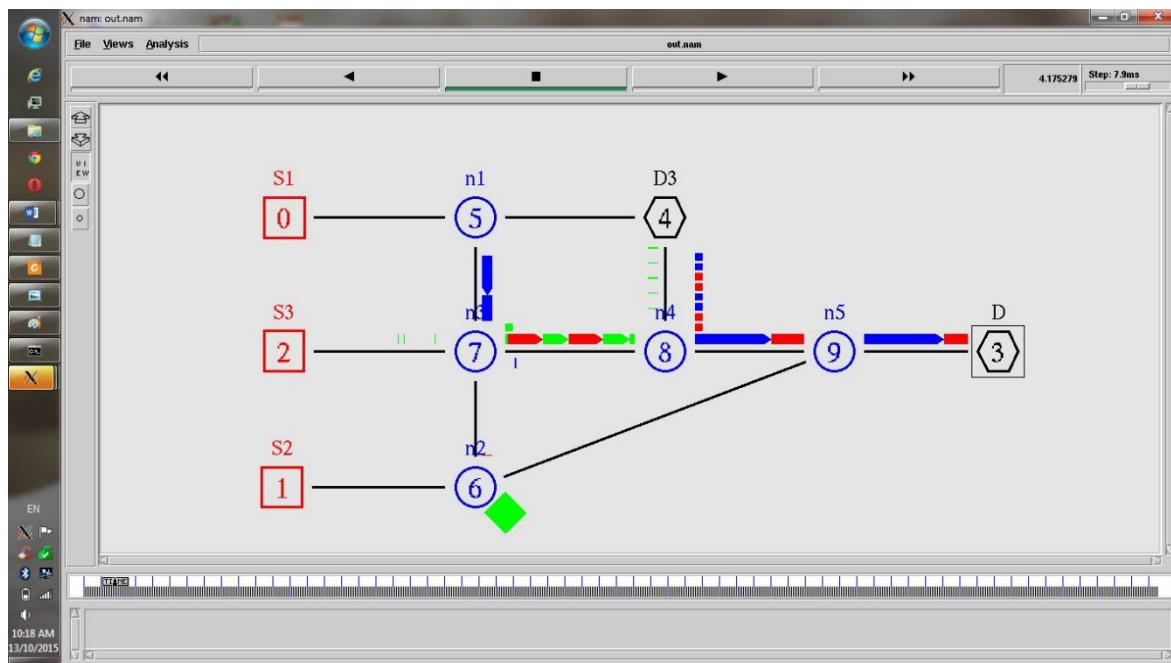
### 1. Kịch bản mô phỏng



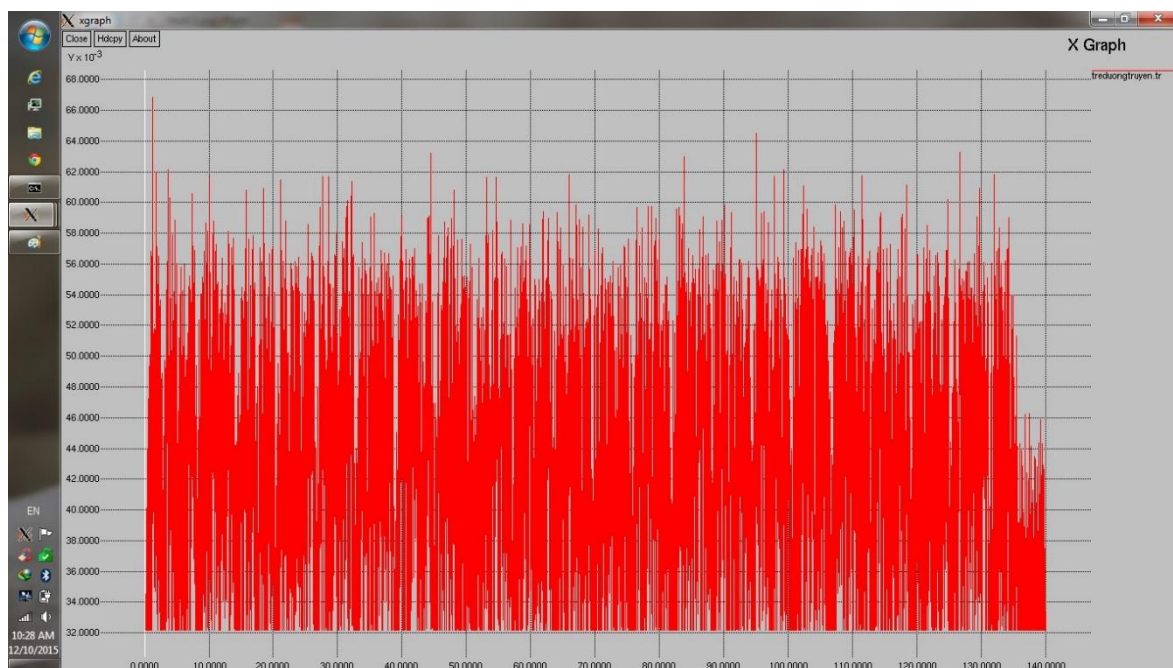
Thời điểm(s)	Mục đích
0.2	Luồng 1, luồng 3 truyền dữ liệu
1.1	Luồng 2 truyền dữ liệu
235.0	Luồng 2 kết thúc truyền dữ liệu
235.5	Luồng 1 kết thúc truyền dữ liệu
240.0	Luồng 3 kết thúc truyền dữ liệu
240.0	Đóng các file trace và data Vẽ đồ thị trễ, tỉ lệ mất gói, trễ truyền thông. Kết thúc mô phỏng.

## 2. Kết quả mô phỏng

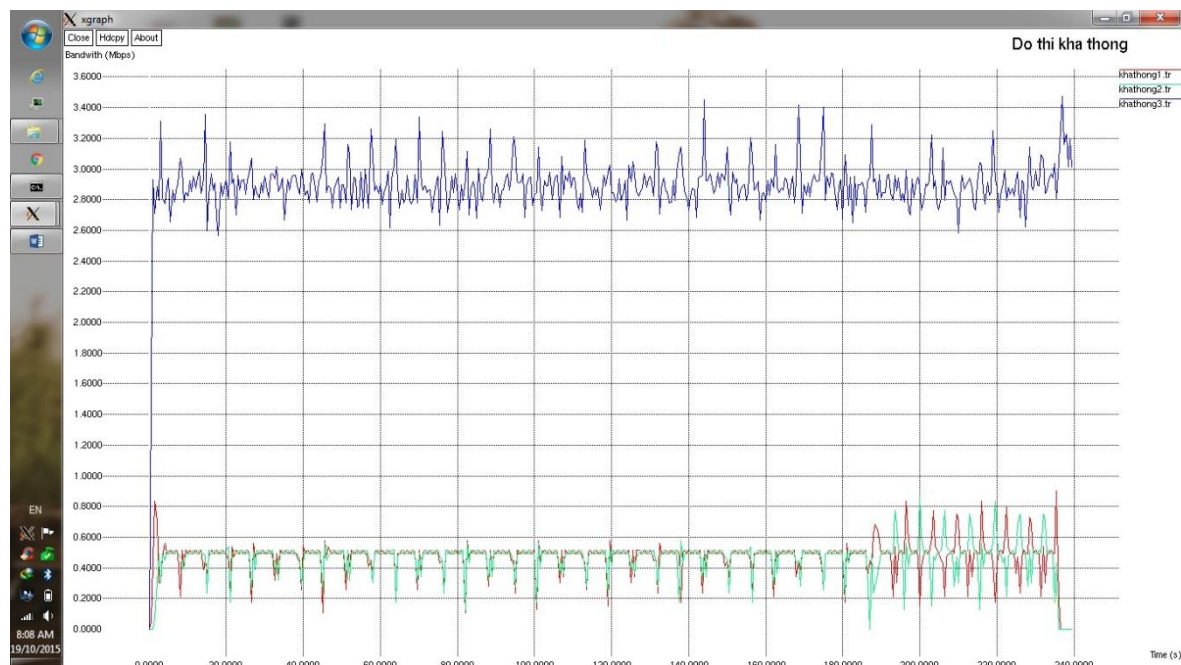
### a) Xây dựng mô hình mạng



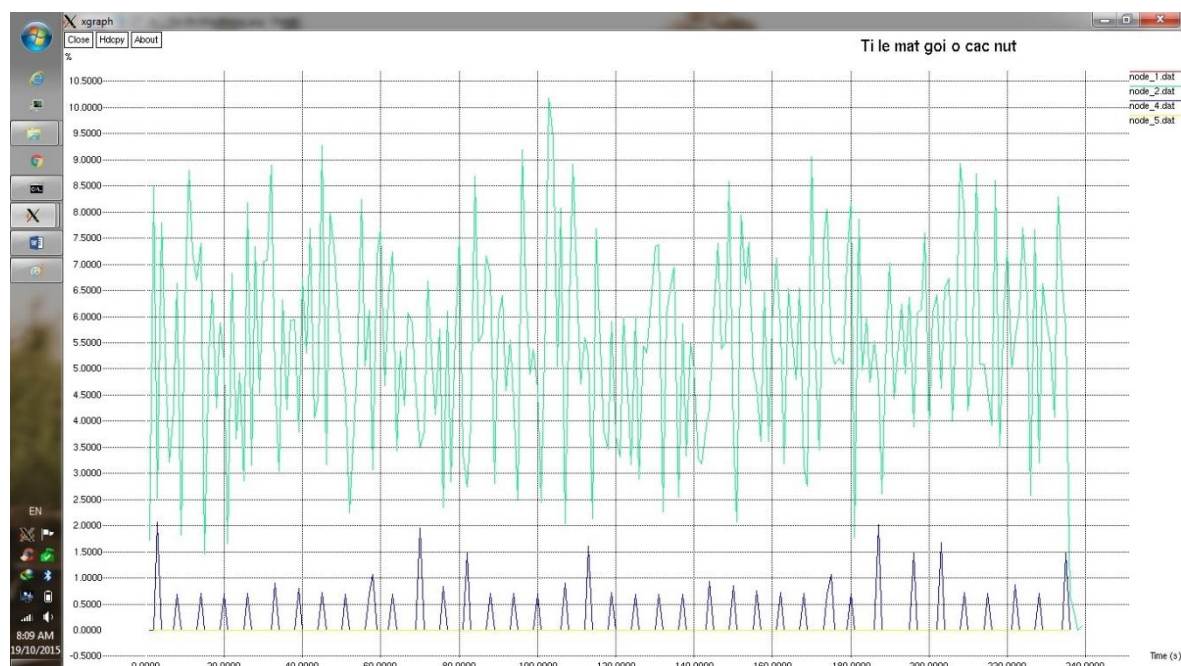
### b) Số lượng gói đến d3 bị trễ nhiều hơn 1s



### c) Đồ thị khả thông



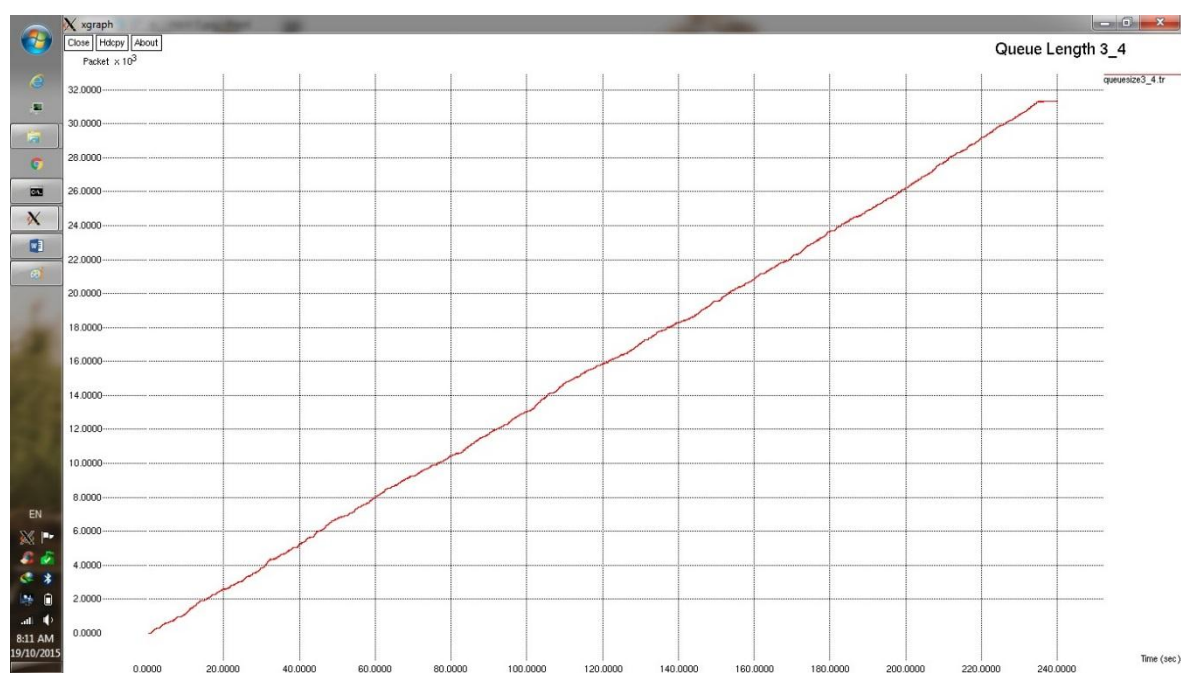
### d) Tỷ lệ mất gói



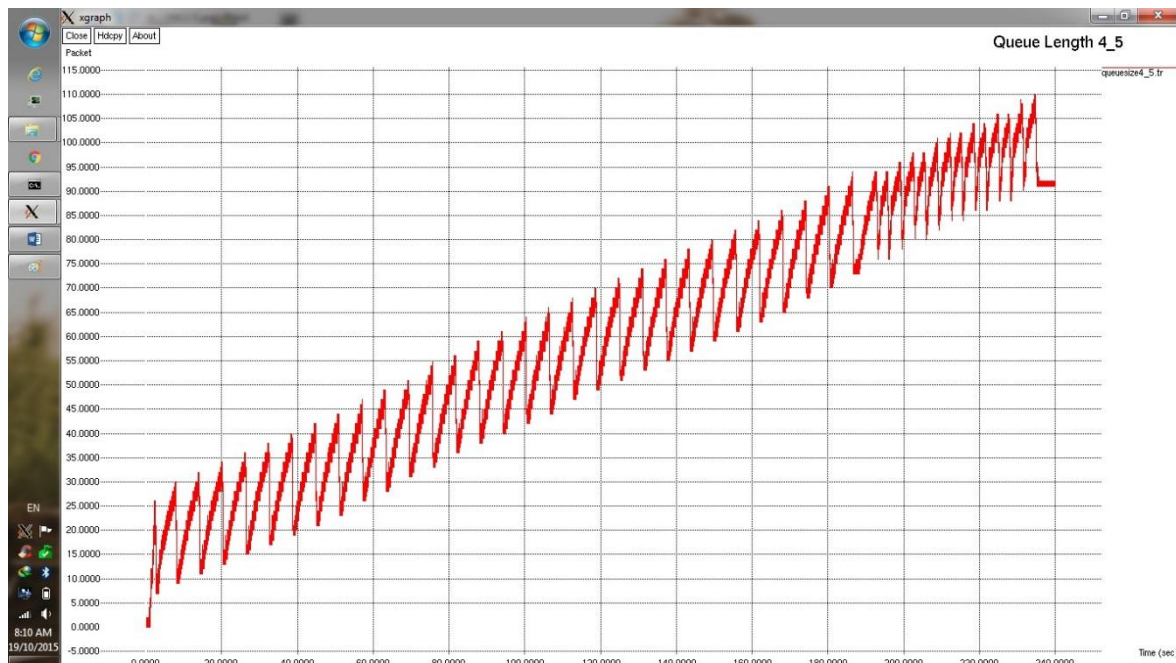
e) Chiều dài hàng đợi trên liên kết node<sub>2</sub> - node<sub>5</sub>



f. Chiều dài hàng đợi trên liên kết node<sub>3</sub> – node<sub>4</sub>



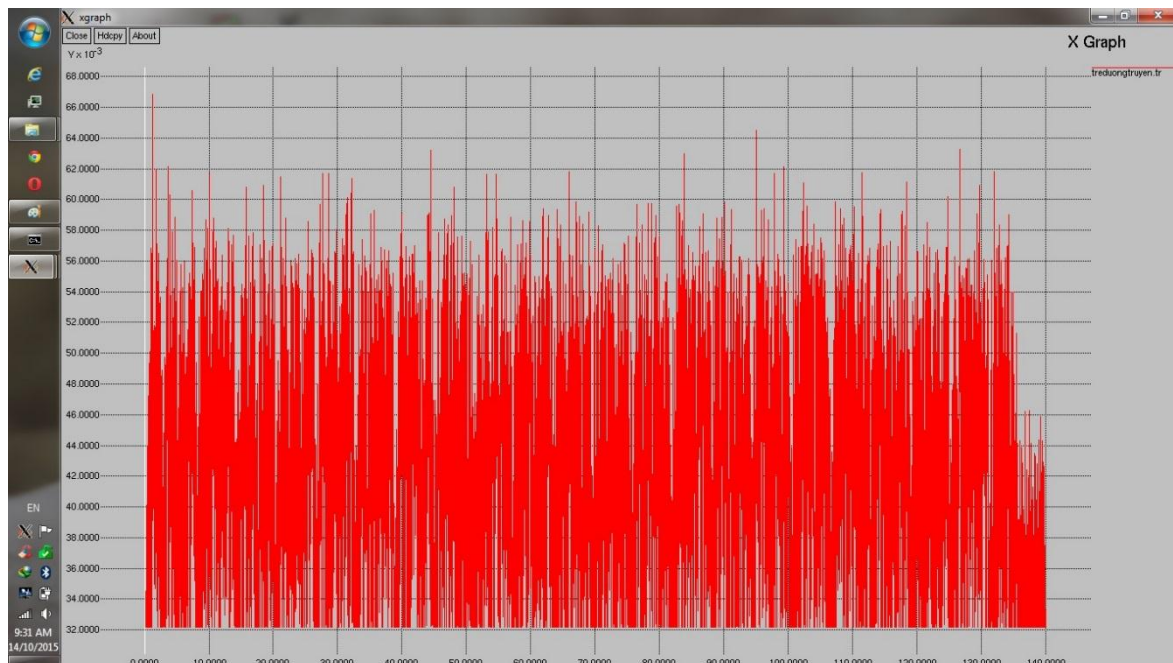
g. Chiều dài hàng đợi trên liên kết node\_4 – node\_5



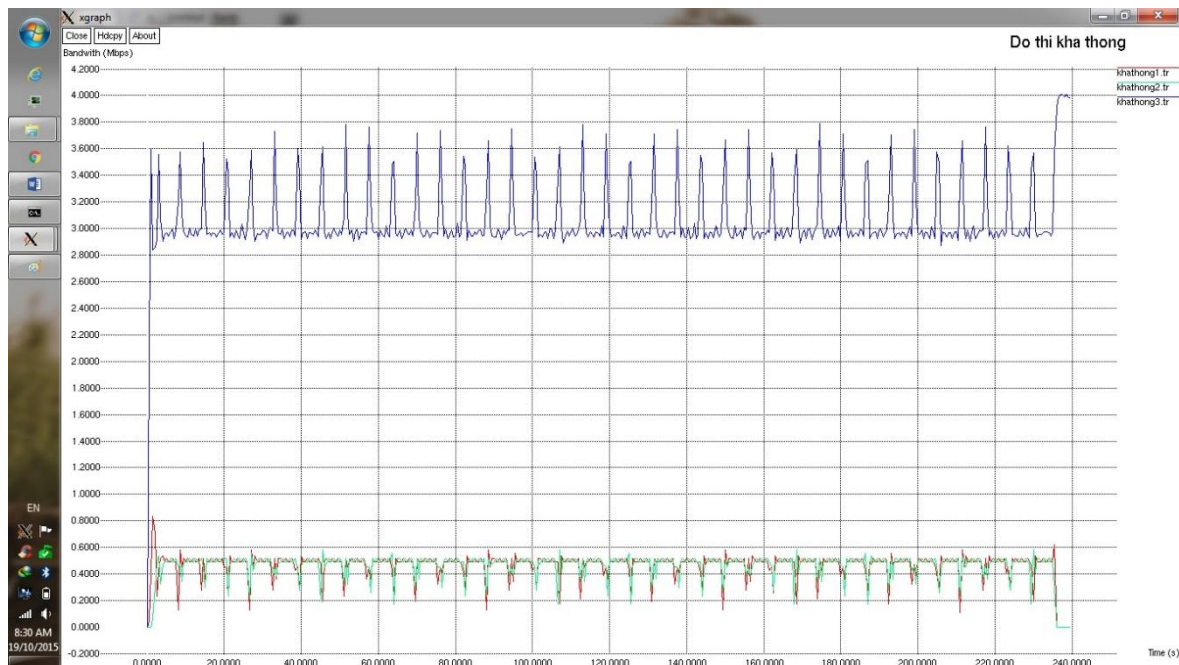
\*\*\* *Khi tăng kích thước gói của luồng UDP ta có*

- *Trễ đường truyền*

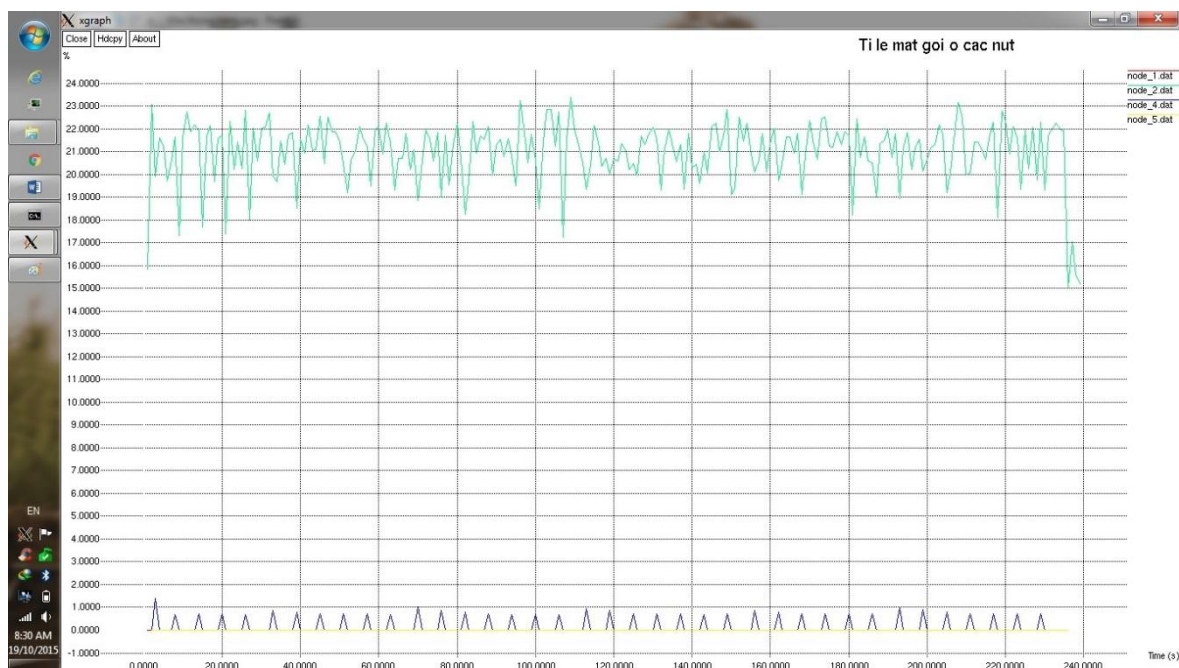




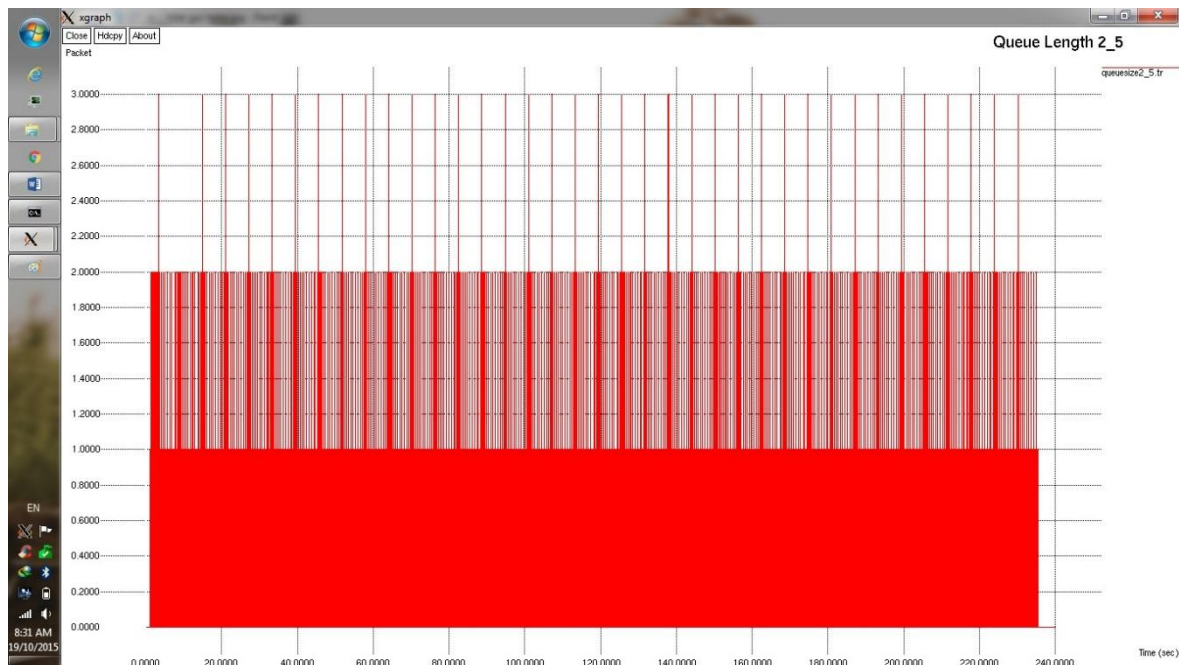
- *Đồ thị khả thông*



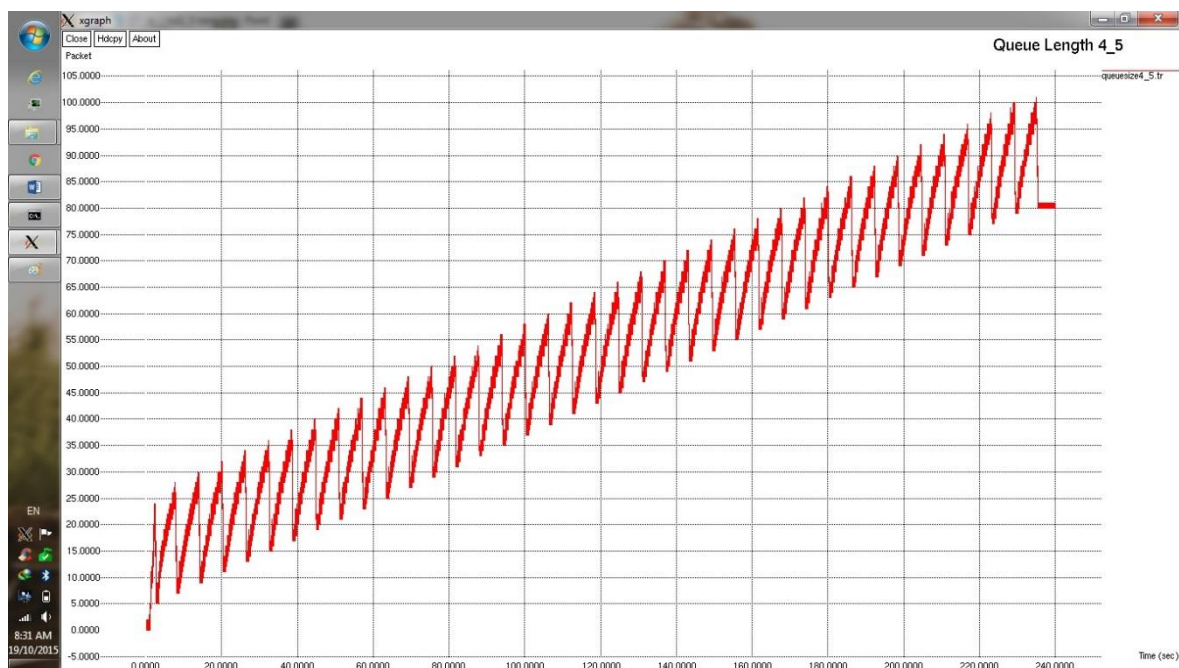
- *Tỷ lệ mất gói*



- Chiều dài hàng đợi trên liên kết node<sub>2</sub> – node<sub>5</sub>

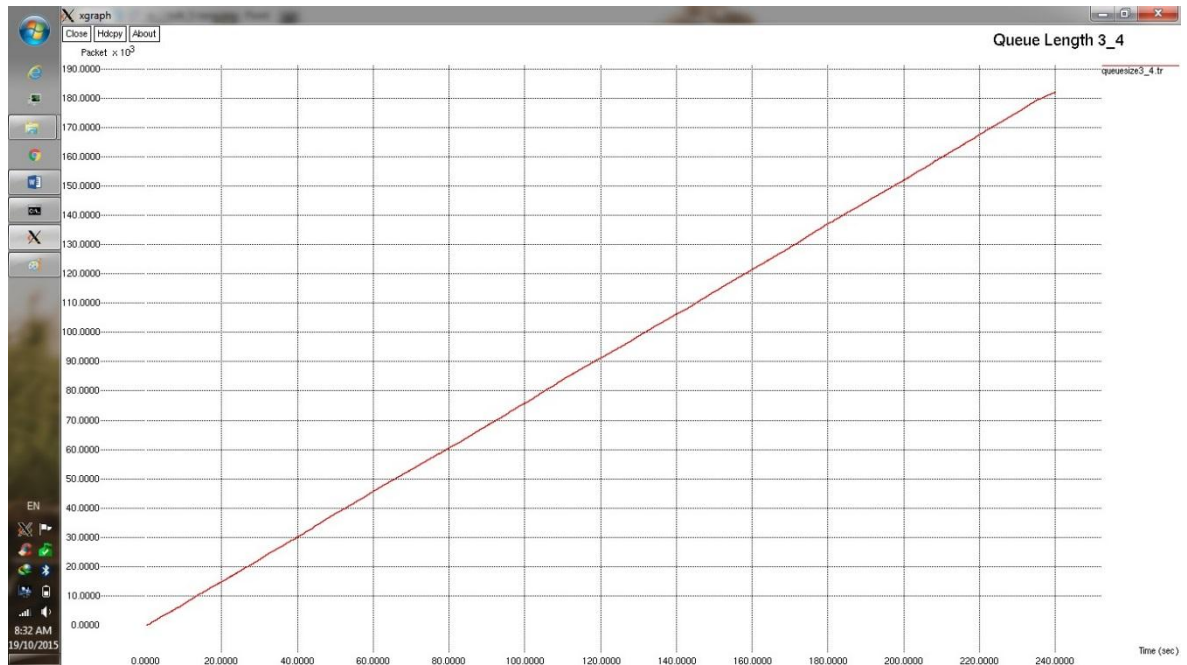


- Chiều dài hàng đợi trên liên kết node<sub>4</sub> – node<sub>5</sub>





- Chiều dài hàng đợi trên liên kết  $node\_3 - node\_4$



**Nhận xét:**

Sau khi thay đổi kích thước gói, tốc độ phát sinh gói của luồng UDP và chạy lại chương trình mô phỏng cho ra kết quả như sau:

- Về đồ thị khả thông:
  - Khả thông trên đường truyền từ nguồn s1 giảm so với ban đầu do có liên kết đường truyền trùng với đường truyền UDP.
  - Khả thông trên đường truyền từ nguồn s2 có tăng so với ban đầu nhưng sự chênh lệch không đáng kể.
  - Khả thông trên đường truyền từ nguồn s3 giảm rõ rệt so với ban đầu do kích thước gói tăng làm đường truyền chậm lại.
- Về tỉ lệ mất gói của từng luồng tại từng nút:
  - Tại nút 1 và nút 5 không mất gói vì không có luồng UDP đi qua 2 nút này nên việc tăng kích thước gói không ảnh hưởng đến 2 nút.
  - Tại nút 2, tỉ lệ mất gói tăng cao so với ban đầu. Ta có thể thấy trên đồ thị là từ 2-10% lên đến 16-23%.
  - Tại nút 3, tỉ lệ mất gói cũng tăng cao do có luồng UDP đi qua.
  - Tại nút 4, tỉ lệ mất gói giảm so với ban đầu do tỉ lệ mất gói luồng 2, 3 tăng dẫn đến số lượng gói đến nút 4 giảm.
- Về kích thước hàng đợi trên các liên kết:
  - Liên kết node\_2 – node\_5: kích thước hàng đợi giảm từ 4 xuống 3 packet theo biểu đồ mô phỏng.
  - Liên kết node\_4 – node\_5: kích thước hàng đợi giảm từ 75 xuống 70 packet.
  - Liên kết node\_3 – node\_4: kích thước hàng đợi tăng rõ rệt do liên kết có luồng UDP và 2 luồng TCP đi qua nên khi tăng kích thước gói làm kích thước hàng đợi tăng rõ rệt.
- Về số lượng gói đến đích d3 bị trễ nhiều hơn 1s:
  - Ta thấy trên đồ thị, số lượng gói đến d3 trễ nhiều hơn 1s tăng do khi tăng kích thước gói, kích thước hàng đợi tăng, khả thông giảm làm trễ đường truyền UDP tăng lên.

## KẾT LUẬN

Qua học phần Mạng thông tin và việc làm bài tập lớn đã giúp chúng em 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, nhưng vẫn chỉ ở mức làm quen và tìm hiểu để biết cách sử dụng chứ chưa thể sử dụng thành thạo được. Tuy nhiên, thông qua việc thực hiện bài tập lớn 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, củng cố 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 sẽ còn nhiều thiếu sót, mong nhận được ý kiến nhận xét quý báu của Thầy, cô để bài tập lớn của chúng em được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

## TÀI LIỆU THAM KHẢO

1. Tập 2 Mô phỏng trong NS2 – vntelecom.org
2. Tài liệu hướng dẫn NS2: <http://www.isi.edu/nsnam/ns/tutorial/>
3. Một số tài liệu do thầy Nguyễn Đăng Hiếu cung cấp.