

THỬ NGHIỆM CƠ CHẾ HOẠT ĐỘNG VÀ ĐÁNH GIÁ HIỆU NĂNG CỦA GIAO THỨC TCP SỬ DỤNG NS-2

I. MỤC ĐÍCH THÍ NGHIỆM

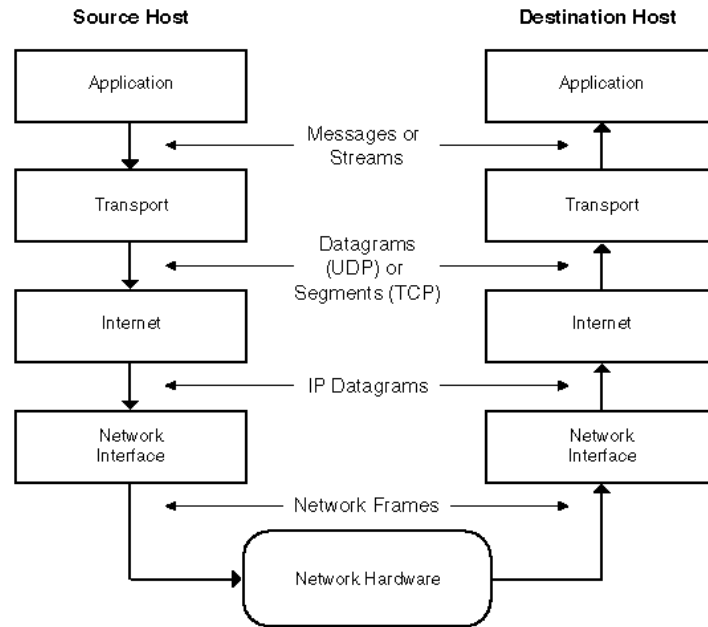
Trong bài 8 và bài 9, sinh viên đã được làm quen với phần mềm mô phỏng NS-2 và cách tính toán các thông số cơ bản của mạng. Bài thí nghiệm 10 này có một số mục đích sau:

- Tìm hiểu cơ chế hoạt động của TCP, cơ chế điều khiển luồng sử dụng congestion window.
- Các thể độ hoạt động của TCP.
- Đánh giá hiệu năng của TCP.

II. THẢO LUẬN

2.1. Mô hình TCP/IP

Mặc dù mô hình tham chiếu OSI được chấp nhận rộng rãi khắp nơi, nhưng chuẩn mở về kỹ thuật mang tính lịch sử của Internet lại là TCP/IP. Mô hình tham chiếu TCP/IP và chồng giao thức TCP/IP tạo khả năng truyền dữ liệu giữa hai máy tính bất kỳ từ bất cứ nơi nào trên thế giới, với vận tốc gần bằng tốc độ của ánh sáng. Mô hình TCP/IP có tầm quan trọng trong lịch sử, gần giống như các chuẩn đã cho phép điện thoại, năng lượng điện, đường sắt, truyền hình và công nghệ băng hình phát triển cực thịnh.



Hình 2.1. Mô hình TCP/IP

- Mô hình TCP/IP bao gồm 4 tầng:
- + Application layer (Tầng ứng dụng)

Lớp ứng dụng kiểm soát các giao thức mức cao, các vấn đề của lớp trình bày, mã hóa và điều khiển hội thoại. TCP/IP tập hợp tất cả các vấn đề liên quan đến ứng dụng vào trong một lớp, và đảm bảo dữ liệu được đóng gói một cách thích hợp cho lớp kế tiếp.

- + Transport layer (Tầng giao vận)

Lớp vận chuyển đề cập đến các vấn đề chất lượng dịch vụ như độ tin cậy, điều khiển luồng và sửa lỗi. Một trong các giao thức của nó là TCP, TCP cung cấp các phương thức linh hoạt và hiệu quả để thực hiện các hoạt động truyền dữ liệu tin cậy, hiệu suất cao và ít lỗi.

- + Internet layer (Tầng mạng)

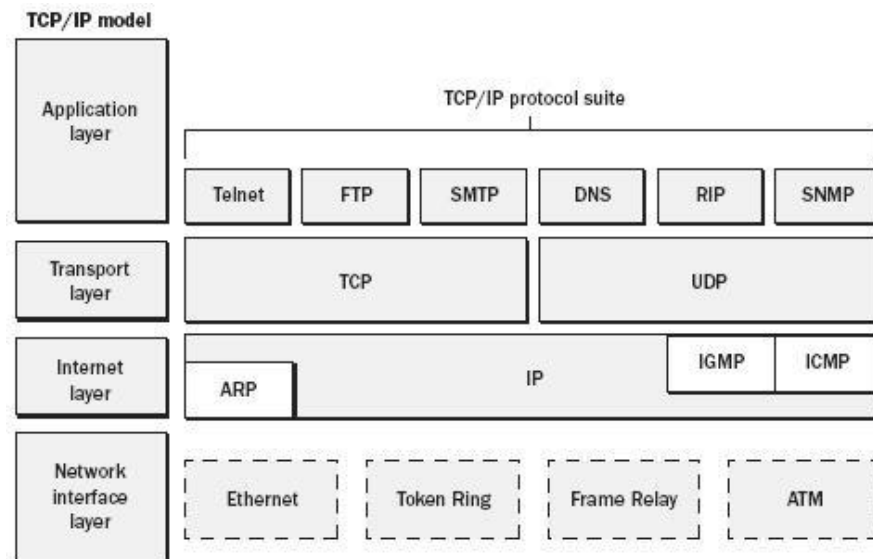
Mục tiêu của lớp Internet là truyền các gói bắt nguồn từ bất kỳ host nào trên liên mạng và đến được đích trong điều kiện độc lập với đường dẫn và các mạng mà chúng đã trải qua. Giao thức đặc trưng không chế lớp này được gọi là IP. Công việc xác định đường dẫn tốt nhất và hoạt động chuyển mạch gói diễn ra tại lớp này.

- + Link layer (Tầng liên kết)

Nó là lớp liên quan đến tất cả các vấn đề mà một gói IP yêu cầu để tạo một liên kết vật lý thực sự, và sau đó tạo một liên kết vật lý khác. Nó bao gồm các chi tiết kỹ thuật LAN và WAN, và tất cả các chi tiết trong lớp liên kết dữ liệu cũng như lớp vật lý của mô hình OSI.

- **Chồng giao thức TCP/IP**

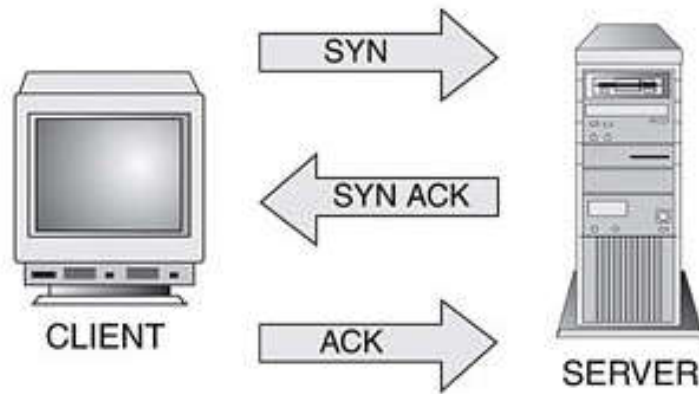
TCP/IP là bộ giao thức truyền thông chung giữa các máy tính trong mạng internet. Nó bao gồm một tập hợp các chuẩn mạng đặc tả chi tiết cách thức mà các máy tính thông tin liên lạc với nhau, cũng như quy ước cho các truyền thông liên mạng. TCP/IP được sử dụng rộng rãi trong mạng cục bộ cũng như mạng n Internet toàn cầu.



Hình 2.2. Chồng giao thức TCP/IP

2.2. Transmission Control Protocol (TCP)

- **Đặc điểm của giao thức TCP**
 - + TCP là giao thức cho phép truyền dữ liệu một cách tin cậy giữa 2 máy tính bất kỳ. Được sử dụng ở tầng Transport Layer.
 - + TCP sử dụng bản tin xác nhận ACK để báo nhận gói tin.
 - + TCP sử dụng phương pháp điều khiển luồng theo cửa sổ trượt với 2 loại cửa sổ là congestion window bên phát và advertised window bên thu.
- ⇒ **Tính chất của TCP:**
 - + Truyền dữ liệu tin cậy, không lỗi do có cơ chế sửa lỗi/truyền lại.
 - + Truyền gói theo thứ tự.
 - + Loại bỏ gói lặp.
 - + Hạn chế tắc nghẽn.
- **Hoạt động của TCP**
 - + Thiết lập kết nối (Threeway handshake)



Hình 2.3. TCP Threeway handshake

+ Quá trình truyền dữ liệu

- Slow start

Ở trạng thái slow start, khi kết nối được thiết lập, giá trị của cửa sổ trượt sẽ được đặt bằng 1, và sau mỗi ACK nhận được, kích thước cửa sổ trượt sẽ tăng lên thành

$$cwnd = cwnd + 1$$

tức kích thước cửa sổ trượt sẽ tăng gấp đôi sau mỗi Round Trip Time (RTT)

Kích thước cửa sổ sẽ tăng theo cấp số nhân cho đến khi xảy ra mất gói, giá trị slow start threshold (sssthresh) sẽ được cập nhật

$$sssthresh = cwnd/2$$

Sau khi mất gói, kết nối trở lại trạng thái slow start ban đầu với $cwnd = 1$, kích thước cửa sổ sẽ tăng theo cấp số nhân đến khi bằng sssthresh, giá trị ước tính cho băng thông khả dụng của mạng. Lúc này, kết nối chuyển sang trạng thái congestion avoidance.

- Congestion avoidance

Giá trị của cửa sổ trượt sẽ tăng chậm hơn ở trạng thái này với

$$cwnd = cwnd + 1/cwnd$$

tức giá trị cửa sổ tăng lên 1 sau mỗi RTT, tăng tuyến tính thay vì tăng theo cấp số nhân. Quá trình này sẽ tiếp tục đến khi phát hiện gói mất.

- Fast Retransmit – Three Duplicate ACKs

Duplicate ACKs là một cách để phát hiện gói mất, và có thể gây ra do các gói được phát lại. Khi nhận được 1 gói ACK lặp, bên gửi không biết chắc chắn được gói đã mất hay bị lỗi nhưng sau vài ACK lặp, có cơ sở để khẳng định rằng đã xảy ra mất gói. Mục đích của cơ chế Fast Retransmit là để tăng tốc quá trình phát lại gói bằng cách cho phép bên gửi gửi lại gói ngay khi có đủ bằng chứng rằng hiện tượng mất gói đã xảy ra. Thường thì sau 3 ACK lặp là có thể coi là đã đủ bằng chứng – Three

Duplicate ACKs. Có nghĩa là thay vì phải đợi hết timeout, bên gửi có thể gửi lại gói ngay lập tức sau khi nhận được 3 ACK lặp.

- Fast Recovery

Trong kết nối TCP Tahoe, kết nối luôn quay trở về trạng thái slow start sau khi mất gói xảy ra. Tuy nhiên vì kích thước cửa sổ lớn và hiện tượng mất gói ít, sẽ tốt hơn nếu kết nối được tiếp tục luôn ở trạng thái congestion avoidance, vì nó sẽ rất mất thời gian để tăng lại kích thước cửa sổ từ 1 đến ssthresh. Mục đích của Fast Recovery trong TCP Reno là để đạt được điều này.

Trong một kết nối sử dụng thuật toán Fast Retransmit, bên gửi có thể dùng luồng Duplicate ACKs để đo thời gian truyền lại gói. Khi có gói mất được truyền lại, giá trị của ssthresh và cwnd sẽ được đặt lại

$$\text{ssthresh} = \text{cwnd}/2$$

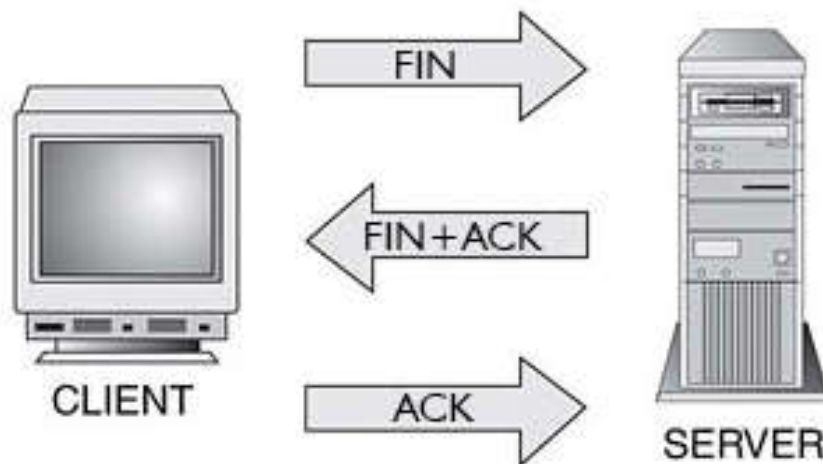
$$\text{cwnd} = \text{ssthresh}$$

Có nghĩa là kết nối sẽ được tiếp tục ở trạng thái congestion avoidance và tăng tuyến tính kích thước cửa sổ.

- Delayed ACK

Để tiết kiệm băng thông, không nhất thiết phải gửi ACK cho mỗi gói tin nhận được. Sau vài gói tin mới cần phải gửi 1 ACK để báo nhận cho toàn bộ. Đó gọi là Delayed ACK.

+ Ngắt kết nối



Hình 2.4. TCP Finish handshake

2.3. User Datagram Protocol (UDP)

UDP là một trong những giao thức cốt lõi của chồng giao thức TCP/IP. Dùng UDP, chương trình trên mạng máy tính có thể gửi những dữ liệu ngắn được gọi là datagram tới máy khác.

UDP không cung cấp sự tin cậy và thứ tự truyền nhận mà TCP làm; 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 các mục tiêu như kích thước nhỏ và yêu cầu khẩn trương về thời gian.

Do bản chất không trạng thái của nó nên nó hữu dụng đối với việc trả lời các truy vấn nhỏ với số lượng lớn người yêu cầu. Những ứng dụng phổ biến sử dụng UDP như Domain Name System (DNS), ứng dụng streaming media, Voice over IP (VoIP), Trivial File Transfer Protocol (TFTP), và game trực tuyến.

2.4. So sánh TCP và UDP

	TCP (Transmission Control Protocol)	UDP (User Datagram Protocol)
Lớp	Transport	Transport
Phương thức truyền	Reliable	Best-effort
Giao thức kết nối	Connection-oriented	Connectionless
Khôi phục gói mất	Có	Không
Thứ tự	Có	Không
Điều khiển luồng	Có	Không
Ứng dụng	E-mail Download File sharing	Voice/ Video chat Voice/Video streaming

Bảng 2.1. Bảng so sánh TCP và UDP

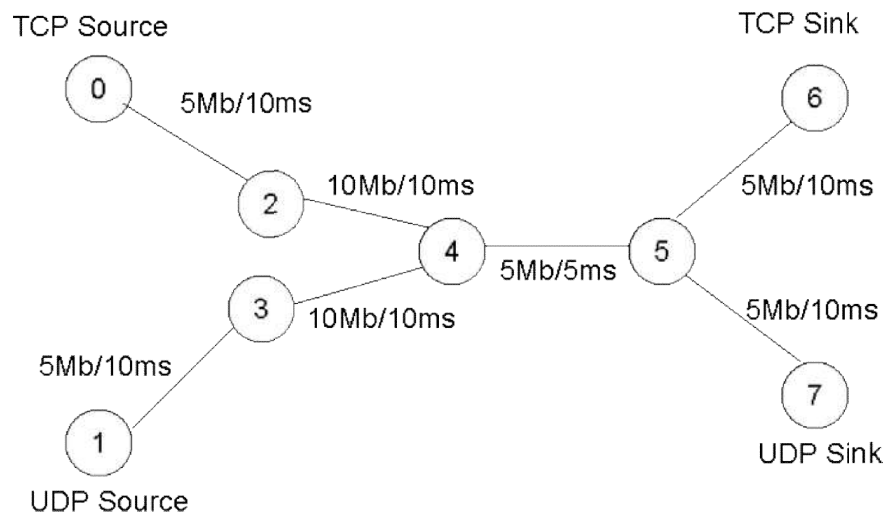
III. YÊU CẦU VỀ THIẾT BỊ

- Máy tính chạy hệ điều hành Ubuntu 14.04
- Cài đặt phần mềm Network Simulator 2 (NS-2)

IV. NỘI DUNG THÍ NGHIỆM

TCP (Transmission Control Protocol) là giao thức cho phép truyền dữ liệu một cách tin cậy giữa 2 máy tính bất kỳ. TCP dựa vào phương pháp điều khiển luồng theo cửa sổ trượt. Mặt khác, UDP là giao thức không đưa ra các cơ chế truyền tin một cách tin cậy, tuy nhiên UDP lại có đặc điểm là đơn giản, trễ truyền dữ liệu nhỏ, hiệu suất truyền lớn nên thích hợp cho các ứng dụng thời gian thực. Một câu hỏi đặt ra là khi truyền cả TCP và UDP trên mạng Internet thì 2 giao thức trên chia sẻ băng thông như thế nào, hoạt động của chúng ra sao trong trường hợp mạng xảy ra tắc nghẽn. Bài thí nghiệm này cho phép sinh viên khảo sát và trả lời các câu hỏi trên thông qua các bước như sau:

- Tạo ra một mạng cần mô phỏng sử dụng NS-2 giống như trên hình vẽ. Tạo ra 2 kết nối TCP và UDP sử dụng nguồn/đích TCP và nguồn/đích UDP. Nguồn của TCP sử dụng TCP Reno (TCP Reno là một phiên bản của TCP đã được hỗ trợ trong NS-2).
- Thiết lập cửa sổ chống tắc nghẽn tối đa (maximal congestion window size) của luồng TCP là 32 gói.
- Kích thước các gói UDP và TCP là 512 bytes.
- Chạy mô phỏng trong 200s.
- Các nguồn UDP và TCP đều phát gói liên tục. Luồng UDP có tốc độ phát là 1 Mbit/s (sử dụng nguồn CBR)



Hình 4.1. Mạng cần mô phỏng

- Chạy mô phỏng, đo và vẽ đồ thị các tham số khi:
 1. Chỉ cho nguồn TCP phát gói. Thay đổi kích thước cửa sổ trượt tối đa là 32 và 128.
 2. Tốc độ phát của nguồn UDP là 1 Mbit/s, 2 Mbit/s, 3 Mbit/s và 4Mbit/s. Kích thước cửa sổ trượt tối đa của nguồn TCP là 32.

3. Tốc độ phát của nguồn UDP là 1Mbit/s, thay đổi kích thước của sổ trượt tối đa lên 64 và 128.

- Tạo thêm một luồng TCP y hệt nguồn cũ với nguồn gắn vào nút số 3 và đích gắn vào nút số 5. Chạy mô phỏng, đo và vẽ đồ thị các tham số khi:

1. Luồng UDP không phát và phát với tốc độ 1 Mbit/s. Kích thước của sổ trượt tối đa của cả 2 luồng TCP là 128.

2. Luồng UDP phát với tốc độ 1 Mbit/s. Luồng TCP mới tạo phát chậm hơn 50 giây.

V. TRÌNH TỰ THÍ NGHIỆM

5.1. Mạng gồm 1 luồng TCP và 1 luồng UDP

5.1.1. Tạo folder

Trong folder *MSSV* đã tạo ở bài 8 và bài 9, tạo folder **bai10**. Trong folder vừa tạo, tiếp tục tạo 2 folder *1tcp1udp* và *2tcp1udp*

5.1.2. Viết file .tcl

Bước 1: Vào folder *1tcp1udp*, tạo file *1tcp1udp.tcl*. Mở file *1tcp1udp.tcl* vừa tạo bằng gedit hoặc trình chỉnh sửa văn bản bất kỳ.

Bước 2: Viết chương trình

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

```
$ns color 1 Red
```

```
$ns color 2 Green
```

```
$ns color 3 Blue
```

```
set pksize 512
```

```
set tcpbw [open tcpbw.tr w]
```

```
set udpbw [open udpbw.tr w]
```

```
set tcpdl [open tcpdl.tr w]
```

```
set udpdl [open udpdl.tr w]
```

```
set tcplr [open tcplr.tr w]
```

```
set udplr [open udplr.tr w]
```

+ congestion window

```
set cwndsize [open "cwnd.tr" w]
```



```

set tf [open Itcp1udp.tr w]
$ns trace-all $tf
set nf [open Itcp1udp.nam w]
$ns namtrace-all $nf

proc finish {} {
    global ns tf nf tcpbw udpbw tcpdl udpdl tcplr udplr
    $ns flush-trace
    close $udpbw
    close $tcpbw
    close $tcpdl
    close $udpdll
    close $tcplr
    close $udplr
    close $nf
    close $tf

    exec nam Itcp1udp.nam &

    exec awk -f bandwidth.awk Itcp1udp.tr
    exec xgraph tcpbw.tr udpbw.tr -bg "black" -fg "white" -geometry 1000x500 -t
    "Bandwidth" -x "s" -y "Mbps" &

    exec awk -f lostrate.awk Itcp1udp.tr
    exec xgraph tcplr.tr udplr.tr -bg "black" -fg "white" -geometry 1000x500 -t
    "Lost Rate" -x "s" -y "%" &

    exec awk -f delay.awk Itcp1udp.tr
    exec xgraph tcpdl.tr udpdl.tr -bg "black" -fg "white" -geometry 1000x500 -t
    "Delay" -x "s" -y "ms" &

    exec xgraph cwnd.tr -bg "black" -fg "white" -geometry 1000x500 -t
    "WindowSize" -x "s" -y "Size" &

    exit 0
}

set s0 [$ns node]
$s0 label "TCPSource"

```

*set s1 [\$ns node]
\$s1 label "UDPSource"
set n2 [\$ns node]
\$n2 label "Node2"
set n3 [\$ns node]
\$n3 label "Node3"
set n4 [\$ns node]
\$n4 label "Node4"
set n5 [\$ns node]
\$n5 label "Node5"
set d6 [\$ns node]
\$d6 label "TCPSink"
set d7 [\$ns node]
\$d7 label "UDPNull"*

*\$s0 shape "hexagon"
\$s0 color "red"
\$s1 shape "hexagon"
\$s1 color "green"
\$d6 shape "square"
\$d6 color "red"
\$d7 shape "square"
\$d7 color "green"*

*\$ns duplex-link \$s0 \$n2 5Mb 10ms DropTail
\$ns duplex-link \$n2 \$n4 10Mb 10ms DropTail
\$ns duplex-link \$s1 \$n3 5Mb 10ms DropTail
\$ns duplex-link \$n3 \$n4 10Mb 10ms DropTail
\$ns duplex-link \$n4 \$n5 5Mb 5ms DropTail
\$ns duplex-link \$n5 \$d6 5Mb 10ms DropTail
\$ns duplex-link \$n5 \$d7 5Mb 10ms DropTail*

*\$ns duplex-link-op \$s0 \$n2 orient right
\$ns duplex-link-op \$s1 \$n3 orient right
\$ns duplex-link-op \$n4 \$n2 orient left-up
\$ns duplex-link-op \$n4 \$n3 orient left-down
\$ns duplex-link-op \$n4 \$n5 orient right
\$ns duplex-link-op \$n5 \$d6 orient right-up
\$ns duplex-link-op \$n5 \$d7 orient right-down*

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

\$ns queue-limit \$n2 \$n4 20
\$ns queue-limit \$n3 \$n4 20
\$ns queue-limit \$n4 \$n5 20

set tcp [new Agent/TCP/Reno]
\$tcp set packetSize_ \$pksize
\$tcp set window_ 128
\$tcp set maxcwnd_ 128
set sink [new Agent/TCPSink]
\$ns attach-agent \$s0 \$tcp
\$ns attach-agent \$d6 \$sink
\$ns connect \$tcp \$sink
\$tcp set fid_ 1
set ftp [new Application/FTP]
\$ftp attach-agent \$tcp
\$ftp set type_ FTP

set udp [new Agent/UDP]
set null [new Agent/LossMonitor]
\$ns attach-agent \$s1 \$udp
\$ns attach-agent \$d7 \$null
\$ns connect \$udp \$null
\$udp set fid_ 2
set cbr [new Application/Traffic/CBR]
\$cbr set packetSize_ \$pksize
\$cbr set rate_ 1Mb
\$cbr set random_ false
\$cbr attach-agent \$udp
\$cbr set type_ CBR

proc plotWindow {tcpSource cwndsize} {
global ns
set now [\$ns now]
set time 1
set cwnd [\$tcpSource set cwnd_]

```
puts $wndsize "$now $wnd"
$ns at [expr $now+$time] "plotWindow $tcpSource $wndsize"
}
```

```
$ns at 0.1 "plotWindow $tcp $wndsize"
$ns at 0.1 "$cbr start"
$ns at 0.1 "$ftp start"
$ns at 200.1 "$ftp stop"
$ns at 200.1 "$cbr stop"
$ns at 200.2 "finish"
```

\$ns run

5.1.3. Viết file .awk

Dựa vào bài 9, viết các file .awk tương ứng để tính 3 tham số: lưu lượng, trễ, tỉ lệ mất gói của từng luồng.

Chú ý:

- Tên file trace .tr tạo ra được đặt như trong phần khai báo ở file .tcl.
- Hàm tính và vẽ kích thước của sổ trượt đã được viết sẵn trong file .tcl.

5.1.4. Chỉ cho nguồn TCP phát gói.

- Thay đổi kích thước cửa sổ trượt tối đa là 32 và 128.
- Điền các giá trị vào bảng sau:

Kích thước cửa sổ trượt tối đa	32	128
Lưu lượng (Mbps)		
Trễ (ms)		
Tỉ lệ mất gói trung bình (%)		
Kích thước cửa sổ trượt trung bình		

- Nhận xét về hoạt động của nguồn TCP:

.....

.....

.....

.....

5.1.5. Thay đổi tốc độ phát của luồng UDP

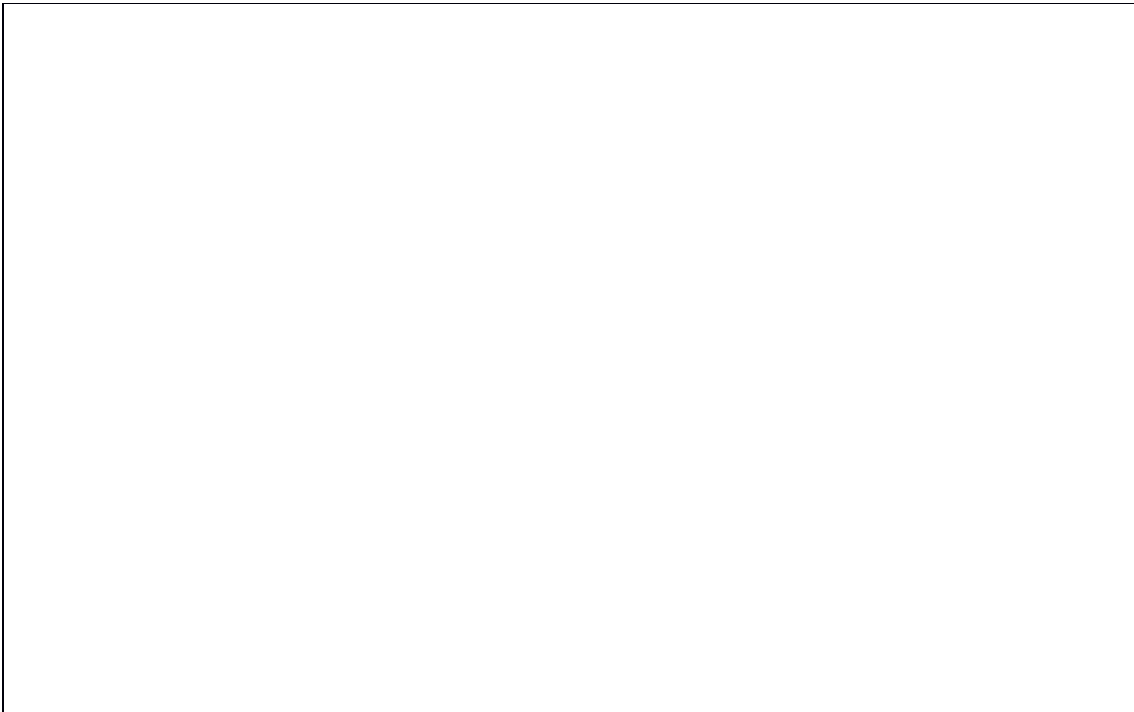
Đặt kích thước cửa sổ trượt tối đa của luồng TCP là 32. Thay đổi tốc độ phát của luồng UDP để mô phỏng các trường hợp khi tốc độ của luồng UDP lần lượt là 1 Mbps, 2 Mbps, 3 Mbps và 4 Mbps.

5.1.5.1. Mô phỏng

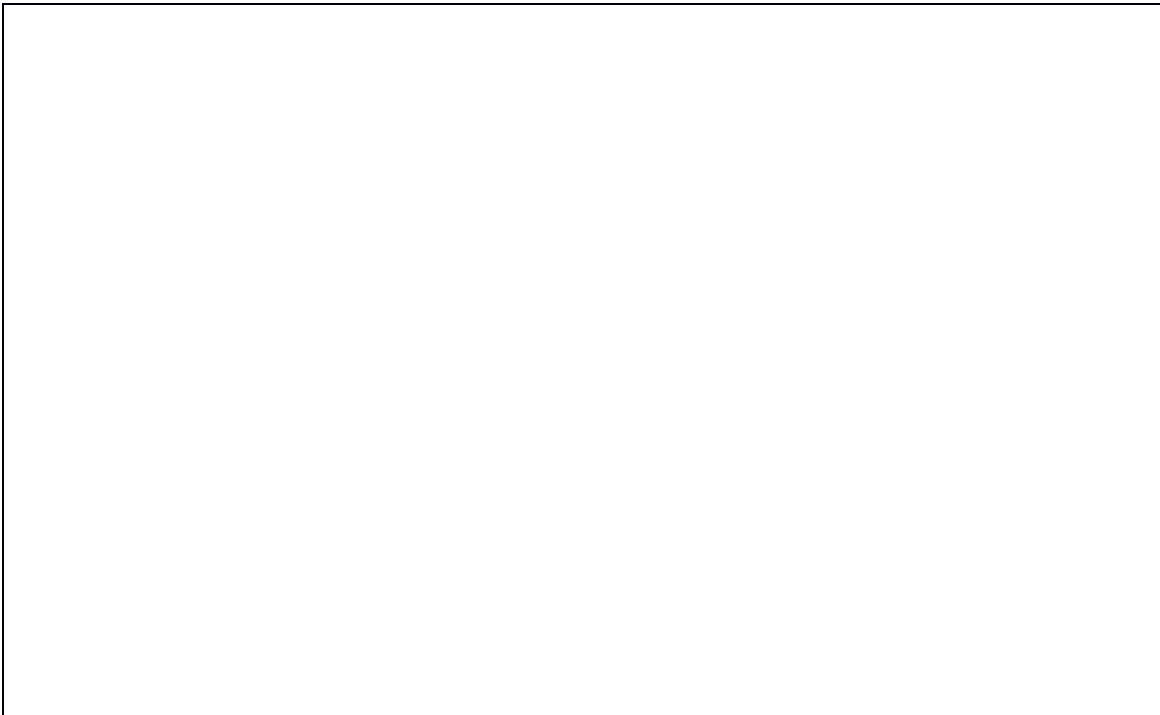
- Điền các giá trị vào bảng sau:

UDP rate (Mbps)		1	2	3	4
Lưu lượng (Mbps)	TCP				
Trễ (ms)	TCP				
	UDP				
Tỉ lệ mất gói (%)	TCP				
	UDP				
Kích thước cửa sổ trượt trung bình	TCP				

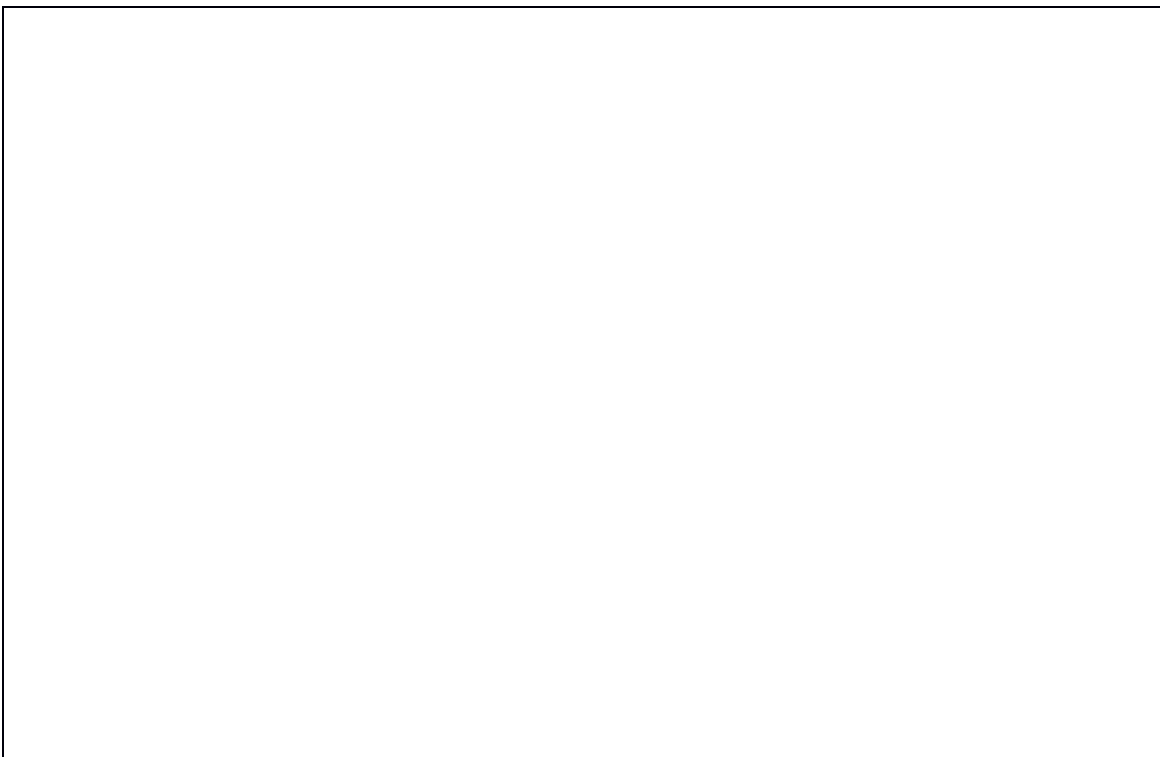
- Đồ thị lưu lượng các luồng trong tất cả các trường hợp:



Đồ thị 5.1. Lưu lượng các luồng khi tốc độ phát của luồng UDP là 1 Mbps.



Đồ thị 5.2. Lưu lượng các luồng khi tốc độ phát của luồng UDP là 2 Mbps.

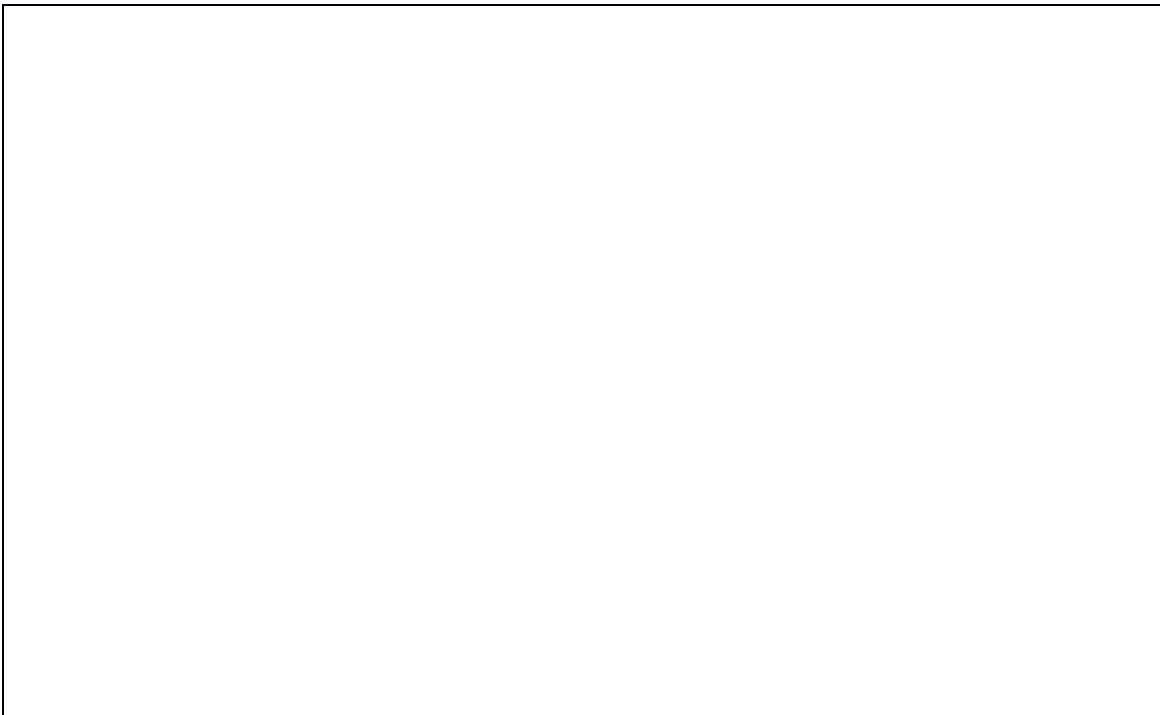


Đồ thị 5.3. Lưu lượng các luồng khi tốc độ phát của luồng UDP là 3 Mbps.



Đồ thị 5.4. Lưu lượng các luồng khi tốc độ phát của luồng UDP là 4 Mbps.

- Đồ thị tỉ lệ mất gói trong các trường hợp tốc độ phát của luồng UDP là 1 Mbps và 4 Mbps.

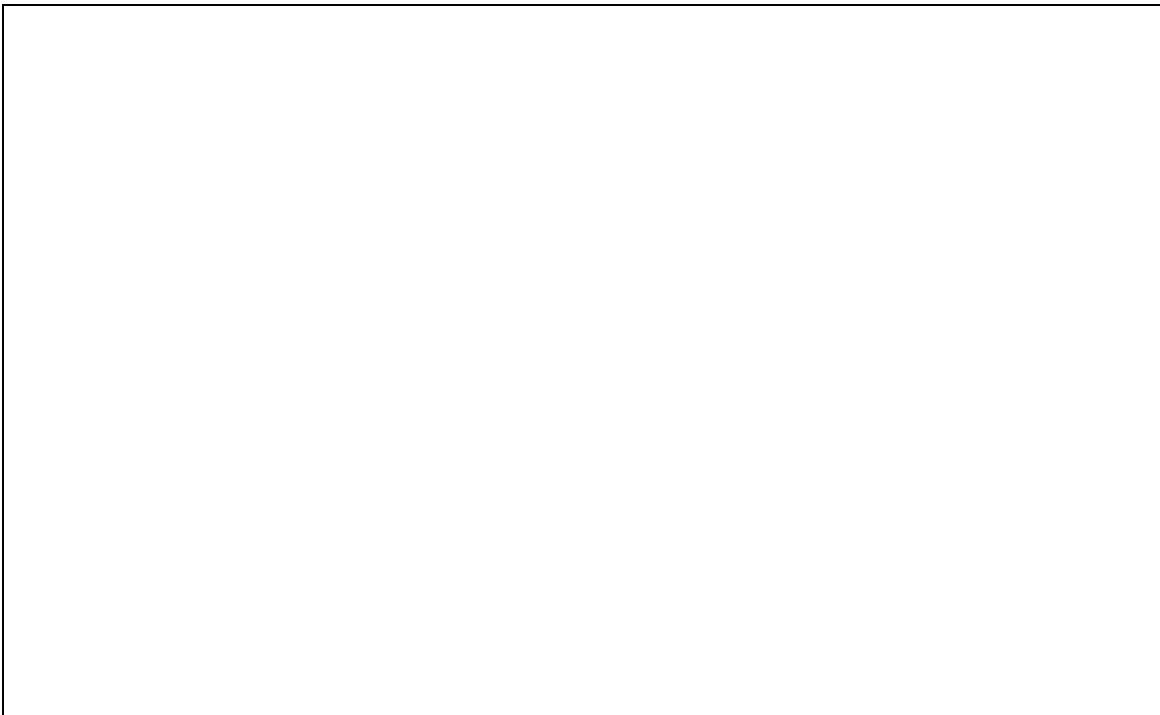


Đồ thị 5.5. Tỉ lệ mất gói các luồng khi tốc độ phát của luồng UDP là 1 Mbps.

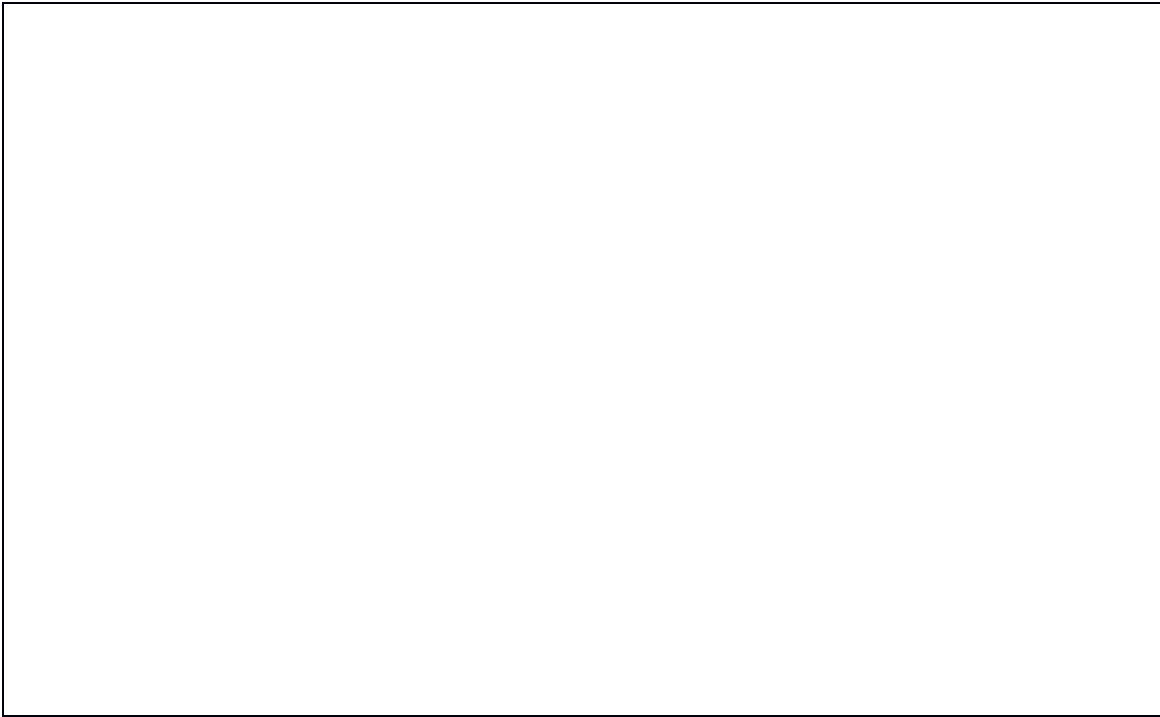


Đồ thị 5.6. Tỷ lệ mất gói các luồng khi tốc độ phát của luồng UDP là 4 Mbps.

- Đồ thị trễ đầu cuối đến đầu cuối trong các trường hợp tốc độ phát của luồng UDP là 1 Mbps và 4 Mbps

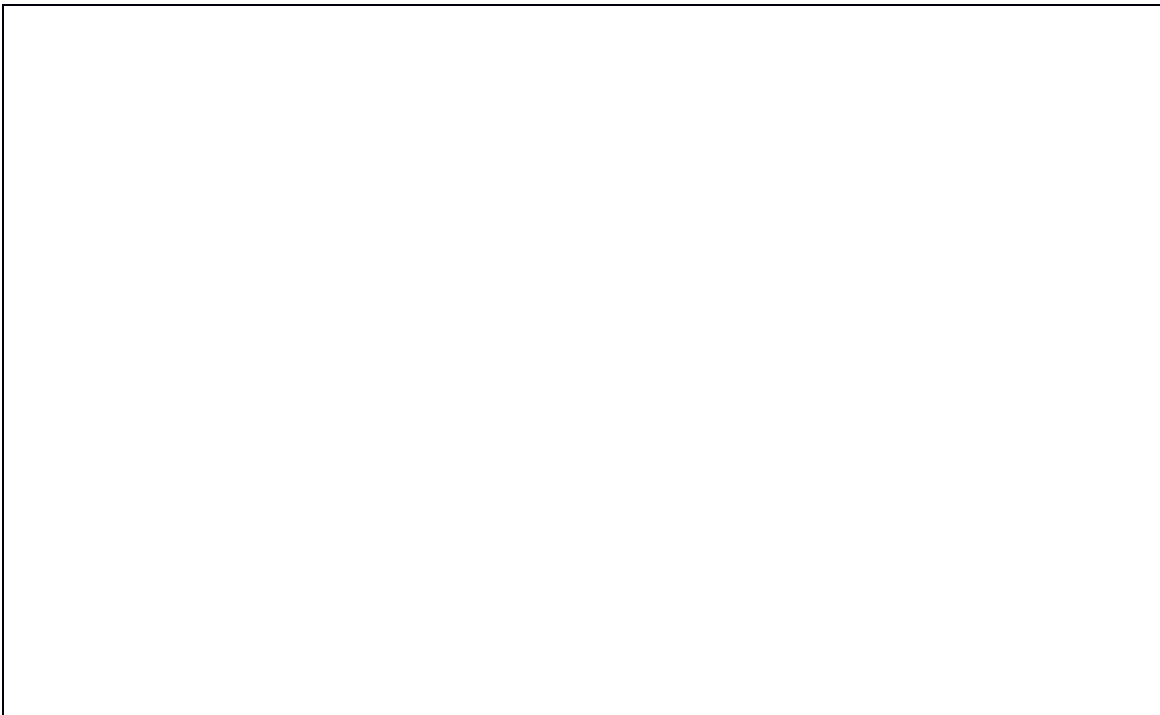


Đồ thị 5.7. Trễ đầu cuối đến đầu cuối khi tốc độ phát của luồng UDP là 1 Mbps.

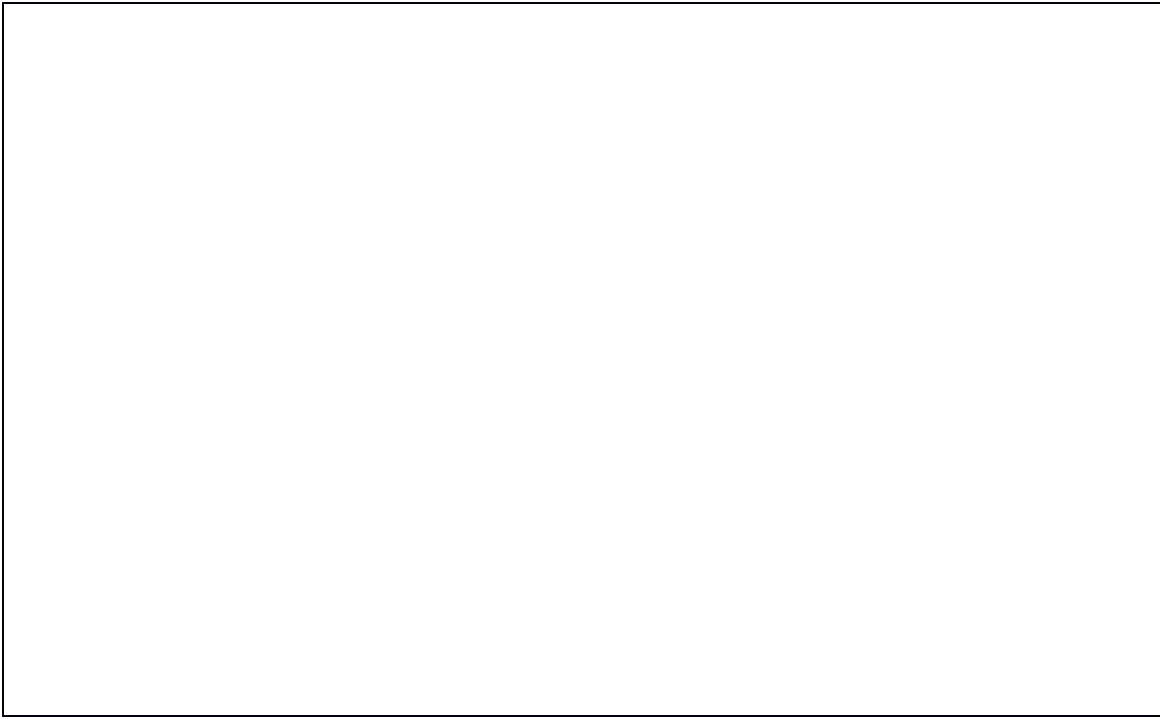


Đồ thị 5.8. Trễ đầu cuối đến đầu cuối khi tốc độ phát của luồng UDP là 4 Mbps.

- Đồ thị kích thước của sổ trượt trong các trường hợp tốc độ phát của luồng UDP là 1 Mbps và 4 Mbps.



Đồ thị 5.9. Kích thước của sổ trượt khi tốc độ phát của luồng UDP là 1 Mbps.



Đồ thị 5.10. Kích thước cửa sổ trượt khi tốc độ phát của luồng UDP là 4 Mbps.

5.1.5.2. Nhận xét và giải thích hiện tượng

Nhận xét về sự thay đổi các thông số của luồng TCP khi có và không có luồng UDP chạy nền:

.....
.....
.....
.....

Nhận xét và giải thích sự thay đổi của các thông số đã tính khi thay đổi tốc độ phát của luồng UDP:

+ Luồng UDP:

.....
.....
.....
.....

+ Luồng TCP:

.....
.....
.....
.....

+ So sánh với tính toán thực tế

.....

.....

.....

.....

5.1.6. Thay đổi kích thước cửa sổ trượt tối đa của luồng TCP

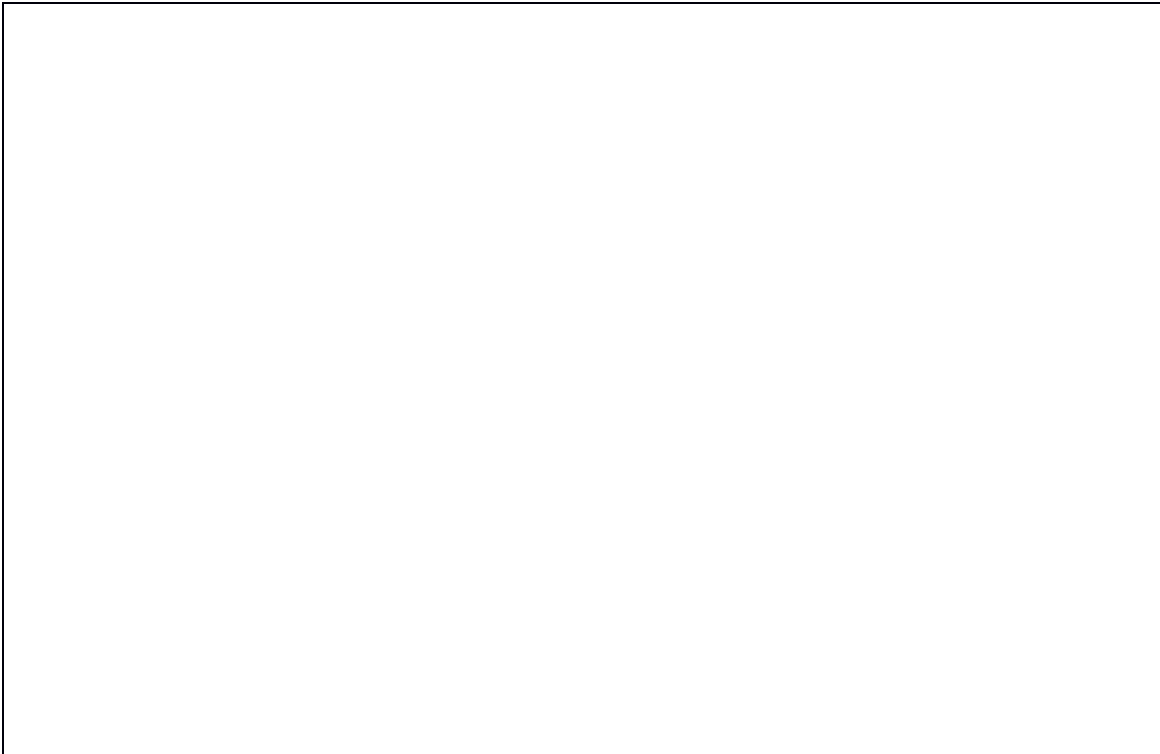
Đặt tốc độ phát của luồng UDP là 1 Mbps. Thay đổi kích thước cửa sổ trượt tối đa lần lượt là 32, 64 và 128.

5.1.6.1. Mô phỏng

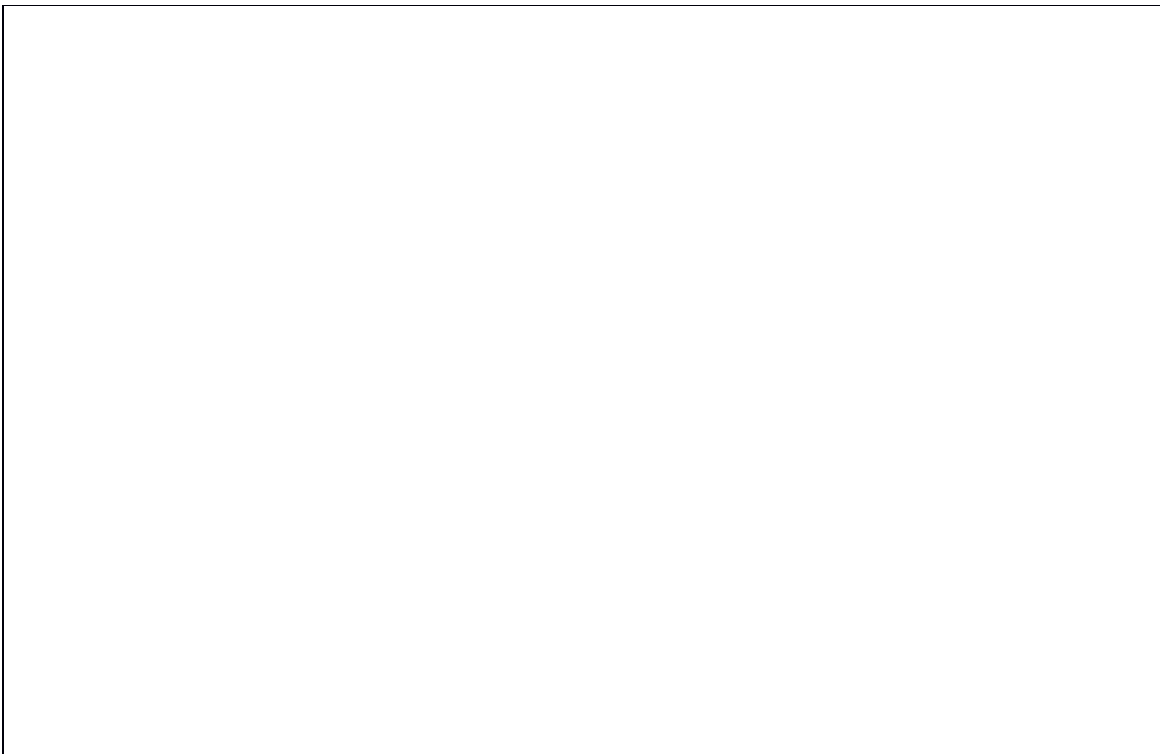
- Điền các giá trị vào bảng sau:

KÍCH THƯỚC CỬA SỔ TỐI ĐA		32	64	128
Lưu lượng (Mbps)	TCP			
	UDP			
Trễ (ms)	TCP			
	UDP			
Tỉ lệ mất gói	TCP			
	UDP			
Kích thước cửa sổ trượt trung bình	TCP			

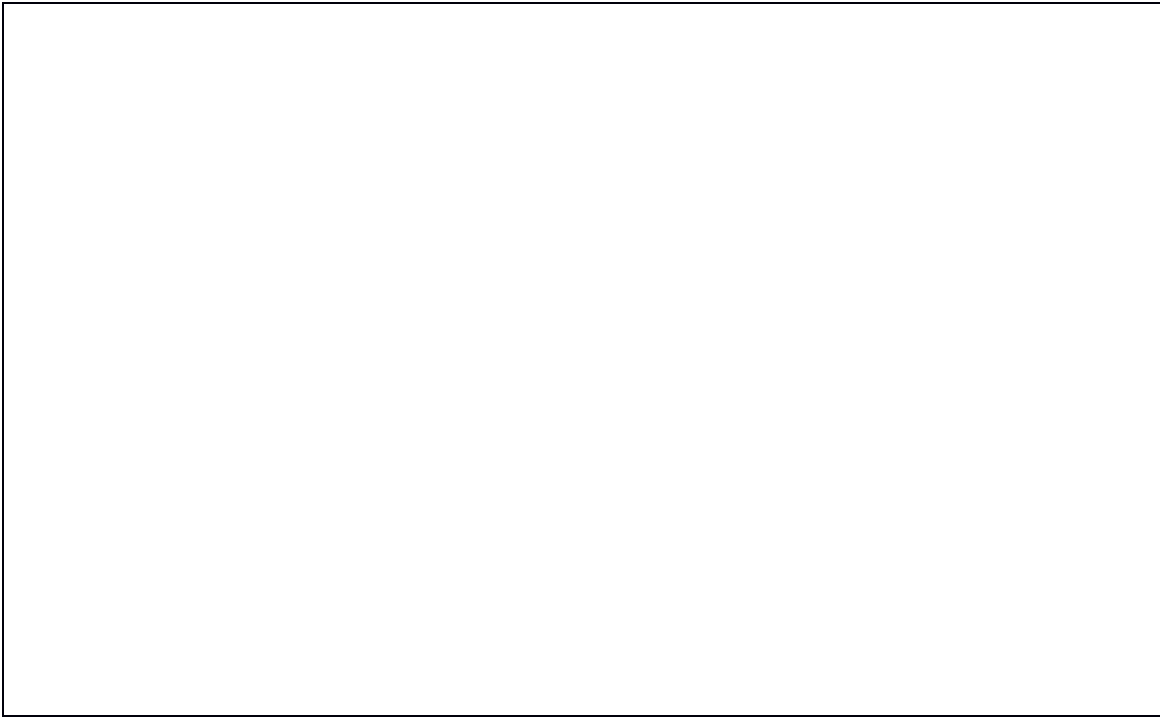
- Đồ thị lưu lượng các trường hợp:



Đồ thị 5.11. Lưu lượng các luồng khi kích thước cửa sổ trượt tối đa là 32.



Đồ thị 5.12. Lưu lượng các luồng khi kích thước cửa sổ trượt tối đa là 64.



Đồ thị 5.13. Bảng thông các luồng khi kích thước cửa sổ trượt tối đa là 128.

5.1.6.2. Nhận xét và giải thích hiện tượng

Nhận xét và giải thích sự thay đổi của bảng thông các luồng và kích thước cửa sổ trượt của luồng TCP khi thay đổi tốc độ phát của luồng UDP:

+ Luồng UDP:

.....

.....

.....

.....

+ Luồng TCP:

.....

.....

.....

.....

+ So sánh với tính toán thực tế

.....

.....

.....

.....

5.2. Mạng gồm 2 luồng TCP và 1 luồng UDP

5.2.1. Tạo file .tcl

Vào folder *2tcp1udp*, tạo file *2tcp2udp.tcl*

5.2.2. Viết chương trình

Bước 1: Mở file *2tcp2udp.tcl* vừa tạo bằng gedit hoặc trình chỉnh sửa văn bản bất kỳ.

Bước 2: Viết chương trình tương tự như đối với trường hợp trước với:

- Tạo thêm 1 luồng TCP mới y hệt luồng cũ. (Kích thước cửa sổ trượt tối đa là 128. Tốc độ phát của luồng UDP là 1 Mbps)
- Đặt tên 2 luồng TCP lần lượt là TCP1, TCP2.
- Đặt luồng TCP1 là luồng 1, TCP2 là luồng 2, UDP là luồng 3.
- Tạo thêm nút s8 và d9 là nguồn và đích của luồng TCP2. Nút s8 nối với n3 và nút d9 nối với n5.
- Chỉnh sửa hàm *cwindow* trong file .tcl để tính toán cho cả luồng TCP2 mới tạo.
- Copy các file .awk từ folder *1tcp1udp* sang. Chỉnh sửa các file .awk để tính toán cho cả luồng TCP2 mới tạo.

5.2.3. Background traffic

Chạy mô phỏng, đo và vẽ đồ thị các tham số khi có mặt và không có mặt luồng UDP với tốc độ phát 1 Mbit/s. Kích thước cửa sổ trượt tối đa của cả 2 luồng TCP là 128.

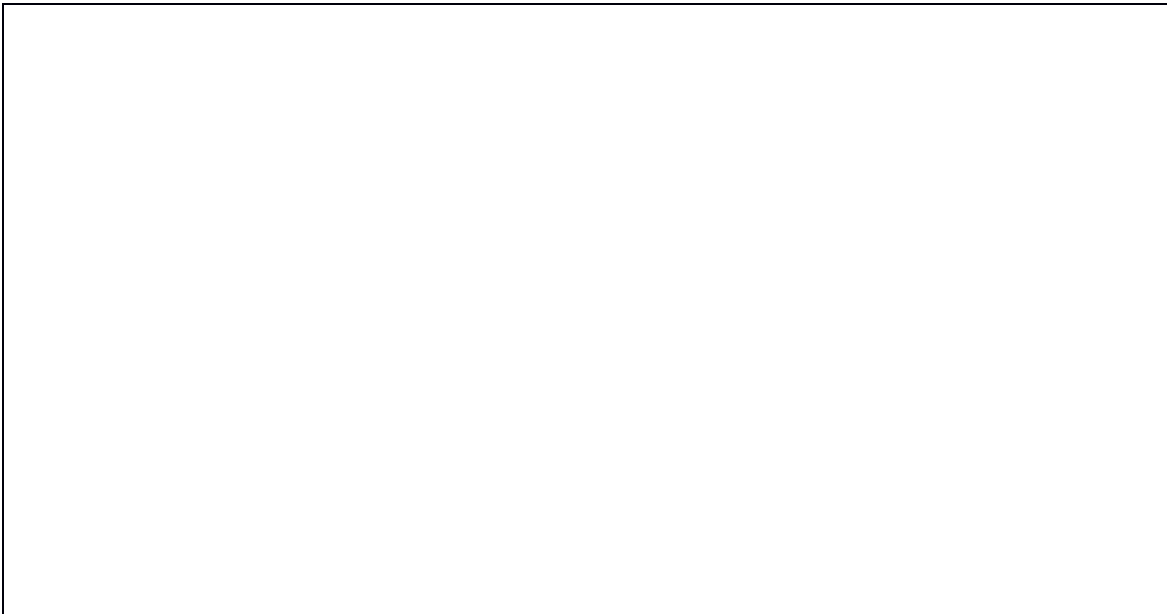
5.2.3.1. Mô phỏng

- Điền các giá trị vào bảng sau:

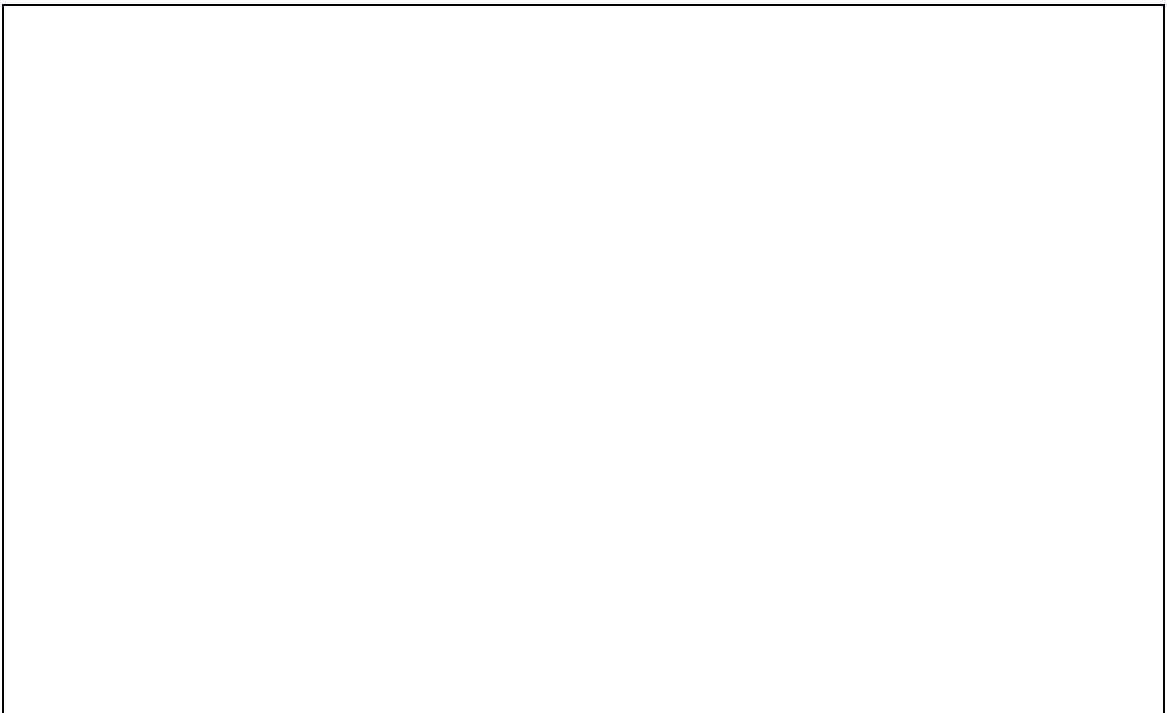
UDP rate (Mbps)		0	1
Lưu lượng (Mbps)	TCP1		
	TCP2		
Trễ (ms)	TCP1		
	TCP2		
	UDP		
Tỉ lệ mất gói	TCP1		

(%)	TCP2		
	UDP		
Kích thước cửa sổ trượt trung bình	TCP1		
	TCP2		

- Đồ thị lưu lượng các luồng trong 2 trường hợp:

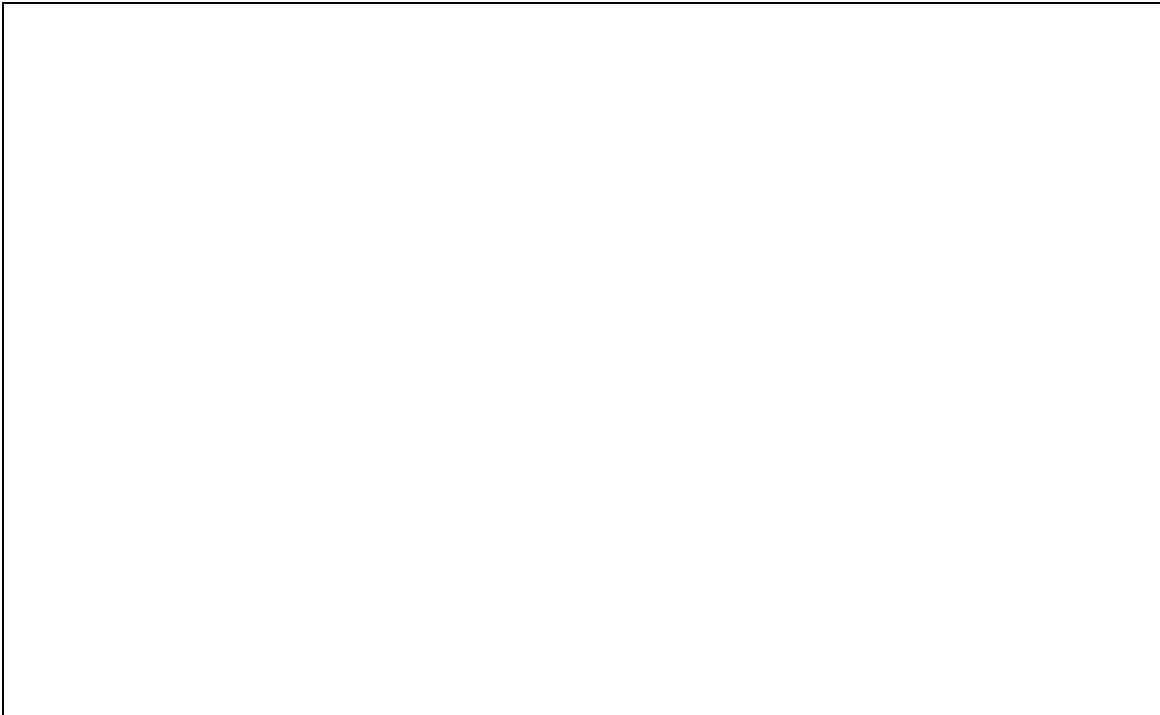


Đồ thị 5.14. Lưu lượng các luồng khi không có luồng UDP.

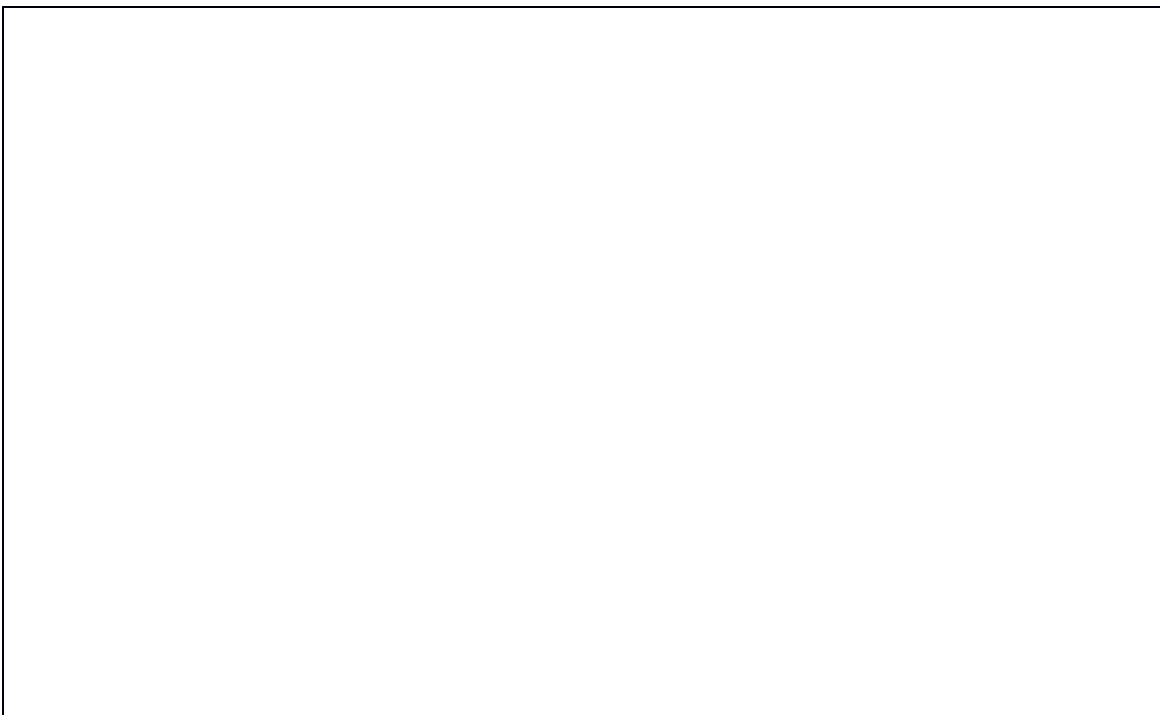


Đồ thị 5.15. Lưu lượng các luồng khi tốc độ phát của luồng UDP là 1 Mbps.

- Đồ thị tỉ lệ mất gói trong 2 trường hợp:

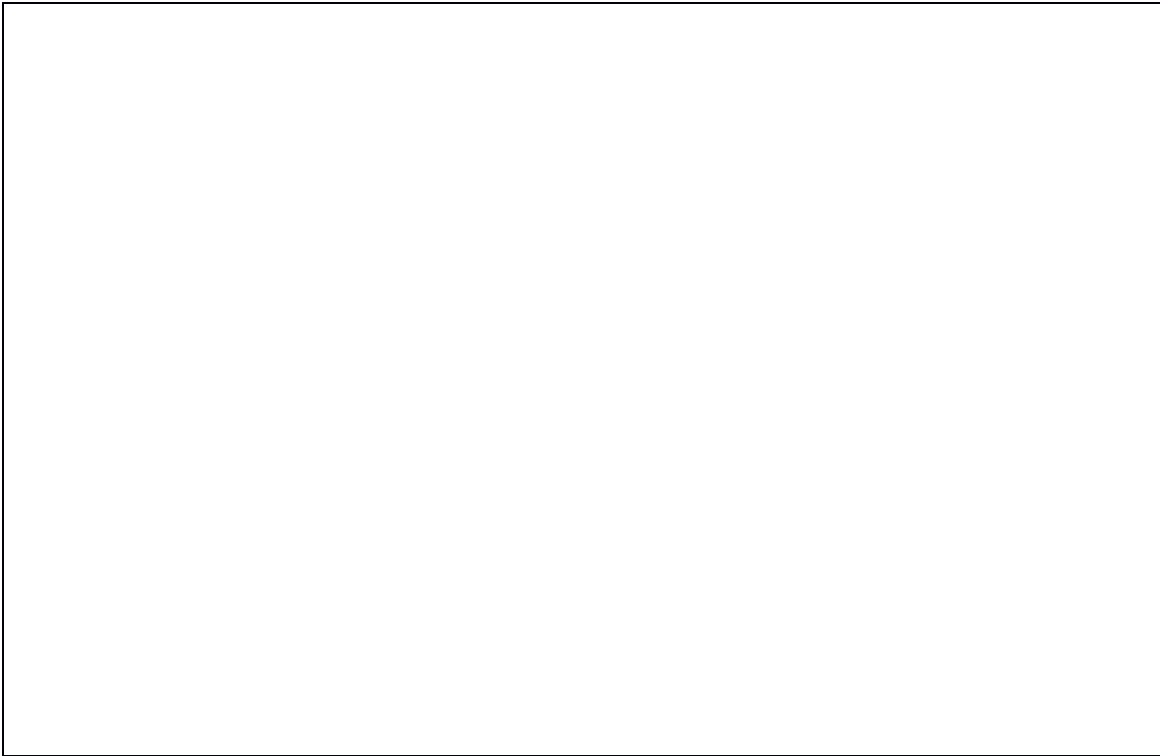


Đồ thị 5.16. Tỉ lệ mất gói các luồng khi không có luồng UDP.

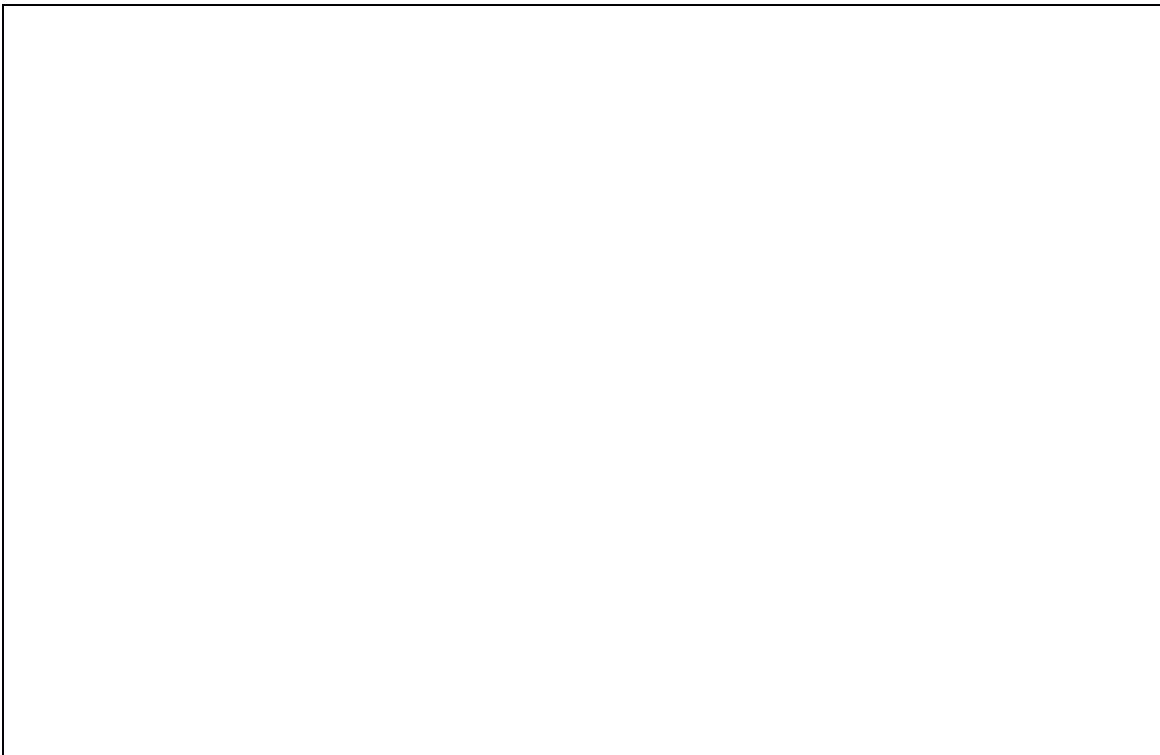


Đồ thị 5.17. Tỉ lệ mất gói các luồng khi tốc độ phát của luồng UDP là 1 Mbps.

- Đồ thị trễ đầu cuối đến đầu cuối trong 2 trường hợp:

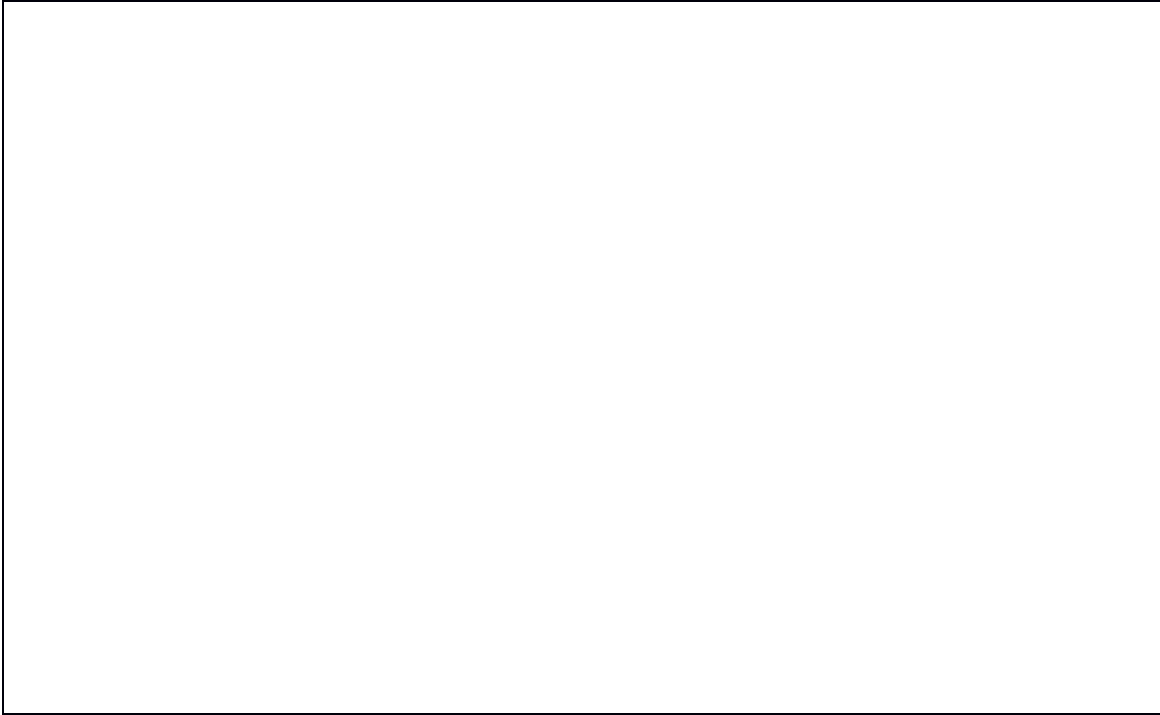


Đồ thị 5.18. Trễ đầu cuối đến đầu cuối khi không có luồng UDP.

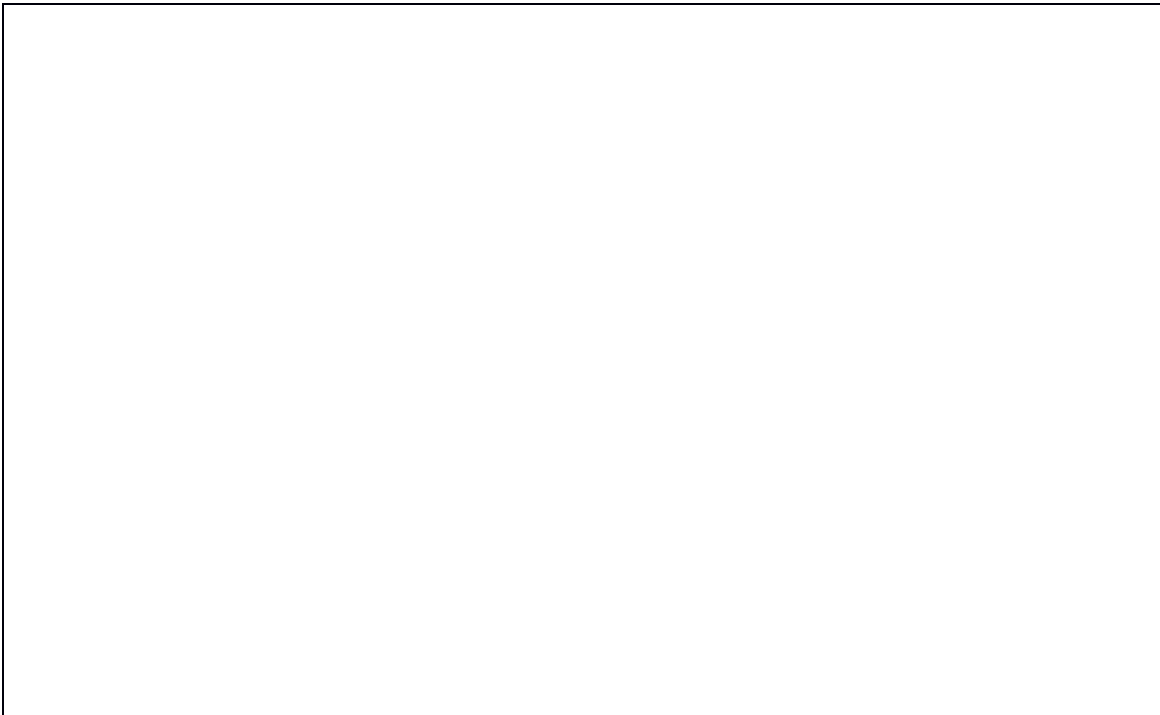


Đồ thị 5.19. Trễ đầu cuối đến đầu cuối khi tốc độ phát của luồng UDP là 1 Mbps.

- Đồ thị kích thước cửa sổ trượt trong 2 trường hợp:



Đồ thị 5.20. Kích thước cửa sổ trượt khi không có luồng UDP.



Đồ thị 5.21. Kích thước cửa sổ trượt khi tốc độ phát của luồng UDP là 1 Mbps.

5.2.3.2. Nhận xét và giải thích hiện tượng

Nhận xét sự thay đổi các tham số của mạng khi có và không có luồng UDP chạy nền:

5.2.4. Thay đổi thời gian phát

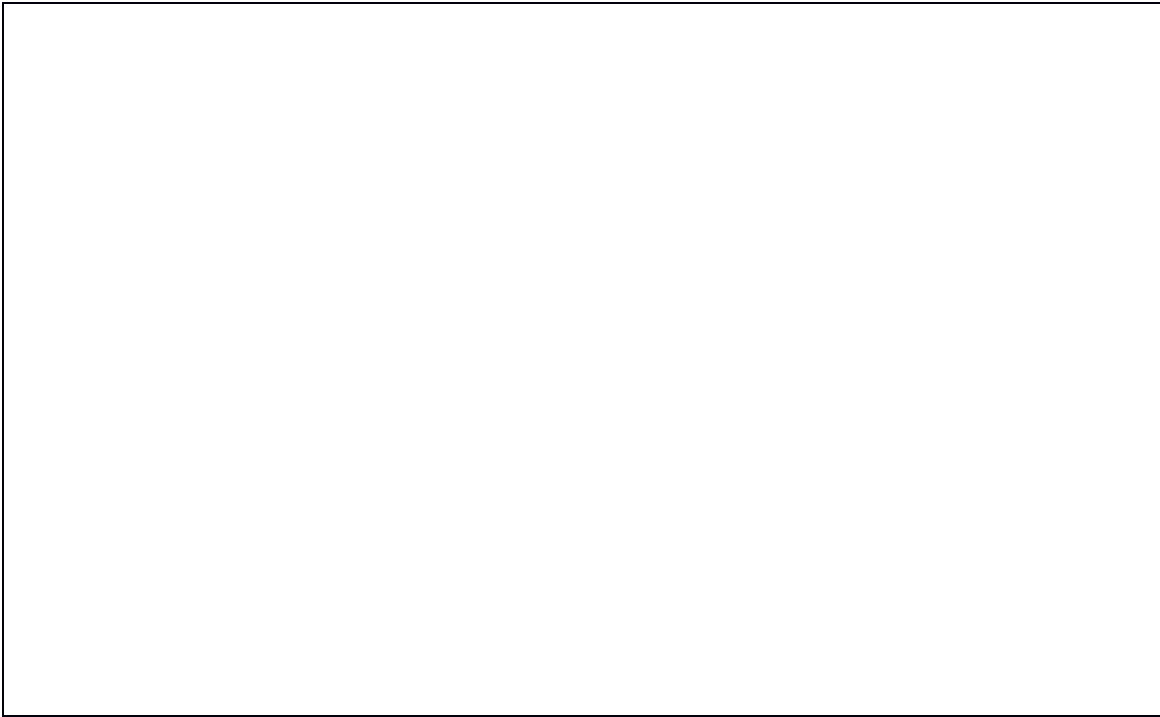
Luồng TCP2 bắt đầu phát sau luồng TCP1 50s.

5.2.4.1. Mô phỏng

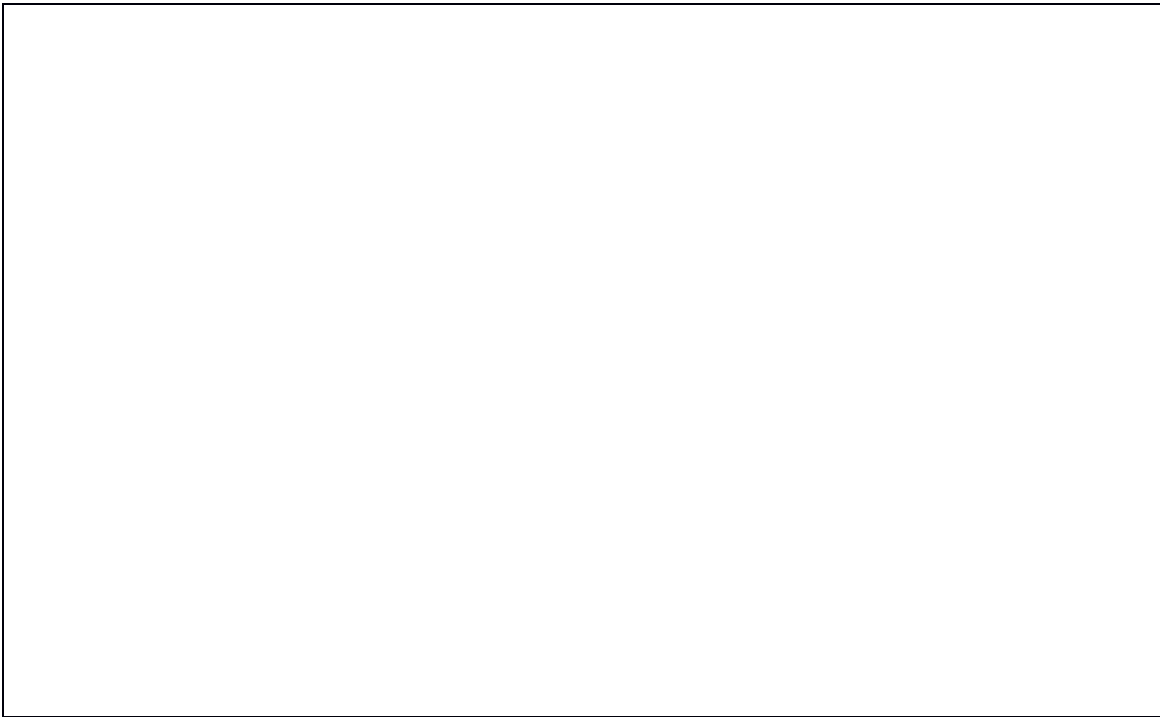
- Điền các giá trị vào bảng sau:

		Cùng lúc	Sau 50s	
			Trước giây 50	Sau giây 50
Lưu lượng (Mbps)	TCP1			
	TCP2			
	UDP			
Trễ (ms)	TCP1			
	TCP2			
	UDP			
Tỉ lệ mất gói (%)	TCP1			
	TCP2			
	UDP			
Kích thước cửa sổ trung bình	TCP1			
	TCP2			

- Đồ thị:



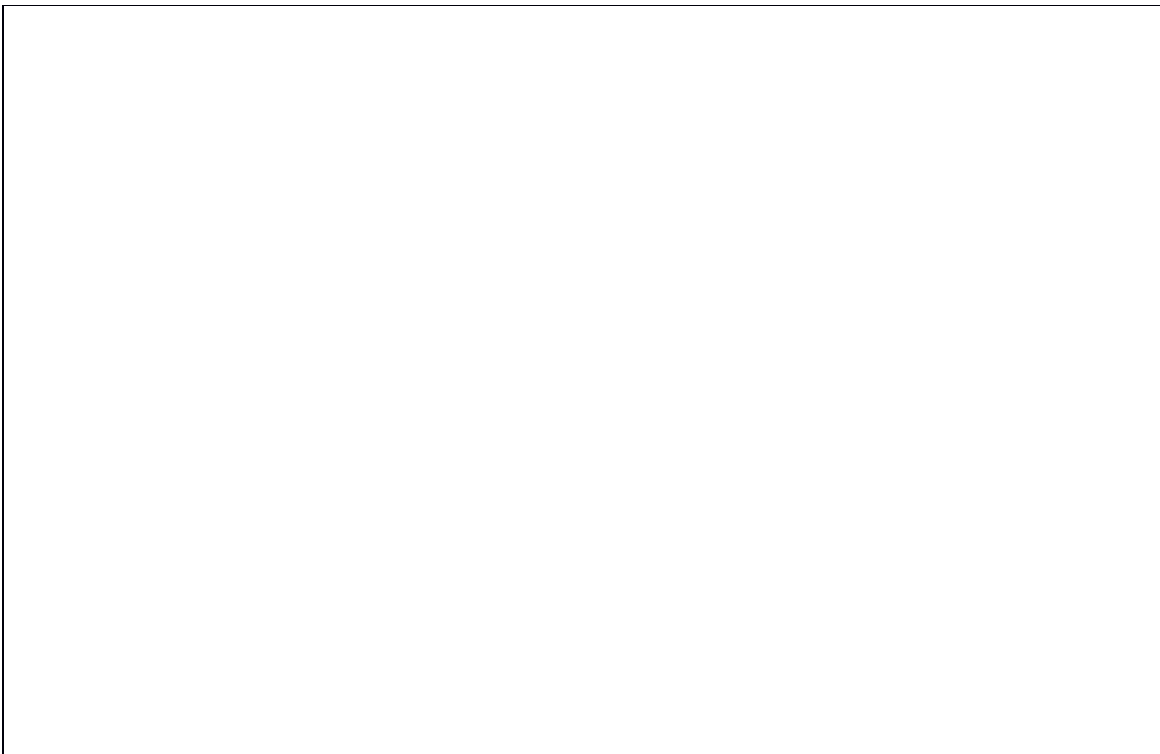
Đồ thị 5.22. Lưu lượng các luồng khi 2 luồng TCP phát lệch nhau 50s.



Đồ thị 5.23. Trễ các luồng khi 2 luồng TCP phát lệch nhau 50s.



Đồ thị 5.24. Tỷ lệ mất gói khi 2 luồng TCP phát lệch nhau 50s.



Đồ thị 5.25. Kích thước cửa sổ trượt khi 2 luồng TCP phát lệch nhau 50s.

5.2.4.2. Nhận xét và giải thích hiện tượng

Nhận xét và giải thích sự thay đổi của các tham số của mạng khi thay đổi thời gian bắt đầu phát của 2 luồng.

+ Luồng UDP:

.....

.....

.....

.....

+ 2 Luồng TCP:

.....

.....

.....

.....

+ So sánh với tính toán thực tế

.....

.....

.....

.....

VI. KẾT LUẬN

Qua bài thí nghiệm này, sinh viên đã được:

- Tìm hiểu cơ chế hoạt động của TCP, cơ chế điều khiển luồng sử dụng congestion window.
- Các thể độ hoạt động của TCP.
- Đánh giá hiệu năng của TCP.

VII. CÂU HỎI KIỂM TRA

1. Giải thích cơ chế slow start của giao thức TCP?

.....

.....

.....

.....

2. Tại sao trễ đầu cuối đến đầu cuối tính được luôn lớn hơn trễ đường truyền đã định sẵn khi xây dựng mạng trong file .tcl

.....

.....

.....

.....

3. Vì sao băng thông của luồng UDP luôn ổn định ở giá trị đặt sẵn trong mọi trường hợp?

.....

.....

.....

.....

4. Vì sao băng thông tổng cộng không bao giờ đạt được đến giá trị dung lượng của kênh truyền?

.....

.....

.....

.....