**TRƯỜNG ĐẠI HỌC KINH TẾ**

**KỸ THUẬT CÔNG NGHIỆP**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ CÔNG THƯƠNG**



**BÁO CÁO**

**ĐÁNH GIÁ HIỆU NĂNG MMT**

**ỨNG DỤNG NS ĐỂ MÔ PHỎNG MỘT MÔ HÌNH MẠNG**

**Sinh viên thực hiện:** Phạm Hoàng Hà My

**Lớp:** DHMT15A1HN

**GVHD:** Nguyễn Hoàng Chiến

HÀ NỘI - 2024

# MỤC LỤC

[**BÀI KIỂM TRA 3**](#_Toc183715276)

[**ỨNG DỤNG NS ĐỂ MÔ PHỎNG MỘT MÔ HÌNH MẠNG 3**](#_Toc183715277)

[1. Môi trường làm việc của NS. 3](#_Toc183715278)

[2. NAM (Network Animator). 3](#_Toc183715279)

[3. Mô hình. 4](#_Toc183715280)

[**Link Drive kịch bản mô phỏng: 5**](#_Toc183715281)

[**a. Khi thực hiện mô phỏng mô hình mạng sử dụng hàng đợi DropTail ta có kết quả. 5**](#_Toc183715282)

[phhmy\_droptail.tcl: 5](#_Toc183715283)

[Giải thích chi tiết phhmy\_droptail.tcl: 10](#_Toc183715284)

[Tóm tắt hoạt động 13](#_Toc183715285)

[**b. Tương tự với hàng đợi RED ta cũng có kết quả sau: 13**](#_Toc183715286)

[phhmy\_red.tcl: 14](#_Toc183715287)

[Giải thích chi tiết phhmy\_red.tcl: 18](#_Toc183715288)

[Tóm tắt hoạt động 22](#_Toc183715289)

[**c. So sánh hai mô hình 22**](#_Toc183715290)

[Thông lượng (Throughput): 24](#_Toc183715291)

[Độ trễ (Delay): 24](#_Toc183715292)

[So sánh chi tiết: 24](#_Toc183715293)

[Kết luận: 25](#_Toc183715294)

# BÀI KIỂM TRA

# ỨNG DỤNG NS ĐỂ MÔ PHỎNG MỘT MÔ HÌNH MẠNG

Để trình bày ứng dụng một mô hình mạng được thực hiện trên môi trường mô phỏng NS. Để cho biết cách thức truyền tin, điều khiển nghẽn mạch, cơ chế quản lý hàng đợi trong NS như thế nào. Ở đây mô hình chỉ sử dụng hai hàng đợi cơ bản là DropTail và RED.

Hàng đợi DropTail hoạt động theo kiểu cắt bớt phần đuôi, có nghĩa là khi hàng đợi lưu gói tin bị đầy, bộ định tuyến sẽ hủy bỏ tất cả các những gói tin gửi theo sau hay nói cách khác bộ định tuyến sẽ hủy bỏ phần đuôi của dữ liệu. Điều này có ảnh hưởng lớn đến hệ thống.

Hàng đợi RED (Random Early Discard), hàng đợi hủy bỏ sớm ngẩu nhiên, tính ngẩu nhiên của RED thể hiện khi hàng đợi bị đầy buộc nhiều kết nối TCP phải chuyển qua trạng thái khởi động chậm, bộ định tuyến sẽ hủy bỏ gói tin một cách ngẫu nhiên và từ từ theo sự gia tăng của sự nghẽn mạch. Hàng đợi RED sẽ khắc phục nhược điểm của DropTail.

## 1. Môi trường làm việc của NS.

Để mô phỏng một mô hình mạng trên NS. NS cung cấp môi trường làm việc một cách tự động, các Agent, Node, Queue, Traffic-source... đã được định nghĩa và chúng ta có thể ứng dụng. Sau khi khởi tạo các đối tượng trong môi trường NS, đoạn mã lệnh tương ứng sẽ được phát sinh. Nếu không chúng ta cũng có thể tạo ra một mô hình mạng thông qua cấu trúc câu lệnh của NS.

## 2. NAM (Network Animator).

Sau khi thực hiện mô phỏng mô hình mạng trên NS, trình biên dịch sẽ dịch chương trình chúng ta đã mô phỏng, kết quả được lưu dưới dạng file NAM. NAM có một giao diện dễ sử dụng (có các nút điều khiển chương trình như: Play, Stop, FastForward, Rewind, Pause...), chúng ta có thể thực hiện chương trình đã biên dịch dưới dạng file NAM trong môi trường NAM một cách dễ dàng. Trên màn hình luôn luôn hiển thị thời gian thực hiện chương trình, tốc độ truyền gói tin. Ngoài ra nó còn có màn hình quan sát lưu lượng gói tin truyền đi và số lượng gói tin rời khỏi đường truyền khi có độ trì hoãn cao hay nghẽn mạch xảy ra trên mạng.

## 3. Mô hình.

Mô hình được mô phỏng ở đây là dạng điểm nối điểm bao gồm 7 nút với các thông số được mô tả như sau:

A diagram of a network

Description automatically generated

### Link Drive kịch bản mô phỏng:

<https://drive.google.com/drive/folders/1RcgF8AmaStC3iUHaH9Cx326Pyi6BJmU6?usp=sharing>

### a. Khi thực hiện mô phỏng mô hình mạng sử dụng hàng đợi DropTail ta có kết quả.

A screenshot of a computer

Description automatically generated

nam: phhmy\_out1.nam(phhmy\_droptail.tcl)

#### phhmy\_droptail.tcl:

#================================

# simulation parameters setup

#================================

set val(stop) 2;

#================================

# INITIALIZATION

#================================

# Create a ns simulator

set ns [new Simulator]

# Open the NS trace file

set tracefile [open phhmy\_out1.tr w]

$ns trace-all $tracefile

# Open the NAM trace file

set namfile [open phhmy\_out1.nam w]

$ns namtrace-all $namfile

#================================

# NODES DEFINITION

#================================

# Create 7 nodes

set s0 [$ns node]

set s1 [$ns node]

set s2 [$ns node]

set r1 [$ns node]

set r2 [$ns node]

set s6 [$ns node]

set s5 [$ns node]

#================================

# LINKS DEFINITION

#================================

# Create duplex links between nodes

$ns duplex-link $s0 $r1 5Mb 5ms DropTail

$ns duplex-link $s1 $r1 5Mb 5ms DropTail

$ns duplex-link $s2 $r1 5Mb 5ms DropTail

$ns duplex-link $r1 $r2 4Mb 2ms DropTail

$ns duplex-link $r2 $s5 3Mb 6ms DropTail

$ns duplex-link $r2 $s6 3Mb 6ms DropTail

# Give node position (for NAM)

$ns duplex-link-op $s0 $r1 orient right-down

$ns duplex-link-op $s1 $r1 orient right

$ns duplex-link-op $s2 $r1 orient right-up

$ns duplex-link-op $r1 $r2 orient right

$ns duplex-link-op $r2 $s5 orient right-up

$ns duplex-link-op $r2 $s6 orient right-down

#================================

# AGENTS DEFINITION

#================================

# Set up a TCP connection

set tcp0 [new Agent/TCP]

$tcp0 set fid\_ 0

$ns attach-agent $s0 $tcp0

set sink0 [new Agent/TCPSink]

$ns attach-agent $s5 $sink0

$ns connect $tcp0 $sink0

# Set up a TCP connection

set tcp1 [new Agent/TCP]

$tcp1 set fid\_ 1

$ns attach-agent $s2 $tcp1

set sink1 [new Agent/TCPSink]

$ns attach-agent $s6 $sink1

$ns connect $tcp1 $sink1

# Set up a UDP connection

set udp0 [new Agent/UDP]

$ns attach-agent $r1 $udp0

set null [new Agent/Null]

$ns attach-agent $r2 $null

$ns connect $udp0 $null

#================================

# ADD COLORS FOR FLOWS

#================================

$ns color 0 Red ;# TCP flow from s0 to s5

$ns color 1 Blue ;# TCP flow from s2 to s6

$ns color 2 Green ;# UDP flow from r1 to r2

#================================

# APPLICATIONS DEFINITION

#================================

# Set up a FTP Application over TCP connection

set ftp0 [new Application/FTP]

$ftp0 attach-agent $tcp0

$ns at 0.1 "$ftp0 start"

$ns at 1 "$ftp0 stop"

# Set up a FTP Application over TCP connection

set ftp1 [new Application/FTP]

$ftp1 attach-agent $tcp1

$ns at 0 "$ftp1 start"

$ns at 0.9 "$ftp1 stop"

# Set up a CBR Application over UDP connection

set cbr0 [new Application/Traffic/CBR]

$cbr0 attach-agent $udp0

$cbr0 set packetSize\_ 1000

$cbr0 set rate\_ 1.5Mb

$cbr0 set random\_ 1

$ns at 0.1 "$cbr0 start"

$ns at 1 "$cbr0 stop"

#================================

# TERMINATION

#================================

# Define a 'finish' procedure

proc finish {} {

global ns tracefile namfile

$ns flush-trace

close $tracefile

close $namfile

exec nam phhmy\_out1.nam &

exit 0

}

$ns at $val(stop) "finish"

$ns run

#### Giải thích chi tiết phhmy\_droptail.tcl:

##### 1. Thiết lập thông số mô phỏng

set val(stop) 2;

* val(stop): Đặt thời điểm kết thúc mô phỏng là **2 giây**.

##### 2. Khởi tạo

set ns [new Simulator]

* Tạo một đối tượng mô phỏng mới (ns).

set tracefile [open phhmy\_out1.tr w]

$ns trace-all $tracefile

* Mở file trace (phhmy\_out1.tr) để lưu thông tin chi tiết về sự kiện mạng (ví dụ: gửi/giao nhận gói).

set namfile [open phhmy\_out1.nam w]

$ns namtrace-all $namfile

* Mở file NAM (phhmy\_out1.nam) để lưu dữ liệu mô phỏng trực quan cho công cụ đồ họa **NAM**.

##### 3. Định nghĩa các nút

set s0 [$ns node]

set s1 [$ns node]

...

* Tạo tổng cộng **7 nút mạng**: s0, s1, s2, r1, r2, s5, và s6.

##### 4. Định nghĩa các liên kết

$ns duplex-link $s0 $r1 5Mb 5ms DropTail

* Tạo liên kết song công (duplex link) giữa các nút.
* Cấu hình:
  + **Băng thông**: 5Mb.
  + **Độ trễ**: 5ms.
  + **DropTail**: Cơ chế quản lý hàng đợi gói.

$ns duplex-link-op $s0 $r1 orient right-down

* Cấu hình vị trí tương đối của các nút để hiển thị đồ họa trong NAM.

##### 5. Định nghĩa các agent

Agents mô phỏng các giao thức truyền tải (TCP/UDP).

**Thiết lập TCP**

set tcp0 [new Agent/TCP]

$tcp0 set fid\_ 0

$ns attach-agent $s0 $tcp0

* Tạo một agent TCP (tcp0) gắn vào nút s0.
* **fid\_**: Gán ID dòng dữ liệu là 0.

set sink0 [new Agent/TCPSink]

$ns attach-agent $s5 $sink0

$ns connect $tcp0 $sink0

* Tạo agent nhận TCP (sink0) gắn vào nút s5.
* Kết nối tcp0 (nút gửi) với sink0 (nút nhận).

**Thiết lập UDP**

set udp0 [new Agent/UDP]

$ns attach-agent $r1 $udp0

set null [new Agent/Null]

$ns attach-agent $r2 $null

$ns connect $udp0 $null

* Tạo agent UDP (udp0) tại r1 và một agent Null (null) tại r2.
* Null agent là nơi nhận dữ liệu mà không xử lý.

##### 6. Định nghĩa màu sắc dòng dữ liệu

$ns color 0 Red

* Gán màu đỏ (Red) cho dòng TCP giữa s0 và s5.
* Tương tự, gán màu xanh (Blue) cho dòng TCP s2 đến s6 và màu xanh lá (Green) cho dòng UDP từ r1 đến r2.

##### 7. Định nghĩa ứng dụng truyền dữ liệu

**Ứng dụng FTP (trên TCP)**

set ftp0 [new Application/FTP]

$ftp0 attach-agent $tcp0

$ns at 0.1 "$ftp0 start"

$ns at 1 "$ftp0 stop"

* Tạo ứng dụng FTP trên TCP agent (tcp0) để truyền dữ liệu.
* FTP bắt đầu từ 0.1 giây và dừng tại 1 giây.

**Ứng dụng CBR (trên UDP)**

set cbr0 [new Application/Traffic/CBR]

$cbr0 attach-agent $udp0

$cbr0 set packetSize\_ 1000

$cbr0 set rate\_ 1.5Mb

* Tạo ứng dụng CBR (Constant Bit Rate) để gửi dữ liệu qua UDP.
* **Kích thước gói**: 1000 bytes.
* **Tốc độ**: 1.5Mb/s.
* Ứng dụng chạy từ 0.1 giây và dừng tại 1 giây.

##### 8. Kết thúc mô phỏng

proc finish {} {

global ns tracefile namfile

$ns flush-trace

close $tracefile

close $namfile

exec nam phhmy\_out1.nam &

exit 0

}

$ns at $val(stop) "finish"

* **finish**: Thủ tục kết thúc mô phỏng, đóng các file trace và NAM.
* Mở công cụ NAM để hiển thị đồ họa.

##### 9. Chạy mô phỏng

$ns run

* Khởi động quá trình mô phỏng.

#### Tóm tắt hoạt động

* Mô phỏng có:
  + 3 dòng TCP (s0->s5, s2->s6) và 1 dòng UDP (r1->r2).
  + Ứng dụng FTP và CBR hoạt động trong các dòng này.
* Mọi sự kiện được ghi lại trong file phhmy\_out1.tr và trực quan hóa bằng phhmy\_out1.nam.
* Mô phỏng kết thúc sau **2 giây**.

### b. Tương tự với hàng đợi RED ta cũng có kết quả sau:

A screenshot of a computer

Description automatically generated

nam: phhmy\_out2.nam(phhmy\_red.tcl)

#### phhmy\_red.tcl:

#================================

# simulation parameters setup

#================================

set val(stop) 2;

#================================

# INITIALIZATION

#================================

# Create a ns simulator

set ns [new Simulator]

# Open the NS trace file

set tracefile [open phhmy\_out2.tr w]

$ns trace-all $tracefile

# Open the NAM trace file

set namfile [open phhmy\_out2.nam w]

$ns namtrace-all $namfile

#================================

# NODES DEFINITION

#================================

# Create 7 nodes

set s0 [$ns node]

set s1 [$ns node]

set s2 [$ns node]

set r1 [$ns node]

set r2 [$ns node]

set s6 [$ns node]

set s5 [$ns node]

#================================

# LINKS DEFINITION

#================================

# Create duplex links between nodes using RED queue management

$ns duplex-link $s0 $r1 5Mb 5ms RED

$ns duplex-link $s1 $r1 5Mb 5ms RED

$ns duplex-link $s2 $r1 5Mb 5ms RED

$ns duplex-link $r1 $r2 4Mb 2ms RED

$ns duplex-link $r2 $s5 3Mb 6ms RED

$ns duplex-link $r2 $s6 3Mb 6ms RED

# Configure RED parameters (example: min threshold, max threshold, max probability)

$ns queue-limit $s0 $r1 50

$ns queue-limit $r1 $r2 30

$ns queue-limit $r2 $s5 20

# Give node position (for NAM)

$ns duplex-link-op $s0 $r1 orient right-down

$ns duplex-link-op $s1 $r1 orient right

$ns duplex-link-op $s2 $r1 orient right-up

$ns duplex-link-op $r1 $r2 orient right

$ns duplex-link-op $r2 $s5 orient right-up

$ns duplex-link-op $r2 $s6 orient right-down

#================================

# AGENTS DEFINITION

#================================

# Set up a TCP connection

set tcp0 [new Agent/TCP]

$tcp0 set fid\_ 0

$ns attach-agent $s0 $tcp0

set sink0 [new Agent/TCPSink]

$ns attach-agent $s5 $sink0

$ns connect $tcp0 $sink0

# Set up a TCP connection

set tcp1 [new Agent/TCP]

$tcp1 set fid\_ 1

$ns attach-agent $s2 $tcp1

set sink1 [new Agent/TCPSink]

$ns attach-agent $s6 $sink1

$ns connect $tcp1 $sink1

# Set up a UDP connection

set udp0 [new Agent/UDP]

$ns attach-agent $r1 $udp0

set null [new Agent/Null]

$ns attach-agent $r2 $null

$ns connect $udp0 $null

#================================

# ADD COLORS FOR FLOWS

#================================

$ns color 0 Red ;# TCP flow from s0 to s5

$ns color 1 Blue ;# TCP flow from s2 to s6

$ns color 2 Green ;# UDP flow from r1 to r2

#================================

# APPLICATIONS DEFINITION

#================================

# Set up a FTP Application over TCP connection

set ftp0 [new Application/FTP]

$ftp0 attach-agent $tcp0

$ns at 0.1 "$ftp0 start"

$ns at 1 "$ftp0 stop"

# Set up a FTP Application over TCP connection

set ftp1 [new Application/FTP]

$ftp1 attach-agent $tcp1

$ns at 0 "$ftp1 start"

$ns at 0.9 "$ftp1 stop"

# Set up a CBR Application over UDP connection

set cbr0 [new Application/Traffic/CBR]

$cbr0 attach-agent $udp0

$cbr0 set packetSize\_ 1000

$cbr0 set rate\_ 1.5Mb

$cbr0 set random\_ 1

$ns at 0.1 "$cbr0 start"

$ns at 1 "$cbr0 stop"

#================================

# TERMINATION

#================================

# Define a 'finish' procedure

proc finish {} {

global ns tracefile namfile

$ns flush-trace

close $tracefile

close $namfile

exec nam phhmy\_out2.nam &

exit 0

}

$ns at $val(stop) "finish"

$ns run

#### Giải thích chi tiết phhmy\_red.tcl:

***phhmy\_red.tcl*** mô phỏng một mạng sử dụng **NS2**, với điểm nổi bật là sử dụng cơ chế quản lý hàng đợi **RED (Random Early Detection)** cho các liên kết. Đây là một cơ chế giảm tắc nghẽn chủ động, khác với DropTail. Mô phỏng bao gồm các nút, liên kết, các giao thức truyền tải (TCP, UDP), và các ứng dụng truyền dữ liệu. Sau đây là giải thích chi tiết từng phần:

##### 1. Thiết lập thông số mô phỏng

set val(stop) 2;

* Đặt thời gian kết thúc mô phỏng là **2 giây**.

##### 2. Khởi tạo

set ns [new Simulator]

* Tạo một đối tượng mô phỏng (ns).

set tracefile [open phhmy\_out2.tr w]

$ns trace-all $tracefile

* Mở file trace (phhmy\_out2.tr) để lưu các sự kiện mạng (gửi/nhận gói, mất gói...).

set namfile [open phhmy\_out2.nam w]

$ns namtrace-all $namfile

* Mở file NAM (phhmy\_out2.nam) để lưu dữ liệu phục vụ hiển thị mô phỏng bằng công cụ NAM.

##### 3. Định nghĩa các nút

set s0 [$ns node]

set s1 [$ns node]

...

* Tạo tổng cộng **7 nút mạng**: s0, s1, s2, r1, r2, s6, và s5.

##### 4. Định nghĩa các liên kết

**Liên kết sử dụng RED**

$ns duplex-link $s0 $r1 5Mb 5ms RED

* Tạo liên kết song công giữa s0 và r1.
* **Thông số**:
  + **Băng thông**: 5Mb.
  + **Độ trễ**: 5ms.
  + **RED**: Cơ chế hàng đợi chủ động.

**Cấu hình RED**

$ns queue-limit $s0 $r1 50

* Đặt giới hạn hàng đợi của liên kết s0 đến r1 là **50 gói**.
* Tương tự, đặt giới hạn cho các liên kết khác (r1 -> r2: 30 gói, r2 -> s5: 20 gói).

**Định vị nút (cho NAM)**

$ns duplex-link-op $s0 $r1 orient right-down

* Cấu hình vị trí các nút khi hiển thị trong NAM.

##### 5. Định nghĩa các agent

Agents mô phỏng các giao thức truyền tải (TCP, UDP).

**TCP**

set tcp0 [new Agent/TCP]

$tcp0 set fid\_ 0

$ns attach-agent $s0 $tcp0

* Tạo agent TCP (tcp0) gắn vào s0.
* **fid\_**: Đặt ID dòng dữ liệu là 0.

set sink0 [new Agent/TCPSink]

$ns attach-agent $s5 $sink0

$ns connect $tcp0 $sink0

* Tạo agent nhận TCP (sink0) tại s5.
* Kết nối tcp0 (nút gửi) với sink0 (nút nhận).

**UDP**

set udp0 [new Agent/UDP]

$ns attach-agent $r1 $udp0

set null [new Agent/Null]

$ns attach-agent $r2 $null

$ns connect $udp0 $null

* Tạo agent UDP (udp0) tại r1 và một agent Null (null) tại r2.

##### 6. Định nghĩa màu sắc dòng dữ liệu

$ns color 0 Red

* Gán màu đỏ cho dòng TCP từ s0 đến s5.
* Tương tự, màu xanh (Blue) cho TCP từ s2 đến s6, và màu xanh lá cây (Green) cho UDP từ r1 đến r2.

##### 7. Định nghĩa các ứng dụng

**FTP (trên TCP)**

set ftp0 [new Application/FTP]

$ftp0 attach-agent $tcp0

$ns at 0.1 "$ftp0 start"

$ns at 1 "$ftp0 stop"

* Tạo ứng dụng FTP trên TCP agent (tcp0).
* Ứng dụng chạy từ 0.1 giây đến 1 giây.

**CBR (trên UDP)**

set cbr0 [new Application/Traffic/CBR]

$cbr0 attach-agent $udp0

$cbr0 set packetSize\_ 1000

$cbr0 set rate\_ 1.5Mb

$cbr0 set random\_ 1

$ns at 0.1 "$cbr0 start"

$ns at 1 "$cbr0 stop"

* Tạo ứng dụng CBR (Constant Bit Rate) để gửi dữ liệu qua UDP:
  + **Kích thước gói**: 1000 bytes.
  + **Tốc độ**: 1.5Mb/s.
  + **Ngẫu nhiên hóa**: 1 (random interval).

##### 8. Kết thúc mô phỏng

proc finish {} {

global ns tracefile namfile

$ns flush-trace

close $tracefile

close $namfile

exec nam phhmy\_out2.nam &

exit 0

}

$ns at $val(stop) "finish"

* **finish**: Thủ tục kết thúc mô phỏng:
* Ghi dữ liệu vào file.
* Đóng các file trace và NAM.
* Chạy công cụ NAM (exec nam phhmy\_out2.nam &).

##### 9. Chạy mô phỏng

$ns run

* Bắt đầu quá trình mô phỏng.

#### Tóm tắt hoạt động

**Liên kết**:

Mô phỏng có các liên kết sử dụng **RED** để quản lý hàng đợi, nhằm giảm thiểu tắc nghẽn chủ động.

**Dòng dữ liệu**:

* + **2 dòng TCP**:
    - s0 -> s5 (dòng 0, Red).
    - s2 -> s6 (dòng 1, Blue).
  + **1 dòng UDP**:
    - r1 -> r2 (dòng Green).

**Ứng dụng**:

* + FTP chạy trên TCP.
  + CBR chạy trên UDP.

**Kết quả**:

* + Kết thúc sau **2 giây**.
  + File phhmy\_out2.nam có thể xem bằng **NAM** để trực quan hóa.

### c. So sánh hai mô hình

So sánh hai mô hình sử dụng hàng đợi Droptail và RED với các thông số đều giống nhau như độ trễ, băng thông, sử dụng cùng giao thức và cùng thời gian truyền tin.

A screenshot of a computer

Description automatically generated

Thông lượng, độ trễ của hàng đợi droptail

A screenshot of a computer

Description automatically generated

Thông lượng, độ trễ của hàng đợi red (dưới droptail)

#### Thông lượng (Throughput):

* + Droptail:
    - FTP0: ~1.587 Mbps
    - FTP2: ~1.067 Mbps
    - UDP: ~1.449 Mbps
  + RED:
    - FTP0: ~1.607 Mbps
    - FTP2: ~0.485 Mbps
    - UDP: ~1.458 Mbps

#### Độ trễ (Delay):

* + Droptail:
    - FTP0: ~0.078 s
    - FTP2: ~0.097 s
    - UDP: ~0.004 s
  + RED:
    - FTP0: ~0.033 s
    - FTP2: ~0.032 s
    - UDP: ~0.004 s

#### So sánh chi tiết:

* **Thông lượng (Throughput):**
  + **Droptail** có thông lượng ổn định và cao hơn ở luồng FTP (FTP0, FTP2) so với RED.
  + **RED** phân phối băng thông không đồng đều hơn, với thông lượng FTP2 giảm rõ rệt (~0.485 Mbps), nhưng thông lượng UDP được ưu tiên và duy trì tốt hơn một chút (~1.458 Mbps).
* **Độ trễ (Delay):**
  + **RED** cải thiện độ trễ rõ rệt trên tất cả các luồng FTP (FTP0, FTP2), đặc biệt là luồng FTP0, từ 0.078 s (Droptail) xuống 0.033 s.
  + Độ trễ UDP không có sự khác biệt đáng kể giữa hai thuật toán (~0.004 s).

#### Kết luận:

* **Droptail:** Phù hợp nếu cần tối đa hóa thông lượng cho các luồng TCP (như FTP), nhưng có thể gây ra vấn đề khi mạng bị tắc nghẽn (bufferbloat), dẫn đến độ trễ cao.
* **RED:** Thích hợp hơn trong môi trường mạng có lưu lượng hỗn hợp (TCP và UDP), vì nó giảm độ trễ và kiểm soát tắc nghẽn hiệu quả hơn, nhưng có thể làm giảm thông lượng cho một số luồng.