

**ĐẠI HỌC ĐÀ NẴNG**  
**TRƯỜNG ĐẠI HỌC BÁCH KHOA**  
**KHOA ĐIỆN TỬ - VIỄN THÔNG**



**BÁO CÁO**  
**LẬP TRÌNH MẠNG**

**ĐỀ TÀI:**

**ẢNH HƯỞNG CỦA KÍCH THƯỚC GÓI TIN ĐẾN  
HIỆU SUẤT CSMA TRONG MẠNG ETHERNET**

**Sinh viên thực hiện: Văn Tiến Lộc (106220257)**  
**Nguyễn Trung Thành (106220270)**  
**Lớp học phần: 22.44**  
**GVHD: TS. Nguyễn Văn Hiếu**



**TP. Đà Nẵng, 2025**

## BẢNG PHÂN CÔNG CÔNG VIỆC TRONG NHÓM

Họ và tên sinh viên	Mã số sinh viên	Lớp sinh hoạt	Phân công nhiệm vụ	Đóng góp
Văn Tiến Lộc	106220220	22KTMT2	Viết code, chạy mô phỏng, làm báo cáo, slide	50%
Nguyễn Trung Thành	106220226		Viết code, chạy mô phỏng, thu thập dữ liệu	50%
Tổng				100%

# MỤC LỤC

1. Giới thiệu.....	1
1.1. Tổng quan về CSMA.....	1
1.2. Mục tiêu nghiên cứu:.....	1
2. Phương pháp thực hiện.....	2
2.1 Mô hình hệ thống/mạng.....	2
2.2 Kịch bản mô phỏng.....	3
3. Công cụ sử dụng.....	3
3.1. NS-3(Network Simulator 3).....	3
3.2 Python.....	4
4. Kết quả.....	4
4.1. Kết quả mô phỏng từ NS-3.....	4
a. Trường hợp mạng lý tưởng (DataRate=100 Mbps, Delay=1 $\mu$ s).....	4
b. Trường hợp mạng nghẽn (DataRate=20 Mbps, Delay=20 $\mu$ s).....	6
4.2. Phân tích đồ thị thông lượng TCP và UDP theo kích thước gói.....	9
4.4. Phân tích đồ thị gói nhận của TCP và UDP theo kích thước gói.....	11
4.5 Phân tích đồ thị Packet Delivery Ratio (PDR) theo kích thước gói:.....	12
4.6. Đánh giá chung:.....	13
a. TCP.....	13
b. UDP.....	13
5. Kết luận.....	13

## 1. Giới thiệu

### 1.1. Tổng quan về CSMA

- Khái niệm: CSMA (Carrier Sense Multiple Access) là một giao thức truy cập môi trường chia sẻ, thường được sử dụng trong các mạng Ethernet để điều khiển việc truyền dữ liệu giữa nhiều nút trên cùng một kênh. Nguyên tắc cơ bản của CSMA là “nghe trước khi gửi”: trước khi truyền gói tin, một nút sẽ kiểm tra xem kênh có đang bận hay không.
- Nguyên lý hoạt động
  1. Kiểm tra kênh (Carrier Sense): Mỗi nút kiểm tra kênh trước khi truyền. Nếu kênh rảnh, nút bắt đầu gửi dữ liệu.
  2. Truy cập nhiều nút (Multiple Access): Nhiều nút có thể truy cập cùng một kênh. Khi nhiều nút truyền cùng lúc, xảy ra va chạm.
  3. Xử lý va chạm (Collision Handling): Khi va chạm xảy ra, các nút dừng truyền và thực hiện backoff ngẫu nhiên trước khi thử lại.
- Ý nghĩa:
  - Tối ưu hóa truy cập kênh: CSMA giúp nhiều nút chia sẻ kênh một cách hiệu quả, giảm thời gian kênh bị lãng phí.
  - Giảm va chạm: Cơ chế kiểm tra kênh trước khi truyền và backoff ngẫu nhiên hạn chế xung đột giữa các nút.
  - Cơ sở cho mạng Ethernet hiện đại: CSMA và các biến thể của nó là nền tảng cho CSMA/CD (Ethernet có dây) và CSMA/CA (Wi-Fi), giúp duy trì hiệu suất mạng ổn định trong nhiều tình huống tải khác nhau.
  - Ứng dụng nghiên cứu: CSMA được dùng trong mô phỏng mạng để phân tích hiệu suất, ảnh hưởng của kích thước gói tin, số nút và tải mạng lên throughput, delay và PDR(Packet Delivery Ratio).

### 1.2. Mục tiêu nghiên cứu:

Các mục tiêu nghiên cứu chính của đề tài được thực hiện trên NS3 bao gồm:

- Đánh giá ảnh hưởng của kích thước gói tin lên hiệu suất mạng CSMA, bao gồm TCP và UDP.
- Phân tích các chỉ số Throughput, Packet Delivery Ratio (PDR), số lượng gói tin truyền và nhận.
- So sánh hiệu suất giữa hai giao thức truyền tải phổ biến là TCP và UDP trong mạng Ethernet 5 nút.
- Xác định xu hướng thay đổi của hiệu suất mạng khi kích thước gói tin tăng, từ gói nhỏ đến gói lớn.

## 2. Phương pháp thực hiện

### 2.1 Mô hình hệ thống/mạng

- Mạng mô phỏng CSMA:
  - Mạng được mô phỏng gồm 5 nút (Node 0 – Node 4) kết nối qua kênh CSMA (Carrier Sense Multiple Access), một giao thức truy cập môi trường chia sẻ phổ biến trong Ethernet. Kênh CSMA được thiết lập với DataRate 100 Mbps và Delay 6560 ns, mô phỏng băng thông và độ trễ thực tế của môi trường mạng có dây.
  - Trong mạng CSMA, khi nhiều nút cùng truyền đồng thời, va chạm (collision) có thể xảy ra, và các nút sử dụng cơ chế CSMA/CD (Collision Detection) để phát hiện và xử lý va chạm, dừng truyền, gửi tín hiệu “jam” và chờ ngẫu nhiên trước khi thử lại.
  - Mạng mô phỏng trong code hoạt động như một mạng Ethernet thu nhỏ, nơi tất cả các nút chia sẻ cùng một kênh truyền dẫn. Việc cài đặt CSMA giúp mô phỏng các tình huống thực tế của mạng có tải vừa phải, phản ánh chính xác cách mà các nút cạnh tranh băng thông, xảy ra va chạm và xử lý lỗi truyền dữ liệu.
- Các khối/nút mạng và nhiệm vụ:
  - Node 0 (Nguồn dữ liệu): Node này chạy ứng dụng TCP BulkSend và UDP OnOff để truyền gói tin đến Node 4. Nó chịu trách nhiệm tạo tải mạng và thay đổi kích thước gói tin theo kịch bản, từ đó đánh giá ảnh hưởng của packet size đến hiệu suất mạng.
  - Node 1 – Node 3 (Nút trung gian): Các nút này không chạy ứng dụng truyền dữ liệu riêng nhưng đóng vai trò chuyển tiếp gói tin qua kênh CSMA. Chúng đảm bảo môi trường mạng nhiều nút cạnh tranh kênh, giúp mô phỏng chính xác hiện tượng va chạm và cơ chế backoff trong CSMA/CD.
  - Node 4 (Đích nhận dữ liệu): Node này cài đặt TCP PacketSink và UDP PacketSink để nhận dữ liệu từ Node 0. Nó thu thập các số liệu về số gói tin truyền, nhận và tổng số byte nhận được, phục vụ phân tích throughput và Packet Delivery Ratio (PDR).
- Mạng mô phỏng này cung cấp một môi trường kiểm soát để đánh giá cách mà CSMA/CD vận hành thực tế, bao gồm kiểm tra kênh, va chạm, backoff, và phân phối gói tin. Nó cho phép phân tích ảnh hưởng của các thông số ảnh đến hiệu suất mạng, từ đó rút ra các kết luận về tối ưu hóa trong mạng Ethernet.

## 2.2 Kịch bản mô phỏng

Trong dự án này, kịch bản mô phỏng được thiết kế để đánh giá ảnh hưởng của kích thước gói tin đến hiệu suất mạng CSMA trong Ethernet

- Mục tiêu: Đánh giá hiệu suất TCP và UDP trong mạng CSMA Ethernet theo các kích thước gói khác nhau và trạng thái kênh (lý tưởng vs nghẽn).
- Số node: 5 node Ethernet.
- Thời gian mô phỏng: 10 giây.
- Các kịch bản:
  - Lý tưởng: Delay thấp 1  $\mu$ s, tốc độ kênh 100 Mbps.
  - Nghẽn: Delay cao 20  $\mu$ s, tốc độ kênh 20 Mbps.
- Kích thước gói: 64, 128, 256, 512, 1024, 1500 byte.
- Mạng: CSMA Ethernet (CsmHelper) với DataRate và Delay tùy theo kịch bản.
- Cấu hình IP: Tất cả node cài Internet Stack, dải 10.1.1.0/24.
- Ứng dụng TCP: BulkSendHelper từ node 0  $\rightarrow$  node 4, MaxBytes không giới hạn, SendSize = kích thước gói, cổng 50000; máy chủ PacketSink ở node 4.
- Ứng dụng UDP: OnOffHelper từ node 0  $\rightarrow$  node 4, DataRate 100 Mbps, PacketSize = kích thước gói, cổng 4000; máy chủ PacketSink ở node 4.
- Thu thập số liệu: Dùng FlowMonitor, các thông số: Transmitted Packets (gói gửi), Received Packets (gói nhận), Throughput (Thông lượng), PDR (Packet Delivery Ratio- tỷ lệ gói tin được chuyển thành công) .
- Quy trình mô phỏng: Lập qua các kịch bản và kích thước gói  $\rightarrow$  cài đặt node, thiết lập CSMA  $\rightarrow$  cài ứng dụng TCP/UDP  $\rightarrow$  chạy mô phỏng  $\rightarrow$  thu thập số liệu  $\rightarrow$  ghi CSV và in console  $\rightarrow$  dọn dẹp.

## 3. Công cụ sử dụng

### 3.1. NS-3(Network Simulator 3)

NS-3 được sử dụng làm nền tảng mô phỏng chính trong dự án. Công cụ này cho phép xây dựng mô hình mạng CSMA Ethernet với khả năng tùy chỉnh chi tiết cấu hình như tốc độ đường truyền, độ trễ, số lượng nút mạng và các ứng dụng truyền tải. Những chức năng chính được sử dụng:

- csma-module: tạo liên kết Ethernet CSMA và mô phỏng cơ chế truy cập đường truyền.
- internet-module: cài đặt giao thức IP và định tuyến cho các nút.
- applications-module: triển khai lưu lượng TCP (BulkSend) và UDP (OnOff) giữa các nút.

- flow-monitor: thu thập và ghi nhận các thông số hiệu suất như throughput, Rx/Tx packets, RxBytes và PDR để phục vụ phân tích.

NS-3 giúp mô phỏng sát thực tế quá trình truyền dữ liệu trong mạng Ethernet sử dụng CSMA/CD, từ đó đánh giá chính xác ảnh hưởng của kích thước gói tin.

### 3.2 Python

Python được sử dụng như công cụ xử lý và trực quan hóa dữ liệu sau mô phỏng. Các thư viện chính gồm:

- pandas: Dùng để đọc file CSV do NS-3 xuất ra, tổ chức lại dữ liệu theo từng kích thước gói tin và giao thức. pandas hỗ trợ thao tác như lọc dữ liệu, tính toán, hoặc tổng hợp các giá trị theo nhóm để chuẩn bị cho việc vẽ biểu đồ.
- matplotlib: cho phép biểu diễn dữ liệu dưới dạng line chart hoặc bar chart, giúp việc phân tích xu hướng trở nên trực quan và rõ ràng.

Python đảm nhiệm toàn bộ giai đoạn hậu xử lý, biến dữ liệu thô từ mô phỏng thành đồ thị trực quan để đánh giá hiệu suất mạng theo từng kích thước gói tin.

## 4. Kết quả

### 4.1. Kết quả mô phỏng từ NS-3

#### a. Trường hợp mạng lý tưởng (DataRate=100 Mbps, Delay=1 $\mu$ s)

Simulation Results for PacketSize = 64 bytes Scenario = ideal	Simulation Results for PacketSize = 128 bytes Scenario = ideal
<p>TCP:</p> <p>Transmitted Packets: 169909</p> <p>Received Packets: 169793</p> <p>Throughput: 58.277 Mbps</p> <p>Packet Delivery Ratio: 99.93 %</p> <p>UDP:</p> <p>Transmitted Packets: 781249</p> <p>Received Packets: 263939</p> <p>Throughput: 27.07 Mbps</p> <p>Packet Delivery Ratio: 33.78 %</p>	<p>TCP:</p> <p>Transmitted Packets: 167123</p> <p>Received Packets: 166889</p> <p>Throughput: 54.94 Mbps</p> <p>Packet Delivery Ratio: 99.86 %</p> <p>UDP:</p> <p>Transmitted Packets: 390624</p> <p>Received Packets: 201033</p> <p>Throughput: 34.55 Mbps</p> <p>Packet Delivery Ratio: 51.46 %</p>

Hình 1: Trường hợp mạng lý tưởng với gói 64 bytes và 128 bytes

```

=====
Simulation Results for PacketSize = 256 bytes
Scenario = ideal
=====
TCP:
Transmitted Packets: 142633
Received Packets: 142411
Throughput: 48.05 Mbps
Packet Delivery Ratio: 99.84 %
UDP:
Transmitted Packets: 195312
Received Packets: 150076
Throughput: 46.04 Mbps
Packet Delivery Ratio: 76.84 %
=====

```

```

=====
Simulation Results for PacketSize = 512 bytes
Scenario = ideal
=====
TCP:
Transmitted Packets: 122157
Received Packets: 121902
Throughput: 42.63 Mbps
Packet Delivery Ratio: 99.79 %
UDP:
Transmitted Packets: 97656
Received Packets: 97080
Throughput: 56.66 Mbps
Packet Delivery Ratio: 99.41 %
=====

```

Hình 2: Trường hợp mạng lý tưởng với gói 256 bytes và 512 bytes

```

=====
Simulation Results for PacketSize = 1024 bytes
Scenario = ideal
=====
TCP:
Transmitted Packets: 128503
Received Packets: 128334
Throughput: 44.98 Mbps
Packet Delivery Ratio: 99.87 %
UDP:
Transmitted Packets: 48828
Received Packets: 48425
Throughput: 56.81 Mbps
Packet Delivery Ratio: 99.17 %
=====

```

```

=====
Simulation Results for PacketSize = 1500 bytes
Scenario = ideal
=====
TCP:
Transmitted Packets: 129156
Received Packets: 129005
Throughput: 44.29 Mbps
Packet Delivery Ratio: 99.88 %
UDP:
Transmitted Packets: 33333
Received Packets: 32894
Throughput: 55.18 Mbps
Packet Delivery Ratio: 98.68 %
=====

```

Hình 3: Trường hợp mạng lý tưởng với gói 1024 bytes và 1500 byte

- Bảng tổng hợp số liệu trong mạng lý tưởng:

TCP:

Kích thước gói	Gói gửi	Gói nhận	Thông lượng(Mbps)	PDR(%)
64	169909	169793	58.277	99.93
128	167123	166889	54.94	99.86
256	142633	142411	48.05	99.84
512	122157	121902	42.63	99.79
1024	128503	128334	44.98	99.87
1500	129156	129005	44.29	99.88

UDP:

Kích thước gói	Gói gửi	Gói nhận	Thông lượng(Mbps)	PDR(%)
64	781249	263939	27.07	33.78
128	390624	201033	34.55	51.46
256	195312	150076	46.04	76.84
512	97656	97080	56.66	99.41
1024	48828	48425	56.81	99.17
1500	33333	32894	55.18	98.68



- Nhận xét số liệu:
  - TCP: Số gói gửi giảm nhẹ khi kích thước gói tăng, gói nhận gần bằng gói gửi, PDR rất cao (99.79-99.93%), thông lượng giảm nhẹ (58.28-44.29 Mbps). Nguyên nhân là do TCP kiểm soát tắc nghẽn và gửi lại gói bị mất, đảm bảo hầu hết gói được nhận. Khi gói lớn, số gói gửi ít hơn nhưng mỗi gói mang nhiều dữ liệu, thông lượng giảm nhẹ do cơ chế điều chỉnh tốc độ và tránh va chạm.
  - UDP: Số gói gửi cao hơn TCP rất nhiều, gói nhận thấp ở gói nhỏ nhưng gần bằng TCP khi gói lớn hơn 512 bytes, PDR tăng từ 33.78% đến 98.68%, thông lượng tăng theo kích thước gói (27.07-56.81 Mbps). Nguyên nhân là do UDP không kiểm soát tắc nghẽn, gửi liên tục nên số gói gửi nhiều. Gói nhỏ dễ bị mất do va chạm hoặc hạn chế kênh; gói lớn mang nhiều dữ liệu hơn, PDR cải thiện và thông lượng tăng.
  - Nhìn chung TCP luôn ổn định, PDR cao, thông lượng giảm nhẹ khi kích thước gói tăng. UDP gửi nhiều gói, thông lượng tăng theo gói lớn, PDR thấp ở gói nhỏ nhưng gần bằng TCP khi gói lớn.

**b. Trường hợp mạng nghẽn (DataRate=20 Mbps, Delay=20  $\mu$ s)**

Simulation Results for PacketSize = 64 bytes Scenario = congest	Simulation Results for PacketSize = 128 bytes Scenario = congest
TCP: Transmitted Packets: 20001 Received Packets: 19933 Throughput: 7.02 Mbps Packet Delivery Ratio: 99.66 % UDP: Transmitted Packets: 781249 Received Packets: 64520 Throughput: 5.94 Mbps Packet Delivery Ratio: 8.26 %	TCP: Transmitted Packets: 21367 Received Packets: 21315 Throughput: 7.55 Mbps Packet Delivery Ratio: 99.76 % UDP: Transmitted Packets: 390624 Received Packets: 45521 Throughput: 7.10 Mbps Packet Delivery Ratio: 11.65 %

Hình 4: Trường hợp mạng nghẽn với gói 64 bytes và 128 bytes

Simulation Results for PacketSize = 256 bytes Scenario = congest	Simulation Results for PacketSize = 512 bytes Scenario = congest
TCP: Transmitted Packets: 21877 Received Packets: 21825 Throughput: 7.79 Mbps Packet Delivery Ratio: 99.76 % UDP: Transmitted Packets: 195312 Received Packets: 29586 Throughput: 8.40 Mbps Packet Delivery Ratio: 15.15 %	TCP: Transmitted Packets: 20140 Received Packets: 20060 Throughput: 7.20 Mbps Packet Delivery Ratio: 99.60 % UDP: Transmitted Packets: 97656 Received Packets: 19162 Throughput: 10.35 Mbps Packet Delivery Ratio: 19.62 %

Hình 5: Trường hợp mạng nghẽn với gói 256 bytes và 512 bytes

<pre> ===== Simulation Results for PacketSize = 1024 bytes Scenario = congest ===== TCP: Transmitted Packets: 17340 Received Packets: 17263 Throughput: 6.14 Mbps Packet Delivery Ratio: 99.56 % UDP: Transmitted Packets: 48828 Received Packets: 11955 Throughput: 12.58 Mbps Packet Delivery Ratio: 24.48 % </pre>	<pre> ===== Simulation Results for PacketSize = 1500 bytes Scenario = congest ===== TCP: Transmitted Packets: 11974 Received Packets: 11906 Throughput: 4.16 Mbps Packet Delivery Ratio: 99.43 % UDP: Transmitted Packets: 33333 Received Packets: 7216 Throughput: 11.03 Mbps Packet Delivery Ratio: 21.65 % </pre>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Hình 6: Trường hợp mạng nghẽn với gói 1024 bytes và 1500 bytes

- Bảng tổng hợp số liệu trong mạng nghẽn:

TCP:

Kích thước gói	Gói gửi	Gói nhận	Thông lượng(Mbps)	PDR(%)
64	20001	19933	7.02	99.66
128	21367	21315	7.55	99.76
256	21877	21825	7.79	99.76
512	20140	20060	7.20	99.60
1024	17340	17263	14,99	99.56
1500	11974	11906	4.16	99.43

UDP:

Kích thước gói	Gói gửi	Gói nhận	Thông lượng(Mbps)	PDR(%)
64	781249	64520	5.94	8.26
128	390624	45521	7.10	11.65
256	195312	29586	8.40	15.15
512	97656	19162	10.35	19.62
1024	48828	11955	12.58	24.48
1500	33333	7216	11.03	21.65

- Nhận xét số liệu:

- TCP: Số gói gửi và nhận giảm mạnh so với lý tưởng, tuy nhiên PDR vẫn cao (99.43-99.76%), thông lượng thấp hơn nhưng ổn định (4.16-7.79 Mbps). Thông lượng giảm do kênh nghẽn nhưng số gói nhận vẫn gần bằng số gói gửi.
- UDP: Số gói gửi cao hơn nhiều so TCP (781249-33333), gói nhận lại rất thấp (64520-7216), PDR quá thấp (8.26–24.48%), thông lượng tăng nhẹ với gói lớn.
- Nhìn chung TCP vẫn ổn định và đáng tin cậy kể cả trong trường hợp mạng nghẽn, số gói nhận gần bằng gói gửi, PDR cao, thông lượng giảm đáng kể. UDP

mất rất nhiều gói, PDR thấp, thông lượng thực tế kém ở gói nhỏ, cải thiện nhẹ khi gói lớn hơn.

### c. File kết quả:

Đây là file kết quả được chương trình tạo ra trong quá trình chạy mô phỏng. File CSV được ghi tự động trong mỗi lần mô phỏng một cặp (Scenario, PacketSize), và chứa đầy đủ các thông số cần thiết để phân tích hiệu suất mạng CSMA dưới hai giao thức TCP và UDP.

Cấu trúc file gồm các trường:

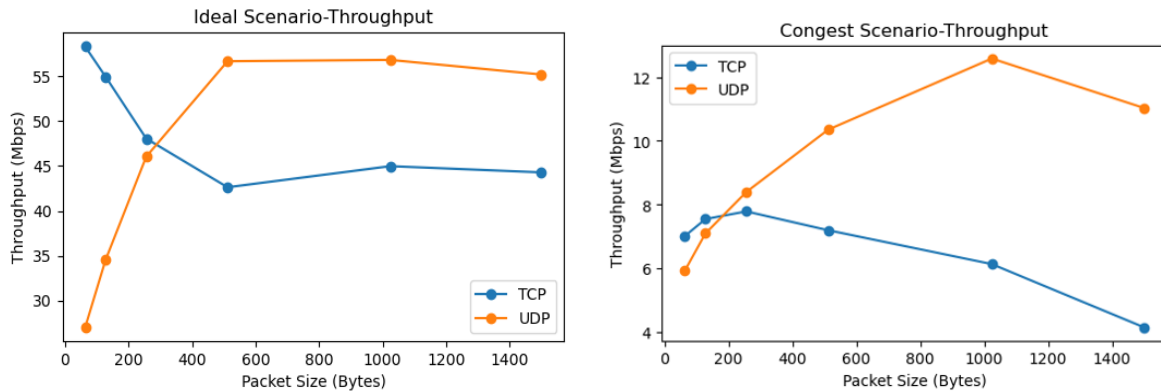
- PacketSize: kích thước gói tin được sử dụng (64–1500 bytes).
- Scenario: trường hợp mô phỏng gồm:
  - ideal – kịch bản lý tưởng (tốc độ cao, độ trễ thấp),
  - congest – kịch bản nghẽn (tốc độ thấp, độ trễ lớn).
- Protocol: TCP hoặc UDP.
- TxPackets: tổng số gói tin đã gửi.
- RxPackets: tổng số gói tin nhận thành công.
- ThroughputMbps: thông lượng đạt được tại đích, tính bằng Mbps.
- PDR (Packet Delivery Ratio): tỷ lệ gói nhận trên gói gửi (%).

File này được xây dựng từ FlowMonitor và được ghi lại theo định dạng bảng trong excel, giúp việc xử lý và vẽ đồ thị sau mô phỏng trở nên thuận tiện.

PacketSize	Scenario	Protocol	TxPackets	RxPackets	ThroughputMbps	PDR
64	ideal	TCP	169909	169793	58.277	99.93
64	ideal	UDP	781249	263939	27.07	33.78
128	ideal	TCP	167123	166889	54.94	99.86
128	ideal	UDP	390624	201033	34.55	51.46
256	ideal	TCP	142633	142411	48.05	99.84
256	ideal	UDP	195312	150076	46.04	76.84
512	ideal	TCP	122157	121902	42.63	99.79
512	ideal	UDP	97656	97080	56.66	99.41
1024	ideal	TCP	128503	128334	44.98	99.87
1024	ideal	UDP	48828	48425	56.81	99.17
1500	ideal	TCP	129156	129005	44.29	99.88
1500	ideal	UDP	33333	32894	55.18	98.68
64	congest	TCP	20001	19933	7.02	99.66
64	congest	UDP	781249	64520	5.94	8.26
128	congest	TCP	21367	21315	7.55	99.76
128	congest	UDP	390624	45521	7.10	11.65
256	congest	TCP	21877	21825	7.79	99.76
256	congest	UDP	195312	29586	8.40	15.15
512	congest	TCP	20140	20060	7.20	99.60
512	congest	UDP	97656	19162	10.35	19.62
1024	congest	TCP	17340	17263	6.14	99.56
1024	congest	UDP	48828	11955	12.58	24.48
1500	congest	TCP	11974	11906	4.16	99.43
1500	congest	UDP	33333	7216	11.03	21.65

Hình 7: Số liệu thô được ghi vào trong file csma\_metrics.csv

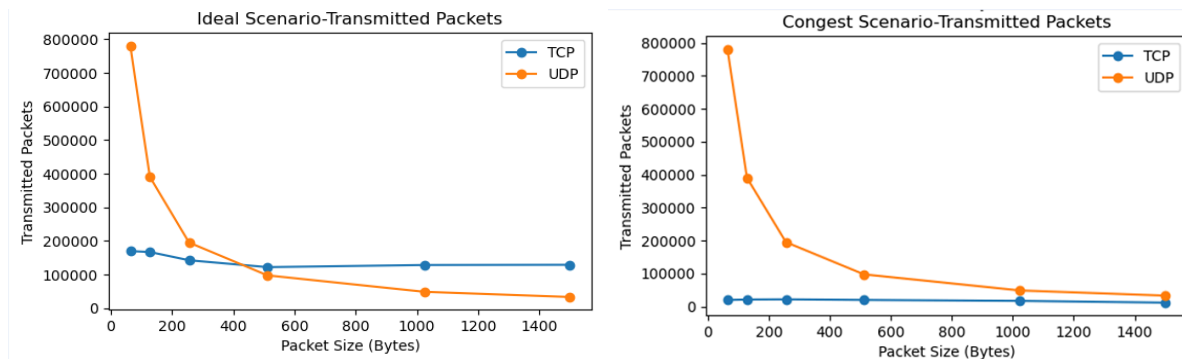
## 4.2. Phân tích đồ thị thông lượng TCP và UDP theo kích thước gói



Hình 8: Đồ thị thông lượng theo kích thước gói trong mạng lý tưởng và mạng nghẽn

- Trường hợp mạng lý tưởng:
  - TCP: Throughput giảm đều khi kích thước gói tăng. Ở 64 bytes đạt khoảng 58 Mbps và giảm về khoảng 42-45 Mbps ở các kích thước lớn. Đồ thị thể hiện dạng giảm dần và tương đối ổn định khi gói lớn hơn 512 bytes.
  - UDP: Throughput tăng theo kích thước gói, từ mức thấp nhất khoảng 27 Mbps tại 64 byte lên trên 55 Mbps tại 512-1500 bytes. Đồ thị có dạng tăng nhanh và ổn định dần ở kích thước lớn
- Trường hợp mạng nghẽn:
  - TCP: Throughput duy trì trong vùng 4–8 Mbps với thay đổi nhỏ theo kích thước. Đồ thị gần như phẳng với dao động nhẹ.
  - UDP: Đồ thị thể hiện xu hướng tăng tuyến tính nhẹ. Throughput tăng theo kích thước gói, từ khoảng 6 Mbps lên khoảng 10–12 Mbps.
- Nhận xét: TCP duy trì thông lượng ổn định và mượt ở cả 2 trường hợp ; UDP biến động nhiều, đặc biệt trong kênh nghẽn. Thông lượng của UDP tăng rõ với kích thước gói khi kênh lý tưởng, nhưng giảm hiệu quả khi kênh nghẽn, cho thấy UDP phụ thuộc mạnh vào trạng thái mạng.

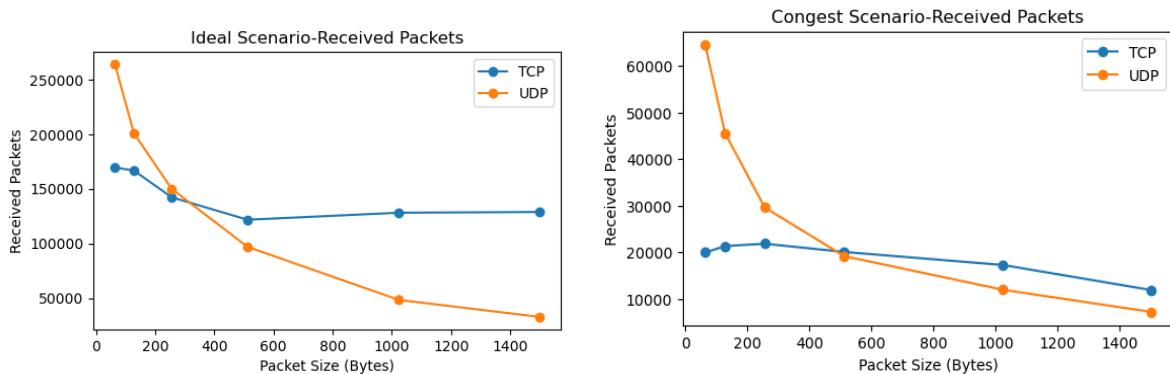
### 4.3. Phân tích đồ thị gói gửi của TCP và UDP theo kích thước gói



Hình 9: Đồ thị gói gửi theo kích thước gói trong mạng lý tưởng và mạng nghẽn

- Trường hợp mạng lý tưởng:
  - TCP: Số gói gửi giảm đều theo kích thước gói. Đồ thị có dạng đường giảm tuyến tính, thể hiện rằng khi kích thước gói tăng, cùng lượng dữ liệu cần truyền, số gói giảm tương ứng. TCP ổn định, không có biến động bất thường giữa các kích thước.
  - UDP: Số gói gửi nhiều hơn hẳn TCP do không kiểm soát tắc nghẽn cũng như không chờ ACK. Số gói gửi cũng giảm theo kích thước, đồ thị giảm nhanh hơn TCP ở gói nhỏ do UDP gửi dữ liệu liên tục và tốc độ cao hơn, nhưng khi gói lớn hơn 512 bytes, đồ thị gần như song song với TCP, thể hiện ổn định.
- Trường hợp mạng nghẽn:
  - TCP: Số gói gửi giảm nhẹ khi tăng kích thước. Đường đồ thị phẳng và mượt, dao động nhỏ, cho thấy TCP giữ tốc độ gửi ổn định nhờ cơ chế kiểm soát tắc nghẽn, dù kênh nghẽn.
  - UDP: Số gói gửi giảm, đường đồ thị giảm nhưng vẫn cao hơn TCP, thể hiện UDP gửi liên tục, không kiểm soát tắc nghẽn, nhưng số gói giảm dần khi gói lớn hơn do giới hạn băng thông và va chạm tăng.
- Nhận xét: Cả TCP và UDP đều thể hiện xu hướng giảm số gói gửi khi kích thước gói tăng. TCP ổn định và mượt hơn, ít biến động, phản ánh cơ chế kiểm soát tắc nghẽn hiệu quả. UDP gửi nhiều gói hơn TCP, đặc biệt trong kênh nghẽn, nhưng giảm dần khi gói lớn do băng thông hạn chế và va chạm tăng.

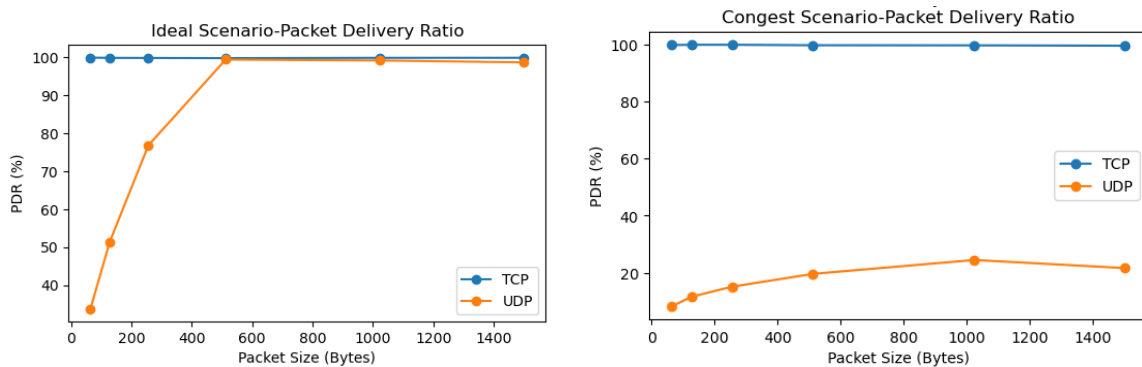
#### 4.4. Phân tích đồ thị gói nhận của TCP và UDP theo kích thước gói



Hình 10: Đồ thị gói nhận theo kích thước gói trong mạng lý tưởng và mạng nghẽn

- Trường hợp mạng lý tưởng:
  - TCP: Gói nhận gần bằng gói gửi, giảm dần khi kích thước gói tăng do số gói gửi giảm theo tải cố định. Đồ thị giảm tuyến tính theo kích thước.
  - UDP: Ở gói nhỏ lượng gói nhận thấp do mất nhiều gói, nhưng tăng nhanh theo kích thước và gần bằng gói gửi khi kích thước từ 512 bytes trở lên. Đồ thị thể hiện xu hướng tăng mạnh rồi tiến tới cân bằng.
- Trường hợp mạng nghẽn:
  - TCP: Gói nhận gần bằng gói gửi và giảm dần theo kích thước gói vì tổng tải cố định làm số gói gửi giảm. Đồ thị giảm tuyến tính.
  - UDP: Gói nhận thấp ở kích thước nhỏ nhưng tăng nhanh khi kích thước lớn và gần bằng gói gửi khi gói  $\geq 512$  byte. Đồ thị tăng mạnh rồi tiến đến cân bằng.
- Nhận xét: TCP duy trì số gói nhận ổn định, gần bằng gói gửi, phản ánh cơ chế đảm bảo độ tin cậy và kiểm soát tắc nghẽn. UDP thay đổi nhiều, dễ mất gói khi kênh nghẽn hoặc gói nhỏ, nhưng cải thiện khi gói lớn và kênh lý tưởng. Đường đồ thị TCP mượt và cũng ít biến động, còn UDP có xu hướng giảm dần theo kích thước gói tin do gói gửi cũng giảm dần.

#### 4.5 Phân tích đồ thị Packet Delivery Ratio (PDR) theo kích thước gói:



Hình 11: Đồ thị PDR theo kích thước gói trong mạng lý tưởng và mạng nghẽn

- PDR là viết tắt của Packet Delivery Ratio, nghĩa là tỷ lệ gói tin được chuyển thành công. Công thức tính:

$$PDR (\%) = \frac{\text{Số gói nhận thành công}}{\text{Số gói gửi}} \times 100$$

- Giá trị càng cao (gần 100%) nghĩa là mạng tin cậy, hầu hết gói tin được truyền đến đích.
- Giá trị thấp cho thấy mất gói nhiều, mạng bị nghẽn hoặc không ổn định.
- Trường hợp mạng lý tưởng:
  - TCP: PDR ổn định trong vùng 99.8–99.9% ở mọi kích thước. Đồ thị tạo thành đường thẳng ổn định.
  - UDP: PDR thấp ở gói nhỏ, tuy nhiên lại tăng mạnh theo kích thước, từ 33.78% ở 64 byte lên trên 98% khi gói lớn hơn 512 bytes. Đồ thị tăng dốc rõ ràng.
- Trường hợp mạng nghẽn:
  - TCP: PDR giữ mức ổn định quanh 99.4–99.8%. Đồ thị vẫn tạo thành 1 đường thẳng.
  - Congest – UDP: PDR cực kỳ thấp ở gói nhỏ (8–15%), tăng đến khoảng 24% ở 1024 byte rồi giảm nhẹ tại 1500 byte. Đồ thị tăng nhẹ nhưng thiếu ổn định.
- Nhận xét: TCP luôn duy trì PDR cao và ổn định trong cả kênh lý tưởng và nghẽn. UDP biến động mạnh, phụ thuộc vào kích thước gói và trạng thái kênh: gói nhỏ trong mạng nghẽn hay lý tưởng đều mất nhiều gói, gói lớn trong mạng lý tưởng cải thiện đáng kể PDR, gần bằng với TCP. Đường đồ thị PDR TCP mượt và cao; đường UDP dốc tăng với gói lớn nhưng vẫn rất thấp và không ổn định trong kênh nghẽn.

#### **4.6. Đánh giá chung:**

##### **a. TCP**

- Ổn định và đáng tin cậy trong cả 2 trường hợp mạng lý tưởng và nghẽn.
- Số gói gửi và nhận đều gần bằng nhau, giảm nhẹ khi kích thước gói tăng (64 bytes đến 1500 bytes).
- Thông lượng: giảm nhẹ khi gói lớn trong mạng lý tưởng (58.277-44.29 Mbps), duy trì ổn định trong mạng nghẽn (7.02-4.16 Mbps).
- PDR luôn cao(99.4–99.93%), cho thấy cơ chế kiểm soát tắc nghẽn và retransmit hoạt động hiệu quả.
- Xu hướng đồ thị phẳng, mượt, gần như không biến động, phản ánh tính ổn định và khả năng chống nghẽn tốt của TCP.

##### **b. UDP**

- Số gói gửi: cao hơn TCP rất nhiều, do cơ chế gửi liên tục, không kiểm soát tắc nghẽn.
- Số gói nhận và PDR: thấp ở gói nhỏ hoặc khi kênh nghẽn, cải thiện khi gói lớn trong kênh lý tưởng.
- Thông lượng: tăng theo kích thước gói trong kênh lý tưởng (27.07-55.18 Mbps) và cao hơn so với TCP, thấp và biến động trong kênh nghẽn (5.94-11.03 Mbps).
- Xu hướng: Đường đồ thị biến động mạnh, PDR thấp khi gói nhỏ hoặc nghẽn, thông lượng phụ thuộc kích thước gói và trạng thái kênh.

#### **5. Kết luận**

- Hiệu suất giao thức: Trong mạng CSMA Ethernet, TCP duy trì độ tin cậy cao và ổn định nhờ cơ chế kiểm soát tắc nghẽn và retransmit, PDR gần 100% ngay cả khi kênh nghẽn. UDP gửi liên tục, dẫn đến số gói gửi lớn nhưng mất nhiều gói khi kênh nghẽn hoặc gói nhỏ; hiệu suất thực tế phụ thuộc vào kích thước gói và trạng thái kênh.
- Ảnh hưởng của kích thước gói:
  - TCP: Số gói gửi giảm khi kích thước gói tăng, thông lượng giảm nhẹ nhưng ổn định.
  - UDP: Gói lớn cải thiện PDR và thông lượng; gói nhỏ dễ mất nhiều, đặc biệt khi kênh nghẽn.
- So sánh trong 2 trường hợp lý tưởng và nghẽn:
  - Lý tưởng (ideal): CSMA Ethernet cho phép cả TCP và UDP hoạt động hiệu quả; TCP ổn định, UDP có thông lượng cao với gói lớn.



- Nghẽn (congest): CSMA Ethernet giới hạn hiệu suất UDP do va chạm và mất gói; TCP vẫn duy trì PDR cao và số gói nhận gần bằng số gói gửi.
- Nhận xét tổng thể về CSMA Ethernet:
  - CSMA hoạt động tốt với TCP, duy trì ổn định, kiểm soát tắc nghẽn hiệu quả.
  - Với UDP, CSMA Ethernet dễ gây va chạm và mất gói khi tải cao hoặc gói nhỏ, phản ánh hạn chế của cơ chế truy cập ngẫu nhiên trong mạng nghẽn.
  - Kích thước gói lớn giúp cải thiện hiệu suất UDP nhờ giảm số lượng gói và tăng lượng dữ liệu mỗi gói, giảm va chạm tương đối.
- Tóm lại: Trong CSMA Ethernet, TCP tối ưu cho độ tin cậy và ổn định, trong khi UDP có tốc độ gửi cao nhưng dễ mất gói, đặc biệt trong kênh nghẽn. Kích thước gói và trạng thái kênh là yếu tố quan trọng ảnh hưởng đến hiệu suất.

#### **Tài liệu tham khảo:**

- [1] ns-3, *ns-3 Tutorial*, 2025
- [2] Geeksforgeeks, *Carrier Sense Multiple Access (CSMA)*, 2025.
- [3] NS-3 Simulation, *What is CSMA Module in Ns3*, 2025
- [4] ns-3, *Building a Bus Network Topology*, 2009
- [5] Geeksforgeeks, *Collision Detection in CSMA/CD*, 2025