

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



BÁO CÁO
LẬP TRÌNH MẠNG

ĐỀ TÀI:

ĐÁNH GIÁ HIỆU SUẤT CỦA CSMA TRONG
MẠNG ETHERNET DÙNG NS3

Sinh viên thực hiện: Trần Quang Khánh (106220220)

Lê Xuân Nam (106220226)

Đặng Quang Lâm (106220256)

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	Số thẻ sinh viên	Lớp sinh hoạt	Phân công nhiệm vụ	Phần trăm đóng góp
Trần Quang Khánh	106220220	22KTMT1	Viết báo cáo phần giới thiệu, 2 phần đầu tiên của phần phương pháp, thực hiện cài đặt, mô phỏng và ra kết quả.	34%
Lê Xuân Nam	106220226		Viết báo cáo phần phân tích kết quả, hỗ trợ đánh giá và viết báo cáo phần kết quả mô phỏng.	33%
Đặng Quang Lâm	106220256	22KTMT2	Viết báo cáo phần kịch bản mô phỏng (ở phần phương pháp) và các công cụ sử dụng cho đề tài.	33%
Tổng				100%

MỤC LỤC

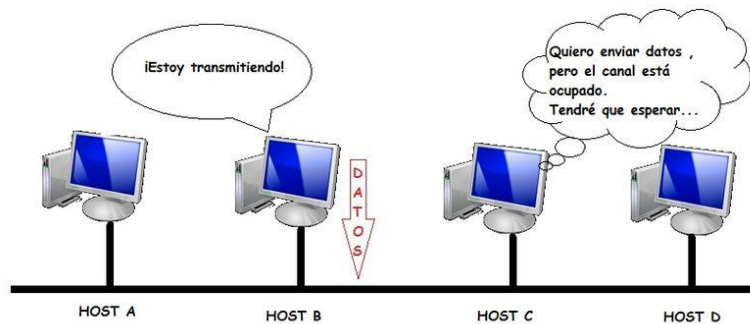
1. Giới thiệu:	3
1.1. Tổng quan về CSMA:	3
1.2. Cơ chế hoạt động:	3
1.3. Mục tiêu nghiên cứu:	4
2. Phương pháp:	5
2.1. Mô hình CSMA và những phát hiện của nhóm:	5
2.2. Sơ đồ topo, sơ đồ CSMA NetDevice và vai trò của các node:	5
2.3. Kịch bản mô phỏng:	7
3. Công cụ sử dụng:	7
3.1. Sử dụng NS3:	7
3.2. Ngôn ngữ lập trình Python:	8
4. Phân tích kết quả:	9
4.1. Tổng hợp kết quả mô phỏng từ NS-3:	9
4.2. Đồ thị thông lượng TCP và UDP theo thời gian:	12
4.3. Phân tích đồ thị Packet Delivery Ratio (PDR) theo thời gian:	13
4.4. Phân tích đồ thị số gói nhận tích lũy theo thời gian:	14
4.5. Phân tích đồ thị dữ liệu nhận tích lũy theo thời gian:	14
4.6. Phân tích đồ thị số gói bị mất theo thời gian:	15
4.7. Phân tích đồ thị số gói gửi tích lũy theo thời gian:	16
4.8. Phân tích đồ thị so sánh các chỉ số tổng hợp:	17
4.9. Phân tích hiệu suất của giao thức theo thời gian:	17
4.10. Bảng tổng hợp đánh giá các chỉ số hiệu suất:	18
5. Kết luận:	19
Tài liệu tham khảo:	19

1. Giới thiệu:

1.1. Tổng quan về CSMA:

CSMA là một phương thức đa truy cập kênh truyền dữ liệu trong mạng máy tính, cho phép nhiều thiết bị cùng chia sẻ một kênh truyền chung mà vẫn giảm thiểu khả năng xảy ra xung đột (collision) giữa các thiết bị [1].

CSMA thường được áp dụng ở lớp liên kết dữ liệu (Data Link Layer) trong mô hình OSI, và được sử dụng rộng rãi trong các công nghệ mạng như Ethernet và Wi-Fi.



Hình 1. Hình ảnh mô tả các máy tính kết nối và truyền thông với nhau qua Ethernet sử dụng cơ chế CSMA [1].

Ý nghĩa:

- CSMA giúp nâng cao hiệu suất sử dụng kênh truyền bằng cách tránh các trường hợp truyền đồng thời khi kênh đã bận.
- Giải pháp này tương đối đơn giản, chi phí thấp, và phù hợp với nhiều môi trường mạng khác nhau – cả có dây và không dây.

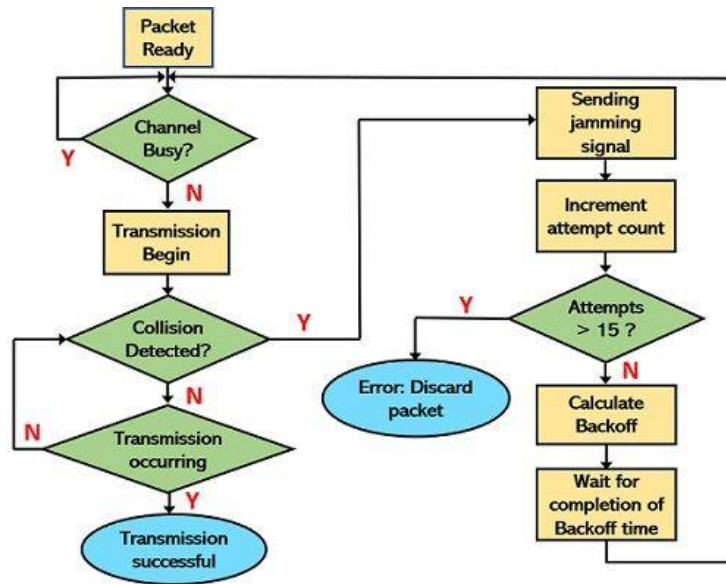
Tuy nhiên, CSMA cũng có giới hạn: khi số lượng thiết bị chia sẻ kênh lớn, hoặc khi lưu lượng mạng cao, việc chờ kênh nhàn và xung đột vẫn có thể dẫn tới độ trễ, hiệu suất giảm.

1.2. Cơ chế hoạt động:

CSMA (Carrier Sense Multiple Access) có thể được chia thành hai biến thể chính dựa trên cách xử lý xung đột trong quá trình truyền dữ liệu:

- CSMA/CD – Carrier Sense Multiple Access with Collision Detection: Dùng trong mạng Ethernet có dây.
- CSMA/CA – Carrier Sense Multiple Access with Collision Avoidance: Dùng trong Wi-Fi (IEEE 802.11).

Trong NS3, module CSMA mô phỏng Ethernet dạng half-duplex, tức là CSMA-CD. Do đó đối tượng nghiên cứu sẽ là CSMA-CD. Hình 2 dưới đây là lưu đồ thuật toán mô tả cơ chế hay nguyên lý hoạt động của CSMA-CD [2]:



Hình 2: Lưu đồ thuật toán mô tả nguyên lý hoạt động của CSMA-CD [2].

Căn cứ vào Hình 2, Cơ chế CSMA-CD (Carrier Sense Multiple Access with Collision Detection) hoạt động theo các bước sau:

- Bước 1: Khi một nút mạng có dữ liệu cần gửi, nó bắt đầu quá trình truy cập kênh truyền.
- Bước 2: Nút sẽ lắng nghe (carrier sense) để xác định xem kênh có đang bận hay không. Nếu kênh bận → nút tiếp tục chờ và kiểm tra lại cho đến khi kênh rảnh. Nếu kênh rảnh → bắt đầu truyền dữ liệu.
- Bước 3: Trong quá trình truyền, nút liên tục giám sát tín hiệu trên đường truyền để phát hiện xem có xảy ra xung đột (collision) hay không.
- Bước 4: Nếu không xảy ra xung đột → quá trình truyền dữ liệu được hoàn tất và gói tin được gửi thành công. Nếu xảy ra xung đột → nút lập tức dừng truyền và phát jam signal để thông báo xung đột cho các nút khác trên mạng.
- Bước 5: Sau khi gửi jam signal, nút tăng bộ đếm số lần thử truyền lại (attempt count). Nếu số lần thử vượt quá giới hạn (16 lần theo chuẩn Ethernet) → gói tin bị loại bỏ. Nếu chưa vượt giới hạn → nút tính toán thời gian chờ để gửi lại gói.
- Bước 6: Sau khi chờ hết thời gian chờ, nút quay lại Bước 2 để kiểm tra kênh và thử truyền lại.

1.3. Mục tiêu nghiên cứu:

Mục tiêu nghiên cứu của chủ đề hiệu suất CSMA trong mạng Ethernet qua NS3 như sau:

- Sử dụng mô hình mô phỏng mạng Ethernet và cơ chế truy cập đa truy nhập cảm nhận sóng mang (CSMA) trong NS-3.

- Đánh giá hiệu suất CSMA trên mạng Ethernet dựa trên thông lượng, số gói nhận thành công, tỉ lệ giao thành công (PDR) và packet loss, phản ánh khả năng khai thác băng thông và độ ổn định của TCP và UDP trong các tình huống.

2. Phương pháp:

2.1. Mô hình CSMA và những phát hiện của nhóm:

Mô hình CSMA trong ns-3 dùng đối tượng `CsmaNetDevice` và kênh `CsmaChannel` để mô phỏng một mạng “bus” kiểu Ethernet đơn giản [3].

Về mô hình kênh `CSMAChannel`:

- `CsmaChannel` mô phỏng môi trường truyền chung cho nhiều thiết bị, không giới hạn số lượng thiết bị kết nối.
- Kênh có 3 trạng thái: IDLE → TRANSMITTING → PROPAGATING → trở lại IDLE. Khi trạng thái thay đổi, tất cả thiết bị kết nối đều “nhận biết” tức thì trạng thái mới.
- Vì trong mô hình tất cả thiết bị “nhận biết” kênh cùng lúc, nên mô hình đơn giản hóa và không cần mô phỏng xung đột thực sự như trong CSMA/CD thật.

Về mô hình thiết bị (`CSMANetDevice`):

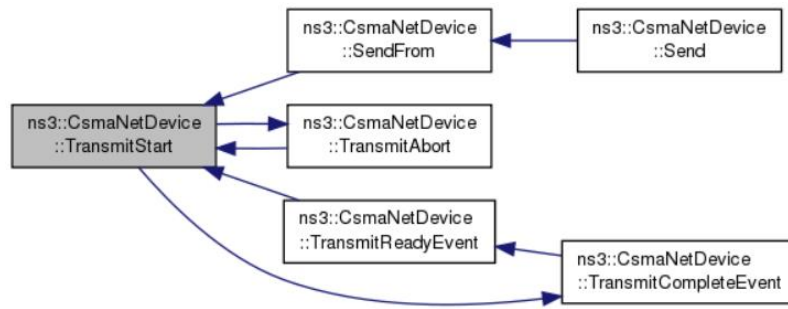
- Mỗi nút (node) mạng được gán cho 1 nút `CsmaNetDevice`, có các thuộc tính như:
 - Address: địa chỉ MAC.
 - SendEnable, ReceiveEnable: bật hoặc tắt các chức năng truyền/nhận.
 - EncapsulationMode: chế độ đóng gói khung Ethernet.
 - TxQueue: hàng đợi truyền dữ liệu.
 - InterframeGap: thời gian chờ giữa các khung.

Những phát hiện của nhóm đối với mô hình:

- Mô hình CSMA trong NS3 không hoàn toàn giống mạng Ethernet thật (IEEE 802.3), ví dụ như không mô phỏng jam signal, do đó nhóm không thể mô phỏng tình huống xung đột theo lớp vật lý [3].
- Do cấu hình thuộc tính trong `CsmaHelper` và `CsmaNetDevice` rất linh hoạt, nếu thay đổi không hợp lý có thể tạo ra kết quả giả lập “không thực tế” và không chính xác, lí do là vì kết quả hiệu suất sẽ còn phụ thuộc nhiều vào thuộc tính của mô hình.

2.2. Sơ đồ topo, sơ đồ CSMA NetDevice và vai trò của các node:

Hình 3 dưới đây mô tả sơ đồ mô hình phương thức `CSMANetDevice` trong NS3, thể hiện các luồng gọi hàm liên quan đến truyền dữ liệu:

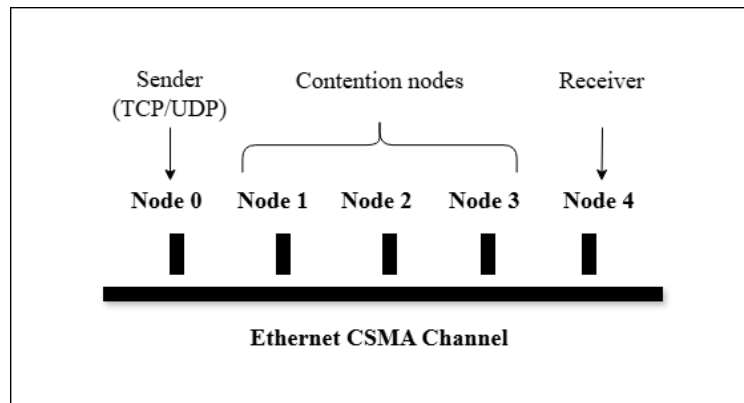


Hình 3. Sơ đồ phương thức CSMA NetDevice trong NS3 [3].

Vai trò của từng phương thức:

- Send / SendFrom: Bắt đầu quá trình gửi gói dữ liệu.
- TransmitStart / TransmitAbort: Quản lý trạng thái bắt đầu truyền hoặc hủy truyền nếu gặp vấn đề.
- TransmitStart: Bắt đầu truyền.
- TransmitReadyEvent: Sự kiện thông báo thiết bị sẵn sàng truyền tiếp.
- TransmitCompleteEvent: Sự kiện báo hoàn thành việc truyền gói dữ liệu.

Sơ đồ topo mô phỏng được minh họa qua Hình 4 dựa vào 2 traffic TCP và UDP dưới đây:



Hình 5. Sơ đồ topo mô phỏng dựa vào 2 traffic TCP và UDP.

Vai trò của các node trong sơ đồ topo:

- Node 0: Ở 2 traffic đều là sender (nơi gửi). Trong TCP: gửi theo tốc độ cấu hình nhưng chịu congestion control; trong UDP: gửi theo tốc độ cố định.
- Node 4: Ở 2 traffic đều là receiver (nơi nhận). Trong TCP: nhận data + gửi ACK ngược; trong UDP: nhận data mà không gửi ACK.
- Node 1, 2, 3: Các node trung gian gây nhiễu, có tác nhân gây tranh cãi đường truyền, tăng thời gian chờ, tạo tình trạng contention để mô phỏng 1 Ethernet thực. Các node này không có ứng dụng gửi/nhận, chỉ có chức năng tham gia vào CSMA protocol.

2.3. Kịch bản mô phỏng:

Mục tiêu mô phỏng: Mô phỏng mạng Ethernet sử dụng giao thức CSMA/CD trong môi trường NS-3 nhằm:

- Đánh giá hiệu suất truyền thông qua các chỉ số: throughput, packet loss, delay, PDR.
- So sánh hiệu suất giữa TCP và UDP trong cùng điều kiện mạng (theo chuẩn Ethernet).
- Điều kiện: Cô lập tác động của tắc nghẽn (appRate) và độ trễ (delay) lên hiệu suất.

Cấu hình mô phỏng:

- Số lượng node: 5 node (Node 0 gửi, Node 4 nhận, Node 1-3 gây nhiễu).
- Mô hình: Mạng CSMA-CD (theo CSMAHelper)
- Kênh truyền: Ethernet CSMA với tốc độ 100 Mbps, độ trễ mặc định 6560ns.
- Ứng dụng cho TCP và UDP: đều sử dụng mẫu OnOffApplication.
- Thời gian mô phỏng: 10 giây.
- Địa chỉ IP: được cấp phát tự động qua Ipv4AddressHelper.

Kịch bản mô phỏng:

- Trường hợp 1: Lí tưởng (Cấu hình appRate cân bằng với dataRate, delay thấp). Kết quả mong đợi: Hiệu suất đạt cực đại.
- Trường hợp 2: Độ trễ cao (tăng delay lên 500 μ s nhưng giữ tải). Kết quả: giảm thông lượng (throughput) của TCP.
- Trường hợp 3: Tắc nghẽn (tăng appRate lên 200Mbps làm quá tải). Kết quả: PDR thấp, TCP giảm tốc độ nhưng đảm bảo PDR.
- Trường hợp 4: Cực đoan (appRate tăng và delay đều tăng quá mức). Kết quả: TCP bị phạt kép dù có cơ chế chống nghẽn, UDP gần như mất hết gói.

Kết quả đầu ra:

- File .css từ FlowMonitor cung cấp chứa dữ liệu về throughput, delay, packet loss và cả PDR.
- Dữ liệu được xử lý bằng Python thông qua các gói hỗ trợ như Pandas hay Matplotlib để vẽ biểu đồ so sánh TCP và UDP.

3. Công cụ sử dụng:

3.1. Sử dụng NS3:

NS-3 (Network Simulator 3) là một trình mô phỏng mạng mã nguồn mở được thiết kế cho mục đích nghiên cứu và giảng dạy trong lĩnh vực mạng máy tính. Đây là công cụ mô phỏng phổ biến trong cộng đồng học thuật và công nghiệp nhờ khả năng mô hình hóa chi tiết các giao thức mạng và môi trường truyền thông.

Các đặc điểm nổi bật của NS-3:

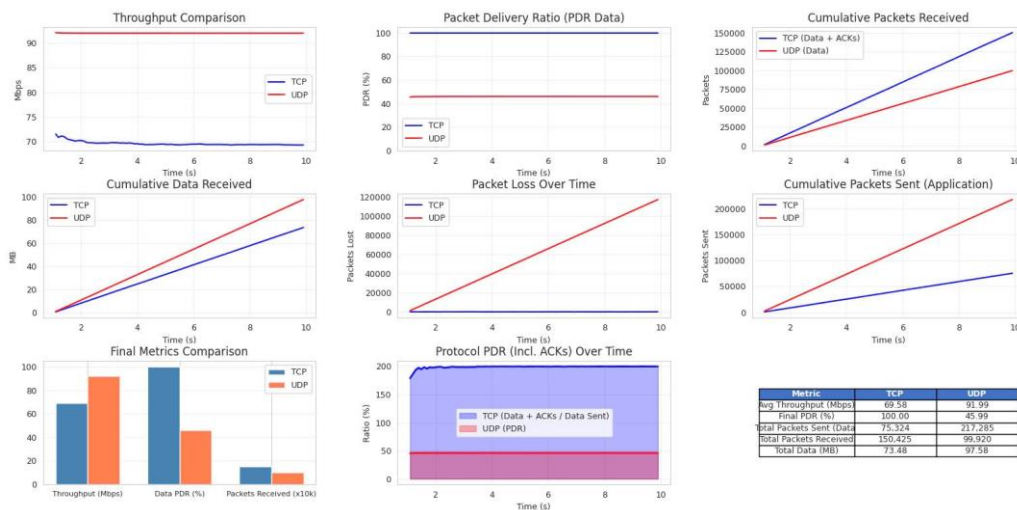
- Mô hình hóa chính xác: NS-3 cho phép mô phỏng các tầng mạng từ lớp vật lý đến lớp ứng dụng, hỗ trợ các giao thức như TCP, UDP, CSMA, Wi-fi, LTE, 5G...
- Môi trường thử nghiệm ảo: Người dùng có thể thiết lập các kịch bản mạng phức tạp để kiểm tra hiệu suất, độ trễ, thông lượng, tỷ lệ mất gói... mà không cần phần cứng thực tế.
- Hỗ trợ đa ngôn ngữ: NS-3 được chủ yếu viết bằng ngôn ngữ C++, nhưng cũng hỗ trợ Python thông qua bindings để tạo kịch bản mô phỏng.
- Tích hợp với công cụ phân tích: Có thể kết hợp với Wireshark để phân tích gói tin, hoặc với Gnuplot để vẽ đồ thị kết quả.

Trong đề tài này, NS-3 được sử dụng để mô phỏng mạng Ethernet sử dụng cơ chế CSMA-CD. Các thành phần chính bao gồm:

- CsmHelper: hỗ trợ cấu hình thiết bị và kênh CSMA.
- CsmNetDevice: mô phỏng thiết bị mạng Ethernet.
- CsmChannel: mô phỏng kênh truyền dạng bus.
- OnOffApplication: tạo lưu lượng UDP và TCP tương ứng.
- FlowMonitor: thu thập dữ liệu hiệu suất như throughput, PDR, packet loss.

3.2. Ngôn ngữ lập trình Python:

Python là ngôn ngữ lập trình bậc cao, dễ học, cú pháp rõ ràng, và được sử dụng rộng rãi trong khoa học dữ liệu, trí tuệ nhân tạo, và mô phỏng hệ thống.



Hình 6. Ví dụ biểu đồ hiệu suất TCP và UDP được vẽ bằng Python từ dữ liệu mô phỏng NS-3.

Lý do sử dụng Python trong NS-3:

- Tạo kịch bản mô phỏng dễ dàng: Python bindings của NS-3 cho phép viết các kịch bản mô phỏng bằng Python thay vì C++, giúp giảm độ phức tạp và tăng tốc độ phát triển.

- Phân tích dữ liệu: Python có nhiều thư viện mạnh như matplotlib, numpy, pandas để xử lý và trực quan hóa kết quả mô phỏng.
- Tự động hóa: Python hỗ trợ viết các script tự động hóa quá trình chạy mô phỏng, thu thập và xử lý kết quả, giúp tiết kiệm thời gian và tăng độ chính xác.

Trong đề tài này, Python được dùng để:

- Viết kịch bản mô phỏng mạng CSMA với các traffic TCP và UDP.
- Cấu hình các node, thiết bị, kênh truyền và ứng dụng mạng.
- Thu thập dữ liệu đầu ra từ NS-3 và xử lý bằng các thư viện Python để vẽ đồ thị và phân tích hiệu suất.

Các gói hỗ trợ: pandas, matplotlib.pyplot, seaborn và numpy.

4. Phân tích kết quả:

4.1. Tổng hợp kết quả mô phỏng từ NS-3:

[TCP TRAFFIC METRICS]		[TCP TRAFFIC METRICS]	
Total Packets Sent:	54913	Total Packets Sent:	4323
Total Packets Received:	109823	Total Packets Received:	7700
Total Bytes Received:	56229888 bytes	Total Bytes Received:	4296872 bytes
Packet Delivery Ratio:	199.99%	Packet Delivery Ratio:	178.12%
Throughput:	50.00 Mbps	Throughput:	3.82 Mbps
Active Duration:	8.997 seconds	Active Duration:	8.992 seconds
[UDP TRAFFIC METRICS]		[UDP TRAFFIC METRICS]	
Total Packets Sent:	54931	Total Packets Sent:	54931
Total Packets Received:	54927	Total Packets Received:	10283
Total Bytes Received:	56245248 bytes	Total Bytes Received:	10529792 bytes
Packet Delivery Ratio:	99.99%	Packet Delivery Ratio:	18.72%
Throughput:	50.00 Mbps	Throughput:	9.36 Mbps
Active Duration:	8.999 seconds	Active Duration:	8.997 seconds

Hình 7. Kết quả hiệu suất TCP và UDP từ NS-3 ở trường hợp 1 và trường hợp 2.

Từ Hình 7, với trường hợp đầu tiên (lí tưởng, cấu hình appRate cân bằng với dataRate là 100 Mbps và độ trễ 2μs):

- Về TCP:
 - o Tổng số gói gửi: 54.913
 - o Tổng số gói nhận: 109.823
 - o Tổng số byte nhận: 5.622.988.800 bytes
 - o PDR: 199.99% (do TCP phân mảnh/ACK nên số gói nhận lớn hơn số gói gửi)
 - o Thông lượng: 50.00 Mbps
 - o Thời gian hoạt động: 8.997 s
- Về UDP:
 - o Tổng số gói gửi: 54.931
 - o Tổng số gói nhận: 54.927
 - o Tổng số byte nhận: 5.625.424.800 bytes

- PDR: 99.99%
- Thông lượng: 50.00 Mbps
- Thời gian hoạt động: 8.999 s

Về trường hợp thứ hai (độ trễ cao, tăng delay lên 500 microsec nhưng giữ tải cân bằng):

- Về TCP:
 - Tổng số gói gửi: 4.323
 - Tổng số gói nhận: 7.700
 - Tổng số byte nhận: 4.296.872 bytes
 - PDR: 178.12%
 - Thông lượng: 3.82 Mbps
 - Thời gian hoạt động: 8.992 s
- Về UDP:
 - Tổng số gói gửi: 54.931
 - Tổng số gói nhận: 10.283
 - Tổng số byte nhận: 10.529.792 bytes
 - PDR: 18.72%
 - Thông lượng: 9.36 Mbps
 - Thời gian hoạt động: 8.997 s

[TCP TRAFFIC METRICS] Total Packets Sent: 76131 Total Packets Received: 152144 Total Bytes Received: 77933080 bytes Packet Delivery Ratio: 199.85% Throughput: 69.30 Mbps Active Duration: 8.997 seconds	[TCP TRAFFIC METRICS] Total Packets Sent: 4323 Total Packets Received: 7699 Total Bytes Received: 4296336 bytes Packet Delivery Ratio: 178.09% Throughput: 3.82 Mbps Active Duration: 8.992 seconds
[UDP TRAFFIC METRICS] Total Packets Sent: 219726 Total Packets Received: 101043 Total Bytes Received: 103468032 bytes Packet Delivery Ratio: 45.99% Throughput: 91.98 Mbps Active Duration: 8.999 seconds	[UDP TRAFFIC METRICS] Total Packets Sent: 219726 Total Packets Received: 10283 Total Bytes Received: 10529792 bytes Packet Delivery Ratio: 4.68% Throughput: 9.36 Mbps Active Duration: 8.997 seconds

Hình 8. Kết quả hiệu suất TCP và UDP từ NS-3 ở trường hợp 3 và trường hợp 4.

Từ Hình 8, với trường hợp thứ 3 (tắc nghẽn, tăng appRate lên 200Mbps làm quá tải nhưng vẫn giữ delay thấp):

- Về TCP:
 - Tổng số gói gửi: 76.131
 - Tổng số gói nhận: 152.144
 - Tổng số byte nhận: 77.993.080 bytes
 - PDR: 199.85% (do TCP phân mảnh/ACK nên số gói nhận lớn hơn số gói gửi)

- Thông lượng: 63.30 Mbps
- Thời gian hoạt động: 8.997 s
- Về UDP:
 - Tổng số gói gửi: 219.726
 - Tổng số gói nhận: 101.043
 - Tổng số byte nhận: 103.468.032 bytes
 - PDR: 45.99%
 - Thông lượng: 91.98 Mbps
 - Thời gian hoạt động: 8.999 s

Về trường hợp thứ 4 (cực đoan, cả appRate tăng và delay đều tăng quá mức):

- Về TCP:
 - Tổng số gói gửi: 4.323
 - Tổng số gói nhận: 7.699
 - Tổng số byte nhận: 4.296.336 bytes
 - PDR: 178.09%
 - Thông lượng: 3.82 Mbps
 - Thời gian hoạt động: 8.992 s
- Về UDP:
 - Tổng số gói gửi: 219.726
 - Tổng số gói nhận: 10.283
 - Tổng số byte nhận: 10.529.792 bytes
 - PDR: 4.68%
 - Thông lượng: 9.36 Mbps
 - Thời gian hoạt động: 8.997 s

Kết luận:

- Trường hợp 1 – Lý tưởng: Cả TCP và UDP đều đạt thông lượng gần 50 Mbps, PDR gần tuyệt đối → hiệu suất tối đa, mạng hoạt động ổn định đúng như kỳ vọng.
- Trường hợp 2 – Độ trễ cao: TCP tụt xuống 3.82 Mbps do RTT lớn, UDP đạt 9.36 Mbps nhưng PDR chỉ 18.72% → TCP bị giảm thông lượng, UDP mất gói nặng.
- Trường hợp 3 – Tắc nghẽn: TCP duy trì 63.3 Mbps với PDR gần 200%, trong khi UDP đạt 91.98 Mbps nhưng PDR chỉ 45.99% → UDP thông lượng cao nhưng rớt gói nhiều, TCP ổn định hơn.
- Trường hợp 4 – Cực đoan: TCP chỉ còn 3.82 Mbps, UDP chỉ còn 4.68% PDR → TCP bị ảnh hưởng nặng bởi delay + tải lớn, UDP gần như mất toàn bộ dữ liệu.

Về file dữ liệu, toàn bộ dữ liệu được ghi vào các file: csma_metrics_1.csv, csma_metrics_2.csv,... . Đây là các file tổng hợp chứa toàn bộ thông tin cần thiết để phân tích hiệu suất hệ thống.

	A	B	C	D	E	F	G	H	I	J	K
1	Time	TCP_Packets_Sent	TCP_Packets_Received	TCP_Bytes_Received	TCP_Throughput_Mbps	TCP_PDR_%	UDP_Packets_Sent	UDP_Packets_Received	UDP_Bytes_Received	UDP_Throughput_Mbps	UDP_PDR_%
2	0.1	0	0	0	0	0	0	0	0	0	0
3	0.2	0	0	0	0	0	0	0	0	0	0
4	0.3	0	0	0	0	0	0	0	0	0	0
5	0.4	0	0	0	0	0	0	0	0	0	0
6	0.5	0	0	0	0	0	0	0	0	0	0
7	0.6	0	0	0	0	0	0	0	0	0	0
8	0.7	0	0	0	0	0	0	0	0	0	0
9	0.8	0	0	0	0	0	0	0	0	0	0
10	0.9	0	0	0	0	0	0	0	0	0	0
11	1	1	0	0	0	0	0	0	0	0	0
12	1.1	2375	2256	922392	0.737914	94.9895	244	243	255636	0.204509	99.5902
13	1.2	4721	4618	1921040	1.53683	97.8183	488	488	513376	0.410701	100
14	1.3	7345	7130	2917992	2.33439	97.0728	732	732	770064	0.616051	100
15	1.4	9838	9616	3893576	3.11486	97.7434	976	976	1026752	0.821402	100
16	1.5	12334	12127	4886356	3.90908	98.3217	1220	1220	1283440	1.02675	100
17	1.6	14795	14667	5912388	4.72991	99.1348	1464	1464	1540128	1.2321	100
18	1.7	17522	17280	6901256	5.521	98.6189	1708	1708	1796816	1.43745	100
19	1.8	20013	19826	7921456	6.33716	99.0656	1953	1952	2053504	1.6428	99.9488
20	1.9	22644	22431	8937068	7.14965	99.0594	2197	2197	2311244	1.849	100

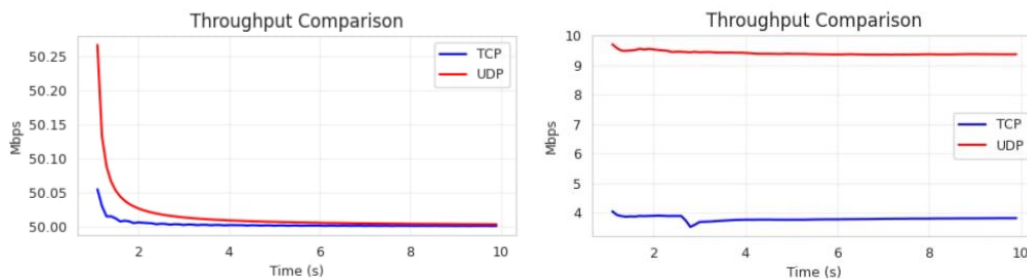
Hình 9. Kết quả trong các file csma_metrics.

Thông tin trong file csv ở Hình 9 này bao gồm các trường dữ liệu sau:

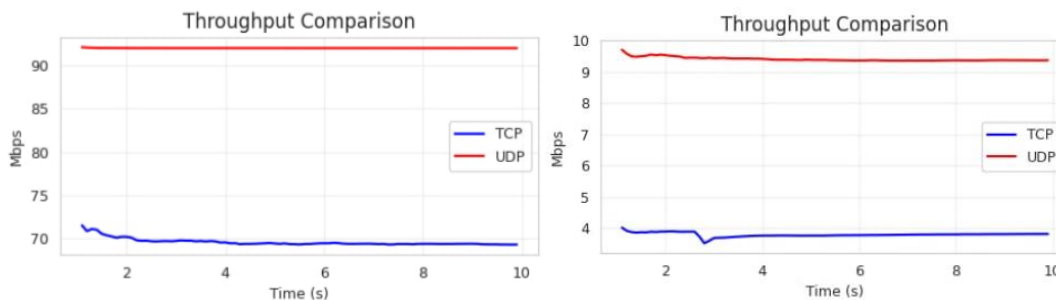
- Time (s): thời gian mô phỏng tại mỗi thời điểm ghi.
- TcpThroughput (Mbps): thông lượng TCP tức thời.
- UdpThroughput (Mbps): thông lượng UDP tức thời.
- TcpPacketsSent / TcpPacketsReceived: số gói TCP đã gửi/nhận.
- UdpPacketsSent / UdpPacketsReceived: số gói UDP đã gửi/nhận.
- TcpLoss / UdpLoss: số lượng gói tổn thất tại thời điểm tương ứng.
- TcpDataReceived / UdpDataReceived: tổng lượng dữ liệu tích lũy tính bằng byte.

File csma_metrics.csv là cơ sở để sinh toàn bộ đồ thị: thông lượng, PDR, số gói tích lũy, dữ liệu tích lũy, hiệu suất, và bảng tổng hợp.

4.2. Đồ thị thông lượng TCP và UDP theo thời gian:



Hình 10. Đồ thị thông lượng TCP và UDP theo trường hợp 1 và trường hợp 2.

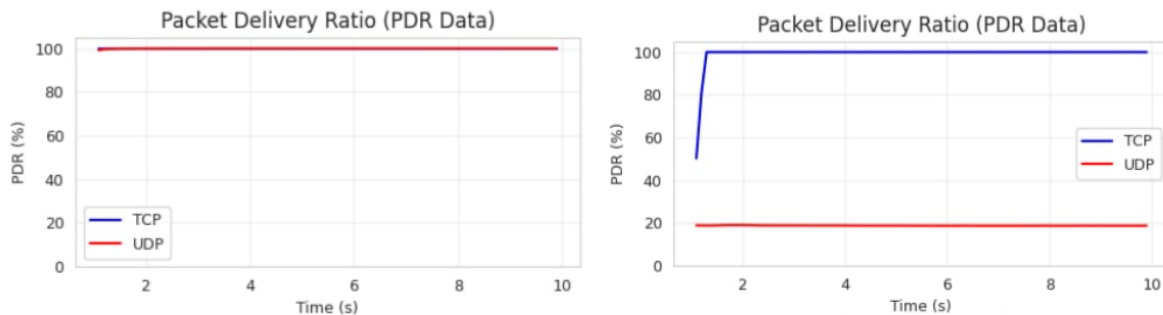


Hình 11. Đồ thị thông lượng TCP và UDP theo trường hợp 3 và trường hợp 4.

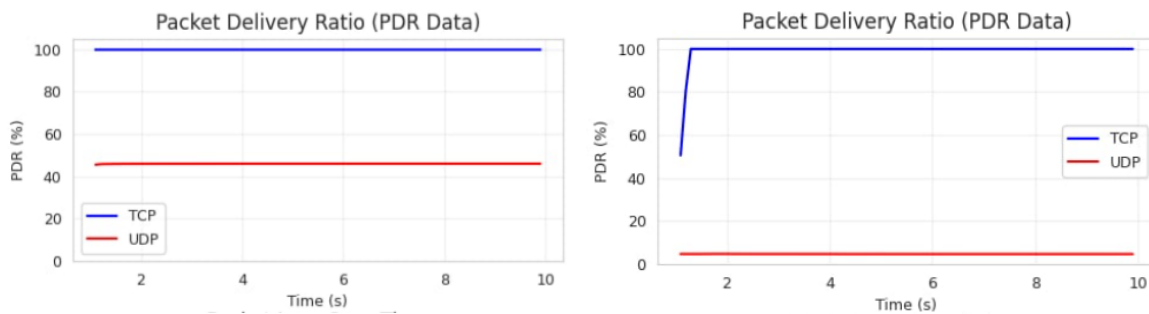
Từ Hình 10 và Hình 11:

- Trường hợp 1 – Lý tưởng: TCP & UDP tiệm cận khoảng 50 Mbps, khai thác tối đa băng thông, hiệu quả tương đương.
- Trường hợp 2 – Delay cao: TCP giảm mạnh ngang 4 tới Mbps do RTT lớn; UDP duy trì ngang 9.5 Mbps → TCP nhạy cảm với độ trễ.
- Trường hợp 3 – Tải cao: UDP ~92 Mbps nhưng mất nhiều gói; TCP giảm còn ~69.6 Mbps nhưng giữ PDR cao → UDP ổn về tốc độ nhưng TCP ổn về độ tin cậy.
- Trường hợp 4 – Cực đoan: TCP thấp ngang 3.8 Mbps, UDP ngang ~9.4 Mbps.

4.3. Phân tích đồ thị Packet Delivery Ratio (PDR) theo thời gian:



Hình 12. Tỷ lệ gói được giao thành công của TCP và UDP theo trường hợp 1 và 2.



Hình 13. Tỷ lệ gói được giao thành công của TCP và UDP theo trường hợp 3 và 4.

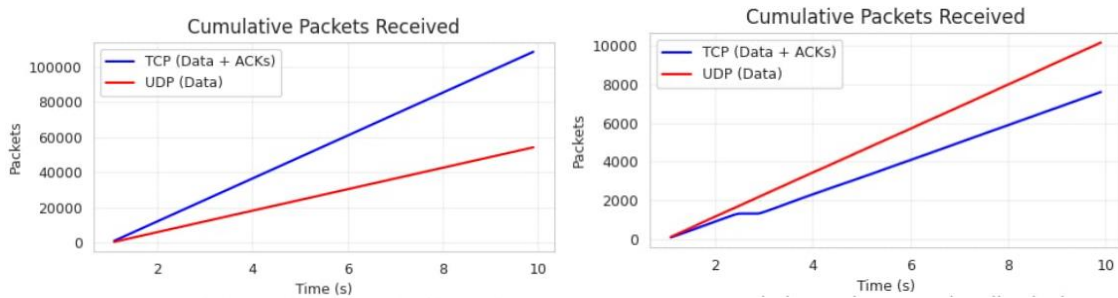
Công thức PDR (tỷ lệ gói được giao thành công) có dạng:

$$\text{PDR} = \frac{\text{Số lượng gói nhận được thành công}}{\text{Tổng số gói được gửi đi}} \cdot 100\%$$

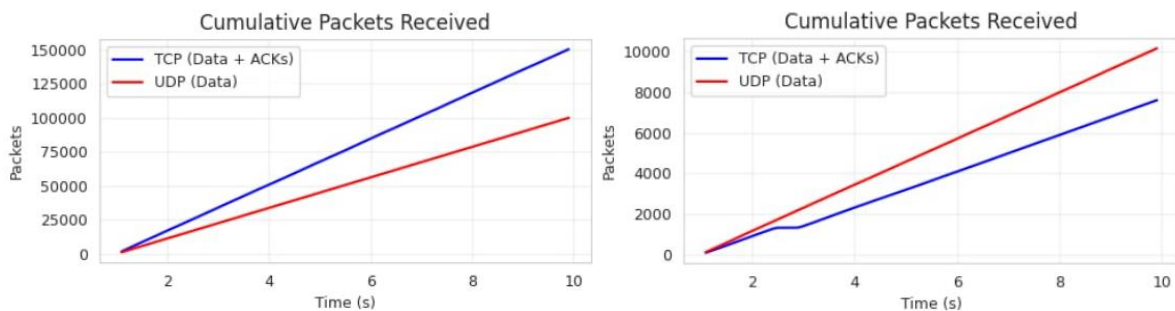
Từ Hình 12 và Hình 13:

- Trường hợp 1 – Lý tưởng: TCP và UDP có PDR 100%, mạng không tắc nghẽn, cả hai giao thức đảm bảo giao gói.
- Trường hợp 2 – Delay cao: TCP có PDR tăng dần lên 100%, UDP giảm khoảng 20% → UDP nhạy cảm với độ trễ.
- Trường hợp 3 – Tải cao: TCP có PDR gần 100%, UDP giảm còn 45.99% → UDP mất nhiều gói do tắc nghẽn.
- Trường hợp 4 – Cực đoan: TCP có PDR lên gần 100%, nhưng UDP chỉ 4.68% → UDP gần như vô dụng, TCP vẫn ưu tiên độ tin cậy.

4.4. Phân tích đồ thị số gói nhận tích lũy theo thời gian:



Hình 14. Số gói nhận tích lũy của TCP và UDP theo trường hợp 1 và 2.

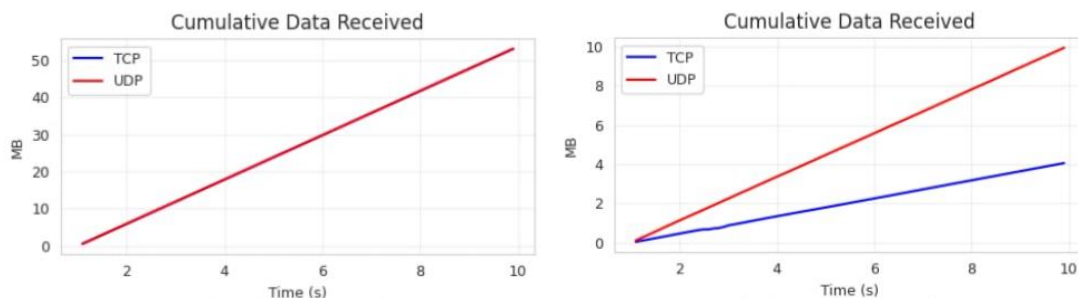


Hình 15. Số gói nhận tích lũy của TCP và UDP theo trường hợp 3 và 4.

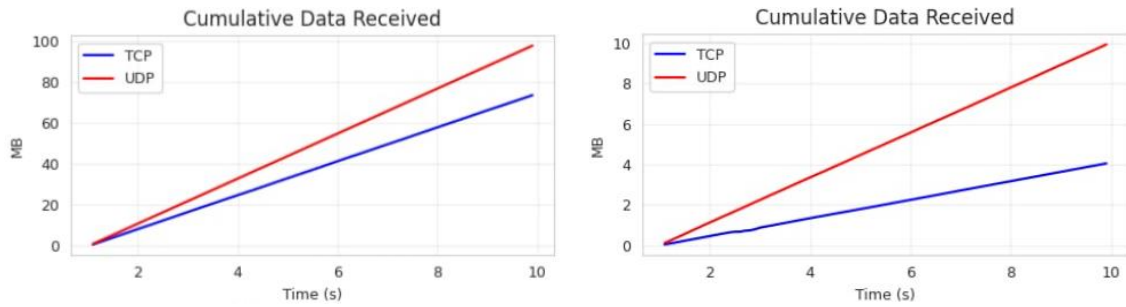
Từ Hình 14 và Hình 15:

- Trường hợp 1 – Lý tưởng: TCP dốc hơn UDP gấp đôi do ACKs → chi phí TCP cao nhưng hiệu quả tương đương.
- Trường hợp 2 – Delay cao: TCP dốc thấp, UDP dốc cao → UDP gửi nhanh.
- Trường hợp 3 – Tải cao: UDP dốc hơn TCP → TCP giữ tốc độ ổn định.
- Trường hợp 4 – Cực đoan: TCP dốc rất thấp; UDP dốc cao, nhưng do PDR chỉ ngang 4.68% → hầu hết gói UDP thất bại.

4.5. Phân tích đồ thị dữ liệu nhận tích lũy theo thời gian:



Hình 16. Đồ thị dữ liệu nhận tích lũy của TCP và UDP theo trường hợp 1 và 2.

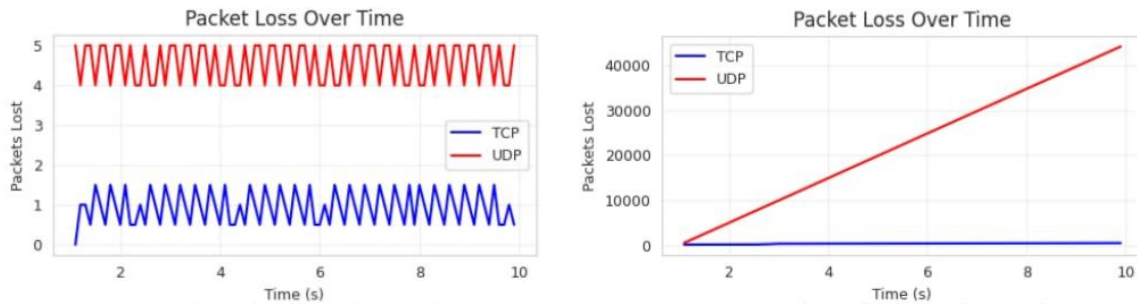


Hình 17. Đồ thị dữ liệu nhận tích lũy của TCP và UDP theo trường hợp 3 và 4.

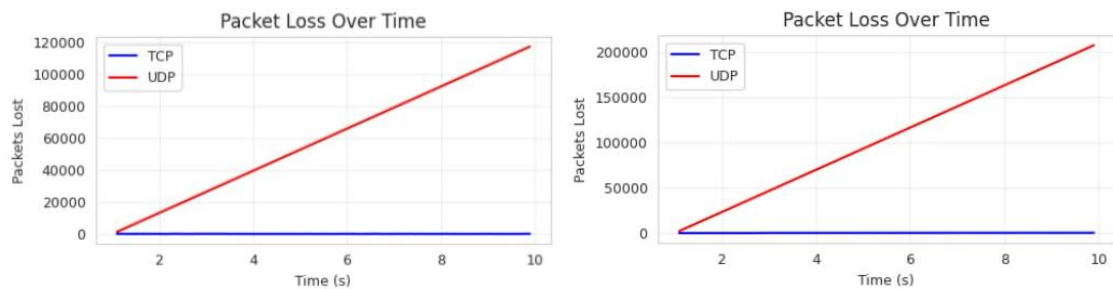
Từ Hình 16 và Hình 17:

- Trường hợp 1 – Lý tưởng: TCP dốc ngang UDP → nhận đủ gói
- Trường hợp 2 – Delay cao: UDP dốc cao hơn TCP → TCP chậm do chờ ACKs.
- Trường hợp 3 – Tải cao: UDP dốc hơn TCP nhưng mất ~54% gói (PDR 45.99%); TCP ổn định và đáng tin cậy hơn.
- Trường hợp 4 – Cực đoan: UDP dốc cao hơn TCP nhưng PDR cực thấp (~4.68%), TCP chậm nhưng giữ dữ liệu hiệu quả.

4.6. Phân tích đồ thị số gói bị mất theo thời gian:



Hình 18. Đồ thị số gói bị mất của TCP và UDP theo trường hợp 1 và 2.



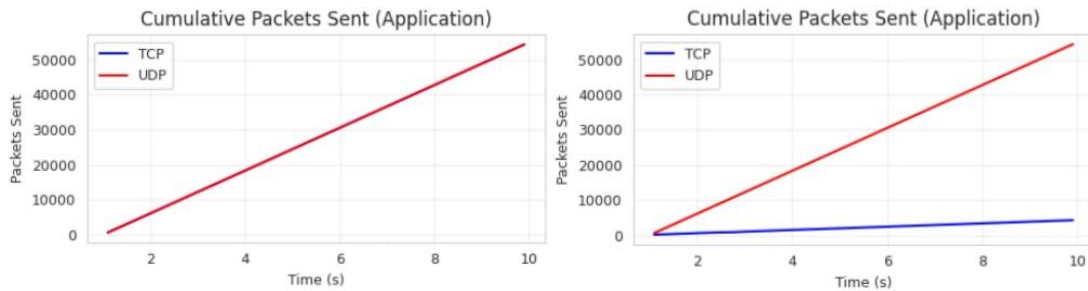
Hình 19. Đồ thị số gói bị mất của TCP và UDP theo trường hợp 3 và 4.

Từ Hình 18 và Hình 19:

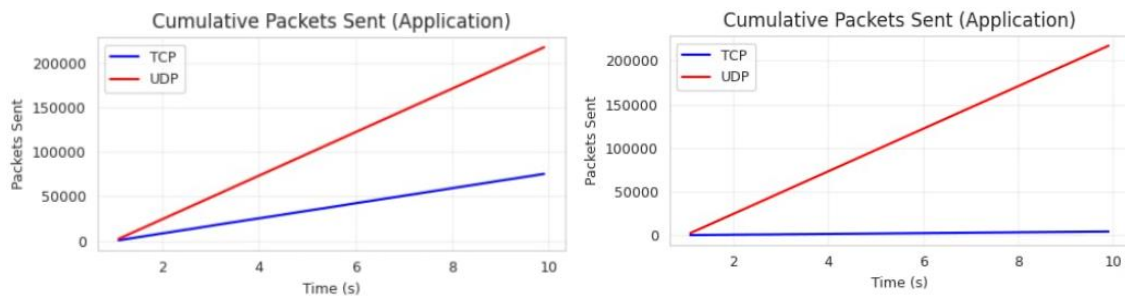
- Trường hợp 1: TCP có số gói mất dao động nhẹ và thấp, thể hiện cơ chế kiểm soát tắc nghẽn hoạt động tốt; trong khi UDP mất gói ổn định ở mức cao hơn do không có cơ chế điều chỉnh tốc độ gửi.
- Trường hợp 2: UDP mất gói tăng tuyến tính rất nhanh khi tải tăng, còn TCP vẫn duy trì mức mất gói rất thấp nhờ tự giảm tốc khi mạng quá tải.

- Trường hợp 3: Khi lưu lượng tăng mạnh hơn, UDP tiếp tục mất gói với tốc độ tăng rõ rệt theo thời gian, còn TCP vẫn gần như giữ ổn định nhờ thích ứng lưu lượng.
- Trường hợp 4: UDP đạt mức mất gói cao nhất trong cả 4 trường hợp, phản ánh nhược điểm khi truyền trên mạng tắc nghẽn; TCP vẫn giữ mức mất gói rất nhỏ nhờ kiểm soát tắc nghẽn hiệu quả.

4.7. Phân tích đồ thị số gói gửi tích lũy theo thời gian:



Hình 20. Đồ thị số gói gửi tích lũy của TCP và UDP theo trường hợp 1 và 2.

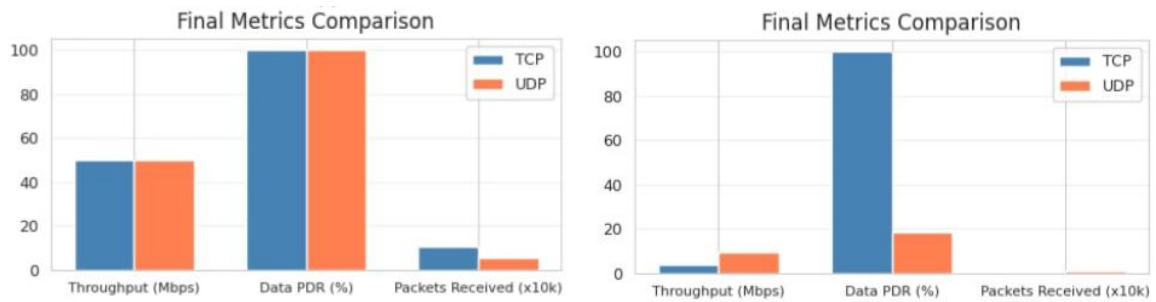


Hình 21. Đồ thị số gói gửi tích lũy của TCP và UDP theo trường hợp 3 và 4.

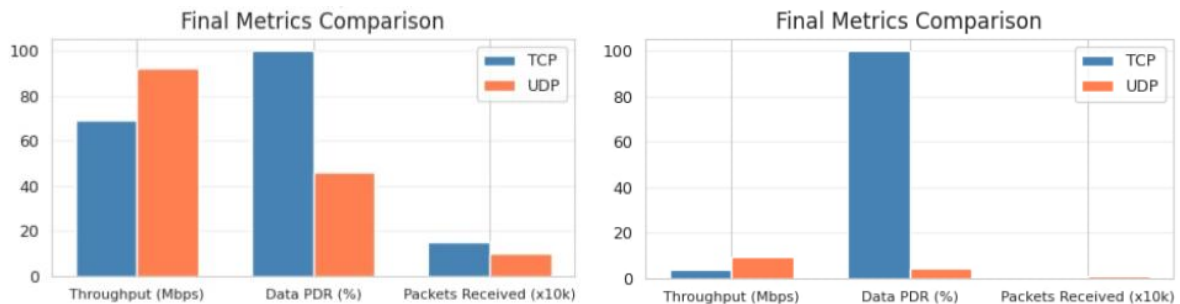
Từ Hình 20 và Hình 21:

- Trường hợp 1: UDP gửi số gói tăng nhanh và đều; TCP gửi chậm hơn đáng kể do bị giới hạn bởi cơ chế kiểm soát tắc nghẽn.
- Trường hợp 2: UDP tiếp tục duy trì tốc độ gửi cao; TCP có tăng nhưng vẫn thấp hơn nhiều so với UDP.
- Trường hợp 3: Khi tải lớn hơn, UDP gửi gói rất nhanh; TCP tăng đều nhưng với tốc độ thấp, thể hiện đặc tính điều chỉnh thận trọng.
- Trường hợp 4: UDP đạt lượng gói gửi tích lũy cao nhất trong bốn trường hợp; TCP vẫn tăng chậm và ổn định theo thời gian.

4.8. Phân tích đồ thị so sánh các chỉ số tổng hợp:



Hình 22. Đồ thị so sánh các chỉ số tổng hợp giữa TCP và UDP theo trường hợp 1 và 2.



Hình 23. Đồ thị so sánh các chỉ số tổng hợp giữa TCP và UDP theo trường hợp 3 và 4.

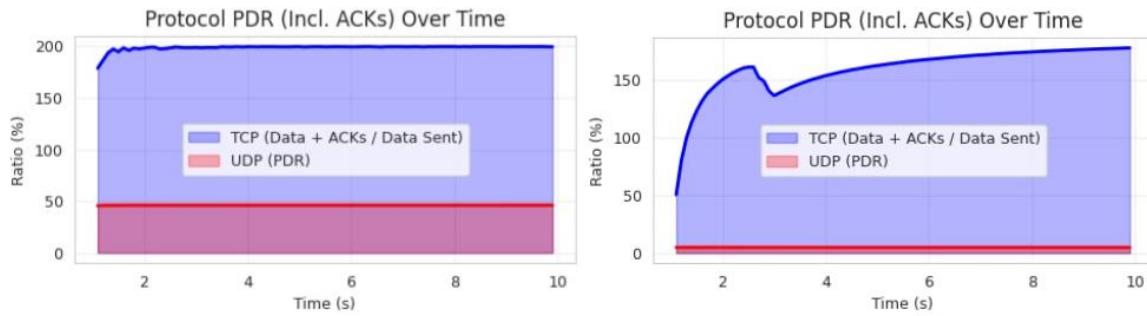
Từ Hình 22 và Hình 23:

- Trường hợp 1: TCP và UDP có throughput gần tương đương; PDR đều đạt gần 100%; số gói nhận của TCP cao hơn nhẹ nhờ kiểm soát truyền tốt.
- Trường hợp 2: UDP có throughput cao hơn TCP, nhưng PDR thấp hơn rõ rệt; TCP giữ PDR gần như tối đa dù throughput thấp.
- Trường hợp 3: UDP đạt throughput vượt trội nhưng PDR giảm mạnh; TCP có throughput trung bình nhưng PDR vẫn giữ ở mức gần 100%.
- Trường hợp 4: UDP throughput cao nhất nhưng PDR rất thấp; TCP giữ PDR tối đa dù throughput thấp, cho thấy ưu tiên độ tin cậy hơn tốc độ.

4.9. Phân tích hiệu suất của giao thức theo thời gian:



Hình 24. Hiệu suất giao thức theo thời gian của TCP và UDP theo trường hợp 1 và 2.



Hình 25. Hiệu suất giao thức theo thời gian của TCP và UDP theo trường hợp 3 và 4. Trường hợp 1: TCP đạt hiệu suất 200% (tỷ lệ Data : ACK = 1:2), phản ánh chi phí điều khiển cơ bản; UDP ổn định ở ~100%.

Trường hợp 2: TCP giảm còn ~150% do độ trễ và truyền lại làm tăng overhead; UDP vẫn ~100%.

Trường hợp 3: TCP giữ 200% dù tắc nghẽn lớn; UDP vẫn ~100%, không thể hiện PDR thấp. Riêng với trường hợp 4: TCP giảm mạnh xuống ~140% vì overhead phục hồi quá lớn; UDP giữ ~100%, không phản ánh mất dữ liệu.

4.10. Bảng tổng hợp đánh giá các chỉ số hiệu suất:

Bảng 1. Bảng các chỉ số hiệu suất của TCP từ 4 trường hợp gộp lại:

Trường hợp	Avg Throughput(Mbps)	Final PDR (%)	Total Packets Sent	Total Packets Received	Total Data(MB)
Trường hợp 1	50.00	100.00	54.302	108.603	53.03
Trường hợp 2	3.8	100.00	4.275	7.608	4.05
Trường hợp 3	69.58	100.00	75.324	150.425	73.48
Trường hợp 4	3.8	100.00	4.274	7.607	4.05

Bảng 1. Bảng các chỉ số hiệu suất của UDP từ 4 trường hợp gộp lại.

Trường hợp	Avg Throughput(Mbps)	Final PDR (%)	Total Packets Sent	Total Packets Received	Total Data(MB)
Trường hợp 1	50.02	99.99	54.321	54.316	53.04
Trường hợp 2	9.40	18.71	54.321	10.165	9.93
Trường hợp 3	91.99	45.99	217.825	99.920	97.58
Trường hợp 4	9.40	4.68	217.825	10.165	9.93

5. Kết luận:

Hiệu suất tối đa khi tải thấp, delay thấp: TCP và UDP đạt 50Mbps, PDR gần 100%, cho thấy overhead của TCP không đáng kể trong điều kiện thuận lợi.

Độ tin cậy: TCP duy trì PDR 100% ở mọi kịch bản, tự điều chỉnh thông lượng khi quá tải, nhưng thông lượng bị phạt kép xuống 3.80 Mbps ở trường hợp 4.

Tốc độ và khả năng mất gói: UDP đạt tốc độ cao (91.99 Mbps ở trường hợp 3) nhưng mất gói nghiêm trọng, sụp đổ khi quá tải (PDR chỉ 4.68% ở trường hợp 4).

Ảnh hưởng độ trễ: TCP bị giảm thông lượng mạnh dưới delay cao (3.8 Mbps ở trường hợp 2 và 4), trong khi phía UDP rất bất ổn (PDR 18.71% ở trường hợp 2).

Các kết quả thu được phù hợp với lý thuyết: khi delay/tải cao thì throughput TCP giảm và UDP mất gói mạnh. Điều này xác nhận mô hình CSMA trong NS-3 hoạt động đúng hành vi của Ethernet.

Do đó, trong mạng CSMA, TCP đảm bảo độ tin cậy cao, duy trì PDR 100% nhưng bị giảm thông lượng khi quá tải hoặc delay cao. UDP ưu tiên tốc độ, đạt thông lượng lớn nhưng mất gói nghiêm trọng và sụp đổ khi mạng xấu. Khi tải thấp và delay thấp, cả hai đều đạt hiệu suất tối đa, cho thấy overhead của TCP không đáng kể.

Tài liệu tham khảo:

- [1] GeeksforGeeks, *Carrier Sense Multiple Access (CSMA)*, GeeksforGeeks, 2025.
- [2] Wikipedia tiếng Việt, *Carrier-sense multiple access with collision detection*, Wikipedia, năm 2023.
- [3] ns-3 Project, *CSMA NetDevice — Model Library*, ns-3 Documentation, 2025.