

NGUYỄN KHẮC LỢI

HỆ THỐNG THÔNG TIN

2022

**BỘ GIÁO DỤC
VÀ ĐÀO TẠO**

**VIỆN HÀN LÂM
KHOA HỌC VÀ CÔNG NGHỆ VN**

HỌC VIỆN KHOA HỌC VÀ CÔNG NGHỆ



Nguyễn Khắc Lợi

**NGHIÊN CỨU GIẢI PHÁP ĐÁNH GIÁ
CHẤT LƯỢNG DỊCH VỤ ĐA PHƯƠNG TIỆN
TRÊN MẠNG KHÔNG DÂY SỬ DỤNG MÔ PHỎNG**

LUẬN VĂN THẠC SĨ HỆ THỐNG THÔNG TIN

Hà Nội – 2022

**BỘ GIÁO DỤC
VÀ ĐÀO TẠO**

**VIỆN HÀN LÂM
KHOA HỌC VÀ CÔNG NGHỆ VN**

HỌC VIỆN KHOA HỌC VÀ CÔNG NGHỆ



Nguyễn Khắc Lợi

**NGHIÊN CỨU GIẢI PHÁP ĐÁNH GIÁ
CHẤT LƯỢNG DỊCH VỤ ĐA PHƯƠNG TIỆN
TRÊN MẠNG KHÔNG DÂY SỬ DỤNG MÔ PHỎNG**

Chuyên ngành : Hệ thống thông tin
Mã số: 8480104

LUẬN VĂN THẠC SĨ NGÀNH MÁY TÍNH

NGƯỜI HƯỚNG DẪN KHOA HỌC :
TS. Ngô Hải Anh

Hà Nội – 2022

LỜI CAM ĐOAN

Tôi xin cam đoan đề tài nghiên cứu trong luận văn này là công trình nghiên cứu của tôi dựa trên những tài liệu, số liệu do chính tôi tự tìm hiểu và nghiên cứu. Chính vì vậy, các kết quả nghiên cứu đảm bảo trung thực và khách quan nhất. Đồng thời, kết quả này chưa từng xuất hiện trong bất cứ một nghiên cứu nào. Các số liệu, kết quả nêu trong luận văn là trung thực nếu sai tôi hoàn chịu trách nhiệm.

TÁC GIẢ LUẬN VĂN

Nguyễn Khắc Lợi

LỜI CẢM ƠN

Với sự chỉ dẫn tận tình và hỗ trợ của giáo viên tại Khoa Công nghệ Thông tin & Viễn thông, Học viện Khoa học và Công nghệ cũng như sự hỗ trợ từ bạn bè và đồng nghiệp, đặc biệt là sự hướng dẫn tận tình của TS. Ngô Hải Anh cộng thêm những nỗ lực, cố gắng bản thân, cho đến nay đề tài đã được hoàn thành. Trong quá trình hoàn thiện làm luận văn, học viên cũng đã cố gắng rất nhiều, nhưng do kiến thức và kinh nghiệm còn hạn hẹp, khó tránh khỏi những sai sót, rất mong nhận được những góp ý của các thầy và các cô để luận văn hoàn thiện hơn.

Tôi xin chân thành cảm ơn.

MỤC LỤC

Trang

Trang phụ bìa	
Bản cam đoan	
Nhiệm vụ luận văn	
Mục lục.....	
Tóm tắt luận văn.....	
Danh mục các ký hiệu, các chữ viết tắt	
Danh mục các hình vẽ	
MỞ ĐẦU	1
Chương 1. TỔNG QUAN VỀ MẠNG KHÔNG DÂY VÀ CHẤT LƯỢNG DỊCH VỤ.....	3
1.1. KHÁI NIỆM VỀ MẠNG KHÔNG DÂY	3
1.2. MẠNG KHÔNG DÂY THEO HỌ TIÊU CHUẨN IEEE 802.11	3
1.3. CHẤT LƯỢNG DỊCH VỤ MẠNG KHÔNG DÂY.....	4
1.3.1. Một số nghiên cứu liên quan.....	4
1.3.2. Các vấn đề chính của chất lượng dịch vụ mạng	7
Chương 2. NGHIÊN CỨU VỀ CHẤT LƯỢNG DỊCH VỤ DỮ LIỆU ĐA PHƯƠNG TIỆN TRÊN MẠNG KHÔNG DÂY	10
2.1. PHÂN TÍCH MỘT SỐ KỸ THUẬT TẦNG MAC CỦA MẠNG KHÔNG DÂY.....	10
2.2. CHUẨN MẠNG KHÔNG DÂY IEEE 802.11E CHO DỮ LIỆU ĐA PHƯƠNG TIỆN.....	13
2.3. MỘT SỐ LOẠI MẠNG DI ĐỘNG CÓ SỬ DỤNG CHUẨN 802.11 .	17
2.3.1. Mạng không dây di động (MANET)	17
2.3.2. Mạng không dây di động trong xe cộ (VANET)	18

Chương 3. ĐÁNH GIÁ CHẤT LƯỢNG DỊCH VỤ DỮ LIỆU ĐA PHƯƠNG TIỆN TRÊN MẠNG KHÔNG DÂY	21
3.1. GIỚI THIỆU VỀ MÔ PHÒNG MẠNG	21
3.2. CÀI ĐẶT, THIẾT LẬP, THỰC THI VÀ XỬ LÝ KẾT QUẢ MÔ PHÒNG TRONG NS-2.....	22
3.3. MÔ PHÒNG VÀ ĐÁNH GIÁ MỘT SỐ KỊCH BẢN MẠNG KHÔNG DÂY	34
3.3.1. Kịch bản thứ 1: mạng không dây ad hoc đơn chặng.....	36
3.3.2. Kịch bản thứ 2: mạng không dây ad hoc đa chặng	41
3.3.3. Kịch bản thứ 3: mạng không dây cho xe cộ.....	45
3.3.4. Kịch bản thứ 4: mạng không dây cho dữ liệu đa phương tiện.....	53
KẾT LUẬN VÀ KIẾN NGHỊ	61
DANH MỤC TÀI LIỆU THAM KHẢO.....	63

DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ CÁI VIẾT TẮT

- AC (Access Category): Phân loại truy cập
- AP (Access Point): điểm truy cập không dây
- BE (Best-Effort): Nỗ lực tối đa (một loại dữ liệu đa phương tiện)
- BEB (Binary Exponential Back-off): Quay lui hàm mũ nhị phân
- BK (Background): Nền (một loại dữ liệu đa phương tiện)
- CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance): Đa truy cập sử dụng sóng mang có tránh xung đột
- CW (Contention Window): Cửa sổ tương tranh
- DCF (Distributed Coordination Function): Chức năng cộng tác phân tán
- EDCA (Enhanced Distributed Channel Access): Truy cập kênh truyền phân tán nâng cao
- FIFO (First In First Out): Vào trước ra trước (một kiểu cấu trúc hàng đợi)
- HCF (Hybrid Coordination Function): Chức năng cộng tác lai
- IEEE (Institute of Electrical and Electronics Engineers): Hội kỹ sư điện và điện tử (một tổ chức định tiêu chuẩn)
- ITU (International Telecommunication Union): Liên minh Viễn thông quốc tế
- MAC (Media Access Control): Điều khiển truy cập môi trường (một phân-tầng trong mô hình phân tầng mạng)
- MANET (Mobile Ad-hoc NETWORK) : Mạng không dây di động phi cấu trúc
- QoE (Quality of Experience): Chất lượng trải nghiệm
- QoS (Quality of Service): Chất lượng dịch vụ
- PHY (PHYSical): tầng Vật lý (trong mô hình phân tầng mạng)
- PQ (Priority Queue): Hàng đợi ưu tiên
- VANET (Vehicular Ad-hoc NETWORK) : Mạng không dây di động trong xe cộ
- VI (Video): Video (một loại dữ liệu đa phương tiện)
- VO (Voice): âm thanh (một loại dữ liệu đa phương tiện)
- WLAN (Wireless Local Area Network): mạng không dây cục bộ

DANH MỤC CÁC BẢNG

Bảng 2. 2. Độ ưu tiên và các mức truy cập	15
Bảng 2. 3. Các tham số EDCA mặc định	17

DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

Hình 1. 1. Giới thiệu về QoS.....	8
Hình 2. 1. Mô hình mạng Ad-hoc	11
Hình 2. 2. Mô hình cơ bản mạng MANET	18
Hình 2. 3. Mô hình mạng VANET.....	19
Hình 2. 4. Nhu cầu định tuyến theo vị trí.....	20
Hình 3. 1. Ubuntu Software Center.....	23
Hình 3. 2. Cửa sổ tìm kiếm trong Software Center.....	23
Hình 3. 3. Tìm gói ns2 trên thư viện phần mềm của Ubuntu	24
Hình 3. 4. Tiến trình cài đặt phần mềm	24
Hình 3. 5. Kiểm tra các gói cài đặt.....	25
Hình 3. 6. Luồng các sự kiện khi chạy mô phỏng trong NS-2	26
Hình 3. 7. Kịch bản mô phỏng mạng đơn giản với NS-2	26
Hình 3. 8. Liên kết (link) giữa các node trong NS2.....	31
Hình 3. 9. Kết quả phân tích	39
Hình 3. 10. Đồ thị biến thiên theo thời gian của throughput	40
Hình 3. 11. Kết quả mô phỏng 1 qua NAM.....	41
Hình 3. 12. Kết quả phân tích	44
Hình 3. 13. Đồ thị biến thiên của throughput theo thời gian	44
Hình 3. 14. Gói tin bị drop	45
Hình 3. 15. Kết quả mô phỏng 2 qua NAM.....	45
Hình 3. 16. Mô hình VANET cơ bản trên NS2	48
Hình 3. 17. Chạy mô phỏng với 2 xe đang di chuyển trên NS2	49
Hình 3. 18. Một đoạn code mẫu của file perl.....	49
Hình 3. 19. Kết quả thu thập từ NS2 qua perl.....	50
Hình 3. 20. Một đoạn code của file awk	50
Hình 3. 21. Phân tích kết quả từ NS2 qua awk	51
Hình 3. 22. Biến thiên throughput data nhận	51
Hình 3. 23. Biến thiên throughput Data gửi.....	51
Hình 3. 24. Đồ trễ gói tin theo thời gian	52
Hình 3. 25. Độ trễ gói tin theo tốc độ xe.....	52
Hình 3. 26. Sự biến thiên thông lượng theo thời gian trong 802.11e	60

MỞ ĐẦU

Với sự phát triển nhanh chóng của Internet và viễn thông hiện nay, những năm gần đây mô hình cung cấp dịch vụ giải trí đa phương tiện trên không gian mạng đã trở nên phổ biến. Một công ty có thể đưa ra rất nhiều gói dịch vụ phù hợp với người dùng tại một quốc gia mà gần như không cần có trụ sở, hạ tầng mạng và lưu trữ dữ liệu tại quốc gia đó, ví dụ như các dịch vụ: nhắn tin (instant message), gọi điện (audio/video call), giải trí (xem phim, nghe nhạc, thu/phát trực tuyến) của các công ty đa quốc gia hiện nay gần như chỉ cần người dùng có kết nối mạng là có thể sử dụng được. Tình huống này đặt ra nhiều thách thức trong việc quản lý nội dung, phân phối lợi nhuận, chủ quyền trên mạng... của mỗi quốc gia.

Hiện nay, các công nghệ liên quan đến mạng máy tính có tốc độ phát triển nhanh chóng, đặc biệt với xu thế người dùng di động hiện nay, mạng không dây đóng vai trò ngày càng quan trọng với đa số người dùng và dần chiếm tỷ lệ lớn hơn rất nhiều so với loại hình mạng có dây. Các loại hình thức truyền dữ liệu cũng có sự đa dạng theo những nhu cầu khác nhau của người dùng. Truyền thông dữ liệu đa phương tiện (multimedia data) là một vấn đề thời sự vì các loại dữ liệu như audio, video đòi hỏi các tiêu chí đa dạng về độ nén, chất lượng hình ảnh, yêu cầu chất lượng mạng... nhằm thích nghi với các dịch vụ được cung cấp cũng như nhu cầu sử dụng khác nhau của người dùng

Các dịch vụ dữ liệu đa phương tiện trên mạng không dây cần có các tiêu chí đánh giá khác nhau về chất lượng dịch vụ, do vậy đề tài chọn tập trung nghiên cứu về *chất lượng dịch vụ* (Quality of Service – QoS). Do việc đánh giá hiệu năng và chất lượng cũng như các tiêu chí kỹ thuật của mạng không dây là khá phức tạp do bản chất di động cũng như cần nhiều thiết bị mới hình thành nên một mạng, luận văn chọn sử dụng phương pháp đánh giá thông qua bộ công cụ phần mềm *mô phỏng mạng* (network simulator) nhằm

đánh giá chất lượng dịch vụ trên một số cấu hình kịch bản (topology) mạng không dây khác nhau.

Với nội dung khái quát vừa nêu, cấu trúc luận văn “**NGHIÊN CỨU GIẢI PHÁP ĐÁNH GIÁ CHẤT LƯỢNG DỊCH VỤ ĐA PHƯƠNG TIỆN TRÊN MẠNG KHÔNG DÂY SỬ DỤNG MÔ PHỎNG**” gồm: phần mở đầu, chương 1, 2 và 3, phần kết luận và kiến nghị, tài liệu tham khảo.

Chương 1 – Tổng quan về mạng không dây và chất lượng dịch vụ mạng không dây.

Chương 2 – Nghiên cứu về chất lượng dịch vụ (QoS) với dữ liệu đa phương tiện trên mạng không dây.

Chương 3 - Đánh giá QoS dữ liệu đa phương tiện trên mạng không dây theo một số kịch bản mô phỏng.

Chương 1. TỔNG QUAN VỀ MẠNG KHÔNG DÂY VÀ CHẤT LƯỢNG DỊCH VỤ

1.1. KHÁI NIỆM VỀ MẠNG KHÔNG DÂY

Mạng không dây hiểu đơn giản là mạng máy tính, hoặc mạng điện thoại sử dụng sóng radio làm sóng truyền dẫn hay hoạt động ở tầng vật lý (PHY) trong mô hình phân tầng mạng. Như vậy, có thể hiểu một *mạng không dây* là bao gồm nhiều máy tính được kết nối với nhau thông qua các kết nối dữ liệu không dây giữa các nút mạng. Loại hình mạng này được dùng trong nhiều phạm vi sử dụng khác nhau, từ cấp hộ gia đình, đến cấp doanh nghiệp, hoặc các cơ sở kinh doanh cỡ nhỏ, hoặc vừa và cỡ lớn,... mọi phạm vi sử dụng đều cần nối mạng Internet hoặc mạng nội bộ nhưng không muốn sử dụng nhiều dây cáp mạng để kết nối. Nhìn chung thì các cấp độ mạng không dây có thể quản lý theo nhiều cách và qua nhiều cấp: với các mạng viễn thông 3G/4G, chúng được quản lý bởi hạ tầng của các hãng viễn thông hoặc nhà cung cấp dịch vụ Internet (ISP), những hệ thống này thường được đặt tập trung hoặc rời rạc tại những cơ sở lưu trữ của các nhà mạng; còn với mạng không dây Wi-Fi, chúng vẫn cần một hạ tầng kết nối có dây (cáp mạng) để kết nối tới các điểm sử dụng dịch vụ (gia đình, công ty, văn phòng,...) và kết nối không dây thực sự mới được triển khai tại các địa điểm đó thông qua hai cách triển khai chính là có cơ sở hạ tầng (ví dụ dùng các điểm truy cập – AP) hoặc không có cơ sở hạ tầng – gọi là mạng tự học – Ad hoc, hoặc mạng không cấu trúc. Phần tiếp theo sẽ trình bày rõ hơn về hai loại này cũng như mô tả chi tiết họ tiêu chuẩn IEEE 802.11 vốn dùng rộng rãi trong các mạng không dây cục bộ (Wireless LAN – WLAN).

1.2. MẠNG KHÔNG DÂY THEO HỌ TIÊU CHUẨN IEEE 802.11

IEEE 802.11 [1] là một tập các chuẩn của tổ chức IEEE (Institute of Electrical and Electronic Engineers) gồm nhiều mô tả kỹ thuật liên quan đến

hệ thống mạng không dây. Chuẩn này mô tả một giao tiếp có thể *truyền qua môi trường không khí* bằng cách sử dụng sóng vô tuyến để truyền và nhận tín hiệu giữa các thiết bị không dây (kiểu phi cấu trúc – Ad hoc), hoặc giữa một thiết bị không dây và *điểm truy cập dịch vụ không dây* (kiểu căn cơ sở hạ tầng). Như vậy, có thể thấy các loại chính của mạng không dây là:

- *Mạng có sử dụng hạ tầng*, hạ tầng ở đây thường được gọi là các điểm truy cập (AP). Các nút mạng không dây trong phạm vi của AP sẽ giao tiếp với nhau thông qua trung gian là thiết bị AP này.
- *Mạng không sử dụng hạ tầng, hay mạng tự học, hay mạng phi cấu trúc, hoặc đơn giản gọi tắt là mạng Ad hoc*, gồm nhiều nút mạng không dây nhưng luôn di động (chuyển động liên tục), các nút này có thể tạo thành một mạng tạm thời mà không cần thiết bị trung gian hoặc hạ tầng mạng có sẵn như AP. Cấu trúc mạng như vậy có thể hoạt động một cách độc lập, hoặc mạng này có thể kết nối với các mạng có hạ tầng khác để tạo thành một mạng mới có phạm vi lớn hơn, thậm chí toàn cầu. Mô hình Ad hoc này khá phù hợp trong những điều kiện không có sẵn hạ tầng mạng như vùng núi xa, hải đảo hoặc trong những điều kiện hạn chế như vừa xảy ra thiên tai, thảm họa hoặc trong những lĩnh vực đặc biệt như quân sự. Việc đảm bảo QoS hay chất lượng dịch vụ đối với loại hình mạng này cũng được quan tâm nghiên cứu nhiều hơn so với loại hình mạng có cơ sở hạ tầng.

1.3. CHẤT LƯỢNG DỊCH VỤ MẠNG KHÔNG DÂY

1.3.1. Một số nghiên cứu liên quan

Hiệu suất truyền dữ liệu đa phương tiện qua các hệ thống truyền thông không dây mà vẫn đảm bảo các tiêu chí chất lượng dịch vụ (QoS) của nó, luôn là một thách thức lớn trong truyền thông giữa các nút mạng di động. Rất nhiều vấn đề cần giải quyết do bản thân mạng không dây không thể đảm bảo

tính ổn định cũng như chất lượng dịch vụ tốt như mạng có dây truyền thông, yếu tố không dây, và sự di chuyển thay vì cố định của các nút mạng khiến rất nhiều yếu tố bên ngoài như nhiệt độ, độ ẩm, nhiễu sóng, suy giảm tín hiệu ảnh hưởng đến việc truyền tin trong mạng không dây. Với loại hình dữ liệu đa phương tiện, nếu QoS không được đảm bảo như: *độ trễ* (delay) phải nhỏ, *tỷ lệ mất gói tin* (packet loss) phải thấp, *độ biến đổi trễ* (jitter) phải ổn định,... sẽ gây ra các hiện tượng méo hình, vỡ tiếng, chất lượng dữ liệu suy giảm khi thực hiện các phiên truyền dữ liệu audio, video. Do vậy trên thế giới có rất nhiều nghiên cứu về vấn đề đảm bảo QoS cho dữ liệu đa phương tiện trong mạng không dây. Ví dụ trong bài báo [2], tầm quan trọng của truyền dữ liệu đa phương tiện với việc đảm bảo QoS cho người dùng thứ cấp đã được nghiên cứu. Các tham số QoS được phân tích đối với các nút di động qua mạng không dây với thuật toán định tuyến phù hợp được xem xét. Các thảo luận trong nghiên cứu này bao gồm cung cấp hỗ trợ QoS hiệu quả cho việc truyền đa phương tiện với các chiến lược chuyển giao (handoff) và hiệu quả tính toán các thông số hiệu suất cũng được đánh giá với công cụ mô phỏng mạng phổ biến là NS-2.

Ngoài yếu tố chất lượng dịch vụ thì một yếu tố nữa cũng có thể được xem xét với chất lượng dữ liệu đa phương tiện, đó là *Chất lượng trải nghiệm* (Quality of Experience – QoE) được định nghĩa là khả năng chấp nhận tổng thể của một ứng dụng hoặc dịch vụ, theo nhận thức chủ quan của người dùng cuối, bao gồm cả các hiệu ứng hệ thống đầu cuối hoàn chỉnh. Điều đó có nghĩa là QoE có thể bị ảnh hưởng bởi kỳ vọng và bối cảnh của người dùng, thêm một thành phần chủ quan vào các phép đo. Tuy nhiên, nhiều nghiên cứu cho rằng một số chỉ số về *chất lượng dịch vụ* (QoS), liên quan nhiều hơn đến ứng dụng và do đó gần gũi với người dùng hơn, có thể tương quan với đánh giá của người dùng về truyền đa phương tiện. Do đó, trong nghiên cứu [3], nhóm tác giả đã xây dựng một framework để nghiên cứu mối quan hệ giữa

các số liệu QoS và QoE cho việc truyền đa phương tiện theo cùng cách phân tầng (layering) như các tầng trong mô hình Internet. Framework được đề xuất trong bài báo này được gọi là *BoxingExperience*, dựa trên các phần mềm mô phỏng mạng nguồn mở ns3 và VLC thiết lập trên hệ điều hành Linux. Các tác giả đã thử nghiệm framework trong bối cảnh truyền video từ camera mạng tới một số máy khách (clients). Các mô phỏng trong nghiên cứu này cho thấy rằng giải pháp đề xuất có thể dễ dàng mô phỏng trên một máy tính để bàn thông thường, một kịch bản trong đó nhiều máy khách được kết nối với một máy chủ phát trực tuyến, do đó rất dễ dàng thực hiện các nghiên cứu có liên quan tới QoS/QoE của mạng dữ liệu đa phương tiện.

Nghiên cứu [4] tập trung vào chuẩn IEEE 802.11e, tiêu chuẩn này đã được phát hành và tích hợp với họ tiêu chuẩn không dây IEEE 802.11 với mục tiêu đảm bảo chất lượng dịch vụ, đặc biệt với dữ liệu đa phương tiện. Tuy nhiên, tiêu chuẩn này chỉ ấn định các ưu tiên cho các loại dữ liệu khác nhau nhưng không kiểm soát việc chia sẻ băng thông giữa các luồng dữ liệu khác nhau. Do vậy nghiên cứu này đã đề xuất một phương pháp chia sẻ băng thông theo tỷ lệ khác nhau cho các luồng dữ liệu đa phương tiện thông qua việc điều khiển động tham số *cửa sổ tranh giành* (Contention Window – CW) của mỗi luồng dữ liệu đa phương tiện vốn có ưu tiên khác nhau, nhằm để đạt được tỷ lệ khác nhau theo nhu cầu sử dụng.

Một vài tìm hiểu ngắn gọn như trên cho thấy rằng vấn đề đảm bảo QoS cho dữ liệu đa phương tiện trên mạng không dây là chủ đề được cộng đồng nghiên cứu quan tâm, tìm hiểu rất nhiều, việc sử dụng phương pháp *mô phỏng mạng* (network simulation) cũng thường được sử dụng trong các nghiên cứu về lĩnh vực này, trong đó các phần mềm mã nguồn mở như NS-2, ns3 được sử dụng khá nhiều trong các nghiên cứu liên quan.

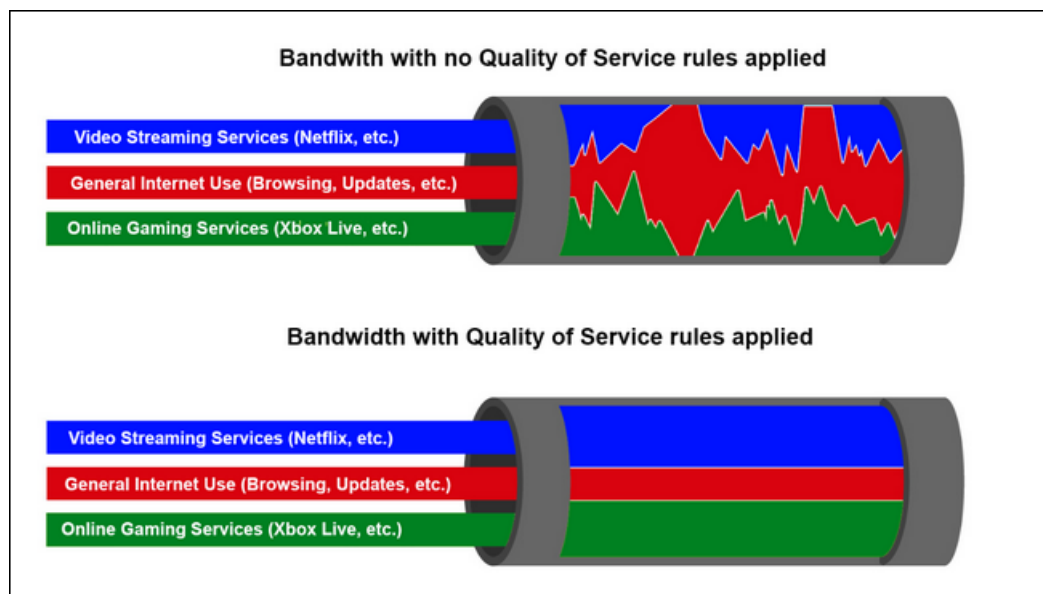
1.3.2. Các vấn đề chính của chất lượng dịch vụ mạng

Khi Internet chưa phát triển bùng nổ, dữ liệu truyền trên đó chủ yếu là truyền dữ liệu thuần túy như văn bản, tệp... thì việc phân biệt các dịch vụ truyền hoặc tạo ưu tiên cho các dịch vụ khác nhau là không cần thiết do tài nguyên mạng (ví dụ băng thông) là đáp ứng đủ cho yêu cầu của các ứng dụng đang sử dụng mạng, do đó các nhà cung cấp dịch vụ sử dụng cơ chế *Nỗ lực tối đa* (BE – Best-Effort) như nhau cho mọi người dùng. Các gói tin (packet) sẽ được truyền đi theo nguyên tắc “vào trước ra trước” hoặc “đến trước phục vụ trước” (FIFO – First In First Out) mà không cần chú ý đến đặc điểm lưu lượng (traffic) của các yêu cầu dịch vụ. Cách truyền tin như vậy sẽ không phù hợp với các loại hình dữ liệu mà đòi hỏi độ trễ (delay) phải càng thấp càng tốt như các dịch vụ thời gian thực (real-time) hay truyền phát video (streaming). Các dịch vụ ứng dụng để truyền và nhận các loại dữ liệu động như âm thanh (audio) hoặc video nói trên được gọi là đa phương tiện (multimedia data application). Loại dữ liệu đa phương tiện như vậy sẽ nhạy cảm với yếu tố *độ trễ* nhưng ngược lại, lại cho phép yếu tố *tỷ lệ mất gói tin* xảy ra ở một ngưỡng nào đó có thể chấp nhận được. Tính chất như vậy lại ngược lại so với các ứng dụng truyền tin văn bản, text,... truyền thống nên sẽ dẫn đến cần yêu cầu chất lượng dịch vụ khác đi.

Chất lượng dịch vụ (QoS) là công nghệ để nhằm mục tiêu quản lý và ưu tiên một số loại lưu lượng dữ liệu nhất định trên mạng với mục tiêu giảm *mất gói* (packet loss), *độ trễ* (latency) và *độ chập chờn* (hoặc còn gọi là biến đổi độ trễ – jitter) cho một loại giao thức hoặc một kiểu lưu lượng mạng cụ thể (ví dụ như VoIP). QoS kiểm soát và quản lý tài nguyên mạng bằng cách đặt các mức độ *ưu tiên* (priority) cho các loại dữ liệu cụ thể trên mạng. QoS tối ưu hóa mạng bằng cách quản lý băng thông, cũng như đặt mức độ ưu tiên lớn hơn cho các ứng dụng mà yêu cầu nhiều tài nguyên hơn. QoS cho phép người quản trị của hệ thống điều chỉnh cài đặt mạng, và chia nhỏ băng thông khả

dụng giữa các ứng dụng.

Ví dụ như trong Hình 1, có ba nhu cầu sử dụng mạng với các dịch vụ khác nhau: xem video trực tuyến, duyệt web, và chơi game trực tuyến, nếu không áp dụng các quy tắc QoS thì băng thông được chia đều cho ba nhu cầu này, dẫn đến việc sử dụng các dịch vụ này không đạt được như mong muốn, khi áp dụng các quy tắc QoS phù hợp, thì các dịch vụ như xem video trực tuyến với Netflix đòi hỏi nhiều tài nguyên hơn nên sẽ được cấp tỷ lệ băng thông nhiều hơn so với các nhu cầu truy cập Internet thông thường như duyệt web hoặc chơi game trực tuyến. Như vậy việc phân chia băng thông sẽ hợp lý hơn và kết quả là cả ba loại nhu cầu sử dụng đều hoạt động đáp ứng tốt nhu cầu của người dùng



Hình 1. 1. Giới thiệu về QoS

Như vậy QoS cho thấy khả năng cung cấp dịch vụ mạng khác nhau cho một loại lưu lượng nhất định. Mục đích chính là điều khiển một số yếu tố như *băng thông* (bandwidth), *độ trễ* (delay, hay latency) và *biến đổi độ trễ* (jitter),... Giảm độ trễ, giảm tỉ lệ mất mát khi truyền các gói tin trong các ứng dụng thời gian thực và các ứng dụng cần tương tác, trong khi đó vẫn đảm bảo phục vụ tốt cho những kiểu dữ liệu khác. Theo Liên minh viễn thông quốc tế

(International Telecommunication Union) tại các tài liệu [5, 6], QoS được định nghĩa là *“Tổng thể các đặc điểm của dịch vụ viễn thông có khả năng đáp ứng các nhu cầu đã được cho biết và đã được nói rõ của người sử dụng dịch vụ.”* Như vậy có thể thấy QoS dựa nhiều vào yếu tố kỹ thuật của dịch vụ do các hãng cung cấp tới khách hàng dựa trên thỏa thuận (kỹ thuật) hai bên, các thỏa thuận này dễ thấy và dễ được lượng hóa thông qua các con số kỹ thuật cụ thể.

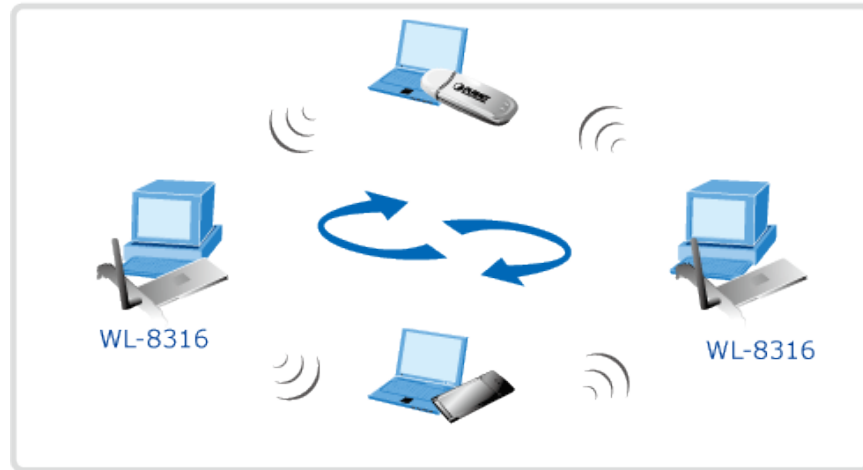
Chương 2. NGHIÊN CỨU VỀ CHẤT LƯỢNG DỊCH VỤ DỮ LIỆU ĐA PHƯƠNG TIỆN TRÊN MẠNG KHÔNG DÂY

2.1. PHÂN TÍCH MỘT SỐ KỸ THUẬT TẦNG MAC CỦA MẠNG KHÔNG DÂY

Trong thực tế mô hình mạng TCP/IP đối với các loại hình mạng *có dây* (wired) và *không dây* (wireless) chủ yếu khác nhau ở các tầng Vật lý (chủ yếu liên quan đến thiết bị) và phân tầng MAC (liên quan đến truy cập đường truyền), do vậy phần này của luận văn sẽ trình bày một số cơ chế hoạt động tại phân tầng MAC của tiêu chuẩn mạng không dây IEEE 802.11, để từ đó thấy được một số tham số sẽ ảnh hưởng tới chất lượng dịch vụ mạng không dây như thế nào.

Chuẩn 802.11 [1] định nghĩa hai chế độ làm việc chính cho các *mạng cục bộ không dây* (Wireless Local Area Networks – WLANs): *dựa trên hạ tầng cơ sở* (infrastructure based) và *không cần hạ tầng cơ sở* (infrastructure-less hay ad hoc) [7].

Kiến trúc dựa trên hạ tầng cơ sở là chế độ thường dùng để xây dựng các điểm truy cập không dây (Wi-Fi hotspot) dựa trên một điểm truy cập mạng. Điều trở ngại với kiểu kiến trúc này là chi phí mua và cài đặt cơ sở hạ tầng, các chi phí loại này có thể không được chấp nhận trong các môi trường động, ở đó người và/hoặc các phương tiện chỉ cần kết nối tạm thời trong một vùng mà không cần một cơ sở hạ tầng truyền thông sẵn có, ví dụ như trường hợp cứu hộ khi có thảm họa động đất, sóng thần, lúc đó hạ tầng mạng gần như bị phá hủy hoàn toàn, hoặc trường hợp các sinh viên trong chuyến xe bus đi du lịch có nhu cầu muốn chia sẻ tài liệu hoặc chơi game tương tác với nhau trong lúc đi trên xe. Trong những trường hợp như vậy, một giải pháp hiệu quả hơn có thể được cung cấp, đó là chế độ hoạt động không cần hạ tầng cơ sở hay *mạng tùy biến không dây* (wireless ad-hoc network), hay gọi tắt là mạng ad hoc.



Hình 2. 1. Mô hình mạng Ad-hoc

Trong mô hình mạng ad hoc như Hình 2.1, kết nối được thiết lập cho khoảng thời gian tương ứng với một phiên làm việc và không cần trạm cơ sở (base station). Các thiết bị sẽ khám phá những thiết bị khác ở trong cùng một miền để hình thành nên mạng. Các thiết bị có thể sẽ tìm kiếm các trạm trong cùng miền bằng cách phát tràn (flooding) các thông điệp quảng bá mà được chuyển tiếp bởi mỗi trạm. Khi hoạt động ở chế độ này, các trạm được xem như đóng vai trò một *tập dịch vụ cơ sở độc lập* (Independent Basic Service Set – IBSS). Bất kỳ trạm nào ở trong *miền phát* (transmission range) của trạm khác, sau một bước đồng bộ hóa, đều có thể bắt đầu truyền thông. *Điểm truy cập* (Access Point – AP) không cần thiết với chế độ mạng này, nhưng nếu một trong số các trạm đang hoạt động ở chế độ ad hoc có kết nối với mạng có dây, các trạm trong mạng ad hoc sẽ có truy cập không dây đến Internet.

Mạng *ad hoc đa chặng* (multi-hop ad hoc networks) là các mạng ad hoc mà các kết nối của chúng có thể qua nhiều trạm. Các giao thức định tuyến do đó sẽ cung cấp các kết nối ổn định cho dù các trạm chuyển động liên tục. Mỗi trạm sẽ cố gắng chuyển tiếp dữ liệu đến các trạm khác, và do đó việc xác định trạm nào chuyển tiếp dữ liệu được thực hiện tự động dựa trên kết nối mạng. Điều này ngược lại với các công nghệ mạng truyền thống ở đó một vài trạm

được thiết kế trước, thông thường với các thiết bị phần cứng như router, switch, hub và firewall, sẽ thực hiện nhiệm vụ chuyển tiếp dữ liệu.

Giao thức MAC (Media Access Control) trong chuẩn IEEE 802.11 cho việc truy cập đường truyền trong WLANs là một chuẩn không chính thức (de facto) cho các mạng không dây ad hoc. Công nghệ 802.11 là một nền tảng tốt để cài đặt các mạng ad hoc *đơn chặng* (single-hop) bởi tính đơn giản của nó. Tính đơn chặng có nghĩa là các trạm phải ở trong cùng miền phát (thường từ 100–200 mét) để có thể truyền thông trực tiếp với nhau. Sự hạn chế đó có thể được khắc phục bởi mạng ad hoc *đa chặng*, ở đó các trạm trung gian ở giữa hai bên gửi/nhận có thể chuyển tiếp thông tin để có thể đến đích.

Trong môi trường mạng ad hoc, các thiết bị di động của người dùng hình thành nên mạng và chúng phải cộng tác với nhau để cung cấp chức năng thông thường được cung cấp bởi cơ sở hạ tầng mạng (ví dụ routers, switches, và servers). Cách tiếp cận này đòi hỏi mật độ người dùng phải đủ lớn để đảm bảo việc chuyển tiếp các gói tin giữa bên gửi và bên nhận. Nếu mật độ người dùng thấp, mạng có thể trở nên không hoạt động được. Tuy nhiên, nếu mật độ người dùng cao thì hiệu năng của mạng như độ trễ, tính công bằng sẽ suy giảm nghiêm trọng. Trong các mạng ad hoc đa chặng, các trạm cộng tác để chuyển tiếp các gói tin từ các trạm khác qua mạng. Do đó, một trạm phải truyền đi cả *luồng trực tiếp* (direct flow), sinh ra bởi chính trạm đó và các *luồng chuyển tiếp* (forwarding flow), được sinh ra bởi các trạm hàng xóm, do đó nó chia sẻ dung lượng kênh truyền với các trạm hàng xóm của nó. Hiệu ứng của sự tranh chấp tại tầng MAC và tầng liên kết sẽ ảnh hưởng đến chất lượng của mạng.

Chức năng cộng tác phân tán (Distributed Coordination Function – DCF) là một kỹ thuật tầng MAC cơ bản trong IEEE 802.11, nó được thiết kế để cung cấp cơ hội công bằng cho mọi trạm để truyền đi các frames trong ngữ

cảnh phân tán. DCF sử dụng giao thức *đa truy cập sử dụng sóng mang có tránh xung đột* (Carrier Sense Multiple Access with Collision Avoidance – CSMA/CA) với cách truy cập dựa trên kỹ thuật *quay lui hàm mũ nhị phân* (Binary Exponential Back-off – BEB). Kỹ thuật BEB điều khiển tần suất truy cập kênh truyền của mỗi trạm bằng cách chọn ngẫu nhiên một giá trị quay lui từ một đến CW dựa trên phân phối ngẫu nhiên chuẩn, với CW là kích thước *cửa sổ tranh chấp* (Contention Window – CW). Do đó, dường như toàn bộ các trạm đang tranh chấp sẽ có cùng cơ hội để truy cập kênh truyền được chia sẻ, tuy nhiên, trong cấu hình mạng đa chặng, kỹ thuật BEB lại gặp phải các vấn đề như *không công bằng* (unfairness) và *lưu lượng thấp* (low throughput), đặc biệt trong trường hợp *tải mạng được cung cấp* (offered load) có lưu lượng lớn khi có quá nhiều yêu cầu sử dụng.

Kỹ thuật BEB xác định kích thước cửa sổ tranh chấp tương ứng với điều kiện tắc nghẽn (khi có nhiều nút mạng cùng truy cập kênh truyền). Kỹ thuật này sẽ nhân đôi giá trị CW sau mỗi lần đụng độ cho đến khi đạt ngưỡng CW_{max} và đặt lại (reset) giá trị CW thành CW_{min} với mỗi lần phát thành công. Tuy nhiên, kỹ thuật BEB không xem xét các điều kiện khác về các trạm hàng xóm, thông tin tầng trên, ... số luồng trong kênh truyền hoặc số người dùng trong hệ thống. Do đó, giá trị CW sau vài lần tắc nghẽn có thể sẽ không phải là giá trị tối ưu cho tính công bằng, đặc biệt trong các mạng đa chặng bất đối xứng.

Mạng Ad-hoc ứng dụng trong rất nhiều lĩnh vực cũng như trong các trường hợp thực tế như ở khách sạn, trường học, sân bay, ...

2.2. CHUẨN MẠNG KHÔNG DÂY IEEE 802.11E CHO DỮ LIỆU ĐA PHƯƠNG TIỆN

Công nghệ giao tiếp không dây dựa trên chuẩn 802.11[1] đã trở nên phổ biến trong nhiều lĩnh vực, ví dụ các điểm truy cập Wi-Fi, mạng đô thị diện

rộng (Metropolitan Area Network – MAN), mạng giao thông (vehicular),... Tại các lĩnh vực đó, truyền thông không dây *đa chặng* (multi-hop) đóng một vai trò quan trọng. Tuy nhiên, chất lượng dịch vụ (QoS) được cung cấp trong các giao tiếp này vẫn là một vấn đề nghiên cứu có nhiều thách thức. Chuẩn IEEE 802.11e[8], dựa trên 802.11 được ra đời nhằm đưa ra các tiêu chuẩn tốt hơn trong việc đảm bảo chất lượng dịch vụ và đã được chấp nhận như một thành phần *mở rộng hỗ trợ đa phương tiện* (WiFi Multimedia – WMM hay Wireless Multimedia Extensions – WME) của IEEE 802.11[1].

Điểm nổi bật của IEEE 802.11e là có *chức năng cộng tác lai* (Hybrid Coordination Function – HCF) – là sự kết hợp của các chức năng DCF, DCF đã có từ 802.11, với những cơ chế đảm bảo QoS nâng cao nhằm hỗ trợ dịch vụ có phân biệt (DiffServ). HCF bao gồm hai cơ chế điều khiển là *truy cập kênh truyền phân tán nâng cao* (EDCA – Enhanced Distributed Channel Access) dựa trên cơ chế có tranh chấp (contention-based) và *điều khiển truy cập tập trung không có tranh chấp* (HCCA – HCF Controlled Channel Access) dựa trên cơ chế hỏi vòng (poll-based). Trong hai cơ chế này, kỹ thuật EDCA được sử dụng nhiều hơn vì phù hợp với đặc tính phân tán của các nút trong mạng không dây ad hoc.

Phương pháp EDCA dùng phương pháp truy cập môi trường truyền có phân biệt, sử dụng những mức ưu tiên khác nhau cho mỗi loại luồng dữ liệu. EDCA định nghĩa bốn loại ưu tiên theo *phân loại truy cập* (AC – Access Categories) cho các kiểu dữ liệu khác nhau và có các dịch vụ phân biệt cho từng loại AC này. Việc các khung dữ liệu khác nhau được ánh xạ vào những AC sẽ phụ thuộc vào sự đòi hỏi chất lượng dịch vụ của tầng trên. Mỗi khung tin từ tầng trên chuyển xuống tầng MAC được đánh trọng số *ưu tiên người dùng* (UP – User Priority) tùy thuộc vào ứng dụng sinh ra khung đó. Có 8 giá trị trọng số ưu tiên được mô tả như trong Bảng 2.1.

Bảng 2. 1. Độ ưu tiên và các mức truy cập

Độ ưu tiên	UP	AC	Kiểu dữ liệu
thấp nhất	1	AC_BK	Background
-	2	AC_BK	Background
-	0	AC_BE	Best effort
-	3	AC_BE	Best effort
-	4	AC_VI	Video
-	5	AC_VI	Video
-	6	AC_VO	Voice
cao nhất	7	AC_VO	Voice

Kỹ thuật EDCA xử lý việc tranh chấp truy cập đường truyền dựa vào các tham số sau đây.

Số khoảng cách giữa các frame xen kẽ (Arbitrary InterFrame Space Number – AIFSN): là số khe thời gian (SlotTime) sau mỗi thời hạn SIFS (Short Interframe Space – là khoảng thời gian ngắn giữa các frame, là lượng thời gian tính bằng micro-giây cần thiết để giao diện không dây xử lý frame đã nhận và phản hồi với frame phản hồi) mà một trạm phải chờ trước khi đi vào pha quay lui hoặc pha truyền dữ liệu. Giá trị AIFSN sẽ ảnh hưởng đến khoảng thời gian giữa các khung (AIFS), là giá trị xác định khoảng thời gian (tính bằng đơn vị thời gian cụ thể thay vì theo số khe thời gian) mà một trạm phải chờ trước khi truyền gói tin tiếp theo hoặc khởi động thuật toán quay lui:

$$AIFS = SIFS + AIFSN \times SlotTime$$

Theo đó một AC với giá trị AIFSN thấp hơn sẽ có AIFS nhỏ và do vậy có được độ ưu tiên cao hơn.

Cửa sổ tương tranh (Contention Window – CW): mỗi trạm tính ra tổng giá trị thời gian quay lui (backoff) từ một giá trị ngẫu nhiên lấy trong khoảng giới hạn của cửa sổ tương tranh:

$$\text{Backoff} = \text{AIFS} + \text{random}[1, \text{CW}]$$

Trong đó CW được xác định trong khoảng (CW_{min} , CW_{max}) phụ thuộc số lần truyền thông. Một AC với giá trị CW càng thấp sẽ có xác suất càng cao để nhận một số ngẫu nhiên nhỏ hơn, do vậy sẽ có được độ ưu tiên cao hơn.

Giới hạn của TXOP (TXOP limit): khoảng thời gian truyền tối đa cho mỗi trạm sau khi giành được một *ơ hội truyền* (Transmission Opportunity). Nếu giá trị này bằng không (0), nghĩa là khi kiểu truy cập (AC) có quyền truy cập kênh truyền, khi đó nó được phép gửi duy nhất một khung từ hàng đợi AC tương ứng. Điều này nhằm giới hạn lưu lượng có độ ưu tiên thấp. Nếu giá trị lớn hơn không (> 0), nghĩa là một AC có thể truyền nhiều khung từ hàng đợi AC của nó (lưu ý rằng giá trị TXOP limit được gán cho một loại AC cụ thể chứ không phải gán cho một trạm). Như vậy trong khoảng thời gian truyền không vượt quá giá trị giới hạn của TXOP, trạm sẽ được phép truyền một số khung từ một hàng đợi AC nào đó. Do vậy, một AC với giá trị TXOP limit cao hơn sẽ có độ ưu tiên cao hơn.

Nhìn chung AC có độ ưu tiên cao hơn sẽ có AIFSN, CW_{min} , CW_{max} nhỏ hơn và TXOP limit lớn hơn so với AC có độ ưu tiên thấp hơn.

Những tham số EDCA này là khác nhau với từng loại AC, và được thể hiện chi tiết trong Bảng 2.2.

Bảng 2. 2. Các tham số EDCA mặc định.

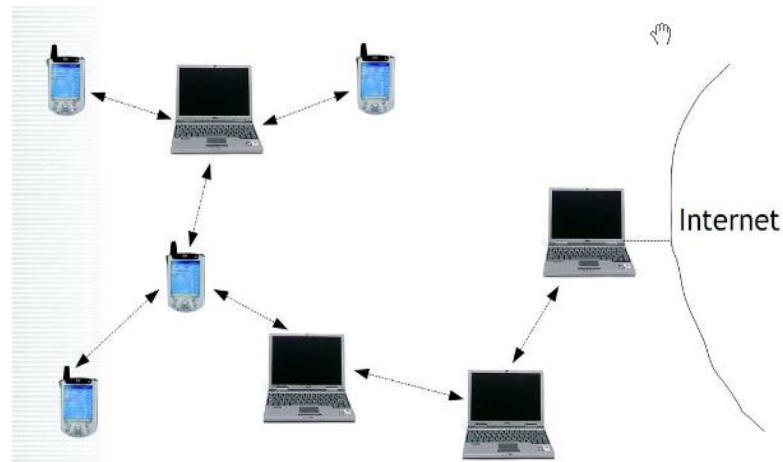
Tham số	BK	BE	VI	VO
AIFSN	7	3	2	2
CWmin	15	15	7	3
CWmax	1023	1023	15	7
TXOPLimit (ms)	0	0	1.504	3.008

Tương ứng với các bộ tham số cho các độ ưu tiên này, các thông số chất lượng mạng như *độ trễ truyền gói tin* (delay), *tỷ lệ mất gói tin khi truyền* (packet loss) và đặc biệt là *thông lượng* (throughput) cũng có sự khác biệt giữa các loại dữ liệu đa phương tiện.

2.3. MỘT SỐ LOẠI MẠNG DI ĐỘNG CÓ SỬ DỤNG CHUẨN 802.11

2.3.1. Mạng không dây di động (MANET)

Mạng không dây di động (Mobile Ad-hoc Networking – MANET) được tổ chức IETF (Lực lượng Chuyên trách về Kỹ thuật Liên mạng, Internet Engineering Task Force) định nghĩa: “*mạng không dây di động MANET là một vùng tự quản (AS – Autonomous System) của các bộ định tuyến (router), có thể coi đó là các nút mạng (network nodes) được gắn kết với nhau nhờ liên kết không dây, các nút mạng có thể di chuyển theo cách không xác định trước (tự do), vì thế kiến trúc mạng MANET sẽ liên tục thay đổi nên khó có thể dự đoán được trước cấu hình mạng sẽ như thế nào*”.



Hình 2. 2. Mô hình cơ bản mạng MANET

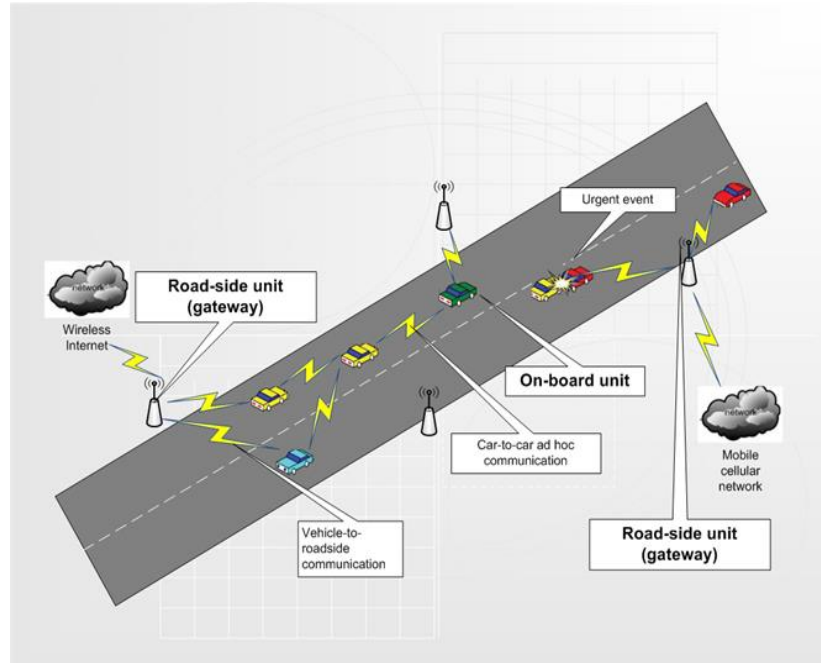
Đặc điểm mạng không dây di động MANET là tập hợp các mobile node trong một mạng có tính chất tạm thời không có sự quản lý của các nhà quản trị. Mô hình này gồm hai hay nhiều *nút mạng không dây* (wireless mobile nodes) kết nối với nhau theo mô hình mạng ngang hàng (peer-to-peer), tức là các nút mạng có vai trò như nhau, có thể kết nối với nhau và không cần đến cơ sở hạ tầng của các mạng trước đó. Các nút trong mạng này còn đóng vai trò như là các router có khả năng tìm kiếm, duy trì và định tuyến các gói dữ liệu cho các nút nằm trong cùng phát sóng của nó. Như vậy, mạng MANET là một mạng tạm thời và không có máy chủ trung tâm (Central Server).

2.3.2. Mạng không dây di động trong xe cộ (VANET)

Vehicular Ad hoc NETWORK (VANET) là mạng tự tổ chức, được tạo thành từ các phương tiện xe cộ lưu thông trên đường. Mỗi phương tiện tham gia vào mạng được trang bị thiết bị thu/phát để có thể liên lạc, chia sẻ và trao đổi thông tin lẫn nhau.

Thông tin trao đổi trong mạng VANET bao gồm thông tin về lưu lượng xe cộ, tình trạng kẹt xe, tai nạn giao thông, nguy hiểm cần tránh và cả những dịch vụ thông thường như dịch vụ đa phương tiện, Internet.... để cung cấp liên

lạc giữa các xe cộ ở gần nhau và với các thiết bị cố định (hay các thiết bị bên đường). Tuy nhiên việc truyền thông tin trong mạng VANET còn rất nhiều thách thức.



Hình 2. 3. Mô hình mạng VANET

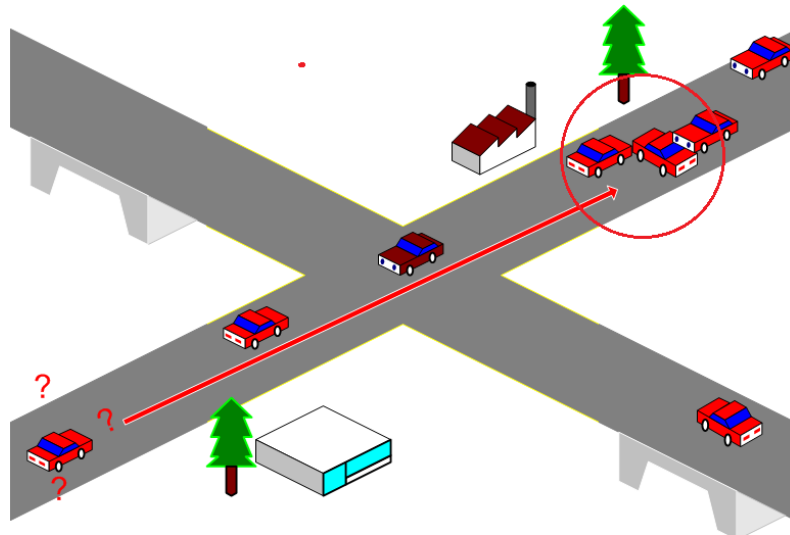
Hai thiết bị chính trong mô hình mạng Vanet là OBU và RSU. OBU (On-Board Unit) là thiết bị gắn trên các phương tiện di chuyển phụ trách việc liên lạc qua mạng ad hoc của phương tiện. RSU (Road-Side Unit) là các thiết bị đặt cố định trên đường, chịu trách nhiệm liên lạc với phương tiện lưu thông qua mạng ad hoc, liên hệ với tổng đài qua mạng di động. Một số pha hoạt động của mạng VANET:

- Hai xe muốn liên lạc (gọi video, chat text, truyền file,...) trên đường qua mạng ad-hoc.
- Một xe nào đó muốn ảnh chụp đường tại vị trí của mình lên server. Xe này sẽ gửi file qua mạng ad hoc các xe khác để đến RSU. RSU sẽ đưa hình ảnh này lên server.

- Một xe muốn xem ảnh đường phố ở vị trí cách nó một khoảng cách nào đó, xe này sẽ tìm cách liên lạc với một xe bất kì trong khoảng đó, và 2 xe chia sẻ hình ảnh cho nhau.

Các chức năng này khi đưa vào mạng giao thông có thể giúp các phương tiện cảnh báo tắc đường, cảnh báo tai nạn... Để thực hiện được các pha liên lạc này, điều tất yếu phải cần một giao thức định tuyến.

Ví dụ bài toán đặt ra khi người dùng đổi mặt trường hợp sau. Người dùng đang đi trên đường và nhìn thấy phía trước đường có vẻ đông. Người này sẽ muốn biết ở trên đó tình trạng đang như thế nào: tắc đường hay có tai nạn hay là một sự việc đặc biệt nào, mức độ ra sao. Thông tin này sẽ ảnh hưởng đến hành động của người tham gia giao thông: tiếp tục đi tiếp hay sẽ chọn một giải pháp khác. Nếu người tham gia giao thông có thể liên lạc được với một người dùng khác đang ở vị trí đó thì chắc chắn người đó sẽ có được thông tin ở đó và đưa ra quyết định chọn đường.



Hình 2. 4. Nhu cầu định tuyến theo vị trí

Chương 3. ĐÁNH GIÁ CHẤT LƯỢNG DỊCH VỤ DỮ LIỆU ĐA PHƯƠNG TIỆN TRÊN MẠNG KHÔNG DÂY

3.1. GIỚI THIỆU VỀ MÔ PHỎNG MẠNG

Để đi đến công nhận chính thức một tiêu chuẩn trong các mạng không dây IEEE 802.11[1] cần trải qua quá trình đánh giá thử nghiệm nhằm chứng minh hiệu quả. Việc đánh giá các thông số mạng nói chung là khó khăn, đặc biệt với mạng không dây do tính đa dạng của phần cứng, phần mềm, thông số mạng và khó có một phương pháp tối ưu nào cho việc đánh giá tất cả các thông số mạng. Một số phương pháp chính thường được thực hiện như *mô hình hóa* (modeling), *mô phỏng* (simulation) và *giả lập* (emulator). Phương pháp mô hình hóa có ưu điểm là dễ đưa thêm tham số, kết quả chính xác do có thể chứng minh bằng toán học, mô hình có tính tổng quát hóa rất cao; tuy nhiên nhược điểm là khó triển khai do môi trường thực tế mạng thường có nhiều yếu tố tác động. Phương pháp mô phỏng có ưu điểm là chi phí thấp, các thông số mạng có thể dễ dàng thay đổi, quan sát nhờ việc thiết lập giá trị trong phần mềm mô phỏng; nhược điểm của phương pháp này là khó chứng minh chính xác như phương pháp mô hình và các yếu tố đánh giá dù sao vẫn là do phần mềm mô phỏng thiết lập nên vẫn không đạt được mức độ giống các mạng chạy trong thực tế. Phương pháp giả lập hay nói cách khác là sử dụng thiết bị thật để giả lập một môi trường mạng gần giống như thực tế để đánh giá thử nghiệm, phương pháp này có ưu điểm hơn hẳn hai phương pháp mô hình và mô phỏng về tính chất giống thật, nhưng cũng có khó khăn do chi phí đầu tư thiết bị, chỉ giả lập một phần của mạng chứ không giống toàn bộ một mạng trong thực tế (do có nhiều thiết bị, nhiều người dùng). Để phù hợp với khả năng, trong khuôn khổ luận văn này sẽ tìm hiểu phương pháp *mô phỏng* và áp dụng để phân tích, đánh giá chất lượng dịch vụ thông qua một số kịch bản truyền dữ liệu đa phương tiện trong mạng không dây.

Theo một số nghiên cứu đánh giá, so sánh các giải pháp phần mềm mô phỏng mạng như [9, 10] thì bộ công cụ mô phỏng ns-2[11] là một trong những phần mềm được sử dụng rộng rãi trong nhiều nghiên cứu thuộc lĩnh vực đánh giá hiệu năng và chất lượng mạng không dây, do tính chất miễn phí, mã nguồn mở, được đông đảo cộng đồng nghiên cứu cùng tham gia trao đổi chia sẻ. công cụ NS-2 là một phần mềm mô phỏng mạng đã có từ lâu và được dùng rất nhiều trong lĩnh vực nghiên cứu về mạng máy tính, đặc biệt là các phần khó thiết lập môi trường thực như mạng không dây (vì đòi hỏi nhiều thiết bị), do được cộng đồng sử dụng nhiều nên ít nhiều việc mô phỏng bằng NS-2 có tính tin cậy cao và vẫn được dùng trong nhiều nghiên cứu liên quan đến mạng không dây cũng như các loại hình dịch vụ dữ liệu khác nhau trên mạng không dây.

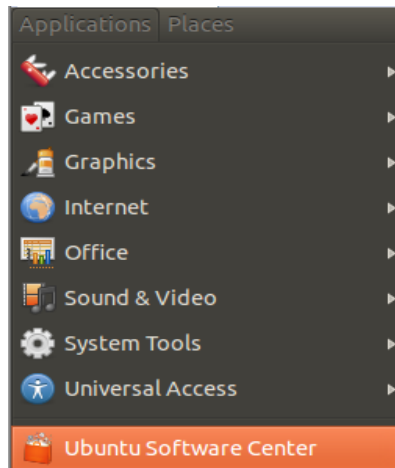
Do vậy phần này của luận văn sẽ tập trung làm rõ các bước cài đặt sử dụng ns-2 cũng như xây dựng một số kịch bản mô phỏng để đánh giá chất lượng dịch vụ mạng không dây, đặc biệt với loại dữ liệu đa phương tiện.

3.2. CÀI ĐẶT, THIẾT LẬP, THỰC THI VÀ XỬ LÝ KẾT QUẢ MÔ PHỎNG TRONG NS-2

NS-2 [11] là phần mềm mô phỏng mạng điều khiển sự kiện riêng rẽ hướng đối tượng, được phát triển tại UC Berkely, viết bằng ngôn ngữ C++ và OTcl. Một số khả năng chính của NS-2 bao gồm: kiểm tra tính ổn định của các giao thức mạng đang tồn tại; đánh giá các giao thức mới trước khi đưa vào sử dụng; thực thi những mô hình mạng lớn mà gần như ta không thể thực thi trong thực tế; khả năng mô phỏng nhiều loại mạng khác nhau. NS-2 có thể dùng thêm với công cụ hiển thị hình ảnh dạng động là Network Animator hay NAM[12] – cho phép hiển thị cấu hình mạng mô phỏng của như hoạt động của mạng một cách động và trực quan, rất thuận tiện cho việc quan sát hoạt động của mạng.

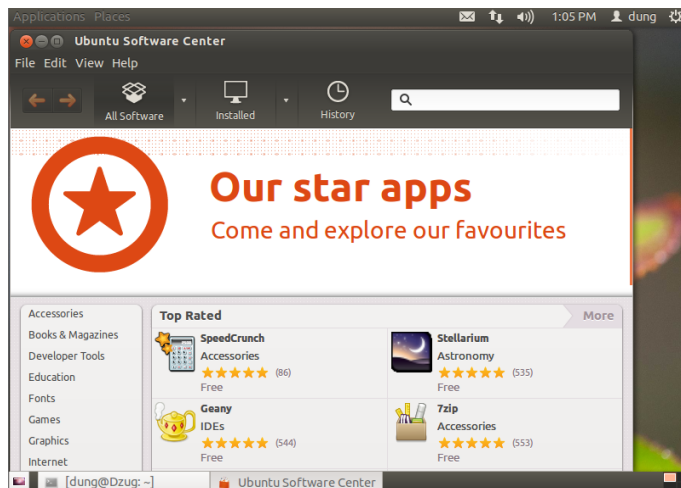
Về cơ bản thì việc cài đặt NS-2 và cả NAM không quá khó, nhất là trên môi trường hệ điều hành Linux. Phần này của luận văn trình bày ngắn gọn cách cài đặt NS-2 trên hệ điều hành Ubuntu Linux qua một số bước sau đây:

Vào hệ điều hành Ubuntu, chọn thư viện phần mềm Ubuntu Software Center



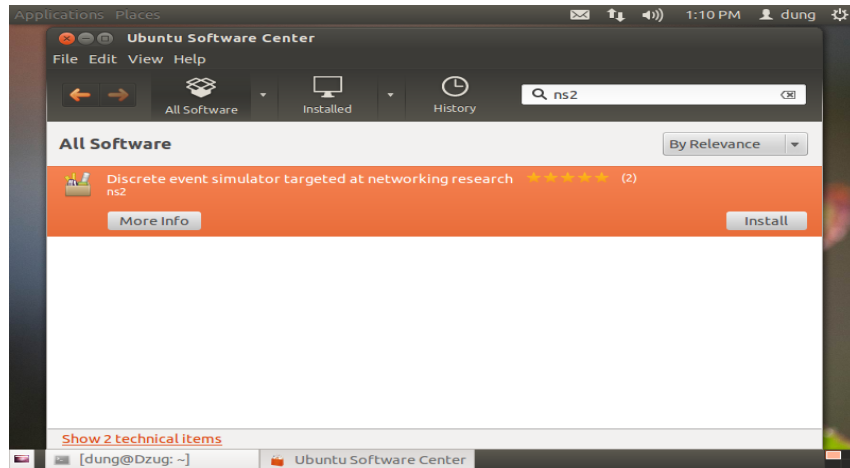
Hình 3. 1. Ubuntu Software Center

Hiện ra cửa sổ như hình :



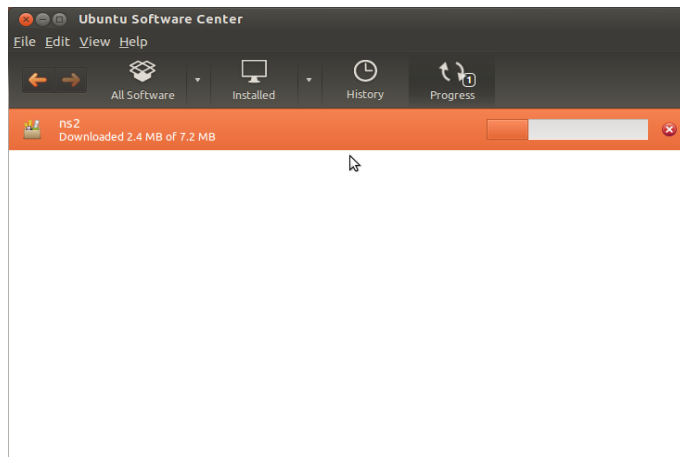
Hình 3. 2. Cửa sổ tìm kiếm trong Software Center

Vào thanh công cụ tìm kiếm gõ từ khóa “ns2” và tìm kiếm:



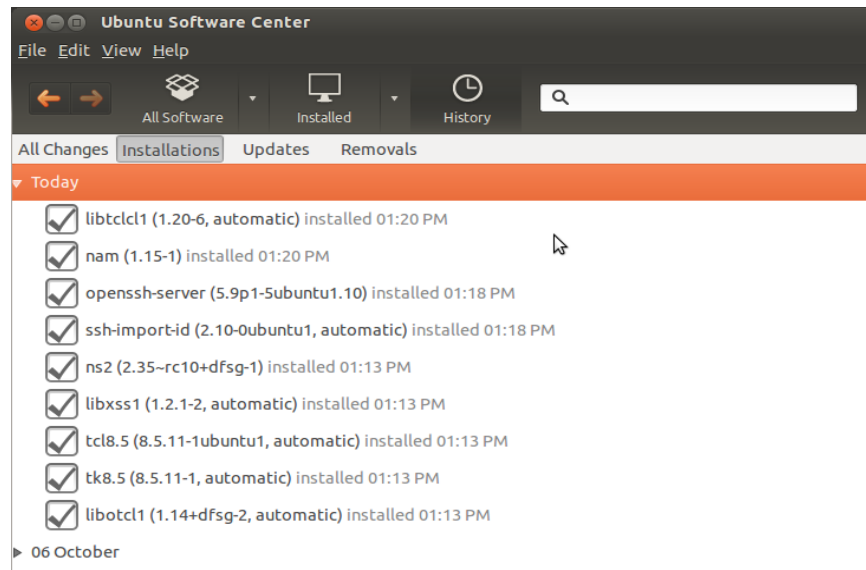
Hình 3. 3. Tìm gói ns2 trên thư viện phần mềm của Ubuntu

Chọn Install và đợi cho tiến trình download và cài đặt tự động hoàn tất.



Hình 3. 4. Tiến trình cài đặt phần mềm

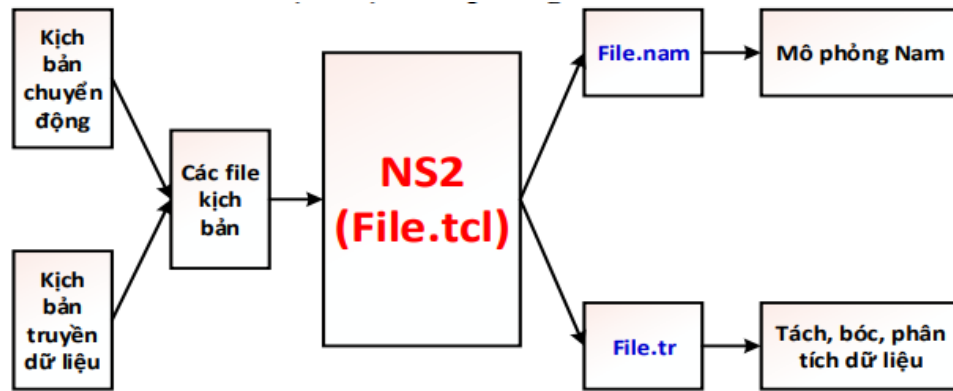
Tương tự cài đặt NAM cũng giống như NS2, sau khi cài đặt xong vào phần history chọn mục Installations để kiểm tra các gói phần mềm của NS2 và NAM đã được cài đặt trên hệ thống hay chưa.



Hình 3. 5. Kiểm tra các gói cài đặt

Việc thực thi kết quả mô phỏng trong NS-2 sẽ cho kết quả đầu ra với định dạng (format) theo hai kiểu là: dạng **.nam** và dạng **.tr**, trong đó kiểu thứ nhất chủ yếu dùng để chạy với NAM để cho ra hình ảnh của kịch bản mô phỏng, còn kiểu thứ hai quan trọng hơn lưu lại toàn bộ các sự kiện đã xảy ra với mô phỏng đã chạy.

Việc xây dựng và thực thi các kịch bản mô phỏng mạng trong NS-2 được thực hiện qua một số bước, được mô tả như trong Hình 3.6. Về cơ bản thì chúng ta mô tả kịch bản mạng sẽ chuyển động như thế nào (hoặc thậm chí đứng yên với mạng có dây) cũng như việc truyền dữ liệu giữa các nút mạng ra sao, nếu kịch bản phức tạp với mã nguồn code dài thì có thể chia thành nhiều file để tiện quản lý, nhưng các file này đều phải có định dạng **.tcl** (dùng cú pháp ngôn ngữ TCL). Sau khi thực thi file kịch bản dạng **.tcl** này thì kết quả sẽ thu được hai tệp *vết* (trace file) dạng **.nam** và **.tr**. Để xem file **.nam** thì đơn giản là sử dụng NAM, còn để phân tích **.tr** thì chúng ta bắt buộc phải tự lập trình để bóc tách dữ liệu từ tệp **.tr** ra thành các thông số về hiệu năng, chất lượng,... mạng như độ trễ, độ mất gói, thông lượng,... để có thể biểu diễn dưới dạng bảng biểu, biểu đồ,... cho tiện phân tích, đánh giá, so sánh...

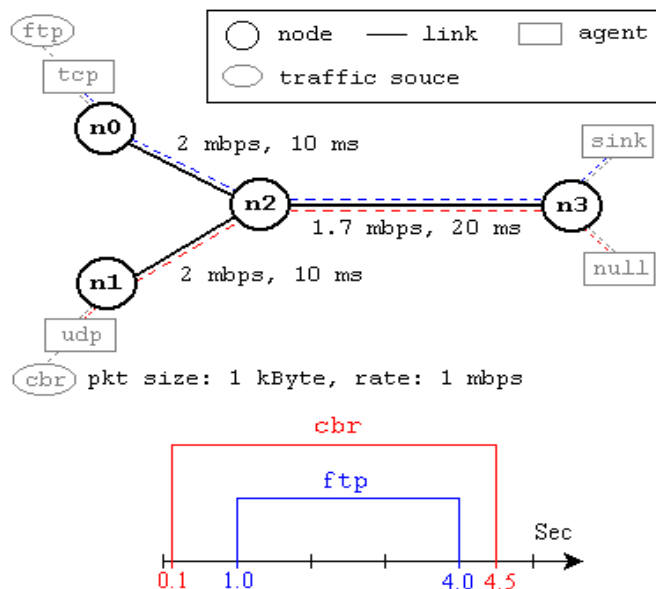


Hình 3. 6. Luồng các sự kiện khi chạy mô phỏng trong NS-2

Dưới đây là mô tả chi tiết từng bước thực hiện mô phỏng với NS-2 với ví dụ đơn giản gồm bốn nút mạng.

Các bước mô phỏng kịch bản trong NS-2:

Nội dung kịch bản mô phỏng được viết trong một file “simple.tcl” (nội dung file được mô tả chi tiết bên dưới). Thực thi mô phỏng kịch bản này bằng cách gõ lệnh “ns ns-simple.tcl” tại dấu nhắc hệ thống.



Hình 3. 7. Kịch bản mô phỏng mạng đơn giản với NS-2

Mạng trên bao gồm 4 node (n_0, n_1, n_2, n_3). Liên kết giữa các node n_0 và n_2 , n_1 và n_2 là loại *song công* (duplex – là kiểu đồng thời truyền và nhận dữ liệu cùng lúc) với các thông số:

- Băng thông (bandwidth): 2 Mbps
- Độ trễ truyền (delay): 10 ms.
- Liên kết kiểu duplex giữa $n2$ và $n3$ có bandwidth là 1.7 Mbps, độ trễ (delay) bằng 20 ms.
- Các nút mạng dùng hàng đợi dạng DropTail, có giá trị lớn nhất của kích thước hàng đợi là 10.

Tác tử “tcp” được gắn vào nút $n0$ và tác tử “sink” được gắn vào nút $n3$. Tác tử “tcp” sẽ tạo gói tin với kích thước lớn nhất là 1 (KB). Tác tử kiểu TCP “sink” sẽ tạo ra rồi gửi gói tin dạng phản hồi (ACK) cho phía gửi (sender) (là tác tử gửi gói tin đi), và sau đó giải phóng gói tin nhận được. Tác tử kiểu UDP “udp” mà gắn với nút $n1$ sẽ kết nối với tác tử “null” vốn được gắn vào nút $n3$. Tác tử “null” sẽ chỉ giải phóng gói tin mà đã nhận được. Các bộ sinh traffic (lưu lượng mạng) kiểu FPT là “ftp”, và kiểu bit-rate-cố định (Constant Bit Rate) là “cbr” tương ứng sẽ được gắn vào các tác tử “tcp” và “udp”. Tác tử “cbr” sẽ được thiết lập để tạo ra các gói tin kích thước 1 (KB) với tốc độ gửi 1 Mbps. Tác tử “cbr” được khởi chạy (start) vào thời điểm bắt đầu là 0.1 (s) và được dừng chạy (kết thúc) vào thời điểm 4.5 (s), còn tác tử “ftp” khởi chạy vào 1.0 (s) và dừng chạy vào 4.0 (s).

```

#Create a simulator object
set ns [new Simulator]

#Define different colors for data flows (for NAM)
$ns color 1 Blue
$ns color 2 Red

#Open the NAM trace file
set nf [open out.nam w]
$ns namtrace-all $nf

#Define a 'finish' procedure
proc finish {} {
    global ns nf
    $ns flush-trace
    #Close the NAM trace file
    close $nf
    #Execute NAM on the trace file
    exec nam out.nam &
    exit 0
}

#Create four nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

#Create links between the nodes
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns duplex-link $n2 $n3 1.7Mb 20ms DropTail
#Set Queue Size of link (n2-n3) to 10
$ns queue-limit $n2 $n3 10
#Give node position (for NAM)
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient right

#Monitor the queue for link (n2-n3). (for NAM)
$ns duplex-link-op $n2 $n3 queuePos 0.5

#Setup a TCP connection
set tcp [new Agent/TCP]
$tcp set class_ 2
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPsink]
$ns attach-agent $n3 $sink
$ns connect $tcp $sink
$tcp set fid_ 1

```

```

#Setup a FTP over TCP connection
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP
#Setup a UDP connection
set udp [new Agent/UDP]
$ns attach-agent $n1 $udp
set null [new Agent/Null]
$ns attach-agent $n3 $null
$ns connect $udp $null
$udp set fid_ 2
#Setup a CBR over UDP connection
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set type_ CBR
$cbr set packet_size_ 1000
$cbr set rate_ 1mb
$cbr set random_ false
#Schedule events for the CBR and FTP agents
$ns at 0.1 "$cbr start"
$ns at 1.0 "$ftp start"
$ns at 4.0 "$ftp stop"
$ns at 4.5 "$cbr stop"
#Detach tcp and sink agents (not really necessary)
$ns at 4.5 "$ns detach-agent $n0 $tcp
$ns detach-agent $n3 $sink"

#Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"

#Print CBR packet size and interval
puts "CBR packet size = [$cbr set packet_size_]"
puts "CBR interval = [$cbr set interval_]"

#Run the simulation
$ns run

```

Ở phần trên của luận văn, chúng tôi đã trình bày về các cách sử dụng đơn giản với NS-2 để tạo nút mạng (nodes), liên kết giữa các nút (links), các tác tử (agents), sinh lưu lượng mạng (traffic)...

Mã nguồn mô phỏng đề: tạo bộ mô phỏng (đối tượng “Simulator”), lập lịch cho sự kiện mô phỏng (chạy ns tại một thời điểm <time>), và khởi chạy bộ mô phỏng (bằng hàm “run”) như sau:

```

set ns [new Simulator]
$ns at <time> <event>
$ns run

```

a. Khởi tạo node

Đối tượng tạo nút mô phỏng là “node” dùng để khởi tạo một nút mạng. Ví dụ sau là tạo ra hai nút với tên biến được đặt là $n0$, và $n1$.

```

set n0 [$ns node]

```

```
set n1 [$ns node]
```

Nếu tạo số lượng nút nhiều có thể dùng vòng lặp for (cú pháp tương tự trong ngôn ngữ C/C++):

```
set n0 [$ns node]
set n1 [$ns node]
```

Để tiện phân biệt các nút có thể gán màu cho các nút bằng đối tượng “color” như sau:

```
$n0 color <colour>
```

Các giá trị màu có thể dùng là BLUE (màu xanh dương), BLACK (màu đen), seaGreen (màu xanh lá), RED (màu đỏ).

Với mạng không dây, sau khi khởi tạo node thì setup tọa độ trong mô phỏng:

```
$node_(0) set X_ 0
$node_(0) set Y_ 0
$node_(0) set Z_ 0.0
```

Và khoảng chuyển động tự do (random-motion):

```
set node_(0) [$ns_ node]
$node_(0) random-motion 0
```

Ở đây ta cho khoảng di động tự do của node bằng 0 để tiện lợi cho việc quan sát và phân tích.

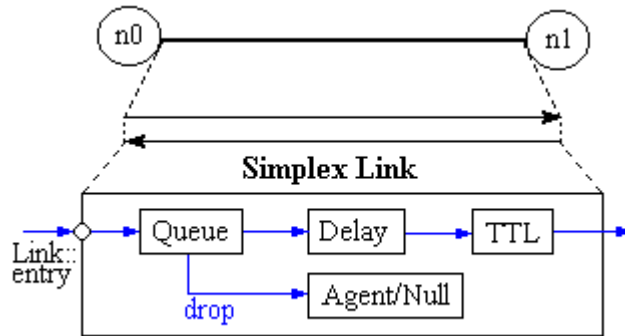
Cuối cùng trong việc tạo node là điều chỉnh kích thước của node, dùng lệnh lặp for để tối ưu :

```
for {set i 0} {$i < $val(nn)} {incr i} {
  $ns_ initial_node_pos $node_($i) 30
}
```

b. Tạo liên kết (link) giữa các nút

Trong NS-2, khi chúng ta tạo liên kết bằng cách sử dụng hàm “*duplex-link*” (song công) thì sẽ tạo ra hai liên kết hai chiều *simplex link* (đơn công).

Một *output queue* (hàng đợi đầu ra) của nút được thực hiện như một phần trong đối tượng “*simplex link*”. Các gói tin khi đi ra queue thì được chuyển tới đối tượng “*delay*”; còn những packet bị bỏ đi (drop) ra queue sẽ được gửi đến tác tử “*Agent/Null*” rồi bị hủy tại đó. Đối tượng *time-to-live* (TTL, hiểu là thời gian tồn tại của gói tin) sẽ tính giá trị thời gian tồn tại này cho mỗi gói tin và cập nhật giá trị thời gian tồn tại mới.



Hình 3. 8. Liên kết (link) giữa các node trong NS2

Để tạo liên kết “*simplex*” (đơn công, hay một chiều) giữa hai nút, sử dụng lệnh sau:

```
$ns simplex-link $n0 $n1 <bandwidth> <delay>
<queue_type>
```

Tạo liên kết “*duplex*” (song công, hay hai chiều) giữa hai nút thì sử dụng lệnh sau đây:

```
$ns duplex-link $n0 $n1 <bandwidth> <delay> <queue_type>
```

Các giá trị như: *băng thông* (bandwidth) và *trễ* (delay) tương ứng, được đặt bằng 1 (Mbps) và 10 (ms). NS-2 hỗ trợ nhiều giá trị “*queue_type*” (kiểu hàng đợi) phổ biến như phát hiện *sớm ngẫu nhiên* (RED – Random Early Detection), hay *loại bỏ phần cuối* (Drop Tail),...

Đối với mạng không dây (wireless) thì ta không cần tạo liên kết giữa hai nút vì chúng hoạt động trong môi trường không dây (vô tuyến), chỉ cần hai nút nằm trong phạm vi phát (transmission range) của nhau là có thể gửi nhận dữ liệu giữa hai bên.

c. Khởi tạo các tác tử mạng (network agent)

Hai lớp **Agent** và **Application** sẽ tạo ra lưu lượng mạng (traffic) trong NS-2. Mỗi nút trong mạng muốn nhận và gửi lưu lượng mạng thì cần có tác tử (agent) tương ứng gắn vào nó. Trên đỉnh của các tác tử sẽ chạy ứng dụng

mạng, và ứng dụng mạng sẽ quyết định loại lưu lượng nào, ví dụ truyền tệp (FTP), web (HTTP), hoặc dữ liệu đa phương tiện,... sẽ được mô phỏng.

Trong NS-2 có hai loại tác tử chính: TCP và UDP, cách tạo mỗi loại tác tử này được mô tả như bên dưới.

- **Tác tử UDP**

```
set udp0 [new Agent/UDP]
set null [new Agent/Null]
$ns attach-agent $n0 $udp0
$ns attach-agent $n1 $null
$ns connect $udp0 $null
```

Đoạn mã trên tạo agent UDP và gắn vào node n0 bằng thủ tục attach-agent. Tạo ra agent Null, hoạt động như một traffic sink và gắn vào node n1. Hai agent được nối lại bằng phương thức connect.

- **Tác tử TCP**

```
set tcp [new Agent/TCP]
set tcp_sink [new Agent/TCPSink]
$ns attach-agent $n0 $tcp
$ns attach-agent $n1 $tcp_sink
$ns connect $tcp $tcp_sink
```

Trong đoạn code trên thì NS-2 tạo ra tác tử TCP và gắn vào nút “tcp” nhờ phương thức “attach-agent”. Tác tử “TCPSink” hoạt động như một bộ chứa lưu lượng TCP (TCP sink) và gắn vào nút “tcp_sink”. Hai tác tử này được nối với nhau thông qua thủ tục “connect”.

Một số dịch vụ sinh dữ liệu cơ bản trong NS-2

Hai tác tử ứng dụng (Application) mô phỏng gửi lưu lượng trên đỉnh đối tượng TCP là “Application/FTP” và “Application/Telnet” sẽ sinh tương ứng lưu lượng theo các giao thức FTP và Telnet; còn với UDP là lưu lượng có tỷ lệ không đổi (Constant Bit Rate – CBR).

Giao thức truyền tệp File Transfer Protocol (FTP)

Code dùng cho giao thức FTP trong NS-2 là:


```
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at <time> "$ftp start"
```

d. Dò dấu vết (Tracing)

Trong NS-2, các sự kiện xảy ra (hay mọi hoạt động mạng) sẽ được *lưu dấu vết* (tracing). Nếu mô phỏng được chỉ trực tiếp đến các hoạt động trace (bằng các lệnh “*\$ns trace-all tên_file*” hay “*\$ns namtrace-all tên_file*”) thì kết quả sẽ có các tệp định dạng **.tr** và **.nam** tương ứng được tạo ra.

Tệp vết (trace file) có nội dung chứa tất cả các thông tin về các sự kiện (các hoạt động) xảy ra trong kịch bản mô phỏng như: thời gian gói tin đến (packet arrival), thời gian gói tin đi (packet departure), thời gian gói tin bị hủy (packet drop),..

Để theo vết các gói tin trên tất cả các liên kết thì dùng đoạn mã sau:

```
set trace_file [open out.tr w]
$ns trace-all $trace_file
$ns flush-trace
close $trace_file
```

Với mạng không dây thì muốn lưu vết ra file **.nam** cần bổ sung thêm dòng lệnh sau:

```

set namtrace [open simple-udp.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)

# set up topography object ( tạo topo và không gian mạng không
dây)
set topo [new Topography]

$topo load_flatgrid $val(x) $val(y)
# topo có kích thước x*y (ví dụ: 300 x 300 hoặc 400 x 500)
#tạo số node xuất hiện trong topo

create-god $val(nn)
# configure node (khởi tạo node không dây với các chuẩn cho trước)

$ns_ node-config -adhocRouting $val(rp) \
    -llType $val(ll) \
    -macType $val(mac) \
    -ifqType $val(ifq) \
    -ifqLen $val(ifqlen) \
    -antType $val(ant) \
    -propType $val(prop) \
    -phyType $val(netif) \
    -channelType $val(chan) \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace ON \
    -movementTrace OFF\

```

3.3. MÔ PHỎNG VÀ ĐÁNH GIÁ MỘT SỐ KỊCH BẢN MẠNG KHÔNG DÂY

Phần sau đây của luận văn sẽ xây dựng bốn kịch bản mô phỏng các trường hợp khác nhau của mạng không dây:

Kịch bản 1: mạng không dây di động (Mobile Adhoc NETWORK) đơn chặng, kịch bản này mô phỏng đơn giản việc truyền nhận giữa hai nút mạng không dây.

Kịch bản 2: mạng không dây di động (MANET) đa chặng, kịch bản này mô phỏng việc truyền nhận giữa hai nút nguồn và đích thông qua một nút trung gian, kịch bản này có ý nghĩa thực tế vì mạng không dây không có cơ sở hạ tầng (ad hoc) thì vị trí cặp nút gửi/nhận có thể ở xa ngoài phạm vi truyền sóng của nhau dẫn đến không truyền tin được, nên việc có thêm nút trung gian sẽ có ý nghĩa thực tế giúp các nút ở xa nhau vẫn có thể gửi nhận dữ liệu cho nhau.

Kịch bản 3: mạng không dây cho xe cộ (VANET): kịch bản này mô tả mạng xe cộ (Vehicular Ad hoc NETwork), loại hình mạng này mới hơn mạng MANET và dùng luôn chuẩn không dây IEEE 802.11e cho mạng dữ liệu đa phương tiện để gửi nhận các thông tin giữa các phương tiện tham gia giao thông.

Kịch bản 4: mạng không dây với dữ liệu đa phương tiện, gồm các loại dữ liệu voice, video, best-effort,... kịch bản mô phỏng nhằm quan sát sự khác biệt về mức độ ưu tiên giữa các loại dữ liệu đa phương tiện, tương ứng là tài nguyên – cụ thể là thông lượng dành cho các loại dữ liệu là khác nhau, với dữ liệu có độ ưu tiên cao hơn sẽ có được nhiều thông lượng hơn so với dữ liệu có độ ưu tiên thấp.

Với các kịch bản trên luận văn sử dụng một số cách đánh giá, phân tích kết quả như sau:

- Sử dụng phần mềm NAM để xem tệp vết dạng **.nam** sinh ra sau khi chạy mô phỏng, nếu các nút có thể gửi/nhận dữ liệu cho nhau thì hình ảnh chuyển động trên NAM sẽ cho thấy đường nối liên tục giữa các nút mạng, còn nếu không có đường kết nối thì có nghĩa các nút đang không thể trao đổi dữ liệu với nhau.
- Phân tích tệp vết dạng **.tr** sinh ra sau khi chạy mô phỏng, code phân tích tệp vết dạng này được viết bằng các ngôn ngữ thông dụng trên Linux như **awk** hoặc **Perl** để từ đó biểu diễn thành các tiêu chí chất

lượng mạng ví dụ như throughput (thông lượng) nhằm thể hiện việc truyền nhận dữ liệu qua thời gian như thế nào, cũng như mức độ thông lượng dành cho các loại dữ liệu đa phương tiện khác nhau ra sao..., thông tin được biểu diễn dưới dạng đồ thị để tiện quan sát.

3.3.1. Kịch bản thứ 1: mạng không dây ad hoc đơn chặng

Kịch bản này sẽ thực hiện những công việc sau:

- Tạo hai node, cài đặt tham số thích hợp cho từng node.
- Thêm agent TCP và TCP sink cho 2 node tương ứng 0 và 1.
- Thêm ứng dụng truyền tệp tin FTP vào agent TCP.
- Kết nối các agent và chạy mô phỏng trong 25s.
- Gọi NAM để minh họa cho hoạt động của topology này.

Nội dung Tcl (mã nguồn mô phỏng) của kịch bản 1 trên NS-2:

```
# test2node.tcl
set val(chan) Channel/WirelessChannel ;#channel type
set val(prop) Propagation/TwoRayGround ;#radio-propagation model
set val(netif) Phy/WirelessPhy ;# network interface type
set val(mac) Mac/802_11 ;# MAC type
set val(ifq) Queue/DropTail/PriQueue ;# interface queue type
set val(ll) LL ;# link layer type
set val(ant) Antenna/OmniAntenna ;# antenna model
set val(ifqlen) 50 ;# max packet in ifq
set val(nn) 2 ;# number of mobilenodes
set val(rp) AODV ;# routing protocol
=====
# Chương trình chính
=====
# Khởi tạo đối tượng
set ns [new Simulator]
set tracefd [open test2node.tr w]
$ns trace-all $tracefd
# Open the NAM trace file
set nf [open test2node.nam w]
$ns namtrace-all-wireless $nf 200 200

# set up topography object
set topo [new Topography]
$topo load_flatgrid 200 200
# tạo số node ( 2 node )
create-god $val(nn)
```

```

# configure nodes - cai dat tham so khong day cho cac node
    $ns node-config -adHocRouting $val(rp) \
        -llType $val(ll) \
        -macType $val(mac) \
        -ifqType $val(ifq) \
        -ifqLen $val(ifqlen) \
        -antType $val(ant) \
        -propType $val(prop) \
        -phyType $val(netif) \
        -channelType $val(chan) \
        -topoInstance $topo \
        -agentTrace ON \
        -routerTrace ON \
        -macTrace OFF \
        -movementTrace OFF
set node(0) [$ns node]
$node(0) random-motion 0
set node(1) [$ns node]
$node(1) random-motion 0
# disable random motion - khong cho cac node di chuyen
# Provide initial (X,Y, for now Z=0) co-ordinates for mobilenodes
$node(0) set X_ 0
$node(0) set Y_ 0
$node(0) set Z_ 0.0
$node(1) set X_ 100
$node(1) set Y_ 100
$node(1) set Z_ 0.0
for {set i 0} {$i < $val(nn)} {incr i}
{
    $ns initial_node_pos $node($i) 30
}
# Khoi tao kich thuc cho moi node tren topo
# Tạo ra một vài di chuyển trong topo
# Node(1) bắt đầu di chuyển đến Node(0)
$ns at 5.0 "$node(1) setdest 30.0 30.0 15.0"
# Node(1) sau đó lại di chuyển ra xa Node(0)
$ns at 10.0 "$node(1) setdest 199.0 199.0 20.0"

# Setup traffic flow between nodes
# TCP connections between node(0) and node(1)
# Ket noi TCP giữa 2 node ( node(0) truyen node(1) nhan )
# Set up TCP connection
set tcp [new Agent/TCP]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node(0) $tcp
$ns attach-agent $node(1) $sink
$ns connect $tcp $sink
# Set up a FTP over TCP connection
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP
$ns at 0.1 "$ftp start"
$ns at 25.0 "$ftp stop" # Ket thuc mo phong

for {set i 0} {$i < $val(nn) } {incr i} {
    $ns at 25.01 "$node($i) reset";
}

$ns at 25.01 "stop"
$ns at 25.01 "puts \"NS EXITING...\" ; $ns halt"
# Định nghĩa tiến trình "stop" - gọi NAM để thực hiện mô phỏng
proc stop {} {
    global ns nf tracefd
    $ns flush-trace
}

```

```

# Close the NAM trace file
  close $nf
  close $tracefd
# Excute NAM on the trace file
  exec nam test2node.nam &
  exit 0
}
puts "Starting Simulation..."
$ns run

```

Kết quả của kịch bản trên:

- Sau khi 2 node được tạo và được gắn các agent tương ứng, tiếp tục thực hiện các bước trong kịch bản
- Node(1) di chuyển gần tới Node(0) , 2 node nhận được tầm phủ sóng của nhau và bắt đầu truyền và nhận tín hiệu, sau đó thực hiện giao thức FTP
- Node(1) lại tiếp tục di chuyển ra xa Node(0), các gói tin vẫn gửi và nhận, tuy nhiên đến một khoảng cách nhất định, các gói tin đều bị mất, 2 node không thể nhận và gửi cho nhau được nữa.

Phân tích:

- Sử dụng file định .awk có chứa các lệnh trích xuất dữ liệu từ file .tr được sinh ra khi chạy NS-2. Một đoạn code trong file .awk, chủ yếu sử dụng các câu lệnh và hàm trong C:

```

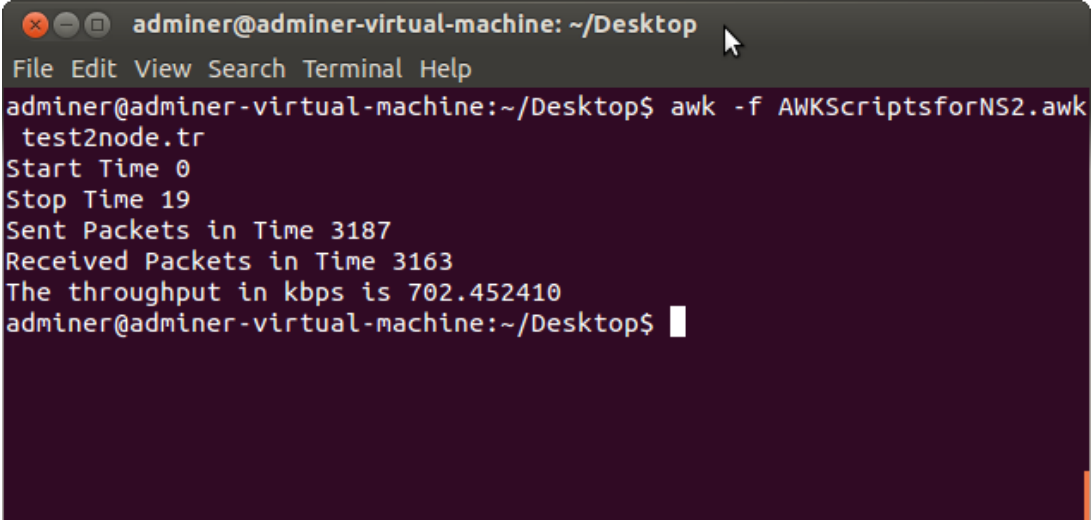
if (packet=="AGT" && sendTime[pkt_id]==0 && (event=="+" || event=="s")) {
    if (time < sTime) {
        sTime = time
    }
    sendTime[pkt_id] = time
    this_flow = flow_type
}

if (packet=="AGT" && event == "r") {
    if(time>spTime) {
        spTime=time
    }
    recv_size = recv_size + packet_size
    recvTime[pkt_id] = time
    NumOfRecd = NumOfRecd + 1
}

```

Hình 3.10: Một đoạn code trong tập tin phân tích

- Sử dụng cú pháp phân tích với awk: “*awk -f <AWK_file> <tr file>*”
Kết quả thu được:



```
adminer@adminer-virtual-machine: ~/Desktop
File Edit View Search Terminal Help
adminer@adminer-virtual-machine:~/Desktop$ awk -f AWKScriptsforNS2.awk
test2node.tr
Start Time 0
Stop Time 19
Sent Packets in Time 3187
Received Packets in Time 3163
The throughput in kbps is 702.452410
adminer@adminer-virtual-machine:~/Desktop$
```

Hình 3. 9. Kết quả phân tích

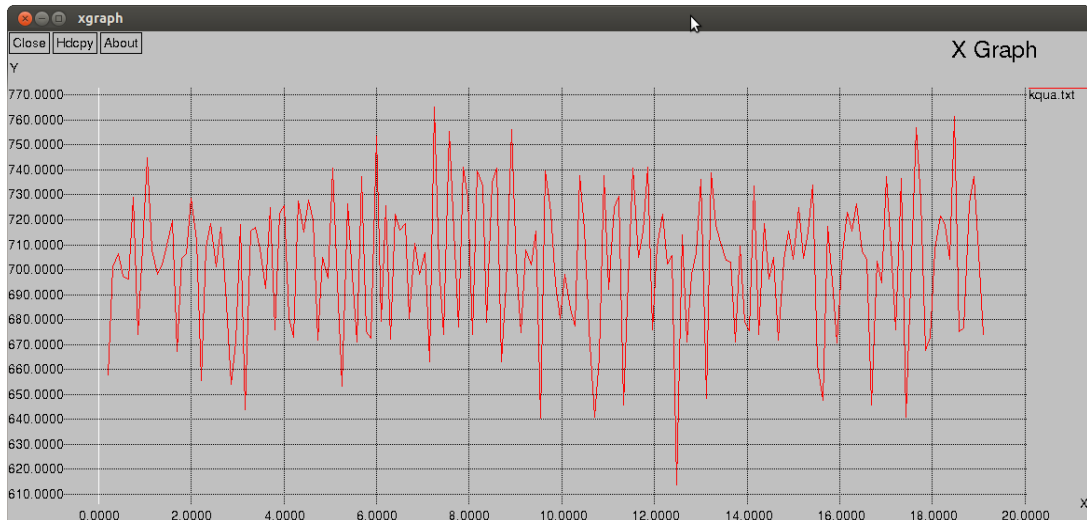
Với các thông số sau:

- *Start Time* : Thời gian bắt đầu chạy
- *Stop Time*: Thời gian 2 node dừng chạy
- *Sent Packets in Time*: Số gói tin gửi
- *Received Packets in Time*: Số gói tin nhận
- *The throughput in kbps is*: Bảng thông trung bình(kbit/s)

Sử dụng Xgraph để xem xét độ biến thiên của throughput hình thành qua biểu đồ sử dụng từ câu lệnh để trace ra file “kqua.txt” :

“*awk -f bienthienthp.awk test2node.tr > kqua.txt*” với *bienthienthp.awk* là một file phân tích khác cho phết in ra các giá trị throughput qua các thời điểm được ghi lại trong file dấu vết .tr .

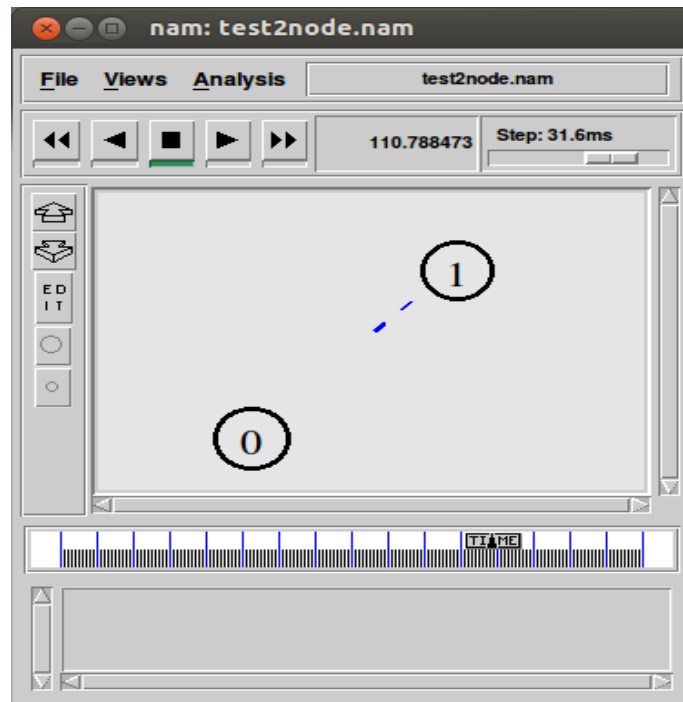
Gõ lệnh “*xgraph kqua.txt*” :



Hình 3. 10. Đồ thị biến thiên theo thời gian của throughput

Đánh giá: Từ đồ thị ta thấy được sự biến thiên của throughput qua các thời điểm khác nhau, rõ ràng ta thấy từ giây thứ 19 trở đi thì băng thông đã mất giữa 2 node mặc dù ta cho kịch bản chạy trong 25s, cho thấy đến khoảng cách vật lý quá xa thì 2 node sẽ không nhận thấy nhau và dừng truyền dữ liệu, đường đồ thị biến thiên cuối cùng đi xuống cho thấy băng thông giảm dần và biến mất trong khoảng 670-680, 2 node liên tiếp gửi các gói tin định tuyến sau 19s nhưng không nhận được phản hồi và chấm dứt kết nối hình thành mạng ngay sau đó.

Ý nghĩa: Mô hình trên thể hiện một mạng MANET cơ bản nhất, 2 thiết bị trong mạng tự thiết lập mạng để trao đổi dữ liệu khi chúng trong vùng sóng của nhau, cho tới khi cách nhau một khoảng cách nhất định thì mạng này sẽ bị hủy bỏ, đây là tính chất của mạng MANET đơn giản.



Hình 3. 11. Kết quả mô phỏng 1 qua NAM

3.3.2. Kịch bản thứ 2: mạng không dây ad hoc đa chặng

- Xuất phát từ kịch bản số 1, nếu node số 0 muốn định tuyến để gửi dữ liệu đến node số 1 nhưng khoảng cách quá xa, vậy phải làm thế nào, bài toán định tuyến xuất hiện, node 0 sẽ gửi thông tin đến các gói lân cận để tìm đường gửi dữ liệu đến cho node 1, ta đặt 1 node số 2 ở giữa để xem kết quả.
- Nội dung code Tcl của kịch bản:

```

#=====  

#      Simulation parameters setup  

#=====  

set val(chan) Channel/WirelessChannel ;#channel type  

set val(prop) | Propagation/TwoRayGround ;#radio-propagation model  

set val(netif) Phy/WirelessPhy ;#network interface type  

set val(mac) Mac/802_11 ;#MAC type  

set val(ifq) Queue/DropTail/PriQueue ;#interface queue type  

set val(ll) LL ;#link layer type  

set val(ant) Antenna/OmniAntenna ;#antenna model  

set val(ifqlen) 50 ;#max packet in ifq  

set val(nn) 3 ;#number of mobilenodes  

set val(rp) AODV ;#routing protocol  

set val(x) 510 ;#X dimension of topography  

set val(y) 309 ;#Y dimension of topography  

set val(stop) 21.0 ;#time of simulation end  

#=====  

#      Initialization  

#=====  

#Create a ns simulator  

set ns [new Simulator]  

$ns use-newtrace  

#Setup topography object  

set topo [new Topography]  

$topo load_flatgrid $val(x) $val(y)  

create-god $val(nn)  

#Open the NS trace file  

set tracefile [open test.tr w]  

$ns trace-all $tracefile  

#Open the NAM trace file  

set namfile [open test.nam w]  

$ns namtrace-all $namfile  

$ns namtrace-all-wireless $namfile $val(x) $val(y)  

set chan [new $val(chan)];#Create wireless channel  

#=====  

#      Mobile node parameter setup  

#=====  

$ns node-config -adhocRouting $val(rp) \  

               -llType $val(ll) \  

               -macType $val(mac) \  

               -ifqType $val(ifq) \  

               -ifqLen $val(ifqlen) \  

               -antType $val(ant) \  

               -propType $val(prop) \  

               -phyType $val(netif) \  

               -channel $chan \  

               -topoInstance $topo \  

               -agentTrace ON \  

               -routerTrace ON \  

               -macTrace ON \  

               -movementTrace ON  

#=====  

#      Nodes Definition  

#=====  

#Create 3 nodes  

set n0 [$ns node]  

$n0 set X_ 103

```

```

$n0 set Y_ 209
$n0 set Z_ 0.0
$ns initial_node_pos $n0 40
set n1 [$ns node]
$n1 set X_ 294
$n1 set Y_ 209
$n1 set Z_ 0.0
$ns initial_node_pos $n1 40
set n2 [$ns node]
$n2 set X_ 194
$n2 set Y_ 101
$n2 set Z_ 0.0
$ns initial_node_pos $n2 40

#=====
#           Generate movement
#=====
$ns at 0.1 " $n1 setdest 410 201 10 "

#=====
#           Agents Definition
#=====
#Setup a TCP connection
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n1 $sink1
$ns connect $tcp0 $sink1
$tcp0 set packetSize_ 1500
#=====
#           Applications Definition
#=====
#Setup a FTP Application over TCP connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 0.1 "$ftp0 start"
$ns at 20.0 "$ftp0 stop"

#=====
#           Termination
#=====
#Define a 'finish' procedure
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam test.nam &
    exit 0
}
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns at $val(stop) "\$n$i reset"
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run

```

Kết quả: Hai node đã gửi và nhận dữ liệu cho nhau đến lúc kịch bản kết thúc trong 20s.

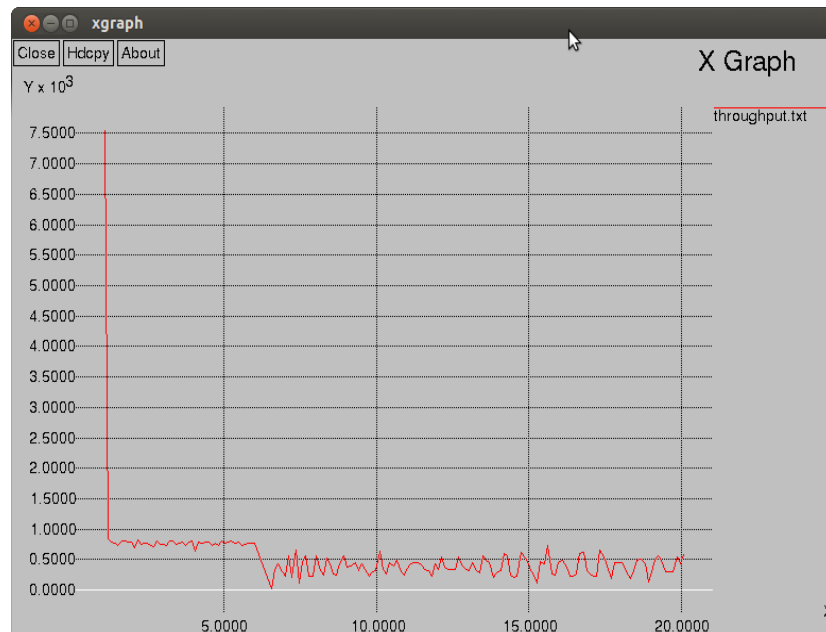
```

adminer@adminer-virtual-machine: ~/Desktop
File Edit View Search Terminal Help
adminer@adminer-virtual-machine:~/Desktop$ awk -f AWKScriptsforNS2.awk test.tr
Start Time 0
Stop Time 20
Sent Packets in Time 1597
Received Packets in Time 1577
The throughput in kbps is 488.538002
adminer@adminer-virtual-machine:~/Desktop$

```

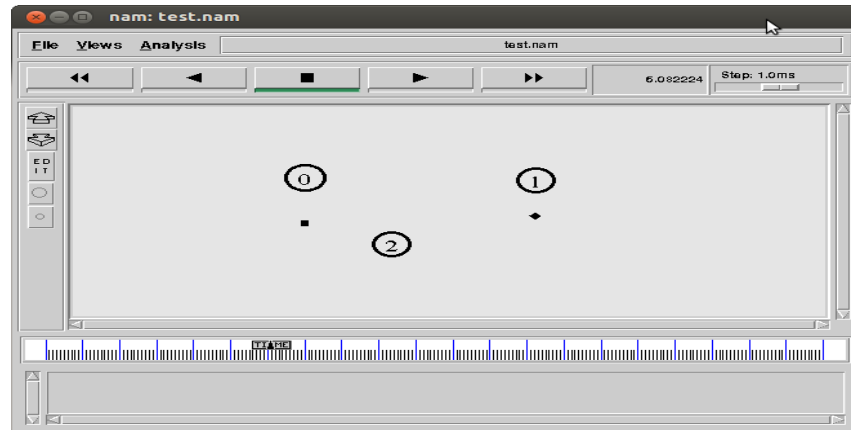
Hình 3. 12. Kết quả phân tích

Đồ thị biến thiên theo thời gian của throughput:



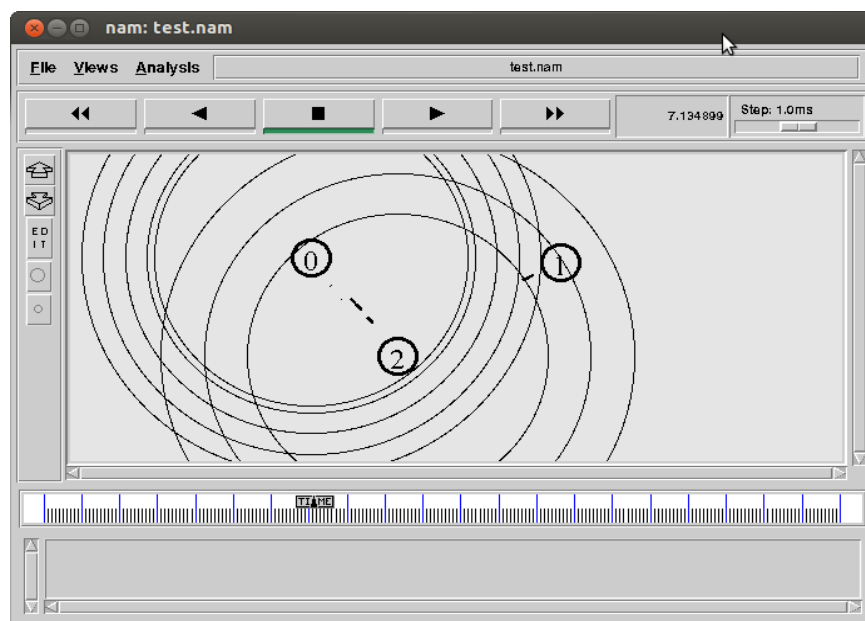
Hình 3. 13. Đồ thị biến thiên của throughput theo thời gian

Đánh giá: Trong khoảng thời gian node 1 di chuyển từ 5s ->10s (cụ thể là từ giây thứ 6) xuất hiện thời điểm khi mà node 1 cách xa node 0 và không thể gửi nhận dữ liệu, định tuyến giữa 2 node bị gián đoạn và bị drop gói tin.



Hình 3. 14. Gói tin bị drop

Thời điểm này, throughput giảm dần và bằng 0 để đợi node 0 tìm đường định tuyến trở lại, sau khi tìm được đường định tuyến nhờ gửi gói tin tìm đường qua node 2, node 0 tiếp tục gửi dữ liệu theo bảng định tuyến qua node 2 để đến đích là node 1, kịch bản chạy liên tục cho đến khi kết thúc, tuy nhiên tốc độ băng thông sẽ giảm sau khi qua node 2.



Hình 3. 15. Kết quả mô phỏng 2 qua NAM

3.3.3. Kịch bản thứ 3: mạng không dây cho xe cộ

- Về nội dung kịch bản mô phỏng cơ bản về mạng VANET, 2 xe muốn giao tiếp với nhau để xem tình trạng đường của bên kia. Ta dựng kịch bản cho các xe di chuyển và trạm RSU ở giữa làn đường 2 xe n1 và n4,

nếu muốn giao tiếp được với nhau thì 2 xe phải giao tiếp qua các xe lân cận và qua các trạm RSU bên đường bằng cách định tuyến, cụ thể là giao thức AODV để đảm bảo dữ liệu chuyển đi nhanh và chính xác nhất.

- Nội dung kịch bản trên NS-2:

```
# Simulation parameters setup
#=====
set val(chan) Channel/WirelessChannel ;# channel type
set val(prop) Propagation/TwoRayGround ;# radio-propagation model
set val(netif) Phy/WirelessPhy ;# network interface type
set val(mac) Mac/802_11 ;# MAC type
set val(ifq) Queue/DropTail/PriQueue ;# interface queue type
set val(ll) LL ;# link layer type
set val(ant) Antenna/OmniAntenna ;# antenna model
set val(ifqlen) 50 ;# max packet in ifq
set val(nn) 7 ;# number of mobilenodes
set val(rp) AODV ;# routing protocol
set val(x) 1288 ;# X dimension of topography
set val(y) 550 ;# Y dimension of topography
set val(stop) 22.0 ;# time of simulation end

#=====
# Initialization
#=====
#Create a ns simulator
set ns [new Simulator]

#Setup topography object
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)

#Open the NS trace file
$ns use-newtrace
set tracefile [open VANET.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open VANET.nam w]
$ns namtrace-all $namfile
$ns namtrace-all-wireless $namfile $val(x) $val(y)
set chan [new $val(chan)];#Create wireless channel

#=====
# Mobile node parameter setup
#=====
$ns node-config -adhocRouting $val(rp) \
                -llType $val(ll) \
                -macType $val(mac) \
                -ifqType $val(ifq) \
                -ifqLen $val(ifqlen) \
                -antType $val(ant) \
                -propType $val(prop) \
```

```

        -phyType      $val(netif) \
        -channel      $chan \
        -topoInstance $topo \
        -agentTrace   ON \
        -routerTrace  ON \
        -macTrace     ON \
        -movementTrace ON

#=====
#           Nodes Definition
#=====
#Create 7 nodes
set n0 [$ns node]
$n0 set X_ 487
$n0 set Y_ 287
$n0 set Z_ 0.0
$ns initial_node_pos $n0 20
set n1 [$ns node]
$n1 set X_ 260
$n1 set Y_ 127
$n1 set Z_ 0.0
$ns initial_node_pos $n1 20
set n2 [$ns node]
$n2 set X_ 384
$n2 set Y_ 184
$n2 set Z_ 0.0
$ns initial_node_pos $n2 20
set n3 [$ns node]
$n3 set X_ 534
$n3 set Y_ 184
$n3 set Z_ 0.0
$ns initial_node_pos $n3 20
set n4 [$ns node]
$n4 set X_ 488
$n4 set Y_ 441
$n4 set Z_ 0.0
$ns initial_node_pos $n4 20
set n7 [$ns node]
$n7 set X_ 781
$n7 set Y_ 313
$n7 set Z_ 0.0
$ns initial_node_pos $n7 20
set n8 [$ns node]
$n8 set X_ 1004
$n8 set Y_ 450
$n8 set Z_ 0.0
$ns initial_node_pos $n8 20

#=====
#           Generate movement
#=====
$ns at 1 " $n1 setdest 1046 135 20"
$ns at 1 " $n4 setdest 1004 450 10"
# speed in setdest ?
#=====
#           Agents Definition
#=====
#Setup a TCP connection
set tcp0 [new Agent/TCP]

```

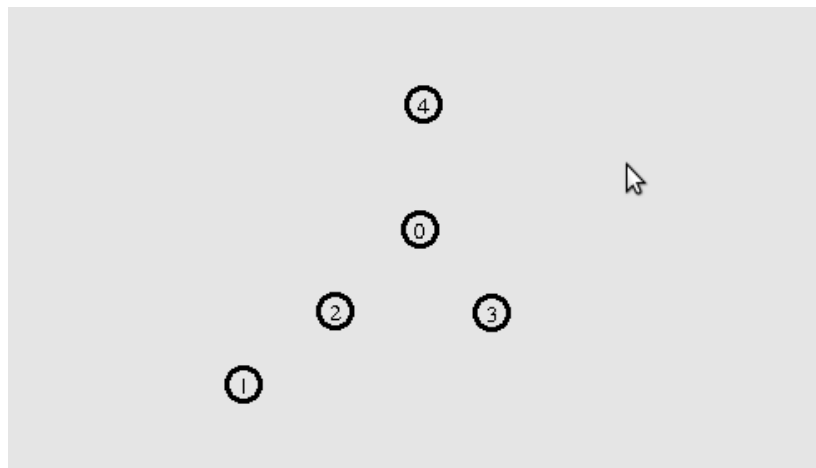
```

$ns attach-agent $n1 $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n4 $sink1
$ns connect $tcp0 $sink1
$tcp0 set packetSize_ 1500

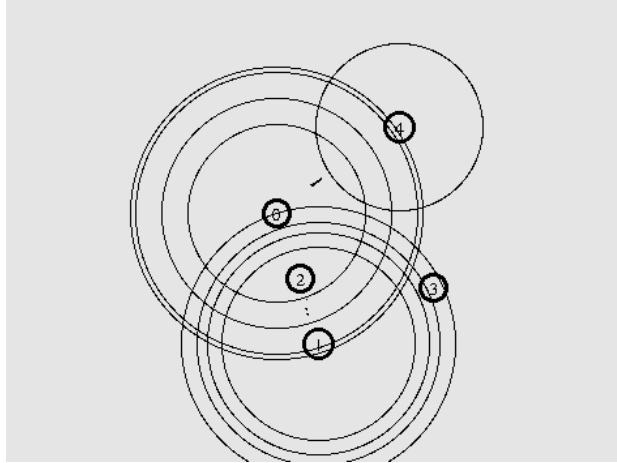
#=====
#           Applications Definition
#=====
#Setup a FTP Application over TCP connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 1.0 "$ftp0 start"
$ns at 20.0 "$ftp0 stop"

#=====
#           Termination
#=====
#Define a 'finish' procedure
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam VANET.nam &
    exit 0
}
for {set i 0} {$i < 4 } { incr i } {
    $ns at $val(stop) "\$n$i reset"
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run

```



Hình 3. 16. Mô hình VANET cơ bản trên NS2



Hình 3. 17. Chạy mô phỏng với 2 xe đang di chuyển trên NS2

- Sử dụng file phân tích “analyze.pl” với hàm perl để phân tích kết quả sinh ra từ kịch bản.

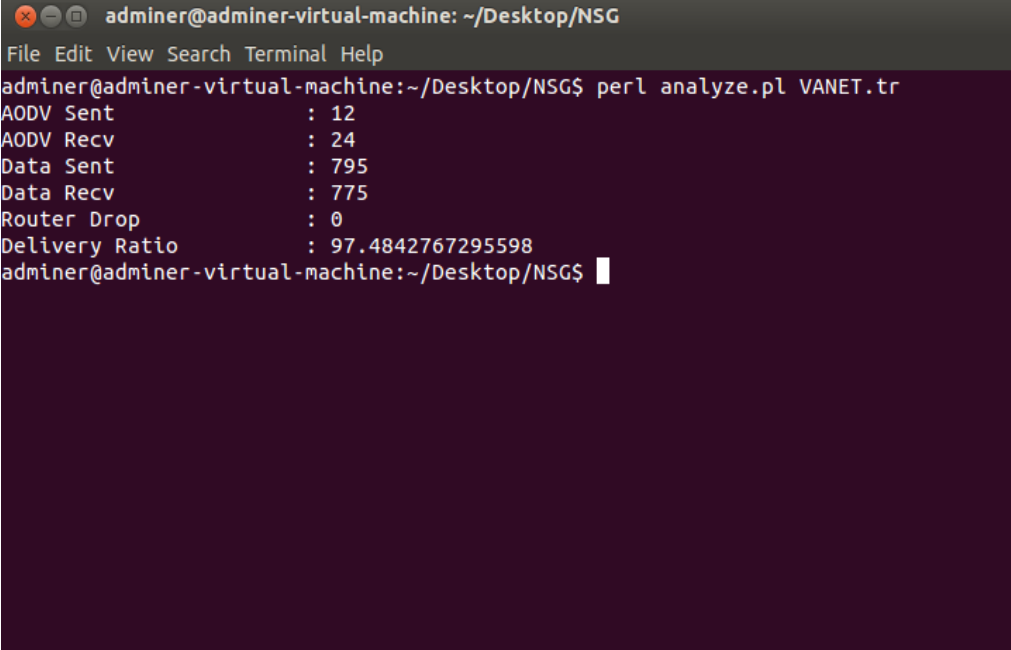
```

analyze.pl ✖
1 #!/usr/bin/perl
2 $tracefile=$ARGV[0];
3 $granularity=$ARGV[1];
4
5 $ofile="simulation_result.csv";
6 open OUT, ">$ofile" or die "$0 cannot open output file $ofile: $!";
7
8 open (DR,STDIN);
9 $gclock=0;
10
11 #Data Packet Information
12 $dataSent = 0;
13 $dataRecv = 0;
14 $routerDrop = 0;
15
16 #AODV Packet Information
17 $aodvSent = 0;
18 $aodvRecv = 0;
19
20 $aodvSentRequest = 0;
21 $aodvRecvRequest = 0;
22 $aodvDropRequest = 0;

```

Hình 3. 18. Một đoạn code mẫu của file perl

Gõ câu lệnh: “*perl <analyze.pl> <tracefile.tr>*” ta thu được kết quả:



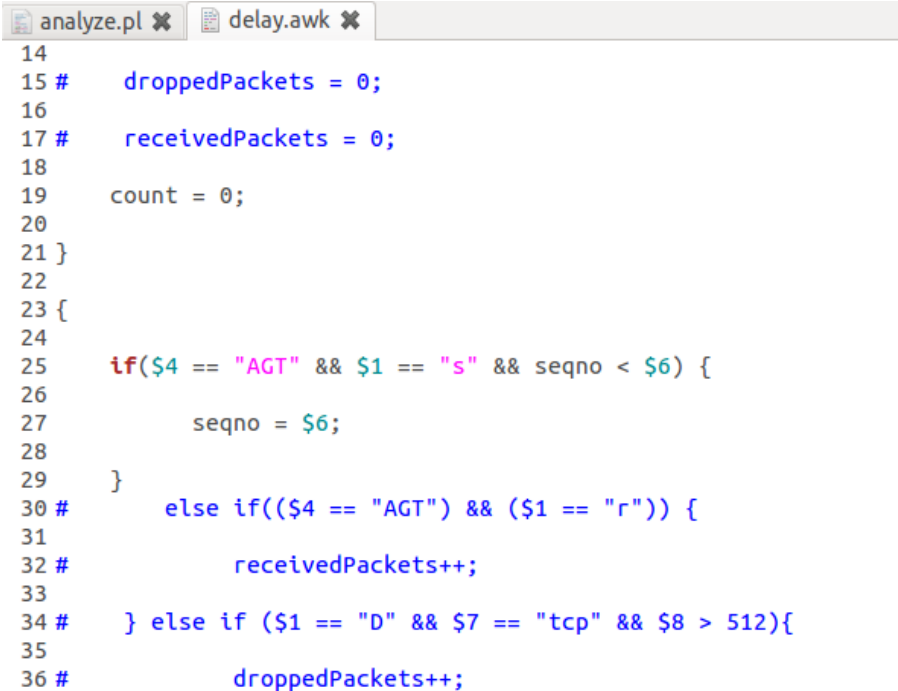
```

adminer@adminer-virtual-machine: ~/Desktop/NSG
File Edit View Search Terminal Help
adminer@adminer-virtual-machine:~/Desktop/NSG$ perl analyze.pl VANET.tr
AODV Sent      : 12
AODV Recv     : 24
Data Sent     : 795
Data Recv    : 775
Router Drop   : 0
Delivery Ratio : 97.4842767295598
adminer@adminer-virtual-machine:~/Desktop/NSG$

```

Hình 3. 19. Kết quả thu thập từ NS2 qua perl

- File .pl có nhiệm vụ đọc và tổng hợp theo lập trình các thông số thu thập được trong tracefile của NS2 như Data Sent, Data Receive, Router Drop, Delivery Ratio,...
- Sử dụng file delay.awk để biết độ trễ trung bình của gói tin gửi và nhận.



```

analyze.pl ✕ delay.awk ✕
14
15 #   droppedPackets = 0;
16
17 #   receivedPackets = 0;
18
19   count = 0;
20
21 }
22
23 {
24
25   if($4 == "AGT" && $1 == "s" && seqno < $6) {
26
27       seqno = $6;
28
29   }
30 #   else if(($4 == "AGT") && ($1 == "r")) {
31
32       receivedPackets++;
33
34 #   } else if ($1 == "D" && $7 == "tcp" && $8 > 512){
35
36 #       droppedPackets++;

```

Hình 3. 20. Một đoạn code của file awk

- Kết quả thu được:

```

adminer@adminer-virtual-machine: ~/Desktop/NSG
File Edit View Search Terminal Help
adminer@adminer-virtual-machine:~/Desktop/NSG$ awk -f delay.awk VANET.tr

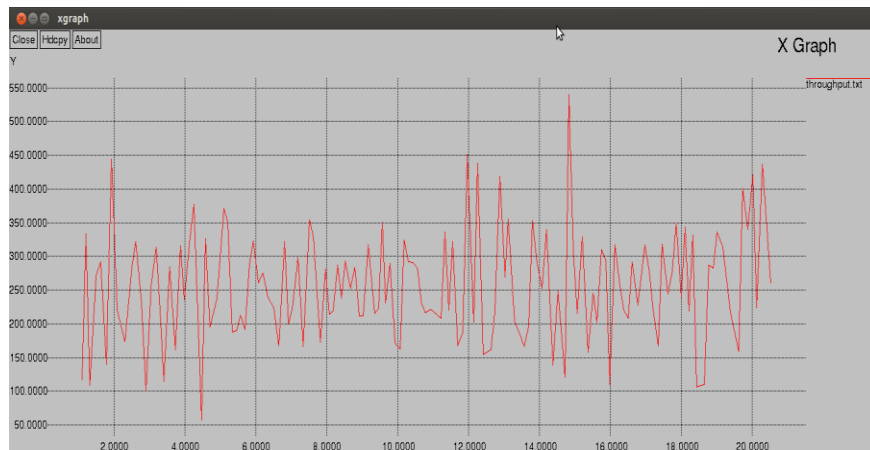
GeneratedPackets          = 795
Packet Delivery Ratio     = 0
Average End-to-End Delay  = 449.159 ms

adminer@adminer-virtual-machine:~/Desktop/NSG$

```

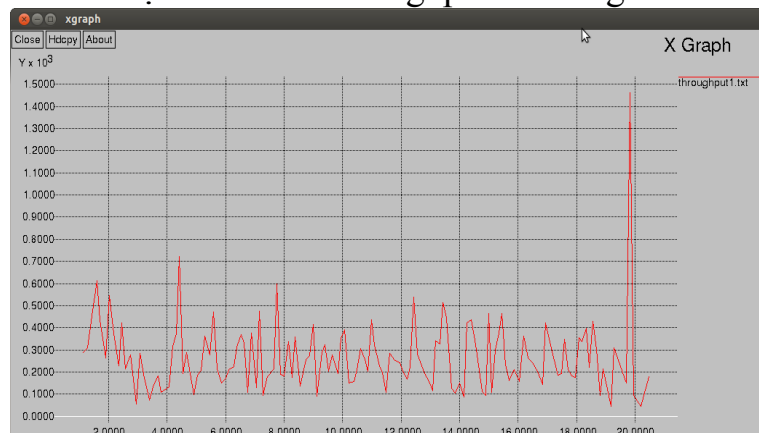
Hình 3. 21. Phân tích kết quả từ NS2 qua awk

- Đồ thị biểu diễn sự biến thiên của throughput data nhận dữ liệu(node số 4):



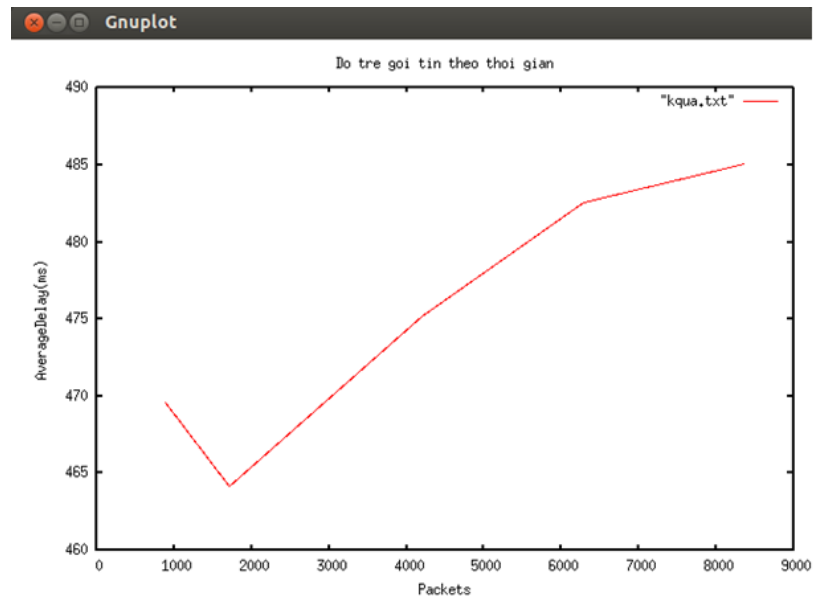
Hình 3. 22. Biến thiên throughput data nhận

- Đồ thị biểu diễn sự biến thiên throughput của xe gửi tin:

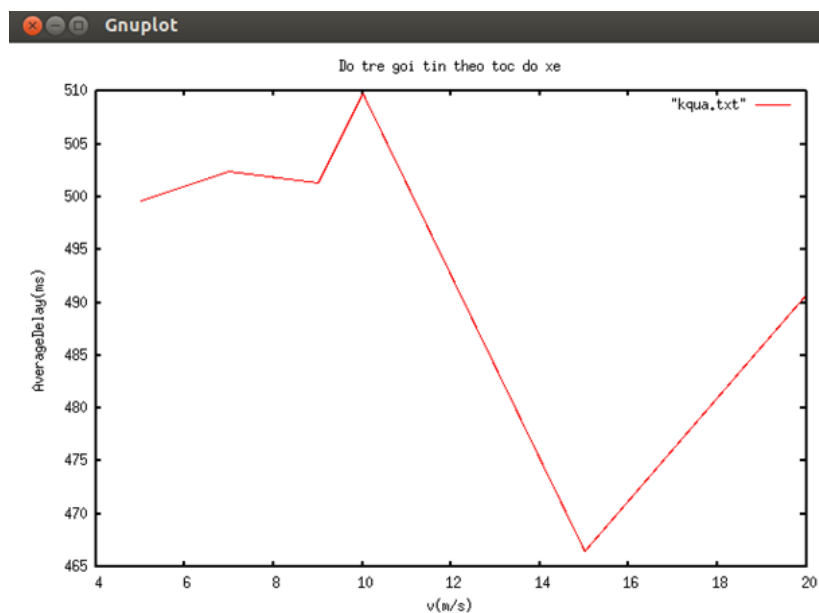


Hình 3. 23. Biến thiên throughput Data gửi

- Sử dụng và thay đổi giá trị thời gian truyền gói tin, ta thu được kết quả phân tích, sử dụng Gnuplot để hiển thị:



Hình 3. 24. Đồ trễ gói tin theo thời gian



Hình 3. 25. Đồ trễ gói tin theo tốc độ xe

- Kết luận: Nhìn đồ thị ta thấy được độ trễ gói tin, tỷ lệ phân phát gói tin đến đích và thông lượng throughput gửi và nhận phụ thuộc vào nhiều yếu tố như tốc độ xe di chuyển trên đường, thời gian phát gói tin, tuy nhiên sự biến thiên trên được coi là chấp nhận được với tốc độ xe đi

nhanh và đảm bảo được dữ liệu được truyền đi nhanh và chính xác nhất có thể.

3.3.4. Kịch bản thứ 4: mạng không dây cho dữ liệu đa phương tiện

- Kịch bản sau sẽ thực hiện mô phỏng và đánh giá chất lượng dịch vụ truyền phát đa phương tiện trong mạng không dây cục bộ WLAN với chuẩn không dây 802.11e.
- Để có thể chạy được chuẩn đa phương tiện ta phải tích hợp giao thức chuẩn 802.11e vào trong NS-2, truy cập file thư mục gốc chứa ứng dụng NS-2 và thêm một vài thông số như sau:
 - Mở file Makefile.in trong thư mục ./ns-2.35
 - thêm vào đối tượng INCLUDES:
 - -I./mac/802_11e
 - thêm vào OBJ_CC:
 - mac/802_11e/mac-802_11e.o mac/802_11e/priq.o
 - mac/802_11e/d-tail.o mac/802_11e/mac-timers_802_11_e.o
 - xóa trong đối tượng NS_TCL_LIB:
 - tcl/lib/ns-mobilenode.tcl \
 - thêm vào NS_TCL_LIB:
 - mac/802_11e/ns-mobilenode_EDCA.tcl \
 - mac/802_11e/priority.tcl \
 - Thay đổi file ns-lib.tcl trong đường dẫn ./ns-2.35/tcl/lib/
 - xóa dòng sau từ source list:
 - source ns-mobilenode.tcl
 - sau đó thêm vào:
 - source ../../mac/802_11e/ns-mobilenode_EDCA.tcl
 - source ../../mac/802_11e/priority.tcl
 - Thay đổi file ns-default.tcl trong đường dẫn ./ns-2.35/tcl/lib/
 - Queue/DTail set drop_front_ false
 - Queue/DTail set summarystats_ false
 - Queue/DTail set queue_in_bytes_ false
 - Queue/DTail set mean_pktsize_ 500

- Queue/DTail/PriQ set Prefer_Routing_Protocols 1
- Queue/DTail/PriQ set Max_Levels 4
- Queue/DTail/PriQ set Levels 4
- Mac/802_11e set SlotTime_ 0.000020 ;# 20us
- Mac/802_11e set SIFS_ 0.000010 ;# 10us
- Mac/802_11e set PreambleLength_ 144 ;# 144 bit
- Mac/802_11e set PLCPHeaderLength_ 48 ;# 48 bits
- Mac/802_11e set PLCPDataRate_ 1.0e6 ;# 1Mbps
- Mac/802_11e set RTSThreshold_ 3000 ;# bytes
- Mac/802_11e set ShortRetryLimit_ 7 ;#etransmittions
- Mac/802_11e set LongRetryLimit_ 4 ;# retransmissions

○ Thay đổi file ns-mac.tcl trong đường dẫn ./ns-2.35/tcl/lan:

- if [TclObject is-class Mac/802_11e] {
 - Mac/802_11e set delay_ 64us
 - Mac/802_11e set ifs_ 16us
 - Mac/802_11e set slotTime_ 16us
 - Mac/802_11e set cwmin_ 16
 - Mac/802_11e set cwmax_ 1024
 - Mac/802_11e set rtxLimit_ 16
 - Mac/802_11e set bssId_ -1
 - Mac/802_11e set sifs_ 8us
 - Mac/802_11e set pifs_ 12us
 - Mac/802_11e set difs_ 16us
 - Mac/802_11e set rtxAckLimit_ 1
 - Mac/802_11e set rtxRtsLimit_ 3
 - Mac/802_11e set basicRate_ 1Mb ;
 - Mac/802_11e set dataRate_ 1Mb ;
 - Mac/802_11e set cfb_ 0 ;# disables CFB }

○ Tại file wireless.phy.h trong đường dẫn: ./ns-2.35/mac:

- **đổi** “enum ChannelStatus {IDLE, RECV, SEND};”
thành “enum ChannelStatus {IDLE, RECVING, SENDING};”

Nội dung kịch bản viết bằng ngôn ngữ Tcl trong NS-2:

```

# Define Node Configuration paramaters
set val(chan) Channel/WirelessChannel |;# channel type
set val(prop) Propagation/TwoRayGround;# radio-propagation
model
set val(netif) Phy/WirelessPhy ;# network interface type
set val(mac) Mac/802_11e ;# MAC type
set val(ifq) Queue/DTail/PriQ ;# interface queue type
set val(ll) LL ;# link layer type
set val(ant) Antenna/OmniAntenna ;# antenna model
set val(ifqlen) 50 ;# max packet in ifq
set val(nn) 3 ;# number of mobilenodes
set val(rp) AODV ;# routing protocol
set val(x) 500 ;# X dimension of the topography
set val(y) 500 ;# Y dimension of the topography
set val(stop) 50 ;#End simulation
set val(stops) 50.1 ;# End NS-2
set maxvi 1 ;# Number of Voice flows
set maxvo 1 ;# Number of Video flows
set maxbe 1 ;# Number of BestEffort flows
set maxbk 1 ;# Number of Background flows

Mac/802_11e set dataRate_ 2Mb
Mac/802_11e set BackoffDelay 20.0
Mac/802_11e set EstimationPeriod 2.0

# Initialize trace file descriptions

# *** Throughput Trace ***
set fvi [open Voice.tr w]
set fvo [open Video.tr w]
set fbe [open BestEffort.tr w]
set fbk [open Background.tr w]

# *** Initialize Simulator ***
set ns_ [new Simulator]

# *** Initialize Trace file ***
set tracefd [open trace.tr w]
$ns_ use-newtrace
$ns_ trace-all $tracefd

# *** Initialize Network Animator ***
set namtrace [open nam.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)

# *** set up topography object ***
set topo [new Topography]
$topo load_flatgrid 500 500

```

```

# Create General Operations Director (GOD) object.
# It is used to store global information about the state of the
environment, network, or nodes that an
# omniscient observer would have, but that should not be made known to
any participant in the simulation.
create-god $val(nn)
# configure nodes
$ns_ node-config -adhocRouting $val(rp) \
                -llType $val(ll) \
                -macType $val(mac) \
                -ifqType $val(ifq) \
                -ifqLen $val(ifqlen) \
                -antType $val(ant) \
                -propType $val(prop) \
                -phyType $val(netif) \
                -channelType $val(chan) \
                -topoInstance $topo \
                -agentTrace OFF \
                -routerTrace OFF \
                -macTrace ON \
                -movementTrace OFF

# Create Nodes
for {set i 0} {$i < $val(nn) } {incr i} {
    set node_($i) [$ns_ node]
    $node_($i) random-motion 0; # disable random motion
}

# Initialize Node Coordinates
$node_(0) set X_ 5.0
$node_(0) set Y_ 5.0
$node_(0) set Z_ 0.0

$node_(1) set X_ 50.0
$node_(1) set Y_ 5.0
$node_(1) set Z_ 0.0

$node_(2) set X_ 95.0
$node_(2) set Y_ 5.0
$node_(2) set Z_ 0.0

# Setup traffic flow between nodes
# Create Constant four Bit Rate Traffic sources
# Voice (VI) flows
for {set vi 0} {$vi < $maxvi } {incr vi} {
    set agentvi($vi) [new Agent/UDP]
    $agentvi($vi) set prio_ 0
    set sinkvi($vi) [new Agent/LossMonitor]
    $ns_ attach-agent $node_(0) $agentvi($vi)
    $ns_ attach-agent $node_(2) $sinkvi($vi)
    $agentvi($vi) set fid_ $vi
    $ns_ connect $agentvi($vi) $sinkvi($vi)
    set appvi($vi) [new Application/Traffic/CBR]
    $appvi($vi) set packetSize_ 512
    $appvi($vi) set rate_ 2000Kb
    $appvi($vi) attach-agent $agentvi($vi)
}

```



```

# Video (VO) flows
for {set vo 0} {$vo < $maxvo } {incr vo} {
    set agentvo($vo) [new Agent/UDP]
    $agentvo($vo) set prio_ 1
    set sinkvo($vo) [new Agent/LossMonitor]
    $ns_ attach-agent $node_ (0) $agentvo($vo)
    $ns_ attach-agent $node_ (2) $sinkvo($vo)
    $agentvo($vo) set fid_ [expr {$vo+$maxvi}]
    $ns_ connect $agentvo($vo) $sinkvo($vo)
    set appvo($vo) [new Application/Traffic/CBR]
    $appvo($vo) set packetSize_ 512
    $appvo($vo) set rate_ 2000Kb
    $appvo($vo) attach-agent $agentvo($vo)
}

# BestEffort (BE) flows
for {set be 0} {$be < $maxbe } {incr be} {
    set agentbe($be) [new Agent/UDP]
    $agentbe($be) set prio_ 2
    set sinkbe($be) [new Agent/LossMonitor]
    $ns_ attach-agent $node_ (0) $agentbe($be)
    $ns_ attach-agent $node_ (2) $sinkbe($be)
    $agentbe($be) set fid_ [expr {$be+$maxvo+$maxvi}]
    $ns_ connect $agentbe($be) $sinkbe($be)
    set appbe($be) [new Application/Traffic/CBR]
    $appbe($be) set packetSize_ 512
    $appbe($be) set rate_ 2000Kb
    $appbe($be) attach-agent $agentbe($be)
}

if 0 {
# Background (BK) flows
for {set bk 0} {$bk < $maxbk } {incr vi} {
    set agentbk($bk) [new Agent/UDP]
    $agentbk($bk) set prio_ 3
    set sinkbk($bk) [new Agent/LossMonitor]
    $ns_ attach-agent $node_ (0) $agentbk($bk)
    $ns_ attach-agent $node_ (2) $sinkbk($bk)
    $agentbk($bk) set fid_ $bk+$be+$vo+$vi
    $ns_ connect $agentbk($bk) $sinkbk($bk)
    set appbk($bk) [new Application/Traffic/CBR]
    $appbk($bk) set packetSize_ 512
    $appbk($bk) set rate_ 2000Kb
    $appbk($bk) attach-agent $agentbk($bk)
}
}

# defines the node size in Network Animator
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns_ initial_node_pos $node_ ($i) 20
}

# Initialize Flags
set holdratevi 0
set holdratevo 0
set holdratebe 0

set throughputvi 0
set throughputvo 0

```

```

set throughputbe 0

# Function To record Statistics (Bit Rate, Delay, Drop)
proc record {} {
    global throughputvi throughputvo throughputbe
    global fvi fvo fbe fbk
    global holdratevi holdratevo holdratebe
    global maxvi maxvo maxbe
    global sinkvi sinkvo sinkbe

    set ns [Simulator instance]
    set time 0.9; # Set Sampling Time to 0.9 Sec
    set now [$ns now]
    # Calculate voice (VI) throughput
    for {set vi 0} {$vi < $maxvi } {incr vi} {
        set thvi [$sinkvi($vi) set bytes_]
        puts $fvi "$now [expr
(($thvi+$holdratevi)*8)/(2*$time*1000000)]"
        set throughputvi [expr
{$throughputvi+((($thvi+$holdratevi)*8)/(2*$time*1000000))]
    }

    # Calculate video (VO) throughput
    for {set vo 0} {$vo < $maxvo } {incr vo} {
        set thvo [$sinkvo($vo) set bytes_]
        puts $fvo "$now [expr
(($thvo+$holdratevo)*8)/(2*$time*1000000)]"
        set throughputvo [expr
{$throughputvo+((($thvo+$holdratevo)*8)/(2*$time*1000000))]
    }

    # Calculate besteffort (BE) throughput
    for {set be 0} {$be < $maxbe } {incr be} {
        set thbe [$sinkbe($be) set bytes_]
        puts $fbe "$now [expr
(($thbe+$holdratebe)*8)/(2*$time*1000000)]"
        set throughputbe [expr
{$throughputbe+((($thbe+$holdratebe)*8)/(2*$time*1000000))]
    }

    # Reset Variables
    for {set vi 0} {$vi < $maxvi } {incr vi} {
        $sinkvi($vi) set bytes_ 0
    }
    for {set vo 0} {$vo < $maxvo } {incr vo} {
        $sinkvo($vo) set bytes_ 0
    }
    for {set be 0} {$be < $maxbe } {incr be} {
        $sinkbe($be) set bytes_ 0
    }

    set holdratevi $thvi
    set holdratevo $thvo
    set holdratebe $thbe

    $ns at [expr $now+$time] "record"; # Schedule Record after $time
    interval sec
}

```

```

# Start Recording at Time 0
$ns_ at 0.0 "record"

# Start Voice (vi) flows
for {set vi 0} {$vi < $maxvi } {incr vi} {
    $ns_ at $vi "$appvi($vi) start"
}

# Start Video (vo) flows
for {set vo 0} {$vo < $maxvo } {incr vo} {
    $ns_ at $vo "$appvo($vo) start"
}

# Start BestEffort (be) flows
for {set be 0} {$be < $maxbe } {incr be} {
    $ns_ at $be "$appbe($be) start"
}

# Stop Simulation
$ns_ at $val(stop) "stop"

# Reset Nodes
for {set i 0} {$i < $val(nn) } {incr i} {
    $ns_ at $val(stop) "$node_($i) reset";
}

# Exit Simulatoion at Time 80.01 sec
$ns_ at $val(stops) "puts \"NS EXITING...\" ; $ns_ halt"

proc stop {} {
    global throughputvi throughputvo throughputbe
    global ns_ tracefd fvi fvo fbe fbk

    # Close Trace Files
    close $fvo
    close $fvi
    close $fbe
    close $fbk

    # Print flows traffic
    puts "VI throughput: $throughputvi Kbps"
    puts "VO throughput: $throughputvo Kbps"
    puts "BE throughput: $throughputbe Kbps"

    # Plot Recorded Statistics
    exec /home/ns-allinone-2.35/xgraph-12.2/xgraph Voice.tr Video.tr
    BestEffort.tr Background.tr -geometry 900x450 -t "Throughput" &

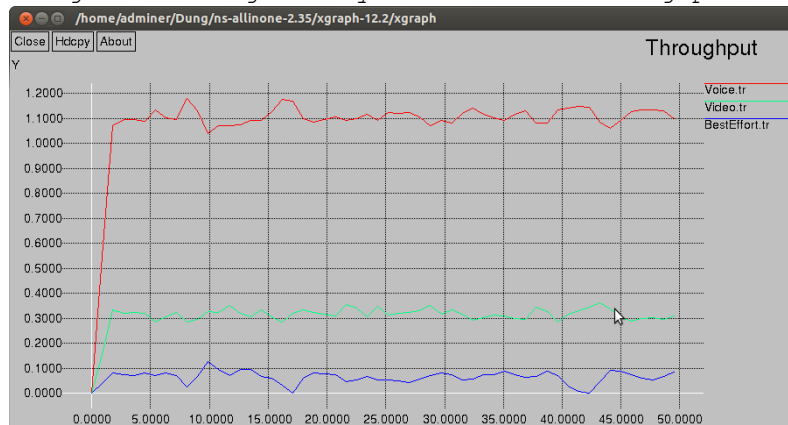
    # Reset Trace File
    $ns flush-trace
    close $tracefd
    exit 0
}

#Ramdomize seed number
$defaultRNG seed 0

puts "Starting Simulation..."

```

Kết quả chạy kịch bản mô phỏng như mã nguồn ở trên, đồ thị được biểu diễn bởi phần mềm vẽ đồ thị xgraph – được gọi luôn trong đoạn code ở trên ở đoạn “`exec /home/adminer/ns-allinone-2.35/xgraph-12.2/xgraph Voice.tr Video.tr BestEffort.tr Background.tr -geometry 900x450 -t "Throughput"`”:



Hình 3. 26. Sự biến thiên thông lượng theo thời gian trong 802.11e

Đánh giá: Từ kết quả phân tích thu được cho thấy thông lượng của Video và Voice luôn được đảm bảo và đều nằm trong khoảng tương ứng là 0.3 Kbps và 1.1 Kbps tuy nhỏ nhưng luôn ổn định, độ thăng giáng lưu lượng theo thời gian là nhỏ, đảm bảo ứng dụng. Độ ưu tiên khác nhau rõ rệt với từng loại thông lượng, Voice luôn được ưu tiên cao nhất, sau đó là chất lượng hình ảnh Video và thông lượng của BestEffort(môi trường truyền tải) luôn được giới hạn để tăng cho các phương tiện còn lại có độ ưu tiên cao hơn.

KẾT LUẬN VÀ KIẾN NGHỊ

Thông qua việc tìm hiểu về mạng không dây đặc biệt là mạng cục bộ không dây WLAN và mô phỏng được chúng qua NS2, học viên đã có được các kiến thức cơ bản về mạng cục bộ không dây, về các chuẩn, cấu trúc mạng và các vấn đề về bảo mật cho mạng Wifi. Việc phát triển mạng không dây thật sự đem lại hiệu quả với sự thuận lợi khi sử dụng các thiết bị có tính di động cao, và hơn nữa còn có thể mô phỏng lại chúng thực tế trên NS2 để kiểm tra.

Luận văn đã hoàn thành các nội dung nghiên cứu và đã đạt được một số kết quả về lý thuyết và thực nghiệm như sau:

Về Lý thuyết: Đã nghiên cứu được lý thuyết về truyền thông đa phương tiện, nghiên cứu và chất lượng dịch vụ (QoS) trên mạng không dây, một số phương pháp đảm bảo chất lượng dịch vụ.

Về thực nghiệm: Nghiên cứu đã phân tích và chọn lựa phương pháp đánh giá chủ quan để đánh giá chất lượng trải nghiệm trên mạng không dây. Từ đó xác định được quan hệ giữa các tham số QoS của mạng không dây. Các giá trị này được làm bộ giá trị tham khảo để phục vụ việc quản lý chất lượng dịch vụ. Luận văn cũng đã khái quát chung về dịch vụ mạng không dây như đặc điểm về cấu trúc, công nghệ, dịch vụ và chất lượng dịch vụ. Trên cơ sở các đặc điểm của dịch vụ mạng không dây nói chung, tác giả đã đi sâu nghiên cứu phân tích ảnh hưởng của các tham số QoS đến chất lượng dịch vụ và đề xuất giải pháp quản lý chất lượng dịch vụ hiệu quả.

Việc xây dựng, thiết kế một mạng hết sức khó khăn nhất là tính thực tế. Dựa trên quy mô thực tế của hệ thống mạng cần xây dựng cũng như đòi hỏi về tính thực tế mà chúng ta thiết lập những cơ sở và chính khác nhau. Tùy thuộc vào là mạng gia đình, trường học, điểm truy cập công cộng hay môi trường doanh nghiệp mà có những phương pháp lắp đặt khác nhau.

Mặc dù chỉ là trên lý thuyết và triển khai trên mô hình NS2 về mạng cục bộ không dây, chưa đi vào chi tiết nhưng nó sẽ góp phần vào việc lựa chọn các giải pháp cho việc xây dựng và mở rộng hệ thống mạng không dây.

Phân mô phỏng QoS đã cho thấy sự cần thiết của các giải pháp QoS để đảm bảo chất lượng dịch vụ mạng không dây. Giải pháp QoS hiệu quả phải được kết hợp nhiều cơ chế QoS phù hợp để đảm bảo chất lượng dịch vụ.

DANH MỤC TÀI LIỆU THAM KHẢO

1. “IEEE Standard for Information Technology--Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks-- Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” in *IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016)*, pp.1-4379, 26 Feb. 2021, doi: 10.1109/IEEEESTD.2021.9363693.
2. B. Appoon and V. Amudha. 2015. “Qos Analysis Of Data Communication Using NS2.” *Middle-East Journal of Scientific Research 23 (Sensing, Signal Processing and Security)*. IDOSI Publications, 2015.
3. Javier Bustos-Jiménez, Rodrigo Alonso, Camila Faúndez, Hugo Méric, “Boxing experience: Measuring QoS and QoE of multimedia streaming using NS3, LXC and VLC,” *39th Annual IEEE Conference on Local Computer Networks Workshops*, Edmonton, AB, Canada, 2018, doi: 10.1109/LCNW.2014.6927717.
4. Ngo Hai Anh, Pham Thanh Giang, “Controlling Contention Window to Ensure QoS for Multimedia Data in Wireless Network,”. *Journal of Research and Development on Information and Communication Technology (ICT Research)*, Vol 2020 No1, doi : <https://doi.org/10.32913/mic-ict-research.v2020.n1.896>
5. “ITU-T: Recommendation E.800 (2008) Definitions of terms related to quality of service, ” <https://www.itu.int/rec/T-REC-E.800-200809-I>, truy cập lần cuối vào 08/08/2022.
6. “ITU-T: Recommendation P.10/G.100 Amendment 3 (2012) New definitions for inclusion in recommendation ITU-T P.10/G.100,” <https://www.itu.int/rec/T-REC-P.10>, truy cập lần cuối ngày 08/08/2022
7. Silvia Giordano, *Handbook of wireless networks and mobile computing*, John Wiley & Sons, Inc., 2002, chapter Mobile Ad Hoc Networks, 325–346.

New York, NY, USA

8. “IEEE 802.11e Amendment”. https://standards.ieee.org/standard/802_11e-2005.html, truy cập lần cuối 08/08/2022
9. A. R. Khan, S. M. Bilal and M. Othman, “A performance comparison of open source network simulators for wireless networks,” *2012 IEEE International Conference on Control System, Computing and Engineering*, 2012, pp. 34-38, doi: 10.1109/ICCSCE.2012.6487111.
10. X. Zhou and H. Tian, “Comparison on Network Simulation Techniques,” *2016 17th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, 2016, pp. 313-316, doi: 10.1109/PDCAT.2016.073.
11. “The Network Simulator - ns-2,” <https://www.isi.edu/nsnam/ns/>, truy cập lần cuối 08/08/2022
12. “Nam: Network Animator,” <https://www.isi.edu/nsnam/nam/>, truy cập lần cuối 09/08/2022