

MỤC LỤC

MỞ ĐẦU.....	1
CHƯƠNG 1. TỔNG QUAN VỀ MẠNG AD HOC.....	3
1.1. Giới thiệu về mạng ad hoc.....	3
1.1.1. Khái niệm mạng ad hoc.....	3
1.1.2. Đặc điểm của mạng ad hoc.....	4
1.1.3. Ứng dụng của mạng ad hoc.....	5
1.2. Chuẩn IEE 802.11 và giao thức tầng MAC của mạng ad hoc.....	6
1.2.1. Vấn đề trạm ẩn trong mạng không dây IEEE 802.11b.....	6
1.2.2 DCF và CSMA/CA tại tầng MAC.....	7
1.2.3. Mạng ad-hoc với IEEE 802.11b.....	11
1.3. Một số chiến lược định tuyến trong mạng ad hoc.....	15
1.3.1. Định tuyến tìm đường trước và tìm đường theo yêu cầu.....	15
1.3.2. Định tuyến cập nhật định kỳ và cập nhật theo sự kiện.....	16
1.3.3. Định tuyến phẳng và định tuyến phân cấp.....	16
1.3.4. Định tuyến với kỹ thuật tính toán tập trung và tính toán phân tán....	17
1.3.5. Định tuyến nguồn và định tuyến từng chặng.....	17
1.3.6. Định tuyến đơn đường và định tuyến đa đường.....	18
1.4. Đánh giá hiệu năng mạng ad hoc.....	18
1.5. Tổng kết Chương 1.....	20
CHƯƠNG 2. MỘT SỐ GIAO THỨC ĐỊNH TUYẾN PHỔ BIẾN TRONG MẠNG AD HOC.....	22
2.1. Yêu cầu đối với giao thức và thuật toán định tuyến trong mạng ad hoc.....	22
2.2. Giao thức định tuyến DSDV.....	23
2.2.1. Tổng quan về giao thức định tuyến DSDV.....	23
2.2.2. Bảng định tuyến và thông tin quảng bá đường.....	24
2.2.3. Tần suất quảng bá đường.....	26
2.2.4. Hoạt động của giao thức DSDV ở Tầng 2.....	29

2.3. Giao thức định tuyến AODV.....	30
2.3.1. Giới thiệu về giao thức AODV.....	30
2.3.2. Cơ chế hoạt động của giao thức AODV.....	32
2.4. Giao thức định tuyến DSR.....	38
2.4.1. Tổng quan về giao thức DSR.....	38
2.4.2. Thủ tục tìm đường cơ bản của DSR.....	39
2.4.3. Thủ tục duy trì đường cơ bản của DSR.....	42
2.5. Tổng kết Chương 2.....	43
CHƯƠNG 3. MÔ PHỎNG VÀ ĐÁNH GIÁ HIỆU NĂNG ĐỊNH TUYẾN.....	45
3.1. Tổng quan về mô phỏng mạng.....	45
3.1.1. Khái niệm mô phỏng mạng.....	45
3.1.2. Các thành phần của mô phỏng.....	45
3.2. Giới thiệu về phần mềm mô phỏng NS2.....	47
3.2.1. Kiến trúc cơ bản của NS2.....	48
3.2.2. Mô phỏng sự kiện rời rạc bằng NS2.....	50
3.2.3. Cấu hình mạng ad hoc trong NS2.....	51
3.3. Mục đích và phạm vi của việc đánh giá hiệu năng các giao thức.....	53
3.4. Các thông số thiết lập mô phỏng.....	54
3.5. Mô phỏng và đánh giá ảnh hưởng của tốc độ di chuyển.....	56
3.5.1. Ảnh hưởng của tốc độ di chuyển.....	56
3.5.2. Ảnh hưởng của thời gian tạm dừng.....	61
3.6. Mô phỏng và đánh giá ảnh hưởng của tải dữ liệu.....	66
3.6.1. Ảnh hưởng của số lượng kết nối.....	66
3.6.2. Tác động của số lượng nút.....	72
3.7. Tổng kết Chương 3.....	77
KẾT LUẬN.....	80
TÀI LIỆU THAM KHẢO.....	82

MỞ ĐẦU

Trong mô hình mạng 4GN, xoay quanh hệ thống mạng lõi Internet, cùng với mạng viễn thông tế bào, mạng viễn thông vệ tinh, mạng không dây diện rộng, mạng không dây cục bộ, đã có sự xuất hiện của mạng ad hoc. Sự xuất hiện này cho thấy tầm quan trọng của mạng ad hoc trong công nghệ mạng truyền thông hiện đại và khả năng ứng dụng rộng rãi của nó vào nhiều lĩnh vực khác nhau từ hỗ trợ tác chiến trong quân đội; dự báo và cảnh báo thiên tai, thảm họa; ứng dụng thương mại, gia đình, văn phòng, giáo dục và giải trí; theo dõi và điều hành giao thông tới các dịch vụ theo dõi và chăm sóc sức khỏe cho con người.

Trong nhiều năm qua, các nghiên cứu để giải quyết các vấn đề cơ bản của mạng ad hoc trải từ tầng Ứng dụng tới tầng Vật lý trong mô hình OSI đã và đang được rất nhiều nhà khoa học quan tâm nhằm đưa mạng ad hoc có khả năng ứng dụng ngày càng rộng rãi hơn phục vụ cuộc sống của con người. Do tính chất di động cũng như khả năng thêm mới rồi bỏ kết nối mạng tự do của các nút mạng di động nên topo mạng ad hoc thường xuyên thay đổi. Thêm vào đó, không có các bộ định tuyến cố định trong mạng ad hoc. Vì vậy, giao thức định tuyến trong mạng ad hoc cần được thiết kế để thích ứng với những tính chất đặc biệt này.

Trong thời gian qua, đã có nhiều giao thức định tuyến dành cho mạng ad hoc được nghiên cứu và công bố. Việc đánh giá tính hiệu quả và mức độ phù hợp của các giao thức định tuyến này đối với các mô hình mạng ad hoc thực tế là rất khó triển khai đối với các nghiên cứu về mạng ad hoc vì lý do kinh tế và các công nghệ, sản phẩm trên thị trường đối với các thiết bị của mạng ad hoc chưa thực sự hoàn thiện. Vì vậy, phần lớn các nghiên cứu về đánh giá hiệu năng định

tuyến dành cho mạng ad hoc hiện nay đang tập trung vào việc sử dụng các hệ thống mô phỏng.

Mục đích của đề tài là nghiên cứu về lý thuyết hoạt động của các giao thức định tuyến và thực nghiệm bằng hệ thống mô phỏng sự tác động của tốc độ di chuyển và tải dữ liệu đối với hiệu năng của một số giao thức định tuyến phổ biến trong mạng ad hoc. Qua đó, rút ra các kết luận và khuyến nghị về việc sử dụng giao thức định tuyến phù hợp đối với các mô hình mạng ad hoc trong các điều kiện cụ thể.

Luận văn được bố cục như sau: Phần mở đầu giới thiệu ý nghĩa, mục tiêu nghiên cứu và bố cục của luận văn. Trong Chương 1, các vấn đề tổng quan về mạng ad hoc, các chuẩn và giao thức tầng MAC dành cho mạng ad hoc, các chiến lược định tuyến dành cho mạng ad hoc và vấn đề đánh giá hiệu năng mạng ad hoc sẽ được trình bày. Nội dung của Chương 2 trình bày chi tiết về cơ chế hoạt động của một số giao thức và thuật toán định tuyến phổ biến trong mạng ad hoc. Chương 3 trình bày vấn đề mô phỏng mạng máy tính và bộ công cụ mô phỏng NS-2, kết quả mô phỏng và những phân tích đánh giá hiệu năng của một số giao thức định tuyến dưới góc nhìn về mức độ ảnh hưởng của tốc độ di chuyển và tải dữ liệu. Cuối cùng là phần kết luận và hướng phát triển của luận văn.

CHƯƠNG 1. TỔNG QUAN VỀ MẠNG AD HOC

1.1. Giới thiệu về mạng ad hoc

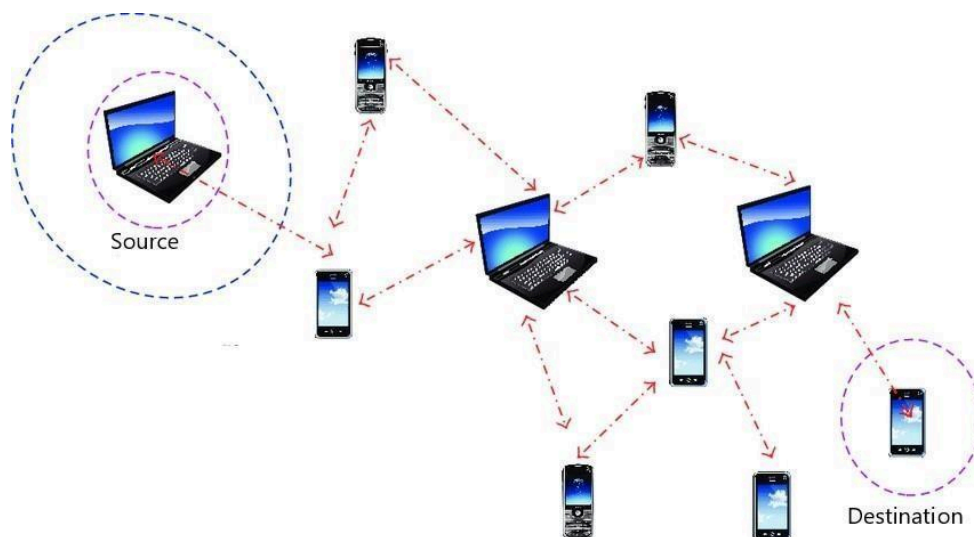
1.1.1. Khái niệm mạng ad hoc

Mạng ad hoc [4] là một tập các nút di động có khả năng tự kết nối và tự tổ chức để hình thành một mạng mà không cần các thiết bị hạ tầng mạng cơ sở đóng vai trò trung gian để thu/phát tín hiệu và chuyển tiếp dữ liệu. Do mỗi nút di động sử dụng công nghệ truyền thông không dây chỉ có thể truyền thông một phạm vi giới hạn nên cần có sự hỗ trợ của các nút lân cận để chuyển tiếp các gói dữ liệu.

Các nút trong mạng ad hoc thông thường sẽ kết nối với nhau trong một khoảng thời gian để trao đổi thông tin. Trong khi trao đổi thông tin, các nút này vẫn có thể di chuyển, do đó, mạng này phải đáp ứng được yêu cầu truyền dữ liệu trong khi hình trạng mạng có thể thay đổi liên tục. Các nút mạng phải có cơ chế tự tổ chức thành một mạng để thiết lập các đường truyền dữ liệu mà không cần sự hỗ trợ từ bên ngoài. Trong mô hình này, mỗi nút mạng có thể đóng vai trò là một nút đầu cuối để chạy các chương trình ứng dụng của người sử dụng hoặc là một bộ định tuyến để chuyển tiếp các gói tin cho các nút mạng khác.

Thuật ngữ “Ad hoc” áp dụng cho mạng không dây mô tả một mạng không có cơ sở hạ tầng cố định, trong đó hình trạng mạng được tạo thành bởi chính các nút mạng. Chế độ “Ad hoc” của chuẩn IEEE 802.11 hoạt động theo mô hình này, mặc dù nó chỉ hỗ trợ để thiết lập một mạng đơn chặng. Các mạng di động không dây kiểu không cấu trúc đã mở rộng khái niệm “Ad hoc” đa chặng theo nghĩa: một nút mạng có thể định tuyến và chuyển tiếp một gói tin nó nhận được từ một nút mạng khác. Nói cách khác, con đường chuyển tiếp gói tin từ nút nguồn tới

nút đích có thể chứa các nút trung gian khác. Các nút trung gian sẽ đọc thông tin trong phần header của các gói tin dữ liệu và chuyển tiếp chúng tới chặng kế tiếp trên một con đường đã được hình thành.



Hình 1.1. Một ví dụ của mạng ad hoc

Hình 1.1 minh họa một mạng ad hoc. Trong ví dụ này, các gói tin từ nút nguồn là một máy tính cần chuyển tới một nút đích là một điện thoại thông minh không nằm trong phạm vi truyền của nút nguồn. Vì vậy, cần có sự trợ giúp của các nút trung gian để chuyển tiếp gói tin từ nút nguồn tới nút đích. Để thực hiện được công việc này, các nút mạng phải sử dụng giao thức định tuyến phù hợp cho mạng ad hoc.

1.1.2. Đặc điểm của mạng ad hoc

Do ad hoc là một mạng không dây hoạt động không cần sự hỗ trợ của hạ tầng mạng cơ sở trên cơ sở truyền thông đa chặng giữa các thiết bị di động vừa đóng vai trò là thiết bị đầu cuối, vừa đóng vai trò là bộ định tuyến nên mạng ad hoc còn có một số đặc điểm nổi bật sau [1]:

- Cấu trúc động.
- Chất lượng liên kết hạn chế.
- Các nút mạng có tài nguyên hạn chế.
- Độ bảo mật thấp ở mức độ vật lý.

Đây là những đặc điểm ảnh hưởng rất nhiều đến hiệu năng của mạng ad hoc. Để có thể triển khai được mạng ad hoc trong thực tế, các thiết kế mạng ad hoc phải giải quyết được những thách thức sinh ra do những đặc điểm đã nêu trên của mạng ad hoc.

1.1.3. Ứng dụng của mạng ad hoc

Ngày nay, mạng ad hoc có nhiều những ứng dụng trong đời sống, kinh tế, xã hội của con người. Mô hình mạng này phù hợp đối với những tình huống cần triển khai hệ thống mạng một cách nhanh chóng, linh động và thường xuyên có sự biến đổi trong cấu trúc mạng. Mạng ad hoc có tiềm năng ứng dụng rất lớn trong những lĩnh vực sau:

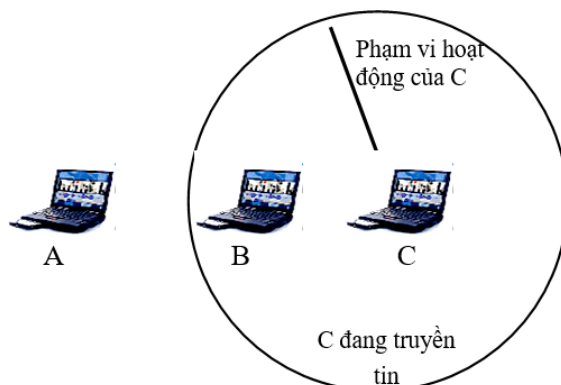
- Ứng dụng trong thương mại
- Ứng dụng trong quân sự
- Ứng dụng trong phòng chống thiên tai, thảm họa
- Ứng dụng trong giáo dục
- Ứng dụng trong quản lý giao thông
- Ứng dụng trong mạng cảm biến

1.2. Chuẩn IEEE 802.11 và giao thức tầng MAC của mạng ad hoc

1.2.1. Vấn đề trạm ẩn trong mạng không dây IEEE 802.11b

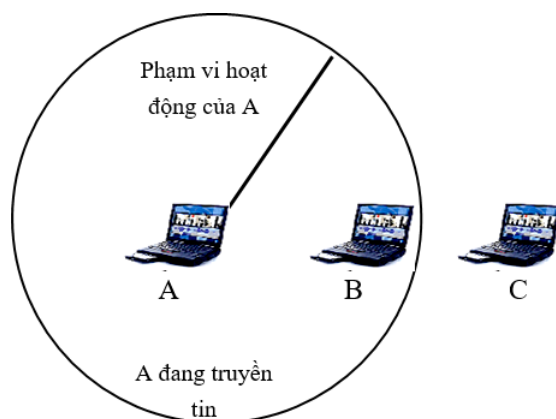
Trong chế độ ad-hoc của IEEE 802.11b, hai nút chỉ có thể truyền thông được với nhau khi chúng ở trong phạm vi hoạt động của nhau. Các nút bên ngoài phạm vi hoạt động được coi là “không trông thấy”. Vấn đề trạm ẩn xảy ra khi hai nút ở bên ngoài phạm vi hoạt động của nhau truyền dữ liệu tại cùng một thời điểm tới một nút thứ ba (ở trong phạm vi hoạt động của hai nút kia). Do hai nút này ở ngoài phạm vi hoạt động của nhau nên không thể “cảm nhận” được tình huống này. Xung đột sẽ xảy ra tại nút thứ ba.

Hình 1.2 minh họa vấn đề trạm ẩn. Tác động của vấn đề trạm ẩn là cả nút A hoặc nút B không thể dò tìm được xung đột do chúng ở ngoài phạm vi hoạt động của nhau. Việc thiếu ACK cho mỗi gói tin sẽ làm cho hai nút giả thiết rằng gói tin bị mất vì một vài lý do nào đó. Kết quả là cả hai sẽ truyền lại gói tin của chúng làm cho có thể xung đột lại xảy ra một lần nữa. Điều này sẽ tiếp diễn cho đến khi hết thời gian của nút A và B và báo lỗi.



Hình 1.2. Ví dụ 1 về tình huống trạm ẩn

Trong Hình 1.3, nút B muốn gửi dữ liệu tới C, nó sẽ cảm nhận kênh truyền. Khi cảm nhận kênh truyền là không rỗi, nó kết luận sai rằng không thể truyền dữ liệu tới C thậm chí còn nghĩ rằng A đang truyền dữ liệu tới D (D không được chỉ ra trong hình vẽ). Hơn nữa, hầu hết các kênh truyền sóng vô tuyến đều là kênh truyền bán song công – có nghĩa là chúng không thể truyền tin và lắng nghe phản hồi tại cùng một thời điểm đối với cùng một tần số đơn. Kết quả là 802.11b không sử dụng chiến lược CSMA/CD như của Ethernet.



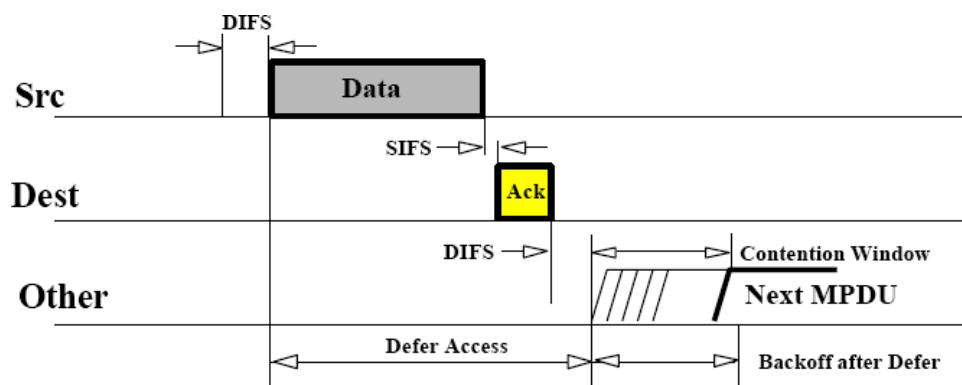
Hình 1.3. Ví dụ 2 về tính hướng trạm ẩn

Để giải quyết vấn đề này, IEEE 802.11b hỗ trợ hai trạng thái hoạt động: DCF – không sử dụng bất cứ điều khiển tập trung nào (ở khía cạnh này tương tự như Ethernet) và PCF – sử dụng một trạm cơ sở để điều khiển tất cả các hoạt động trong tế bào (cell) của nó. Tất cả các cài đặt đều phải hỗ trợ DCF nhưng PCF là tùy chọn.

1.2.2 DCF và CSMA/CA tại tầng MAC

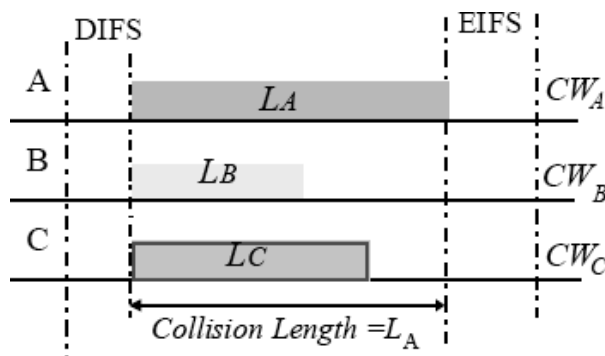
Để khắc phục những vấn đề không tương thích trong dò tìm xung đột giữa mạng có dây và mạng không dây, IEEE 802.11b sử dụng kỹ thuật tránh xung đột cùng với một chiến lược biên nhận tích cực như sau (Hình 1.4): Trạm muốn

truyền tin cảm nhận kênh truyền. Nếu kênh truyền được cảm nhận là bận, nó sẽ chờ. Nếu kênh truyền là rỗi trong một khoảng thời gian xác định (được gọi là DIFS – Distributed Inter Frame Space), trạm được phép truyền tin. Bên nhận khi nhận được gói tin sẽ thực hiện thuật toán CRC để dò tìm lỗi, sau đó đợi trong một khoảng thời gian được gọi là SIFS ($SIFS < DIFS$) và gửi gói ACK. ACK sẽ không được gửi đi nếu gói tin do trạm nguồn gửi bị lỗi hoặc bị mất. Nếu bên gửi không nhận được ACK, nó sẽ giả thiết có xung đột và lập kế hoạch truyền lại.



Hình 1.4. Cơ chế CSMA/CA

Khi phát hiện thấy gói tin gửi đi bị lỗi, kênh truyền phải rỗi trong một khoảng thời gian nhỏ nhất là EIFS trước khi trạm được kích hoạt trở lại thuật toán back-off để lập kế hoạch truyền tin tiếp theo (Hình 1.5).



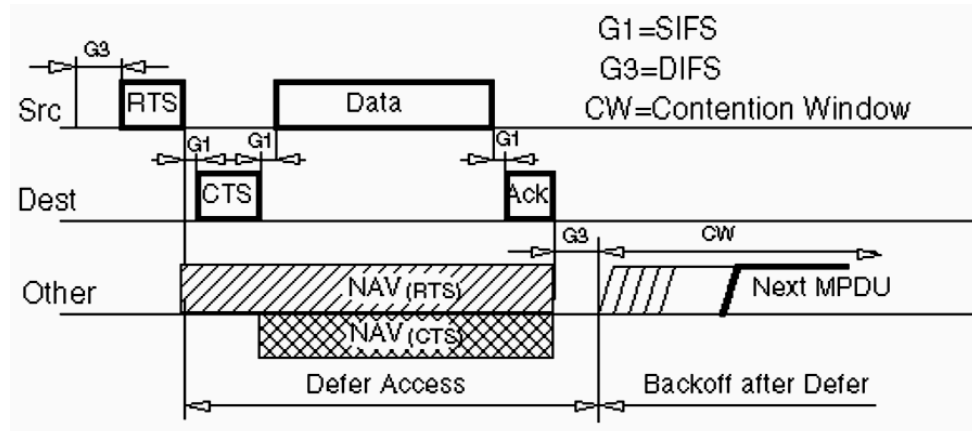
Hình 1.5. Xung đột

Để làm giảm xác suất xung đột, IEEE 802.11b sử dụng kỹ thuật back-off: Khi trạm S muốn truyền tin đi cảm nhận thấy kênh truyền đang bận, nó sẽ chờ cho đến khi kết thúc quá trình truyền tin hiện tại. Tại thời điểm cuối của quá trình truyền tin hiện tại, trạm S khởi tạo một bộ đếm (gọi là back-off timer) bằng cách chọn một khoảng thời gian ngẫu nhiên (back-off interval) để lập lịch cho việc truyền tin của nó. Bộ đếm sẽ giảm trong thời gian kênh truyền được cảm nhận là rỗi, dừng lại khi có phát hiện thấy kênh truyền đang truyền tin và được kích hoạt lại khi kênh truyền được cảm nhận là rỗi trong một khoảng thời gian lớn hơn DIFS. Khi bộ đếm bằng 0, trạm được phép truyền tin. Ở đây DCF sử dụng kỹ thuật back-off hàm mũ hai theo khe thời gian. Thời gian theo sau một DIFS hoặc EIFS được phân chia thành khe; một trạm chỉ được phép truyền tin tại thời điểm bắt đầu mỗi khe thời gian (Slot Time). Thời gian back-off thường được chọn trong khoảng $(0, CW)$, được gọi là Cửa sổ Back-off (Back-off Window/Contention Window). Tại thời điểm thử truyền tin lần đầu tiên, $CW = CW_{min}$, giá trị CW được gấp đôi sau mỗi lần thử truyền tin lại ($CW_i = 2^{k+i-1} - 1$, trong đó i là số lần thử truyền tin – tính cả lần đang xét, k là hằng số xác định giá trị CW_{min}), tới giá trị tối đa là CW_{max} . Giá trị cụ thể của CW_{min} và CW_{max} phụ thuộc vào từng kiểu tầng vật lý, ví dụ nếu tầng vật lý là FHSS thì $CW_{min} = 16$ và $CW_{max} = 1024$.

Việc cảm nhận kênh truyền như trên là cảm nhận vật lý kênh truyền. Chức năng cảm nhận do tầng vật lý cung cấp. Tuy nhiên, trong nhiều trường hợp cảm nhận vật lý kênh truyền không cung cấp đủ các thông tin cần thiết, ví dụ như vấn đề trạm ẩn. Do đó, trong IEEE 802.11b còn hỗ trợ một chiến lược cảm nhận ảo được cung cấp bởi NAV (Network Allocation Vector).

Phần lớn các gói tin (frame) của IEEE 802.11b có một trường “duration”, được dùng để dành kênh truyền trong một khoảng thời gian cố định. NAV là một bộ định thời (timer) cho biết kênh truyền được để dành trong thời gian bao lâu. Các trạm thiết lập giá trị NAV bằng thời gian chúng muốn sử dụng kênh truyền – là khoảng thời gian cần để truyền đi tất cả các frame cần thiết để hoàn thành hành động hiện tại. Các trạm khác sẽ thực hiện đếm ngược từ giá trị NAV tới 0. Khi NAV khác 0, chức năng cảm nhận sóng mang ảo cho biết kênh truyền là bận, khi NAV được giảm tới 0, chức năng cảm nhận sóng mang ảo cho biết kênh truyền là rỗi.

Với NAV, cơ chế cảm nhận sóng mang ảo RTS/CTS được thực hiện như sau (Hình 1.6): Sau khi giành được quyền truy cập kênh truyền, trước khi bắt đầu truyền tin, trạm phải gửi đi một gói tin điều khiển RTS (Request To Send) tới trạm nhận để thông báo về việc truyền tin sắp tới. Phía nhận sẽ trả lời lại gói tin RTS bằng gói tin CTS để cho biết đã sẵn sàng nhận tin. Cả RTS và CTS đều chứa độ dài dự kiến của việc truyền tin (thời gian truyền gói tin và ACK). Tất cả các trạm khi nhận được RTS hoặc CTS sẽ thiết lập chỉ số cảm nhận sóng mang ảo của nó hay còn gọi là NAV bằng khoảng thời gian dự kiến truyền tin. Thông tin này sẽ được sử dụng cùng với cảm nhận vật lý kênh truyền khi cảm nhận kênh truyền.



Hình 1.6. CSMA/CA với cảm nhận sóng mang ảo.

Cơ chế này giải quyết được vấn đề trạm ẩn vì tất cả các trạm ở trong phạm vi hoạt động của trạm gửi hoặc trạm nhận đều biết được kênh truyền sẽ được sử dụng cho việc truyền tin hiện tại trong bao lâu, đảm bảo được rằng không một nút nào có thể làm dừng quá trình truyền tin cho đến khi nút nhận đã gửi ACK cho nút gửi. Tuy nhiên, do sử dụng RTS và CTS nên tổng phí truyền tin tăng, xuất hiện dưới dạng độ trễ trước khi dữ liệu thực được truyền đi. Vì vậy truyền đi một gói dữ liệu lớn có lợi hơn là gửi nhiều gói dữ liệu nhỏ. Chuẩn IEEE 802.11b còn định nghĩa một tham số gọi là RTSThreshold (ngưỡng RTS) cho phép các gói tin nhỏ được truyền đi không cần quá trình trao đổi RTS/CTS.

1.2.3. Mạng ad-hoc với IEEE 802.11b

Trong IEEE 802.11b, chế độ IBSS cho phép triển khai một mạng ad-hoc.

Để xác định các mạng LAN không dây khác nhau trong cùng một vùng,

mỗi một mạng được gán với một số định danh. Trong chế độ ad-hoc (IBSS), số định danh của IBSS là IBSSID. Khi một trạm nào đó khởi tạo một IBSS, IBSSID là một số 46bits được sinh theo một thuật toán tạo số ngẫu nhiên sao cho xác suất để các trạm khác cũng tạo ra số đó là nhỏ nhất.

Do tính linh động của thuật toán CSMA/CA, sự đồng bộ hóa của các trạm theo một đồng hồ chung là không đủ để nhận hoặc truyền dữ liệu đúng. IEEE 802.11b sử dụng hai chức năng đồng bộ hóa các trạm trong một IBSS là: (1) thực hiện đồng bộ hóa và (2) bảo trì đồng bộ hóa.

Thực hiện đồng bộ hóa: Chức năng này là cần thiết để một nút tham gia vào một IBSS đã có. Để phát hiện ra một IBSS đã có, thực hiện một thủ tục quét kênh truyền không dây. Trong quá trình quét, trạm nhận được bật lên với các tần số vô tuyến khác nhau, tìm kiếm frame điều khiển. Nếu thủ tục quét không phát hiện thấy một IBSS nào, trạm có thể tự khởi tạo một IBSS.

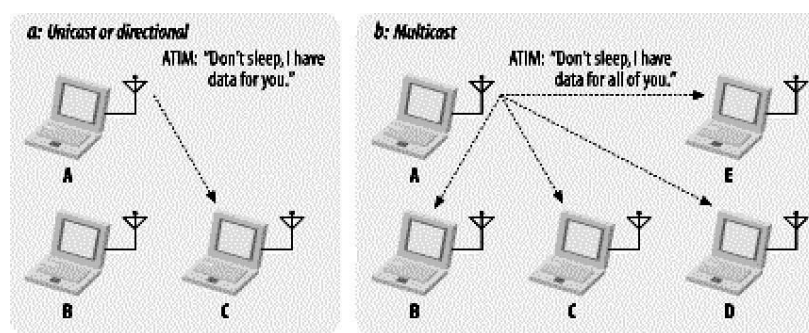
Bảo trì đồng bộ hóa: Do không có điểm truy cập trong IBSS nên chức năng đồng bộ hóa được triển khai thông qua một thuật toán phân tán – được thực hiện bởi tất cả các thành viên của IBSS. Thuật toán này được dựa trên việc truyền lại các frame báo hiệu (Beacon)¹ theo một tỷ lệ rất nhỏ đã được biết trước. Khoảng thời gian các lần truyền frame báo hiệu được xác định bởi trạm đã khởi tạo IBSS.

Trong môi trường di động, chức năng quản lý nguồn nuôi là đặc biệt quan trọng do các nút mạng cần thực hiện chính sách tiết kiệm năng lượng. Trong IBSS, chính sách tiết kiệm năng lượng được triển khai hoàn toàn phân tán. Một trạm có thể ở trong hai trạng thái năng lượng khác nhau:

- Hoạt động (awake): trạm có đủ năng lượng
- Ngủ (doze): trạm không thể nhận hoặc truyền dữ liệu

¹ Frame báo hiệu: chứa các thông tin để đồng bộ hóa và tập các tham số của mạng hiện tại (IBSSID, beacon period, công nghệ của tầng vật lý, các tham số liên quan đến quản lý mạng, các tham số liên quan đến chính sách tiết kiệm năng lượng,...)

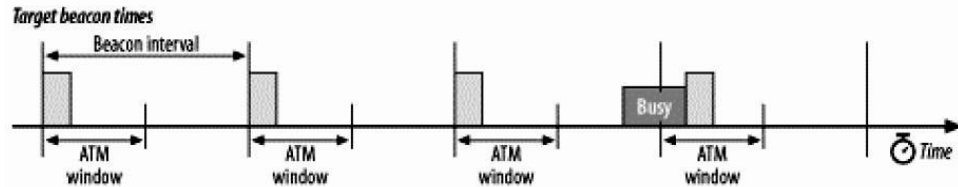
Các trạm sử dụng thông điệp chỉ thị lưu lượng ATIMs để thông báo về việc truyền dữ liệu sắp tới và các trạm nhận dữ liệu không được chuyển sang trạng thái “ngủ”. Trong Hình 2-11, trong bộ đệm của trạm A có một frame cần gửi tới C. Do đó A gửi một frame ATIM theo kiểu unicast cho trạm C trong thời gian ứng với cửa sổ truyền tin ATIM để thông báo cho C rằng A sắp truyền tin tới C và C không nên ở trạng thái tiết kiệm năng lượng. Tuy nhiên trạm B là tự do khi muốn chuyển trạng thái năng lượng. Hình minh họa frame ATIM được phát multicast tới một nhóm trạm, thông báo cho các trạm này biết không chuyển sang chế độ tiết kiệm năng lượng.



Hình 1.7. Sử dụng frame ATIM.

Rõ ràng rằng frame ATIM phải được truyền đi trong khoảng thời gian tất cả các trạm là hoạt động. Do đó định nghĩa cửa sổ ATIM (ATIM window) là một khoảng thời gian xác định tính từ thời điểm frame báo hiệu bắt đầu được truyền đi. Trong khoảng thời gian ứng với cửa sổ ATIM, tất cả các trạm phải ở trạng thái hoạt động (các trạm đang trong trạng thái tiết kiệm năng lượng đều được “đánh thức”), chỉ được truyền đi frame báo hiệu và ATIM. Giá trị của cửa sổ ATIM là một tham số của IBSS và được xác định khi IBSS được tạo ra. Nếu giá trị cửa sổ ATIM bằng 0, có nghĩa là IBSS không áp dụng quản lý năng lượng. Hình 1.8 minh họa cửa sổ ATIM và mối quan hệ của nó với khoảng thời gian

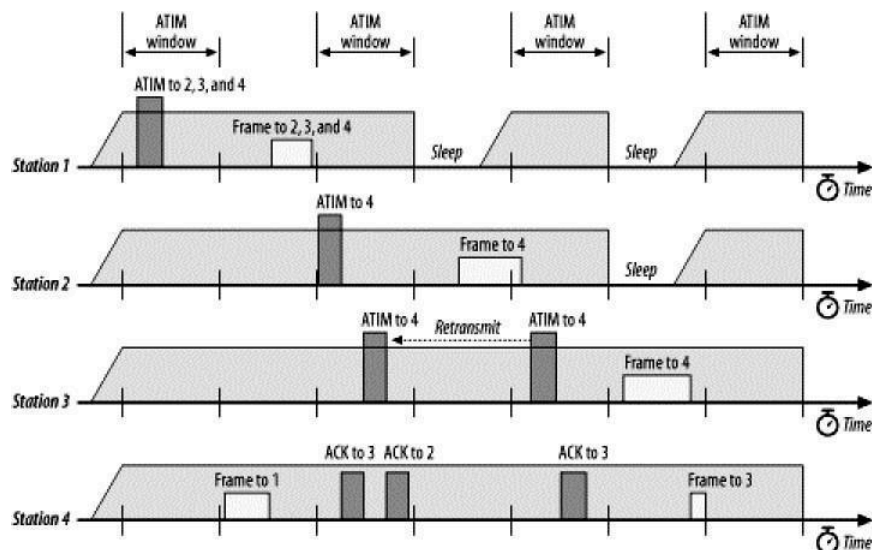
giữa các frame báo hiệu. Trong hình vẽ, frame báo hiệu thứ tư bị truyền trễ do kênh truyền bận. Giá trị của số ATIM là hằng số, do đó khoảng thời gian sử dụng thực sự của cửa sổ ATIM bị co lại.



Hình 1.8. Cửa sổ ATIM

Hình 1.9 minh họa tác động của frame ATIM đối với chế độ tiết kiệm năng lượng trong một IBSS: Trong khoảng thời gian báo hiệu thứ nhất, trạm 1 truyền đi frame ATIM theo kiểu multicast tới trạm 2, 3 và 4. Khi truyền đi frame ATIM theo kiểu multicast, trạm 1 không cần nhận được biên nhận, tuy nhiên việc truyền này có ý nghĩa rằng tất cả các trạm phải ở trong trạng thái hoạt động trong suốt khoảng thời gian của cửa sổ báo hiệu thứ nhất để nhận các frame do trạm 1 gửi đi theo kiểu multicast. Khi hết thời gian của cửa sổ ATIM, trạm 1 có thể truyền đi frame tới các trạm kia. Trạm 4 có thể lợi dụng khoảng thời gian còn lại để gửi đi một frame tới trạm 1. Trong khoảng thời gian báo hiệu thứ hai, trạm 2 và 3 đều có frame trong bộ đệm cần gửi tới trạm 4, do đó chúng gửi frame ATIM tới trạm 4. Trạm 4 khi nhận được các frame ATIM này, gửi frame biên nhận cho trạm gửi tương ứng. Khi hết thời gian của cửa sổ ATIM, trạm 1 không có nhu cầu truyền tin cũng như không nhận được frame ATIM nào, do đó nó có thể bước vào trạng thái “ngủ” cho tới khi bắt đầu khoảng thời gian báo hiệu thứ ba. Trạm 2 gửi một frame với kích thước lớn tới trạm 4, do đó lấy đi cơ hội truyền tin của trạm 3 tới trạm 4 trong khoảng thời gian báo hiệu này. Khi bắt đầu khoảng thời gian báo hiệu thứ ba, trạm 3 vẫn có frame trong hàng đợi cần gửi

đến trạm 4. Do đó, nó truyền lại một frame ATIM tới trạm 4. Trạm 4 khi nhận được frame ATIM này sẽ gửi lại một frame biên nhận. Trạm 2 không tham gia vào việc trao đổi frame ATIM và không có nhu cầu truyền tin nữa, do đó bước vào trạng thái “ngủ”. Trạm 3 gửi frame tới trạm 4. Sau đó, trạm 4 có thể lợi dụng khoảng thời gian còn lại để gửi đi một frame tới trạm 4 (trạm 3 biết trạm 4 ở trong trạng thái hoạt động do quá trình trao đổi frame trước đó)



Hình 1.9. Tác động của frame ATIM đối với chế độ tiết kiệm năng lượng

1.3. Một số chiến lược định tuyến trong mạng ad hoc

1.3.1. Định tuyến tìm đường trước và tìm đường theo yêu cầu

Kiểu định tuyến tìm đường trước còn được gọi là “định tuyến kích hoạt trước” hay “định tuyến điều khiển dạng bảng”. Đối với kiểu định tuyến này, các con đường tới mọi đích được tìm ra trước khi có nhu cầu truyền dữ liệu tại mọi nút mạng.

Đối với các giao thức định tuyến tìm đường theo yêu cầu, chỉ khi có nhu cầu sử dụng đường truyền dữ liệu, các nút liên quan mới khởi tạo tiến trình tìm

đường và trao đổi thông tin định tuyến. Phương pháp này có ưu điểm là tiết kiệm băng thông mạng dành cho tải định tuyến nhưng cũng có nhược điểm là quá trình tìm kiếm tuyến đường có thể gây ra một độ trễ truyền tin đáng kể.

1.3.2. Định tuyến cập nhật định kỳ và cập nhật theo sự kiện

Chiến lược định tuyến theo chu kỳ sẽ duy trì độ ổn định của mạng và quan trọng nhất là cho phép các nút mạng học được thông tin về hình trạng và trạng thái của toàn mạng. Tuy nhiên, nếu sử dụng chu kỳ dài để cập nhật thông tin định tuyến, các nút mạng có thể chứa các thông tin định tuyến đã cũ và không chính xác. Ngược lại, nếu chu kỳ cập nhật thông tin định tuyến là quá ngắn, sẽ có quá nhiều gói tin định tuyến được sinh ra và quảng bá trong mạng gây ra sự lãng phí về tài nguyên mạng.

Đối với chiến lược định tuyến theo sự kiện, khi có một sự kiện diễn ra trong mạng, những nút mạng chịu tác động trực tiếp của các sự kiện này mới quảng bá các gói tin cập nhật thông tin định tuyến. Vì vậy, thông tin về những thay đổi của trạng thái mạng sẽ nhanh chóng được cập nhật tới các nút mạng. Tuy nhiên, khi topo mạng thay đổi với tốc độ nhanh, sẽ có rất nhiều các gói tin quảng bá cập nhật định tuyến được sinh ra làm lãng phí băng thông mạng và biến động đối với các con đường truyền dữ liệu.

1.3.3. Định tuyến phẳng và định tuyến phân cấp

Trong định tuyến phẳng, mọi nút trong mạng đều có cùng cấp độ và chức năng định tuyến. Chiến lược định tuyến này tương đối đơn giản và hiệu quả đối với các mạng nhỏ. Các giao thức AODV, DSDV, DSR là những giao thức điển hình sử dụng chiến lược định tuyến phẳng.

Trong chiến lược định tuyến phân cấp, các nút mạng được tổ chức một cách link động thành các vùng. Mỗi vùng lại có thể chia tiếp thành các vùng con theo kiểu cây phân cấp. Cấu trúc phân cấp này nhằm duy trì tính ổn định tương đối của hình trạng mạng.

1.3.4. Định tuyến với kỹ thuật tính toán tập trung và tính toán phân tán

Trong chiến lược định tuyến với kỹ thuật tính toán tập trung, mọi nút trong mạng sẽ duy trì thông tin đầy đủ về toàn bộ hình trạng mạng để có thể tự thực hiện các thuật toán tìm đường khi cần thiết.

Trong chiến lược định tuyến với kỹ thuật tính toán phân tán, mọi nút mạng chỉ duy trì thông tin cục bộ về hình trạng mạng. Khi có nhu cầu tìm đường, nhiều nút mạng sẽ cùng tham gia vào tiến trình tìm đường.

1.3.5. Định tuyến nguồn và định tuyến từng chặng

Có một vài giao thức định tuyến đưa thông tin về toàn bộ con đường vào trong header của các gói tin dữ liệu để các nút trung gian có thể chuyển tiếp những gói tin này theo các thông tin định tuyến mà nó đọc được trong phần header. Chiến lược định tuyến này được gọi là định tuyến nguồn.

Trong chiến lược định tuyến từng chặng, con đường tới nút đích được phân bố trong các “chặng kế tiếp” của các nút thuộc con đường này. Khi một nút nhận được một gói tin cần truyền tới một đích xác định, nó sẽ chuyển tiếp gói tin này tới chặng kế tiếp tương ứng trên con đường. Vì mỗi nút mạng không có thông tin đầy đủ về toàn bộ các liên kết trong mạng nên thuật toán định tuyến phải đảm bảo không chọn các con đường gây ra định tuyến lặp.

1.3.6. Định tuyến đơn đường và định tuyến đa đường

Đối với các giao thức định tuyến đơn đường, chỉ có tối đa một con đường tối ưu theo độ đo định tuyến của chúng được cài đặt vào bảng định tuyến sau mỗi tiến trình tìm đường.

Để tiết kiệm tài nguyên hệ thống mạng trong các tiến trình tìm đường, các giao thức định tuyến đa đường cho phép tìm và cài đặt nhiều hơn một con đường không giao nhau tới cùng một đích vào bảng định tuyến của chúng. Tại một nút, khi có yêu cầu chuyển tiếp dữ liệu tới nút đích, con đường tốt nhất sẽ được sử dụng và những con đường còn lại sẽ đóng vai trò là đường dự phòng.

1.4. Đánh giá hiệu năng mạng ad hoc

Hiệu năng mạng là những tiêu chuẩn chính được sử dụng trong thiết kế và khai thác hệ thống mạng. Việc đánh giá hiệu năng mạng nhằm mục đích so sánh giữa các thiết kế để tìm ra thiết kế tốt nhất. Có ba phương pháp phổ biến thường được sử dụng để đánh giá hiệu năng mạng [3] là đánh giá bằng kỹ thuật đo lường, đánh giá bằng mô hình phân tích và đánh giá bằng mô phỏng. Các phương pháp đánh giá hiệu năng mạng này có thể được sử dụng để đánh giá hiệu năng của các giao thức định tuyến trong mạng ad hoc.

- Đánh giá bằng kỹ thuật đo lường: phương pháp này có thể được thực hiện trong một mạng test-bed. Nó yêu cầu các thiết bị thật, các chương trình và thời gian để chạy các thí nghiệm. Phương pháp này chỉ được áp dụng hệ thống mạng thực đã tồn tại và được phép truy cập để đánh giá. Hiệu năng mạng được đánh giá theo phương pháp này tương đối chính xác với hoạt động của hệ thống thực tế. Tuy nhiên phương pháp này có nhược điểm là mạng test-bed thường khó cấu hình và khó có khả năng chia sẻ cho những người nghiên cứu, chi phí cao và

yêu cầu người nghiên cứu phải có khả năng thống kê, phân tích các kết quả thu được sau các thí nghiệm.

- Đánh giá bằng mô hình phân tích: Phương pháp này sử dụng các mô hình và khái niệm toán học để mô tả các tham số hiệu năng, các yếu tố ảnh hưởng tới hiệu năng và mối tác động giữa chúng. Ưu điểm của phương pháp này là chi phí và thời gian thử nghiệm thấp, kết quả dễ dự đoán hơn so với phương pháp đo lường và mô phỏng. Tuy nhiên, kết quả của phương pháp này có thể không chính xác với thực tế vì nó thường sử dụng nhiều giả định toán học để giải bài toán của mô hình và có thể bỏ qua các yếu tố động của mạng.

- Đánh giá bằng mô phỏng: Khi không có điều kiện để triển khai phương pháp đo lường trên hệ thống thực hoặc mô hình toán học sau khi đã rút gọn vẫn không giải được, phương pháp đánh giá bằng mô phỏng thường được sử dụng. Đây là một phương pháp được các nhà nghiên cứu về mạng và truyền thông sử dụng rất rộng rãi. Ưu điểm của phương pháp này là có thể nghiên cứu hoạt động, đánh giá hiệu năng của các hệ thống mạng đã hoặc chưa tồn tại trong thực tế. Khi bộ mô phỏng đã được xây dựng, người nghiên cứu có thể chạy các thử nghiệm với chi phí thấp với độ chính xác theo yêu cầu. Tuy nhiên, phương pháp này yêu cầu thời gian để xây dựng và kiểm nghiệm tính chính xác của bộ mô phỏng.

Trong ba phương pháp trên, phương pháp đánh giá bằng mô phỏng đã được sử dụng trong luận văn này để đánh giá sự tác động của tốc độ di chuyển và tải dữ liệu đối với hiệu năng định tuyến trong mạng ad hoc trên cơ sở sử dụng bộ mô phỏng NS2. Đây là một bộ mô phỏng được sử dụng rộng rãi trong các

nghiên cứu về mạng và truyền thông nói chung cũng như trong các nghiên cứu về các mạng ad hoc nói riêng.

1.5. Tổng kết Chương 1

Mạng ad hoc là một tập các nút di động có khả năng tự kết nối và tự tổ chức để hình thành một mạng mà không cần các thiết bị hạ tầng mạng cơ sở đóng vai trò trung gian để thu/phát tín hiệu và chuyển tiếp dữ liệu. Mỗi nút mạng vừa đóng vai trò là một thiết bị đầu cuối vừa đóng vai trò là một bộ định tuyến để tìm đường và chuyển tiếp dữ liệu cho các nút mạng khác.

Với những ưu điểm về độ linh động cao và không yêu cầu hạ tầng mạng có sẵn, mạng ad hoc có tiềm năng ứng dụng vào rất nhiều lĩnh vực phục vụ cho cuộc sống của con người như trong các mạng truyền thông trong quân sự, cảnh báo thiên tai hiểm họa, giao thông, thương mại, giải trí, giáo dục, ...

Do các nút mạng có khả năng di chuyển tự do và ngẫu nhiên nên topo mạng ad hoc thường xuyên thay đổi, các liên kết và đường truyền dữ liệu trong mạng thường xuyên bị phá vỡ và hình thành mới. Vì vậy các giao thức định tuyến trong mạng ad hoc cần phải được thiết kế phù hợp với yêu cầu này. Có nhiều chiến lược định tuyến đã được thiết kế dành cho mạng ad hoc. Mỗi chiến lược đều có những ưu và nhược điểm khác nhau và được sử dụng phù hợp trong những điều kiện, hoàn cảnh khác nhau.

Chuẩn IEEE 802.11 là chuẩn được sử dụng rộng rãi tại tầng Vật lý và tầng Liên kết dữ liệu của các mạng không dây. Tuy nhiên cơ chế truy cập môi trường truyền CSMA/CD được sử dụng trong công nghệ IEEE 802.11 Ethernet không phù hợp với mạng không dây ad hoc vì tính chất vật lý của mạng không dây và vấn đề trạm ẩn. Cơ chế truy cập môi trường truyền CSMA/CA trong DCF của

chuẩn IEEE 802.11 đã được thiết kế để giải quyết các vấn đề kỹ thuật tại tầng MAC dành cho mạng ad hoc. Kỹ thuật cảm nhận sóng mang ảo trên cơ sở hội thoại bằng các gói RTS, CTS, ACK đã được trình bày chi tiết trong chương này.

Hiệu năng mạng là những tiêu chuẩn chính được sử dụng trong thiết kế và khai thác hệ thống mạng. Việc đánh giá hiệu năng mạng nhằm mục đích so sánh giữa các thiết kế để tìm ra thiết kế tốt nhất. Có ba phương pháp phổ biến thường được sử dụng để đánh giá hiệu năng mạng là đánh giá bằng kỹ thuật đo lường, đánh giá bằng mô hình phân tích và đánh giá bằng mô phỏng. Các phương pháp đánh giá hiệu năng mạng này có thể được sử dụng để đánh giá hiệu năng của các giao thức định tuyến trong mạng ad hoc.

Trong ba phương pháp trên, phương pháp đánh giá bằng mô phỏng đã được sử dụng trong luận văn này để đánh giá sự tác động của tốc độ di chuyển và tải dữ liệu đối với hiệu năng định tuyến trong mạng ad hoc trên cơ sở sử dụng bộ mô phỏng NS2.

CHƯƠNG 2. MỘT SỐ GIAO THỨC ĐỊNH TUYẾN PHỔ BIẾN TRONG MẠNG AD HOC

2.1. Yêu cầu đối với giao thức và thuật toán định tuyến trong mạng ad hoc

Do mạng ad hoc có nhiều điểm khác biệt so với mạng truyền thống nên các giao thức và thuật toán định tuyến trong mạng ad hoc phải được thiết kế để đảm bảo các yêu cầu kỹ thuật sau:

- Thuật toán phải được thiết kế sao cho phù hợp với tính động của topo mạng và các liên kết bất đối xứng.
- Có khả năng hoạt động phân tán: cách tiếp cận tập trung cho mạng Ad hoc sẽ thất bại do sẽ tốn rất nhiều thời gian để tập hợp các thông tin trạng thái hiện tại của mạng để tính toán rồi lại phát tán lại nó cho các nút mạng. Trong thời gian đó, cấu hình mạng có thể đã thay đổi rất nhiều.
- Tiết kiệm năng lượng và băng thông của mạng: Do các nút mạng có nguồn năng lượng hạn chế lên cần phải tính toán đến vấn đề tiết kiệm năng lượng. Giao thức định tuyến có thể cung cấp yêu cầu bảo tồn năng lượng ở các nút mạng khi có thể. Băng thông của mạng cũng cần được tính đến để tránh gây lãng phí băng thông không cần thiết.
- Không để xảy ra hiện tượng lặp định tuyến: Hiện tượng này xảy ra khi một phần nhỏ các gói tin di chuyển lòng vòng quanh mạng trong một khoảng thời gian nào đó. Giải pháp đưa ra có thể là sử dụng bộ đếm chặng trong mỗi gói tin. Mỗi khi gói tin di chuyển đến một nút mạng mới, bộ đếm chặng sẽ tăng lên một, và đến một giá trị nào đó thì gói tin sẽ bị loại bỏ.

- Bảo mật: Giao thức định tuyến của mạng Ad hoc có thể bị tấn công dễ dàng ở một số dạng như đưa ra các cập nhật định tuyến không chính xác hoặc ngăn cản việc chuyển tiếp gói tin, gián tiếp gây ra việc từ chối dịch vụ dẫn đến các gói tin không bao giờ đến được đích. Chúng cũng có thể thay đổi thông tin định tuyến trong mạng, cho dù các thông tin đó là không nguy hiểm nhưng cũng gây tổn băng thông và năng lượng, vốn là những tài nguyên "quý hiếm" trong mạng Ad hoc. Do vậy cần có những phương pháp bảo mật thích hợp để ngăn chặn việc sửa đổi hoạt động của giao thức.

2.2. Giao thức định tuyến DSDV

2.2.1. Tổng quan về giao thức định tuyến DSDV

Giao thức định tuyến DSDV [6] là một giao thức định tuyến "tìm đường trước" dạng véc tơ khoảng cách điển hình. Nó không chỉ giải quyết vấn đề định tuyến đối với mạng ad hoc mà còn mô tả cách thực hiện các chức năng định tuyến ở Tầng 2. Các gói tin được truyền giữa các trạm của một mạng bằng cách sử dụng các bảng định tuyến được lưu trữ tại mỗi một trạm. Mỗi một bảng định tuyến tại mỗi một trạm làm việc là một danh sách các đích có thể đến từ trạm đó và số lượng các chặng phải chuyển qua để đến được mỗi đích. Mỗi dòng trong bảng định tuyến được gán thêm một số thứ tự, số này được trạm đích thành lập đầu tiên. Để duy trì tính nhất quán của các bảng định tuyến khi hình trạng mạng thay đổi động, mỗi một trạm phải định kỳ gửi và nhận các thông tin cập nhật hoặc gửi ngay các thông tin cập nhật khi phát hiện ra sự thay đổi. Vì vấn đề lưu trữ bất kỳ dạng thông tin nào về thời gian đồng bộ trên các máy không đặt ra ở đây nên mối quan hệ theo pha về thời gian cập nhật giữa các trạm di động cũng không được tính đến. Trong các thuật toán định tuyến dạng véc tơ khoảng cách,

mỗi trạm sẽ sử dụng các gói tin được quảng bá từ các trạm khác để xác định từ nó có thể truy cập được tới những trạm nào và số lượng các chặng cần thiết để tới được trạm cần truy cập. Vấn đề đưa ra ở đây chỉ tập trung vào việc đưa các con số về thứ tự liên kết với một con đường chứ không tập trung vào việc đưa ra các độ đo về khoảng cách cho một con đường đó. Các gói tin được truyền sẽ chứa hoặc là địa chỉ Tầng 2 (MAC) hoặc là địa chỉ Tầng 3 (Network).

2.2.2. Bảng định tuyến và thông tin quảng bá đường

Thông tin định tuyến trong DSDV được đưa ra bằng các gói tin quảng bá được truyền dưới dạng broadcasting hoặc multicasting. Các gói tin này được truyền một cách định kỳ và sẽ được truyền bổ sung khi phát hiện thấy hình trạng mạng thay đổi, chẳng hạn như khi các trạm di chuyển trong một mạng. Dữ liệu về khoảng thời gian từ lúc *con đường đầu tiên* đến nơi và *con đường tốt nhất* đến nơi cho mỗi một trạm đích sẽ được lưu lại. Quyết định về thời gian trễ giữa hai lần quảng bá thông tin định tuyến sẽ được đưa ra trên cơ sở dữ liệu này để làm giảm bớt sự thay đổi bất thường trong các bảng định tuyến. Khoảng thời gian trễ giữa hai lần quảng bá có thể không cần cố định để giảm bớt số lượng các con đường đã được quảng bá đã có trong các bảng định tuyến với cùng một số thứ tự.

Giao thức DSDV yêu cầu mỗi một trạm di động phải quảng bá tới các trạm lân cận với nó về bảng định tuyến của chính nó (ví dụ, quảng bá các dòng của nó). Các dòng trong danh sách này có thể thay đổi động theo thời gian vì vậy số lần quảng bá phải được thực hiện phải đủ lớn để đảm bảo rằng mọi máy tính di động đều luôn luôn có thể xác định được các máy tính khác trong vùng. Thêm vào đó, mỗi một máy tính di động phải chuyển tiếp các gói dữ liệu tới các máy tính khác trên cơ sở được yêu cầu. Việc chuyển tiếp này tạo khả năng xác định

một con đường tới đích với số lượng các chặng chuyển tiếp là nhỏ nhất. Một máy tính di động có thể trao đổi dữ liệu với bất kỳ một máy tính di động nào khác trong nhóm mặc dù máy tính kia không nằm trong vùng truyền thông trực tiếp. Nếu DSDV hoạt động được tại tầng 2 thì nó sẽ hoạt động được ở các tầng cao hơn.

Mọi máy tính phải hợp tác hoạt động với nhau để tạo các con đường truyền dữ liệu giữa chúng và quảng bá các dữ liệu cần thiết một cách định kỳ, thông thường là vài giây một lần. Trong môi trường không dây, điều quan trọng là thông tin quảng bá bị giới hạn bởi phạm vi truyền của môi trường vật lý. Điều này khác với tình huống của các môi trường truyền bằng cáp là môi trường mà phạm vi truyền tới đích nhận được xác định một cách chính xác. Dữ liệu quảng bá từ mỗi một máy tính di động bao gồm một số thứ tự mới của nó và các thông tin sau về mỗi một con đường mới:

- Địa chỉ đích;
- Số lượng các nút chuyển tiếp để đến được đích;
- Số thứ tự của các thông tin nhận được đối với đích, được gán nhãn ban đầu bằng đích.

Các bảng định tuyến được truyền cũng chứa địa chỉ phần cứng, và (nếu thích hợp) địa chỉ mạng của các máy tính truyền chúng trong các header của các gói tin. Các con đường có các số thứ tự mới nhất sẽ luôn được ưu tiên trong các quyết định chuyển tiếp, nhưng không cần thiết phải quảng bá. Với các con đường có cùng số thứ tự, con đường có độ đo nhỏ nhất sẽ được sử dụng. Khi truyền đi các bảng định tuyến, số thứ tự cũng sẽ được gửi theo tới tất cả mọi máy

tính di động khác và mỗi máy sẽ tự quyết định duy trì mỗi dòng định tuyến đến máy nguồn.

Các con đường được nhận được theo cách quảng bá sẽ được máy nhận tiếp tục gửi tới các máy khác khi nó quảng bá các thông tin của mình; máy nhận sẽ tăng thêm độ đo về chiều dài con đường trước khi quảng bá con đường này trong trường hợp các gói tin dữ liệu yêu cầu một hoặc nhiều nút chuyển tiếp để tới được đích.

2.2.3. Tần suất quảng bá đường

Một trong những tham số quan trọng nhất khi lựa chọn là khoảng thời gian giữa hai lần quảng bá các gói tin chứa thông tin định tuyến. Tuy nhiên, khi một máy tính di động nhận được bất kỳ một thông tin mới hoặc thông tin sửa đổi về đường đi, thông tin này sẽ được truyền đi ngay lập tức. Điều này tạo nên hiệu ứng phân tán thông tin định tuyến mới một cách nhanh nhất có thể giữa các máy tính di động trong vùng làm việc. Việc quảng bá lại một cách nhanh chóng này đưa ra một yêu cầu mới cho các giao thức phải hội tụ một cách nhanh nhất có thể. Việc một máy tính di động di chuyển có thể gây ra một cơn bão quảng bá, làm thiệt hại và giảm giá trị của môi trường không dây. Các máy tính di động gây ra sự gián đoạn liên kết khi chúng di chuyển từ nơi này đến nơi khác. Liên kết bị đứt có thể nhận ra bằng các giao thức ở Tầng 2 hoặc khi một máy không nhận được các thông tin quảng bá từ một trạm lân cận nào đó trong một khoảng thời gian xác định. Một liên kết bị đứt được mô tả bằng một độ đo có giá trị là ∞ (chẳng hạn, là bất kỳ một giá trị nào lớn hơn đơn vị đo lớn nhất có thể có). Khi một liên kết tới nút tiếp theo bị đứt, bất kỳ con đường nào tới nút đó cũng đều sẽ được gán độ đo khoảng cách bằng ∞ . Khi có một sự thay đổi nào đó về đường đi

xảy ra, thông tin về con đường sửa đổi được đưa vào các gói tin quảng bá định tuyến. Việc xây dựng thông tin mô tả về các liên kết bị đứt chỉ xảy ra khi số thứ tự được các trạm di động khác với trạm di động đích sinh ra. Các số thứ tự được các trạm di động ban đầu định nghĩa là các số chẵn và các số thứ tự được sinh ra để chỉ định các độ đo ∞ là các số lẻ. Theo cách này, bất cứ một số thứ tự “thực” lẻ nào cũng có thể được sử dụng để thay thế một độ đo bằng ∞ . Khi một nút nhận được một độ đo ∞ , và sau đó nó có một số thứ tự có độ đo hữu hạn, nó sẽ kích hoạt công việc quảng bá thông tin định tuyến cập nhật để phân bố các tin tức quan trọng về một đích nào đó.

Trong các mạng có số lượng các trạm di động lớn, sự thay đổi có thể diễn ra trong khoảng giữa các lần quảng bá thông tin định tuyến định kỳ. Để giảm bớt lượng thông tin trong các gói tin quảng bá định tuyến này, có hai dạng gói tin này được định nghĩa. Dạng thứ nhất sẽ mang đầy đủ các thông tin trong gói tin được gọi là dạng “*đầy đủ*”. Dạng thứ hai sẽ chỉ mang thông tin bị thay đổi kể từ lần định tuyến quảng bá đầy đủ cuối cùng trước đó và được gọi là dạng “*bổ sung*”. Theo thiết kế, thông tin ở dạng cập nhật định tuyến *bổ sung* sẽ chứa vừa trong một đơn vị dữ liệu giao thức mạng NPDU. Dạng thông tin cập nhật định tuyến *đầy đủ* thường yêu cầu nhiều NPDU, thậm chí đối với một mạng có số lượng các trạm di động nhỏ. Khi các trạm di động không di chuyển, dạng thông tin cập nhật định tuyến đầy đủ sẽ được truyền đi. Khi sự di chuyển thường xuyên xảy ra và kích thước của thông tin định tuyến bổ sung gần với kích thước của một NPDU thì dạng thông tin định tuyến đầy đủ lại được đưa vào kế hoạch. Mỗi một nút di động đều muốn xác định sự thay đổi đường đi nào là đủ quan trọng để gửi đi trong mỗi một lần quảng bá bổ sung. Ví dụ, khi một con đường ổn định đưa ra một độ đo khác cho một vài đích, sự thay đổi này là một sự thay đổi quan

trọng cần được quảng bá dạng bổ sung. Nếu một trạm nhận được một con đường có số thứ tự mới nhưng độ đo lại không thay đổi thì đây không phải là thông tin quan trọng và thường không được đưa vào các quảng bá bổ sung.

Khi một trạm di động nhận thông tin định tuyến mới (thông thường dưới dạng một gói quảng bá bổ sung đã mô tả ở trên), thông tin này được so sánh với các thông tin đã có từ các gói tin mang thông tin định tuyến trước đó. Con đường nào có số thứ tự mới nhất sẽ được sử dụng. Các con đường có số thứ tự cũ hơn sẽ bị bỏ qua. Một con đường có số thứ tự bằng với một con đường đã có, nếu có độ đo “tốt hơn” thì nó sẽ được sử dụng, và con đường đã có sẽ bị bỏ qua hoặc được lưu trữ dưới dạng kém ưu tiên hơn. Các độ đo cho các con đường được chọn từ thông tin quảng bá mới nhận được sẽ tăng thêm một chặng trước khi lưu trữ. Các con đường mới được ghi lại sẽ được đưa vào kế hoạch quảng bá tiếp theo ngay lập tức tới các trạm di động lân cận của trạm hiện tại. Các con đường có độ đo được cải thiện sẽ được đưa vào kế hoạch quảng bá tại thời điểm tiếp theo. Thời điểm tiếp theo này phụ thuộc vào thời gian dàn xếp trung bình cho các con đường tới được đích xác định.

Thời gian lệch giữa các trạm di động là hoàn toàn có thể xảy ra. Việc quảng bá các gói tin định tuyến từ các trạm di động đôi khi còn được xem là các sự kiện bất đồng bộ. Với trường hợp có một mật độ các tác tử truyền độc lập, việc sử dụng các thủ tục cập nhật đường đi ở trên có thể làm xuất hiện những sự thay đổi bất thường. Điều này có thể gây ra việc một trạm di động nhận được các thông tin định tuyến mới ở dạng mà nó sẽ thay đổi các con đường từ một chặng tiếp theo này sang một chặng tiếp theo khác, thậm chí cả khi trạm di động đích không di chuyển. Điều này xảy ra bởi vì có hai cách chọn một con đường mới; cách thứ nhất là con đường này có số thứ tự mới nhất, cách thứ hai là con đường

này có độ đo tốt nhất. Một trạm di động có thể luôn luôn nhận được hai con đường đi tới cùng một đích nhưng con đường có số thứ tự mới nhất lại đến sau (qua các trạm hàng xóm khác) thì trạm này luôn chọn con đường có độ đo tốt hơn đến trước. Điều này sẽ làm cho trạm đó tiếp tục quảng bá con đường đã chọn đến các trạm khác, và quá trình này tiếp tục xảy ra khi nó đến một trạm tiếp theo.

Một phương án là làm trễ thời gian quảng bá những con đường kiểu như vậy, khi một trạm di động có thể xác định rằng một con đường với độ đo tốt hơn có thể truyền đến nó. Con đường có số thứ tự mới nhất vẫn được sử dụng ở trạm đó nhưng nó chưa được quảng bá ngay lập tức trừ khi nó là con đường tới được đích mà trước đó không thể tới được. Do đó, sẽ có hai bảng định tuyến được lưu trữ trong mỗi một trạm di động; một bảng được sử dụng để chuyển tiếp các gói tin dữ liệu và một bảng kia được sử dụng trong các gói tin định tuyến quảng bá bổ sung. Để xác định xác suất nhận được thông tin định tuyến có độ đo tốt hơn, trạm di động này sẽ phải lưu trữ một lịch sử về trọng số thời gian trung bình của các con đường tới một đích xác định nào đó bị thay đổi cho đến khi nhận được con đường có độ đo tốt nhất. Thủ tục này cho phép một trạm dự đoán được khoảng thời gian phải chờ đợi trước khi tiến hành quảng bá thông tin về một con đường mới.

2.2.4. Hoạt động của giao thức DSDV ở Tầng 2

Các địa chỉ lưu trữ trong các bảng định tuyến sẽ tương ứng với tầng mà giao thức định tuyến của mạng không dây kiểu không cấu trúc hoạt động. Nghĩa là, nếu giao thức định tuyến hoạt động tại Tầng 3, nó sẽ sử dụng địa chỉ mạng làm địa chỉ cho các chặng tiếp theo và địa chỉ đích. Nếu giao thức định tuyến

hoạt động tại Tầng 2, nó sẽ sử dụng địa chỉ MAC. Tuy nhiên, việc sử dụng địa chỉ MAC của Tầng 2 sẽ phát sinh một số yêu cầu mới. Điều khó khăn là các giao thức mạng Tầng 3 cung cấp cơ chế truyền thông trên cơ sở địa chỉ mạng và cơ chế để chuyển đổi từ các địa chỉ Tầng 3 thành các địa chỉ MAC. Mặt khác, khi có nhiều cơ chế chuyển đổi địa chỉ khác nhau có thể được sử dụng và kết quả là làm giảm băng thông của môi trường truyền không dây khi thực hiện các cơ chế chuyển đổi này.

Điều đáng nói là những cơ chế chuyển đổi địa chỉ như vậy có thể yêu cầu quảng bá tại một trạm và liên tiếp quảng bá tại các trạm tiếp theo trong mạng không dây kiểu không cấu trúc. Do đó, hầu như mọi cơ chế chuyển đổi địa chỉ đều hoạt động không đồng đều trong các thao tác cơ bản của mạng này. Phương án giải quyết ở đây cho hoạt động của giao thức tại Tầng 2 là đưa vào các thông tin của giao thức Tầng 3 cùng với các thông tin của giao thức Tầng 2. Mỗi một trạm đích sẽ quảng bá giao thức Tầng 3 nào mà nó hỗ trợ và mỗi một trạm di động sẽ liên tiếp quảng bá cho tới đích với thông tin quảng bá bao gồm cả các thông tin về giao thức Tầng 3 mà trạm đích hỗ trợ. Thông tin này chỉ được truyền khi nó thay đổi. Sự thay đổi sẽ được truyền đi như là một phần của thông tin định tuyến quảng bá bổ sung.

2.3. Giao thức định tuyến AODV

2.3.1. Giới thiệu về giao thức AODV

AODV [5] là một giao thức định tuyến động, hoạt động theo yêu cầu, đa chặng và tự khởi động giữa các nút di động trong mạng không dây phi cấu trúc. Nó cho phép các nút tìm được các đường tới một đích một cách nhanh chóng và không yêu cầu các nút duy trì các con đường tới đích khi không truyền thông.

Đồng thời, giao thức này cho phép các nút di động làm việc được với sự thay đổi hình trạng của mạng hoặc liên kết bị đứt.

AODV là giao thức có khả năng tránh định tuyến lặp và có tốc độ hội tụ nhanh khi hình trạng mạng thay đổi. Khi một liên kết bị đứt, AODV sẽ tạo ra hiệu ứng để báo cho tập các nút liên quan cập nhật thông tin về đường bị lỗi. Giao thức này sử dụng số thứ tự đích cho mỗi entry trong bảng định tuyến để biểu diễn “độ mới” của đường. Số thứ tự đích do nút đích tạo ra được đưa vào các gói tin điều khiển cùng với các thông tin định tuyến khác và được gửi đi đến nút có yêu cầu tìm đường. Nút yêu cầu sẽ lựa chọn một con đường có số thứ tự lớn nhất.

Các gói yêu cầu đường (RREQ), trả lời đường (RREP), báo lỗi đường (RERR) và gói Hello là các gói điều khiển được định nghĩa trong AODV. Khi một nút cần tìm đường đến đích, nó sẽ quảng bá gói RREQ. Quá trình quảng bá gói RREQ tạo ra các đường nghịch (reverse route) hướng tới nút nguồn tại các nút nhận gói. Khi một nút nhận được gói RREQ, nếu nó là nút đích hoặc là nút trung gian nhưng có thông tin về đường “*đủ mới*” thỏa mãn yêu cầu của nút nguồn, nó sẽ gửi gói RREP dạng unicast tới nút nguồn để trả lời đường. Quá trình truyền gói RREP tạo ra các đường thuận (forward route) hướng tới nút đích tại các nút nhận gói. Gói Hello được sử dụng để theo dõi trạng thái của liên kết. Khi một liên kết thuộc một đường bị đứt gói RERR được sử dụng để báo lỗi đường cho các nút láng giềng qua “*danh sách con trở trước*”.

Quản lý số thứ tự là một việc thiết yếu để tránh định tuyến lặp. Một nút đích sẽ trở thành nút không đến được khi một liên kết bị đứt hoặc đang ở trạng thái không hợp lệ. Khi những điều kiện này xảy ra, đường chứa liên kết này sẽ

được coi là mất hiệu lực bằng thao tác gán số thứ tự và đánh dấu trong bảng định tuyến là đường không hợp lệ.

2.3.2. Cơ chế hoạt động của giao thức AODV

◆ Duy trì các số thứ tự

Số thứ tự được gán cho mỗi đường trong bảng định tuyến là “độ mới” của con đường. Nó được gọi là “số thứ tự đích” vì nó biểu diễn “độ mới” cho con đường tới một đích xác định để tránh định tuyến lặp. Nó được cập nhật khi một nút nhận được thông tin mới hơn từ các thông điệp RREQ, RREP hoặc RERR liên quan đến đích. Số thứ tự đích được mỗi nút duy trì một cách độc lập. Một nút sẽ tăng số thứ tự của nó: (1) trước khi gửi gói RREQ; (2) trước khi gửi gói RREP; (3) khi nhận được thông tin về trạng thái liên kết tới chặng kế tiếp của đường tới đích bị lỗi.

Để khẳng định rằng thông tin về đường đi tới một đích là mới, một nút chỉ cập nhật thông tin từ các gói điều khiển của AODV nó nhận được khi số thứ tự đích của gói lớn hơn số thứ tự đích hiện tại của nó.

◆ Bảng định tuyến và các danh sách con trỏ trước

Bảng định tuyến của AODV bao gồm các entry, mỗi entry là biểu diễn một đường tới một đích, chứa các thông tin về IP đích, số thứ tự đích, các cờ trạng thái, giao tiếp mạng, số chặng, chặng kế tiếp, danh sách con trỏ trước và thời gian sống của đường.

Khi một nút nhận được gói RREQ, RREP hoặc RERR, nó sẽ kiểm tra bảng định tuyến đã có entry biểu diễn đường tới đích. Nếu chưa có, nó sẽ tạo entry mới. Một entry chỉ được cập nhật nếu số thứ tự đích của nó: (i) cao hơn số thứ tự

đích trong bảng định tuyến; (ii) bằng với số thứ tự đích trong bảng định tuyến nhưng số chặng của đường mới cộng một nhỏ hơn số chặng hiện tại của entry hiện tại; (iii) chưa được biết đến.

Thời gian hoạt động của mỗi entry được xác định từ gói điều khiển nhận được hoặc được khởi tạo tới bằng giá trị `ACTIVE_ROUTE_TIMEOUT`. Khi entry được sử dụng để chuyển tiếp gói dữ liệu, giá trị trường này được cập nhật bằng thời gian hiện tại cộng với `ACTIVE_ROUTE_TIMEOUT`. Nếu thời gian hiện tại lớn hơn giá trị trường này, entry sẽ được đánh dấu là không hợp lệ.

Với mỗi entry trong bảng định tuyến duy trì danh sách các con trỏ trước. Các nút trong danh sách này sẽ nhận các thông báo về sự kiện liên kết tới chặng kế tiếp bị đứt. Danh sách các con trỏ trước chứa các địa chỉ các nút láng giềng của các đường nghịch.

◆ Tạo gói yêu cầu tìm đường RREQ

Giao thức AODV sử dụng gói tin RREQ để gửi yêu cầu tìm đường. Cấu trúc gói RREQ được biểu diễn trong Hình 1.6.

Một nút sẽ gửi gói RREQ khi cần chuyển tiếp một gói dữ liệu tới một đích nhưng nó không có entry hợp lệ trong bảng định tuyến. Trường Destination Sequence Number được thiết lập bằng số thứ tự đích của entry có đích tương ứng trong bảng định tuyến. Nếu entry này không tồn tại, cờ „U” được thiết lập là True. Trường Originator Sequence Number được thiết lập giá trị bằng số thứ tự của nút cộng một. Trường RREQ ID được thiết lập bằng giá trị RREQ ID của nút cộng một. Mỗi nút duy trì giá trị RREQ ID một cách độc lập. Trường Hop Count được thiết lập bằng 0. Trường Originator IP Address và Destination IP Address chứa địa chỉ IP tương ứng của nút hiện tại (nguồn) và nút đích.

0								1								2								3								
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
Type								J	R	G	D	U	Reversed								CF				Hop Count							
RREQ ID																																
Destination IP Address																																
Destination Sequence Number																																
Source IP Address																																
Source Sequence Number																																

Hình 2.1. Cấu trúc gói RREQ

Trước khi quảng bá gói RREQ, nút nguồn lưu giá trị trường RREQ ID và Originator IP Address trong khoảng thời gian PATH_DISCOVERY_TIME. Khi một nút nhận lại gói tin chính gói tin này từ các nút láng giềng, gói tin sẽ không được xử lý và chuyển tiếp. Nếu cờ „G” được thiết lập, khi nút trung gian gửi gói RREP để trả lời đường, nó sẽ thông tin cho nút đích đường quay trở lại nút nguồn.

Số gói RREQ được tạo trong một giây phải nhỏ hơn RREQ_RATELIMIT. Sau khi gửi gói RREQ, nút nguồn đợi gói RREP trong khoảng thời gian NET_TRAVERSAL_TIME. Sau khoảng thời gian này, nút nguồn sẽ quảng bá một gói RREQ khác với số lần gửi lại lớn nhất là RREQ_RETRIES. Để giảm tắc nghẽn, giá trị của khoảng thời gian đợi gói RREP ở lần truyền lại gói RREQ thứ n sẽ là ($2^n \times \text{NET_TRAVERSAL_TIME}$).

Dữ liệu cần chuyển tiếp trong thời gian tìm đường được lưu trữ vào bộ nhớ đệm kiểu FIFO. Nếu sau RREQ_RETRIES lần gửi lại gói RREQ, nút nguồn

không tìm được đường, dữ liệu trong bộ nhớ đệm sẽ bị xóa và thông điệp Destination Unreachable sẽ được gửi tới ứng dụng.

❖ Điều khiển truyền gói RREQ

Để hạn chế sự quảng bá của gói RREQ, các nút trong giao thức AODV có thể sử dụng thuật toán tìm đường mở rộng dần theo vòng trên cơ sở thay đổi giá trị trường TTL trong gói IP chứa gói RREQ. Giá trị trường này được khởi tạo ở lần tìm đường thứ nhất là TTL_START. Nếu sau thời gian RING_TRAVERAL_TIME, nút nguồn không nhận được gói trả lời đường RREP, nó sẽ tăng giá trị trường TTL lên TTL_INCREMENT đơn vị và gửi lại gói RREQ. Việc này sẽ tiếp tục cho đến khi TTL được thiết lập trong RREQ đạt đến ngưỡng TTL_THRESHOLD, ngoại trừ trường hợp TTL=NET_DIAMETER được sử dụng cho mỗi lần thử truyền lại RREQ. Sau mỗi lần, khoảng thời gian timeout chờ để nhận một thông điệp RREP là RING_TRAVERAL_TIME. Khi muốn thông điệp RREQ đi qua toàn bộ mạng trong mọi lần thử truyền lại, cần thiết lập cả giá trị TTL_START và TTL_INCREMENT bằng giá trị NET_DIAMETER.

Nếu trong bảng định tuyến tồn tại một entry tới đích nhưng không còn hợp lệ, giá trị khởi tạo của trường TTL được thiết lập bằng tổng của số chặng trong entry và TTL_INCREMENT.

❖ Xử lý và chuyển tiếp các thông điệp RREQ

Khi một nút nhận được một gói RREQ, đầu tiên nó sẽ tạo hoặc cập nhật entry biểu diễn đường tới nút gửi gói RREQ cho nó. Sau đó, nó sẽ kiểm tra để xác định nó đã được nhận gói này trước đó chưa. Nếu đã nhận, nút này sẽ huỷ bỏ gói RREQ. Nếu chưa nhận được, nó sẽ thực hiện các việc sau: (1) tăng giá trị

trường hop count trong gói RREQ được tăng lên một đơn vị; (2) tìm đường nghịch có đích là Originator IP Address của gói RREQ trong bảng định tuyến của mình. Nếu chưa có thì tạo đường nghịch mới. Nếu đã có thì cập nhật đường nghịch nếu đường nghịch nhận được có số thứ tự mới hơn số thứ tự của đường hiện tại; (3) cập nhật số thứ tự đích của nút nếu số thứ tự đích của gói RREQ lớn hơn số thứ tự hiện tại của nút; (4) chuyển tiếp gói RREQ nếu không có đường tới đích hoặc trả lời bằng gói RREP nếu có đường tới đích. Trong trường hợp gói RREQ cần được chuyển tiếp, giá trị trường TTL được trừ đi 1 đơn vị và gói này được gửi kiểu broadcast trên tất cả các giao tiếp mạng của nút. Trong trường hợp trả lời bằng gói RREP, nếu giá trị cờ „G” trong gói RREQ bằng 1, gói RREQ sẽ tiếp tục được gửi kiểu unicast đến nút đích.

◆ Tạo gói trả lời đường RREP

Giao thức AODV sử dụng gói tin RREP để trả lời truy vấn đường. Cấu trúc gói RREP được biểu diễn trong Hình 1.7.

Một nút tạo gói RREP nếu nó là nút đích hoặc nó có entry biểu diễn đường tới nút đích có giá trị số thứ tự đích lớn hơn hoặc bằng với số thứ tự đích của gói RREQ nó nhận được.

Khi tạo ra một thông điệp RREP, giá trị trường Destination IP Address và trường Originator IP Address được sao chép từ thông điệp RREQ. Sau đó, gói RREP được truyền kiểu unicast theo đường nghịch mà gói RREQ đã đi qua. Tại mỗi nút trung gian, giá trị trường Hop Count sẽ được tăng 1 đơn vị.

Nếu nút tạo gói RREP là nút đích, nó phải tăng số thứ tự của nó lên một đơn vị nếu số thứ tự trong thông điệp RREQ lớn hơn số thứ tự hiện tại của nó. Nút đích sẽ thiết lập giá trị cho trường Destination Sequence Number bằng số

thứ tự của nó, thiết lập giá trị trường Hop Count bằng 0, thiết lập giá trị trường Lifetime bằng giá trị MY_ROUTE_TIMEOUT của nó.

0								1								2								3									
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31		
Type								R	A	Reversed								Prefix Sz								Hop Count							
RREQ ID																																	
Destination IP Address																																	
Destination Sequence Number																																	
Originator IP Address																																	
Lifetime																																	

Hình 2.2. Cấu trúc gói RREP

Gọi entry trong bảng định tuyến chứa đường tới nút đích là entry hiện tại. Nếu nút tạo thông điệp RREP không phải là nút đích, nó sẽ thiết lập giá trị trường Destination Sequence Number bằng giá trị entry hiện tại; cập nhật các danh sách con trở trước của entry biểu diễn đường thuận tới đích và đường nghịch tới nguồn tương ứng bằng địa chỉ IP của nút gửi gói RREQ cho nó và địa chỉ IP của chặng kế tiếp thuộc đường tới đích; thiết lập trường Hop Count bằng giá trị Hop Count của entry hiện tại; thiết lập giá trị trường Lifetime bằng hiệu giữa thời gian timeout của hiện tại và thời gian hiện tại.

❖ Nhận và chuyển tiếp gói RREP

Khi một nút nhận gói RREP, nó sẽ tạo một entry chứa đường tới nút gửi gói RREP và entry chứa đường thuận tới nút đích nếu những entry này chưa tồn tại. Sau đó, nút này sẽ tăng giá trị trường Hop Count lên một đơn vị. Nếu đã tồn tại entry chứa đường thuận tới đích, nút hiện tại sẽ cập nhật entry này nếu: (1) Số

thứ tự của entry được đánh dấu là không hợp lệ; (2) Giá trị trường Destination Sequence Number lớn hơn số thứ tự của nó; (3) Số thứ tự đích trong gói RREP bằng nhau số thứ tự của nó và giá trị Hop Count của gói RREP nhỏ hơn giá trị Hop Count của entry hiện tại. Các công việc cập nhật entry đường thuận bao gồm: Thiết lập trạng thái entry là hợp lệ; Thiết lập trạng thái số thứ tự đích của entry là hợp lệ; Thiết lập Next Hop của entry là địa chỉ IP của nút gửi gói RREP tới nó; Thiết lập giá trị trường Hop Count, Lifetime và Destination Sequence Number của entry tương ứng bằng giá trị Hop Count và Lifetime và Destination Sequence Number của gói RREP. Nút hiện tại sau đó có thể sử dụng đường biểu diễn bởi entry này để chuyển tiếp các gói dữ liệu đến đích.

Nếu nút nhận gói RREP không phải là nút khởi tạo yêu cầu tìm đường, nó sẽ tìm entry chứa đường tới nút nguồn trong bảng định tuyến để xác định nút tiếp theo nhận gói RREP được nó chuyển tiếp.

2.4. Giao thức định tuyến DSR

2.4.1. Tổng quan về giao thức DSR

Giao thức DSR [2] là một giao thức định tuyến đơn giản và hiệu quả được thiết kế để sử dụng trong các mạng không dây kiểu không cấu trúc có nhiều chặng. Nó cho phép việc tổ chức và cấu hình trên các máy di động trong một mạng có thể diễn ra tự động hoàn toàn. Giao thức này bao gồm hai cơ chế chính là cơ chế *Tìm đường* và cơ chế *Duy trì đường*. Hai cơ chế này phối hợp với nhau để cho phép các nút di động tìm và duy trì các con đường tới các đích bất kỳ trong mạng không dây kiểu không cấu trúc. Việc sử dụng kiểu định tuyến nguồn cho phép tránh khỏi vấn đề định tuyến vòng, các nút mạng trung gian không cần phải cập nhật liên tục các thông tin định tuyến và cho phép các nút chuyển tiếp

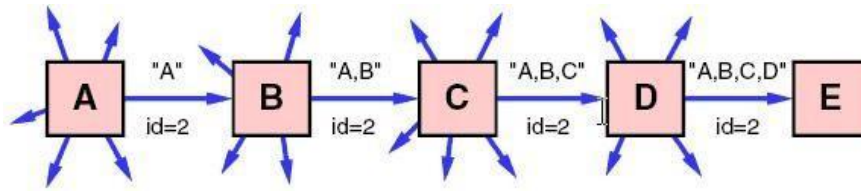
hoặc đọc và lưu các thông tin định tuyến cần thiết từ các gói dữ liệu để sau đó sử dụng.

Giao thức DSR cho phép các nút mạng tự khám phá một con đường nguồn qua các nút mạng trung gian tới bất kỳ một nút đích nào trong mạng ad hoc. Mỗi một gói dữ liệu được gửi đi sau đó sẽ chứa một danh sách đầy đủ các nút trung gian mà gói này phải đi qua để đến được đích mà không có vấn đề di chuyển theo vòng diễn ra đồng thời tránh khỏi việc cập nhật liên tục các thông tin định tuyến trên các nút trung gian chuyển tiếp gói tin dữ liệu này. Bằng cách đưa con đường nguồn vào trong phần header của các gói dữ liệu, mỗi một nút khi chuyển tiếp bất kỳ một gói tin nào dạng này cũng dễ dàng lưu trữ lại để sử dụng.

Giao thức DSR là một giao thức yêu cầu mức độ xử lý rất nhẹ nhàng để đáp ứng lại với sự thay đổi rất nhanh của hình trạng mạng và là một dịch vụ có độ tác động trở lại cao để đảm bảo có thể truyền các gói dữ liệu một cách thành công dọc theo một chuỗi các nút mạng di động hoặc điều kiện về mạng thay đổi thường xuyên.

2.4.2. Thủ tục tìm đường cơ bản của DSR

Khi một nút nguồn S tạo ra một gói tin mới có đích là nút D, nó sẽ đưa vào trong phần header của gói tin mới tạo một *con đường nguồn* cung cấp một thứ tự các chặng mà gói tin này phải đi qua trên con đường đi đến nút D. Thông thường, nút S sẽ có được một con đường phù hợp bằng cách tìm kiếm trong bộ nhớ của nó lưu các con đường mà nó đã học được. Khi không tìm thấy một con đường nào phù hợp trong bộ nhớ, nó sẽ khởi tạo tiến trình *Tìm đường* để tìm một con đường mới tới nút D. Trong trường hợp này, ta gọi S là *nút khởi tạo* và D là *nút đích* của tiến trình *Tìm đường*.



Hình 2.3. Ví dụ về một thủ tục tìm đường trong DSR

Hình 3.3 minh họa một ví dụ tìm đường, trong đó nút A đang cố gắng tìm một con đường đến nút E. Để khởi tạo thủ tục *Tìm đường*, nút A truyền một thông điệp **ROUTE REQUEST**. Đây là một thông điệp dạng broadcast mà tất cả các nút mạng trong phạm vi truyền của nút A sẽ nhận được. Trong mỗi thông điệp **ROUTE REQUEST** phải chỉ ra nút khởi tạo và nút đích của một quá trình Tìm đường và đồng thời cũng chứa một *request id* duy nhất do nút khởi tạo xác định. Ngoài ra, thông điệp này còn chứa một danh sách ghi lại các địa chỉ của mỗi nút trung gian đã chuyển tiếp nó. Danh sách ghi đường được nút khởi tạo thiết lập ban đầu là một danh sách rỗng.

Khi một nút nào đó nhận được một thông điệp **ROUTE REQUEST**, nếu nó là nút đích của thủ tục Tìm đường, nó sẽ gửi thông điệp **ROUTE REPLY** lại cho nút khởi tạo. Thông điệp **ROUTE REPLY** chứa một danh sách ghi đường được sao chép lại từ thông điệp **ROUTE REQUEST**. Khi nút khởi tạo nhận được thông điệp **ROUTE REPLY**, nó lưu trữ con đường này lại trong *bộ nhớ đường* của nó để sau đó sử dụng nhằm gửi các gói tin dữ liệu đến đích. Trường hợp nếu nút này nhận một thông điệp **ROUTE REQUEST** mà trước đó đã thấy một thông điệp **ROUTE REQUEST** khác từ nút khởi tạo có cùng *request id* với *request id* của thông điệp mà nó vừa nhận được hoặc nó tìm thấy địa chỉ của mình đã có trong danh sách ghi đường trong thông điệp **ROUTE REQUEST**, nó sẽ bỏ qua thông điệp này. Ngược lại, nút này sẽ gắn địa chỉ của nó vào danh sách ghi

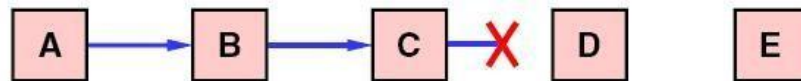
đường trong thông điệp ROUTE REQUEST và truyền broadcast thông điệp này đi.

Trong quá trình thông điệp ROUTE REPLY quay trở về nút khởi tạo, chẳng hạn như nút E trả lời lại nút A trong Hình 3.3, nút E thường sẽ kiểm tra bộ nhớ đường của nó đã có con đường quay về nút A hay chưa. Nếu nó tìm thấy, nó sẽ sử dụng con đường này làm con đường nguồn để truyền gói tin chứa thông điệp ROUTE REPLY về nút A. Ngược lại, nút E có thể thực hiện thủ tục Tìm đường của nó để tìm con đường quay về nút A. Tuy nhiên, để tránh khỏi việc này sẽ lặp lại vô hạn, khi nút E gửi thông điệp ROUTE REQUEST cho nút A, nó phải đưa thông điệp trả lời ROUTE REPLY vào cùng với gói tin chứa thông điệp ROUTE REQUEST của nó. Để đơn giản, nút E cũng có thể đảo ngược lại thứ tự các nút trong danh sách ghi đường của thông điệp ROUTE REQUEST và sử dụng danh sách đảo này như là một con đường nguồn cho gói tin mang thông điệp ROUTE REPLY của nó.

Khi khởi tạo một thủ tục Tìm đường, nút gửi sẽ ghi lại một bản sao của gói tin gốc trong bộ đệm nội bộ của nó gọi là *Bộ đệm gửi*. Bộ đệm gửi chứa một bản sao của các một gói tin chưa được nút này gửi đi bởi vì nút này chưa biết đường đến đích của các gói tin đó. Mỗi một gói tin khi đưa vào trong bộ đệm gửi sẽ được gán nhãn thời gian và sẽ bị xóa khỏi bộ đệm gửi sau khoảng thời gian timeout. Nếu cần thiết để tránh cho bộ đệm gửi bị tràn, chiến lược điều phối kiểu FIFO hoặc một chiến lược điều phối khác có thể được áp dụng để xóa các gói tin trong bộ đệm gửi trước khoảng thời gian chúng hết hạn.

2.4.3. Thủ tục duy trì đường cơ bản của DSR

Khi khởi tạo hoặc chuyển tiếp một gói tin sử dụng một con đường nguồn, mỗi một nút truyền gói tin này có nghĩa vụ phải khẳng định rằng chẳng tiếp theo dọc theo con đường gói tin đi đến đích đã nhận được gói tin. Gói tin sẽ được truyền lại cho đến khi nút này biết chắc chắn rằng nút tiếp theo đã nhận được nó.



Hình 2.4. Ví dụ về một thủ tục duy trì đường

Ví dụ, trong tình huống được minh họa trong Hình 3.4, nút A đã khởi tạo một gói tin tới nút E sử dụng một con đường nguồn thông qua các nút trung gian là B, C và D. Trong trường hợp này, nút A có trách nhiệm đảm bảo gói tin đã đến B, nút B có trách nhiệm đảm bảo gói tin đã đến C, nút C có trách nhiệm đảm bảo gói tin đã đến D và nút D có trách nhiệm đảm bảo gói tin đã đến đích E. Việc khẳng định đã nhận được gói tin trong nhiều trường hợp có thể là không có giá trị đối với DSR bởi vì hoặc là nó đã có trong một phần của giao thức MAC đang được sử dụng hoặc là nó đã được thực hiện bởi cơ chế báo nhận chủ động (ví dụ trong trường hợp này là việc B xác nhận rằng C đã nhận được bằng cách nghe xem C có chuyển tiếp gói tin này đến D hay không). Trong trường hợp không có cả hai cơ chế trên, một nút khi truyền một gói tin phải thiết lập một bit trong phần header của gói tin để yêu cầu nút tiếp theo phải cung cấp cơ chế báo nhận phần mềm. Cơ chế báo nhận phần mềm thường sẽ được truyền trực tiếp tới nút gửi nhưng nếu liên kết giữa hai nút là kiểu một chiều thì thông tin báo nhận phần mềm phải đi theo một con đường khác.

Nếu gói tin được một vài chặng khác truyền lại với số lần truyền tối đa mà các nút truyền đi không nhận được thông tin báo nhận từ các nút tiếp theo, nút này sẽ gửi một thông điệp báo lỗi ROUTE ERROR tới nút ban đầu gửi gói tin để chỉ ra rằng gói tin không thể chuyển tiếp được theo liên kết hiện tại.

2.5. Tổng kết Chương 2

Do những đặc điểm khác biệt cơ bản về tính di động, sự vắng mặt của các thiết bị hạ tầng mạng cơ sở, tính chất phân tán nên các giao thức và thuật toán định tuyến trong mạng ad hoc phải giải quyết được những yêu cầu cơ bản như khả năng tìm đường qua các liên kết bất đối xứng, tiết kiệm tài nguyên mạng, khả năng định tuyến hiệu quả khi topo mạng thay đổi liên tục, đảm bảo tính bảo mật.

Nội dung Chương 3 tập trung trình bày về cơ chế hoạt động của 3 giao thức định tuyến tiêu biểu dành cho mạng ad hoc là giao thức định tuyến dạng bảng tìm đường trước DSDV, giao thức định tuyến theo yêu cầu AODV và giao thức định tuyến nguồn DSR.

Ngay sau khi khởi động, mỗi nút mạng sử dụng giao thức định tuyến DSDV để trao đổi thông tin định tuyến của mình dưới dạng bảng với các nút lân cận. Đây là một dạng giao thức định tuyến “tìm đường trước”. Số chặng được sử dụng trong giao thức DSDV để làm độ đo của mỗi con đường. Có hai dạng quảng bá định tuyến trong giao thức DSDV. Trong khi cơ chế quảng bá định kỳ gửi đầy đủ thông tin về bảng định tuyến giữa các nút mạng thì cơ chế quảng bá bổ sung sẽ chỉ gửi những thông tin cập nhật so với lần gửi thông tin đầy đủ gần nhất để tiết kiệm tài nguyên mạng. Giao thức định tuyến DSDV sử dụng số thứ tự đích để phân biệt độ mới của các con đường.

Giao thức định tuyến AODV không tìm đường trước khi có yêu cầu truyền dữ liệu. Khi có yêu cầu truyền dữ liệu tới một đích nào đó, nút mạng mới khởi động tiến trình khám phá đường. Tiến trình này được thực hiện bằng thuật toán lan truyền gói tin yêu cầu đường RREQ qua các nút trung gian. Tại mỗi nút mạng, con đường tới nút nguồn được hình thành. Khi gói RREQ đến được nút đích hoặc nút trung gian biết đường tới nút đích, gói trả lời đường RREP sẽ được gửi lại theo một đường duy nhất tới nút nguồn. Khi có lỗi xảy ra, gói RRER được sử dụng để báo lỗi đường và kích hoạt lại tiến trình tìm đường mới. Giao thức AODV cũng sử dụng số thứ tự đích để phân biệt tính mới của một con đường và sử dụng số chặng làm độ đo giá trị của một con đường.

Giao thức DSR cũng là một giao thức định tuyến tìm đường theo yêu cầu dạng véc tơ khoảng cách, sử dụng số chặng làm độ đo định tuyến tương tự như giao thức định tuyến AODV. Tuy nhiên thông tin về các nút trung gian trên con đường được đưa vào trong các gói tin điều khiển định tuyến để mỗi nút mạng đều biết được thông tin đầy đủ về con đường được hình thành qua các nút trung gian nào. Thuật toán định tuyến này còn được gọi là định tuyến nguồn.

Việc đánh giá hiệu năng của mạng ad hoc sử dụng các giao thức trên dưới sự ảnh hưởng của tốc độ di chuyển các nút mạng và tải dữ liệu sẽ được trình bày trong Chương 3.

CHƯƠNG 3. MÔ PHỎNG VÀ ĐÁNH GIÁ HIỆU NĂNG ĐỊNH TUYẾN

3.1. Tổng quan về mô phỏng mạng

3.1.1. Khái niệm mô phỏng mạng

Theo [7], mô phỏng là “tiến trình thiết kế một mô hình của hệ thống thực và áp dụng các thí nghiệm với mô hình này nhằm mục đích hiểu được hành vi của hệ thống và/hoặc đánh giá các chiến lược cho hoạt động của hệ thống”.

Một quá trình mô phỏng có thể được xem như là một dòng các tiến trình của các thực thể trong mạng (chẳng hạn như các nút, các gói tin). Khi những thực thể này di chuyển qua hệ thống, chúng tương tác, kết hợp với các thực thể khác, kích hoạt các sự kiện làm thay đổi trạng thái của hệ thống và kết thúc tiến trình. Theo thời gian, một thực thể có thể cạnh tranh hoặc đợi các tài nguyên với các thực thể khác. Điều này dẫn đến việc phải có một trình tự thực thi một cách logic sao cho mọi hành động trong hệ thống đều được xảy ra theo nghĩa có thể hiểu và quản lý được chúng. Một trình tự thực thi đóng một vai trò quan trọng trong việc giám sát một quá trình mô phỏng và đôi khi chúng được sử dụng để nhận biết loại mô phỏng

3.1.2. Các thành phần của mô phỏng

Các thành phần cấu thành của một mô phỏng bao gồm:

❖ Thực thể (Entity)

Thực thể là các đối tượng tương tác với các đối tượng khác trong một chương trình mô phỏng để gây ra những thay đổi về tình trạng của hệ thống. Trong ngữ cảnh mạng máy tính, thực thể có thể là các nút mạng, các gói tin, các luồng gói tin hoặc các đối tượng phi vật lý chẳng hạn như đồng hồ mô phỏng. Để

phân biệt các thực thể khác nhau, người ta sử dụng các thuộc tính định danh duy nhất thực thể chẳng hạn như độ dài gói tin, số thứ tự, độ ưu tiên và phần header.

❖ Tài nguyên

Tài nguyên là một phần của một hệ thống phức tạp. Thông thường, các thực thể chia sẻ nhau một số lượng tài nguyên có hạn. Ví dụ trong mạng máy tính các tài nguyên có thể là băng thông, thời gian truyền, số lượng máy chủ.

❖ Hoạt động và Sự kiện

Theo thời gian, các thực thể sẽ tiến hành các hoạt động của mình. Hoạt động của các thực thể sẽ tạo ra các sự kiện và kích hoạt sự thay đổi trạng thái của hệ thống. Ví dụ điển hình của hoạt động là đợi và xếp hàng. Khi một máy tính cần gửi một gói tin nhưng môi trường truyền đang bận, nó sẽ đợi cho đến khi môi trường truyền rồi, khi đó thực thể gói tin đang tiến hành hoạt động đợi.

❖ Bộ lập lịch

Một bộ lập lịch bao gồm một danh sách các sự kiện và thời gian thực hiện chúng. Trong quá trình mô phỏng, các sự kiện trong bộ lập lịch sẽ được kích hoạt chạy bởi đồng hồ hệ thống mô phỏng.

❖ Biến toàn cục

Trong mô phỏng, một biến toàn cục (public/global) có thể được truy cập bởi bất kỳ một hàm hoặc một thực thể nào trong hệ thống và về cơ bản nó thường lưu trữ các giá trị chung dùng trong mô phỏng. Ví dụ như đối với mạng máy tính, biến toàn cục có thể biểu diễn độ dài của hàng đợi gói tin trong một mạng có một máy chủ, có thể biểu diễn thời gian đợi tổng của không dây hoặc có thể biểu diễn số gói tin đã được truyền.

◆ Bộ sinh số ngẫu nhiên

Một mô hình mô phỏng yêu cầu một bộ sinh số ngẫu nhiên (RNG) để sinh ra các số ngẫu nhiên trong hệ thống. Các số ngẫu nhiên được sinh ra bằng cách lấy liên tiếp các số từ một dãy xác định các số ngẫu nhiên giả cho đến khi số được lấy ra từ dãy được xem như là số ngẫu nhiên. Trong hầu hết mọi trường hợp, dãy số ngẫu nhiên giả được xác định trước và được sử dụng bởi mọi bộ sinh số ngẫu nhiên.

Trong những tình huống yêu cầu nhiều kết quả thống kê, một bộ RNG cần bắt đầu lấy số từ một vị trí khác (hạt giống – seed) trong cùng một dãy số ngẫu nhiên giả. Nói cách khác rất có thể kết quả của mọi lần chạy sẽ là giống nhau. Khi triển khai trong thực tế, một bộ RNG sẽ khởi tạo một hạt giống. Một hạt giống sẽ xác định vị trí bắt đầu trong dãy số ngẫu nhiên giả mà trong đó bộ RNG sẽ bắt đầu lấy số. Các mô phỏng khác nhau sẽ khởi tạo các giá trị hạt giống khác nhau do đó kết quả sinh ra sẽ khác nhau.

◆ Bộ thu thập thống kê

Nhiệm vụ chính của một bộ thu thập thống kê là thu thập dữ liệu sinh ra từ một mô phỏng để suy ra ý nghĩa của dữ liệu thu thập được

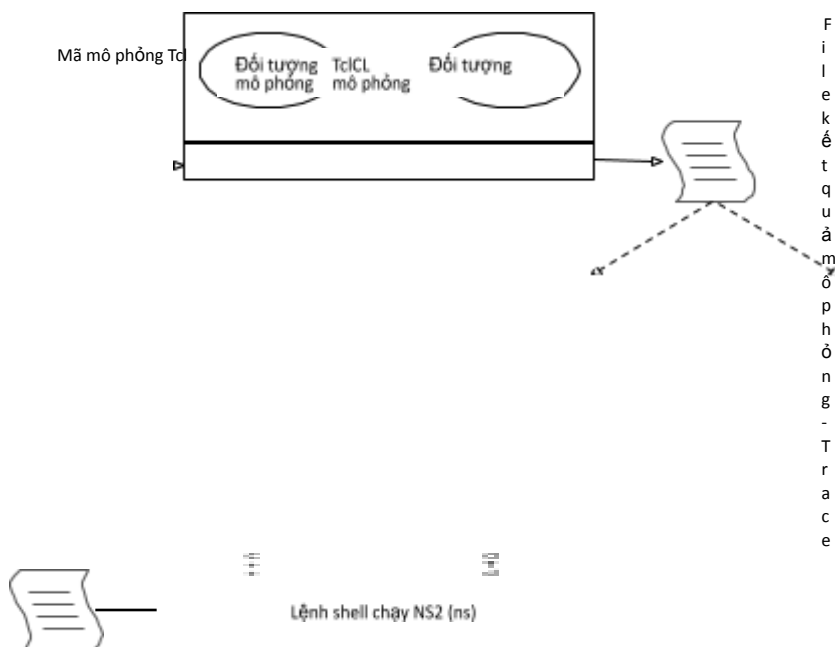
3.2. Giới thiệu về phần mềm mô phỏng NS2

NS2 (Network Simulator version 2) là một công cụ mô phỏng hướng sự kiện được xây dựng để sử dụng trong nghiên cứu về lĩnh vực mạng và truyền thông. NS2 có thể mô phỏng các chức năng và các giao thức của cả mạng có dây cũng như mạng không dây chẳng hạn như các thuật toán định tuyến, giao thức TCP, giao thức UDP,... NS2 cung cấp cho người dùng cách thức đặc tả các giao thức mạng và mô phỏng hoạt động của chúng.

Do tính chất linh hoạt và mô đun hóa tự nhiên, NS2 đã được sử dụng rộng rãi trong cộng đồng những nhà nghiên cứu về lĩnh vực mạng truyền thông kể từ khi được sinh ra vào năm 1989. Kể từ khi ra đời cho đến nay, NS2 đã tiến hóa và biến đổi mạnh mẽ đánh dấu sự lớn mạnh và trưởng thành của công cụ này với sự đóng góp của rất nhiều tổ chức và cá nhân trong lĩnh vực nghiên cứu về mạng và truyền thông. Trong số đó phải kể đến trường đại học California và trường đại học Cornell đã phát triển công cụ mô phỏng mạng có tên là REAL là nền móng cơ sở để xây dựng NS. Vào năm 1995, Cục nghiên cứu các dự án quốc phòng cao cấp của Mỹ (DARPA) hỗ trợ phát triển NS qua dự án VINT. Hiện nay, Quỹ khoa học quốc gia của Mỹ đã phối hợp tài trợ để phát triển NS. Cuối cùng, một số lượng không nhỏ những nhà nghiên cứu và nhà phát triển trên toàn thế giới đang làm việc không ngừng để góp phần làm cho NS2 ngày càng mạnh mẽ và linh hoạt hơn.

3.2.1. Kiến trúc cơ bản của NS2

Hình 3.1 cho ta thấy kiến trúc cơ bản của NS2. Người dùng sẽ sử dụng lệnh ns cùng với tham số đi kèm là tên file script mô phỏng. Thông thường sau khi chạy mô phỏng, người dùng sẽ sử dụng file trace được tạo ra để vẽ đồ thị hoặc tạo mô phỏng trực quan.



NAM (mô phỏng
trực quan)

Xgraph (vẽ biểu đồ,
đồ thị)

Hình 3.1. Kiến trúc cơ bản của NS2

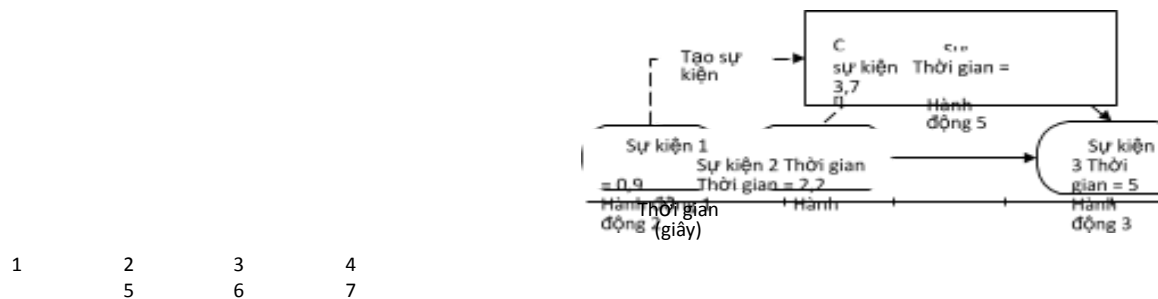
Hai ngôn ngữ chính trong NS2 là C++ và OTcl (Object-Oriented Tool Command Language). Trong khi C++ tạo ra các mô tả bên trong cho các đối tượng mô phỏng thì OTcl thiết lập mô phỏng bằng cách liên kết và cấu hình các đối tượng cũng như lập lịch cho các sự kiện rời rạc. C++ và OTcl được liên kết với nhau bằng TclCL. Một biến trong OTcl ánh xạ vào một đối tượng C++ được gọi là một tham chiếu. Về bản chất, một tham chiếu là một biến chuỗi trong miền OTcl và nó không chứa bất kỳ một hàm nào. Các hàm này được định nghĩa trong đối tượng C++ được nó ánh xạ tới. Trong miền OTcl, một tham chiếu hoạt động như một điểm truy cập để tương tác với người dùng và các đối tượng OTcl khác. Thủ tục và biến có thể được định nghĩa trong một tham chiếu để thực hiện các tương tác. Chú ý rằng trong một lớp miền OTcl, thủ tục thành viên được gọi là thủ tục instproc và biến thành viên được gọi là biến instvar.

NS2 cung cấp một số lượng lớn các đối tượng C++ đã được xây dựng sẵn. Ta nên sử dụng các đối tượng C++ này trong khi thiết lập trình mô phỏng bằng cách sử dụng mã lệnh Tcl (script). Tuy nhiên, những người dùng cấp cao có thể thấy các đối tượng C++ này chưa phù hợp với nhu cầu của họ. Trong tình huống này, họ cần phải phát triển các đối tượng C++ của chính mình và sử dụng giao tiếp cấu hình OTcl để đưa các đối tượng này vào sử dụng.

Sau khi thực hiện một mô phỏng, NS2 kết xuất kết quả dưới dạng text hoặc mô phỏng trực quan. Để dịch và biểu diễn kết quả mô phỏng trực quan, người ta sử dụng 2 công cụ là NAM (Network AniMator) và Xgraph. Một phần kết quả dạng text có thể được trích ra nhằm phục vụ cho các phân tích sâu hơn về kết quả mô phỏng.

3.2.2. Mô phỏng sự kiện rời rạc bằng NS2

NS2 là một bộ mô phỏng sự kiện rời rạc trong đó các hành động được liên kết với các sự kiện thay vì với thời gian. Một sự kiện trong bộ mô phỏng sự kiện rời rạc gồm: thời gian thực hiện, tập các hành động và tham chiếu tới sự kiện tiếp theo (Hình 4.1). Mỗi sự kiện được kết nối với các sự kiện khác hình thành lên một chuỗi các sự kiện diễn ra theo thời gian. Không giống như một bộ mô phỏng theo thời gian, trong một bộ mô phỏng theo sự kiện thời gian giữa một cặp sự kiện không nhất thiết phải là hằng số. Khi khởi động mô phỏng, các sự kiện trong chuỗi sự kiện được thực hiện từ trái qua phải (theo thứ tự thời gian).



Hình 3.2. Ví dụ về một chuỗi các sự kiện trong một mô phỏng sự kiện rời rạc.

Mô phỏng bằng NS2 bao gồm hai pha chính:

Pha I. Cấu hình mạng

Trong pha này, NS2 xây dựng mạng và thiết lập chuỗi các sự kiện ban đầu. Chuỗi các sự kiện ban đầu bao gồm các sự kiện được lập lịch để diễn ra tại các thời điểm xác định trước (Ví dụ khởi động lưu lượng FTP tại thời điểm 1 giây). Các sự kiện này được gọi là các sự kiện at-event. Pha này tương ứng với các dòng trong mã lệnh mô phỏng Tcl trước khi thực thi thủ tục run{} của đối tượng Simulator.

Pha II. Mô phỏng

Pha này tương ứng với một dòng lệnh Simulator::run{} để thực thi thủ tục run{} của đối tượng Simulator.

Trong pha này, NS2 dịch chuyển theo chuỗi các sự kiện và thực hiện từng sự kiện một theo trình tự thời gian. Ở đây, thủ tục Simulator::run{} khởi động quá trình mô phỏng bằng các thực thi sự kiện đầu tiên trong chuỗi các sự kiện. Trong NS2, thuật ngữ “*thực thi một sự kiện*” hay “*đốt cháy một sự kiện*” có nghĩa là “*thực hiện các hành động tương ứng với sự kiện*”. Ví dụ về một hành động là việc khởi động lưu lượng FTP hoặc tạo ra một sự kiện mới và chèn sự kiện được tạo ra vào chuỗi các sự kiện. Trong Hình 4.1 tại thời điểm 0,9 giây, Sự kiện 1 tạo ra Sự kiện 5 tại thời điểm 3,7 giây và chèn Sự kiện 5 vào sau Sự kiện 2. Sau khi thực thi một sự kiện, NS2 tiếp tục di chuyển theo chuỗi sự kiện và thực thi các sự kiện tiếp theo. Tiến trình này lặp lại cho đến khi sự kiện cuối cùng tương ứng với thủ tục thành viên halt{} của lớp Simulator được thực thi.

3.2.3. Cấu hình mạng ad hoc trong NS2

Việc cấu hình mạng ad hoc trong NS2 trở nên khá đơn giản khi NS2 đã được phát triển và tích hợp các thành phần mô phỏng mạng. Để cấu hình một mạng ad hoc, chỉ cần cấu hình các tham số cho mỗi nút mạng khi tạo ra. Cú pháp script TCL cấu hình nút mạng ad hoc trong NS2 như sau:

```
$ns_ node-config -<para1> <value1> [-<para2> <value2>]...[-<paran> <valuen>]
```

trong đó <para_{*i*}> và <value_{*i*}> là tên và giá trị tương ứng của tham số thứ *i*.

Danh sách các tham số có thể được cấu hình được liệt kê trong Bảng 2.1.

Tên tham số	Giá trị có thể nhận	Ý nghĩa
addressType	flat, hierarchical	Loại địa chỉ
MPLS	ON, OFF	Cho phép MPLS
wiredRouting	ON, OFF	Định tuyến mạng có dây
llType	LL, LL/Sat	Cơ chế tầng Logical Link
macType	Mac/802_11, Mac/Csma/Ca, Mac/Sat, Mac/Sat/Unslotted Aloha, Mac/Tdma	Cơ chế tầng MAC
ifqType	Queue/DropTail, Queue/DropTail/PriQueue	Kiểu hàng đợi
phyType	Phy/WirelessPhy, Phy/Sat	Cơ chế tầng Vật lý
adhocRouting	DIFFUSION/RATE, DIFFUSION/PROB, DSDV, DSR, FLOODING, OMNIMCAST, AODV, TORA, M-DART, PUMA	Loại giao thức định tuyến cho mạng ad hoc
propType	Propagation/TwoRayGround, Propagation/Shadowing	Cơ chế truyền
propInstance	Propagation/TwoRayGround, Propagation/Shadowing	Phiên bản cơ chế truyền
antType	Antenna/OmniAntenna	Loại ăng ten
channel	Channel/WirelessChannel, Channel/Sat	Loại kênh truyền
topoInstance	<topology file>	File topo
mobileIP	ON, OFF	Cho phép cơ chế mobileIP
energyModel	EnergyModel	Mô hình năng lượng
initialEnergy	<value in Joules>	Năng lượng khởi tạo
rxPower	<value in W>	Năng lượng nhận
txPower	<value in W>	Năng lượng truyền
idlePower	<value in W>	Năng lượng kênh truyền rỗi

agentTrace	ON, OFF	Kích hoạt lưu vết agent
routerTrace	ON, OFF	Kích hoạt lưu vết định tuyến
macTrace	ON, OFF	Kích hoạt lưu vết tầng MAC
movementTrace	ON, OFF	Kích hoạt lưu vết di chuyển
errProc	UniformErrorProc	Mô hình lỗi
toraDebug	ON, OFF	Kích hoạt chế độ debug của giao thức TORA
satNodeType	polar, geo, terminal, geo- repeater	Kiểu nút vệ tinh
downlinkBW	<bandwidth value>	Băng thông kênh truyền xuống

Bảng 3.1. Danh sách các tham số cấu hình nút trong NS2

3.3. Mục đích và phạm vi của việc đánh giá hiệu năng các giao thức

Các giao thức định tuyến tiêu biểu DSDV, AODV và DSR đã được trình bày trong Chương 2 sử dụng các chiến lược, thuật toán, cơ chế và kỹ thuật định tuyến khác nhau hướng tới mục tiêu định tuyến hiệu quả trong mạng ad hoc.

Đối với một hệ thống mạng bất kỳ, hiệu năng của giao thức định tuyến khi tải lưu lượng dữ liệu truyền qua mạng thay đổi là một trong những phép đo cần thiết để xem xét về tính hiệu quả và phù hợp của giao thức. Đặc biệt, đối với mạng ad hoc, tính di động của các nút mạng sẽ ảnh hưởng rất nhiều đến hiệu năng mạng. Vì vậy mục đích của chương này là đánh giá hiệu năng của mạng sử dụng các giao thức định tuyến DSDV, AODV và DSR dưới tác động của tải dữ liệu và tốc độ di chuyển của các nút mạng.

Với mục đích như trên, các thử nghiệm mô phỏng trong chương này sẽ phân tích kết quả mô phỏng để so sánh hiệu năng của mạng sử dụng giao thức định tuyến DSDV, AODV và DSR dưới tác động của các yếu tố sau:

- Tốc độ di chuyển của các nút mạng
- Thời gian tạm dừng giữa các lần di chuyển
- Số lượng kết nối dữ liệu
- Số lượng nút mạng

Các giao thức được đánh giá 4 tham số hiệu năng sau:

- Tỷ lệ truyền gói thành công: Tỷ lệ tổng số gói tin nhận được trên tổng số gói tin gửi đi. Đơn vị đo là %.
- Độ trễ đầu cuối trung bình: Thời gian trung bình để truyền một gói dữ liệu từ nút nguồn tới nút đích, đơn vị đo là giây.
- Thông lượng: Tổng lượng dữ liệu nhận được trên thời gian từ lúc gửi gói tin đầu tiên đến khi nhận gói tin cuối cùng Đơn vị đo là kpbs.
- Tải định tuyến: Tổng số gói tin điều khiển của giao thức định tuyến được các nút mạng tạo ra và chuyển tiếp.

3.4. Các thông số thiết lập mô phỏng

Các thử nghiệm mô phỏng để đánh giá hiệu năng của các giao thức định tuyến DSDV, AODV và DSR được thực hiện trên cơ sở thiết lập các thông số mô phỏng theo 4 kịch bản khác nhau nhằm đánh giá hiệu năng các giao thức này dưới tác động thay đổi của tốc độ di chuyển và tải định tuyến.

Để đánh giá tác động của tốc độ di chuyển với hiệu năng định tuyến, có 2 kịch bản được sử dụng là (1) thay đổi tốc độ di chuyển tối đa của nút mạng và (2) thay đổi thời gian tạm dừng giữa 2 lần di chuyển. Để kiểm nghiệm tác động của tải dữ liệu đối với hiệu năng định tuyến, có 2 kịch bản được sử dụng là (3) thay đổi số lượng kết nối truyền dữ liệu (số lượng luồng dữ liệu) và (4) thay đổi số lượng nút mạng.

Các thông số mô phỏng được liệt kê trong Bảng 3.2

Loại thông số	Giá trị
Giao thức định tuyến	DSDV, AODV, DSR
Giao thức tầng Transport	UDP
Diện tích vùng mô phỏng	1000 x 800
Kiểu địa chỉ	Flat
Tầng liên kết dữ liệu	LLC
Tầng MAC	802.11
Kiểu kênh truyền	Wireless
Loại ăng ten	Omni antenna
Thời gian mô phỏng	1000 s
Loại dữ liệu	CBR
Kích thước gói dữ liệu	512 bytes

Bảng 3.2. Các thông số mô phỏng

Mỗi kịch bản mô phỏng được chạy 10 lần với topo mạng được sinh ngẫu nhiên mỗi lần chạy và được sử dụng trong kịch bản mô phỏng cho cả 3 giao thức. Kết quả về hiệu năng của giao thức được tính theo giá trị trung bình cộng của 10 lần chạy cho mỗi giao thức.

3.5. Mô phỏng và đánh giá ảnh hưởng của tốc độ di chuyển

3.5.1. Ảnh hưởng của tốc độ di chuyển

Để kiểm nghiệm tác động của tốc độ di chuyển của nút mạng đến hiệu năng, mô phỏng đã được thực hiện với các thông số chung được thiết lập trong Bảng 3.2, số lượng nút là 40; tốc độ di chuyển của các nút mạng nhận các giá trị 0, 5, 10, 15, 20, 25, 30 (m/s); thời gian tạm dừng giữa 2 lần di chuyển là 10 giây; số lượng kết nối dữ liệu là 10. Kết quả của các tham số hiệu năng được đưa ra trong các Bảng 3.3 – Bảng 3.6 được tính trung bình từ kết quả của 10 lần chạy mô phỏng với topo mạng và lưu lượng khác nhau được sinh ngẫu nhiên.

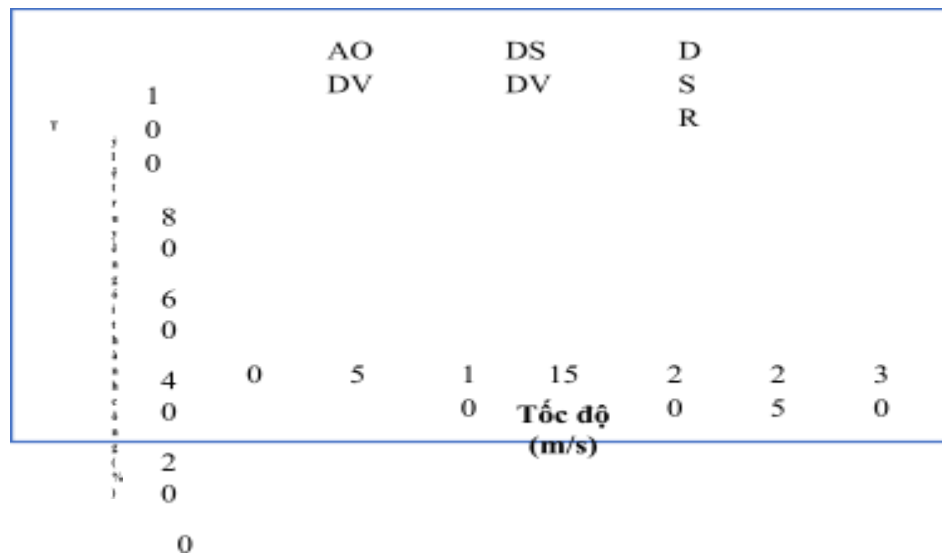
◆ Tỷ lệ truyền gói tin dữ liệu thành công

Tỷ lệ truyền gói tin dữ liệu thành công khi thay đổi tốc độ di chuyển của các nút mạng được đưa ra trong Bảng 3.3 và mối tương quan về sự thay đổi hiệu năng tỉ lệ truyền gói thành công các giao thức được biểu diễn trong Hình 3.1.

Tốc độ (m/s)	Tỷ lệ truyền gói thành công (%)		
	AODV	DSDV	DSR
0	92	94.8	89.2
5	91.2	84.6	86.6
10	85.4	76.6	72.4
15	87.4	74.8	75.6
20	90.8	76.2	74.8

25	86.2	71.6	66.6
30	84	69.4	58.6
Trung bình	88.1	78.3	74.8

Bảng 3.3. Tỷ lệ truyền gói thành công theo Tốc độ di chuyển



Hình 3.1. Biểu đồ Tỷ lệ truyền gói thành công theo Tốc độ di chuyển

Dữ liệu và hình ảnh trong Bảng 3.3 và Hình 3.1 cho thấy khi tốc độ di chuyển tăng lên, tỷ lệ truyền gói thành công của cả 3 giao thức đều có xu hướng giảm. Tuy nhiên mức độ sụt giảm về tỷ lệ truyền gói thành công là ít nhất đối với giao thức AODV và là nhiều nhất đối với DSR.

◆ Trễ đầu cuối trung bình

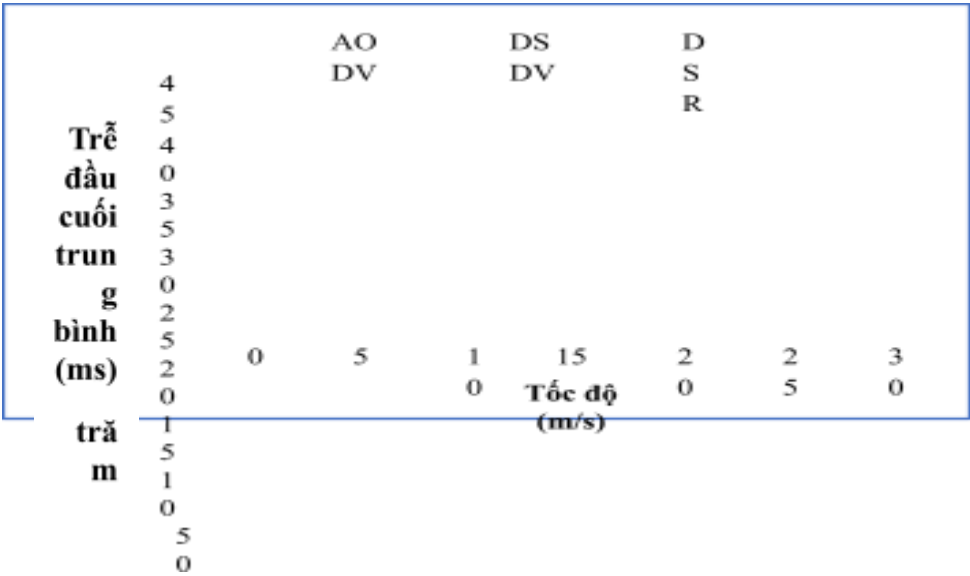
Số liệu trễ đầu cuối trung bình khi truyền các gói tin dữ liệu của 3 giao thức được đưa ra trong Bảng 3.4 và mối tương quan của sự thay đổi được minh họa trong Hình 3.2.

Dữ liệu và hình ảnh trong Bảng 3.4 và Hình 3.2 cho thấy khi tốc độ di chuyển tăng lên, thời gian trễ trung bình khi truyền một gói tin dữ liệu từ nút

nguồn tới nút đích của giao thức DSR tăng nhanh. Đối với giao thức AODV và giao thức DSDV, thời gian trễ này không bị tác động nhiều bởi tốc độ di chuyển của nút mạng. Tuy nhiên giao thức AODV vẫn có thời gian trễ trung bình truyền gói tin dữ liệu từ nguồn tới đích thấp hơn so với giao thức DSDV.

Tốc độ (m/s)	Trễ truyền gói tin trung bình (ms)		
	AODV	DSDV	DSR
0	305.4	331.6	1376.8
5	220.2	243.6	1539.8
10	224.8	366.2	2515.2
15	293.6	329.2	2491.6
20	161.8	201.2	2641.6
25	244.6	364.8	3425.4
30	279.4	395.4	4139.2
Trung bình	247.1	318.9	2589.9

Bảng 3.4. Trễ đầu cuối trung bình theo Tốc độ di chuyển



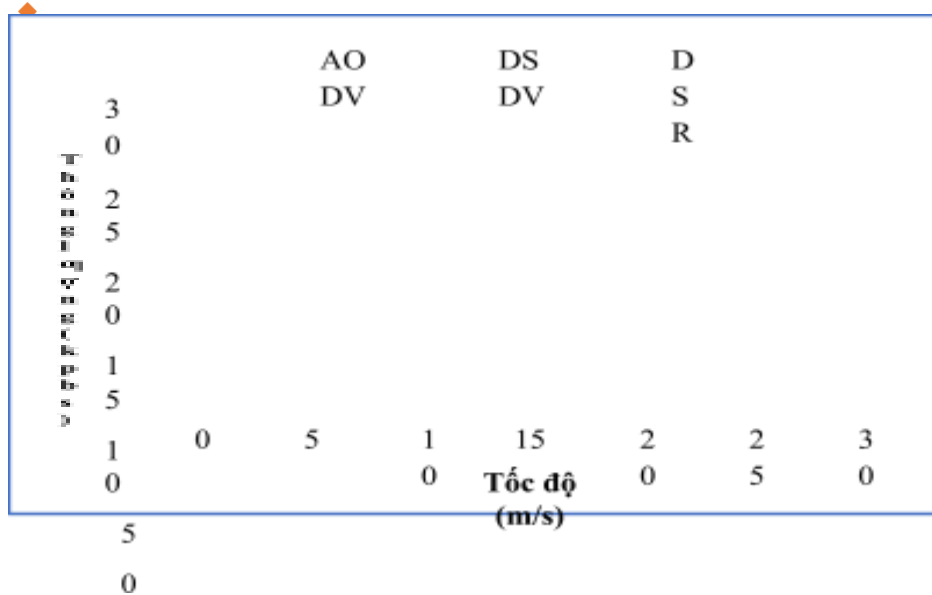
Hình 3.2. Biểu đồ Trễ đầu cuối trung bình theo Tốc độ di chuyển

❖ Thông lượng

Số liệu về thông lượng của 3 giao thức được đưa ra trong Bảng 3.5 và mối tương quan của sự thay đổi thông lượng được minh họa trong Hình 3.3.

Tốc độ (m/s)	Thông lượng (kbps)		
	AODV	DSDV	DSR
0	27	28	25.2
5	27	25	24.4
10	24.8	22.4	20.6
15	25.8	21.8	21.4
20	26.8	22.4	21
25	25.2	20.8	19
30	24.6	20.2	16.6
Trung bình	25.9	22.9	21.2

Bảng 3.5. Thông lượng theo Tốc độ di chuyển



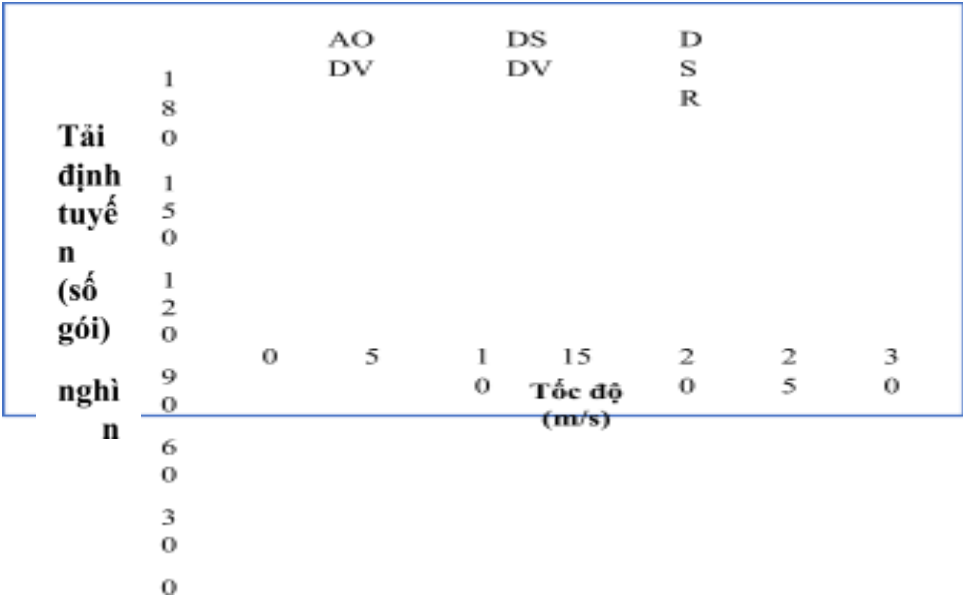
Hình 3.3. Biểu đồ Thông lượng theo Tốc độ di chuyển

Dữ liệu và hình ảnh trong Bảng 3.5 và Hình 3.3 cho thấy khi tốc độ di chuyển tăng lên, thông lượng của cả 3 giao thức đều có xu hướng giảm. Tuy nhiên mức độ giảm về thông lượng của giao thức AODV là ít nhất và của giao thức DSR là nhiều nhất.

❖ Tải định tuyến

Tốc độ (m/s)	Tải định tuyến (nghìn gói)		
	AODV	DSDV	DSR
0	105.7	74.5	86.8
5	114.4	74.3	97.5
10	144.3	75.9	120.3
15	133.0	76.1	117.4
20	127.0	77.1	91.2
25	145.1	78.6	118.3
30	154.4	81.1	123.1
Trung bình	132.0	76.8	107.8

Bảng 3.6. Tải định tuyến theo Tốc độ di chuyển



Hình 3.4. Biểu đồ Tải định tuyến theo Tốc độ di chuyển

Dữ liệu và hình ảnh trong Bảng 3.6 và Hình 3.4 cho thấy khi tốc độ di chuyển tăng lên, tải định tuyến của cả 3 giao thức đều tăng. Điều này là do tần suất phá vỡ liên kết và tuyến đường trong mạng tăng lên khi tốc độ di chuyển của các nút mạng tăng. Tuy nhiên tải định tuyến của giao thức DSDV tăng lên không đáng kể và thấp hơn so với tải định tuyến của hai giao thức AODV và DSR. Tải định tuyến của giao thức AODV là cao nhất trong ba giao thức.

3.5.2. Ảnh hưởng của thời gian tạm dừng

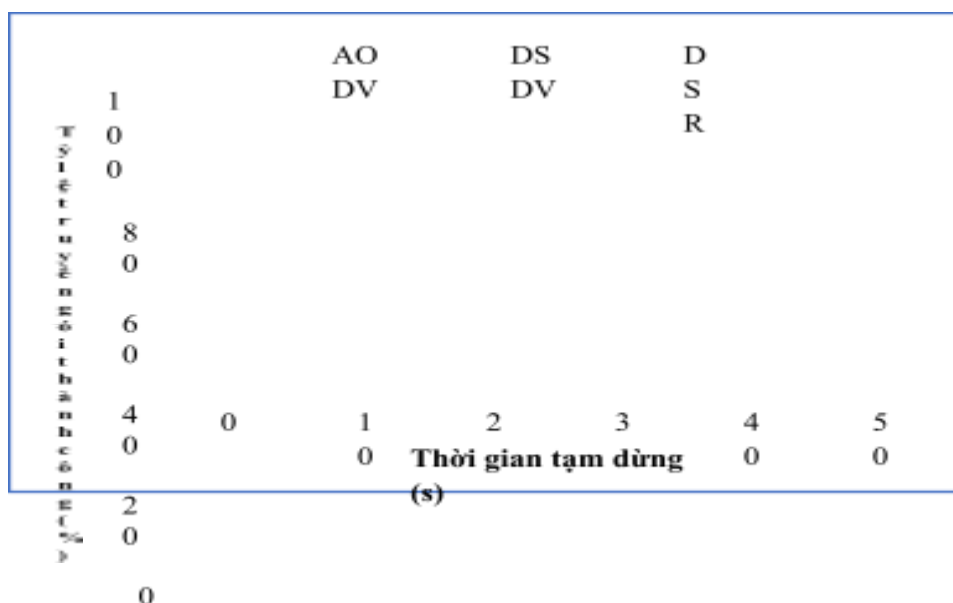
Để kiểm nghiệm tác động của thời gian tạm dừng di chuyển của nút mạng đến hiệu năng mạng sử dụng các giao thức định tuyến AODV, DSDV và DSR, mô phỏng đã được thực hiện với các thông số chung được thiết lập trong Bảng 4.1, số lượng nút là 40, tốc độ di chuyển của các nút mạng là 30 (m/s), thời gian tạm dừng giữa 2 lần di chuyển được thay đổi từ 0, 10, 20, 30, 40 đến 50 giây, số lượng kết nối truyền dữ liệu giữa một cặp nút nguồn-đích là 10. Kết quả của các tham số hiệu năng được đưa ra trong các Bảng 3.7 – Bảng 3.10 được tính trung bình từ kết quả của 10 lần chạy mô phỏng với topo mạng và lưu lượng khác nhau được sinh ngẫu nhiên.

◆ Tỷ lệ truyền gói tin dữ liệu thành công

Tỷ lệ truyền gói tin dữ liệu thành công khi thay đổi thời gian tạm dừng giữa hai lần di chuyển của các nút mạng được đưa ra trong Bảng 3.7 và mối tương quan về sự thay đổi hiệu năng về tỉ lệ truyền gói thành công theo thời gian tạm dừng của các giao thức được biểu diễn trong Hình 3.5.

Thời gian tạm dừng (s)	Tỷ lệ truyền gói thành công (%)		
	AODV	DSDV	DSR
0	85.8	70.4	66
10	85.6	71.2	63.4
20	90.4	74.8	71.28
30	85	70.6	60.4
40	82.8	70.6	58.6
50	81.2	70.4	62.4
Trung bình	85.1	71.3	63.7

Bảng 3.7. Tỷ lệ truyền gói thành công theo Thời gian tạm dừng



Hình 3.5. Biểu đồ Tỷ lệ truyền gói thành công theo Thời gian tạm dừng

Dữ liệu trong Bảng 3.7 và hình ảnh trong Hình 3.5 cho thấy thời gian tạm dừng giữa hai lần di chuyển của nút mạng không có ảnh hưởng nhiều đến tỷ lệ truyền gói thành công của các giao thức. Đối với các thông số mô phỏng được thiết lập, tỷ lệ truyền gói thành công có xu hướng đạt được giá trị cao nhất ở thời

gian tạm dừng 20 giây. Khi thay đổi thời gian tạm dừng, tỉ lệ truyền gói thành công của giao thức AODV vẫn là cao nhất và của giao thức DSR vẫn là thấp nhất trong số 3 giao thức được kiểm nghiệm.

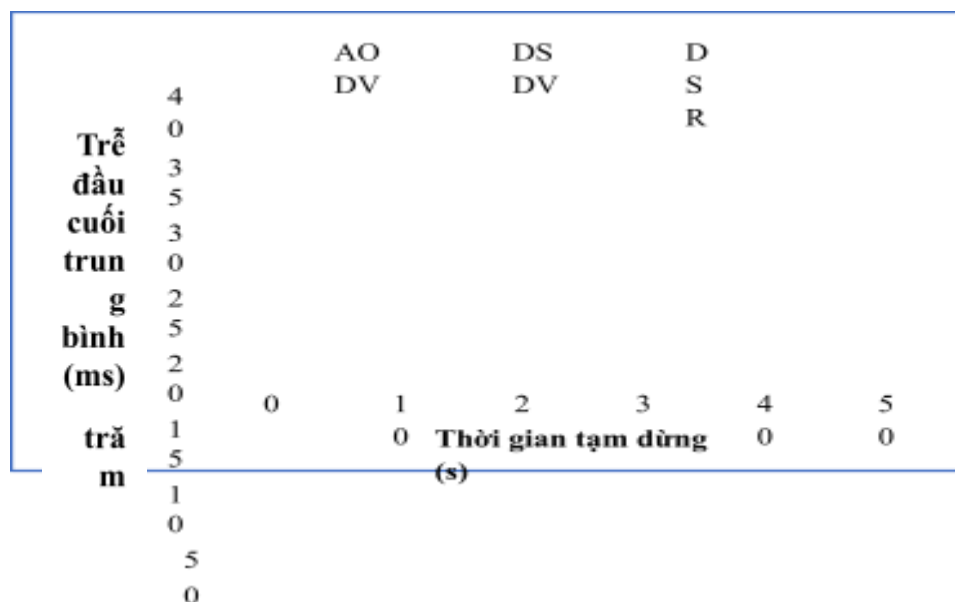
❖ Trễ đầu cuối trung bình

Số liệu trễ đầu cuối trung bình khi truyền các gói tin dữ liệu của 3 giao thức được đưa ra trong Bảng 3.8 và mối tương quan của sự thay đổi được minh họa trong Hình 3.6.

Thời gian tạm dừng (s)	Trễ truyền gói tin trung bình (ms)		
	AODV	DSDV	DSR
0	257.2	343.2	3567.2
10	237.2	342.2	3592.2
20	168.6	251.2	2606.84
30	260	361	3717.2
40	342.4	481.8	3786.2
50	422.6	505.2	3640.2
Trung bình	281.3	380.8	3485.0

Bảng 3.8. Trễ đầu cuối trung bình theo Thời gian tạm dừng

Dữ liệu và hình ảnh trong Bảng 3.8 và Hình 3.6 cho thấy đối với giao thức AODV và DSDV, thời gian tạm dừng không làm ảnh hưởng nhiều tới trễ truyền gói tin trung bình. Đối với giao thức DSR, thời gian trễ trung bình khi truyền gói tin dữ liệu có xu hướng đạt giá trị nhỏ nhất ở thời gian tạm dừng là 20 giây. Trong số 3 giao thức, thời gian trễ đầu cuối trung bình của giao thức AODV là nhỏ nhất và của giao thức DSR là lớn nhất.



Hình 3.6. Biểu đồ Trễ đầu cuối trung bình theo Tốc độ di chuyển

◆ Thông lượng

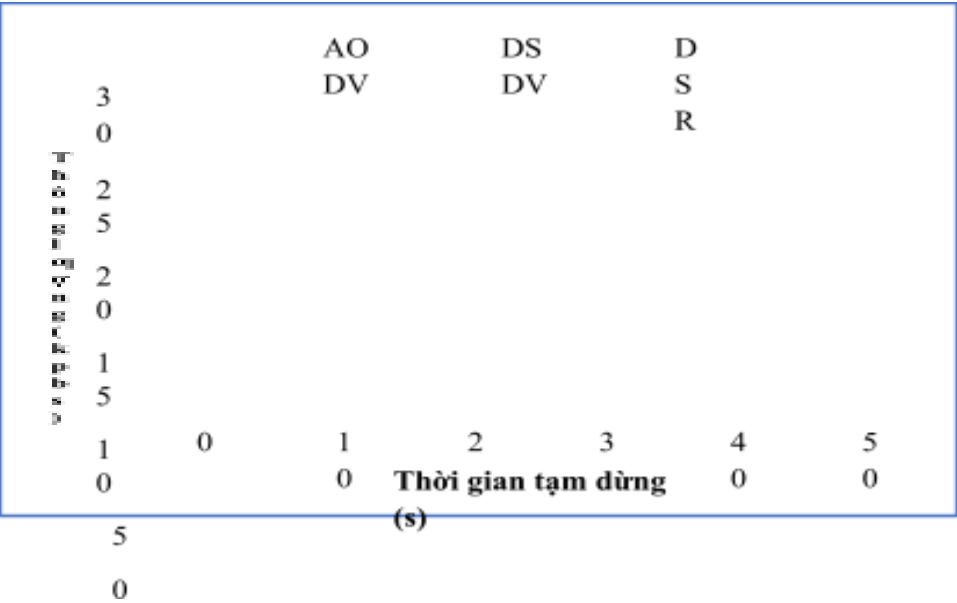
Số liệu về thông lượng của 3 giao thức được đưa ra trong Bảng 3.9 và mối tương quan của sự thay đổi thông lượng được minh họa trong Hình 3.7.

Thời gian tạm dừng (s)	Thông lượng (kbps)		
	AODV	DS DV	DSR
0	25.4	20.6	18.6
10	25.2	20.8	17.8
20	26.4	22	19.96
30	25	20.6	17
40	24.4	20.6	16.6
50	23.8	20.6	17.6
Trung bình	25.0	20.9	17.9

Bảng 3.9. Thông lượng theo Thời gian tạm dừng

Dữ liệu và hình ảnh trong Bảng 3.9 và Hình 3.7 cho thấy khi thời gian tạm dừng thay đổi, mức độ thay đổi về thông lượng của ba giao thức là không đáng

kể. Tuy nhiên, thông lượng của cả 3 giao thức có xu hướng đạt được ở mức cao nhất khi thời gian tạm dừng được thiết lập ở mức 20 giây. Thông lượng của giao thức AODV là cao nhất và của giao thức DSR là thấp nhất khi thay đổi thời gian tạm dừng.

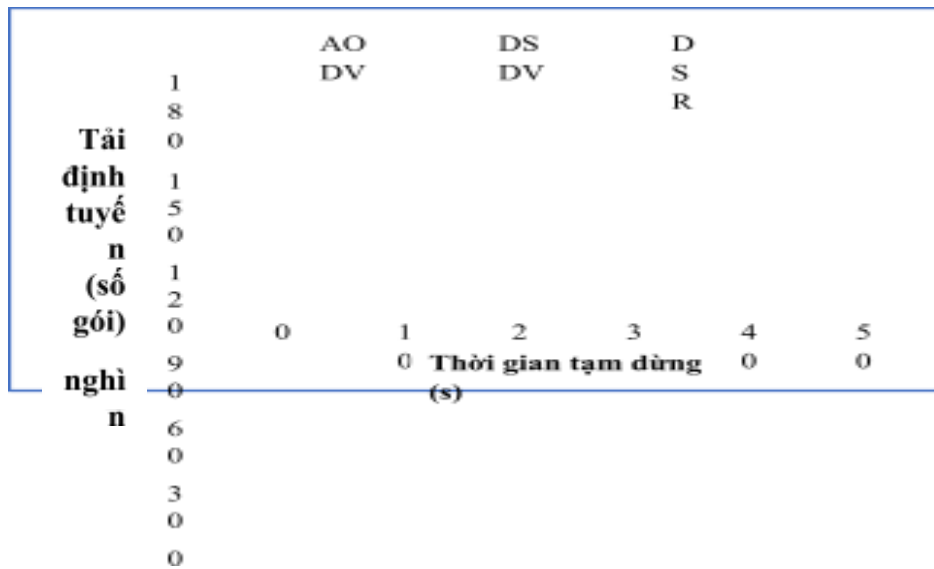


Hình 3.7. Biểu đồ Thông lượng theo Thời gian tạm dừng

◆ **Tải định tuyến**

Thời gian tạm dừng (s)	Tải định tuyến (nghìn gói)		
	AODV	DSDV	DSR
0	149.4	79.3	131.4
10	149.2	80.2	137.2
20	128.4	78.7	103.3
30	151.3	79	114.4
40	154.1	80.5	134.4
50	158.9	80.3	120.2
Trung bình	148.6	79.7	123.5

Bảng 3.10. Tải định tuyến theo Thời gian tạm dừng



Hình 3.8. Biểu đồ Tải định tuyến theo Thời gian tạm dừng

Dữ liệu và hình ảnh trong Bảng 3.10 và Hình 3.8 cho thấy thời gian tạm dừng không tác động nhiều tới tải định tuyến của giao thức DSDV. Đối với giao thức AODV và DSR, tải định tuyến có xu hướng đạt giá trị thấp nhất ở thời gian tạm dừng 20 giây. Tải định tuyến của giao thức AODV là cao nhất và của giao thức DSDV là thấp nhất trong ba giao thức.

3.6. Mô phỏng và đánh giá ảnh hưởng của tải dữ liệu

3.6.1. Ảnh hưởng của số lượng kết nối

Để kiểm nghiệm tác động của số lượng kết nối truyền dữ liệu đến hiệu năng mạng sử dụng các giao thức định tuyến AODV, DSDV và DSR, mô phỏng đã được thực hiện với các thông số chung được thiết lập trong Bảng 3.1, số lượng nút là 40, tốc độ di chuyển của các nút mạng được thiết lập là 5 (m/s), thời gian tạm dừng giữa 2 lần di chuyển là 10 giây, số lượng kết nối truyền dữ liệu giữa một cặp nút nguồn-đích được thay đổi từ 6, 8, 10, 12, 14, 16 đến 18 kết nối. Kết quả của các tham số hiệu năng được đưa ra trong các Bảng 3.11 – Bảng 3.14

được tính trung bình từ kết quả của 10 lần chạy mô phỏng với topo mạng và lưu lượng khác nhau được sinh ngẫu nhiên.

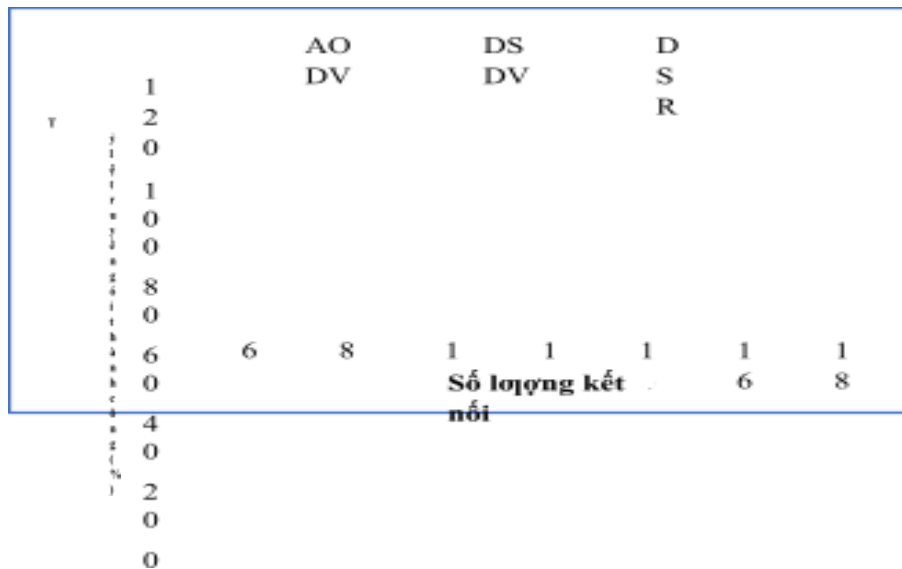
❖ Tỷ lệ truyền gói tin dữ liệu thành công

Số liệu và biểu đồ tương quan về tỷ lệ truyền gói tin thành công khi số lượng kết nối truyền dữ liệu thay đổi được đưa ra trong Bảng 3.11 và mối tương quan về tỉ lệ truyền gói thành công giữa các giao thức được biểu diễn trong biểu đồ trong Hình 3.9.

Dữ liệu và hình ảnh trong Bảng 3.11 và Hình 3.9 cho thấy khi số lượng kết nối truyền dữ liệu tăng lên đồng nghĩa với tải dữ liệu tăng thì tỉ lệ truyền gói thành công của cả 3 giao thức đều giảm. Với tải dữ liệu thấp (nhỏ hơn 12 kết nối), tỉ lệ truyền gói thành công của giao thức AODV là cao nhất và của giao thức DSDV là thấp nhất. Tuy nhiên, khi số lượng kết nối tăng lên (lớn hơn 12 kết nối), tỉ lệ truyền thành công của giao thức DSDV lại là cao nhất và của giao thức DSR là nhỏ nhất.

Số lượng kết nối	Tỷ lệ truyền gói thành công (%)		
	AODV	DSDV	DSR
6	98.6	87.2	97
8	96.2	87.4	93.8
10	89	84	82
12	75	75.8	62.8
14	68	69.2	55.4
16	61	63.2	50.4
18	52.8	56.8	41.4
Trung bình	77.2	74.8	69.0

Bảng 3.11. Tỷ lệ truyền gói thành công theo Số lượng kết nối



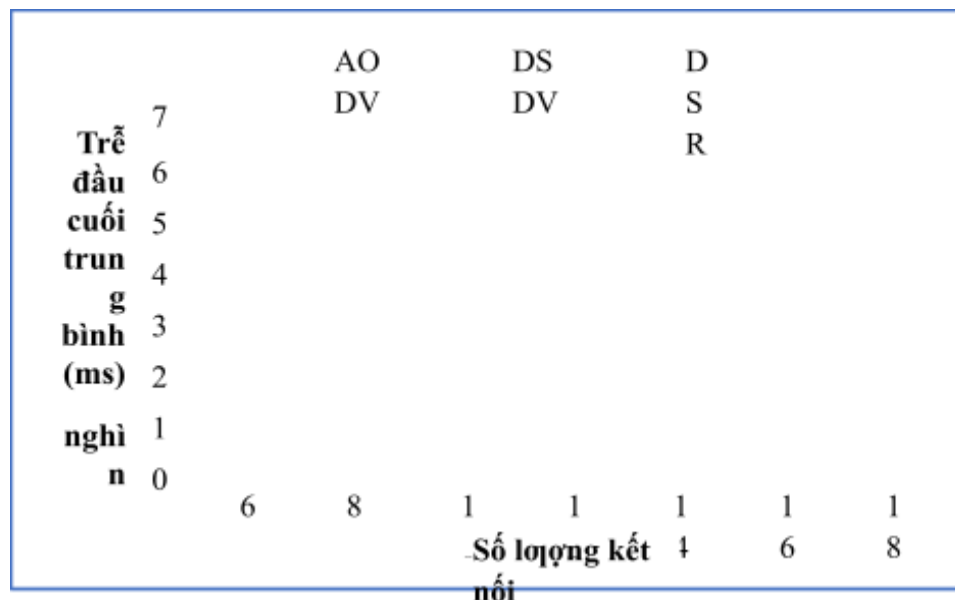
Hình 3.9. Biểu đồ Tỷ lệ truyền gói thành công theo Số lượng kết nối

◆ Trễ đầu cuối trung bình

Số liệu trễ đầu cuối trung bình khi truyền các gói tin dữ liệu của 3 giao thức được đưa ra trong Bảng 3.12 và mỗi tương quan sự thay đổi trễ đầu cuối trung bình của ba giao thức được minh họa bằng biểu đồ trong Hình 3.10.

Số lượng kết nối	Trễ truyền gói tin trung bình (ms)		
	AODV	DSDV	DSR
6	28.2	25.4	311.4
8	61.8	69.6	668.4
10	202	273.6	1933
12	625.4	858.8	3768.6
14	1041.2	1308	4400.8
16	1625	1741.4	5119.4
18	2272.8	2260.6	5987.2
Trung bình	836.6	933.9	3169.8

Bảng 3.12. Trễ đầu cuối trung bình theo Số lượng kết nối



Hình 3.10. Biểu đồ Trễ đầu cuối trung bình theo Số lượng kết nối

Dữ liệu và hình ảnh trong Bảng 3.12 và Hình 3.10 cho thấy khi tải dữ liệu (số lượng kết nối) tăng lên, thời gian trễ trung bình khi truyền một gói tin dữ liệu từ nút nguồn tới nút đích của giao thức DSR tăng rất mạnh. Mức độ tăng của trễ đầu cuối đối với giao thức AODV và giao thức DSDV thấp hơn. Giá trị trễ đầu cuối trung bình của hai giao thức này xấp xỉ nhau và đều nhỏ hơn trễ đầu cuối của giao thức DSR.



❖ Thông lượng

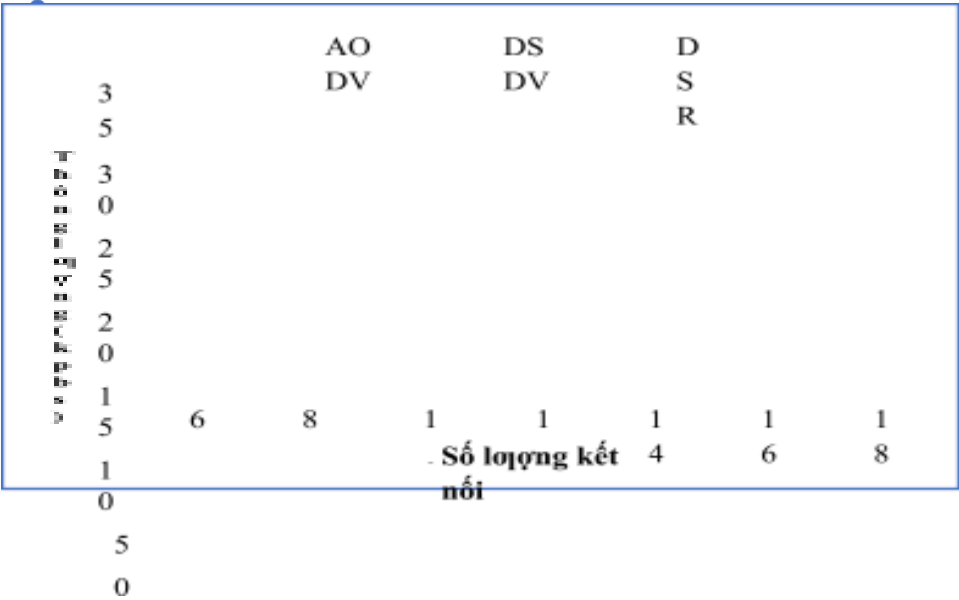
Số liệu về thông lượng của 3 giao thức được đưa ra trong Bảng 3.13 và mối tương quan của sự thay đổi thông lượng được minh họa trong Hình 3.11.

Dữ liệu và hình ảnh trong Bảng 3.13 và Hình 3.11 cho thấy khi số lượng kết nối tăng lên, thông lượng của cả 3 giao thức đều có xu hướng giảm. Với tải dữ liệu thấp và trung bình (nhỏ hơn 12 kết nối), thông lượng của giao thức AODV là lớn nhất và của giao thức DSDV là nhỏ nhất. Tuy nhiên, khi tải dữ liệu

tăng cao, giao thức DSDV lại đạt được thông lượng cao nhất và giao thức DSR có thông lượng thấp nhất trong số 3 giao thức.

Số lượng kết nối	Thông lượng (kbps)		
	AODV	DSDV	DSR
6	29	25.6	27.4
8	28.2	25.6	26.4
10	26.2	24.8	23.2
12	22.2	22.2	17.6
14	19.8	20.2	15.6
16	17.8	18.8	14.4
18	15.2	16.6	11.2
Trung bình	22.6	22.0	19.4



 Bảng 3.13. Thông lượng theo Số lượng kết nối



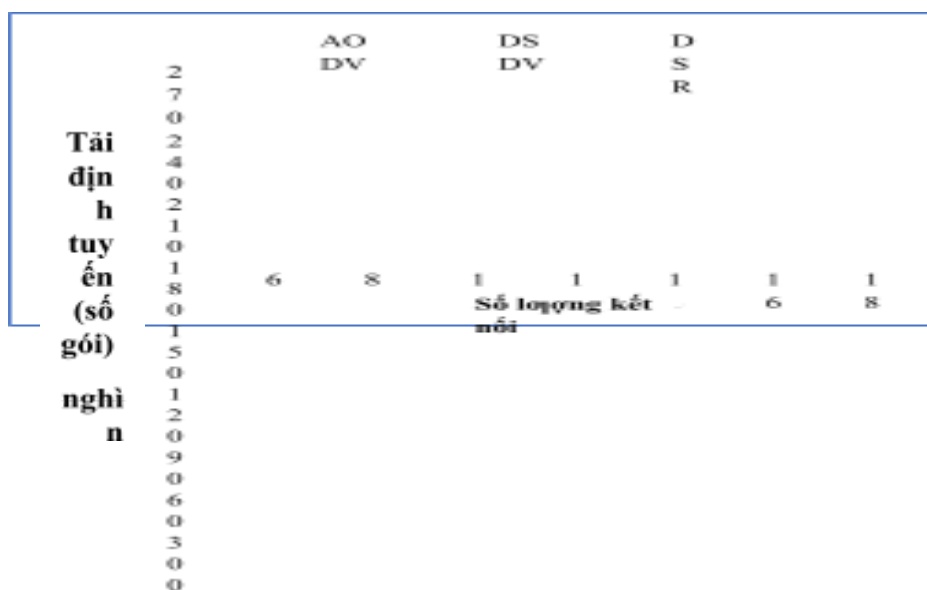
Hình 3.11. Biểu đồ Thông lượng theo Số lượng kết nối

❖ Tải định tuyến

Số liệu về tải định tuyến (số lượng gói tin điều khiển) của 3 giao thức được đưa ra trong Bảng 3.14 và mối tương quan của sự thay đổi tải định tuyến được minh họa trong Hình 3.12.

Số lượng kết nối	Tải định tuyến (nghìn gói)		
	AODV	DSDV	DSR
6	47.6	47.3	46
8	76.8	60.7	68.9
10	128.9	74.2	107.7
12	183.5	88.6	128.2
14	206.3	101.3	168.5
16	219.2	113.9	167.8
18	245.4	127.3	211.1
Trung bình	158.2	87.6	128.3

— Bảng 3.14. — Tải định tuyến theo Số lượng kết nối



Hình 3.12. Biểu đồ Tải định tuyến theo Số lượng kết nối

Dữ liệu và đồ thị trong Bảng 3.14 và Hình 3.12 cho thấy khi số lượng luồng dữ liệu tăng lên, tải định tuyến của cả 2 giao thức AODV và DSR đều tăng. Tuy nhiên tải định tuyến của giao thức DSDV biến đổi không đáng kể và thấp hơn so với tải định tuyến của hai giao thức AODV và DSR. Tải định tuyến của giao thức AODV là cao nhất trong ba giao thức.

3.6.2. Tác động của số lượng nút

Để kiểm nghiệm tác động của số lượng nút mạng đến hiệu năng mạng sử dụng các giao thức định tuyến AODV, DSDV và DSR, mô phỏng đã được thực hiện với các thông số chung được thiết lập trong Bảng 4.1. Số lượng nút được thiết lập giá trị thay đổi từ 20 đến 30, 40, 50, 60 và 70 nút. Tốc độ di chuyển của các nút mạng là 30 (m/s). Thời gian tạm dừng giữa 2 lần di chuyển là 10 giây. Số lượng kết nối truyền dữ liệu giữa một cặp nút nguồn-đích là 10. Kết quả của các tham số hiệu năng được đưa ra trong các Bảng 3.15 – Bảng 3.18 được tính trung bình từ kết quả của 10 lần chạy mô phỏng với topo mạng và lưu lượng khác nhau được sinh ngẫu nhiên.

◆ Tỷ lệ truyền gói tin dữ liệu thành công

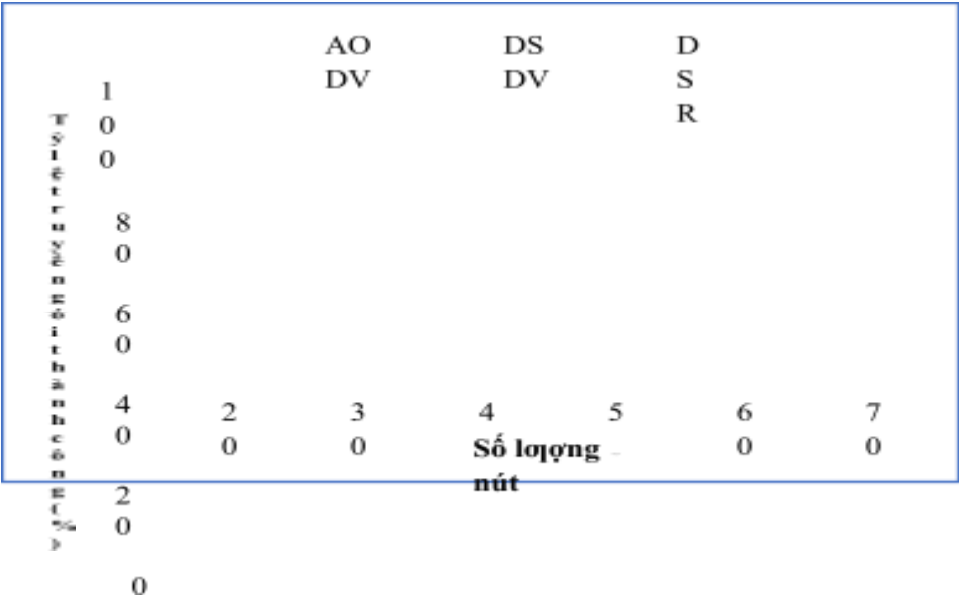
Dữ liệu và biểu đồ tương quan về tỷ lệ truyền gói tin dữ liệu thành công khi thay đổi số lượng nút mạng được đưa ra trong Bảng 3.15 và Hình 3.13.

Dữ liệu và hình ảnh trong Bảng 3.15 và Hình 3.13 cho thấy khi số lượng nút mạng tăng lên, tỉ lệ truyền gói thành công của giao thức AODV không thay đổi nhiều, tỉ lệ truyền gói thành công của giao thức DSDV có xu hướng tăng khi số lượng nút mạng tăng từ 20 đến 50 và có xu hướng giảm khi số lượng nút là từ 60 nút trở lên, tỉ lệ truyền gói thành công của giao thức DSR giảm khi số lượng nút mạng tăng lên. Hiệu năng về tỉ lệ truyền gói thành công của giao thức AODV

đạt được cao nhất trong số 3 giao thức. Khi số lượng nút mạng tăng lớn hơn 40 nút, tỉ lệ truyền gói thành công của giao thức DSR là thấp nhất.

Số lượng nút	Tỷ lệ truyền gói thành công (%)		
	AODV	DSDV	DSR
20	85.8	56	75.4
30	84.8	60.4	67
40	84.2	68.8	65.6
50	81.4	74	52.2
60	85.8	72.8	46.2
70	81.6	63.8	27.6
Trung bình	83.9	66.0	55.7

Bảng 3.15. Tỷ lệ truyền gói thành công theo Số lượng nút mạng



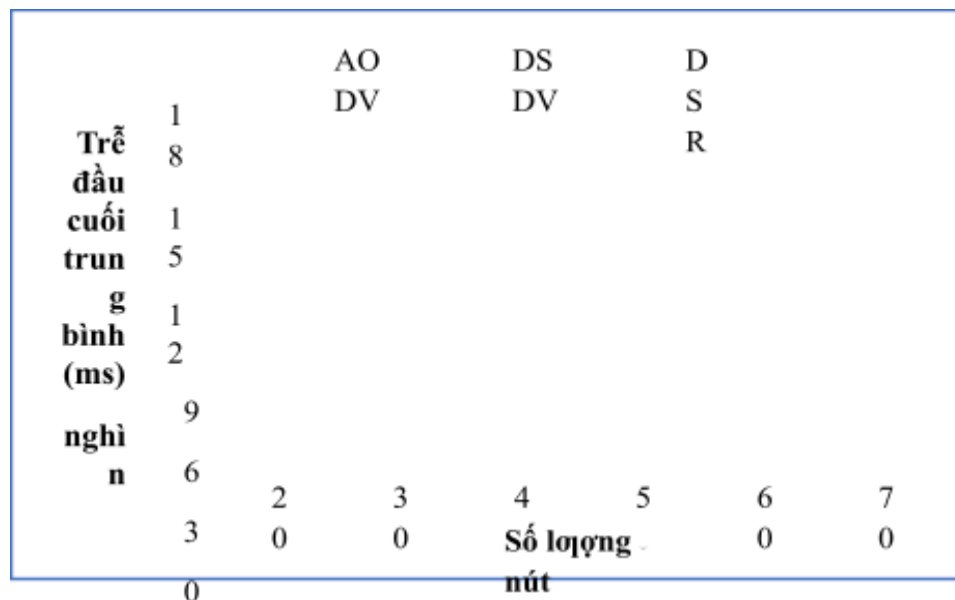
Hình 3.13. Biểu đồ Tỷ lệ truyền gói thành công theo Số lượng nút mạng

❖ Trễ đầu cuối trung bình

Số liệu trễ đầu cuối trung bình khi truyền các gói tin dữ liệu của 3 giao thức được đưa ra trong Bảng 3.16 và mối tương quan của sự thay đổi tham số hiệu năng này được minh họa trong Hình 3.14.

Số lượng nút	Trễ truyền gói tin trung bình (ms)		
	AODV	DSDV	DSR
20	277	361.2	1704.2
30	285.6	407.8	2794.8
40	312	464.4	3199.6
50	352.6	724.4	6085.8
60	288	981.4	8190.8
70	419.8	1698	16414.6
Trung bình	322.5	772.9	6398.3

Bảng 3.16. Trễ đầu cuối trung bình theo Số lượng nút



Hình 3.14. Biểu đồ Trễ đầu cuối trung bình theo Số lượng nút mạng

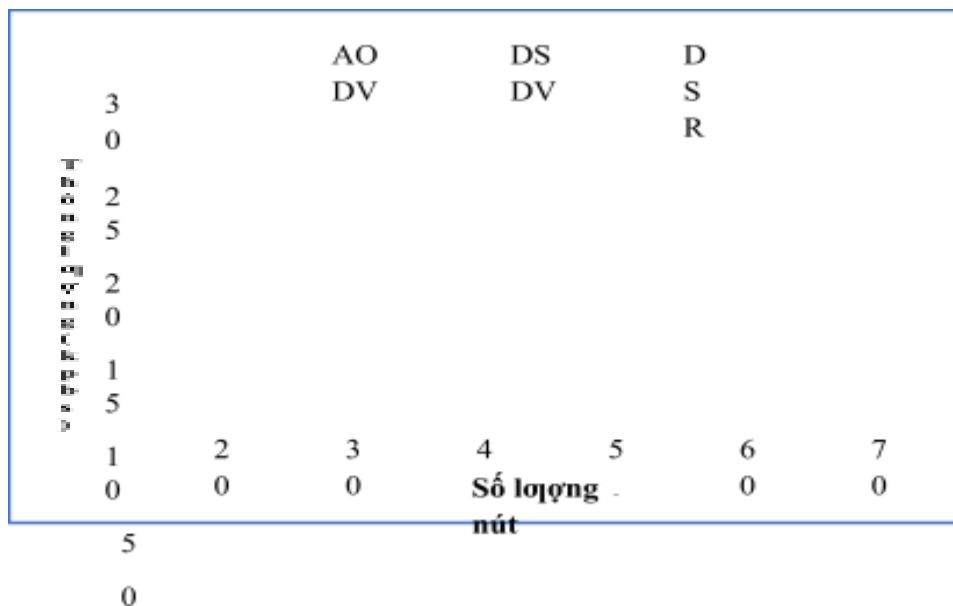
Dữ liệu trong Bảng 3.16 và biểu đồ trong Hình 3.14 cho thấy khi số lượng nút mạng tăng lên, thời gian trễ trung bình khi truyền một gói tin dữ liệu từ nút nguồn tới nút đích của giao thức DSR tăng nhanh. Đối với giao thức AODV và giao thức DSDV, thời gian trễ này không bị tác động nhiều. Khi số lượng nút mạng tiếp tục tăng ở ngưỡng lớn hơn 60 nút, trễ đầu cuối trung bình của giao thức DSDV có xu hướng tăng. Giao thức AODV có trễ đầu cuối trung bình nhỏ nhất trong số 3 giao thức.

❖ Thông lượng

Dữ liệu và hình ảnh trong Bảng 3.17 và Hình 3.15 cho thấy khi số lượng nút mạng tăng lên, thông lượng của giao thức AODV không thay đổi nhiều; thông lượng của giao thức DSDV có xu hướng tăng khi số lượng nút mạng tăng từ 20 đến 50 và có xu hướng giảm khi số lượng nút là từ 60 nút trở lên; thông lượng của giao thức DSR giảm khi số lượng nút mạng tăng lên. Thông lượng của giao thức AODV đạt được cao nhất trong số 3 giao thức. Khi số lượng nút mạng tăng lớn hơn 30 nút, thông lượng của giao thức DSR là thấp nhất.

Số lượng nút	Thông lượng (kbps)		
	AODV	DSDV	DSR
20	25	16.2	21.4
30	25	17.6	18.8
40	24.8	20.4	18.4
50	23.8	21.8	14.6
60	25.4	21.4	13
70	24	18.6	7.4
Trung bình	24.7	19.3	15.6

Bảng 3.17. Thông lượng theo Số lượng nút mạng



Hình 3.15. Biểu đồ Thông lượng theo Số lượng nút mạng

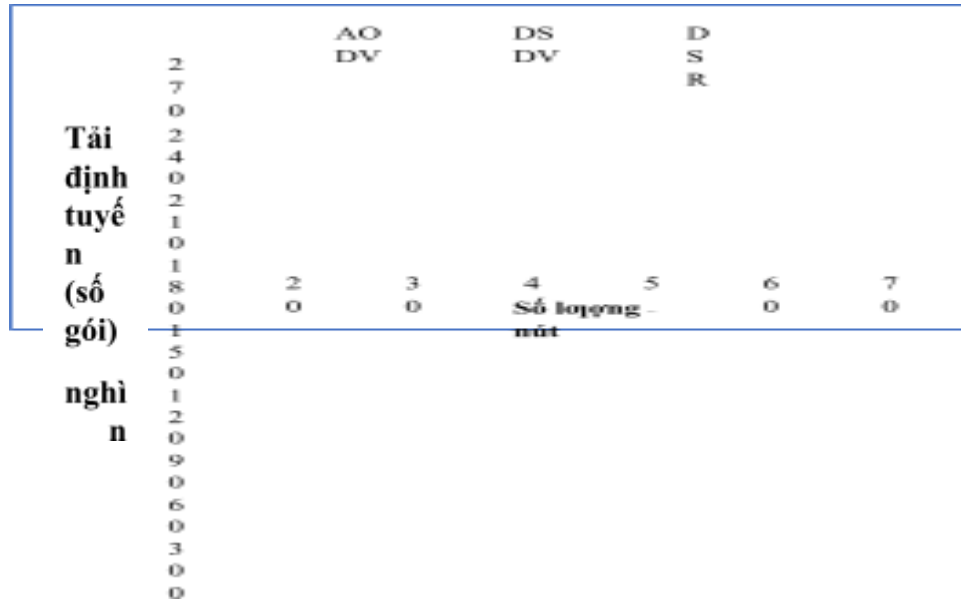
❖ Tải định tuyến

Số lượng nút	Tải định tuyến (nghìn gói)		
	AODV	DSDV	DSR
20	94.4	66.4	94
30	128.1	71.7	93.4
40	145.6	80.9	114.3
50	189.9	100.3	158.9
60	176.5	111.3	182
70	213.2	122.3	255.5
Trung bình	158.0	92.2	149.7

Bảng 3.18. Tải định tuyến theo Số lượng nút mạng

Dữ liệu và hình ảnh trong Bảng 3.18 và Hình 3.16 cho thấy khi số lượng nút mạng tăng lên, tải định tuyến của cả 3 giao thức đều tăng. Tuy nhiên mức độ tăng tải định tuyến của giao thức DSDV là ít nhất và thấp hơn so với tải định tuyến của hai giao thức AODV và DSR. Tải định tuyến của giao thức AODV là

cao nhất khi số lượng nút mạng nhỏ hơn 50 nút. Tải định tuyến của giao thức DSR là cao nhất khi số lượng nút mạng lớn hơn 60 nút.



Hình 3.16. Biểu đồ Tải định tuyến theo Số lượng nút mạng

3.7. Tổng kết Chương 3

Mô phỏng là “tiến trình thiết kế một mô hình của hệ thống thực và áp dụng các thí nghiệm với mô hình này nhằm mục đích hiểu được hành vi của hệ thống và/hoặc đánh giá các chiến lược cho hoạt động của hệ thống”. Một quá trình mô phỏng có thể được xem như là một dòng các tiến trình của các thực thể trong mạng.

NS2 là một công cụ mô phỏng hướng sự kiện được xây dựng để sử dụng trong nghiên cứu về lĩnh vực mạng và truyền thông. NS2 có thể mô phỏng các chức năng và các giao thức mạng, đồng thời cung cấp cho người dùng cách thức đặc tả các giao thức mạng và mô phỏng hoạt động của chúng.

Trong kiến trúc của NS2, có hai ngôn ngữ chính trong NS2 là C++ và OTcl. Trong khi C++ tạo ra các mô tả bên trong cho các đối tượng mô phỏng thì OTcl

thiết lập mô phỏng bằng cách liên kết và cấu hình các đối tượng cũng như lập lịch cho các sự kiện rời rạc. C++ và OTcl được liên kết với nhau bằng TclCL.

Việc cấu hình mạng ad hoc trong NS2 trở nên khá đơn giản khi NS2 đã được phát triển và tích hợp các thành phần mô phỏng mạng. Để cấu hình một mạng ad hoc, chỉ cần cấu hình các tham số cho mỗi nút mạng khi tạo ra.

Để đánh giá mức độ ảnh hưởng của tốc độ di chuyển của các nút mạng và tải dữ liệu đối với hiệu năng các giao thức định tuyến AODV, DSDV và DSR, có 4 kịch bản mô phỏng đã được thử nghiệm bao gồm: thay đổi tốc độ di chuyển của nút mạng; thay đổi tốc độ tạm dừng giữa các lần di chuyển của nút mạng; thay đổi số lượng luồng truyền dữ liệu và thay đổi số lượng nút mạng. Có 4 tham số hiệu năng được đánh giá trong các kịch bản bao gồm: Tỷ lệ truyền gói tin thành công, độ trễ đầu cuối trung bình, thông lượng và tải định tuyến.

Kết quả thử nghiệm mô phỏng cho thấy:

- Khi tốc độ di chuyển tăng lên: tỷ lệ truyền gói thành công của cả 3 giao thức đều có xu hướng giảm; thời gian trễ truyền gói trung bình giao thức DSR tăng nhanh; giao thức AODV và DSDV ít bị ảnh hưởng bởi tốc độ di chuyển của nút mạng; thông lượng của cả 3 giao thức đều có xu hướng giảm; tải định tuyến của cả 3 giao thức đều tăng.

- Thời gian tạm dừng giữa hai lần di chuyển của nút mạng không có ảnh hưởng nhiều đến tỷ lệ truyền gói thành công, trễ đầu cuối trung bình, thông lượng và tải định tuyến của 3 giao thức định tuyến.

- Khi số lượng kết nối truyền dữ liệu tăng lên: tỷ lệ truyền gói thành công của cả 3 giao thức đều giảm; thời gian trễ trung bình của cả 3 giao thức đều tăng nhưng mạnh nhất đối với giao thức DSR; thông lượng của cả 3 giao thức đều có

xu hướng giảm. Với tải dữ liệu thấp và trung bình, thông lượng của giao thức AODV là lớn nhất và của giao thức DSDV là nhỏ nhất. Khi tải dữ liệu tăng cao, giao thức DSDV lại đạt được thông lượng cao nhất và giao thức DSR có thông lượng thấp nhất; tải định tuyến của cả 2 giao thức AODV và DSR đều tăng. Tuy nhiên tải định tuyến của giao thức DSDV biến đổi không đáng kể. Tải định tuyến của giao thức AODV là cao nhất trong ba giao thức.

- Khi số lượng nút mạng tăng lên: tỉ lệ truyền gói thành công của giao thức AODV không thay đổi nhiều; thời gian trễ trung bình của giao thức DSR tăng nhanh nhưng ít biến đổi đối với giao thức AODV và DSDV; thông lượng của giao thức AODV không thay đổi nhiều trong khi thông lượng của giao thức DSR giảm. Thông lượng của giao thức DSDV có xu hướng tăng khi số lượng nút mạng tăng đến 50 nút và có xu hướng giảm khi số lượng nút là từ 60 trở lên; tải định tuyến của cả 3 giao thức đều tăng. Mức độ tăng tải định tuyến của giao thức DSDV là ít nhất và thấp hơn so với tải định tuyến của hai giao thức AODV và DSR.

KẾT LUẬN

Với những ưu điểm về độ linh động cao và không yêu cầu hạ tầng mạng có sẵn, mạng ad hoc có tiềm năng ứng dụng vào rất nhiều lĩnh vực phục vụ cho cuộc sống của con người như trong các mạng truyền thông trong quân sự, cảnh báo thiên tai hiểm họa, giao thông, thương mại, giải trí, giáo dục, ...

Do các nút mạng có khả năng di chuyển tự do và ngẫu nhiên nên topo mạng ad hoc thường xuyên thay đổi, các liên kết và đường truyền dữ liệu trong mạng thường xuyên bị phá vỡ và hình thành mới. Vì vậy các giao thức định tuyến trong mạng ad hoc cần phải được thiết kế phù hợp với yêu cầu này. Có 3 giao thức định tuyến tiêu biểu cho các chiến lược định tuyến trong mạng ad hoc là AODV, DSDV và DSR.

Hiệu năng mạng là những tiêu chuẩn chính được sử dụng trong thiết kế và khai thác hệ thống mạng. Việc đánh giá hiệu năng mạng nhằm mục đích so sánh giữa các thiết kế để tìm ra thiết kế tốt nhất. Phương pháp đánh giá bằng mô phỏng đã được sử dụng trong luận văn này để đánh giá sự tác động của tốc độ di chuyển và tải dữ liệu đối với hiệu năng định tuyến trong mạng ad hoc trên cơ sở sử dụng bộ mô phỏng NS2.

Các kết quả chính đã đạt được trong luận văn bao gồm:

- Nghiên cứu tổng quan về mạng ad hoc, các chuẩn và giao thức tầng MAC dành cho mạng ad hoc, các chiến lược định tuyến dành cho mạng ad hoc và vấn đề đánh giá hiệu năng mạng ad hoc.
- Tìm hiểu và trình bày về vấn đề mô phỏng mạng máy tính và bộ công cụ mô phỏng NS-2.

- Nghiên cứu về cơ chế hoạt động và thuật toán định tuyến của giao thức định tuyến AODV, DSDV và DSR trong mạng mạng ad hoc.
- Đánh giá tác động của sự di chuyển của các nút mạng và tải dữ liệu đối với hiệu năng của các giao thức định tuyến với 4 tham số hiệu năng gồm có tỉ lệ truyền gói thành công, trễ đầu cuối trung bình, thông lượng và tải định tuyến trên cơ sở mô phỏng hoạt động của 3 giao thức định tuyến AODV, DSDV và DSR theo 4 kịch bản mô phỏng khác nhau.

Tuy nhiên, kết quả của đánh giá hiện mới đang được giới hạn ở mô hình di động ngẫu nhiên cho các nút mạng sử dụng trong mô phỏng. Để đánh giá được một cách toàn diện và chính xác hơn về hiệu năng định tuyến của các giao thức trong mạng ad hoc, cần triển khai mô phỏng theo các mô hình phân bố và di chuyển khác cho các nút mạng cũng như sử dụng một phạm vi rộng hơn các thông số biến đổi của các kịch bản mô phỏng. Ngoài ra, cần mở rộng phạm vi đánh giá cho các giao thức khác và các giao thức đã được cải tiến từ các giao thức này. Đây cũng chính là các hướng phát triển tiếp theo của đề tài này.

TÀI LIỆU THAM KHẢO

Tiếng Anh

- [1]. Corson S., Macker J. (1999), “Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations”, RFC 2501, Available at: <https://tools.ietf.org/html/rfc2501>.
- [2]. Johnson D., Hu Y., Maltz D. (2007), “The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4”, RFC 4728, Available at: <https://www.ietf.org/rfc/rfc4728.txt>.
- [3]. Kadhun M. (2010), “Network Performance and NS2”, Available at: http://www.internetworks.my/netapps2010/documents/ns2_tutorial.pdf.
- [4]. Perkins C. (2001), Ad Hoc Networking, Addison-Wesley, USA.
- [5]. Perkins C., Belding-Royer E., Das S. (2003), “Ad hoc On-Demand Distance Vector (AODV) Routing”, RFC 3561, Available at: <https://www.ietf.org/rfc/rfc3561.txt>.
- [6]. Perkins C., Bhagwat P. (1994), “Highly Dynamic Destination-Sequenced Distance-Vector (DSDV) Routing for mobile computers”, Proceeding of ACM SIGCOMM Symposium on Communications Conference, pp. 234-44.
- [7]. Teerawat I., Ekram H. (2012), Introduction to Network Simulator NS2, second edition, Springer, USA.

THÔNG TIN HỎI ĐÁP:

*Bạn còn nhiều thắc mắc hoặc muốn tìm kiếm thêm nhiều tài liệu luận văn mới mẻ khác của Trung tâm [Best4Team](#) ,
Liên hệ [dịch vụ viết thuê luận văn](#)
Hoặc qua SĐT Zalo: 091.552.1220 hoặc email: best4team.com@gmail.com để hỗ trợ ngay nhé!*