ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

HOÀNG HỒNG SƠN

NGHIÊN CỨU VÀ ĐÁNH GIÁ HIỆU SUẤT CÁC GIAO THỨC ĐỊNH TUYẾN TRONG MẠNG MANET

LUẬN VĂN THẠC SĨ CÔNG NGHỆ THÔNG TIN

Hà Nội - Năm 2016

ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

HOÀNG HỒNG SƠN

NGHIÊN CỨU VÀ ĐÁNH GIÁ HIỆU SUẤT CÁC GIAO THỨC ĐỊNH TUYẾN TRONG MẠNG MANET

NGÀNH: CÔNG NGHỆ THÔNG TIN

CHUYÊN NGÀNH: TRUYỀN DỮ LIỆU VÀ MẠNG MÁY TÍNH

MÃ SỐ:

LUẬN VĂN THẠC SĨ CÔNG NGHỆ THÔNG TIN

NGƯỜI HƯỚNG DẪN KHOA HỌC: PGS. TS. NGUYỄN ĐÌNH VIỆT

Hà Nội - Năm 2016

LỜI CAM ĐOAN

Tôi xin cam đoan kết quả đạt được trong luận văn là sản phẩm của riêng cá nhân, không sao chép lại của người khác. Trong toàn bộ nội dung của luận văn, những điều được trình bày hoặc là của cá nhân hoặc là được tổng hợp từ nhiều nguồn tài liệu. Tất cả các tài liệu tham khảo đều có xuất xứ rõ ràng và được trích dẫn hợp pháp.

Tôi xin hoàn toàn chịu trách nhiệm về những lời cam đoan của mình.

Hà Nội, tháng 11 năm 2016

HỌC VIÊN

Hoàng Hồng Sơn

LỜI CẢM ƠN

Trong quá trình làm luận văn tôi đã rất cố gắng tuy nhiên luận văn có thể vẫn còn một số thiếu sót nhất định. Tôi rất mong nhận được sự góp ý của thầy cô giáo để luận văn hoàn thiện hơn. Qua đây, tôi cũng xin chân thành cảm ơn PGS.TS. Nguyễn Đình Việt, thầy đã gợi ý và tận tình chỉ bảo, cung cấp nhiều tài liệu quan trọng liên quan tới quá trình thực hiện luận văn. Tôi cũng xin chân thành cảm ơn các thầy cô giáo trường Đại học công nghệ - Đại học quốc gia Hà Nội đã dạy và giúp đỡ trong suốt quá trình nghiên cứu và học tập tại trường.

Tôi xin trân trọng cảm ơn.

Tác giả: Hoàng Hồng Sơn

MỤC LỤC

DANH MỤC CÁC CHỮ VIẾT TẮT

ACK	Acknowledgement
AODV	Ad hoc On Demand Distance Vector
AP	Access Point
BSS	Basic Service Set
CSMA/CA	Carrier Sense Multiple Access / Collision Avoidance
DARPA	Defense Advanced Research Projects Agency
ESS	Extended Service Set
IBSS	Independent Basic Service Set
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
ISM	Industrial, Scientific and Medical
LAN	Local Area Network
MAC	Media Access Control
MANET	Mobile Adhoc Network
NAM	Network Animator
NS-2	Network Simulation version 2
Prnet	Packet Radio network
REQ_ID	Route Request ID
RREP	Route Reply
RREQ	Route Request
RRER	Route Error
SEQ	Sequence Number

WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
WWAN	Wireless Wide Area Network

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG BIỂU

MỞ ĐẦU

Ngày nay mạng không dây tồn tại trong rất nhiều ứng dụng. Được biết tới với sự tiện lợi khi sử dụng và mang tính thẩm mĩ cao khi không cần dây dẫn, mạng không dây có mặt trong các lĩnh vực như giải trí, giáo dục, phương tiện giao thông và đặc biệt mạng không dây đáp ứng được những yêu cầu khắt khe trong quân sự.

Với việc không cần dây dẫn để truyền tải tín hiệu, mạng không dây sử dụng sóng radio làm môi trường truyền dẫn, các node trong mạng có thể tự do di chuyển. Hơn thế nữa các node mạng có thể vừa đóng vai trò là thiết bị đầu cuối lại vừa có thể là node trung gian truyền tải tín hiệu như router.

Trong khuôn khổ của luận văn này, tác giả tập trung nghiên cứu về mạng tùy biến di động - một mô hình mạng không dây mà các node mạng có đặc tính di chuyển liên tục, năng lượng cho các node là hạn chế và do bản chất truyền tin qua sóng radio nên rất dễ bị tấn công làm sai lệch gói tin hoặc thậm chí phá hỏng toàn bộ cấu hình mạng.

Trong bài toán được đặt ra là sự tấn công của các node độc hại đã bị nhiễm mã độc làm cho giao thức định tuyến của các node này bị thay đổi dẫn tới gói tin khi truyền tới node bị nhiễm mã độc sẽ bị hủy bỏ thay vì chuyển tiếp tới node đích.

Mục tiêu chính của luận văn là nghiên cứu các hình thức tấn công trong mạng MANET đặc biệt là hình thức tấn công kiểu blackhole vào giao thức định tuyến AODV, từ đó tiến hành cài đặt đánh giá hiệu năng theo các tham số như tỉ lệ chuyển gói tin dữ liệu thành công, độ trễ trung bình. Phân tích đánh giá hiệu năng một số giao thức định tuyến mở rộng như idsAODV, phr-aodv. Từ những phân tích đó đưa ra các ý tưởng, giải pháp phòng, đề xuất cải tiến giao thức AODV để chống tấn công blackhole.

Luận văn này được trình bày trong 3 chương như sau:

Chương 1: Tổng quan về mạng không dây, giới thiệu một cách tổng quan về mạng không dây và mạng tùy biến di động, các vấn đề quan trọng phải giải quyết trong mạng tùy biến di động

Chương 2: Tấn công kiểu lỗ đen (blackhole) trong giao thức AODV, phân tích về lỗ hổng bảo mật các hình thức tấn công trong mạng tùy biến di động , phân tích các giao thức được mở rộng từ cách thức cơ bản trong mạng MANET để chống tấn công blackhole, từ đó đưa ra ý tưởng cải tiến giao thức AODV.

Chương 3: Đánh giá bằng mô phỏng các đề xuất chống tấn công lỗ đen trong giao thức AODV. Từ các phân tích ở chương 2, chương 3 mô phỏng lại các ý tưởng và giải thuật cải tiến giao thức AODV nhằm chống lại tấn công blackhole. Đề xuất cải tiến giao thức AODV nhằm nâng cao tỉ lệ chuyển gói tin thành công.

CHƯƠNG 1. MẠNG TÙY BIẾN DI ĐỘNG VÀ VẤN ĐỀ BẢO MẬT

1.1. Mạng không dây

1.1.1. Giới thiệu mạng không dây

Mạng không dây (*wireless network*) là mạng điện thoại hoặc mạng máy tính sử dụng sóng radio làm sóng truyền dẫn.[1]

Mạng không dây có tính linh hoạt cao, hỗ trợ các thiết bị di động nên không bị ràng buộc cố định về phân bố địa lí như trong mạng hữu tuyến (có dây). Trong quá trình hoạt động, hệ thống mạng có thể dễ dàng bổ sung hoặc thay thế các thiết bị tham gia mạng mà không cần cấu hình lại toàn bộ kiến trúc mạng. Mạng không dây cho phép triển khai mà không cần đòi hỏi nhiều về cơ sở hạ tầng, không bắt buộc phải có các thiết bị đóng vai trò trung tâm điều khiển và không phụ thuộc vào điều kiện địa lí.

Bên cạnh những thuận lợi trong quá trình triển khai, mạng không dây cũng bộc lộ một số điểm yếu như cơ chế định tuyến trong mạng không dây khá phức tạp, khả năng gây nhiễu và mất gói tin trong quá trình truyền dữ liệu cao. Hơn thế nữa vấn đề bảo mật thông tin trên mạng không dây đã và đang đặt nhiều thách thức cho việc nghiên cứu và thử nghiệm trên các mô hình mạng nhằm nâng cao hiệu quả và chất lượng của hệ thống mạng. Có thể nói, nghiên cứu và cải tiến mạng không dây là yêu cầu cấp thiết và có tính thực tế cao.

1.1.2. Phân loại mạng không dây

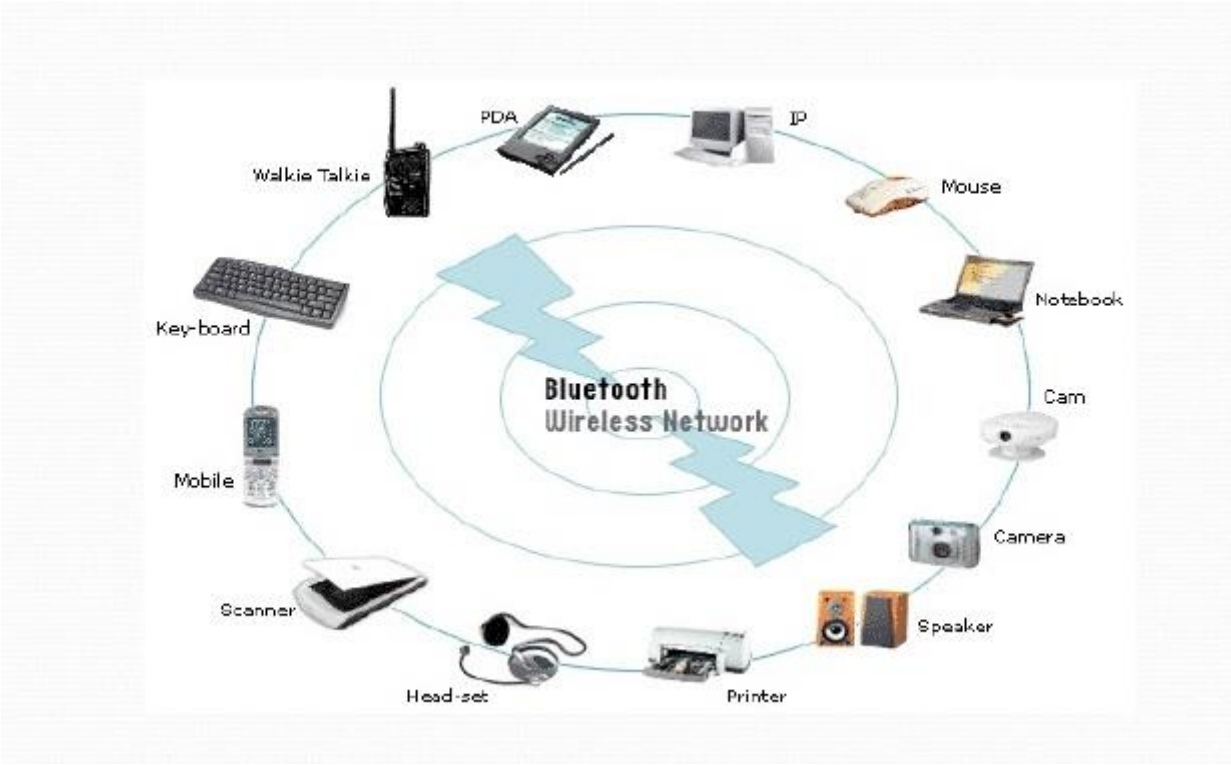
Mạng không dây có thể triển khai trong nhiều dạng khu vực địa lí khác nhau kết hợp với công nghệ hạ tầng cho phù hợp. Phân loại mạng không dây có thể dựa trên 2 tiêu chí đó là:

1. Theo qui mô triển khai mạng.
2. Theo sự di động của các thiết bị di động trong mạng.

1.1.2.1. Phân loại theo qui mô triển khai mạng

- ***WPAN(Wireless Personal Area Network) : Mạng không dây cá nhân***

Là một công nghệ mạng cho phép các thiết bị giao tiếp với nhau bằng sóng radio qua băng tần ISM (In, Scientific and Medical) 2.4 GHz [2]. WPAN còn được gọi là Bluetooth, được SIG chính thức giới thiệu phiên bản 1.0 của Bluetooth vào tháng 7 năm 1999. Các thiết bị có thể kết nối với nhau với khoảng cách tối đa là 10m, hỗ trợ tố đa 8 kết nối đồng thời. Băng thông tối đa là 1Mbps chia sẻ cho các kết nối. Nhược điểm khoảng cách liên lạc nhỏ, băng thông thấp.



Hình 1. 1 Mạng WPAN

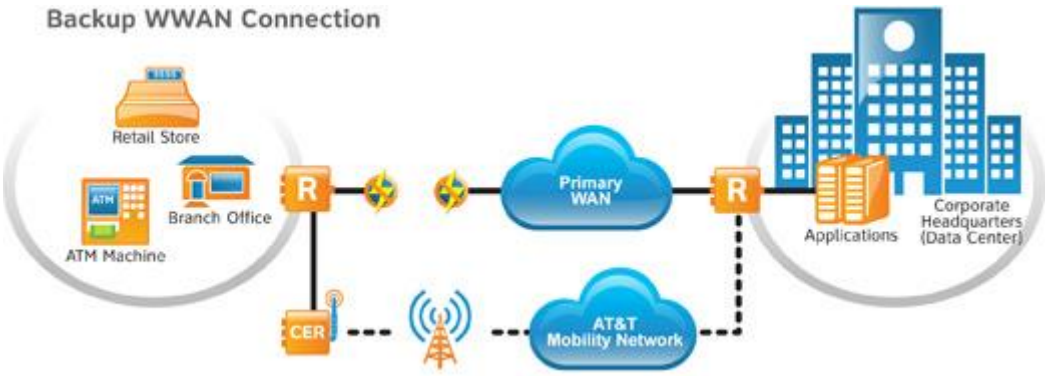
- ***WLAN(Wireless Lan Area Network): Mạng Lan không dây***

Các thiết bị trong mạng có khả năng kết nối rộng hơn với nhiều vùng phủ sóng khác nhau, phạm vi di chuyển từ 100m tới 500m, tốc độ truyền dữ liệu tối đa 54Mbps. Ưu điểm của mạng WLAN là dễ cấu hình và cài đặt, tiết kiệm chi phí mở rộng mạng. Nhược điểm là tốc độ chậm hơn mạng LAN và dễ bị nhiễu.

Hình 1. 2 WLAN

- ***WWAN(Wireless Wide Area Network)***

Mạng không dây triển khai trên qui mô rộng. Sử dụng băng tần đã được đăng kí trước sử dụng các chuẩn mở GMS, CDMA. Phạm vi hoạt động hàng trăm km, tốc độ truyền từ 5Kbps tới 20Kbps. Ưu điểm là dễ mở rộng hệ thống mạng, các thiết bị di chuyển trong phạm vi rộng nhưng nhược điểm là dễ bị ảnh hưởng của tác động môi trường, không an toàn, chất lượng mạng chưa cao, chi phí lắp đặt lớn.



Hình 1. 3 WWAN

1.1.2.2. Phân loại theo sự di động của các thiết bị di động trong mạng

- ***Mạng không dây cố định (Fixed wireless network)***

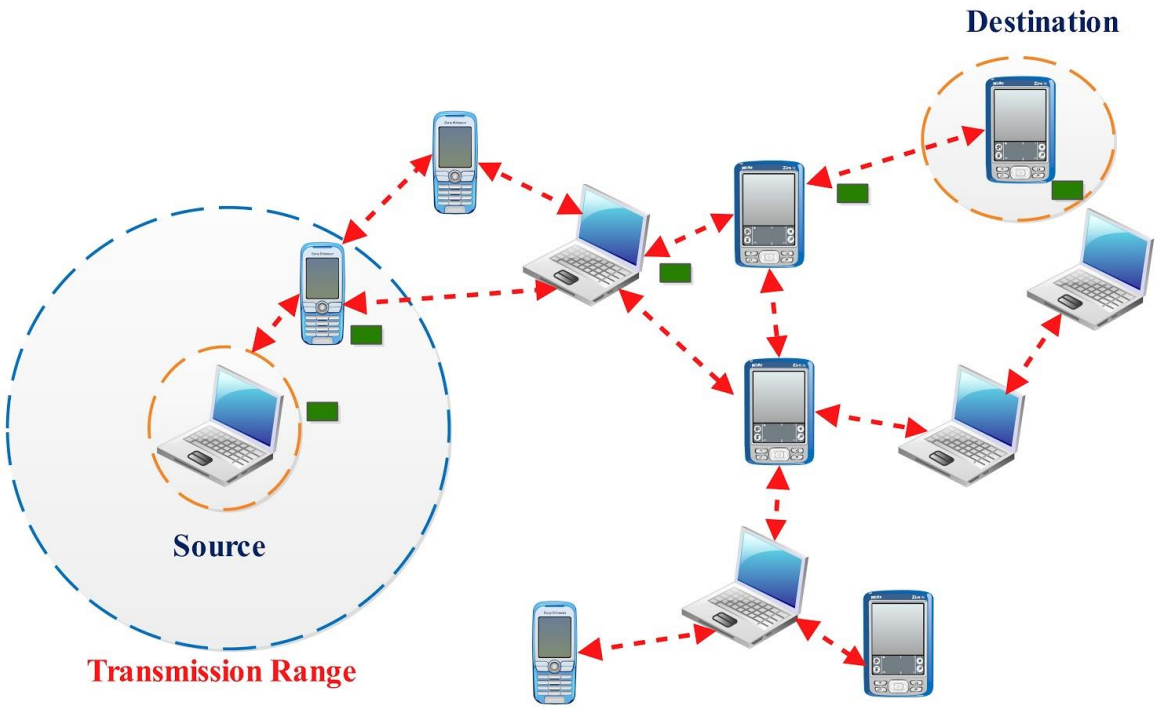
Các bộ định tuyến và node mạng sử dụng các kênh không dây để kết nối với nhau chẳng hạn sử dụng anten để kết nối.

- ***Mạng không dây với các điểm truy cập cố định (Wireless network with fixed access point)***

Các điểm cố định đóng vai trò như một thiết bị định tuyến cho các node mạng khác.

- ***Mạng tùy biến (MANET – Mobile Ad hoc Network)***

Trong mô hình này các node mạng là ngang hàng, không yêu cầu có các thiết bị trung tâm, sử dụng thuận tiện cho các vùng không thể xây dựng hạ tầng mạng.

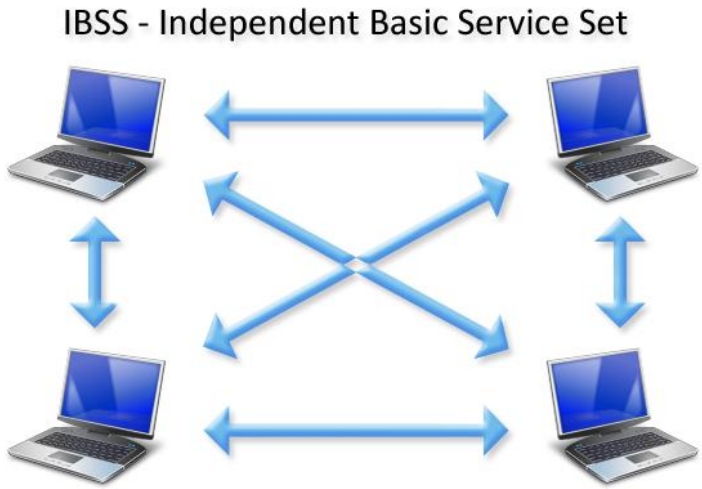


Hình 1. 4 MANET

1.1.3. Mô hình mạng không dây

1.1.3.1. Mô hình mạng độc lập (IBSS)

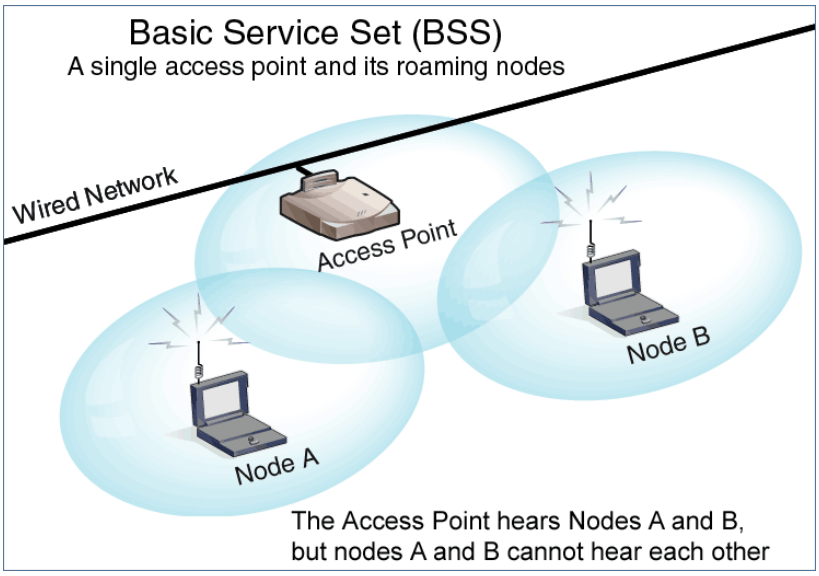
Các trạm kết nối trực tiếp ngang hàng với nhau nên không cần thông qua hạ tầng mạng nào.



Hình 1. 5 IBSS

1.1.3.2. Mô hình mạng cơ sở (BSS)

Đòi hỏi phải có một thiết bị đặc biệt làm trung tâm (AP) để liên lạc cho mọi thiết bị trong cùng một dịch vụ cơ bản, các thiết bị không liên lạc trực tiếp với nhau, AP trong mạng có thể kết nối với mạng có dây.



Hình 1. 6 BSS

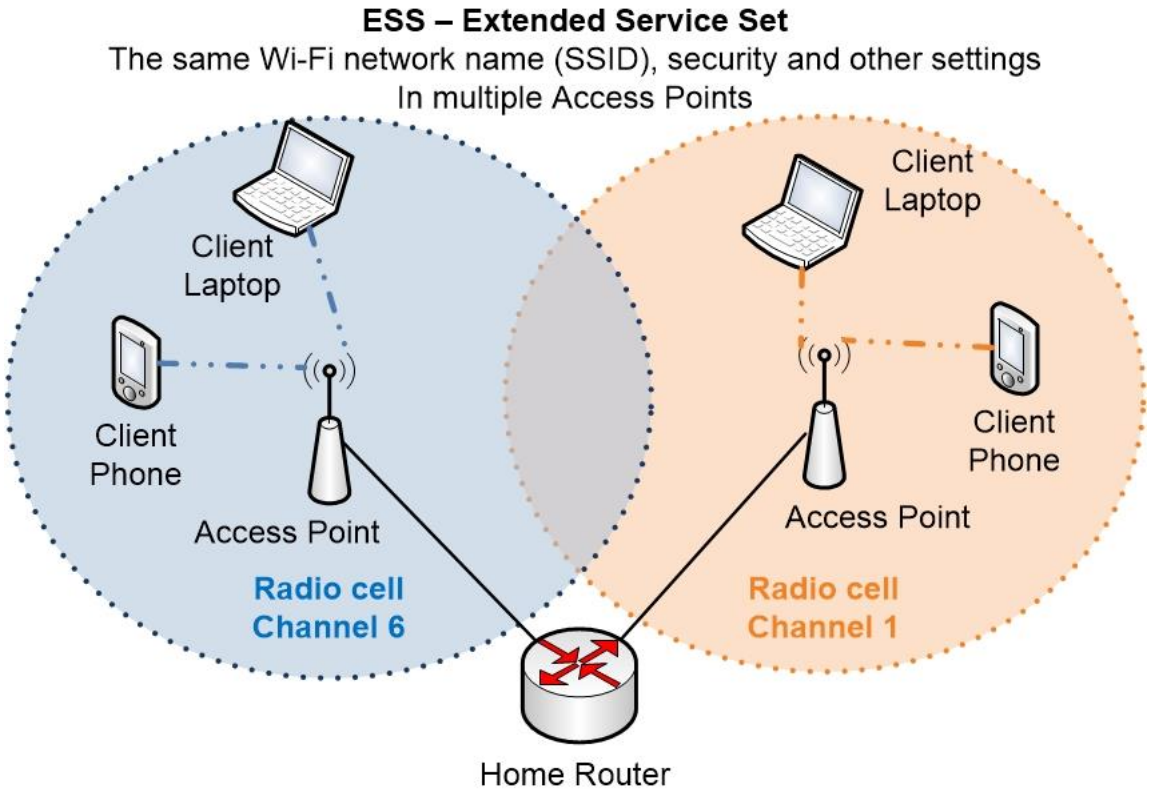
1.1.3.3. Mô hình mạng mở rộng (ESS) ghép nối các BSS thành mạng lớn được gọi là ESS

Yêu cầu thiết bị sử dụng mạng không dây.

Điểm truy cập (AP – Access Point).

AP là thiết bị phổ biến nhất trong hệ thống mạng không dây, cung cấp cho các máy khách một điểm truy cập vào mạng. AP là một thiết bị song công Full duplex có mức độ thông minh tương đương với một chuyển mạch phức tạp – Switch.

AP có thể giao tiếp với các máy không dây, các mạng có dây truyền thống và các AP khác. Trong từng cơ chế giao tiếp cụ thể, AP sẽ hoạt động dưới các chế độ khác nhau. Có 3 chế độ hoạt động chính của AP là: Root mode, Repeter mode và Bridge mode.



Hình 1. 7 ESS

- ***Chế độ Root mode***

  Được sử dụng khi AP kết nối với mạng backbone có dây thông qua giao diện có dây, thường là mạng Ethernet. Hầu hết các AP sẽ hỗ trợ các mode khác ngoài root mode. Tuy nhiên root mode là cấu hình mặc định, khi một AP được kết nối với cùng một hệ thống phân phối có dây thông qua cổng Ethernet của nó thì sẽ được cấu hình để hoạt động trong root mode. Khi ở trong root mode , các AP được kết nối với cùng một hệ thống phân phối có dây có thể nói chuyện được với nhau thông qua phân đoạn có dây. Các client không dây có thể giao tiếp với các client khác nằm trong những cell (vùng phủ sóng của AP) khác nhau thông qua AP tương ứng mà chúng kết nối vào, sau đó các AP này sẽ giao tiếp với nhau thông qua phân đoạn có dây.

- ***Chế độ cầu nối (Bridge Mode)***

  Trong chế độ này, AP hoạt động hoàn toàn giống một cầu nối không dây. Chỉ một số ít các AP trên thị trường có chức năng này do giá thành cao.

- ***Chế độ lặp (repeater mode)***

  AP có khả năng cung cấp một đường kết nối không dây vào mạng có dây thay vì một kết nối có dây bình thường, ở chế độ này một AP hoạt động như một AP và AP còn lại hoạt động như một Repeater không dây.

1.1.4. Đặc điểm mạng không dây

- Cung cấp tất cả các tính năng của công nghệ mạng LAN mà không bị giới hạn bởi kết nói vật lí, tạo ra sự thuận lợi trong việc truyền tải dữ liệu giữa các thiết bị trong hệ thống mạng.
- Tiết kiệm chi phí trong triển khai mạng, phí thiết kế dây dẫn, bảo dưỡng. Tiết kiệm thời gian triển khai, có khả năng mở rộng và linh động khi triển khai hệ thống mạng.
- Vấn đề bảo mật trong mạng không dây là mối quan tâm hàng đầu. Trong mạng cố định truyền thống tín hiệu truyền được truyền qua dây dẫn nên có tính bảo mật cao hơn. Trong mạng không dây, việc thâm nhập vào hệ thống mạng sẽ trở nên dễ dàng do mạng này sử dụng sóng vô tuyến truyền trong không khí nên có thể được bắt bởi bất kì thiết bị nhận nào nằm trong phạm vi cho phép.
- Mạng không dây không có ranh giới rõ ràng nên cũng khó quản lí.

1.2. Mạng tùy biến di động – MANET

1.2.1. Giới thiệu mạng tùy biến di động

Mạng đặc biệt di động MANET (Mobile Ad hoc NETwork) được hình thành bởi các nút di động có trang bị các giao tiếp mạng không dây cần thiết lập truyền thông không cần tới sự hiện diện của các cơ sở hạ tầng mạng và các quản trị trung tâm. Mục đích của làm việc mạng MANET là mở rộng sự di động sang miền tự trị, di động, không dây. Các nút trong mạng chạy các ứng dụng và có thể chuyển tiếp các gói tin cho các nút khác. Khả năng về làm việc của mạng MANET bắt nguồn từ 1968 khi các mạng ALOHA được thực hiện. Mục tiêu của mạng này là kết nối các cơ sở giáo dục ở Hawaii. Mặc dù các trạm làm việc là cố định, giao thức ALOHA đã thực hiện việc quản lý truy cập kênh truyền dưới dạng phân tán, do đó đã cung cấp cơ sở cho sự phát triển về sau của các lược đồ truy cập kênh phân tán cho phép hoạt động của mạng AD HOC.[2]

Khởi nguồn từ các mạng ALOHA, và những phát triển ban đầu của mạng cố định chuyển mạch gói, tổ chức DARPA đã bắt đầu làm việc trên các mạng vô tuyến gói tin PRnet (Packet Radio network) vào năm 1973. Đây là mạng vô tuyến gói tin đa chặng đầu tiên. Trong ngữ cảnh này, đa chặng có nghĩa là các nút hợp tác để chuyển tiếp truyền thông cho các nút ở xa nằm ngoài dải truyền

thông của một nút khác. PRnet đã cung cấp cơ chế cho việc quản lý hoạt động trên cơ sở tập trung cũng như phân tán. Người ta cũng bắt đầu nhận thấy nhiều lợi điểm của việc làm việc đa chặng so với đơn chặng. Triển khai đa chặng tạo điều kiện thuận lợi cho việc dùng lại các tài nguyên kênh truyền về cả không gian và thời gian và làm giảm năng lượng phát cần thiết. Trong khi đó, làm việc đơn chặng chỉ chia sẻ các tài nguyên kênh về thời gian và yêu cầu năng lượng cao hơn để có thể giao tiếp được với các nút ở xa. Trong cả hai trường hợp, ngữ cảnh mạng là hoàn toàn giống nhau về sự phân bố của nút, các nguồn phát và các đích. Trong trường hợp đa chặng, các gói tin được định tuyến thông qua nhiều điểm chuyển phát. Tuy nhiên, trong mạng đơn chặng, gói tin được gửi trực tiếp từ nguồn tới đích [2][3].

Mặc dù nhiều mạng vô tuyến gói tin đã được phát triển sau đó, các hệ thống không dây này vẫn chưa bao giờ xuất hiện với người dùng. Khi chuẩn IEEE 802.11, một chuẩn cho mạng cục bộ không dây được phát triển, viện IEEE đã thay thế khái niệm mạng vô tuyến gói tin thành mạng AD HOC. Các mạng vô tuyến gói tin do đó thường gắn với các mạng đa chặng rộng lớn trong quân sự . Với việc đưa ra một tên mới cho mạng vô tuyến gói tin đa chặng, IEEE hi vọng sẽ cho thấy những ngữ cảnh triển khai hoàn toàn mới của loại mạng này.[3]

Một số các công nghệ không dây hiện tại hỗ trợ làm việc mạng MANET là Bluetooth và IEEE 802.11. Trong đó, IEEE 802.11 là chuẩn cục bộ không dây có cơ sở hạ tầng được bổ sung chức năng hỗ trợ làm việc mạng MANET. Mạng IEEE 802.11b làm việc ở dải băng tần 2.4GHz với tốc độ dữ liệu 11Mbps và hiện tại đã đạt tới 20 Mbps. Chuẩn IEEE 802.11a tiếp theo hoạt động ở dài băng tần 5GHz và tốc độ dữ liệu đạt tới 54Mbps. Trong khi đó, Bluetooth là kiến trúc làm việc mạng MANET không dây dải sóng ngắn cho các mạng cá nhân WPAN (Wireless Personal Area Network). Mạng làm việc này nhằm mục đích kết nối các thiết bị cá nhân di động như các máy tính laptop, PDA, các thiết bị ngoại vi, điện thoại cầm tay, các máy quay kỹ thuật số, các headset, và các thiết bị điện tử khác. Diện hoạt động của mạng do vậy rất nhỏ thường dưới 10m xung quanh cá nhân thường được gọi là không gian hoạt động cá nhân POS-Personal Operating Space.[4]

1.2.2. Ứng dụng mạng MANET

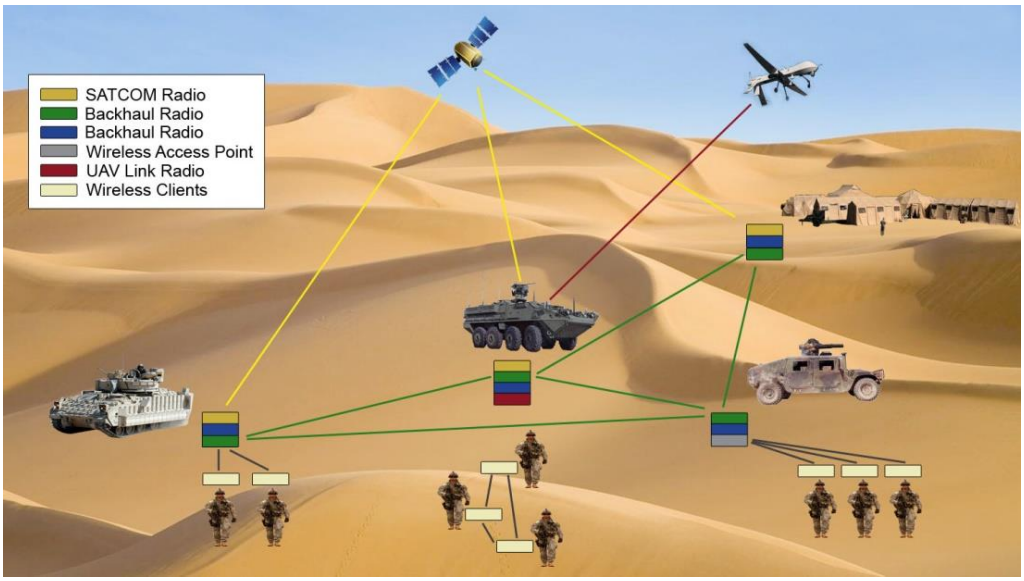
Các ứng dụng đầu tiên của mạng vô tuyến gói tin MANET là ở trong quân sự. Hoạt động phi tập trung của mạng chính là nhu cầu cần thiết đối với hoạt động quân sự. Ngày nay, các thiết bị tính toán không dây, di động vẫn ở mức giá

khá cao. Tuy nhiên, khả năng của các máy tính di động sẽ tăng lên, và nhu cầu về làm việc mạng không giới hạn do vậy cũng sẽ tăng. Các mạng MANET có thể được dùng trong các tính huống khi không có cơ sở hạ tầng cố định hoặc tế bào tồn tại. Mạng MANET có thể được triển khai trong truy cập công cộng không dây ở các khu vực thành phố, trường học giúp thực hiện nhanh các truyền thông và mở rộng diện hoạt động. Các điểm truy cập có thể dùng như các trạm tiếp sóng cố định thực hiện việc định tuyến giữa chúng và giữa các nút người dùng. Một số điểm truy cập có thể dùng như gateway cho phép người dùng kết nối tới mạng xương sống cố định.

Ở mức cục bộ, mạng MANET liên kết các notebook hoặc các máy tính laptop để phân phát và chia sẻ thông tin giữa những người tham gia trong một hội nghị hay lớp học. Mạng MANET cũng thích hợp cho các ứng dụng trong mạng gia đình.

Trong đó, các thiết bị có thể truyền thông trực tiếp với nhau để trao đổi thông tin dữ liệu như âm thanh, hình ảnh, báo thức, và các cập nhật cấu hình. Mạng MANET còn được biết đến như mạng cảm ứng (sensor network) trong các ứng dụng về kiểm soát môi trường. Các mạng này có thể được dùng để dự báo những ô nhiễm về nguồn nước hoặc những cảnh báo sớm về lũ lụt hoặc sóng thần. Các mạng MANET dải sóng ngắn làm đơn giản hóa truyền thông giữa các thiết bị di động khác nhau như điện thoại tế bào và PDA bằng việc hình thành các mạng WPAN và loại bỏ sự kết nối bởi các cáp. Mạng có thể giúp chia sẻ khả năng truy cập Internet và các tài nguyên trong mạng như máy in giữa các thiết bị. Khả năng này giúp mở rộng tính di động của người dùng.

Với sự hợp tác cùng truyền thông vệ tinh, công nghệ MANET có thể cung cấp phương pháp linh động cho việc thiết lập các truyền thông trong các hoạt động cứu hộ, chữa cháy, an toàn, các ngữ cảnh yêu cầu sự truyền thông được triển khai nhanh. Ngoài ra, nhiều ngữ cảnh sử dụng mạng MANET có thể phong phú hơn khi các mạng này được triển khai một cách mở rộng.



Hình 1. 8 Ứng dụng MANET trong quân sự

Hình 1. 9 Ứng dụng MANET trong dân sự

1.2.3. Các đặc điểm mạng MANET

Trong mạng MANET, các nút là di động và được trang bị các bộ phát và nhận không dây sử dụng các loại ăng ten khác nhau. Tại một thời điểm, phụ thuộc vào vị trí của nút và dạng bao phủ của bộ nhận và phát tín hiệu, mức năng lượng phát và các mức giao thoa cùng kênh, kết nối không dây giữa các nút có dạng ngẫu nhiên, đồ thị đa chặng. Cấu hình này thay đổi theo thời gian do các nút di chuyển hoặc điều chỉnh các tham số phát và nhận sóng. **[2][3]**

Cấu hình mạng động: Do sự di chuyển của các nút, mạng thông thường là đa chặng, có thể thay đổi một cách ngẫu nhiên và nhanh chóng tại bất kỳ thời điểm nào và có thể chứa các liên kết hai chiều cũng như một chiều**. [2]**

Băng thông hạn chế, khả năng của các liên kết có thể biến đổi: các liên kết không dây có băng thông thấp hơn đáng kể so với các đường truyền cáp. Thêm vào đó, thông lượng của các truyền thông không dây do các ảnh hưởng của đa truy cập, sự suy giảm, nhiễu, và các điều kiện giao thoa thường nhỏ hơn tốc độ truyền lớn nhất của sóng vô tuyến.

Các nút có năng lượng thấp: Một số hoặc tất cả các nút trong mạng MANET dùng pin để cung cấp năng lượng hoạt động cho các thành phần trong thiết bị. Do vậy, các nút trong mạng MANET hạn chế về khả năng tính toán của CPU, kích thước bộ nhớ, khả năng xử lý tín hiệu, và mức năng lượng phát và nhận sóng

Bảo mật vật lý giới hạn: Do việc truyền qua không khí, các mạng không dây tiềm ẩn nhiều nguy cơ bảo mật hơn các mạng cáp. Nhiều khả năng tấn công bảo mật như nghe trộm, giả mạo, và từ chối dịch vụ (DoS) có thể xảy ra. Các kỹ thuật bảo mật cần được triển khai trên nhiều tầng giao thức để làm giảm các nguy cơ bảo mật.

1.3. Các vấn đề quan trọng phải nghiên cứu, giải quyết đối với mạng MANET

1.3.1. Vấn đề định tuyến trong mạng MANET

Định tuyến mạng là việc tìm đường đi từ nguồn tới đích qua hệ thống mạng. Giao thức định tuyến có chức năng chính là lựa chọn đường cho các cặp nguồn -đích và phân phát gói tin tới đích chính xác. Truyền thông trong mạng MANET dựa trên các đường đi đa chặng, do vậy định tuyến các gói tin là hoạt động quan trọng. Khác với các mạng cố định có cấu hình ít thay đổi hoặc gần như không thay đổi, các vấn đề về không dây và tính chất động của mạng AD HOC khiến cho các giao thức định tuyến được thiết kế cho các mạng cố định không thể áp dụng hoặc gần như thất bại trong mạng AD HOC. Việc thiết kế một giao thức định tuyến làm việc hiệu quả trong mạng AD HOC là một bài toàn khó.

Do đó, thiết kế của các giao thức định tuyến trong mạng AD HOC thường xem xét một số các yếu tố sau đây:

- ***Hoạt động phân tán***

  Cách tiếp cận tập trung sẽ thất bại do sẽ tốn rất nhiều thời gian để tập hợp một trạng thái hiện tại và phát tán lại nó. Trong thời gian đó, cấu hình có thể đã có các thay đổi khác.

- ***Không có lặp định tuyến***

  Hiện tượng xảy ra khi một phần nhỏ các gói tin quay vòng trong mạng trong một khoảng thời gian nào đó. Một giải pháp có thể là sử dụng giá trị thời gian quá hạn.

- ***Tính toán đường dựa trên yêu cầu***

  Thay thế việc duy trì định tuyến tới tất cả các nút tại tất cả các thời điểm bằng việc thích ứng với dạng truyền thông. Mục đích là tận dụng hiệu quả năng lượng và băng thông, mặc dù độ trễ tăng lên do sự phát hiện đường.

- ***Tính toán đường trước***

  Khi độ trễ có vai trò quan trọng, và băng thông, các tài nguyên năng lượng cho phép, việc tính toán đường trước sẽ giảm độ trễ phân phát.

- ***Bảo mật***

  Giao thức định tuyến mạng AD HOC có khả năng bị tấn công dễ dàng ở một số dạng như xâm nhập truyền thông, phát lại, thay đổi các tiêu đề gói tin, điều hướng các thông điệp định tuyến. Do vậy, cần có các phương pháp bảo mật thích hợp để ngăn chặn việc sửa đổi hoạt động của giao thức.

- ***Hoạt động nghỉ***

  Giao thức định tuyến cần cung cấp yêu cầu bảo tồn năng lượng của các nút khi có thể.

- ***Hỗ trợ liên kết đơn hướng:***

  Hỗ trợ trường hợp khi các liên kết đơn hướng tồn tại trong mạng AD HOC.

1.3.2. Vấn đề bảo mật trong mạng MANET

Mạng Manet có đặc điểm như giới hạn về băng thông, năng lượng, cấu trúc mạng động, tỉ lệ lỗi truyền tin cao, các node trong mạng tự do di chuyển, các node vừa đóng vai trò như router tìm và duy trì đường đường khi kết hợp với các node khác trong mạng.

Bởi vậy, giao thức định tuyến trong mạng có dây không thể sử dụng được trong mạng MANET, nhiều giao thức được thiết kế cho hình thức mạng đặc thù này.

Các giao thức trong mạng có thể chia thành hai loại dựa trên tiêu chí quản lí bảng định tuyến:

- Giao thức sử dụng bảng định tuyến

- Giao thức định tuyến theo yêu cầu

Tấn công mạng AD HOC trong tầng mạng có hai mục đích:

- Không chuyển tiếp gói tin
- Chèn làm thay đổi một vài tham số của bản tin định tuyến như số tuần tự của gói tin (sq#) và địa chỉ IP

1.3.2.1. Table Driven Routing Protocols

Mỗi node luôn cập nhật bảng định tuyến. Để cập nhật bảng định tuyến mọi node lan truyền tin nhắn để cập nhật cho toàn bộ mạng khi cấu trúc mạng thay đổi.

Bởi vì mỗi node đều có thông tin của toàn bộ mạng nên có nhiều nhược điểm sau:

- Toàn bộ mạng được update định kì nên tốn nhiều băng thông.

- Mỗi node cập nhật toàn bộ cấu trúc mạng nên tốn năng lượng.

- Nhiều thông tin định tuyến thừa vô ích trong bảng định tuyến.

1.3.2.2. Giao thức định tuyến theo yêu cầu

- Bảng định tuyến không được cập nhật theo chu kì mà chỉ cập nhật khi có yêu cầu.

- Khi node nguồn có nhu cầu kết nối với node đích, nó quảng bá gói tin yêu cầu tuyến tới các node láng giềng.

- Chỉ có node láng giềng với node nguồn mới nhận được RREQ, khi nhận được các node này chuyển tiếp tới các node hàng xóm cho tới khi node đích được tìm thấy.

- Sau đó node đích gửi gói tin trả lời tuyến tới node nguồn theo đường ngắn nhất.

- Các đường định tuyến còn lại trong bảng định tuyến của các node thông qua đường ngắn nhất cho tới khi không còn cần tới nữa.

CHƯƠNG 2. TẤN CÔNG LỖ ĐEN TRONG GIAO THỨC ĐỊNH TUYẾN AODV VÀ MỘT SỐ GIẢI PHÁP PHÒNG CHỐNG

Với các đặc điểm riêng có của mình, mạng MANET nói riêng và mạng không dây nói chung phải đối mặt với rất nhiều hình thức tấn công. Tấn công vào giao thức định tuyến là hình thức tấn công phổ biến . Trong chương này tác giả sẽ trình bày tổng quan về các hình thức tấn công trong mạng MANET, cách thức hoạt động của giao thức định tuyến điển hình AODV và tấn công lỗ đen vào giao thức định tuyến này.

2.1. Giao thức định tuyến AODV

- **Giao thức AODV (Ad hoc On Demand Distance Vector):[9][10][11]**

Giao thức định tuyến AODV là một trong những giao thức định tuyến theo cơ chế phản ứng trong hệ thống mạng MANET. Giao thức AODV phát gói tin broadcast để yêu cầu tìm đường khi có nhu cầu. AODV sử dụng nhiều cơ chế để duy trì thông tin bảng định tuyến, chẳng hạn như nó sử dụng bảng định tuyến truyền thống để lưu trữ thông tin định tuyến với mỗi entry cho một địa chỉ đích.

AODV không cần biết thông tin về các node láng giềng của nó mà dựa trên thông tin của bảng định tuyến để phát gói tin RREP về node nguồn và node nguồn dùng thông tin đó để gởi dữ liệu đến đích. Để đảm bảo rằng thông tin trong bảng định tuyến là mới nhất thì AODV sử dụng sequence Number (dùng để nhận ra và loại bỏ các đường đi không còn giá trị trong quá trình cập nhật bảng định tuyến). Mỗi node sẽ có một bộ tăng số sequence number riêng cho nó.

Quá trình định tuyến của AODV bao gồm 2 cơ chế chính: cơ chế tạo thông tin định tuyến và cơ chế duy trì thông tin định tuyến.

2.1.1. Cơ chế tạo thông tin định tuyến (route discovery)[12][13]

Mỗi node luôn có hai bộ đếm (counter): bộ đếm số sequence number và bộ đếm REQ_ID. Số sequence number được tăng lên trong các trường hợp:

- Trước khi một node khởi động tiến trình route discovery, điều này chống sự xung đột với các gói RREP trước đó.
- Trước khi node đích gởi gói RREP trả lời gói RREQ , nó sẽ cập nhật lại giá trị sequence number lớn nhất của số sequence number hiện hành mà nó lưu giữ với số sequence number trong gói RREQ.

- Khi có một sự thay đổi trong mạng cục bộ của nó (mạng cục bộ là mạng các node láng giềng). Số REQ_ID được tăng lên khi node khởi động một tiến trình route discovery mới.

  Tiến trình Route Discovery được khởi động khi nào một node muốn trao đổi dữ liệu với một node khác mà trong bảng định tuyến của nó không có thông tin định tuyến đến node đích đó. Khi đó tiến trình sẽ phát broadcast một gói RREQ cho các node láng giềng của nó. Thông tin trong RREQ ngoài địa chỉ đích, địa chỉ nguồn, số hop-count. Còn có các trường: số sequence number của node nguồn, số broadcast ID, số sequence number của node đích. Cặp thông tin <địa chỉ nguồn, số REQ_ID> là số định danh duy nhất cho một gói RREQ. Khi các node láng giềng nhận được gói RREQ, thì nó sẽ kiểm tra tuần tự theo các bước:

- *Bước 1*: Xem các gói RREQ đã được xử lý chưa? Nếu đã được xử lý thì nó sẽ loại bỏ gói tin đó và không xử lý thêm. Ngược lại chuyển qua bước 2.
- *Bước 2*: Nếu trong bảng định tuyến của nó chứa đường đi đến đích, thì sẽ kiểm tra giá trị destination sequence number trong entry chứa thông tin về đường đi với số destination sequence number trong gói RREQ, nếu số destination sequence number trong RREQ lớn hơn số destination squence number trong entry thì nó sẽ không sử dụng thông tin trong entry của bảng định tuyến để trả lời cho node nguồn mà nó sẽ tiếp tục phát broadcast gói RREQ đó đến cho các node láng giềng của nó. Ngược lại nó sẽ phát unicast cho gói RREP ngược trở lại cho node láng giềng của nó để báo đã nhận gói RREQ. Gói RREP ngoài các thông tin như: địa chỉ nguồn, địa chỉ đích,… còn chứa các thông tin: destination sequence number, hop-count, TTL. Ngược lại thì qua bước 3.
- *Bước 3*: Nếu trong bảng định tuyến của nó không có đường đi đến đích thì nó sẽ tăng số Hop-count lên 1, đồng thời nó sẽ tự động thiết lập một đường đi ngược (reverse path) từ nó đến node nguồn bằng cách ghi nhận lại địa chỉ của node láng giềng mà nó nhận gói RREQ lần đầu tiên. Entry chứa đường đi ngược này sẽ được tồn tại trong một khoảng thời gian đủ để gói RREQ tìm đường đi đến đích và gói RREP phản hồi cho node nguồn, sau đó entry này sẽ được xóa đi.

Quá trình kiểm tra này sẽ lặp tuần tự cho đến khi gặp node đích hoặc một node trung gian mà có các điều kiện thỏa bước 2. Trong quá trình trả về gói RREP, một node có thể nhận cùng lúc nhiều gói RREP, khi đó nó sẽ chỉ xử lý gói RREP có số Destination Sequence number lớn nhất, hoặc nếu cùng số Destination sequence number thì nó sẽ chọn gói RREP có số Hop-count nhỏ

nhất. Sau đó nó sẽ cập nhật các thông tin cần thiết vào trong bảng định tuyến của nó và chuyển gói RREP đi.

2.1.2. Cơ chế duy trì thông tin định tuyến (Route Maintenance)[12][13]

Khi một node nhận thấy rằng chặng tiếp theo trong đường đi tới đích của nó không thể tìm thấy, thì nó sẽ phát một gói RREP khẩn cấp với số sequence number bằng số sequence number trước đó cộng với 1 và gởi đến tất cả các node láng giềng đang ở trạng thái active, những node đó sẽ tiếp tục chuyển gói tin đó đến các node láng giềng khác, cứ như vậy cho đến khi tất cả các node trong mạng mà ở trạng thái active nhận được gói tin này.

Sau khi nhận thông báo đó, các node có thể sẽ khởi động lại tiến trình route discovery nếu nó có nhu cầu định tuyến dữ liệu đến node bị hỏng, để biết node cần có nhu cầu định tuyến đến đích hay không thì nó sẽ kiểm tra ở các giao thức bên dưới có kết nối nào đến node đích mà còn mở hay không? Nếu thấy cần có nhu cầu định tuyến nó sẽ gởi một gói RREQ (với số sequence number bằng số sequence number mà nó biết trước đó cộng thêm 1) đến các node láng giềng để tìm đến địa chỉ đích. Để kiểm tra trạng thái một node có active hay không ADOV sử dụng một bộ đếm thời gian. Một entry của bảng định tuyến sẽ bị xem là không active nếu nó không được sử dụng thường xuyên.

Thông điệp hello định kỳ có thể được sử dụng để đảm bảo các liên kết, cũng như để phát hiện các lỗi liên kết. Khi một nút trung gian không thể kết nối, nút nguồn truyền một RREQ với một số RREQ-ID mới (một số thứ tự lớn hơn số thứ tự trước đây được biết).

Những nút sau đó chuyển tiếp tin nhắn đó đến các node láng giềng hoạt động của nó. Quá trình này tiếp tục cho đến khi tất cả các nút nguồn hoạt động được thông báo.

Khi nhận được thông báo của một liên kết bị hỏng, nút nguồn có thể khởi động lại quá trình định tuyến nếu vẫn cần yêu cầu một tuyến đường đến đích. Để xác định xem một tuyến đường vẫn còn cần thiết, một nút có thể kiểm tra xem tuyến đường đã được sử dụng gần đây có sử dụng đường định tuyến này hay không, nếu không thì xóa bỏ nó ra khỏi bảng định tuyến.

2.2. Lỗ hổng bảo mật và một số kiểu tấn công giao thức định tuyến AODV

2.2.1. Lỗ hổng bảo mật trong giao thức định tuyến AODV

Giao thức AODV dễ bị kẻ tấn công làm sai lệch thông tin đường đi để chuyển hướng đường đi và bắt đầu các cuộc tấn công khác. Sự sai sót của bất cứ trường nào trong gói tin điều khiển có thể khiến AODV gặp sự cố. Các trường dễ bị phá hoại trong thông điệp định tuyến AODV như số squence number, hopcout, ID của gói tin… Để thực hiện một cuộc tấn công lỗ đen trong giao thức AODV, nút độc hại chờ gói tin RREQ gửi từ các nút láng giềng của nó. Khi nhận được gói RREQ, nó ngay lập tức gửi trả lời gói tin RREP với nội dung sai lệch trong đó thiết lập giá trị squence number cao nhất và giá trị hopcout nhỏ nhất mà không thực hiện kiểm tra bảng định tuyến xem có tuyến đường tới đích nào không trước khi các nút khác (trong đó gồm các nút trung gian có tuyến đường hợp lệ hoặc chính nút đích) gửi các bảng tin trả lời tuyến. Sau đó mọi dữ liệu truyền từ nút nguồn tới nút đích được nút độc hại loại bỏ (drop) toàn bộ thay vì việc chuyển tiếp tới đích thích hợp.**[13][14][15][22]**

2.2.2. Một số kiểu tấn công vào giao thức AODV

2.2.2.1. Hình thức tấn công lỗ đen trong giao thức định tuyến AODV

Để thực hiện cuộc tấn công black hole attack trong giao thức AODV, có thể thực hiện theo hai cách:**[16][17]**

- RREQ black hole attack
- RREP black hole attack

Giao thức AODV sử dụng một đường đi từ node nguồn tới node đích. Để tìm đường tới đích, các node trong mạng hợp tác chia sẻ thông thông tin thông qua gói tin điều khiển.

Lợi ích:

- Đáp ứng nhanh
- Ít xử lí
- Bộ nhớ ít
- Băng thông mạng ít
- Ít gói tin điều khiển

Sử dụng số Sq# cho mỗi bảng định tuyến. Số Sq# được sinh ra bởi node đích để đáp ứng:

- Đường đi từ nguồn tới đích không được lặp vòng và phải là đường đi ngắn nhất
- Gói tin điều khiển bao gồm: RREQ - Route Requests, RREPs - Route Reply, RRERs - Route Errors.
- Sử dụng giao thức UDP/IP để gửi gói tin data

Tấn công blackhole vào gói tin RREP được thực hiện như sau:

- Node bị tấn công khi nhận được yêu cầu tuyến sẽ ngay lập tức đáp ứng lại với bản tin RREP có số sequence number lớn nhất và số hopcount nhỏ nhất.

- Node nguồn sau khi nhận được các gói tin RREP trả lời sẽ tiến hành chọn lựa, lấy RREP có sequence number lớn nhất và hopcount nhỏ nhất.

- Quá trình tạo thông tin định tuyến được thiết lập, gói tin dữ liệu được chuyển tới node độc hại, tuy nhiên thay vì chuyển tiếp gói tun dữ liệu tới node đích thật sự thì node độc hại xóa bỏ hoàn toàn gói tin dữ liệu.

2.2.2.2. Các kiểu tấn công khác

- ***Passive Eavesdropping (Nghe lén):*** **[18][19]**
  - Kẻ tấn công lắng nghe bất kì mạng không dây nào để biết cái gì sắp diễn ra trong mạng. Đầu tiên nó lắng nghe các gói tin điều khiển để luận ra cấu trúc mạng từ đó hiểu được các node được giao tiếp với các node khác như thế nào. Bởi vậy kẻ tấn công có thể đoán biết được thông tin về mạng trước khi tấn công.
  - Nó cũng lắng nghe thông tin được chuyển giao mặc dù thông tin đó đã được mã hóa bí mật trên tầng ứng dụng.
  - Loại tấn công này cũng vi phạm quyền riêng tư về vị trí địa lí khi nó thông báo sự tồn tại của chủ thể trong vùng địa lí mà không được cho phép.

- ***Selective Existence (Selfish Nodes - Node ích kỉ):*** **[21]**
  - Node độc hại được biết tới như một node ích kỉ trong mạng khi không tham gia vào hệ thống mạng. Nó vẫn tham gia chiếm tài nguyên hệ thống bằng việc phát thông báo đã có những node tồn tại trong mạng để hạn chế sự gia nhập của các node khác.
  - Node độc hại không gửi HELLO message và hủy toàn bộ các gói tin tới nó. Khi node độc hại muốn bắt đầu kết nối với các node khác nó tính toán đường và sau đó gửi các gói tin cần thiết. Khi node này không được sử dụng trong mạng nó chuyển về chế độ im lặng (silent mode). Những node

hàng xóm với nó không thể duy trì kết nối tới node này và khi đó nó chuyển sang vô hình trong mạng.

- **Gray hole Attack [20][21]**
  - Tương tự như cách thức tấn công blackhole, tấn công grayhole cũng có hành vi hủy gói tin dữ liệu thay vì chuyển tiếp tới đích.
  - Tuy nhiên khác với cách tấn công blackhole, tấn công grayhole có thể giữ lại hoặc hủy một số loại gói tin nhất định, ví dụ giữ lại gói tin TCP nhưng hủy gói tin UDP.
  - Tấn công grayhole khó phát hiện hơn blackhole vì sau khi thể hiện hành vi tấn công, node độc hại có thể trở lại trạng thái thông thường.

2.3. Một số giải pháp chống tấn công lỗ đen trong giao thức AODV

2.3.1. Giao thức bảo mật ids-AODV

2.3.1.1 Ý tưởng giao thức

Tấn công blackhole sẽ sinh ra gói tin giả mạo RREP với số Seq# lớn nhất có thể. Khi đó tất cả các gói tin RREP khác đều không được chọn do có Seq# nhỏ hơn. Giao thức ids-AODV **[7]** giả sử rằng RREP có số Seq# lớn thứ 2 mới là gói tin RREP thực vì thế nó sẽ bỏ qua gói tin có số Seq# lớn nhất do tấn công blackhole giả mạo.

Để thực hiện được ý tưởng này, giao thức idsAODV xây dựng cơ chế lưu trữ gói tin RREP với mục đích lấy gói tin có số Seq# lớn thứ 2.

Trong RREP function:

+ nếu gói tin RREP đã được lưu lại từ trước đó cho cùng 1 địa chỉ đích thì thực hiện function RREP như thông thường

+ nếu gói tin RREP chưa từng được lưu lại thì chèn (insert) gói tin vào bộ nhớ đệm, giải phóng gói tin đồng thời thoát khỏi hàm.

```
void idsAODV::rrep_insert(nsaddr_t id)
{idsBroadcastRREP *r = new idsBroadcastRREP(id);
   assert(r);
   r->expire = CURRENT_TIME + BCAST_ID_SAVE;
   r->count++;
   LIST_INSERT_HEAD(&rrephead, r, link);
}
idsBroadcastRREP *
idsAODV::rrep_lookup(nsaddr_t id)
{ idsBroadcastRREP *r = rrephead.lh_first;
   for (; r; r = r->link.le_next) {
      if (r->dst == id)  return r;
   }
   return NULL;
}
```

Hình 2. 1 Các hàm xử lí bộ đệm RREP giao thức ids-AODV

```
idsAODV::recvReply(Packet *p)
{
   idsBroadcastRREP *r = rrep_lookup(rp->rp_dst);
   if (ih->daddr() == index)
   {
      if (r == NULL){
         count = 0;
         rrep_insert(rp->rp_dst);
      }else
      {
         r->count++;
         count = r->count;
      }
      UPDATE ROUTE TABLE
   }else
      Forward(p); }
}
```

Hình 2. 2 Hàm nhận RREP giao thức ids-AODV

2.3.1.2. Cài đặt ids-AODV trên NS-2

Chi tiết về việc cài đặt được tôi trình bày trong phụ lục, ở cuối luận văn.

2.3.2. Giao thức định tuyến ngược PHR-AODV

2.3.2.1. Ý tưởng giao thức

- Giao thức AODV chỉ duy trì 1 đường duy nhất từ node nguồn tới đích do vậy khi đường bị đứt phải khởi tạo đường đi khác.
- Với giao thức phr-AODV**[5][6**] sử dụng nhiều đường để thiết lập truyền thông, khi 1 đường dẫn bị đứt những đường thay thế sẽ được dùng ngay mà không cần khởi tạo.
- Số lượng đường đi từ nguồn tới đích là số cạnh từ node nguồn.
- Dữ liệu sẽ được gửi đi thông qua nhiều đường.
- Đường được chọn sẽ được quyết định thông qua selection process.
- Nếu đường nào bị đứt kết nối thì sẽ được loại bỏ khỏi danh sách đường.
- Khi không còn đường nào trong list thì node nguồn sẽ gửi lại 1 request mới để tìm đường.
- Giao thức phr-AODV yêu cầu node độc hại không phá hủy sự truyền thông giữa node nguồn và đích.

2.3.2.2. Cài đặt giao thức phr-AODV trên NS2

Chi tiết về việc cài đặt được tôi trình bày trong *phụ lục, ở cuối luận văn.*

2.4. Đề xuất cải tiến giao thức bảo mật idsAODV

2.4.1. Ý tưởng

Giao thức idsAODV có nhược điểm là loại bỏ ngay bản tin RREP có số Seq# lớn nhất tuy nhiên nếu trong mạng node đích hoặc node trung gian có Seq# lớn bằng với số cực đại Seq# thì có thể dẫn tới bỏ qua RREP hợp lệ.

2.4.2. Cải tiến ids-AODV 1

Đề xuất chỉ loại bỏ RREP khi số Seq = max Seq và hopcount = 1 bởi rất ít trường hợp node có đường đi hợp lệ thật sự có số hopcount đúng bằng 1 và số Seq max.

```
if (count > 1 ||
    (rt->rt_seqno < rp->rp_dst_seqno) || // newer route
    ((rt->rt_seqno == rp->rp_dst_seqno) &&
     (rt->rt_hops > rp->rp_hop_count) &&
     (rp->rp_dst_seqno < 4294967295 && rp->rp_hop_count > 1)))
{ // shorter or better route
    printf("valid: %f\t", rp->rp_timestamp);
  // do someting
}
```

Hình 2. 3 Điều kiện gói tin RREP là hợp lệ trong cải tiến ids-AODV

Tuy nhiên: Nếu kẻ tấn công không sử dụng node độc hại có số maxSeq và hopcout =1 thì cũng không loại bỏ RREP được sinh ra bởi node độc hại.

2.4.3. Cải tiến ids-AODV 2

Node độc hại lắng nghe nếu có RREQ yêu cầu định tuyến thì sẽ trả lời ngay tức khắc RREP mà không qua quá trình xử lí gói tin (đưa gói tin RREQ vào hàng đợi, tìm kiếm trong bảng định tuyến).

Thời gian tối thiểu kể từ khi node có đường đi tới đích nhận được RREQ tới khi node nguồn nhận RREP:

TimeMin = Transmission Time + queuing Time + ProcessingTime

Trong đó:

Transmission Time = Packet size / Bit rate. Vì RREP là gói tin điều khiển nên Packet size = CTS size = 14 bytes; Bit rate = 4 packets/s.

Queueing time = RREP_WAIT_TIME ¼*kích thước hàng đợi = 1s * 50 = 50s; Kích thước hàng đợi Queue/DropTail/PriQueue = 50.

Processing time = thời gian tìm đường đi trong bảng định tuyến

Do đó thời gian từ lúc node độc hại phản hồi cho tới khi node đầu tiên nhận được sẽ nhỏ hơn TimeMin.

2.4.4. Cài đặt giao thức cải tiến ids-AODV

Chi tiết về việc cài đặt được tôi trình bày trong *phụ lục, ở cuối luận văn.*

2.5. Đề xuất cải tiến giao thức bảo mật PHR-AODV

2.5.1. Ý tưởng

Giao thức PHR-AODV lưu trữ tối đa số đường đi từ node nguồn tới node đích, tuy nhiên trong trường hợp cấu hình mạng có nhiều node tham gia, số node độc hại không nhiều thì việc lưu trữ này là không cần thiết, tốn tài nguyên, tốn thời gian tính toán lưu trữ. Gói tin dữ liệu được gửi đi qua quá nhiều đường cũng có thể dẫn tới mất gói nhiều hơn do tắc nghẽn.

2.5.2 Cải tiến phr-AODV

- Duy trì số đường đi từ nguồn tới đích đúng bằng số 2 *node độc hại tham gia cấu hình mạng +1, khi số lượng node độc hại tăng lên số đường đi cũng tăng lên để giảm số lượng gói tin dữ liệu chuyển qua node độc hại
  Routes = 2*n+1 với n: số node độc hại tham gia mô phỏng
- Đường đi được chọn được lấy theo thứ tự được tìm thấy trong danh sách bảng định tuyến chứa đường đi hợp lệ.

2.6 Tổng kết chương 2

Chương 2 tập trung trình bày cách thức hoạt động của giao thức định tuyến AODV, từ việc hiểu và nắm rõ quá trình hoạt động của giao thức nội dụng của chương tập trung vào việc phân tích cách thức tấn công lỗ đen, các ý tưởng và giải thuật được đưa ra nhằm chống tấn công lỗ đen. Kiến thức chủ yếu được thể hiện ở việc mô phỏng lại hai ý tưởng chống tấn công lỗ đen ids-AODV và phr-AODV.

Thêm vào đó, tác giả cũng đưa vào 3 ý tưởng cải tiến của cá nhân nhằm nâng cao tỉ lệ truyền tin thành công cho các biến thể của giao thức AODV. Hai ý tưởng cải tiến đưa ra cho biến thể ids-AODV và 1 ý tưởng cho biến thể phr-AODV. Trên cơ sở ý tưởng của các biến thể và cải tiến cho các biến thể, tác giả sẽ trình bày cách thức mô phỏng lại trên công cụ NS-2 trong chương 3.

CHƯƠNG 3. ĐÁNH GIÁ BẰNG MÔ PHỎNG CÁC ĐỀ XUẤT CHỐNG TẤN CÔNG KIỂU LỖ ĐEN VÀO GIAO THỨC AODV

Chương 3 tập trung vào việc mô phỏng lại các ý tưởng tấn công và chống tấn công vào giao thức định tuyến AODV trên bộ công cụ mô phỏng NS-2. Các độ đo hiệu năng được được ra nhằm đánh giá hiệu quả tấn công cũng như chống tấn công của các biến thể giao thức AODV.

3.1. Cài đặt mô phỏng AODV và chống tấn công kiểu lỗ đen vào AODV

3.1.1. Giới thiệu bộ lập lịch sự kiện NS-2

NS2 (Network Simulation 2) là bộ mô phỏng đa giao thức thuộc dự án nghiên cứu và phát triển của các nhà nghiên cứu tại trường đại học UC Berkeley từ năm 1989 phục vụ cho các nghiên cứu về làm việc mạng. NS2 có chứa một thư viện phong phú các mô hình cho việc dùng trong nghiên cứu mạng. Khác với các chương trình mô phỏng riêng lẻ được phát triển cho các mục đích nghiên cứu cụ thể, ví dụ các chương trình mô phỏng ATM hoặc PIM muticast, khả năng mô phỏng của NS2 bao gồm các mạng có dây và không dây. Bên cạnh đó, NS2 là phần mềm mã nguồn mở được quan tâm và phát triển bởi nhiều nhà nghiên cứu thuộc các viện, trường đại học và các trung tâm nghiên cứu.[2][3]

Trong hỗ trợ mô phỏng mạng AD HOC, phần mã mô phỏng lớp vật lý, lớp liên kết và lớp MAC được xây dựng bởi nhóm Mornach trường CMU. Với các hỗ trợ mô phỏng này, NS2 được dùng rộng rãi trong nghiên cứu mạng AD HOC. Đặc biệt, việc mở rộng các chức năng mô phỏng mạng AD HOC của NS2 nằm trong mối quan tâm và chủ đề thảo luận của nhóm làm việc MANET, tổ chức IETF.

Về thiết kế chung, NS2 là bộ mô phỏng vận hành theo các sự kiện rời rạc (Discrete Event-Drivent Simulator). Để thực hiện điều đó, NS2 sử dụng một hàng đợi chứa các sự kiện được sắp xếp theo thứ tự thời gian xảy ra.

Dưới khía cạnh này, NS2 là bộ thông dịch ngôn ngữ kịch bản hướng đối tượng OTcl bao gồm bộ lập lịch sự kiện, các thư viện đối tượng thành phần mạng và các thư viện hàm thiết đặt mạng. Chương trình mô phỏng được viết bằng ngôn ngữ OTcl khởi tạo bộ lập lịch, thiết lập cấu hình mạng với các đối tượng mạng và các hàm thiết đặt mạng.

Các nguồn truyền thông được điều khiển phát và dừng thông qua bộ lập lịch sự kiện. Sự kiện trong NS2 được đánh dấu bởi các packetID cho mỗi gói tin với thời gian được lập lịch và con trỏ tới đối tượng thao tác với sự kiện. Bộ lập

lịch sự kiện quản lý thời gian mô phỏng và cho thi hành các sự kiện trong hàng đợi sự kiện tại thời điểm được lập lịch và gọi tới thành phần mạng thích hợp. Ví dụ, một thành phần chuyển mạch mạng (switch) được mô phỏng với thời gian trễ chuyển mạch là 20 Ms, gói tin qua chuyển mạch sẽ được làm trễ 20 Ms trước khi chuyển tới thành phần chuyển mạch để phát ra đường ra thích hợp.

3.1.2. Mô phỏng không dây

Các thành phần mạng chính được dùng để cấu trúc nên tầng giao thức cho mỗi nút di động gồm có kênh (channel), giao tiếp mạng (network interface), mô hình phát sóng vô tuyến (radio propagation model), các giao thức MAC, hàng đợi giao diện (interface queue), lớp liên kết (link layer), mô hình giao thức phân giải địa chỉ ARP và thành phần định tuyến (routing agent**).[2][3][4]**

- **Mô phỏng lớp vật lý thực:**

Các mô hình phát sóng quyết định khoảng cách gói tin có thể được truyền đi trong không khí. Sự suy yếu của sóng vô tuyến giữa các ăng ten ở gần mặt đất được mô hình là 1/r2 (r: khoảng cách giữa các ăng ten) với khoảng cách ngắn và và 1/r4 với các khoảng cách xa. Điểm giao giữa hai khoảng cách được gọi là khoảng cách tham chiếu (reference distance). Khoảng cách này thông thường là 100 m đối với các ăng ten 1,5m có độ lợi thấp (low-gain), ngoài trời, hoạt động ở dải băng tần 1-2GHz.

Đặc tả của mô hình phát sóng trong NS2 tương tự như giao tiếp sóng vô tuyến Lucent's WaveLAN với tốc độ bit danh định có thể đạt tới 2,5Mb/s và phạm vi truyền sóng vô tuyến là 250m. Các mô hình cũng thể hiện độ trễ truyền, các ảnh hưởng và cảm nhận sóng mang.

- **Mô phỏng lớp MAC:**

Lớp liên kết của bộ mô phỏng NS2 cài đặt hoàn chỉnh chuẩn giao thức MAC của IEEE 802.11 DCF(Distributed Coordination Function). Các chức năng của lớp MAC được cài đặt bao gồm phát hiện xung đột, phân mảnh, biên nhận và đặc biệt có khả năng phát hiện các lỗi truyền (transmision error). 802.11 là giao thức CSMA/CA. Việc tránh xung đột được thực hiện bằng việc kiểm tra kênh truyền trước khi sử dụng. Nếu kênh rỗi, nút có thể bắt đầu gửi. Nếu không, nút phải đợi một khoảng thời gian ngẫu nhiên trước khi kiểm tra lại. Mỗi lần cố gắng không thành công, giải thuật rút lui theo hàm mũ được sử dụng. Vấn đề trong môi trường không dây là thiết bị đầu cuối ẩn (hidden terminal). Việc khắc phục được thực hiện bằng cơ chế tránh xung đột CA cùng với lược đồ biên nhận tích cực (RTS/CTS). 802.11 cũng hỗ trợ tiết kiệm năng lượng và bảo mật. Các gói tin

được lưu trong bộ đệm khi hệ thống ở trạng thái nghỉ (sleep); bảo mật được cung cấp bởi giải thuật WEP xác thực và mã hóa. Một trong các đặc điểm quan trọng nhất của 802.11 là chế độ AD HOC cho phép xây dựng các mạng WLAN không có cơ sở hạ tầng.

- **Mô phỏng giao thức phân giải địa chỉ ARP:**

Giao thức ARP dịch địa chỉ IP thành địa chỉ phần cứng MAC. Việc này được thực hiện trước khi gói tin được gửi tới lớp MAC.

- **Hàng đợi giao diện:**

Mỗi nút có hàng đợi các gói tin đang chờ để được truyền bởi giao diện mạng. Hàng đợi được cài đặt là DropTail và có khả năng chứa 50 gói tin (giá trị mặc định).

- **Giao diện sóng vô tuyến:**

Đây là mô hình phần cứng thực sự chuyển gói tin vào kênh. Giao diện sóng vô tuyến được mô hình với các mức năng lượng và lược đồ điều biến.

- **Năng lượng truyền**:

Bán kính vùng thu phát sóng phụ thuộc vào dạng ăngten và công suất phát.

3.1.3. Tổng quan quá trình mô phỏng

Quá trình bao gồm việc tạo hai tệp đầu vào cho NS2:
- Tệp ngữ cảnh (scenario file): là file kịch bản mô tả dạng di chuyển của các nút.
- Tệp truyền thông (communication file): là file kịch bản mô tả các truyền thông trong mạng.

Khi chương trình mô phỏng được chạy, bộ mô phỏng ghi nhận các hoạt động mạng tại các lớp trong một file vết (trace file). Trước khi mô phỏng, các tham số cần cho việc ghi tệp vết được lựa chọn. Tệp vết sau đó có thể được duyệt và phân tích để xác định các tham số cần tính tóan. Các kết quả tính tóan, phân tích có thể dùng làm dữ liệu cho các chương trình vẽ như gnuplot, xgraph, tracegraph. Tệp vết cũng có thể được dùng để trực quan hóa việc chạy mô phỏng bằng ad-hockey hoặc NAM.

3.1.4. Cách thức viết giao thức định tuyến mở rộng trong NS2

Tất cả các giao thức định tuyến trong bộ phần mềm ns2 đều được đặt trong thư mục:

~/ns-allinone-2.35/Ns-2.35

Giao thức mới được viết cần phải khai báo trong các file :

- cmu_trace.cc
- cmu-trace.h
- priqueue.cc
- packet.h
- ns-packet.tcl
- ns-lib.tcl
- ns-agent.tcl
- ns-mobilenode.tcl
- makefile

Triển khai giao thức blackholeAODV:

Tạo thư mục với tên blackholeAODV trong thư mục ns-2.35;

Tạo các file:

- *blackholeaodv.cc*: chứa các hàm thiết lập định tuyến cho giao thức;
- *blackholeaodv.h*: file thư viện chứa các biến dùng chung và thiết lập cấu hình cho giao thức;
- *blackholeaodv.tcl*: định nghĩa các agent trong tcl để có thể dùng ngôn ngữ tcl mô phỏng giao thức;
- *blackholeaodv_rqueue.cc*: chứa các hàm thiết lập cấu trúc hàng đợi cho node trong giao thức;
- *blackholeaodv_rqueue.h*: thư viện hỗ trợ choblackholeaodv_rqueue.cc;
- *aodv_packet.h*: file định nghĩa gói tin aodv.

Giao thức blackholeaodv cũng gửi gói tin aodv như giao thức AODV (khó phát hiện khi bắt các gói tin trong mạng). Vì thế, giao thức blackholeaodv vẫn sử dụng aodv_packet.h

3.1.5 Thực hiện giao thức tấn công blackhole AODV

- Khi node được khai bảo sử dụng giao thức định tuyến blackholeaodv để thực hiện được cần tạo ra 1 agent blackhole. Thiện sửa đổi “\tcl\lib\ ns-lib.tcl”.

```
;# ============================================
blackholeAODV {
set ragent [$self create-blackholeaodv-agent $node]
}
```

Simulator instproc create-blackholeaodv-agent { node } {

set ragent [new Agent/blackholeAODV [$node node-addr]]

$self at 0.0 "$ragent start" # start BEACON/HELLO Messages

$node set ragent_ $ragent

return $ragent

Tấn công blackhole giả mạo gói tin RREP với max seq# và min hopcount

;# ==

sendReply(rq->rq_src, // IP Destination
1, // Hop Count
index, // Dest IP Address
4294967295, // Highest Dest Sequence Num
MY_ROUTE_TIMEOUT, // Lifetime
rq->rq_timestamp); // timestamp

3.1.6 Mô phỏng tấn công và chống tấn công với ngôn ngữ kịch bản tcl

- vị trí của các node trong mô phỏng được thiết lập bằng lệnh ./setdest

- Nguồn sinh lưu lượng CBR với:

Packet size: 512 bytes

Data rates: 10 Kbits

```
set val(chan) Channel/WirelessChannel ;# Channel Type
set val(prop) Propagation/TwoRayGround ;# radio-propagation model
set val(netif) Phy/WirelessPhy ;# network interface type
set val(mac) Mac/802_11 ;# MAC type
set val(ifq) Queue/DropTail/PriQueue ;# interface queue type
set val(ll) LL ;# link layer type
set val(ant) Antenna/OmniAntenna ;# antenna model
set val(ifqlen) 150 ;# max packet in ifq
set val(rp) AODV ;# routing protocol
```

Hình 2. 4 Cấu hình cho node mạng

```
for {set i 0} {$i < $val(nnaodv)} {incr i} {
set node_($i) [$ns_ node]
$node_($i) random-motion 0 ; # disable random motion
}
# The last node behave as blackhole
$ns_ node-config -adhocRouting blackholeAODV
for {set i $val(nnaodv)} {$i < $val(nn)} {incr i} {
set node_($i) [$ns_ node]
$node_($i) random-motion 0 ; # disable random motion
$ns_ at 0.01 "$node_($i) label \"blackhole node\""
}
```

Hình 2. 5 Tạo các node bị tấn công blackhole

3.2. Đánh giá hiệu quả chống tấn công kiểu lỗ đen của giao thức idsAODV

3.2.1 Các độ đo hiệu năng

- Tỉ lệ chuyển gói tin thành công : Packet Delivery Ratio.
- Độ trễ đầu cuối – đầu cuối trung bình: Avarage End to end Delay.

3.2.2 Kịch bản và cấu hình mô phỏng

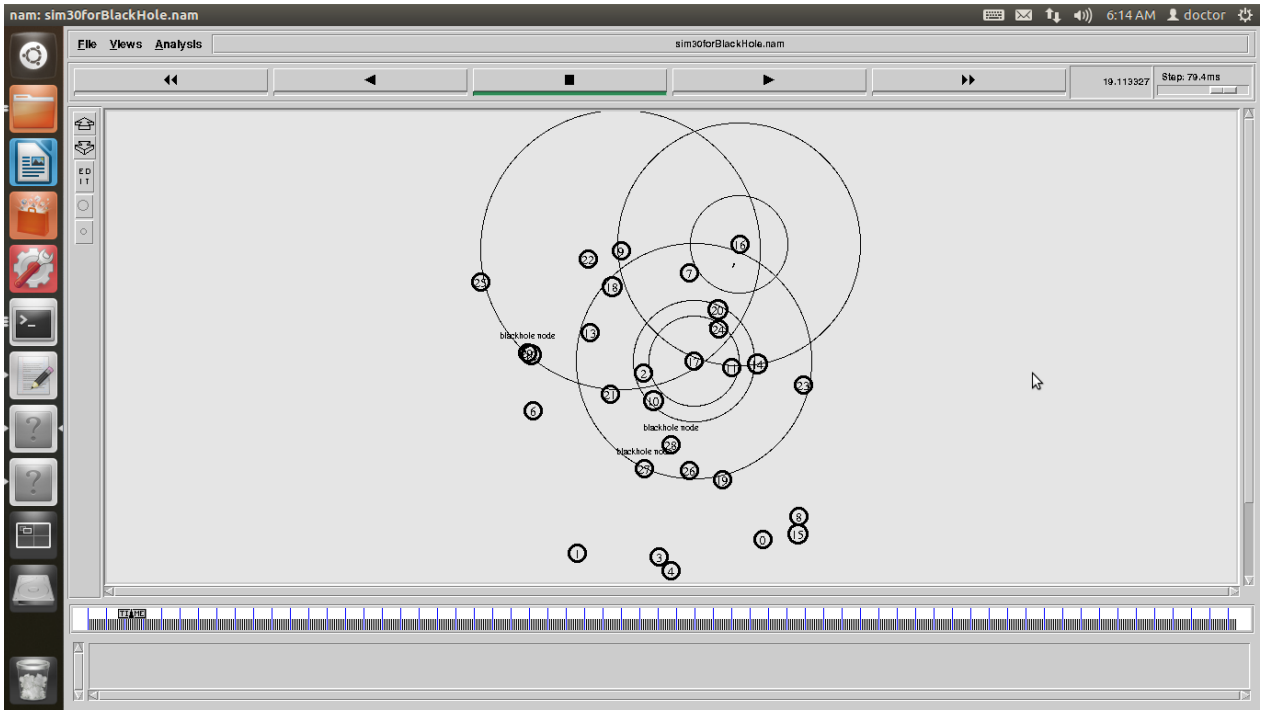
✓ Kịch bản với 20, 30, 40, 50 node tham gia mô phỏng

Thông số	Giá trị
Cấu hình chung	
Khu vực địa lý	750x750 m
Tổng số nút	20, 30, 40, 50
Vùng thu phát sóng	500m
Cấu hình truyền dữ liệu	
Nguồn sinh lưu lượng	CBR
Số kết nối	8
Kích thước gói tin	512 bytes

Tốc độ phát gói	4 gói/s

Bảng 3. 1 Kịch bản với 20, 30, 40, 50 node tham gia mô phỏng chống tấn công blackhole với giao thức ids-AODV

3.2.3 Kết quả mô phỏng



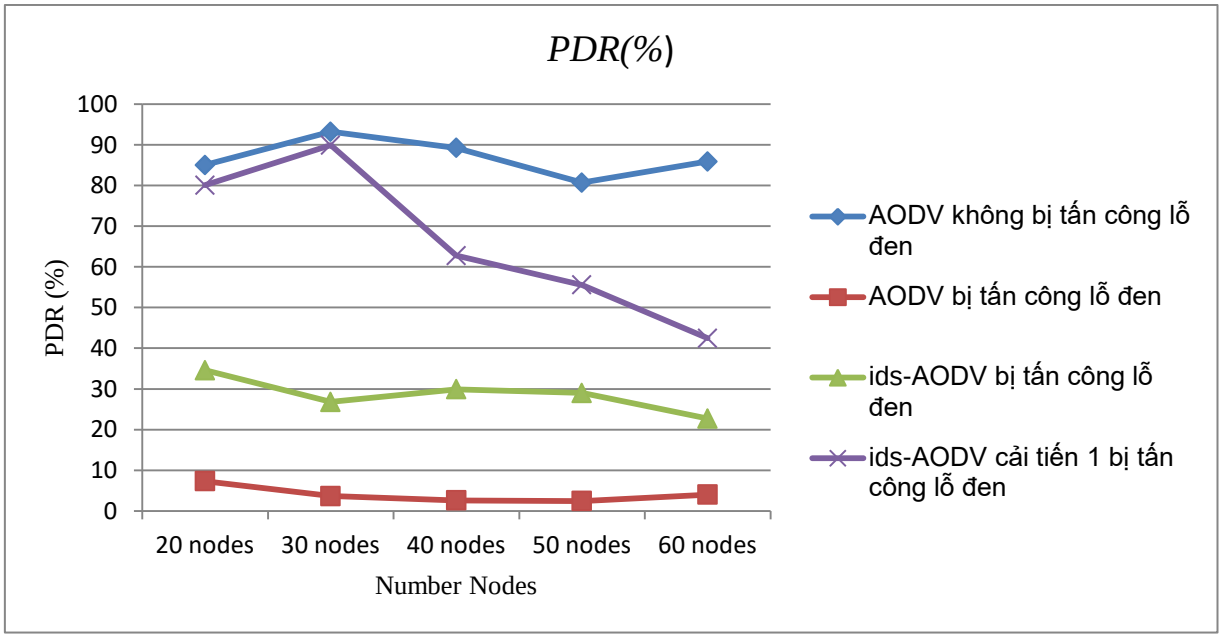
Hình 3. 1 Mô phỏng tấn công blackhole với giao thức ids-AODV

Số node	AODV không bị tấn công lỗ đen PDR(%)	AODV bị tấn công lỗ đen PDR(%)	ids-AODV bị tấn công lỗ đen PDR(%)	ids-AODV cải tiến 1 bị tấn công lỗ đen PDR(%)
20 nodes	85.04	7.34	34.59	80.05
30 nodes	93.21	3.68	26.79	89.92
40 nodes	89.23	2.62	29.89	62.75
50 nodes	80.70	2.45	29	55.54
60 nodes	85.89	4.03	22.72	42.44

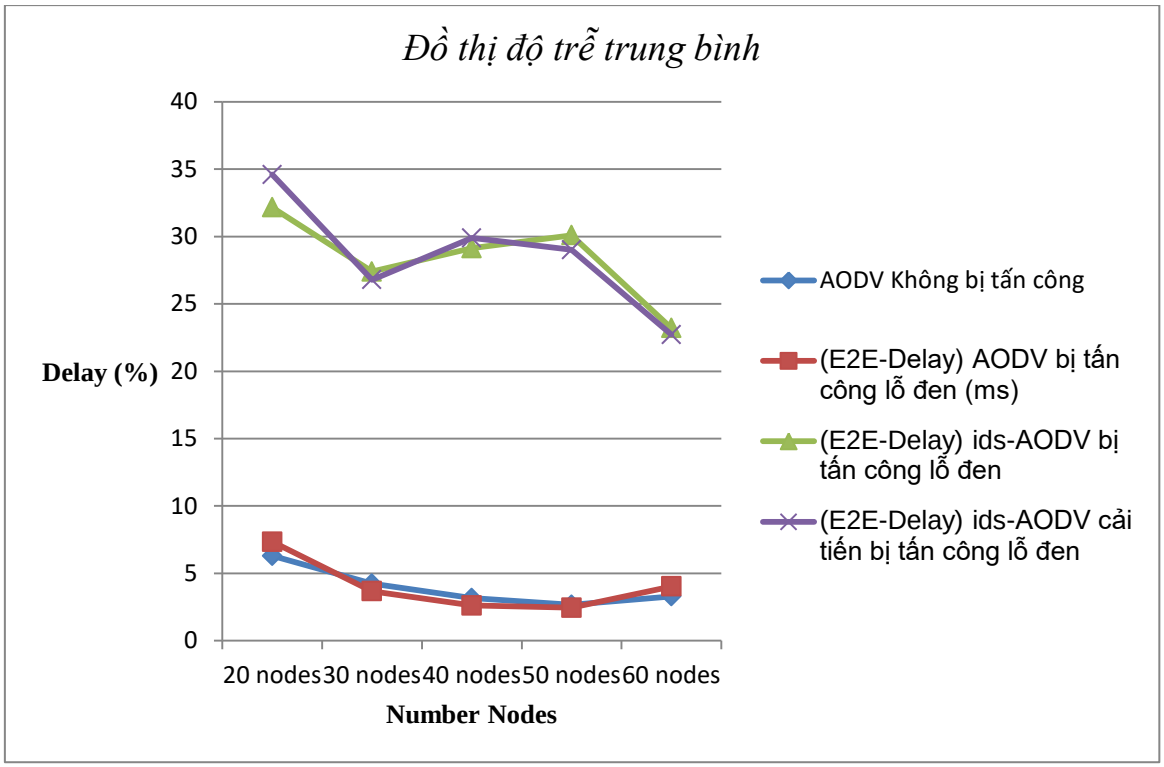
Bảng 3. 2 Tỉ lệ phân phát gói tin thành công giao thức ids-AODV, ids-AODV cải tiến 1, AODV bị tấn công lỗ đen

Số node	(E2E-Delay) AODV Không bị tấn công	(E2E-Delay) AODV bị tấn công lỗ đen (ms)	(E2E-Delay) ids-AODV bị tấn công lỗ đen	(E2E-Delay) ids-AODV cải tiến bị tấn công lỗ đen
20 nodes	6.30	7.34	32.16	34.59
30 nodes	4.23	3.68	27.40	26.79
40 nodes	3.16	2.62	29.13	29.89
50 nodes	2.67	2.45	30.09	29.01
60 nodes	3.29	4.03	23.20	22.72

Bảng 3. 3 Độ trễ trung bình (end to end delay) ids-AODV, ids-AODV cải tiến 1, AODV trước sự tấn công blackhole



Hình 3.1: Đồ thị PDR so sánh giữa các giao thức ids-AODV, AODV



Hình 3. 2 Đồ thị End to End delay giao thức ids-AODV

3.3. Đánh giá hiệu quả chống tấn công kiểu lỗ đen của giao thức PHR-AODV

3.3.1 Các độ đo hiệu năng

- Tỉ lệ chuyển gói tin thành công (Packet Delivery Ratio);
- Độ trễ đầu cuối – đầu cuối trung bình (Avarage End to end Delay);
- Số lượng gói tin điều khiển (Number control message).

3.3.2 Kịch bản và cấu hình mô phỏng

Kịch bản 1: Tăng số node mô phỏng, số node độc hại không đổi = 1

Thông số	Giá trị
Khu vực địa lý	750x750 m
Tổng số nút	20, 30, 40, 50
Vùng thu phát sóng	500m
Nguồn sinh lưu lượng	CBR
Số kết nối	8

Kích thước gói tin	512 bytes
Tốc độ phát gói	4 gói/s

Bảng 3. 4 Kịch bản với nhiều node tham gia mô phỏng chống tấn công lỗ đen của giao thức phr-AODV, phr-AODV cải tiến, số lượng các node độc hại =1

Kịch bản 2: Số node mô phỏng = 30, số node độc hại thay đổi 1, 2, 3, 5, 10

Thông số	Giá trị
Khu vực địa lý	750x750 m
Tổng số nút	30
Tổng số nút độc hại	1, 2, 3, 5, 10
Vùng thu phát sóng	500m
Nguồn sinh lưu lượng	CBR
Số kết nối	8
Kích thước gói tin	512 bytes
Tốc độ phát gói	4 gói/s

Bảng 3. 5 Kịch bản với nhiều node tham gia mô phỏng chống tấn công lỗ đen của giao thức phr-AODV, phr-AODV cải tiến, số lượng các node độc hại thay đổi

3.3.3 Kết quả mô phỏng

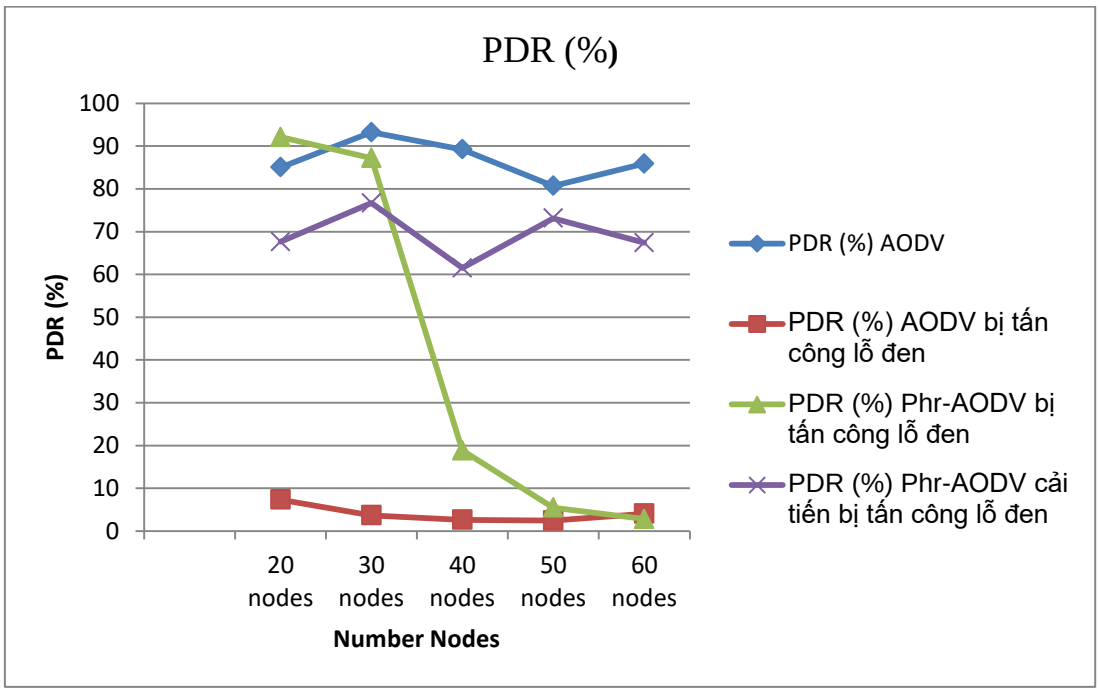
Kết quả mô phỏng kịch bản 1:

Số node	PDR (%) AODV không bị tấn công	PDR (%) AODV bị tấn công lỗ đen	PDR (%) Phr-AODV bị tấn công lỗ đen	PDR (%) Phr-AODV cải tiến bị tấn công lỗ đen
20 nodes	85.04	7.34	92.09	67.63
30 nodes	93.21	3.68	87.21	76.67
40 nodes	89.23	2.62	18.81	61.50
50 nodes	80.70	2.45	5.46	73.12
60 nodes	85.89	4.03	2.82	67.39

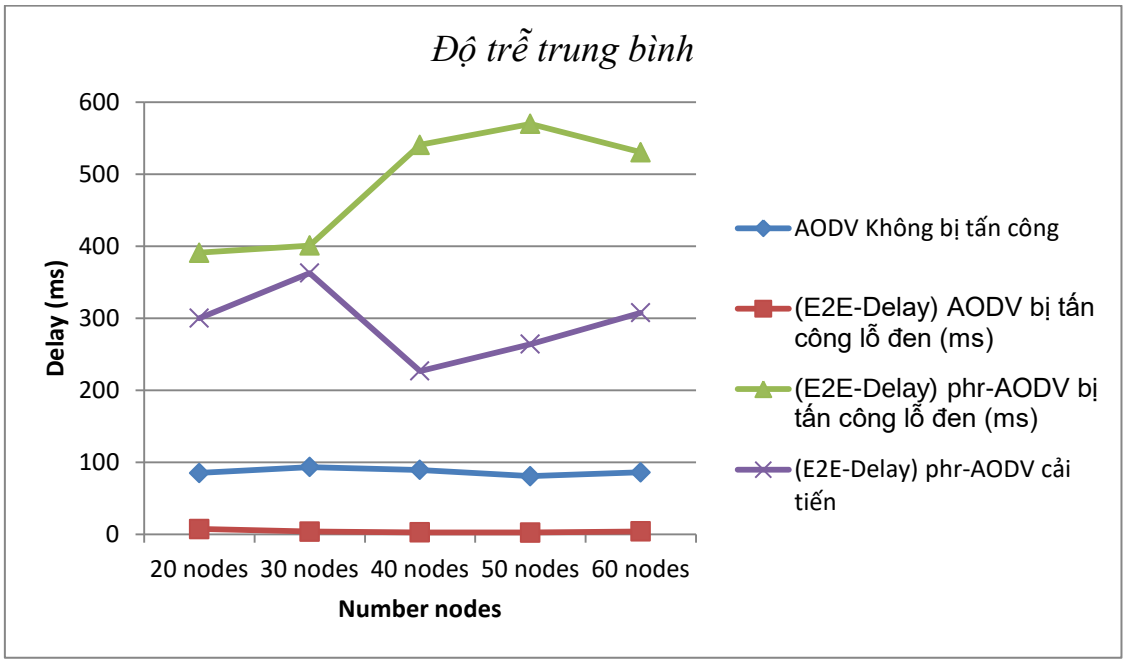
Bảng 3. 6 Tỉ lệ chuyển gói tin thành công giao thức phr-AODV

Số node	AODV Không bị tấn công	(E2E-Delay) AODV bị tấn công lỗ đen (ms)	(E2E-Delay) phr-AODV bị tấn công lỗ đen (ms)	(E2E-Delay) phr-AODV cải tiến
20 nodes	85.04	7.34	390.89	299.94
30 nodes	93.21	3.68	400.89	362.60
40 nodes	89.23	2.62	540.56	226.72
50 nodes	80.70	2.45	569.73	263.86
60 nodes	85.89	4.03	530.45	307.61

Bảng 3. 7 Độ trễ trung bình giao thức phr-AODV



Hình 3. 3 Tỉ lệ chuyển gói tin thành công trước tấn công black hole giao thức phr-AODV



Hình 3. 4 Độ trễ trung bình trước tấn công black hole giao thức phr-AODV

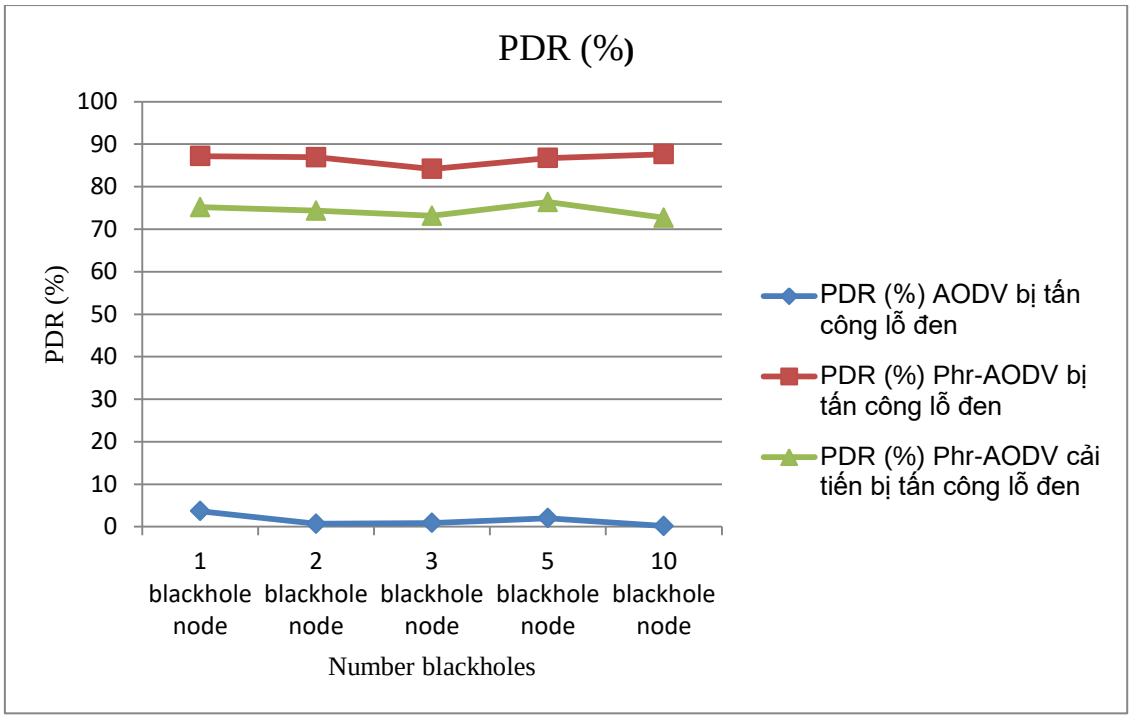
Kết quả mô phỏng kịch bản 2:

Kịch bản (30 node tham gia mô phỏng, số lượng blackhole node thay đổi)	PDR (%) AODV bị tấn công lỗ đen	PDR (%) Phr-AODV bị tấn công lỗ đen	PDR (%) Phr-AODV cải tiến bị tấn công lỗ đen
1 blackhole node	3.68	87.21	75.17
2 blackhole node	0.69	86.94	74.34
3 blackhole node	0.89	84.17	73.15
5 blackhole node	2.01	86.74	76.40
10 blackhole node	0.17	87.64	72.68

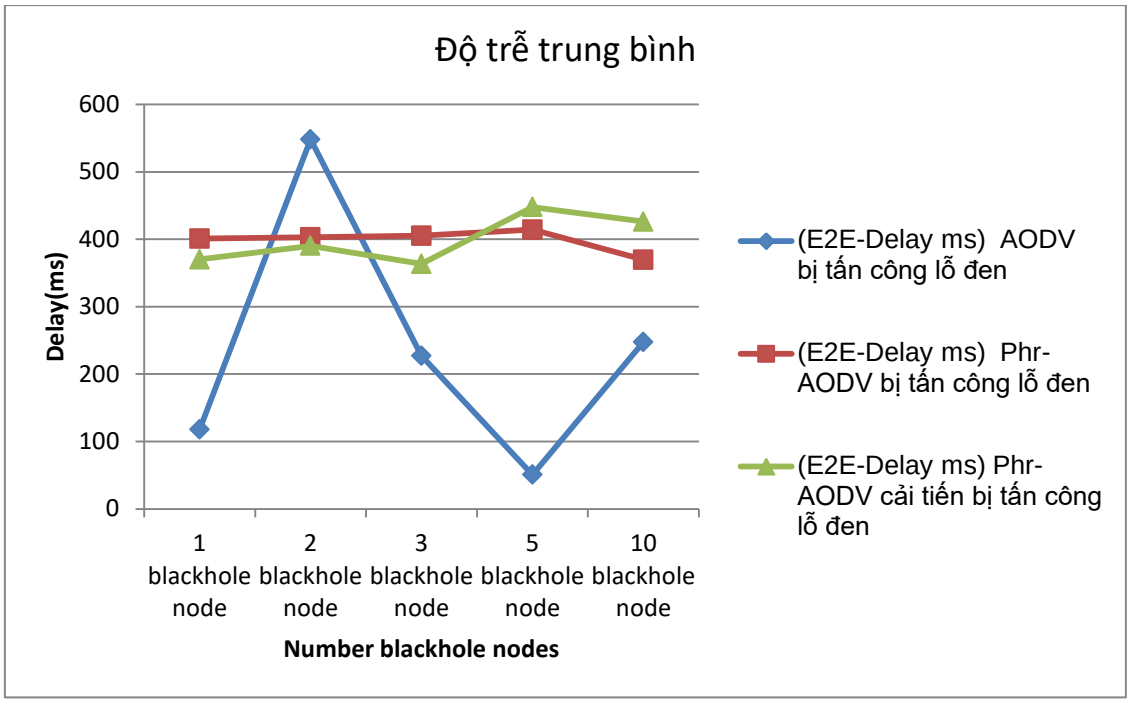
Bảng 3. 8 Tỉ lệ chuyển gói tin thành công giao thức phr-AODV, phr-AOD V cải tiến và giao thức AODV trước sự tấn công của nhiều node blackhole

Kịch bản (30 node tham gia mô phỏng, số lượng blackhole node thay đổi)	(E2E-Delay ms) AODV bị tấn công lỗ đen	(E2E-Delay ms) Phr-AODV bị tấn công lỗ đen	(E2E-Delay ms) Phr-AODV cải tiến bị tấn công lỗ đen
1 blackhole node	117.65	400.89	370.12
2 blackhole node	548.04	402.72	390.23
3 blackhole node	227.2	405.14	363.51
5 blackhole node	50.76	414.28	447.64
10 blackhole node	247.48	369.62	426.16

Bảng 3. 9 Độ trễ trung bình của giao thức phr-AODV, phr-AODV cải tiến , AODV trước sự tấn công của nhiều node blackhole



Hình 3. 5 Tỉ lệ chuyển gói tin thành công trước tấn công nhiều node black hole giao thức phr-AODV



Hình 3. 6 Độ trễ trung bình trước tấn công nhiều node black hole giao thức phr-AODV

3.4. Tổng kết chương 3

Từ các kết quả mô phỏng của giao thức aodv, idsAODV, phr-aodv và AODV trước tấn công blackhole có thể nhận thấy rằng:

- Nhìn vào kết quả hình 3.1, 3.3, 3.5, giao thức cơ bản AODV trước tấn công blackhole cho kết quả tỉ lệ chuyển gói tin thành công rất thấp. Do trong cấu hình mạng có 1 hoặc nhiều node bị tấn công blackhole đã xóa bỏ gói tin dữ liệu thay vì chuyển tới đích nên kết quả tỉ lệ chuyển thành công tới đích rất thấp
- Nhìn vào kết quả hình 3.1 giao thức ids-AODV và cải tiến ids-AODV cho kết quả tỉ lệ gói tin chuyển thành công tương đối ổn định mặc dù vẫn giảm khi số node trong mạng tăng lên. Bởi vì khi số node trong mạng tăng đồng nghĩa với số node trung gian có đường đi tới đích cũng tăng theo, giao thức ids-AODV cần tính toán để loại bỏ RREP không hợp lệ vì thế việc lưu trữ tính toán để lựa chọn các gói tin RREP có thể gây timeout cho quá trình khám phá tuyến dẫn tới tỉ lệ chuyển thành công gói tin giảm xuống

- Nhìn vào kết quả hình 3.3, 3.5 giao thức phr-aodv cho kết quả tỉ lệ gói tin chuyển thành công cao hơn đáng kể khi số lượng node tham gia mô phỏng nhỏ (20 - 30 node), tuy nhiên khi số node trong mạng tăng lên > 30 node tỉ lệ chuyển gói tin thành công tới đích giảm xuống đáng kể. Bởi vì khi số node tăng lên phr-AODV duy trì tối đa các đường đi có thể từ nguồn tới đích để gửi dữ liệu do vậy khi số đường đi có thể quá nhiều việc tìm, lưu trữ cũng như gửi dữ liệu qua nhiều đường như vậy có thể gây tắc nghẽn mạng do có quá nhiều gói tin điều khiển được sinh ra để thiết lập và duy trì đường do vậy làm PDR của toàn mạng giảm xuống. Giao thức cải tiến phr-AODV duy trì số lượng đường đi có thể bằng 2* số node độc hại + 1 nên số đường đi phụ thuộc vào số node độc hại, nhiều node độc hại thì duy trì nhiều đường hơn để hạn chế đường đi qua node độc hại làm mất gói tin
- Nhìn vào hình 3.4, 3.6 giao thức phr-aodv cho độ trễ trung bình cao hơn đáng kể so với giao thức AODV truyền thống. Bởi vì giao thức phr-AODV duy trì tối đa số đường đi có thể tới đích nên độ trễ đầu cuối tăng do các gói tin phải đi qua nhiều đường để có thể tới được đích. Cải tiến giao thức phr-AODV cho kết quả độ trễ đầu cuối nhỏ hơn do duy trì đường đi ít hơn.
- Nhìn vào hình 3.2 giao thức ids-AODV và cải tiến ids-AODV cho độ trễ đầu cuối gần bằng nhau nhưng cao hơn đáng kể so với giao thức AODV truyền thống bởi lẽ khi các node di chuyển khiến đường đi được khám phá bị bẻ gãy, lúc này giao thức ids-AODV và AODV đều tiến hành thủ tục khám phá tuyến mới. Giao thức ids-AODV khám phá tuyến mất nhiều thời gian để tính toán và duy trì đườn đi hơn do đó làm độ trễ đầu cuối tăng lên
- Có thể nói, để tăng được tỉ lệ gói tin được gửi tới đích thành công trước sự tấn công blackhole giao thức ids-AODV và phr-AODV đều phải trả giá bằng sự tiêu tốn tài nguyên hệ thống. Với ids-AODV phải có bộ đệm RREP và với phr-AODV duy trì nhiều đường đi tới đích dẫn tới sinh ra rất nhiều gói tin điều khiển để thiết lập và duy trì đường định tuyến. Cải tiến 1 giao thức ids-AODV mặc dù hạn chế được rủi ro loại bỏ mất RREP hợp lệ tuy nhiên nếu kẻ tấn công không tấn công với số sequence number khác max sequence number và số hopcount khác 1 khả năng nhận biết để loại bỏ RREP của node độc hại là không khả thi. Cải tiến phr-AODV mặc duy trì số đường đi phụ thuộc vào số lượng node blackhole tuy nhiên nếu cấu hình mạng bao gồm nhiều node, các node lại di chuyển liên tục việc

duy trì thêm đường đi tới đích cũng gây tiêu tốn tài nguyên hệ thống một cách đáng kể.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết luận

Trong luận văn này tôi đã nghiên cứu giao thức AODV trong mạng MANET, thực hiện tấn công và chống tấn công. Tôi đã đạt được các kết quả như sau:

Về kiến thức chung:

1. Tổng quan về mạng không dây, mạng MANET.
2. Phân tích giao thức định tuyến AODV
3. Nghiên cứu giao thức mở rộng cho giao thức AODV

Về thực nghiệm:

1. Đề xuất cải tiến giao thức ids-AODV, phr-AODV. Giao thức ids-AODV được cải tiến để chỉ loại bỏ RREP nhận được có sequence number cực đại và số hopcout =1, đề xuất nhưng chưa mô phỏng sử dụng thời gian tối thiểu để nhận RREP giúp tăng khả năng nhận biết node độc hại. Giao thức phr-AODV được cải tiến để chỉ duy trì số đường từ nguồn tới đích bằng 2*số node độc hại+1 giúp giảm chi phí mà vẫn giữ được tỉ lệ chuyển gói tin tới đích thành công
2. Nghiên cứu, tìm hiểu về các công cụ mô phỏng.
3. Cài đặt bộ mô phỏng NS-2.35, cài đặt giao thức mở rộng cho giao thức AODV
4. Tiến hành các thử nghiệm trên các kịch bản mô phỏng khác nhau, đưa ra các kết quả cụ thể về hiệu năng mạng trước sự tấn công blackhole

2. Hướng phát triển của luận văn

- Xây dựng giao thức cải tiến giao thức AODV nhằm nâng cao hiệu năng mạng, thực hiện mô phỏng cải tiến thứ 2 ids-AODV
- Nghiên cứu giải pháp bảo mật dữ liệu cho các giao thức khác trong mạng MANET

TÀI LIỆU THAM KHẢO

Tiếng Việt

1. PGS.TS. Nguyễn Đình Việt (2010), Bài giảng Mạng và Truyền số liệu nâng cao.
2. PGS .TS. Nguyễn Đình Việt (2010), Bài giảng Đánh giá hiệu năng mạng.
3. Nguyễn Minh Nguyệt, Nguyễn Đình Việt, "Đánh giá các giao thức định tuyến trong mạng AD HOC không dây", *Kỷ yếu Hội thảo quốc gia "Một số vấn đề chọn lọc của Công nghệ thông tin",* Đà Nẵng, 08/2004.
4. Nguyen Manh Ha, Nguyen Minh Nguyet, Nguyen Dinh Viet. *Simulation-based evaluation of routing protocols and Internet access in mobile wireless ad hc networks*, Giải Nhất - Hội nghị nghiên cứu khoa học sinh viên và học viên cao học, Khoa Công nghệ, ĐHQGHN, 05/2004.

Tiếng Anh

5. Reverse AODV (R-AODV) Routing Protocol in Mobile Ad hoc Networks,
   C. Kim, E. Talipov, B. Ahn, The 2006 IFIP International Conference On Embedded and Ubiquitous Computing" (EUC'06), LNCS 4097, pp. 522 – 531, Seoul, Korea, August 2006
6. *Path Hopping Based on Reverse AODV for Security C*. Kim, E. Talipov, B. Ahn, *The 2006 IFIP International Conference On Embedded and Ubiquitous Computing*" (EUC'06), LNCS 4097, pp. 522 – 531, Seoul, Korea, August 2006.
7. *S. Dokurer, Y. M. Erten and E. A. Can, "Performance Analysis of Ad-Hoc Networks under Black Hole Attacks," Proceeding from SECON'07: IEEE Southeast Conference, Richmond, 22-25 March 2007, pp. 148-153.*
8. https://rahimanuddin.wordpress.com/2010/03/01/maodv-simulation-in-ns-2/
9. Perkin, C.E., Royer, E.M.: Ad-hoc on demand distance vector routing, *In: Proceedings of 2nd IEEE Workshop on Mobile Computer Systems and Applications*, New Orleans (1999)
10. Abolhasan, M., Wysocki, T., Dutkiewicz, E.: *A review of routing protocols for mobile ad hoc networks*, Elsevier, Amsterdam (2004)
11. Mahmood, R.A., Khan, A.I.: A Survey on Detecting Black Hole Attack in AODVbased Mobile Ad Hoc Networks, *In: International Symposium on High Capacity Optical Networks and Enabling Technologies* (2007)

12. Perkin, C.E.: Ad hoc On Demand Distance Vector (AODV) Routing. Internet draft, draft-ietf-manetaodv-02.txt (November 1988)
13. Kumar, V.: Simulation and Comparison of AODV and DSR Routing Pro
14. Xing, F., Wang, W.: Understanding Dynamic Denial of Service Attacks in Mobile Ad hoc Networks. *In: IEEE Military Communication conference, MILCOM* (2006)
15. Shalini Jain, Mohit Jain, Himanshu Kandwal, *Algorithm for Detection and Prevention of Cooperative Black and Gray Hole Attacks in Mobile Ad Hoc Networks, International Journal of Computer Applications Volume 1* (2010)
16. Abderrahmane Baadache, Ali Belmehdi, *Avoiding Black hole and Cooperative Black Protocols for Mobile Ad hoc networks using Certificate Chaining, International Journal of Computer Applications* (0975 – 8887) Volume 1 – No. 12 (2010)
17. Suman Deswal and Sukhbir Singh, *Implementation of Routing Security Aspects in AODV, International Journal of Computer Theory and Engineering*, Vol. 2, No. 1 February, 2010
18. Sanzgiti, K., Dahill, B., Levine, B.N., Shields, C., Elizabeth, M., Belding-Royer: A secure Routing Protocol for Ad hoc networks. *In: Proceedings of the 10th EEE International Conference on Network Protocols*, ICNP 2002 (2002)
19. Hu, Y.-C., Johnson, D.B., Perrig, A.: SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad hoc Networks. *In: Proc. 4th IEEE Workshop on Mobile Computing Systems and Applications, Callicoon, NY, pp. 3–13 (June 2002) hole Attacks in Wireless Ad hoc Networks (IJCSIS) International Journal of Computer Science and Information Security,* Vol. 7, No. 1, 2010
20. E. A .Mary Anita, V. Vasudevan, *Black Hole Attack Prevention in Multicast Routing*
21. Loay Abusalah, Ashfaq Khokhar, and Mohsen Guizani, “A Survey of Secure Mobile Ad Hoc Routing Protocols,” *IEEE communications surveys & tutorials*, Vol. 10, no. 4, pp. 78- 93, 2008.
22. Arshad, J.; Azad, M.A.; , "Performance Evaluation of Secure on-Demand Routing Protocols for Mobile Ad-hoc Networks," *Sensor and Ad Hoc Communications and Networks*, SECON '06. 2006 3rd Annual IEEE Communications Society on , vol.3, no., pp.971-975, 28-28 Sept. 2006.

PHỤ LỤC CÀI ĐẶT CÁC GIAO THỨC

Tùy biến các file:

;# ==

- cmu-trace.cc

```
//RAODV
void
CMUTrace::format_raodv(Packet *p, int offset)
{
        struct hdr_raodv *ah = HDR_RAODV(p);
        struct hdr_raodv_request *rq = HDR_RAODV_REQUEST(p);
        struct hdr_raodv_reply *rp = HDR_RAODV_REPLY(p);
        switch(ah->ah_type) {
        case RAODVTYPE_RREQ:
              if (pt_->tagged()) {
                 sprintf(pt_->buffer() + offset,
                         "-raodv:t %x -raodv:h %d -raodv:b %d -raodv:d %d "
                         "-raodv:ds %d -raodv:s %d -raodv:ss %d "
                         "-raodv:c REQUEST ",
                         rq->rq_type,
                        rq->rq_hop_count,
                        rq->rq_bcast_id,
                        rq->rq_dst,
                        rq->rq_dst_seqno,
                        rq->rq_src,
                        rq->rq_src_seqno);
              } else if (newtrace_) {
                 sprintf(pt_->buffer() + offset,
                      "-P aodv -Pt 0x%x -Ph %d -Pb %d -Pd %d -Pds %d -Ps %d -
Pss %d -Pc REQUEST ",
```

```
                rq->rq_type,
                rq->rq_hop_count,
                rq->rq_bcast_id,
                rq->rq_dst,
                rq->rq_dst_seqno,
                rq->rq_src,
                rq->rq_src_seqno);
        } else {
            sprintf(pt_->buffer() + offset,
                "[0x%x %d %d [%d %d] [%d %d]] (REQUEST)",
                rq->rq_type,
                rq->rq_hop_count,
                rq->rq_bcast_id,
                rq->rq_dst,
                rq->rq_dst_seqno,
                rq->rq_src,
                rq->rq_src_seqno);
        }
        break;
case RAODVTYPE_RQREP:
        if (pt_->tagged()) {
            sprintf(pt_->buffer() + offset,
                "-raodv:t %x -raodv:h %d -raodv:b %d -raodv:d %d "
                "-raodv:ds %d -raodv:s %d "
                "-raodv:c REVERSE ",
                rp->rp_type,
                rp->rp_hop_count,
                rp->rp_bcast_id,
                rp->rp_dst,
```

```
                rp->rp_dst_seqno,
                rp->rp_src);
        } else if (newtrace_) {
            sprintf(pt_->buffer() + offset,
                "-P aodv -Pt 0x%x -Ph %d -Pb %d -Pd %d -Pds %d -Ps %d -
Pc REVERSE ",
                rp->rp_type,
                rp->rp_hop_count,
                rp->rp_bcast_id,
                rp->rp_dst,
                rp->rp_dst_seqno,
                rp->rp_src);
        } else {
            sprintf(pt_->buffer() + offset,
                "[0x%x %d %d [%d %d] [%d]] (REVERSE)",
                rp->rp_type,
                rp->rp_hop_count,
                rp->rp_bcast_id,
                rp->rp_dst,
                rp->rp_dst_seqno,
                rp->rp_src);
        }
        break;
    case RAODVTYPE_HELLO:
  case RAODVTYPE_RERR:
        if (pt_->tagged()) {
            sprintf(pt_->buffer() + offset,
                "-raodv:t %x -raodv:h %d -raodv:d %d -radov:ds %d "
                "-raodv:l %f -raodv:c %s ",
```

```
                rp->rp_type,
                rp->rp_hop_count,
                rp->rp_dst,
                rp->rp_dst_seqno,
                rp->rp_lifetime,
                (rp->rp_type == RAODVTYPE_RERR ? "ERROR" :
                 "HELLO"));
        } else if (newtrace_) {
                sprintf(pt_->buffer() + offset,
                   "-P raodv -Pt 0x%x -Ph %d -Pd %d -Pds %d -Pl %f -Pc
%s ",
                        rp->rp_type,
                        rp->rp_hop_count,
                        rp->rp_dst,
                        rp->rp_dst_seqno,
                        rp->rp_lifetime,
                        (rp->rp_type == AODVTYPE_RERR ? "ERROR" :
                         "HELLO"));
        } else {
                sprintf(pt_->buffer() + offset,
                        "[0x%x %d [%d %d] %f] (%s)",
                        rp->rp_type,
                        rp->rp_hop_count,
                        rp->rp_dst,
                        rp->rp_dst_seqno,
                        rp->rp_lifetime,
                        (rp->rp_type == AODVTYPE_RERR ? "ERROR" :
                         "HELLO"));
        }
```

```
            break;
        default:
    #ifdef WIN32
            fprintf(stderr,
                "CMUTrace::format_raodv: invalid RAODV packet type\n");
    #else
            fprintf(stderr,
                "%s: invalid RAODV packet type\n", __FUNCTION__);
    #endif
            abort();
        }
    }
;# ============================================
```

- cmu-trace.h

```
void    format_aodv(Packet *p, int offset);
    //RAODV
    void    format_raodv(Packet *p, int offset);
;# ============================================
```

- priqueue.cc

```
void
PriQueue::recv(Packet *p, Handler *h)
{
    struct hdr_cmn *ch = HDR_CMN(p);

    if(Prefer_Routing_Protocols) {
        switch(ch->ptype()) {
        case PT_DSR:
        case PT_MESSAGE:
```

```
                case PT_TORA:
                case PT_AODV:
                case PT_RAODV:
                case PT_idsAODV:
                case PT_blackholeAODV:
                case PT_WFRP:
                case PT_AOMDV:
                case PT_MDART:
                        recvHighPriority(p, h);
                        break;

                default:
                        Queue::recv(p, h);
                }
        }
        else {
                Queue::recv(p, h);
        }
    }
;# ============================================
```

- packet.h

```
    class p_info {
    public:
        p_info()
        {
                initName();
        }
        const char* name(packet_t p) const {
                if ( p <= p_info::nPkt_ ) return name_[p];
```

```
        return 0;
}
static bool data_packet(packet_t type) {
        return ( (type) == PT_TCP || \
                (type) == PT_TELNET || \
                (type) == PT_CBR || \
                (type) == PT_AUDIO || \
                (type) == PT_VIDEO || \
                (type) == PT_ACK || \
                (type) == PT_SCTP || \
                (type) == PT_SCTP_APP1 || \
                (type) == PT_HDLC \
               );
}
static packetClass classify(packet_t type) {
        if (type == PT_DSR ||
           type == PT_MESSAGE ||
           type == PT_TORA ||
           type == PT_PUMA ||
           type == PT_AODV ||
           type == PT_blackholeAODV ||
           type == PT_idsAODV ||
           type == PT_RAODV ||
          type == PT_WFRP ||
           type == PT_MDART)
                return ROUTING;
        if (type == PT_TCP ||
           type == PT_TELNET ||
           type == PT_CBR ||
```

```
                type == PT_AUDIO ||
                type == PT_VIDEO ||
                type == PT_ACK ||
                type == PT_SCTP ||
                type == PT_SCTP_APP1 ||
                type == PT_HDLC)
                    return DATApkt;
            if (pc_)
                    return pc_->classify(type);
            return UNCLASSIFIED;
        }
;# ============================================
```

- ns-packet.tcl

```
# Mobility, Ad-Hoc Networks, Sensor Nets:
    AODV        # routing protocol for ad-hoc networks
    # WFRP patch
    WFRP
    # RAODV patch
    RAODV
    # idsAODV patch
    idsAODV
    # backholeAODV patch
    blackholeAODV
    Diffusion   # diffusion/diffusion.cc
    IMEP        # Internet MANET Encapsulation Protocol, for ad-hoc networks
      MIP       # Mobile IP, mobile/mip-reg.cc
    Smac # Sensor-MAC
    TORA        # routing protocol for ad-hoc networks
```

```
        MDART      # routing protocol for ad-hoc networks
        # AOMDV patch
        AOMDV
;# ============================================
```

- ns-lib.tcl

```
    if {$rtAgentFunction_ != ""} {
            set ragent [$self $rtAgentFunction_ $node]
        } else {
            switch -exact $routingAgent_ {
              DSDV {
                    set ragent [$self create-dsdv-agent $node]
              }
              DSR {
                    $self at 0.0 "$node start-dsr"
              }
              AODV {
                    set ragent [$self create-aodv-agent $node]
              }
              RAODV {
                    set ragent [$self create-raodv-agent $node]
              }
              idsAODV {
                    set ragent [$self create-idsaodv-agent $node]
              }
              blackholeAODV {
                    set ragent [$self create-blackholeaodv-agent $node]
              }
;# ============================================
```

- ns-agent.tcl

```
Agent/AODV instproc init args {
        $self next $args
}
Agent/AODV set sport_   0
Agent/AODV set dport_   0
Agent/idsAODV instproc init args {
        $self next $args
}
Agent/idsAODV set sport_   0
Agent/idsAODV set dport_   0
Agent/blackholeAODV instproc init args {
        $self next $args
}
Agent/blackholeAODV set sport_   0
Agent/blackholeAODV set dport_   0
Agent/RAODV instproc init args {
        $self next $args
}
Agent/RAODV set sport_   0
Agent/RAODV set dport_   0
;# ============================================
```

- ns-mobilenode.tcl

```
# Special processing for idsAODV
    set idsaodvonly [string first "idsAODV" [$agent info class]]
    if {$idsaodvonly != -1 } {
            $agent if-queue [$self set ifq_(0)]   ;# ifq between LL and MAC
    }
```

```
        # Special processing for blackholeAODV
        set blackholeaodvonly [string first "blackholeAODV" [$agent info class]]
        if {$blackholeaodvonly != -1 } {
                $agent if-queue [$self set ifq_(0)]   ;# ifq between LL and MAC
        }
        # Special processing for RAODV
        set raodvonly [string first "RAODV" [$agent info class]]
        if {$raodvonly != -1 } {
                $agent if-queue [$self set ifq_(0)]   ;# ifq between LL and MAC
        }
;# ============================================
```

- makefile

```
aodv/aodv_logs.o aodv/aodv.o \
aodv/aodv_rtable.o aodv/aodv_rqueue.o \
raodv/raodv_logs.o raodv/raodv.o \
raodv/raodv_rtable.o raodv/raodv_rqueue.o \
idsaodv/idsaodv_logs.o idsaodv/idsaodv.o \
idsaodv/idsaodv_rtable.o idsaodv/idsaodv_rqueue.o \
blackholeaodv/blackholeaodv_logs.o blackholeaodv/blackholeaodv.o \
blackholeaodv/blackholeaodv_rtable.o
blackholeaodv/blackholeaodv_rqueue.o \
;# ============================================
set val(chan) Channel/WirelessChannel ;#Channel Type
set val(prop) Propagation/TwoRayGround ;# radio-propagation model

set val(netif) Phy/WirelessPhy ;# network interface type
set val(mac) Mac/802_11 ;# MAC type
set val(ifq) Queue/DropTail/PriQueue ;# interface queue type
set val(ll) LL ;# link layer type
set val(ant) Antenna/OmniAntenna ;# antenna model
```

```
set val(ifqlen) 150 ;# max packet in ifq
set val(nn) 30 ;# total number of mobilenodes
set val(nnaodv) 20 ;# number of AODV mobilenodes
set val(rp) RAODV ;# routing protocol
set val(x) 750 ;# X dimension of topography
set val(y) 750 ;# Y dimension of topography
set val(cstop) 451 ;# time of connections end
set val(stop) 500 ;# time of simulation end
set val(cp) "scenarios/scenforAODV-n30-t500-x750-y750" ;#Connection Pattern
#set val(cc) "scenarios/cbr" ;#CBR Connections
# Initialize Global Variables
set ns_ [new Simulator]
$ns_ use-newtrace
set tracefd [open sim30forBlackHole.tr w]
$ns_ trace-all $tracefd
set namtrace [open sim30forBlackHole.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)
# set up topography object
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
# Create God
create-god $val(nn)
# Create channel #1 and #2
set chan_1_ [new $val(chan)]
set chan_2_ [new $val(chan)]
# configure node, please note the change below.
$ns_ node-config -adhocRouting $val(rp) \
-llType $val(ll) \
-macType $val(mac) \
-ifqType $val(ifq) \
-ifqLen $val(ifqlen) \
-antType $val(ant) \
```

```
-propType $val(prop) \
-phyType $val(netif) \
-topoInstance $topo \
-agentTrace ON \
-routerTrace ON \
-macTrace ON \
-movementTrace ON \
-channel $chan_1_
# Creating mobile AODV nodes for simulation
puts "Creating nodes..."
for {set i 0} {$i < $val(nnaodv)} {incr i} {
set node_($i) [$ns_ node]
$node_($i) random-motion 0 ;#disable random motion
}
# Creating Black Hole nodes for simulation
$ns_ node-config -adhocRouting blackholeAODV
for {set i $val(nnaodv)} {$i < $val(nn)} {incr i} {
set node_($i) [$ns_ node]
$node_($i) random-motion 0 ;#disable random motion
$ns_ at 0.01 "$node_($i) label \"blackhole node\""
}
set god_ [God instance]
source $val(cp)
 set j 0
 for {set i 0} {$i < 18} {incr i} {
 #Create a UDP and NULL agents, then attach them to the appropriate nodes

set udp_($j) [new Agent/UDP]
$ns_ attach-agent $node_($i) $udp_($j)
set null_($j) [new Agent/Null]
$ns_ attach-agent $node_([expr $i + 1]) $null_($j)
set cbr_($j) [new Application/Traffic/CBR]
puts "cbr_($j) has been created over udp_($j)"
```

```
$cbr_($j) set packet_size_ 512
$cbr_($j) set interval_ 1
$cbr_($j) set rate_ 10kb
$cbr_($j) set random_ false
$cbr_($j) attach-agent $udp_($j)
$ns_ connect $udp_($j) $null_($j)
puts "udp_($j) and null_($j) agents has been connected each other"
$ns_ at 1.0 "$cbr_($j) start"
set j [expr $j + 1]
set i [expr $i + 1]
}
# Define initial node position
for {set i 0} {$i < $val(nn) } {incr i} {
$ns_ initial_node_pos $node_($i) 30
}
# CBR connections stops
for {set i 0} {$i < 9 } {incr i} {
$ns_ at $val(cstop) "$cbr_($i) stop"
}

# Tell all nodes when the simulation ends
for {set i 0} {$i < $val(nn) } {incr i} {
$ns_ at $val(stop).000000001 "$node_($i) reset";
}
# Ending nam and simulation
$ns_ at $val(stop) "finish"
$ns_ at $val(stop).0 "$ns_ trace-annotate \"Simulation has ended\""
$ns_ at $val(stop).00000001 "puts \"NS EXITING...\" ; $ns_ halt"
proc finish {} {
global ns_ tracefd namtrace
$ns_ flush-trace
close $tracefd
close $namtrace
```

```
#exec nam sim30forBlackHole.nam &
exit 0
}
puts "Starting Simulation..."
$ns_ run
```