TRƯỜNG ĐẠI HỌC ĐIỆN LỰC

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO MÔN HỌC**

**ĐỀ TÀI:**

**AN TOÀN THÔNG TIN CỦA GIAO THỨC ĐỊNH TUYẾN**

**TRONG MOBILE ADHOC NETWORK**

|  |  |
| --- | --- |
|  |  |
| |  |  | | --- | --- | | **Sinh viên thực hiện** | **: ĐÀM NGỌC SƠN** | | **Giảng viên hướng dẫn** | **: NGÔ HẢI ANH** | | | **Chuyên ngành** | **: QUẢN TRỊ AN NINH MẠNG** | | | **Lớp** | **: D11 QTANM** | | |  |  | | |  |
|  |  |

***Hà Nội, ngày 30 tháng 5 năm 2019***

**LỜI NÓI ĐẦU**

Ngày nay, mạng không dây trong đời sống con người đang ngày càng đóng vị trí quan trọng. Trong số các mạng không dây, mạng ad-hoc được quan tâm một cách đặc biệt. Không giống như mạng có dây truyền thống hay mạng không dây có cơ sở hạ tầng, với việc dễ dàng thiết lập định, chi phí hoạt động thấp, triển khai nhanh chóng và có tính di động cao nên mạng adhoc đang được ứng dụng và có thể làm thay đổi về giao thông, cũng như mọi mặt của đời sống. Cùng với đó việc bảo mật dữ liệu trong mạng ad hoc cũng cần đặc biệt quan tâm cần phát triển các cơ chế bảo mật của các giao thức định tuyến, vì các giao thức định tuyến trong mạng MANET tiềm ẩn nhiều rủi ro dễ bị tấn công, sửa đổi, cũng như giả mạo dễ bị khai thác sơ hở.

**MỤC LỤC**

[CHƯƠNG 1: TỔNG QUAN VỀ MẠNG MANET 1](#_Toc13383280)

[1.1 Sự hiệu quả mạng Ad-Hoc trong đời sống 1](#_Toc13383281)

[1.2 Định Nghĩa 2](#_Toc13383282)

[1.3 Ý tưởng của mạng MANET 3](#_Toc13383283)

[1.4 Lịch sử phát triển 3](#_Toc13383284)

[1.5 Phân loại mạng không dây 3](#_Toc13383285)

[1.5.1 PHÂN LOẠI TRONG MẠNG MANET: 5](#_Toc13383286)

[1.6 Đặc điểm mạng MANET: 8](#_Toc13383287)

[1.7. Kiểu kết nối trong mạng MANET 11](#_Toc13383288)

[1.7.1 Các kiểu kết nối topo mạng Ad-Hoc không dây 11](#_Toc13383289)

[1.8 Ứng dụng 12](#_Toc13383290)

[1.9 Các phương pháp định tuyến trong mạng AdHoc 13](#_Toc13383291)

[1.9.1 Phân loại các giao thức định tuyến trong mạng Ad Hoc 13](#_Toc13383292)

[1.9.1.1 Các giao thức định tuyến theo bảng điều khiển (Table-Driven) 14](#_Toc13383293)

[1.9.1.2 Các giao thức định tuyến theo yêu cầu (On-Demand) 15](#_Toc13383294)

[1.9.1.3 Các giao thức định tuyến lai (hybird Protocol) 15](#_Toc13383295)

[1.9.1.4 Các giao thức định tuyến theo nhận biết vị trí (Location-aware protocol) 16](#_Toc13383296)

[1.9.1.5 Các giao thức định tuyến đa đường (Multipath Protocol) 17](#_Toc13383297)

[1.9.2 Một số loại giao thức phổ biến 17](#_Toc13383298)

[1.9.2.1 Giao thức DSDV (Destination – Sequenced Distance – Vector) 17](#_Toc13383299)

[1.9.2.2 Giao thức DSR (Dynamic Source Routing) 19](#_Toc13383300)

[1.9.2.3 Giao thức AODV (Ad-hoc On-Demand Distance Vector) 21](#_Toc13383301)

[CHƯƠNG 2: NHỮNG VẤN ĐỀ VỀ BẢO MẬT TRONG MẠNG MANET VÀ MỘT SỐ CÁCH THỨC TẤN CÔNG TRONG MẠNG MANET 25](#_Toc13383302)

[2.1. Những vấn đề về an ninh trong mạng MANET 25](#_Toc13383303)

[2.1.1 Thách thức về an ninh trong mạng MANET 25](#_Toc13383304)

[2.1.2. Các yêu cầu về an ninh trong mạng MANET 25](#_Toc13383305)

[2.2 Các phương thức tấn công trong giao thức định tuyến mạng MANET 26](#_Toc13383306)

[2.2.1. Tấn công bằng cách sửa đổi thông tin định tuyến 27](#_Toc13383307)

[2.2.2. Tấn công bằng cách mạo danh 28](#_Toc13383308)

[2.2.3. Tấn công bằng cách tạo ra thông tin bịa đặt 29](#_Toc13383309)

[2.3.4. Một vài kiểu tấn công đặc biệt 29](#_Toc13383310)

[CHƯƠNG 3: TẤN CÔNG BLACKHOLE VỚI GIAO THỨC AODV VÀ MỘT SỐ GIẢI PHÁP PHÒNG CHỐNG TẤN CÔNG BLACKHOLE 32](#_Toc13383311)

[3.1. Lỗ hổng của giao thức AODV 32](#_Toc13383312)

[3.2. Phân loại tấn công kiểu lỗ đen 33](#_Toc13383313)

[3.3. Một số giải pháp phòng chống tấn công lỗ đen trong giao thức AODV 34](#_Toc13383314)

[3.3.1. ARAN (Authenticated Routing for Ad hoc Networks) 34](#_Toc13383315)

[3.3.2. SAODV (Secure Ad hoc On-demand Distance Vector) 36](#_Toc13383316)

[3.3.3. RAODV (Reverse Ad hoc On-demand Distance Vector) 41](#_Toc13383317)

[3.3.4. IDSAODV (Intrusion Detection System Ad hoc On-demand Distance Vector) 43](#_Toc13383318)

[CHƯƠNG 4: SỬ DỤNG NS2 VÀ CÁC CÔNG CỤ LIÊN QUAN ĐỂ ĐÁNH GIÁ HIỆU NĂNG MẠNG VÀ THỰC HÀNH TẤN CÔNG VÀ CHỐNG LẠI BLACKHOLE TRONG GIAO THỨC AODV 45](#_Toc13383319)

[4.2. Bộ mô phỏng NS-2 và cài đặt mô phỏng 45](#_Toc13383320)

[4.2.1. Giới thiệu NS-2 45](#_Toc13383321)

[4.2.2. Các thành phần của bộ chương trình mô phỏng NS-2 46](#_Toc13383322)

[4.2.3. Các chức năng mô phỏng chính của NS-2 47](#_Toc13383323)

[4.2.4. Thiết lập mô phỏng mạng MANET trong NS-2 48](#_Toc13383324)

[4.3. Cài đặt bổ sung các giao thức 53](#_Toc13383325)

[4.3.1. Cài đặt giao thức blackholeAODV mô phỏng tấn công lỗ đen 53](#_Toc13383326)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 54](#_Toc13383327)

[TÀI LIỆU THAM KHẢO 54](#_Toc13383328)

**BẢNG GIẢI THÍCH CÁC THUẬT NGỮ**

**DV** Distance Vector

**LS** Link State

**NS-2** Network Simulator 2

**MAC** Media Access Control

**RTR** Network

**AGT** Transpot

**CBR** Constant Bit Rate

**VBR** Variable bit rate

**s (trong file tr)** sent

**r (trong file tr)** receive

**MANET** Mobile Adhoc NETwork

**AODV** Adhoc On-demand Distance Vector

**DSDV** Destination-Sequenced Distance Vector

**DSR** Dynamic Source Routing

**RAODV** Reverse Adhoc On-demand Distance Vector

**IDSAODV** Intrusion Detection System Adhoc On-demand Distance Vector **SAODV** Secure Adhoc On-demand Distance Vector

**ARAN** Authenticated Routing for Ad hoc Networks

**IP** Internet Protocol

**RREQ** Route Request

**RREP** Route Reply

**R-RREQ** Reverse Route Request

**SN** Sequence Number

**HC** Hop count

**ID** Identification

**MỤC LỤC HÌNH**

[Hình 1. 1 Mô hình mạng không dây có cơ sơ hạ tầng ………………….…………..4](#_Toc13383651)

[Hình 1. 2 Mô hình mạng Ad Hoc ……………………………………….…...……..4](#_Toc13383652)

[Hình 1. 3 Mô hình Singal-hop……………………………………………...……… 5](#_Toc13383653)

[Hình 1. 4 Mô hình Multi-hop ……………………………………………..…….….6](#_Toc13383654)

[Hình 1. 5 Mô hình mạng Hierarchical…………………………………….………. 7](#_Toc13383655)

[Hình 1. 6Mô hình mạng Aggregate ……………………………………………….8](#_Toc13383656)

[Hình 1. 7 Cấu hình mạng động trong mạng MANET……………………..……… 9](#_Toc13383657)

[Hình 1. 8 Hai kiểu topo mạng trong mạng MANET…………………………….. 12](#_Toc13383658)

[Hình 1. 9 Các giao thức định tuyến trong mạng ad-hoc ………………………….14](#_Toc13383659)

[Hình 2. 1 Các kiểu tấn công giao thức định tuyến trong mạng MANET ………..26](#_Toc13383664)

[Hình 2. 2 Ví dụ tấn công bằng cách sửa đổi ……………………………………..27](#_Toc13383665)

[Hình 2. 3 Ví dụ về tấn công bằng cách mạo danh………………………………. 28](#_Toc13383666)

[Hình 2. 4 Tấn công bằng cách tạo ra thông tin bịa đặt …………………………..29](#_Toc13383667)

[Hình 2. 5 Tấn công Wormhole trong mạng MANET ……………………………30](#_Toc13383668)

[Hình 3. 1 Thực hiện tấn công lỗ đen bằng việc giả mạo gói tin RREQ………… 34](#_Toc13384265)

[Hình 3. 2 Thực hiện tấn công lỗ đen bằng việc giả mạo gói tin RREP…………. 34](#_Toc13384266)

[Hình 3. 3 Định dạng của thông điệp định tuyến RREQ (RREP) mở rộng ………38](#_Toc13384267)

[Hình 3. 4 Cách tính hàm băm khi bắt đầu phát sinh RREQ hay RREP …………40](#_Toc13384268)

[Hình 3. 5 Cách tính hàm băm tại nút trung gian …………………………………40](#_Toc13384269)

[Hình 3. 6 Định dạng gói tin RREQ ………………………………………………41](#_Toc13384270)

[Hình 3. 7 Định dạng gói tin R-RREQ ……………………………………………42](#_Toc13384271)

[Hình 3. 8 Sơ đồ định tuyến RAODV …………………………………………….43](#_Toc13384272)

**MỤC LỤC BẢNG**

[Bảng 3. 1 Một số nguy cơ bảo mật trong AODV……………………………….. 32](#_Toc13384612)

[Bảng 3. 2 Các giá trị có thể của trường Hash\_Function………………………… 39](#_Toc13384613)

# TỔNG QUAN VỀ MẠNG MANET

* 1. **Sự hiệu quả mạng Ad-Hoc trong đời sống:**
* **Trong tìm kiếm và cứu nạn:** Khi chúng ta phải đối mặt với những tình huống như động đất, bão, sóng thần, khi đó các cơ sở hạ tầng bị phá hủy. Mạng Ad-Hoc có thể chứng minh sự hữu ích trong việc tìm kiếm và cứu hộ. Mạng adhoc được thiết lập mà không cần cơ sở hạ tầng, và cung cấp thông tin liên lạc giữa các tổ chức cứu trợ để phối hợp hoạt động cứu hộ.

● **Cần tạo một mạng ngay lập tức:** Các cuộc họp không có kế hoạch, giữa các cá nhân tự phát thông tin liên lạc … không thể dựa vào bất kỳ cơ sở hạ tầng. Cơ sở hạ tầng cần lập kế hoạch và hành chính. Sẽ mất quá nhiều thời gian để thiết lập loại này cơ sở hạ tầng; do đó, một kết nối đặc biệt phải được thiết lập.

* **Khi không có cơ sở hạ tầng:**

Trong quân sự: nhiều hoạt động quốc phòng diễn ra tại nhiều địa điểm khác nhau và bí mật nên sẽ không có cơ sở hạ tầng mạng. Sử dụng mạng ad-hoc trong những tình huống như vậy trở nên rất hữu ích và thiết thực.

Các đơn vị khác nhau (bộ binh, hải quân, không quân) tham gia vào các hoạt động quốc phòng cũng cần phải duy trì thông tin liên lạc với nhau.

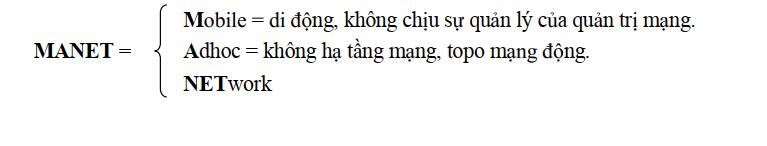
Các Máy bay của không quân bay trong một đội hình có thể thiết lập một mạng Ad-Hoc để giao tiếp với nhau và chia sẻ hình ảnh, dữ liệu với nhau. Một nhóm bộ binh di chuyển cũng có hể sử dụng mạng ad-hoc để giao tiếp với nhau. Việc lạc trong rừng cũng vậy không hề có cơ sở hạ tầng từ trước, vậy việc dùng mạng ad hoc lúc này rất hữu ích.

● **Vùng sâu vùng xa**: tại những nơi xa trung tâm thành phố, nơi có dân cư thưa thớt như ở vùng sâu, vùng xa, việc thiết lập các hệ thống mạng có cơ sở hạ tầng là rất khó khăn và quá tốn kém. Vậy ở những nơi này, giải pháp được đưa ra là sử dụng các mạng vệ tinh hoặc mạng Ad Hoc là tối ưu và hiệu quả nhất.

● **Hiệu quả:** Các dịch vụ được cung cấp bởi cơ sở hạ tầng hiện tại có thể là

quá đắt cho các ứng dụng nhất định. Ví dụ ứng dụng của tôi dùng thỉnh thoảng nó gửi 1 vài tín hiệu lên mạng, việc chúng ta phải đăng ký 1 gói cước với chi phí cao là rất lãng phí và không thiết thực.

**1.2 Định Nghĩa:**



**Mạng tùy biến không dây** (Wireless AdHoc Network - WANET hay Mobile Ad Hoc Network – MANET) là một loại mạng không dây phân tán.

Thuật ngữ ad hoc ngụ ý rằng, mạng này không dựa trên cơ sở hạ tầng có sẵn từ trước như Router, Switch, Hub … Thay vào đó, mỗi thiết bị (nút) tham gia định tuyến mạng bằng cách chuyển tiếp dữ liệu cho nút khác, mà không cần sự hỗ trợ của một sự quản trị trung tâm nào. (tức là các nút mạng hoạt động ngang hàng với nhau).

Mỗi nút trong một mạng vừa như một [máy chủ](https://vi.wikipedia.org/wiki/M%C3%A1y_ch%E1%BB%A7) (*host*) vừa như một [thiết bị định tuyến](https://vi.wikipedia.org/wiki/Router).

Dựa trên kiến trúc mạng AdHoc, các thiết bị cầm tay (điện thoại di động, table, laptop …) và các thiết bi cố định (các trạm cơ sở-BS, các điểm truy cập internet không dây … có thể được kết nối với nhau và tạo thành mạng toàn cầu.

Việc quyết định xem nút nào sẽ thực hiện việc chuyển tiếp [dữ liệu](https://vi.wikipedia.org/wiki/D%E1%BB%AF_li%E1%BB%87u) được dựa trên tình trạng kết nối của mạng.

**1.3 Ý tưởng của mạng MANET:**

Tạo ra 1 mạng vẫn có thể chia sẻ dữ liệu cho nhau mà:

- Sử dụng mạng không dây.

- Không cần cơ sở hạ tầng có sẵn như các trạm BTS …

- Không cần các thiết bị để quản lí tập trung như Router, Switch …

* 1. **Lịch sử phát triển**

Mạng MANET trước đây còn được gọi là mạng vô tuyến gói có khởi nguồn từ mạng PRNet *(*Packet Radio Network*)* và dự án SURAN (Survivable Radio Network) của tổ chức DARPA *(*Defense Advance Research Projects Agency*)* thuộc chính phủ Mỹ.

*-* PRnetlà mạng vô tuyến gói tin đa chặng đầu tiên, vàcung cấp cơ chế  
để quản lý hoạt động trên cơ sở tập trung cũng như phân tán.

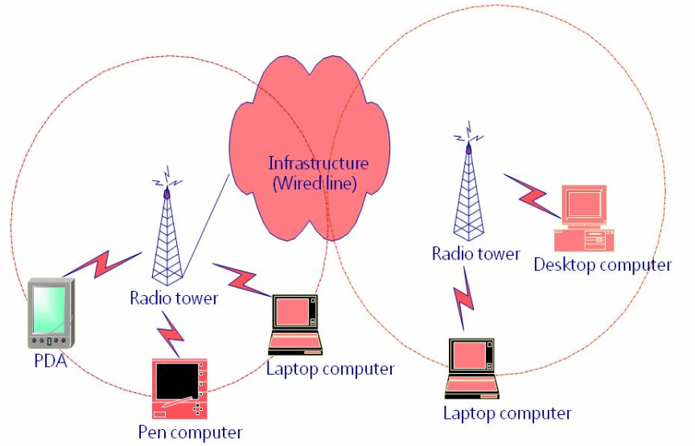
- SUSAN (Adaptive Survivable Network) đã được đề xuất bởi DARPA vào năm 1983 để hỗ trợ một mạng quy mô lớn hơn, mạnh mẽ hơn. Thời gian này, mạng Ad-Hoc đã được sử dụng để mô tả 1 loại mạng như tiêu chuẩn IEEE 802.11.

- Mobile Ad-Hoc Network đã được định nghĩa bởi IETF.

## 1.5 Phân loại mạng không dây

**Mạng di động không dây có thể chia thành hai kiểu mạng**:

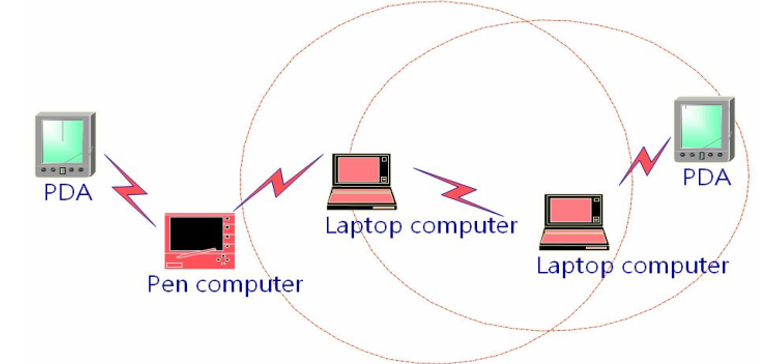
* Mạng có cơ sở hạ tầng (Infrastructure).
* Mạng không hạ tầng (Wireless Ad-Hoc Network).



*Hình 1. 1 Mô hình mạng không dây có cơ sơ hạ tầng*

**Trong mạng có cơ sở hạ tầng**, truyền thông giữa các phần tử mạng phụ thuộc vào sự hỗ trợ của hạ tầng mạng, các thiết bị đầu cuối di động, truyền thông đơn bước không dây qua các điểm truy nhập (các trạm cơ sở) để tới hạ tầng mạng cố định.

Kiểu **mạng không phụ thuộc hạ tầng** còn được gọi chung là các mạng tùy biến di động MANET (Mobile AdHoc Network) hoặc Wireless AdHoc Network là một tập hợp của những node mạng không dây, những node này có thể được thiết lập tại bất kỳ thời điểm, và tại bất cứ nơi nào.

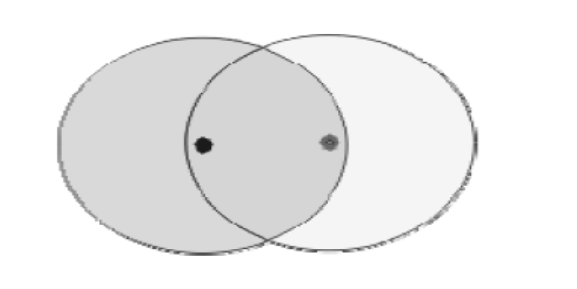


Hình 1. 2 Mô hình mạng Ad Hoc

### 1.5.1 PHÂN LOẠI TRONG MẠNG MANET:

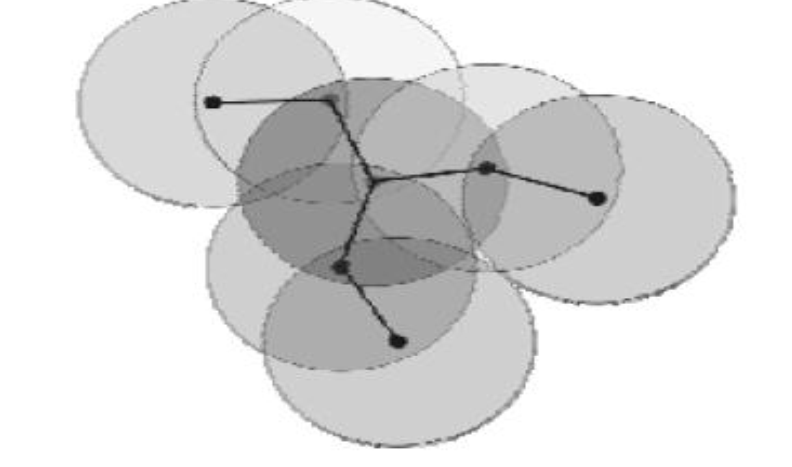
**1.5.1.1 Theo giao thức:**

**•Singal-hop:**Mang Manet định tuyến singal-hop là loại mô hình mạng Ad-Hoc đơn giản nhất.  
Trong đó, tất cả các node đều nằm trong cùng một vùng phủ sóng, nghĩa là các  
node có thể kết nối trực tiếp với nhau mà không cần các node trung gian.  
Mô hình này các node có thể di chuyển tự do nhưng chỉ trong một phạm vi nhất  
định đủ để các node liên kết trực tiếp với các node khác trong mạng.



Hình 1. 3 Mô hình Singal-hop

•**Multi-hop:** Đây là mô hình phổ biến nhất trong mạng MANET, nó khác với mô hình  
trước là các node có thể kết nối với các node khác trong mạng mà có thể  
không cần kết nối trực tiếp với nhau. Các node có thể định tuyến với các  
node khác thông qua các node trung gian trong mạng. Để mô hình này hoạt  
động một cách hoàn hảo thì cần phải có giao thức định tuyến phù hợp với mô  
hình mạng MANET.

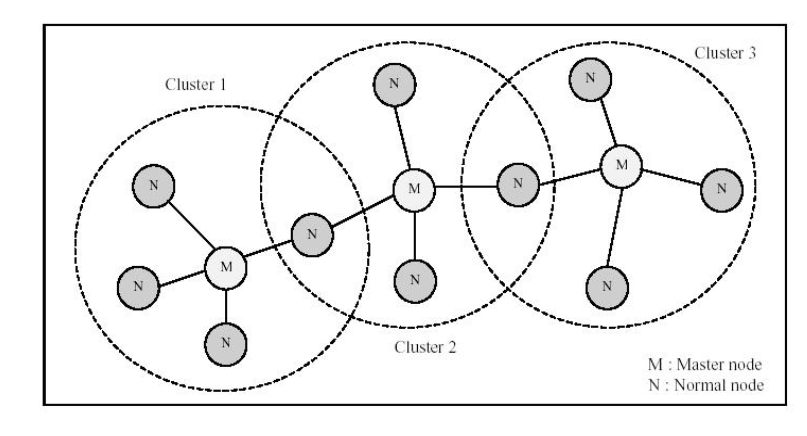


Hình 1. 4 Mô hình Multi-hop

**•Mobile multi-hop:** Mô hình này cũng tương tự với mô hình thứ hai nhưng sự khác biệt ở đây là  
mô hình này tập trung vào các ứng dụng có tính chất thời gian thực: audio,  
video.  
**1.5.1.2. Theo chức năng:**• **Mạng Manet Flat:** Trong kiến trúc này tất cả các node có vai trò ngang hàng với nhau (peer-topeer), và các node đóng vai trò như các router, định tuyến dữ liệu gói trên  
mạng. Trong những mạng lớn, thì cấu trúc Flat không tối ưu hóa việc sử dụng  
tài nguyên băng thông của mạng, vì những tin nhắn điều khiển phải truyền  
trên toàn bộ mạng. Tuy nhiên nó thích hơp trong những topo có các node di  
chuyển nhiều.  
• **Mạng Manet phân cấp (Hierarchical):** Đây là mô hình sử dụng phổ biến nhất.

Trong mô hình này thì mạng chia thành các domain, trong mỗi domain bao gồm một hoặc nhiều cluster mỗi cluster chia thành nhiều node.

Có hai loại node là master node và nomal node. Master node là node quản trị một router có nhiệm vị chuyển dữ liệu của các node trong cluster đến các node trong cluster khác và ngược lại. Nói cách khác nó có nhiệm vụ như một gateway. Normal node là các node nằm trong cùng một cluster. Nó có thể kết nối với các node trong cluster hoặc kết nối với các cluster khác thông qua master node.



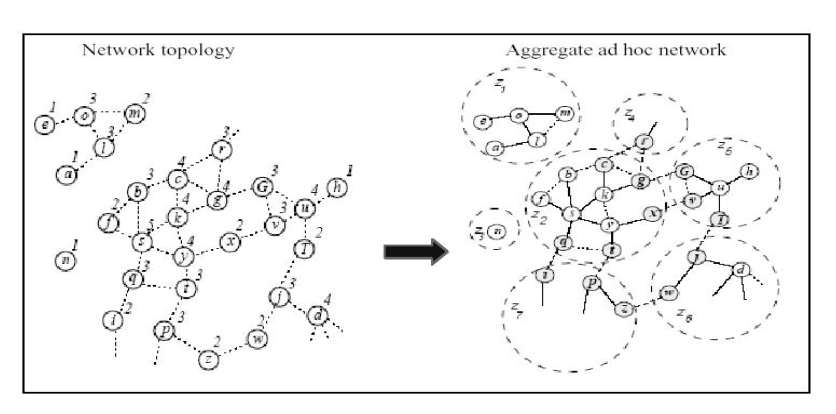
Hình 1. 5 Mô hình mạng Hierarchical

Với các cơ chế trên mạng sử dụng tài nguyên băng thông hiệu quả hơn vì  
các tin nhắn chỉ phải truyền trong 1 cluster. Tuy nhiên việc quản lý tính  
chuyển động của các node trở nên phức tạp hơn. Kiến trúc mạng phân cấp  
thích hợp cho các mạng có tính chuyển động thấp.  
**• Mạng MANET kết hợp (Aggregate):**

Mạng = Zones, Zone = nodes Mỗi node bao gồm hai mức topo:

* Topo mức thấp (node level)
* topo mức cao (zone level)

Mỗi node đặc trưng bởi: node ID và zone ID. Trong một Zone có thể áp dụng kiến trúc đẳng cấp hoặc kiến trúc phân cấp.

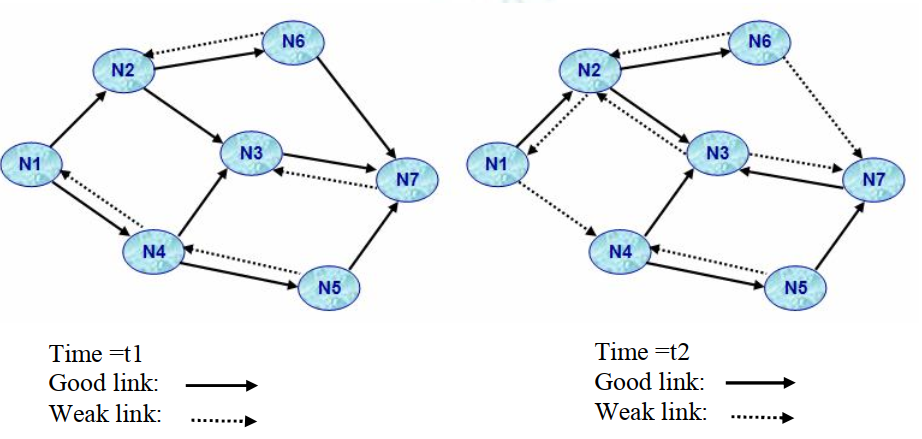


Hình 1. 6Mô hình mạng Aggregate

## 1.6 Đặc điểm mạng MANET:

**-** Mỗi nút di động khác nhau trong mạng MANET đều có những đặc điểm về nguồn năng lượng, bộ phận thu phát sóng khác nhau. Chúng có thể di chuyển về mọi hướng theo các tốc độ khác nhau, do đó ta có thể nhận thấy rõ một số đặc điểm chính của mạng MANET như sau:

**- Cấu hình mạng động:** vì các node là di động, nên cấu trúc mạng có thể thay đổi nhanh và không biết trước và các kết nối giữa các thiết bị đầu cuối có thể thay đổi theo thời gian.



Hình 1. 7 Cấu hình mạng động trong mạng MANET

Hình trên mô tả một mạng Ad hoc đơn giản gồm có 7 nút, các nút mạng được ký hiệu từ N1 đến N7. Nhìn vào hình vẽ chúng ta có thể dễ dàng thấy được: ở thời điểm t1, các liên kết từ N1 đến N2 là những liên kết mạnh (good link), còn các liên kết từ N6 đến N2 ... là những những liên kết yếu (weak link).

Như vậy ở đây một đặc điểm của mạng Ad Hoc đã được thể hiện rõ.

Đó là liên kết giữa 2 nút mạng của mạng có thể không giống nhau dù có chung điểm đầu và điểm cuối. Hiện tượng này được gọi là hiện tượng liên kết hai chiều không đối xứng. Liên kết từ N4 đến N5 là liên kết mạnh nhưng liên kết từ N5 đến N4 lại là liên kết yếu.

Điều này là do vị trí an-ten của 2 nút mạng khác nhau, hoặc do năng lượng phát của các nút mạng trong mạng là khác nhau ...

Tương tự chúng ta cũng có thể thấy N3 có thể nhận tín hiệu từ N2 là một liên kết mạnh nhưng mà N2 lại không thu được tín hiệu từ N3.

Sang đến thời điểm t2, lúc này topo mạng đã thay đổi do các nút di chuyển đến các vị trí khác nhau do đó các liên kết giữa các nút mạng cũng thay đổi theo. Lúc này, N1 chỉ có liên kết mạnh với N2, liên kết với N4 lại là liên kết yếu và N1 không còn thu được tín hiệu từ N4. Liên kết từ N2 đến N3 và N6 lại là liên kết mạnh. Lúc này, N2 cũng có thể thu được tín hiệu từ N3 mặc dù đó là liên kết yếu. Điều này ở thời điểm t1 là không có. Mặt khác chúng ta cũng có thể thấy hai nút mạng nằm trong vùng phủ sóng của nhau có thể truyền thông trực tiếp cho nhau. Ví dụ như trong thời điểm t1, việc truyền thông giữa hai nút mạng N1 và N4 là trực tiếp với nhau.

**-** Ngay cả khi không nằm trong vùng phủ sóng của nhau thì giữa các nút mạng vẫn hoàn toàn có thể thực hiện việc truyền thông với nhau thông qua các nút mạng trung gian. Ví dụ N1 có thể thực hiện truyền dữ liệu cho N7 thông qua nút mạng trung gian N2 và N3, còn N6 có thể truyền dữ liệu cho N1 thông qua nút mạng N2.

**- Thiết bị tự trị đầu cuối (Autonomous terminal):**Trong Manet, mỗi thiết bị di động đầu cuối là một node tự trị. Nó có thể mang chức năng của host và  
router. Bên cạnh khả năng xử lý cơ bản của một host, các node di động này có  
thể chuyển đổi chức năng như một router. Vì vậy thiết bị đầu cuối và chuyển  
mạch là không thể phân biệt được trong mạng Manet.

- **Tự phân chia hoạt động (Distributed operation):**Vì không có hệ thống mạng  
nền tảng cho trung tâm kiểm soát hoạt động của mạng, nên việc kiểm sát và  
quản lý hoạt động của mạng được chia cho các thiết bị đầu cuối. Các node  
trong MANET đòi hỏi phải có sự phối hợp với nhau. (*Vd: như bảo mật  
và định tuyến*).

**Định tuyến đa đường:** Thuật toán định tuyến không dây cơ bản có thể định  
tuyến một chặng hay nhiều chặng dựa vào các thuộc tính liên kết khác nhau và  
giao thức định tuyến. Khi truyền các gói dữ liệu từ một nguồn của nó đến điểm trong phạm vi truyền tải trực tiếp không dây, các gói dữ liệu sẽ được chuyển tiếp qua một hoặc nhiều nút trung gian.

**- Khoảng cách sóng ngắn:** Khoảng cách sóng của các thiết bị di động là rất hạn chế.

**- Năng lượng hạn chế:** Tất cả các thiết bị di động đều sử dụng pin nên khi tham gia vào mạng MANET chúng bị hạn chế về năng lượng, khả năng xử lý của CPU, kích thước bộ nhớ.

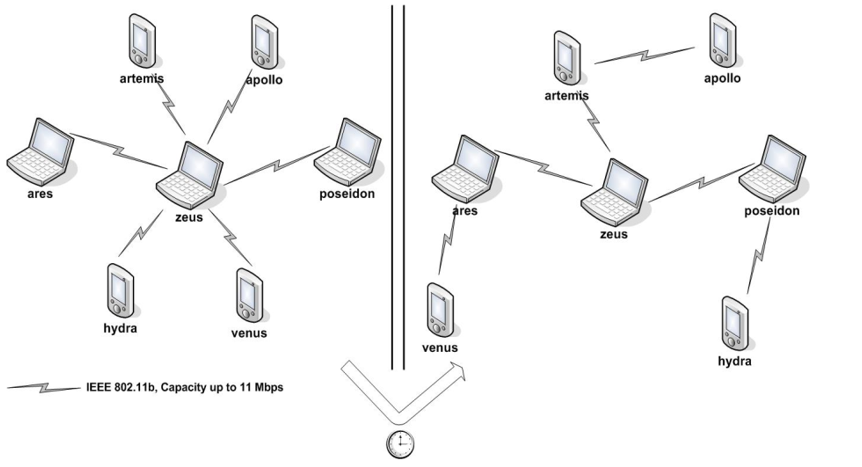
**- Băng thông hạn chế:** Các liên kết không dây có băng thông thấp hơn so với đường truyền cáp và chúng còn chịu ảnh hưởng của sự nhiễu, suy giảm tín hiệu, các điều kiện giao thoa vì thế mà thường nhỏ hơn tốc độ truyền lớn nhất của sóng vô tuyến.

**- Bảo mật yếu:** Đặc điểm của mạng MANET là truyền sóng qua môi trường không khí, điều này khiến cho cơ chế bảo mật kém hơn so vơi môi trường truyền cáp vì nó tiềm ẩn nhiều nguy cơ bị tấn công, nghe lén đường truyền, giả mạo, DoS …

## 1.7. Kiểu kết nối trong mạng MANET:

### 1.7.1 Các kiểu kết nối topo mạng Ad-Hoc không dây

* **Mạng máy chủ di động:** Ở topo này các thiết bị chỉ liên kết với một máy chủ duy nhất. các thiết bị khác liên kết qua máy chủ.
* **Mạng có các thiết bị di động không đồng nhất:** Ở topo này các máy có thể liên kết trực tiếp với nhau trong phạm vi phủ sóng của mình.



Hình 1. 8 Hai kiểu topo mạng trong mạng MANET

## 1.8 Ứng dụng:

* **Giáo dục**: cung cấp cho sinh viên và giảng viên một môi trường thuận tiện để tương tác và việc kết nối sử dụng thiết bị trong phòng sẽ dễ dàng. Cụ thể: những thiết bị như máy chiếu, loa, màn hình được kết nối bằng mạng ad-hoc vừa không phải dây dựa lằng nhằng, vừa đơn giản vừa thao tác nhanh thuận tiện.
* **Trong công nghiệp:** Việc có kết nối có dây dẫn đến sự ngổn ngang và bừa bộn trong một không gian, ra có thể gây mối nguy hiểm cháy nổ. Sử dụng mạng không dây loại bỏ rất nhiều những quan ngại này. Nếu là kết nối mạng không dây ad-hoc sẽ thêm nhiều khía cạnh hấp dẫn và có tính di động. Các thiết bị có thể dễ dàng di chuyển. Đồng thời, nắm bắt các thông tin mới trong cuộc họp của công ty có thể diễn ra mà không cần có mặt tất cả nhân viên trong cùng một phòng.
* **Ứng dụng trong đời sống**: Mạng ad-hoc di động liên kết một mạng lưới đa phương tiện để truyền bá và chia sẻ thông tin. Ví dụ như sử dụng điện thoại/ máy tính để phát wifi cho điện thoại/máy tính khác sử dụng ...
* **Trong giao thông**, nếu các phương tiện đều kết nối mạng manet và nhờ công nghệ an toàn thì các xe hoàn toàn có thể tránh nhau hay tránh tai nạn hay nhường đường cho xe ưu tiên, vì tất cả phương tiện đều di chuyển được => không còn tình trạng tắc đường xảy ra.
* **Trong các lĩnh vực khác** như taxi, thể thao, sân vận động, thuyển và máy bay nhỏ…. Mạng di động ad-hoc tầm ngắn có thể đơn giản hóa việc truyền thông giữa các thiết bị di dộng (PDA, laptop, điện thoại di động). Những dây cáp được thay thể bằng việc kết nối vô tuyến. Mạng di động ad-hoc có thể mở rộng chức năng truy cập Internet như các mạng khác như WLAN, GPRS…. PAN là một lĩnh vực có tiềm năng ứng dụng đầy hứa hẹn của mạng di động ad-hoc phổ biến trong tương lai.

**1.9 Các phương pháp định tuyến trong mạng AdHoc**

Giao thức định tuyến có hai chức năng:

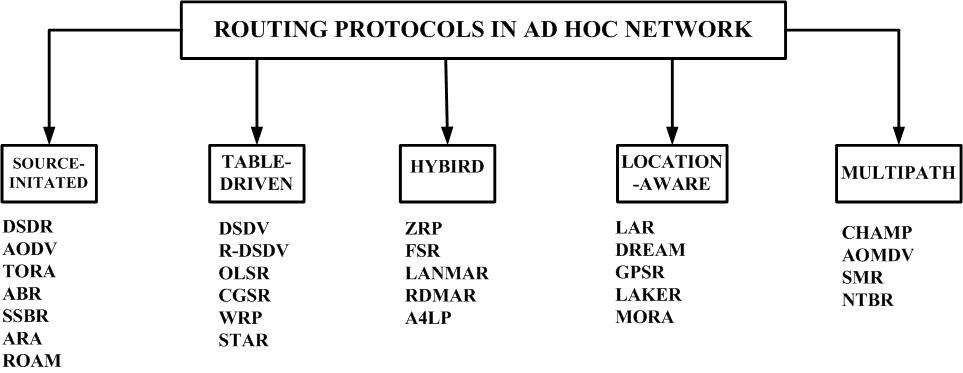
* Tìm, chọn đường tốt nhất.

- Chuyển gói tin đến đúng đích.

Ta sẽ đề cập sâu hơn về việc tìm, chọn đường của các nút.

### 1.9.1 Phân loại các giao thức định tuyến trong mạng Ad Hoc

Giao thức trong mạng Ad hoc được chia làm 5 loại chính. Mỗi loại đều có ưu nhược điểm riêng phù hợp với các trường hợp ứng dụng riêng biệt. Chúng ta sẽ đi sâu vào tìm hiểu kỹ từng loại giao thức.



*Hình 1. 9 Các giao thức định tuyến trong mạng ad-hoc*

### 1.9.1.1 Các giao thức định tuyến theo bảng điều khiển (Table-Driven)

Các giao thức định tuyến theo bảng điều khiển luôn luôn duy trì một bảng định tuyến hay ma trận chứa thông tin cập nhật về các đường đi gồm khoảng cách (số node phải đi qua để đến đích) và thông tin về node kế tiếp trên đường đi đến một node đích bất kỳ. Để cập nhật thông tin về đường đi ngắn nhất mỗi node sẽ thường xuyên trao đổi bảng định tuyến với các node bên cạnh nó. Mỗi node khi nhận được bảng định tuyến của node bên cạnh, nó sẽ tự động tính toán và cập nhật bảng định tuyến của mình. Quá trình gửi cập nhật lại tiếp tục được lặp lại tai node lân cận với mục tiêu giữ các thông tin định tuyến là chính xác nhất có thể. Định tuyến này có ưu điểm là đơn giản và tính toán hiệu quả do thường xuyên được cập nhật. Tuy nhiên nhược điểm của nó là tốn nhiều băng thông và năng lượng do phải thực hiện gửi nhiều bản tin cập nhật. Một nhược điểm nữa là mạng sẽ hội tụ chậm khi topo mạng có thay đổi và khi có lỗi xảy ra.

Một số giao thức trong họ này có thể kể đến là:

* DSDV (Destination-Sequenced Distance-Vector)
* OLSR (Optimized Link State Routing)
* CGSR (Cluster-Head Gateway Switch routing)
* WRP (Wireless routing Protocol)
* STAR (Source-Tree Adaptive Routing)

### 1.9.1.2 Các giao thức định tuyến theo yêu cầu (On-Demand)

Khác với giao thức định tuyến theo bảng điều khiển, giao thức định tuyến theo yêu cầu chỉ thực hiện tìm đường khi có yêu cầu cần cần truyền tin đến một node nào đó ở trong mạng. Khi có yêu cầu tìm đường, node cần truyền sẽ phát vào trong mạng một bản tin quảng bá với nội dung yêu cầu thông tin về node đích cần truyền. Khi một gói tin quảng bá đến được node đích hoặc node trung gian có đường đi đến node đích cần tìm, gói tin hồi đáp sẽ được tạo ra và gửi về node nguồn. Khi node nguồn đã có đường đi tới node đích, nó mới bắt đầu thực hiện truyền các gói tin chứa dữ liệu. Giao thức này có ưu điểm là độ chính xác cao và phản ứng nhanh hơn với sự thay đổi của mạng. Nhưng nhược điểm của nó là thời gian trễ do tìm đường cao và khi các node di chuyển nhanh, lưu lượng dày, các gói tin quảng bá sẽ tăng nhanh số lượng gây nghẽn. Các giao thức định tuyến théo yêu cầu chỉ phù hợp với mạng không dây băng thông rộng, trễ truyền gói nhỏ và lưu lượng rất mỏng.

Một số giao thức trong họ này có thể kể đến là:

* AODV (Ad hoc On-Demand Distance Vector)
* DSR (Dynamic Source Routing)
* TORA (Temporally Ordered Routing Algorithm)
* ABR (Associativity-Based Routing)
* SSBR (Signal Stability-Based Adaptive Routing)

### 1.9.1.3 Các giao thức định tuyến lai (hybird Protocol)

Giao thức định tuyến lai là kết hợp của giao thức định tuyến theo bảng điểu khiên và giao thức định tuyến theo yêu cầu với các ưu điểm của cả hai. Các định tuyến tĩnh được sử dụng ở những phần của mạng mà sự thay đổi không xảy ra thường xuyên. Còn những phần có các node thay đổi liên tục sẽ thực hiện định tuyến theo yêu cầu. Qua đó giao thức đã tạo được cầu nối giữa hai giao thức quan trọng của mạng Ad hoc, thừa hưởng ưu điểm của cả hai, hiệu năng tổng thể do đó cũng được nâng cao đáng kể

Một số giao thức trong họ này có thể kể đến là:

* ZRP (Zone Routing Protocol)
* FSR (Fisheye State Routing)
* LANMAR (Landmark Ad Hoc Routing)
* RDMAR (Relative Distance Micro-Discovery Ad Hoc Routing)
* SLURP (Scalable Location Update-Based Routing Protocol)

### 1.9.1.4 Các giao thức định tuyến theo nhận biết vị trí (Location-aware protocol)

Các giao thức định tuyến theo nhận biết vị trí trong mạng Ad hoc quy định mỗi node trong mạng Ad hoc đều chứa thông tin về vị trí của tất cả các node trong mạng. Cách tốt nhất và dễ dàng nhất là sử dụng hệ thống định vị toàn cầu GPS để xác định chính xác tọa độ của các node trong bất kỳ khu vực địa lý nào. Thông tin về địa điểm sau đó được tối ưu hóa bởi giao thức định tuyến để tìm ra đường đi đúng nhất.

Một số giao thức trong họ này có thể kể đến là:

* LAR (Location-Aided Routing)
* DREAM (Distance Routing Effect Algorithm for Mobility)
* GPSR (Greedy Perimeter Stateless Routing)
* LAKER (Location Aided Knowledge Extraction Routing)
* MORA (Movement-Based Algorithm for Ad Hoc Networks)

### 1.9.1.5 Các giao thức định tuyến đa đường (Multipath Protocol)

Các giao thức định tuyến đa đường tạo nên nhiều đường đi từ node nguồn đến node đích. Ưu điểm chính của việc tìm kiếm nhiều đường khả dụng là băng thông giữa các liên kết được sử dụng hiệu quả hơn. Với cơ chế chia tải truyền dữ liệu theo nhiều đường sẽ giảm nghẽn trong mạng.

Một số giao thức trong họ này có thể kể đến là:

* CHAMP (CacHing and Multipath routing Protocol)
* AOMDV (Ad hoc On-Demand Multipath Distance Vector Routing)
* SMR (Split Multipath Routing)

**1.9.2 Một số loại giao thức phổ biến**

### 1.9.2.1 Giao thức DSDV (Destination – Sequenced Distance – Vector)

Là một trong những giao thức đầu tiên được phát triển cho mạng Ad hoc. DSDV là một biến thể của giao thức định tuyến distance vector theo kiểu proactive, dựa trên ý tưởng của thuật toán định tuyến kinh điển Bell – Man –Ford với một chút cải tiến.

Cải tiến mới của DSDV là sử dụng kỹ thuật đánh số sequence number. Kỹ thuật này dùng để nhận ra các con đường đi không còn giá trị trong quá trình cập nhật bảng định tuyến, do đó, sẽ tránh được vòng lặp trong quá trình định tuyến. Mỗi node sẽ tăng số sequence number khi gởi thông tin về bảng định tuyến của nó cho các node khác trong mạng.

Các cơ chế trong DSDV:

* *Quản lý bảng định tuyến:* Mỗi node luôn duy trì một bảng định tuyến đến các node khác trong mạng. Thông tin của một mục trong bảng định tuyến bao gồm:
  + Địa chỉ của node đích
  + Số hop đến đích (hop – count)
  + Next hop
  + Số sequence number của node đích

Để đảm bảo cho bảng định tuyến luôn luôn phù hợp với những thay đổi trong mạng thì các node phải thường xuyên cập nhật bảng định tuyến theo một khoảng thời gian nhất định khi mạng có sự thay đổi. Do đó, các node phải quảng bá thông tin định tuyến của đó cho các node khác trong mạng bằng cách phát broadcast những thay đổi trong bảng định tuyến của nó. Khi một node nhận gói tin cập nhật bảng định tuyến, nó sẽ kiểm tra số sequence number trong gói tin được cập nhật lớn hơn hoặc bằng với số sequence number trong bảng định tuyến và số có hop – count nhỏ hơn thì node sẽ cập nhật thông tin đó vào bảng định tuyến.

* *Cách thức cập nhật bảng định tuyến trong DSDV*

Bảng định tuyến sẽ được cập nhật theo hai cách:

Thứ nhất, cập nhật toàn bộ bảng định tuyến cho các node láng giềng và có thể truyền trong nhiều packet, gọi là full-dump.

Thứ hai, cập nhật các phần thay đổi trong bảng định tuyến của nó cho các node láng giềng và thông tin thay đổi chỉ được gửi đi trong một packet, gọi là incremental – update.

Đối với một mạng Ad hoc tương đối ổn định, thì kiểu cập nhật incremental – update sẽ thường được sử dụng để hạn chế lưu lượng truyền trên mạng. Trong khi đó, full – dump sẽ được dùng trong những mạng thiếu ổn định

* *Quản lý sự thay đổi của Topology*

Khi một node di chuyển từ nơi này đến nơi khác thì các liên kết của nó với các node láng giềng có thể không còn hiệu lực. Khi một node phát hiện rằng liên kết đến next hop của nó không còn tồn tại thì đường đi qua next hop đó lập tức sẽ có hop – count là ∞ và số sequence number được tăng lên 1. Sau đó node sẽ phát broadcast thông tin đó cho tất cả các node trong mạng và các node sẽ cập nhật lại bảng định tuyến của mình.

**Ưu điểm** của DSDV là đảm bảo không có đường định tuyến kín bằng cách sử dụng số thứ tự để đánh dấu mỗi đường. Số thứ tự cho biết mức độ “mới” của đường định tuyến, số càng lớn thì mức độ đảm bảo càng cao (đường R được coi là tốt hơn R’ nếu số thứ tự của R lớn hơn, trong trường hợp có cùng số thứ tự thì R phải có số bước nhỏ hơn). Số thứ tự sẽ tăng khi nút A phát hiện ra đường đến đích D đị phá vỡ, sau đó nút A quảng bá đường định tuyến của nó tới nút D với số bước không giới hạn và số thứ tự sẽ tăng lên.

DSDV phụ thuộc vào thông tin quảng bá định kỳ nên nó sẽ tiêu tốn thời gian để tổng hợp thông tin trước khi đường định tuyến được đưa vào sử dụng. Thời gian này là không đáng kể đối với mạng có cấu trúc cố định nói chung (bao gồm cả mạng có dây), nhưng với mạng Ad hoc thời gian này là đáng kể, có thể gây ra mất gói tin trước khi tìm ra được định tuyến hợp lý. Ngoài ra, bản tin quảng cáo định kỳ cũng là nguyên nhân gây ra lãng phí tài nguyên mạng.

### 1.9.2.2 Giao thức DSR (Dynamic Source Routing)

DSR là giao thức định tuyến đơn giản và hiệu quả được thiết kế riêng cho mạng MANET. DSR cho phép mạng tự động tổ chức và cấu hình mà không cần đến sự quản trị và cơ sở hạ tầng sẵn có của mạng. Giao thức định tuyến DSR bao gồm hai cơ chế: Route Discovery và Route Maintenance, nhờ hai cơ chế này mà các node có thể tìm và duy trì được các đường đi đến các node trong mạng.

Một đặc tính nổi bật khác của DSR là nó sử dụng kỹ thuật định tuyến Source Routing, khi đó bên gởi sẽ biết toàn bộ thông tin đường đi đến đích, điều này giúp cho việc định tuyến trên mạng không bị hiện tượng vòng lặp (loop) làm tăng hiệu năng của mạng. Để định tuyến được thì trong phần header của packet lưu giữ thêm thông tin về source route. Thông tin về bảng định tuyến được lưu trong bảng route cache. Khi một node trong mạng Ad hoc muốn gởi dữ liệu đến một node đích nó sẽ tìm kiếm thông tin trong route cache, nếu chưa có thông tin về đường đi thì node nguồn sẽ khởi động tiến trình route discovery để tìm kiếm con đường đi đến đích.

* *Cơ chế Route Discovery*

Route Discovery cho phép các host trong mạng Ad hoc tìm kiếm đường đi đến đích một cách tự động thông qua các node trung gian. Tiến trình route discovery sẽ phát một broadcast gói Route Request (RREQ) lên mạng. trong gói RREQ chứa một số request \_ ID là một số được tạo ra bởi node nguồn. Khi một node nhận gói RREQ, nó sẽ tiến hành kiểm tra như sau:

* Bước 1: Nó kiểm tra xem đây có phải là lần đầu tiên nó nhận gói RREQ có địa chỉ đích và số request \_ ID hay không? Nếu không phải thì nó sẽ loại bỏ gói tin này và không xử lý. Ngược lại, sang bước tiếp theo.
* Bước 2: Nó kiểm tra trong trường nguồn định tuyến của gói RREQ đã có địa chỉ của nó hay chưa? Nếu đã tồn tại thì nó cũng sẽ loại bỏ gói tin đó và không xử lý thêm. Ngược lại, qua bước 3.
* Bước 3: Nó kiểm tra trong route cache của nó có đường đi đến node đích mà còn hiệu lực hay không? Nếu có, nó sẽ phải hồi lại cho node nguồn bằng gói route reply (RREP) chứa thông tin về đường đi đến đích và kết thúc tiến trình. Ngược lại, qua bước 4.
* Bước 4: Nó kiểm tra địa chỉ đích cần tìm có trùng với địa chỉ của nó hay không? Nếu trùng, nó cũng sẽ gởi lại cho node nguồn gói RREP chứa đường đi đến đích và kết thúc tiến trình. Ngược lại, nó sẽ phát broadcast đến các láng giềng của nó.
* *Cơ chế Route Maintenance*

Trong giao thức định tuyến DSR, các node khi chuyển gói tin trên mạng đều phải có nhiệm vụ xác nhận rằng gói tin đó đã chuyển đến next hop hay chưa? Trong một trường hợp nào đó mà node đó phát hiện rằng không thể truyền gói tin đến next hop. Nó sẽ gởi gói Route Error (RRER) cho node nguồn để thông báo tình trạng hiện thời của liên kết và địa chỉ hiện thời của next hop mà không thể chuyển đi. Khi node nguồn nhận được gói RRER, nó sẽ xóa con đường đi mà sử dụng liên kết bị hỏng trong route cache và tìm một đường đi khác mà nó biết trong route cache hoặc khởi động một tiến trình route discovery mới nếu đường đi này đang có nhu cầu sử dụng.

### ****1.9.2.3 Giao thức AODV (Ad-hoc On-Demand Distance Vector)****

**Một mạng ad-hoc là một tập hợp của các nút di động mà không có sự can thiệp cần thiết của bất kì một điểm truy cập trung tâm hoặc cơ sở hạ tầng hiện có. AODV là một thuật toán mới với sự hoạt động như vậy của mạng ad-hoc.**

**Mỗi host di động hoạt động như một router chuyên dụng, và các tuyến đường thu được khi cần thiết. Thuật toán định tuyến AODV là khá phù hợp cho một mạng tự khởi động theo nhu cầu của người sử dụng có nhu cầu sử dụng mạng ad-hoc. AODV cung cấp các tuyến đường không lặp, ngay cả khi sửa chữa các đường liên kết bị hỏng. Bởi vì giao thức không quảng bá định tuyến định kì, nhu cầu về băng thông khả dụng cho mỗi nút di động là ít hơn đáng kể so với những giao thức yêu cầu quảng bá định kì.**

**Mục tiêu chủ yếu của thuật toán như sau:**

**- Để truyền broadcast các gói tin chỉ khi cần thiết.**

**- Phân biệt các lân cận và duy trì kết cấu chung.**

**- Để phổ biến thông về những thay đổi trong kết nối nội bộ với những nút lân cận nếu chúng cần thông tin.**

**- AODV sử dụng một cơ chế truyền broadcast route discovery cũng như được sử dụng với những thay đổi trong thuật toán DSR. Thay vì định tuyến nguồn, AODV dựa vào việc vào việc tự động thiết lập các mục trong bảng định tuyến tại các nút trung gian.**

**a. Path discovery**

**Quá trình khám phá con đường được khởi đầu bất kì khi nào một nút nguồn cần giao tiếp với những nút khác mà không có thông tin định tuyến trong bảng của mình.**

**Mỗi nút duy trì hai thành phần riêng biệt: một số thứ tự của nút và một ID broadcast. Một nút nguồn phát hiện con đường bằng cách truyền broadcast một gói tin yêu cầu định tuyến (RREQ) đến các nút lân cận. Gói tin RREQ chứa các trường sau: <source\_addr source sequence# broadcast id dest\_addr dest sequence# hop cnt>**

**Cặp <source\_addr, broadcast\_id> định dạng một gói RREQ. Trường broadcast\_id được tăng lên mỗi khi nguồn phát một gói RREQ mới. Mỗi nút lân cận đáp ứng gói RREQ bằng cách gửi hồi đáp RREP về nguồn và phát broadcast gói RREQ tới những nút lân cận của nó sau khi tăng trường hop\_cnt. Chú ý rằng một nút có thể nhân nhiều bản copy của cùng một một gói broadcast từ nhiều nút khác nhau. Khi một nút trung gian nhận được một gói RREQ, nếu nó đã nhận một gói RREQ với cùng trường broadcast\_id và source\_addr thì nó sẽ loại gói RREQ dư và không truyền broadcast gói tin này.**

**b. Khởi tạo đường ngược (Reverse-Path)**

**Có hai số thứ tự (ngoài broadcast\_id) được bao gồm trong một gói RREQ: số thứ tự nguồn và số thứ tự điểm đích cuối cùng. Số thứ tự nguồn được sử dụng để duy trì các thông tin mới về các tuyến đường ngược về nguồn, và số thứ tự điểm đích chỉ ra sự mới mẻ của một định tuyến tới đích trước khi nó được chấp nhận bởi nguồn. Khi RREQ đi từ nguồn tới những điểm đích khác nhau, nó sẽ tự động thiết lập các con đường ngược lại từ tất cả các nút về nguồn.**

**Để thiết lập một đường ngược lại, một nút ghi lại địa chỉ của những nút lân cận mà nó nhận được những gói RREQ đầu tiên. Những mục tuyến đường ngược được duy trì trong thời gian vừa đủ để gói RREQ đi qua mạng và gửi hồi đáp về nút gửi tin.**

**c. Khởi tạo forward-path**

**Một gói tin đến một nút (có thể là điểm đích của nó) mà sở hữu một nút hiện tại tới đích. Nút nhận đầu tiên sẽ kiểm tra rằng RREQ được nhận thông qua liên kết hai chiều. Nếu một nút trung gian có một mục định tuyến cho đến điểm đích, nó sẽ quyết định tuyến đường bằng cách so sánh các số thứ tự của điểm đích trong RREQ với mục định tuyến của nó. Nếu số thứ tự đích đến của RREQ lớn hơn thì nút trung gian không được sử dụng bảng định tuyến của nó để hồi đáp RREQ. Thay vào đó, có nút trung gian sẽ truyền broadcast lại gói RREQ. Các nút trung gian có thể hồi đáp chỉ khi nó có một đường định tuyến có số thứ tự lớn hơn hoặc bằng số thú tự được lưu trong RREQ. Nếu nó không có đường hiện tại tới đích và nếu RREQ không được xử lý trước đó, các nút sẽ truyền unicast RREP trả về nút lân cận đã truyền RREQ cho nó. Một gói RREP chứa những thông tin sau: <source\_addr, des\_addr, des\_sequnce #, hop\_cnt, lifetime>**

**Khi RREP trở về nguồn, mỗi nút trên đường thiết lập một con trở về phía trước tới nút đã gửi gói RREP, cập nhật thông tin timeout cho các mục định tuyến tới nguồn và đích. Ghi số thứ tự điểm đến mới nhất cho các nút đích được yêu cầu.**

**d. Quản lý bảng định tuyến**

**Bảng định tuyến của mỗi nút sẽ chứa những thông tin sau:**

**- Đích, nguồn.**

**- Danh sách nút tiếp theo.**

**- Số thứ tự nút đích.**

**- Thời gian hết hạn của bảng định tuyến.**

**- ….**

**e. Duy trì đường định tuyến**

**Sự di chuyển của cac nút không nằm trên đường hoạt động không gây ảnh hường đến việc định tuyến đến đích của con đường đó.**

**Nếu nút nguồn di chuyển trong quá trình hoạt động, nó có thể bắt đầu lại thủ tục khám phá tuyến để thiết lập một tuyến đường mới tới đích.**

**Khi nút đích hoặc các nút trung gian di chuyển, một RREP đặc biết được gửi đến các nút nguồn bị ảnh hưởng.**

**Khi nhận được thông báo một đường liên kết bị hỏng, nút nguồn có thể bắt đầu lại quá trình khám phá nếu chúng đòi hỏi một đường định tuyến tới đích.**

**f. Hồi đáp định tuyến (RREP)**

**Một nút trung gian (không phải nút đích) cũng có thể gửi một RREP với điều kiện nó biết một con đường mới hơn so với tuyến đường trước đây mà S biết.**

**Một RREQ mới bởi nguồn S đến một nút đích được gán số thứ tự đích cao hơn. Một nút trung gian biết một đường nhưng số thứ tự nhỏ hơn không thể gửi RREP.**

# ****CHƯƠNG 2: NHỮNG VẤN ĐỀ VỀ BẢO MẬT TRONG MẠNG MANET VÀ MỘT SỐ CÁCH THỨC TẤN CÔNG TRONG MẠNG MANET****

## ****2.1. Những vấn đề về an ninh trong mạng MANET****

### 2.1.1 Thách thức về an ninh trong mạng MANET

- Môi trường truyền sóng điện từ là không khí, vì vậy nguy cơ bị nghe trộm là rất lớn, từ đó kẻ tấn công có thể phân tích lưu lượng mạng phục vụ cho các mục đích tấn công tiếp theo.

- Việc các nút gia nhập và rời mạng bất kỳ lúc nào tạo nên sự thay đổi thường xuyên về cấu trúc mạng đòi hỏi các giao thức định tuyến liên tục phát các yêu cầu quảng bá trong toàn mạng cũng dẫn đến việc mất an ninh trong mạng. Đồng thời, việc cấu trúc mạng liên tục thay đổi cũng là một khó khăn để các giao thức định tuyến phát hiện rằng thông điệp điều khiển sai lệch được sinh ra bởi nút độc hại hay là do quá trình thay đổi cấu trúc mạng.

- Giới hạn về tài nguyên như băng thông và năng lượng làm giảm khả năng chống đỡ của mạng trước các cuộc tấn công.

- Thiếu một cơ sở hạ tầng trợ giúp gây khó khăn khi triển khai các cơ chế bảo mật trong mạng.

### 2.1.2. Các yêu cầu về an ninh trong mạng MANET

Để đảm bảo an toàn trong giao thức định tuyến mạng MANET yêu cầu:

- **Tính bảo mật (Confidentiality):** Đảm bảo thông điệp truyền trong mạng

phải được giữ bí mật. Trong một số trường hợp cần đảm bảo bí mật cả với các

thông điệp định tuyến quảng bá trong mạng vì từ thông tin các thông điệp này có

thể khai thác một số thông tin giúp ích cho việc tấn công.

- **Tính xác thực (Authentication):** Đảm bảo một nút phải xác định được

danh tính rõ ràng của một nút khác trong quá trình truyền dữ liệu với nó.

- **Tính toàn vẹn (Integrity):** Đảm bảo các thông điệp không bị chỉnh sửa

trong toàn bộ quá trình truyền.

- **Tính chống chối bỏ (Non-Repudiation):** Đảm bảo luôn xác định được

nguồn gốc thông điệp truyền từ nút nào.

- **Tính sẵn sàng (Availability):** Đảm bảo tính sẵn sàng của các nút mặc dù

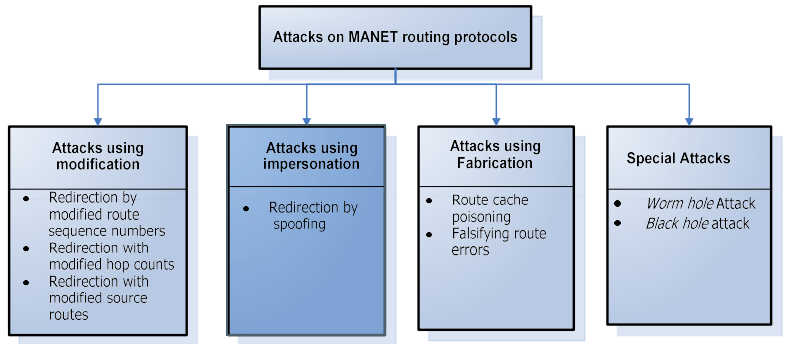
bị các cuộc tấn công. Trong đó tấn công từ chối dịch vụ đe dọa tới bất kỳ tầng

nào trong mạng ad hoc. Ở tầng điều khiển môi trường truy nhập, kẻ tấn công có thể sử dụng hình thức chèn ép kênh truyền vật lý; ở tầng mạng sự gián đoạn

trong hoạt động của các giao thức định tuyến, ở các tầng cao hơn có thể là tấn

công vào các ứng dụng bảo mật ví dụ như hệ thống quản lý khóa.

### 2.2 Các phương thức tấn công trong giao thức định tuyến mạng MANET



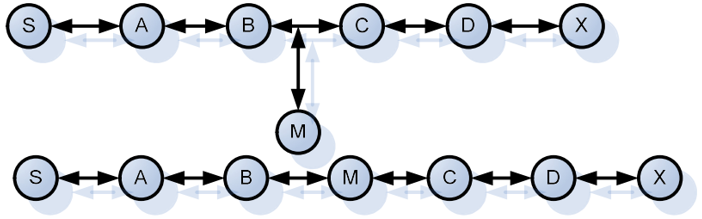
*Hình 2. 1 Các kiểu tấn công giao thức định tuyến trong mạng MANET*

### 2.2.1. Tấn công bằng cách sửa đổi thông tin định tuyến

Trong kiểu tấn công bằng cách sửa đổi, một số trường của thông điệp định tuyến đã bị sửa đổi dẫn đến việc làm rối loạn các tuyến đường, chuyển hướng hoặc hình thành một cuộc tấn công từ chối dịch vụ. Bao gồm các hình thức sau:

- Sửa đổi số tuần tự đích (destination sequence number), số chặng (hop\_count) của tuyến đường: Thể hiện rõ ràng nhất trong giao thức định tuyến AODV. Kẻ tấn công sửa đổi số tuần tự đích, hoặc số chặng của gói tin yêu cầu tuyến (RREQ) hoặc gói tin trả lời tuyến (RREP) để tạo nên tuyến đường mới có hiệu lực, qua đó chiếm quyền điều khiển quá trình truyền dữ liệu từ nguồn tới đích.

- Sửa đổi nguồn của tuyến đường: Thể hiện rõ trong giao thức định tuyến nguồn DSR. Kẻ tấn công chặn thông điệp sửa đổi danh sách các nguồn trước khi gửi tới nút đích trong quá trình truyền.



Hình 2. 2 Ví dụ tấn công bằng cách sửa đổi

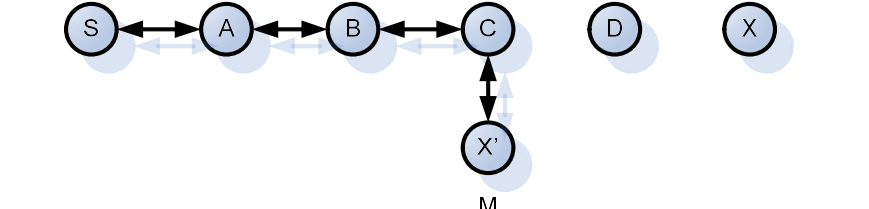
Ví dụ: Ở hình 2.2, tuyến đường ngắn nhất giữa nút nguồn S và đích X là (S – A – B – C – D – X). S và X không thể truyền dữ liệu trực tiếp cho nhau và kịch bản như sau: Nút M là nút độc hại cố gắng thực hiện tấn công từ chối dịch vụ. Giả sử nút nguồn S cố gắng gửi dữ liệu tới nút X nhưng nút M chặn gói tin và bỏ đi nút D trong danh sách và gói tin được chuyển tiếp đến nút C, nút C cố gắng gửi gói tin đến nút X nhưng không thể vì nút C không thể truyền tin trực tiếp tới nút X. Dẫn tới nút M thực hiện thành công cuộc tấn công DDOS trên X.

### 2.2.2. Tấn công bằng cách mạo danh

Kiểu tấn công bằng cách mạo danh đe dọa tính xác thực và bảo mật trong mạng. Nút độc hại có thể giả mạo địa chỉ của nút khác để thay đổi cấu trúc mạng hoặc ẩn mình trên mạng.

Nút độc hại mạo danh bằng cách thay đổi địa chỉ IP nguồn trong thông điệp điều khiển. Một lý do khác để mạo danh là để thuyết phục các nút khác thay đổi thông tin trong bảng định tuyến của chúng rằng tôi là một nút tin cậy, cách này còn được biết đến như tấn công vào bảng định tuyến.

Một trong những điển hình của loại tấn công này là “Man in the midle attack”. Nút độc hại thực hiện cuộc tấn công này bằng cách kết hợp giữa mạo danh và loại bỏ gói tin. Về mô hình vật lý, nó phải được chọn sao cho là nút nằm trong phạm vi tới đích, tức là nằm giữa tuyến đường để có thể chặn bất kỳ thông tin nào từ nút khác tới đích. Để thực hiện việc này, khi nút nguồn gửi RREQ, nút độc hại loại bỏ RREQ và gửi trả lời RREP giả mạo như là nút đích trả lời, song song với quá trình này nó gửi gói tin RREQ tới nút đích và loại bỏ gói RREP từ nút đích trả lời. Bằng cách này, nút độc hại đứng giữa nắm giữ thông tin trong quá trình truyền thông nút nguồn, nút đích. Với thông tin thu được, sử dụng các phương pháp bảo mật ở các lớp trên, kẻ tấn công có thể giải mã hoặc tìm ra các khóa để tiếp tục quá trình tấn công vào các lớp phía trên của hệ thống.

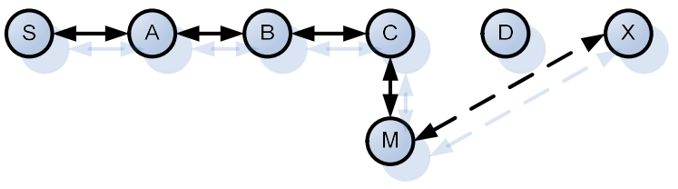


*Hình 2. 3 Ví dụ về tấn công bằng cách mạo danh*

Ví dụ: Ở hình 2.3, nút S muốn gửi dữ liệu tới nút X và trước khi gửi dữ liệu tới nút X, nó khởi động quá trình tìm đường tới X. Nút M là nút độc hại, khi nút M nhận được gói tin tìm đường tới X, M thực hiện sửa đổi địa chỉ của nó thành địa chỉ của X, đóng giả nó thành nút X’. Sau đó nó gửi gói tin trả lời rằng nó chính là X tới nút nguồn S. Khi S nhận được gói tin trả lời từ M, nó không chứng thực và chấp nhận tuyến đường và gửi dữ liệu tới nút độc hại. Kiểu tấn công này cũng được gọi là tấn công lặp định tuyến trong mạng.

### 2.2.3. Tấn công bằng cách tạo ra thông tin bịa đặt

Trong cách tấn công này, nút độc hại cố gắng để “bơm” vào mạng các thông điệp giả mạo hoặc các thông điệp định tuyến sai để phá vỡ cơ chế định tuyến trong mạng. Các cuộc tấn công kiểu này rất khó để phát hiện bởi việc gói tin định tuyến bơm vào mạng đều là gói tin hợp lệ được xử lý bởi các nút trong mạng.



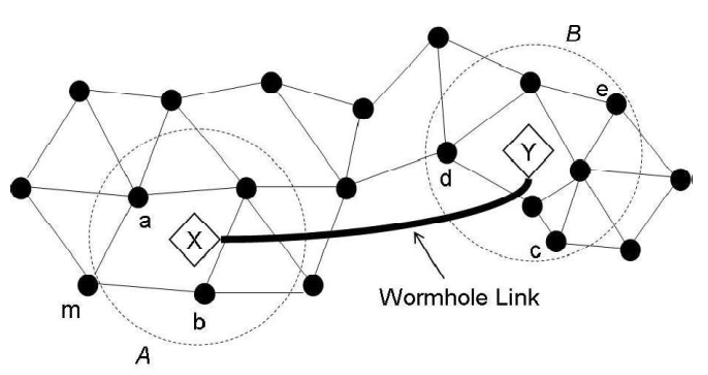
Hình 2. 4 Tấn công bằng cách tạo ra thông tin bịa đặt

Hình 2.4. Ví dụ về tấn công bằng cách tạo ra thông tin bịa đặt. Nút nguồn S muốn gửi dữ liệu tới nút X, do đó nó phát một yêu cầu tuyến đường để tìm ra con đường tới nút X. Nút độc hại M giả vờ có một tuyến đường đến nút đích X, và trả lời RREP tới nút nguồn S. Các nút nguồn, không có sự kiểm tra tính hợp lệ của RREP, chấp nhận RREP và gửi dữ liệu đến thông qua nút M.

### 2.3.4. Một vài kiểu tấn công đặc biệt

**- Tấn công lỗ sâu (Wormhole Attack):** Trong một cuộc tấn công kiểu wormhole, hai hoặc nhiều hơn các nút độc hại thông đồng với nhau bằng cách thiết lập một đường. Trong giai đoạn tìm đường của giao thức định tuyến, thông báo yêu cầu định tuyến được chuyển tiếp giữa các nút độc hại sử dụng các đường hầm đã được thiết lập sẵn. Do đó, thông báo yêu cầu định tuyến đầu tiên đến đích là một trong những thông báo được chuyển tiếp từ nút độc hại.

Do vậy nút độc hại được thêm vào trong đường dẫn từ nút nguồn đến nút đích. Một khi các nút độc hại đã có trong đường dẫn định tuyến, các nút độc hại hoặc sẽ bỏ tất cả các gói dữ liệu dẫn đến việc từ chối hoàn toàn dịch vụ, hoặc loại bỏ có chọn lọc gói tin để tránh bị phát hiện.



Hình 2. 5 Tấn công Wormhole trong mạng MANET

Ở hình 2.5 là một ví dụ về tấn công wormhole. Nút X và nút Y là hai đầu của kiểu tấn công wormhole. Nút X phát lại mọi gói tin trong vùng A mà nút Y nhận được trong vùng B và ngược lại. Dẫn tới việc tất cả các nút trong vùng A cho rằng là hàng xóm của các nút trong vùng B và ngược lại. Kết quả là, nút X và nút Y dễ dàng tham gia vào tuyến đường truyền dữ liệu. Khi đó chúng chỉ việc hủy bỏ mọi gói tin truyền qua chúng và đánh sập mạng.  
- **Tấn công lỗ đen (Blackhole Attack):**Trong cuộc tấn công lỗ đen, nút độc hại tuyên bố rằng nó có tuyến đường hợp lệ tới tất cả các nút khác trong mạng để chiếm quyền điều khiển lưu lượng giữa các thực thể truyền. Sau khi nhận dữ liệu truyền, nó không chuyển tiếp mà loại bỏ (drop) tất cả các gói tin này. Do đó, nút lỗ đen này có thể giám sát và phân tích lưu lượng toàn bộ các nút trong mạng mà nó đã tấn công.

# CHƯƠNG 3: TẤN CÔNG BLACKHOLE VỚI GIAO THỨC AODV VÀ MỘT SỐ GIẢI PHÁP PHÒNG CHỐNG TẤN CÔNG BLACKHOLE

## ****3.1. Lỗ hổng của giao thức AODV****

**Giao thức AODV dễ bị kẻ tấn công làm sai lệch thông tin đường đi để chuyển hướng đường đi và sau đó sẽ thực hiện các cuộc tấn công khác. Trong mỗi gói tin định tuyến AODV, một số trường quan trọng như số đếm chặng HC, số tuần tự đích SN của nguồn và đích, IP header, địa chỉ IP nguồn và đích của AODV, chỉ số RREQ ID, là những yếu tố cần thiết để thực thi giao thức đúng đắn. Sự sai sót của bất cứ trường nào kể trên cũng có thể khiến AODV gặp sự cố. Bảng sau ghi lại vài trường dễ bị phá hoại trong thông điệp định tuyến AODV và sự thay đổi khi chúng bị tấn công.**

|  |  |
| --- | --- |
| **Tên trường** | **Sự thay đổi khi bị tấn công** |
| RREQ ID | Tăng lên để tạo ra một yêu cầu RREQ mới. |
| Số chặng - HC | Nếu số tuần tự là giống, giảm đi để cập nhật bảng chuyển tiếp của nút khác, hoặc tăng để bỏ cập nhật. |
| IP header cũng như địa chỉ IP đích và nguồn của AODV | Thay thế nó cùng với địa chỉ IP khác hoặc không có giá trị. |
| Số tuần tự của đích và nguồn - SN | Tăng lên để cập nhật bảng định tuyến chuyến tiếp, hoặc giảm để bỏ cập nhật. |

*Bảng 3. 1 Một số nguy cơ bảo mật trong AODV*

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 SN cao nhất và giá trị HC 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.

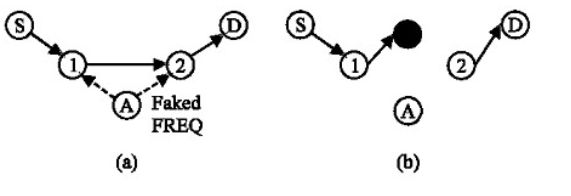
## 3.2. Phân loại tấn công kiểu lỗ đen

Để thực hiện cuộc tấn công kiểu lỗ đen trong giao thức AODV, có thể phân loại theo hai cách: **RREQ Blackhole attack** và **RREP Blackhole attack**.

1. **RREQ Blackhole attack**

Trong kiểu tấn công này, kẻ tấn công có thể giả mạo gói tin RREQ để thực hiện tấn công lỗ đen. Theo đó, kẻ tấn công đánh lừa bằng cách quảng bá gói tin RREQ với một địa chỉ của nút không có thật. Các nút khác sẽ cập nhật bảng định tuyến của mình với tuyến đường tới đích thông qua nút không có thật này. Vì vậy, tuyến đường tới đích sẽ bị đứt làm gián đoạn quá trình truyền thông giữa nút nguồn và nút đích.

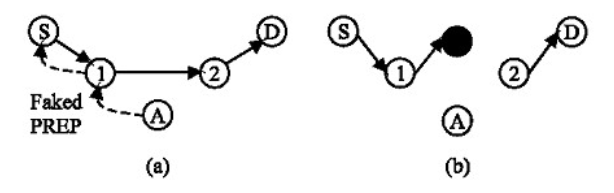
Kẻ tấn công thực hiện tấn công lỗ đen giữa nút nguồn và nút đích bằng việc giả mạo gói tin RREQ được thể hiện trên Hình 3.1, khi nút lỗ đen A gửi gói tin RREQ giả mạo đến hai nút lân cận là 1 và 2 với thông tin về một địa chỉ nút không có thật. Khi đó nút 1 và 2 sẽ cập nhật vào bảng định tuyến của mình con đường tới nút đích D thông qua nút không có thật này, dẫn đến tuyến đường từ nút nguồn S đến nút đích D bị gián đoạn.



Hình 3. 1 Thực hiện tấn công lỗ đen bằng việc giả mạo gói tin RREQ

1. **RREP blackhole attack**

Hình 3.2 thể hiện quá trình tấn công RREP blackhole attack. Nút tấn công A phát gói tin RREP giả mạo tới nút khởi tạo gói RREQ là nút nguồn S, khi nhận được gói tin giả mạo này, các nút tham gia vào quá trình khám phá tuyến sẽ cập nhật bảng định tuyến tuyến đường từ nút nguồn tới nút đích thông qua nút lỗ đen này.



Hình 3. 2 Thực hiện tấn công lỗ đen bằng việc giả mạo gói tin RREP

## 3.3. Một số giải pháp phòng chống tấn công lỗ đen trong giao thức AODV

### 3.3.1. ARAN (Authenticated Routing for Ad hoc Networks)

ARAN là giao thức định tuyến an toàn dựa trên giao thức AODV. Giao thức ARAN sử dụng mã hóa, chứng chỉ số để ngăn chặn hầu hết các cuộc tấn công và phát hiện những hành vi không bình thường. Chứng chỉ chứng thực nút ARAN sử dụng mật mã, chứng chỉ số để đảm bảo tính xác thực, tính toàn vẹn và không chối bỏ trong quá trình định tuyến.

ARAN yêu cầu một trung tâm chứng thực T, trong đó tất cả các nút đều biết được khóa công khai của T. Các nút sử dụng chứng chỉ để chứng thực chính nó với các nút khác trong quá trình định tuyến. Trước khi tham gia vào mạng, mỗi nút phải gửi yêu cầu một chứng chỉ từ T. Sau khi chứng thực định danh với T, mỗi nút sẽ nhận được một chứng chỉ số.

Ví dụ:



Chứng chỉ bao gồm địa chỉ IP của A, khóa công khi của A, thời gian tạo t và thời gian hết hạn e.

Chứng thực quá trình khám phá tuyến nút nguồn A bắt đầu quá trình tìm đường tới nút đích X bằng cách gửi quảng bá tới hàng xóm của nó gói tin tìm đường (RDP – route discovery packet), kèm chứng chỉ của nó.



RDP bao gồm: định danh “RDP”, địa chỉ IP của đích là X, một số N(A) để định danh duy nhất RDP đến từ nguồn (mỗi lần cần tìm đường nó tăng giá trị này lên 1), tất cả được ký bằng khóa bí mật của A. Khi một nút nhận được gói RDP, nó lưu đường quay trở lại nguồn bằng cách lưu nút hàng xóm mà từ đó nó nhận được RDP. Nó sử dụng khóa công khai của A được trích xuất từ chứng chỉ của A. Nó cũng kiểm tra (N(A) và IP(A)) để xác thực rằng nó chưa xử lý RDP. Sau đó nó ký nội dụng thông điệp và kèm chứng chỉ của nó và phát quảng bá tới các hàng xóm của nó. Chữ ký số ngăn chặn tấn công kiểu lừa bịp (spoofing).



Nút C là hàng xóm của B nhận được gói tin. Nó xác thực chữ ký số của cả A và B. Sau khi xác thực C tách bỏ chứng chỉ và chữ ký của B, lưu B và ký nội dung gói tin gốc từ A và phát quảng bá kèm chứng chỉ của nó.



Mỗi nút trung gian lặp lại các bước như trên cho tới khi gói tin đạt đến đích. Nút đích unicast gói tin trả lời (REP) ngược trở lại nút nguồn.



REP bao gồm, định danh “REP”, địa chỉ IP của A, số N(A), chứng chỉ của X. Mỗi nút nhận được REP sẽ ký REP và ghi đè chứng chỉ của nó trước khi chuyển gói.



C nhận được sẽ gỡ bỏ chữ ký và chứng chỉ của D. Sau đó ký và gửi kèm chứng chỉ của nó đến B.



Khi nút nguồn nhận được REP, nó sẽ xác thực chữ ký của đích và giá trị nonce trả lại bởi đích. Khi xác thực thành công, nguồn lưu tuyến đường và bắt đầu gửi dữ liệu.

### 3.3.2. SAODV (Secure Ad hoc On-demand Distance Vector)

Giao thức SAODV được công bố bởi Manel Guerrero Zapata vào năm 2006 để giải quyết các thử thách an ninh trong mạng MANET.

SAODV là một giao thức mở rộng của giao thức AODV, nó được sử dụng để bảo vệ cơ chế quá trình khám phá tuyến bằng cách cung cấp các tính năng an ninh như toàn vẹn, xác thực và chống chối bỏ.

SAODV giả định rằng mỗi nút trong mạng có một chữ ký số cung cấp từ một hệ thống mật mã khóa bất đối xứng để bảo mật các thông điệp định tuyến AODV. Hơn nữa, mỗi nút có khả năng xác minh mối liên hệ giữa địa chỉ của nút và khóa công khai của nút đó. Do đó, cần phải có một lược đồ quản lý khóa cho giao thức SAODV. SAODV sử dụng hai cơ chế để bảo mật các thông điệp định tuyến AODV:

* Chữ ký số để xác thực các trường không thay đổi trong các thông điệp định tuyến.
* Chuỗi hàm băm để bảo vệ các thông tin về số chặng (thay đổi trong quá trình phát hiện tuyến).

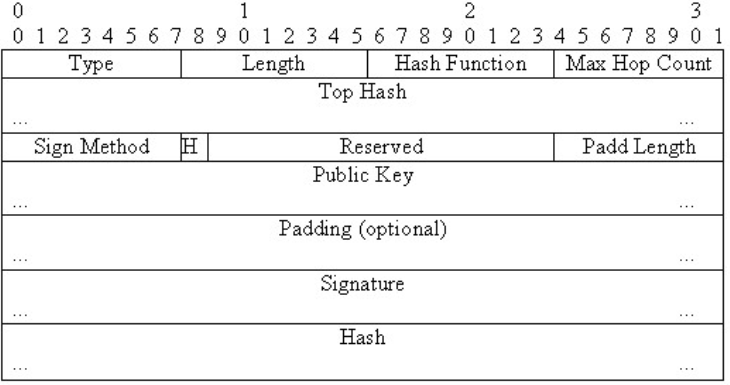
Đối với các trường không thay đổi, chứng thực có thể được thực hiện theo cách thức từ điểm tới điểm, tuy nhiên kỹ thuật này không thể áp dụng cho các trường thông tin thay đổi.

Các thông điệp RREQ, RREP mở rộng thêm một số trường sau:

**<Type, Length, Hash function, Max Hop Count, Top hash, Signature, Hash>**

Thông điệp RRER mở rộng thêm các trường sau:

**<Type, Length, Reserved, Signature>**



Hình 3. 3 Định dạng của thông điệp định tuyến RREQ (RREP) mở rộng

Trong đó:

- Type: 64 cho RREQ và 65 cho RREP.

- Length: Độ dài của thông điệp không bao gồm Type and Length.

- Hash Function: Thuật toán hàm băm được sử dụng.

- Max Hop Count: Số bước nhảy lớn nhất có thể hỗ trợ, sử dụng để xác thực số bước nhảy.

- Top Hash: Giá trị hàm băm tương ứng với số bước nhảy lớn nhất.

- Sign Method: Thuật toán sử dụng trong chữ ký số.

- Padding Length: Độ dài của phần số nhồi thêm.

- Public Key: Khóa công khai của nút nguồn phát gói tin.

- Padding: Số nhồi thêm.

- Signature: Chữ ký số tính từ tất cả các trường không thay đổi của thông điệp định tuyến.

- Hash: Giá trị hàm băm tương ứng với số bước nhảy tại nút hiện tại

**a) SAODV sử dụng chuỗi hàm băm để xác thực trường thay đổi của thông điệp:**

SAODV sử dụng các chuỗi hàm băm để xác thực Số chặng HC của RREQ và RREP chuyển tiếp giữa các nút xung quanh trong quá trình thăm dò tuyến. Một chuỗi hàm băm được hình thành từ hàm băm một chiều và một giá trị khởi đầu ngẫu nhiên. Trong mọi thời điểm, một nút khởi tạo gói tin RREQ hoặc RREP nó thực hiện các tiến trình như sau:

- Khởi tạo một giá trị ngẫu nhiên *seed*

- Thiết lập trường Max\_Hop\_Count với giá trị *TimeToLive* (từ IP header)

*Max\_Hop\_Count = TimeToLive*

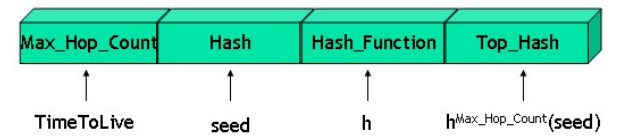
- Thiết lập trường Hash với giá trị seed vừa khởi tạo *Hash = seed*

- Thiết lập trường *Hash\_Function* định danh của hàm băm được sử dụng được cho bởi Bảng 3.1

|  |  |
| --- | --- |
| Value | Hash function |
| 0 | Dự trữ |
| 1 | MD5HMAC96 |
| 2 | SHA1HMAC96 |
| 3 - 127 | Dự trữ |
| 128 - 255 | Tùy thuộc cách triển khai |

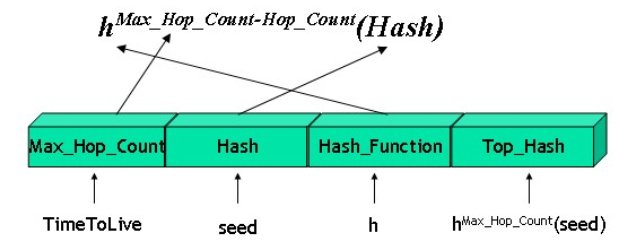
Bảng 3. 2 Các giá trị có thể của trường Hash\_Function

Giá trị hàm băm cao nhất (Top Hash) được tính nhờ sử dụng hàm băm “h” và giá trị khởi đầu ngẫu nhiên



Hình 3. 4 Cách tính hàm băm khi bắt đầu phát sinh RREQ hay RREP

Khi nút tiếp nhận RREQ hoặc RREP thì nó sẽ xác minh số chặng như sau: tính hàm băm h số lần là n [n = (Số chặng - Số chặng của nút hiện thời)] rồi so sánh với giá trị được chứa trong giá trị hàm băm cao nhất (*Top Hash*).



Hình 3. 5 Cách tính hàm băm tại nút trung gian

Nút trung gian sau khi đã xác minh được tính toàn vẹn và xác thực, thì nó sẽ chuẩn bị thông báo RREQ hoặc RREP.

Xác thực các trường không thay đổi của thông điệp bằng chữ ký số

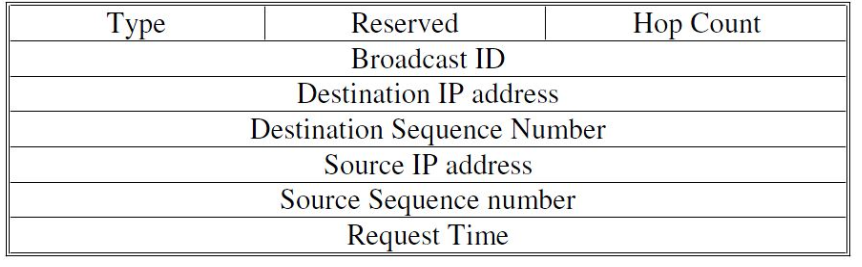
Khi một nút lần đầu tiên nhận RREQ, nó sẽ xác minh chữ ký trước khi tạo hoặc cập nhật một tuyến ngược lại tới nút nguồn. Khi RREQ đến được nút đích, RREP sẽ được gửi với một chữ ký RREP. Khi một nút nhận được RREP, thì đầu tiên nó sẽ xác minh chữ ký trước khi tạo hoặc cập nhật một tuyến tới nút nguồn. Chỉ khi chữ ký được xác minh, thì nút sẽ lưu lại tuyến với chữ ký của RREP và thời hạn tồn tại.

Khi việc thăm dò tuyến thành công, các nút nguồn và nút đích sẽ truyền thông theo các tuyến đã tìm được. Nếu xuất hiện một liên kết hỏng trong mạng do nút nào đó rời đi, thì thông báo lỗi tuyến RERR (Route Error) được tạo ra như trong AODV. RERR cũng được bảo vệ bằng chữ ký số.

### 3.3.3. RAODV (Reverse Ad hoc On-demand Distance Vector)

Giao thức RAODV được đề xuất bởi Chonggun Kim, Elmurod Talipov và Byoungchul Ahn, khám phá tuyến đường sử dụng thủ tục khám phá tuyến ngược nơi nút đích sẽ gửi một gói tin yêu cầu tuyến ngược (R-RREQ) tới các nút láng giềng để tìm một con đường hợp lệ đến nút nguồn sau khi nhận RREQ từ nút nguồn. Cơ chế hoạt động của RAODV như sau:

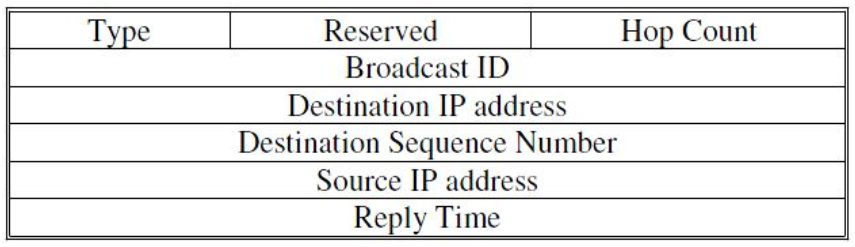
Giao thức RAODV không có tuyến đường cố định được lưu trữ trong các nút. Các nút nguồn khởi tạo thủ tục khám phá tuyến bằng cách quảng bá. Các gói tin RREQ chứa các thông tin sau: Loại tin, địa chỉ nguồn, địa chỉ đích, ID quảng bá, hop count, số thứ tự nguồn, số thứ tự đích, thời gian yêu cầu (timestamp).



Hình 3. 6 Định dạng gói tin RREQ

Mỗi khi nút nguồn gửi một RREQ mới, ID quảng bá được tăng lên một. Vì vậy, các địa chỉ nguồn và đích, cùng với các ID quảng bá, xác định duy nhất gói tin RREQ này. Nút nguồn quảng bá các RREQ đến tất cả các nút trong phạm vi truyền của nó, những nút láng giềng sẽ chuyển tiếp RREQ đến các nút khác. Khi RREQ được quảng bá trên toàn mạng, một số nút có thể nhận được nhiều bản sao của cùng một RREQ. Khi một node trung gian nhận RREQ, kiểm tra nút nếu đã nhận một RREQ với cùng ID quảng bá và nguồn địa chỉ. Nút mạng sẽ lấy ID quảng bá và địa chỉ nguồn lần đầu tiên và loại bỏ RREQ dư thừa. Thủ tục cũng tương tự với RREQ của AODV.

Khi nút đích nhận được gói tin yêu cầu tuyến đường lần đầu, nút đích sẽ tạo ra một gói tin yêu cầu ngược (R-RREQ) và quảng bá cho các nút hàng xóm trong phạm vi truyền giống như RREQ của nút nguồn đã làm. Gói tin R-RREQ chứa các trường thông tin sau: ID trả lời nguồn, ID trả lời đích, ID trả lời quảng bá, hop count, số thứ tự nguồn, thời gian trả lời (timestamp).



Hình 3. 7 Định dạng gói tin R-RREQ

Khi đã quảng bá gói tin R-RREQ đến nút trung gian, nó sẽ kiểm tra dự phòng. Nếu nút trung gian nhận được cùng một thông báo, gói tin đó sẽ bị hủy, nếu không sẽ chuyển tiếp tới các nút tiếp theo. Ngoài ra, nút mạng còn lưu trữ hoặc cập nhật các thông tin sau vào bảng định tuyến:

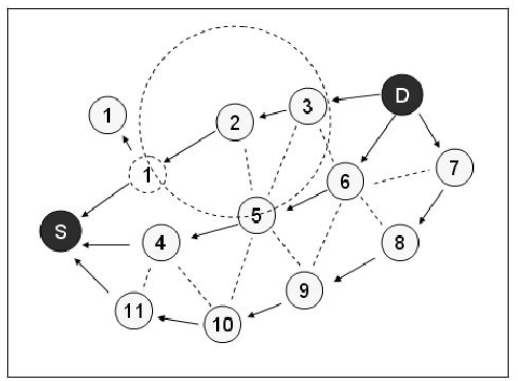
- Địa chỉ nút đích.

- Địa chỉ nút nguồn.

- Số hop đến nút đích.

- Số tuần tự đích.

- Thời gian tuyến đường hết hạn và hop tiếp theo đến nút đích Khi nào nút nguồn gốc nhận được gói tin R-RREQ đầu tiên, nó bắt đầu truyền gói tin dữ liệu, và các gói R-RREQ cuối sẽ được lưu lại để sử dụng sau. Các con đường thay thế có thể được sử dụng khi đường chính bị lỗi.



Hình 3. 8 Sơ đồ định tuyến RAODV

Trên hình 3.8, nút đích D không trả lời gói tin unicast ngay trước khi quyết định con đường ngược lại D -> 3 -> 2 -> 1 -> S. Thay vào đó, nút đích D sẽ gửi nhiều gói tin R-RREQ để tìm nút nguồn S. Và con đường chuyển tiếp tới đích được xây dựng thông qua R-RREQ này. Ví dụ: S -> 4 -> 5 -> 6 -> D, S -> 11->10->9 -> 8 ->7 -> D … Nút nguồn S sẽ chọn con đường tốt nhất và bắt đầu truyền dữ liệu.

### 3.3.4. IDSAODV (Intrusion Detection System Ad hoc On-demand Distance Vector)

Giao thức IDSAODV được đề xuất bởi Semih Dokurer, dựa trên ý tưởng hết sức đơn giản theo cơ chế làm việc của giao thức AODV đó là kiểm tra số SN của gói tin RREP trả lời.

Nếu trong mạng hiện diện nút lỗ đen thì ngay lập tức nút lỗ đen này sẽ trả lời gói tin RREP với giá trị số SN được gán cao nhất và đương nhiên sẽ trả lời ngay lập tức tới nút nguồn gửi yêu cầu RREQ. Do đó, chỉ cần loại bỏ gói tin RREP đầu tiên nhận được và chấp nhận gói tin RREP thứ hai với giá trị số SN cao nhất để thiết lập tuyến đường truyền thông bằng cơ chế bộ đệm gói tin. Tuy nhiên, trong một số trường hợp không phải bao giờ gói tin RREP với giá trị số SN lớn nhất nhận đầu tiên cũng đến từ nút lỗ đen, đó là khi nút đích 40 hay nút trung gian trả lời gói RREP với giá trị số SN lớn nhất có vị trí gần nút đích hơn so với nút lỗ đen.

# CHƯƠNG 4: SỬ DỤNG NS2 VÀ CÁC CÔNG CỤ LIÊN QUAN ĐỂ ĐÁNH GIÁ HIỆU NĂNG MẠNG VÀ THỰC HÀNH TẤN CÔNG VÀ CHỐNG LẠI BLACKHOLE TRONG GIAO THỨC AODV

## 4.2. Bộ mô phỏng NS-2 và cài đặt mô phỏng

### 4.2.1. Giới thiệu NS-2

**NS-2** là phần mềm mô phỏng mạng, hoạt động của nó được điều khiển bởi các sự kiện rời rạc.

NS-2 được thiết kế và phát triển theo kiểu hướng đối tượng, được phát triển tại đại học California, Berkely.

Bộ phần mềm này được viết bằng ngôn ngữ C++ và OTcl.

Ban đầu bộ mô phỏng này được xây dựng trên nền hệ điều hành Unix/Linux, sau này một số phiên bản có thể cài đặt để chạy trên môi trường Windows.

Việc sử dụng NS-2 rất linh hoạt, người dùng sau khi xây dựng xong các kịch bản mô phỏng của mình có thể cho nó chạy lặp đi lặp lại nhiều lần với các tham số cấu hình hệ thống khác nhau.

Tùy vào mục đích của người dùng đối với kịch bản mô phỏng mà kết quả mô phỏng có thể được lưu trữ vào tệp vết (trace file) với khuôn dạng (format) được những người phát triển NS định nghĩa trước hoặc theo khuôn dạng do người sử dụng NS quyết định khi viết kịch bản mô phỏng. Nội dung tệp vết sẽ được tải vào trong các ứng dụng khác để thực hiện phân tích. NS đã định nghĩa 2 loại tệp vết:

- Nam trace file (file.nam): Chứa các thông tin về tô-pô mạng như các nút mạng, đường truyền, vết các gói tin; dùng để minh họa trực quan mạng đã thiết lập.

- Trace file (file.tr): Tệp ghi lại vết của các sự kiện mô phỏng, tệp có định dạng text, có cấu trúc, dùng cho các công cụ lần vết và giám sát mô phỏng như: Gnuplot, XGRAPH hay TRACEGRAPH.

NS-2 hỗ trợ mô phỏng tốt cho cả mạng có dây và mạng không dây. Bao gồm các ưu điểm nổi bật sau:

- **Khả năng trừu tượng hóa:** Với bộ mô phỏng NS-2, chúng ta có thể thay đổi độ mịn các tham số đầu vào cho phù hợp với mục đích nghiên cứu.

**- Khả năng phát sinh kịch bản:** Bên trong phần mềm NS-2 bao gồm rất nhiều các gói bổ sung với nhiều công cụ hỗ trợ cho việc xây dựng các kịch bản như việc tạo mẫu lưu lượng, mô hình chuyển động của các nút.

**- Khả năng mô phỏng tương tác với mạng thực:** Đây là một điều khá đặc biệt ở bộ mô phỏng NS-2. Nó có một giao diện đặc biệt, cho phép lưu lượng thực đi qua nút mạng tương tác với bộ mô phỏng chạy trên nút mạng đó.

**- Khả năng hiển thị trực quan:** Trong bộ mô phỏng NS-2, có công cụ hiển thị NAM giúp chúng ta thấy được hình ảnh đã miêu tả trong kịch bản mô phỏng qua đó tìm ra điểm chưa hợp lý trong tham số đầu vào.

**- Khả năng mở rộng được:** Bộ mô phỏng NS-2 hỗ trợ khả năng mở rộng dễ dàng. Vì đây là phần mềm mã nguồn mở nên cộng đồng mã nguồn mở có thể đóng góp trực tiếp tùy theo mục đích nghiên cứu.

### 4.2.2. Các thành phần của bộ chương trình mô phỏng NS-2

Bộ mô phỏng mạng NS-2 gồm rất nhiều thành phần chức năng trong đó chương trình mô phỏng NS-2 là thành phần chính. Với chương trình mô phỏng NS-2 chúng ta có thể làm rất nhiều việc từ việc tạo ra các nút mạng, các đường truyền, các nguồn sinh lưu lượng theo các phân bố được định nghĩa trước, các thực thể giao thức khác nhau cho đến việc quản lý các chính sách hàng đợi cũng như các mô hình sinh lỗi của đường truyền…NS-2 hỗ trợ rất tốt các ứng dụng phổ biến hiện nay như web, FTP, telnet cũng như các giao thức giao vận phổ biến như TCP, UDP. Với mô phỏng mạng không dây, chương trình NS-2 hỗ trợ một số giao thức định tuyến mạng MANET phổ biến như AODV, DSDV, DSR hay TORA.

Ngoài chương trình NS-2 thì bộ mô phỏng NS-2 còn có các công cụ hiển thị trực quan là NAM và XGRAPH. Người nghiên cứu có thể nhìn thấy hình trạng mạng gồm các nút mạng, liên kết giữa các nút nhờ NAM. Ngoài ra, NAM cũng có thể cho phép hiển thị sự chuyển động của các gói số liệu trong không gian hai chiều. Một công cụ khác là XGRAPH có thể vẽ đồ thị từ các dữ liệu nhận được do chương trình mô phỏng sinh ra. Các số liệu sinh ra từ chương trình mô phỏng có thể được xử lý nhờ các ngôn ngữ như Awk hay Perl để thu được các kết quả mong muốn.

### 4.2.3. Các chức năng mô phỏng chính của NS-2

**\* Đối với mạng có dây:**

- Hỗ trợ các đường truyền điểm

- điểm đơn công, song công, mạng cục bộ LAN.

- Hỗ trợ các nguồn sinh lưu lượng với một số phân bố khác nhau.

- Hỗ trợ một số chính sách phục vụ hàng đợi.

- Hỗ trợ một số mô hình sinh lỗi điển hình.

–Hỗ trợ nghiên cứu vấn đề định tuyến đơn hướng/đa hướng (Unicast/Multicastrouting).

- Hỗ trợ các giao thức tầng giao vận: TCP/Tahoe/Reno/NewReno/Sack/Vegas, UDP, điều khiển lưu lượng và điều khiển tắc nghẽn

- Hỗ trợ một số giao thức tầng ứng dụng, web caching và truyền luồng dữ liệu đa phương tiện.

**\* Đối với mạng không dây:** Ngoài phần lớn các chức năng mô phỏng trên, NS-2 còn có khả năng: - Hỗ trợ việc di chuyển của các nút mạng trong không gian hai chiều.

- Hỗ trợ mạng LAN không dây (WLAN) 802.11.

- Hỗ trợ Mobile IP.

- Hỗ trợ một số giao thức định tuyến trong mạng không dây đặc biệt MANET: AODV, DSDV, DSR, TORA...

- Hỗ trợ liên mạng sử dụng vệ tinh (Satellite Networking).

**\* Trong lĩnh vực mạng hỗn hợp có dây và không dây:**

- Hỗ trợ các trạm cơ sở đóng vai trò cổng chuyển giữa mạng có dây và mạng không dây.

- Hỗ trợ Snoop TCP.

### 4.2.4. Thiết lập mô phỏng mạng MANET trong NS-2

Mô hình không dây chủ yếu bao gồm nút di động (MobileNode) ở lõi, với các đặc điểm hỗ trợ mở rộng cho phép mô phỏng các mạng MANET, mạng không dây… Đối tượng MobileNode là một đối tượng tách biệt. Lớp MobileNode được thừa kế từ lớp cha Node trong C++. Do đó lớp MobileNode là đối tượng Node cơ bản với các chức năng thêm vào của nút không dây và di động như là khả năng di chuyển trong hình trạng mạng cho trước, khả năng nhận và truyền các tín hiệu tới và từ kênh không dây. Trong mục này chúng ta sẽ mô tả các thành phần bên trong của MobileNode bao gồm các kỹ thuật định tuyến, việc tạo ngăn xếp mạng cho phép truy cập kênh trong MobileNode, mô tả ngắn gọn của mỗi thành phần ngăn xếp, hỗ trợ dò vết và tạo ra các ngữ cảnh di chuyển/lưu lượng cho mô phỏng không dây.

**\*Nút di động:** Nút di động (MobileNode) là một đối tượng ns Node cơ bản với các chức năng mở rộng như sự di chuyển, khả năng truyền và nhận trên kênh mà cho phép nó có thể sử dụng để tạo sự di động trong các môi trường mô phỏng không dây. Các đặc điểm di động bao gồm việc di chuyển nút, các cập nhật vị trí định kỳ, bảo đảm đường biên của hình trạng mạng được thực thi trong C++ trong khi các thành phần mạng trong bản thân MobileNode (như bộ phân loại, dmux, LL, Mac, Channel…) lại được thực thi trong Otcl.

Giao thức định tuyến mạng không dây được hỗ trợ trong NS-2 bao gồm AODV, DSDV, DSR và TORA. Việc tạo các nút di động phụ thuộc vào giao thức định tuyến được sử dụng. Hiện nay NS-2 hỗ trợ API tạo nút di động được mô tả như sau:

*$ns\_ node-config*

*-adhocRouting $opt(adhocRouting) \*

*-llType $opt(ll) \*

*-macType $opt(mac) \*

*-ifqType $opt(ifq) \*

*-ifqLen $opt(ifqlen) \*

*-antType $opt(ant) \*

*-propType $opt(prop) \*

*-propInstance [new $opt(prop)] \*

*-phyType $opt(netif) \*

*-channel [new $opt(chan)] \*

*-topoInstance $topo \*

*-agentTrace ON \*

*-routerTrace ON \*

*-macTrace ON \*

API ở trên cấu hình cho nút di động với tất cả các giá trị của giao thức định truyến trong MANET như ngăn xếp mạng, cấu trúc mạng, mô hình kênh truyền. Tiếp đến chúng ta tạo các nút di động như sau:

*for {set i 0} {$i < $val(nnaodv)} {incr i} {*

*set node\_($i) [$ns\_ node]*

*$node\_($i) random-motion 0; # disable random motion*

*}*

Thủ tục trên tạo ra các đối tượng nút di động, tạo tác tử định tuyến mạng MANET, tạo ngăn xếp mạng bao gồm lớp liên kết, hàng đợi, lớp MAC và giao diện mạng với ăng ten, sử dụng mô hình truyền thông được định nghĩa, kết nối các thành phần và kết nối ngăn xếp vào kênh.

**\* Quy trình mô phỏng mạng MANET với NS-2:**

Quá trình mô phỏng mạng MANET với bộ mô phỏng NS-2 thông thường trải qua các bước như sau trong việc xây dựng tệp kịch bản:

**- Tạo bộ lập lịch các sự kiện**

+ Tạo bộ lập lịch

*set ns\_ [new Simulator]*

+ Lập lịch sự kiện

*$ns at <time><event>*

+ Khởi động bộ lập lịch

*$ns run*

**- Ghi lại vết các sự kiện của mạng mô phỏng**

+ Dò vết các gói tin trên tất cả các liên kết, xuất ra tệp traceout.tr

$ns trace-all [open traceout.tr w]

+ Dò vết các gói tin trên tất cả các liên kết, xuất ra định dạng dùng trong NAM

$ns namtrace-all [open traceout.nam w]

+ Cũng có thể dò vết một số tham số

ví dụ:

$ns trace-queue $n0 $n1 $ns nam trace-queue $n0 $n1

**- Thiết lập mạng mô phỏng**

+Thiết lập tô-pô mạng

set topo [new Topography]

$topo load\_flatgrid $val(x) $val(y) # Create God create-god $val(nn)

**- Cấu hình nút di động**

+ Cấu hình nút di động

$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\_

+ Thiết lập vị trí ban đầu

$node\_(0) set X\_ 1.0 $node\_(0) set Y\_ 3.0

$node\_(0) set Z\_ 0.0 $node\_(1) set X\_ 4.0

$node\_(1) set Y\_ 6.0 $node\_(1) set Z\_ 0.0

+ Thiết lập sự di chuyển cho các nút di động ta dùng

$ns\_ at 10.0 “$node\_(0) setdest 20.0 10.0 15.0”

**- Tạo ra các nguồn sinh lưu lượng**

+ Tạo lưu lượng

set udp\_(0) [new Agent/UDP]

$ns\_ attach-agent $node\_(0) $udp\_(0)

set null\_(0) [new Agent/Null]

$ns\_ attach-agent $node\_(1) $null\_(0)

set cbr\_(0) [new Application/Traffic/CBR]

$cbr\_(0) set packetSize\_ 512

$cbr\_(0) set interval\_ 4.0

$cbr\_(0) set random\_ 1

$cbr\_(0) set maxpkts\_ 10000

$cbr\_(0) attach-agent $udp\_(0)

$ns\_ connect $udp\_(0) $null\_(0)

$ns\_ at 158.7047093914378 "$cbr\_(0) start"

+ Việc tạo lưu lượng có thể lấy từ tệp sinh lưu lượng của công cụ có sẵn  
cbrgen trong bộ mô phỏng NS-2.

## 4.3. Cài đặt bổ sung các giao thức

Trong phiên bản NS-2.35 đã hỗ trợ các giao thức định tuyến trong mạng MANET như AODV, DSDV, DSR… Tuy nhiên, không hỗ trợ sẵn giao thức mô phỏng lỗ đen, chống tấn công lỗ đen như IDSAODV và RAODVnên chúng ta buộc phải cài bổ sung cho chúng mới dùng được.

## 4.3.1. Cài đặt giao thức blackholeAODV mô phỏng tấn công lỗ đen

Chúng ta cần sửa đổi 2 file là aodv.h và aodv.cc

Đầu tiên chúng ta vào đường dẫn đã cài ns2.35 là ns-allinone-2.35/ns-2.35/aodv

|  |
| --- |
| **Sửa file aodv.h**  Tại dòng 276 chúng ta thêm  bool malicious; // Added for Blackhole Attack  **Sửa file aodv.cc**  Tại dòng 85 ta thêm  // Added for Blackhole Attack  if(strncasecmp(argv[1], "hacker", 6) == 0) {  malicious = true;  return TCL\_OK;  }  Tại dòng 154 ta thêm  malicious = false; // Added for Blackhole Attack  Tại dòng 454 ta thêm  // Added for Blackhole Attack  // If the node is a malicious node - drop the packet and specify a reason for dropping it!  if (malicious == true) {  drop(p, DROP\_RTR\_ROUTE\_LOOP); // DROP\_RTR\_ROUTE\_LOOP is added for no reason.  return;  }  // Added for Blackhole Attack  Tại dòng 789 ta thêm  // Added for Blackhole Attack  else if (malicious == true) {  seqno = max(seqno, rq->rq\_dst\_seqno)+1;  if (seqno%2) seqno++;  sendReply(rq->rq\_src, // IP Destination  1, // Hop Count is set to 1 to confuse the source node!  rq->rq\_dst, // Dest IP Address  seqno, // Dest Sequence Num  MY\_ROUTE\_TIMEOUT, // Lifetime  rq->rq\_timestamp); // timestamp  Packet::free(p);  }  // Added for Blackhole |

Ở file aodv.h tạo bool malicious mục đích để khởi tạo 1 biến gồm 2 giá trị true, false kích hoạt, hoặc vô hiệu hóa.

ở trong file aodv.cc

Tại dòng 85 ta thêm

// Added for Blackhole Attack

if(strncasecmp(argv[1], "hacker", 6) == 0) {

malicious = true;

return TCL\_OK;

}

Mục đích bắt trường hợp blackhole sẽ kích hoạt biến malicious là true;

Tại dòng 454 ta thêm

// Added for Blackhole Attack

// If the node is a malicious node - drop the packet and specify a reason for dropping it!

if (malicious == true) {

drop(p, DROP\_RTR\_ROUTE\_LOOP); // DROP\_RTR\_ROUTE\_LOOP is added for no reason.

return;

}

Nếu phát hiện node là blackhole thì nó sẽ hủy tất các gói đến nó.

Tại dòng 789 ta thêm

// Added for Blackhole Attack

else if (malicious == true) {

seqno = max(seqno, rq->rq\_dst\_seqno)+1;

if (seqno%2) seqno++;

sendReply(rq->rq\_src, // IP Destination

1, // Hop Count is set to 1

rq->rq\_dst, // Dest IP Address

seqno, // Dest Sequence Num

MY\_ROUTE\_TIMEOUT, // Lifetime

rq->rq\_timestamp); // timestamp

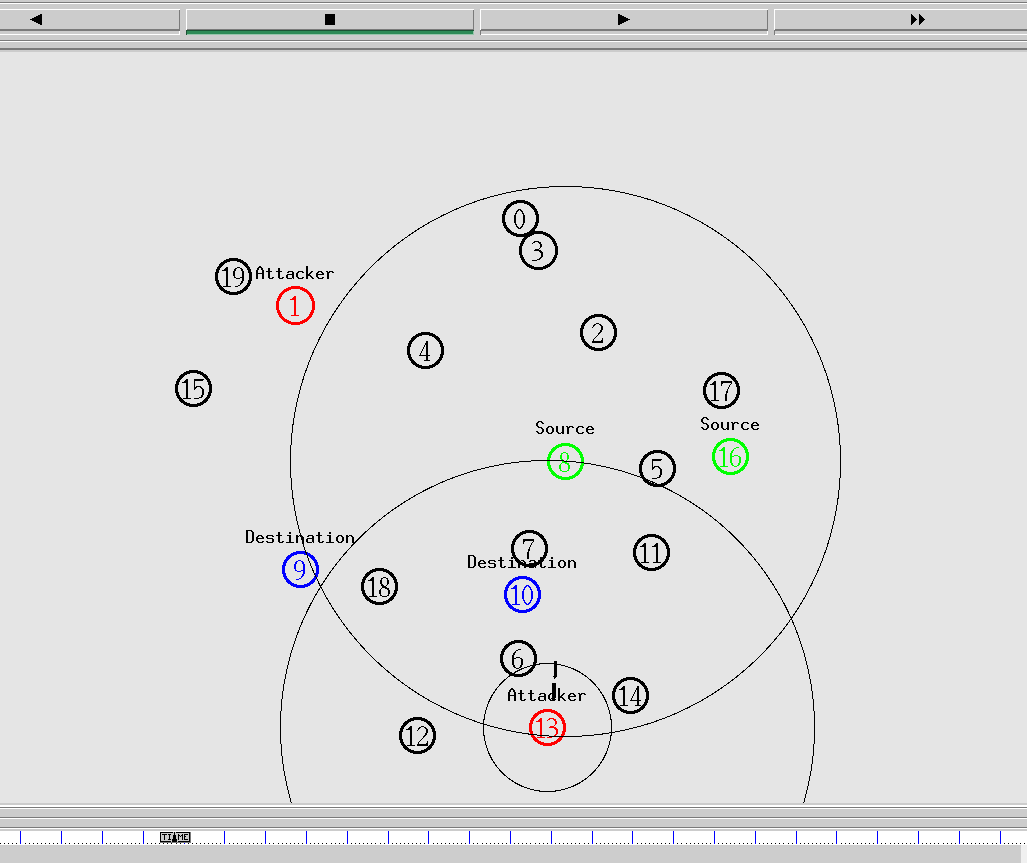
Packet::free(p);

}

Nút độc hại cố gắng đánh lừa các nút gửi bằng gói RREP của mình với giá trị của số SN được gán giá trị cao nhất và hop count gán bằng 1. Giá trị cụ thể của gói tin sai RREP được minh họa ở trên

Sau khi thực hiện những thay đổi trên, chúng ta tiến hành biên dịch lại NS-2 bằng cách vào ns-allinone-2.35/ns-2.35/aodv gõ makeclean, make để biên dịch lại toàn bộ chương trình.

Kết quả ta nhận được:



Chúng ta có thể dùng trình tracegraph hay awk , trong ví dụ này tôi dùng perl để phân tích file, Perl là một ngôn ngữ thông dịch có rất nhiều tác dụng, nhưng những lợi ích quan trọng nhất khi sử dụng Perl đó là tìm kiếm, trích tách và báo cá.

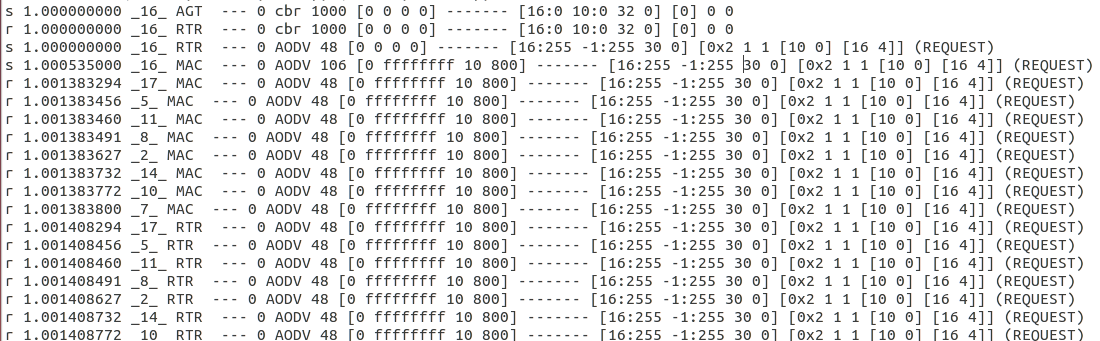
Chúng ta tạo 1 file **analyze.pl** với nội dung như sau:

|  |
| --- |
| #!/usr/bin/perl  $tracefile=$ARGV[0];  $granularity=$ARGV[1];  $ofile="simulation\_result.csv";  open OUT, ">$ofile" or die "$0 cannot open output file $ofile: $!";  open (DR,STDIN);  $gclock=0;  #Data Packet Information  $dataSent = 0;  $dataRecv = 0;  $routerDrop = 0;  #AODV Packet Information  $aodvSent = 0;  $aodvRecv = 0;  $aodvSentRequest = 0;  $aodvRecvRequest = 0;  $aodvDropRequest = 0;  $aodvSentReply = 0;  $aodvRecvReply = 0;  $aodvDropReply = 0;  if ($granularity==0) {$granularity=30;}  while(<>){  chomp;  if (/^s/){  if (/^s.\*AODV/) {  $aodvSent++;  if (/^s.\*REQUEST/) {  $aodvSendRequest++;  }  elsif (/^s.\*REPLY/) {  $aodvSendReply++;  }  }  elsif (/^s.\*AGT/) {  $dataSent++;  }    } elsif (/^r/){  if (/^r.\*AODV/) {  $aodvRecv++;  if (/^r.\*REQUEST/) {  $aodvRecvRequest++;  }  elsif (/^r.\*REPLY/) {  $aodvRecvReply++;  }    }  elsif (/^r.\*AGT/) {  $dataRecv++;  }  #############################################################  ###########################################################    } elsif (/^D/) {  if (/^D.\*AODV/) {  if (/^D.\*REQUEST/) {  $aodvDropRequest++;  }  elsif (/^D.\*REPLY/) {  $aodvDropReply++;  }    }  if (/^D.\*RTR/) {  $routerDrop++;  }  }    }  close DR;  $delivery\_ratio = 100\*$dataRecv/$dataSent;  print "AODV Sent : $aodvSent\n";  print "AODV Recv : $aodvRecv\n";  print "Data Sent : $dataSent\n";  print "Data Recv : $dataRecv\n";  print "Router Drop : $routerDrop\n";  #print "Energy Drop : $routerDrop\n";  print "Delivery Ratio : $delivery\_ratio \n";  print OUT "Messages Sent,$dataSent\n";  print OUT "Messages Recieved,$dataRecv\n";  print OUT "Messages Dropped,$routerDrop\n";  print OUT "Delivery Rate,$delivery\_ratio\n";  close OUT; |

Chúng ta cần hiểu thêm về file .tr như sau:

File này nó sẽ lưu dấu vết các gói tin đã gửi, nhận, hủy, tầng thực thi là tầng nào, gửi dung lượng bao nhiêu, kiểu gửi gói, …

Chẳng hạn như sau:

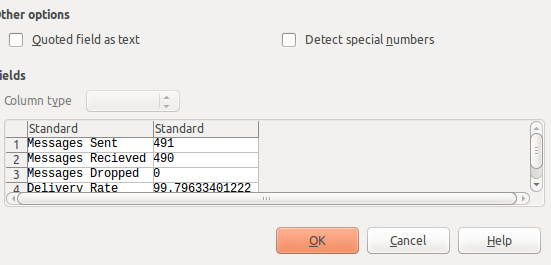


Mời các bạn xem thêm ở phần chú thích.

Và sau khi thực hiện câu lệnh

|  |
| --- |
| sonlanggtu@ubuntu:~/Desktop/tcl$ **perl analyze.pl blackhole.tr** |

Nó sẽ phân tích cho chúng ta, và ghi ra cho 1 file **simulation\_result.csv** file này chứa toàn bộ nội dung phân tích mà chúng ta cần



**4.4 Mô phỏng, đánh giá ảnh hưởng và giải pháp làm giảm hiệu ứng của tấn công lỗ đen**

**4.4.1 Các độ đo hiệu năng được đánh giá**

- **Tỷ lệ phân phát gói tin thành công (PDR - Packet Delivery Ratio):** Là tỷ lệ giữa số gói tin được phân phát thành công tới đích so với số gói tin được gửi đi nguồn phát. Giá trị của tỉ số này càng lớn chứng tỏ hiệu năng hoạt động của mạng càng cao.

- **Độ trễ trung bình (End-to-End Delay):** Thời gian trung bình để gửi một gói dữ liệu tới đích. Nó bao gồm cả độ trễ gây ra bởi quá trình khám phá tuyến và dữ liệu truyền tải nằm trong hàng đợi. Chỉ các gói dữ liệu được truyền thành công tới đích mới được tính.

- **Tổng phí (Routing Overhead - Throughput):** Là tổng của tất cả các gói tin điều khiển như gói tin yêu cầu tuyến, gói tin trả lời tuyến và các gói tin lỗi trên đường truyền từ cả hai phía nguồn và đích.

**4.4.2 Thiết lập các lựa chọn, các tham số mô phỏng**

Như đã trình bày ở trên, do đặc thù của mạng MANET là mạng động với băng thông thấp, năng lượng của nút hạn chế cho nên khi nghiên cứu về ảnh hưởng của tấn công lỗ đen trong mạng MANET ta cần chú ý đến một số lựa chọn, tham số mô phỏng như:

- **Kích thước mạng (độ lớn của mạng):** Số lượng nút trong mạng

- **Mật độ nút:** Tính theo số hàng xóm trung bình hoặc theo số nút trung bình trong diện tích phủ sóng (radio range) của một nút.

- **Độ linh động của mạng:** Được đo bằng tốc độ chuyển động trung bình của các nút mạng.

- **Các mẫu lưu lượng:** Hệ thống với các mẫu lưu lượng như CBR (truyền bằng giao thức UDP) hoặc FTP (truyền bằng giao thức TCP)

**4.4.3 Các thông số mô phỏng**

|  |  |
| --- | --- |
| **Thông số** | **Giá trị** |
| *Cấu hình chung* | |
| Khu vực địa lý | 1186 x 600 m2 |
| Tổng số nút | 20, 25, 30 nút |
| Vùng thu phát sóng | 250 m |
| *Cấu hình di chuyển* | |
| Tốc độ di chuyển nhanh nhất |  |
| Tốc độ di chuyển chậm nhất | 0 m/s |
| *Cấu hình truyền dữ liệu* | |
| Nguồn sinh lưu lượng | CBR |
| Số kết nối | 20 |
| Kích thước gói tin | 1000Mb |
| Tốc độ phát gói | 0.1MB |

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. **KẾT LUẬN**

Ngày nay, mạng không dây trong đời sống con người đang ngày càng đóng vị trí quan trọng. Trong số các mạng không dây, mạng Ad-Hoc được quan tâm một cách đặc biệt. Không giống như mạng có dây truyền thống hay mạng không dây có cơ sở hạ tầng, với việc dễ dàng thiết lập định, chi phí hoạt động thấp, triển khai nhanh chóng và có tính di động cao nên mạng AdHoc đang được ứng dụng và có thể làm thay đổi về giao thông, cũng như mọi mặt của đời sống. Cùng với đó việc bảo mật dữ liệu trong mạng ad hoc cũng cần đặc biệt quan tâm cần phát triển các cơ chế bảo mật của các giao thức định tuyến, vì các giao thức định tuyến trong mạng MANET tiềm ẩn nhiều rủi ro dễ bị tấn công, sửa đổi, cũng như giả mạo dễ bị khai thác sơ hở.

Bài báo cáo đã chỉ ra những vấn đề, những nguy cơ tìm ẩn cũng như các giải pháp để đảm bảo các tính an toàn thông tin trong mạng MANET, và cũng đã khai thác cụ thể được lỗ hổng blackhole và cách chống lại lỗ hổng này.

1. **HƯỚNG PHÁT TRIỂN**

Mạng MANET là mạng có tiềm năng, ứng dụng cao sẽ rất phát triển sau này. Tuy nhiên đi đôi với nó chúng ta cần nhiều nhà nghiên cứu hơn cũng như các giải pháp tốt hơn để vừa đảm bảo được tốc độ truyền gói tin tốt vừa đảm bảo an toàn thông tin.

Sau bài báo cáo này em sẽ tích cực nghiên cứu thêm về các lỗ hổng khác trong mạng MANET cũng như tìm ra thuật toán đảm bảo an toàn thông tin vừa có tốc độ truyền gói tốt.

# TÀI LIỆU THAM KHẢO

1. Đánh giá về an toàn giao thức định tuyến trong mạng MANET – Ngô thế hải anh

2. C.Perkins,“(RFC) Request for Comments 3561”, Category: Experimental, Network, Working Group, July 2003

3. Dr.Satya Prakash Singh, Ramveer Singh (2012), “Security challenges in mobile adhoc network”, International Journal of Applied Engineering Research, Volume 7 (11)

4. S. Mehla, B. Gupta, P. Nagrath, “Analyzing Security of Authenticated Routing Protocol (ARAN)”, International Journal of Computer Science and Engineering (IJCSE), Vol. 02, No. 03, 2010, 664-668, 2010

5. Manel Guerrero Zapata (2002), “Secure Ad hoc On-Demand Distance Vector Routing”, ACM Mobile Computing and Communications Review (MC2R), 6(3):106-107, July 2002

6. C. Kim, E. Talipov, and B. Ahn, “A Reverse AODV Routing Protocol in Ad Hoc Mobile Networks”, The International Conference on Emerging Directions in Embedded and Ubiquitous Computing (EUC’06), Seoul, 1-4 August 2006, pp.522-531. Springer, 2006

7. S. Dokurer, “Simulation of Black hole attack in wireless ad-hoc networks” Thesis Master in Computer Engineering Atihm University, September 2006 8. F.J.Ros and P.M.Ruiz (2004), “Implementing a New Manet Unicast Routing Protocol in NS-2”, December, 2004

9. C. P. Vandana and A. F. S. Devaraj, “MLDW - A MultiLayered Detection mechanism for Wormhole attacks in AODV based MANET”, in International Journal of Security, Privacy and Trust Management (IJSPTM) vol. 2, no. 3, (2013) June

10. H. Deng, W. Li and D. P. Agrawal (2002), “Routing Security in Wireless Ad Hoc Networks”,University of Cincinnati, IEEE Communication Magazine, October 2002

11. The VINT Project, “The NS manual”, A Collaboration between researches at UC Berkeley, LBL, USC/ISI, and Xerox PARC, March 14,2008

12. <https://en.wikipedia.org/wiki/Open_Shortest_Path_First>

13. Elizabeth M.Royer, C-K Toh (1999), “A review of current routing protocol for Ad hoc Mobile Wireless Networks”, IEEE Personal Communications, 6(2):46–55, April 1999

14. Charles E. Perkins, Pravin Bhagwat, “Highly Dynamic DestinationSequenced Distance Vector Routing (DSDV) for Mobile Computers”, Comp Commun Rev, Oct. 1994

15. David B. Johnson, David A. Maltz, Josh Broch, “DSR: The Dynamic Source Routing Protocol for Multi-hop Wireless Ad Hoc Networks”, t. Imielinski and H.Korth , Eds Kluwer, 1996

16. Jathe S.R., Dakhane D.M. (2012), “A Review Paper on Black Hole Attack and Comparison of Different Black Hole Attack Tech-niques”, International Journal of Cryptography and Security, ISSN: 2249-7013 & E-ISSN: 2249-7021, Volume 2, Issue 1, pp.-22-26

17. Kimaya Sanzgiri, Bridget Dahill, Brian N. Levine, and Elizabeth M. Belding-Royer, “A Secure Routing Protocol for Ad Hoc Networks”. Proceedings of 10th IEEE International Conference on Network Protocols (ICNP’02), Paris, France, November 2002, pp. 78-90

18. Yi-an Huang and Wenke Lee, “Attack analysis and Detection for Ad-hoc Routing protocols”. Proceedings of the 7th International Symposium on Recent Advances in Intrusion Detection (RAID'04), French Riviera, France. September 2004

19. Mishra Amitabh, Nadkarni Ketan M., Ilyas Mohammad, “Chapter 30: Security in wireless ad-hoc networks, the handbook of Ad hoc wireless network”, CRC PRESS Publisher, 2003

20. C.Siva Ram Murthy, B. S. Manoj, “Ad hoc wireless networks: Architecture and Protocols”. Prentice Hall Publishers, May 2004, ISBN 013147023X

21. William Stallings, “Network Security essentials: Application and Standards”, Pearson Education , Inc 2003, ISBN 0130351288

22. Preeti Sachan, Pabitra Mohan Khilar (2011), “Securing AODV routing protocol in MANET based cryptographic authentication”, International Journal of Network Security & Its Applications, Volume 3 (5)