BÁO CÁO THỰC HÀNH

**Bài thực hành số 02: Routing Protocol trong mạng MANET**

**Môn học:** Hệ Thống Nhúng & Mạng Không Dây

**Lớp:** NT131.O12.MMCL

**THÀNH VIÊN THỰC HIỆN:**

|  |  |  |
| --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** |
| 1 | Lê Hoàng Khánh | 21522205 |
| 2 | Ngô Thế Anh | 21521826 |

|  |
| --- |
| **Điểm tự đánh giá** |
| **10** |

**ĐÁNH GIÁ KHÁC:**

|  |  |
| --- | --- |
| Tổng thời gian thực hiện | 2 tuần. |
| Phân chia công việc | Thế Anh: AODV Routing, hiệu chỉnh báo cáo.  Hoàng Khánh: OLSR Routing, chạy mô phỏng. |
| Ý kiến *(nếu có)*  + Khó khăn  + Đề xuất, kiến nghị |  |

MỤC LỤC

[**A.** OLSR Routing 4](#_Toc149612631)

[1. Compile và chạy mô phỏng, quan sát quá trình chạy của kịch bản. Thu được các file pcap. 4](#_Toc149612632)

[2. Địa chỉ IP của Node 0, Node 1, Node 2, Node 3 lần lượt là gì? 6](#_Toc149612633)

[3. Xác định các gói OLSR thể hiện quá trình định tuyến giữa các Node 0, Node 1, Node 2 từ giây thứ 1 đến giây thứ 20. 7](#_Toc149612634)

[4. Địa chỉ IP đích của các gói tin OLSR từ các Node là gì? Chúng có ý nghĩa như thế nào? Giải thích. 8](#_Toc149612635)

[a. Địa chỉ IP đích. 8](#_Toc149612636)

[b. Ý nghĩa. 8](#_Toc149612637)

[5. Tìm kiếm trường thông tin TTL của các gói tin OLSR, cho biết chúng có điều gì đặt biệt? Từ đó rút ra ý nghĩa tại sao chúng có giá trị đó? 9](#_Toc149612638)

[a. TTL là gì? 9](#_Toc149612639)

[b. Trường TTL của các gói tin OLSR. 9](#_Toc149612640)

[c. Ý nghĩa. 9](#_Toc149612641)

[6. Các gói tin OLSR này định kỳ bao nhiêu thời gian sẽ gửi lại và thông điệp mà chúng gửi là gì đến giây thứ 10? Tìm kiếm trường thông tin Neighbor Address, giá trị của chúng là bao nhiêu? Từ đó, suy ra ý nghĩa của thông điệp này? 10](#_Toc149612642)

[7. Tìm kiếm thông tin Link Type: Symmetric Link và Asymetric Link. Cho biết chúng hoạt động như thế nào? 12](#_Toc149612643)

[8. Từ giây thứ 10 trở đi, tìm các MPR Link? Giải thích ý nghĩa của MPR và mô tả quá trình xảy ra đối với kịch bản? 13](#_Toc149612644)

[a. Các MPR Link. 13](#_Toc149612645)

[b. Giải thích và ý nghĩa. 15](#_Toc149612646)

[c. Mô tả kịch bản. 16](#_Toc149612647)

[9. Tìm các thông điệp TC, cho biết thông điệp này đang cố quảng bá điều gì? 17](#_Toc149612648)

[a. Các thông điệp TC. 17](#_Toc149612649)

[b. Giải thích. 18](#_Toc149612650)

[10. Xác định các gói tin UDP từ Node 0 → Node 1 trước giây thứ 20. 19](#_Toc149612651)

[11. Xác định các gói tin OLSR, ARP thể hiện lại quá trình tái định tuyến giữa Node 0 và Node 1 sau giây thứ 20 khi Node 3 đi vào vùng giữa Node 0 và Node 1 và Node 2 di chuyển ra ngoài vùng này. Nêu rõ ý nghĩa của các gói ARP lúc này và các gói OLSR thể hiện điều gì? 20](#_Toc149612652)

[12. Tăng khoảng cách của Node 0 và Node 1 lên 350m. Điều chỉnh vị trí ban đầu của Node 3, vận tốc của Node 2 và Node 3 sao cho ở giây thứ 15 cả hai Node 2 và Node 3 sẽ gặp nhau ở cùng một tọa độ x. 22](#_Toc149612653)

[a. Sửa đổi code. 22](#_Toc149612654)

[b. Xác nhận lại qua mô phỏng: 23](#_Toc149612655)

[**B.** AODV Routing 24](#_Toc149612656)

[1. Compile và chạy mô phỏng, quan sát quá trình chạy của kịch bản. Thu được các file pcap. 24](#_Toc149612657)

[2. Địa chỉ IP của Node 0, Node 1, Node 2, Node 3 lần lượt là gì? 26](#_Toc149612658)

[3. Xem file pcap của Node 3, xác định các gói AODV thể hiện quá trình định tuyến giữa các Node 0, Node 3, Node 1 từ giây thứ 1 đến 13. 27](#_Toc149612659)

[4. Khoảng bao nhiêu giây thì gồm Node 0, Node 2, Node 3 gửi gói Route Reply đến địa chỉ Broadcast. 28](#_Toc149612660)

[5. Tìm gói tin Route Request thể hiện quá trình tìm đường đi từ Node 0 đến Node 1. Thông tin nào thể hiện quá trình đó? 29](#_Toc149612661)

[6. Tìm các gói tin Route Reply thể hiện quá trình kịch bản đã khám phá được đường đi từ Node 0 đến Node 1 và ngược lại. Thông tin nào giúp ta xác định được điều đó? Giải thích. 30](#_Toc149612662)

[a. Quá trình tìm đường trong AODV 30](#_Toc149612663)

[b. Quá trình tìm đường từ Node 0 đến 1 trong kịch bản. 30](#_Toc149612664)

[7. Sau giây thứ 13, tìm các gói tin Route Error. Tại sao có thông điệp này xảy ra? Trường thông tin nào chứng minh được điều đó? 32](#_Toc149612665)

[8. Xác định các gói tin AODV, ARP thể hiện lại quá trình tái định tuyến giữa Node 0 và Node 1 sau giây thứ 13 khi Node 3 đi ra khỏi vùng truyền thông của Node 0 và Node 1. Mô tả lại quá trình đó. 33](#_Toc149612666)

[**C.** TÀI LIỆU THAM KHẢO 35](#_Toc149612667)

# OLSR Routing

## Compile và chạy mô phỏng, quan sát quá trình chạy của kịch bản. Thu được các file pcap.

Tái sử dụng môi trường ns3 đã build được từ Lab 01.

1. Tại folder **ns3-3.40** copy file **lab2-olsr.cc** và **myapp.h** được cung cấp vào folder **scratch**.
2. Re-build lại ns3 bằng lệnh.

$ ./ns3 build

1. Sau khi đã Re-build xong, để chạy lab2-olsr.cc kèm theo mô phỏng ta thực hiện lệnh.

$ ./ns3 run scratch/lab2-olsr.cc -–vis

Sau khi chạy ta được giao diện mô phỏng như hình.

A screenshot of a video game

Description automatically generated

Quan sát mô phỏng ta nhận thấy kịch bản chạy giống như mô tả của đề bài như sau:

A diagram of a process

Description automatically generated

Sau khi chạy mô phỏng sau qua giây 35 ta tắt mô hỏng, các file PCAP sẽ được phần mềm tự động tạo ra với số thứ tự ứng với từng node.

A screen shot of a computer

Description automatically generated

## Địa chỉ IP của Node 0, Node 1, Node 2, Node 3 lần lượt là gì?

A black background with white text

Description automatically generated

Ta có biến c kiểu NodeContainer. NodeContainer là một tập hợp các node. Tạo 4 node trong mạng và thêm chúng vào biến c.



Cài đặt các thiết bị mạng cho các node trong biến **c.**

A black screen with white text

Description automatically generated

Cài đặt địa chỉ của mạng sẽ là 10.1.1.0, từ đó các node sẽ được gặn địa chỉ tăng dần theo từng Node, IP 10.1.1.1 cho Node 0, địa chỉ IP 10.1.1.2 cho Node 1, địa chỉ IP 10.1.1.3 cho Node 2 và địa chỉ IP 10.1.1.4 cho Node 3.

A computer screen shot of text

Description automatically generated

Từ các lệnh cài đặt vị trí ta có thể khẳng định rằng thứ tự các Node trong kịch bản tương đương với thứ tự các Node từ đề bài.

Tóm lại địa chỉ IP của Node 0, Node 1, Node 2, Node 3 lần lượt là:

* Node 0: 10.1.1.1
* Node 1: 10.1.1.2
* Node 2: 10.1.1.3
* Node 3: 10.1.1.4

## Xác định các gói OLSR thể hiện quá trình định tuyến giữa các Node 0, Node 1, Node 2 từ giây thứ 1 đến giây thứ 20.

Các gói tin OLSR thể hiện quá trình địch tuyến giữa các Node 0, Node 1, Node 2 được xác định trong file PCAP của Node 2:

A screenshot of a computer

Description automatically generated

## Địa chỉ IP đích của các gói tin OLSR từ các Node là gì? Chúng có ý nghĩa như thế nào? Giải thích.

### Địa chỉ IP đích.

A red rectangular object with numbers

Description automatically generated

Có thể thấy địa chỉ IP đích của các gói tin OLSR là 10.1.1.255 và đây là địa chỉ Boardcast của mạng.

### Ý nghĩa.

Sử dụng địa chỉ IP địch là địa chỉ BoảdCast thể hiện gói tin sẽ được gửi đến tất các các Node trong mạng.

Điều này là cần thiết vì OLSR là một giao thức định tuyến dựa trên trạng thái, trong đó các Node trao đổi thông tin về trạng thái kết nối của chúng với nhau. Để đảm bảo rằng tất cả các Node đều nhận được thông tin này, các gói tin OLSR được gửi đến địa chỉ Boardcast.

## Tìm kiếm trường thông tin TTL của các gói tin OLSR, cho biết chúng có điều gì đặt biệt? Từ đó rút ra ý nghĩa tại sao chúng có giá trị đó?

### TTL là gì?

TTL là viết tắt của Time To Live, xác định số Hop tối đa mà gói tin có thể đi qua trước khi bị loại bỏ. Khi một node nhận được một gói tin OLSR, nó sẽ giảm TTL của gói tin đi 1. Nếu TTL bằng 0, gói tin sẽ bị loại bỏ.

### Trường TTL của các gói tin OLSR.

A screenshot of a computer

Description automatically generated

Có thể thầy TTL của gói tin HELLO là 1. Vì gói tin HELLO trong OLSR chỉ được gửi cho các Node lân cận cách 1 Hop.

A screenshot of a computer

Description automatically generated

Có thể thấy TTL của gói tin TC là 255. Vì gói tin TC không có ràng buộc về số Hop nên đây có thể là giá trị TTL mặc định của ns3.

### Ý nghĩa.

Trong giao thức OLSR, TTL được sử dụng để kiểm soát luồng lưu lượng trong mạng. Bằng cách giới hạn số bước nhảy mà một gói tin có thể đi qua, TTL giúp ngăn chặn lưu lượng truy cập tràn lan và đảm bảo rằng các gói tin chỉ được gửi đến các nút liên quan.

## Các gói tin OLSR này định kỳ bao nhiêu thời gian sẽ gửi lại và thông điệp mà chúng gửi là gì đến giây thứ 10? Tìm kiếm trường thông tin Neighbor Address, giá trị của chúng là bao nhiêu? Từ đó, suy ra ý nghĩa của thông điệp này?

A screenshot of a computer

Description automatically generated

Các gói tin OLSR đến giây thứ 10 có thông điệp là HELLO.

A screenshot of a computer

Description automatically generated

Các gói tin OLSR mang thông điệp HELLO sẽ được gửi định kỳ trong 2 giây.

A screenshot of a computer

Description automatically generated

Trường thông tin Neighbor Address chưa địa chỉ IP của node lân cận.

Trong giao thức định tuyến OLSR, thông điệp HELLO là một thông điệp định kỳ được phát đi bởi mỗi node trong mạng để quảng bá thông tin về bản thân và các nút lân cận. Và được sử dụng để:

* Phát hiện các nút lân cận
* Xây dựng bản đồ mạng
* Duy trì thông tin định tuyến

Thông điệp HELLO được phát đi với tần suất cố định, thường là 2 giây. Các node nhận thông điệp HELLO sẽ cập nhật thông tin về các nút lân cận và bản đồ mạng của mình.

Trong OLSR, thông điệp HELLO được sử dụng để xây dựng danh sách các node MPR (Multi-Point Relay). Các node MPR là các node được chọn để chuyển tiếp các gói tin giữa các node không trực tiếp liên kết với nhau. Việc sử dụng các node MPR giúp giảm tải lưu lượng mạng và cải thiện hiệu suất định tuyến.

## Tìm kiếm thông tin Link Type: Symmetric Link và Asymetric Link. Cho biết chúng hoạt động như thế nào?

A screenshot of a computer

Description automatically generated

Trong thông điệp HELLO của OLSR, **Link Type** là một trường 8 bit dùng để chỉ định loại liên kết giữa hai nút. Có hai loại Link Type chính:

* **Symmetric Link:** Liên kết hai chiều, có nghĩa là cả hai nút đều có thể truyền dữ liệu cho nhau.
* **Asymmetric Link:** Liên kết một chiều, có nghĩa là chỉ một nút có thể truyền dữ liệu cho nút kia.

**Link Type** được sử dụng để xây dựng bản đồ mạng của mỗi nút. Nếu hai nút có **Link Type** là **Symmetric Link**, thì chúng được coi là lân cận của nhau. Nếu hai nút có **Link Type** là **Asymmetric Link**, thì chỉ nút có **Link Type** là **Symmetric Link** mới được coi là lân cận của nút kia.

## Từ giây thứ 10 trở đi, tìm các MPR Link? Giải thích ý nghĩa của MPR và mô tả quá trình xảy ra đối với kịch bản?

### Các MPR Link.

Node 0:

A screenshot of a computer

Description automatically generated

Node 1:

A screenshot of a computer

Description automatically generated

Node 2:

A screenshot of a computer

Description automatically generated

Node 3:

A screenshot of a computer

Description automatically generated

Các giá trị của trườn MPR Link của các Node gửi đi sau 10 giầy là không thay đổi.

### Giải thích và ý nghĩa.

MPR Link là một liên kết giữa một node và một MPR (Multi-Point Relay). MPR là các node được chọn để chuyển tiếp các gói tin giữa các nút không trực tiếp liên kết với nhau.

MPR Link được xây dựng dựa trên thông tin trong bản tin Hello. Trong bản tin Hello, mỗi nút sẽ liệt kê danh sách các nút MPR của mình. Các nút nhận bản tin Hello sẽ cập nhật thông tin này vào bản đồ mạng của mình.

Khi một nút cần gửi một gói tin đến một nút không trực tiếp liên kết với mình, nó sẽ gửi gói tin đó đến một MPR Link. MPR Link sẽ chuyển tiếp gói tin đến nút đích thông qua các MPR Link khác.

Việc sử dụng MPR Link giúp giảm tải lưu lượng mạng vì các gói tin chỉ được truyền qua các MPR Link. Điều này giúp cải thiện hiệu suất định tuyến và giảm thiểu khả năng tắc nghẽn mạng.

### Mô tả kịch bản.

Qua các gói tin bắt được ta biết được MPR Link của các node sau giây thứ 10 như sau.

MPR(0) = {2}

MPR(1) = {2}

MPR(2) = {1}

MPR(3) = {1}

## Tìm các thông điệp TC, cho biết thông điệp này đang cố quảng bá điều gì?

### Các thông điệp TC.

Thông điệp TC từ Node 1:

A screenshot of a computer

Description automatically generated

Thông điệp TC từ Node 2 và thông điệp TC từ Node 1 được Node 2 forward.

A screenshot of a computer

Description automatically generated

Thông điệp TC từ Node 2 được Node 1 Forward.

A screenshot of a computer

Description automatically generated

### Giải thích.

TC (Topology Control) là một loại thông điệp được sử dụng trong giao thức định tuyến OLSR để quảng bá thông tin về bản đồ mạng. TC được gửi bởi các nút MPR (Multi-Point Relay) đến các nút lân cận của mình. Thông điệp TC chứa thông tin về các node lân cận của node MPR.

Cơ chế hoạt động của TC:

* **Trường hợp 1:** Nút MPR gửi TC tin nhắn đến các nút lân cận của mình trong tập hợp MS (Multipoint Set).
* **Trường hợp 2:** Nút nhận được TC tin nhắn từ một nút MPR khác.

## Xác định các gói tin UDP từ Node 0 → Node 1 trước giây thứ 20.

A screen shot of a computer

Description automatically generated

Xét Code ta có sinkPort là 9, đây chinhs là Port hoạt động UDP Sink App.

Xét file PCAP của Node 0, filter các gói tin UDP từ Port là 9 ta thu được các gói tin UDP từ Node 0 đến Node 1.

A screenshot of a computer

Description automatically generated

## Xác định các gói tin OLSR, ARP thể hiện lại quá trình tái định tuyến giữa Node 0 và Node 1 sau giây thứ 20 khi Node 3 đi vào vùng giữa Node 0 và Node 1 và Node 2 di chuyển ra ngoài vùng này. Nêu rõ ý nghĩa của các gói ARP lúc này và các gói OLSR thể hiện điều gì?

Tại **frame** 913 Node 0 lần đầu tiên nhận được gói tin OLSR với thông điệp **HELLO** từ Node 3. Tuy nhiên lúc này Node 4 vẫn chưa mất kết nối đến Node 0 nên Node 3 vẫn được sử dụng để truyền gói tin đến Node 1.

A screenshot of a computer

Description automatically generated

Bắt đầu tại **frame** 1334 Node 0 bắt đầu phát hiện mất hiện mất kết nối với Node 3 và cập nhập bảng định tuyến của mình dựa vào các gói tin mang thông điệp HELLO trước đó và sẽ chọn Node 3 làm **MPR Link**. Và gói tin **ARP Boardcast** được gửi từ Node 0 để lấy địa chỉ MAC của Node 3 phục vụ cho IEEE 802.11.

A screenshot of a computer

Description automatically generated

Node 3 cũng gửi gói tin ARP Broadcast để lấy địa chỉ MAC của Node 1 phục vụ cho IEEE 802.11. Sau quá trình này các gói tin từ Node 0 đến Node 1 sẽ thông qua Node trung gian là 3.

A screenshot of a computer

Description automatically generated

Node 0 gửi gói tin mang thông điệp HELLO kèm theo trường Lost Link thông báo mất kết nối đến Node 2 và thông báo MRP Link mới là Node 3.

A screenshot of a computer

Description automatically generated

## Tăng khoảng cách của Node 0 và Node 1 lên 350m. Điều chỉnh vị trí ban đầu của Node 3, vận tốc của Node 2 và Node 3 sao cho ở giây thứ 15 cả hai Node 2 và Node 3 sẽ gặp nhau ở cùng một tọa độ x.

### Sửa đổi code.

Ban đầu ta có vị trí các Node:

* Node 0: x0=0, y0=0
* Node 1: x1=300, y1=0
* Node 2: x2=0, y2=110
* Node 3: x3=600, y3=-150

Tăng khoảng cách của Node 0 và 1 lên 350m.

Ta sẽ có vị trí của:

* Node 0: x0=0, y=0
* Node 1: x1=350, y=0

Tại giây thứ 15 cả 2 Node 2 và Node 3 gặp nhau tại cùng một toà độ x.

Ta có tại giây thứ 15:

* Node 2: x2=0, y2=110
* Node 3: x3=x, y3=-150

Suy ra:

* x2 + 10 \* 15 = x3 – v3 \* 15
* x2 + 150 = x3 – 300
* x3 = x2 + 450
* x3 = 0 + 450 = 450

Vậy tạ có x3 cần thay đổi thành 450.

Thay đổi Code:

A computer screen shot of a program code

Description automatically generated

### Xác nhận lại qua mô phỏng.

A screenshot of a video editing

Description automatically generated

Qua mô phỏng ta có thể thấy ở giây 15 hai Node 2 và 3 đã gặp nhau tại cùng tại cùng toà x. Và Node 0 cách Node 1 một khoảng 350m.

# AODV Routing

## Compile và chạy mô phỏng, quan sát quá trình chạy của kịch bản. Thu được các file pcap.

1. Tại folder **ns3-3.40** copy file **lab2-aodv.cc** và **myapp.h** được cung cấp vào folder **scratch**.
2. Re-build lại ns3 bằng lệnh.

$ ./ns3 build

1. Sau khi đã Re-build xong, để chạy lab2-aodv.cc kèm theo mô phỏng ta thực hiện lệnh.

$ ./ns3 run scratch/lab2-aodv.cc -–vis

Sau khi chạy ta được giao diện mô phỏng như hình.

A screen shot of a computer

Description automatically generated

Quan sát mô phỏng ta nhận thấy kịch bản chạy giống như mô tả của đề bài như sau:

A diagram of a diagram

Description automatically generated

Sau khi chạy mô phỏng sau qua giây 40 ta tắt mô hỏng, các file PCAP sẽ được phần mềm tự động tạo ra với số thứ tự ứng với từng node.

A screen shot of a computer screen

Description automatically generated

## Địa chỉ IP của Node 0, Node 1, Node 2, Node 3 lần lượt là gì?

A black background with white text

Description automatically generated

Ta có biến c kiểu NodeContainer. NodeContainer là một tập hợp các node. Tạo 4 node trong mạng và thêm chúng vào biến c.

A black background with white text

Description automatically generated

Cài đặt các thiết bị mạng cho các node trong biến **c.**

A screen shot of a computer

Description automatically generated

Cài đặt địa chỉ của mạng sẽ là 10.1.1.0, từ đó các node sẽ được gặn địa chỉ tăng dần theo từng Node, IP 10.1.1.1 cho Node 0, địa chỉ IP 10.1.1.2 cho Node 1, địa chỉ IP 10.1.1.3 cho Node 2 và địa chỉ IP 10.1.1.4 cho Node 3.

A computer screen shot of text

Description automatically generated

Từ các lệnh cài đặt vị trí ta có thể khẳng định rằng thứ tự các Node trong kịch bản tương đương với thứ tự các Node từ đề bài.

Tóm lại địa chỉ IP của Node 0, Node 1, Node 2, Node 3 lần lượt là:

* Node 0: 10.1.1.1
* Node 1: 10.1.1.2
* Node 2: 10.1.1.3
* Node 3: 10.1.1.4

## Xem file pcap của Node 3, xác định các gói AODV thể hiện quá trình định tuyến giữa các Node 0, Node 3, Node 1 từ giây thứ 1 đến 13.

A red screen with white lines

Description automatically generated

Các gói thông điệp AODV dùng để định tuyến giữa các Node 0, 1, 3:

Route:

* Route Request
* Route Reply
* Route Reply Acknowledgment

## Khoảng bao nhiêu giây thì gồm Node 0, Node 2, Node 3 gửi gói Route Reply đến địa chỉ Broadcast.

Node 0:

A screenshot of a computer

Description automatically generated

Node 2:

A screenshot of a computer

Description automatically generated

Node 3:

A screenshot of a computer

Description automatically generated

Có thể thấy khoảng 1 giây Node 0, Node 2, Node 3 gửi gói Route Reply đến địa chỉ Broadcast.

## Tìm gói tin Route Request thể hiện quá trình tìm đường đi từ Node 0 đến Node 1. Thông tin nào thể hiện quá trình đó?

Gói tin Route Request thể hiện quá trình tìm đường đi từ Node 0 đến Node 1:

A screenshot of a computer

Description automatically generated

Trường Originator IP thể hiện địa chỉ Node nguồn.

Trường Destination IP thể hiện địa chỉ Node địch.

Và các địa chỉ IP tại hai trường này đều tương ứng với Node nguồn là 0, Node đích là 1 dựa trên địa chỉ IP từ câu 1.

## Tìm các gói tin Route Reply thể hiện quá trình kịch bản đã khám phá được đường đi từ Node 0 đến Node 1 và ngược lại. Thông tin nào giúp ta xác định được điều đó? Giải thích.

### Quá trình tìm đường trong AODV

Quá trình phát hiện đường đi (route discovery) trong AODV diễn ra như sau:

* Node nguồn gửi **Route Request** đến tất cả các nút hàng xóm của nó.
* Các Node lân cận lần lượt kiểm tra xem mình có đường đi đến đích hay không. Nếu có, chúng sẽ gửi **Route Replay** trở lại nguồn. Nếu không, chúng sẽ phát lại RREQ đến tất cả các node hàng xóm của mình.
* Quá trình này tiếp tục cho đến khi nguồn nhận được **Route Replay** từ đích hoặc một nút trung gian có đường đi đến đích.

### Quá trình tìm đường từ Node 0 đến 1 trong kịch bản.

Node 0 gửi gói tin **Route Request** đến các Node lân cận để tìm đường đi đến Node 1, lúc này các Node lân cận của 0 lần lượt là 2 và 3.

A screenshot of a computer

Description automatically generated

Khi này Node 3 nhận được gói tin **Route Request** và Node 3 có đường đi đến Node 1. Node 3 sẽ phản hồi lại gói tin **Route Replay** cho Node 0 để thông báo cho đường đi đến Node 1.

A screenshot of a computer

Description automatically generated

Khi Node 0 nhận lại gói tin **Route Replay** thì sẽ thông báo lại cho Node 3 gói tin **Route Replay-ACK** để thông báo rằng đã nhận được và chấp nhận gói tin **Route Replay** từ Node 3.

A screenshot of a computer

Description automatically generated

Đồng thời Node 3 sẽ gởi gói tin **Route Replay** cho Node 1 để thông báo cho Node 1 đường đi đến Node 0.

A screenshot of a computer

Description automatically generated

## Sau giây thứ 13, tìm các gói tin Route Error. Tại sao có thông điệp này xảy ra? Trường thông tin nào chứng minh được điều đó?

A screenshot of a computer

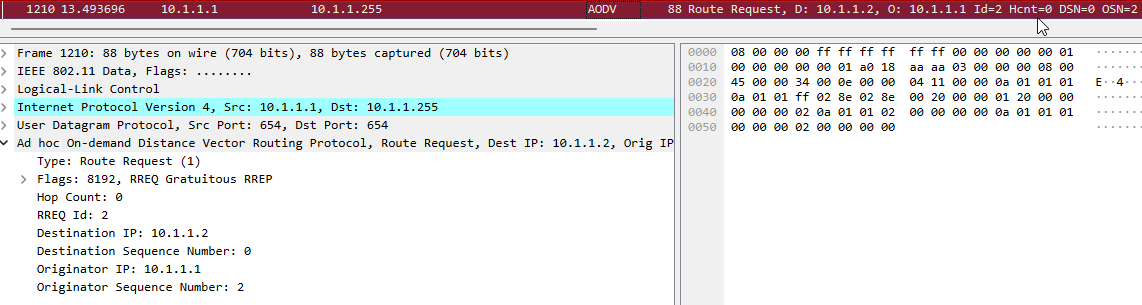
Description automatically generated

Gói tin **Route Error** để thông báo cho các node khác về một đường đi bị hỏng. **Route Error** được gửi đi bởi một node khi nó phát hiện ra rằng đường đi đến một node đích cụ thể không còn hoạt động.

Xe gói tin **Route Error** với trường **Unreachable Destinations** cho biết rằng đã mất kết nối đến Node 1 vàđược gửi từ Node 3 đến Node 0 để thông báo sự cố.

## Xác định các gói tin AODV, ARP thể hiện lại quá trình tái định tuyến giữa Node 0 và Node 1 sau giây thứ 13 khi Node 3 đi ra khỏi vùng truyền thông của Node 0 và Node 1. Mô tả lại quá trình đó.

Sau giây thứ 13 Node 3 mất kết nối tới Node 1 nên đường đi từ Node 0 đến Node 1 lúc này sử dụng Node 3 làm trung gian sẽ không còn khả thi. Lúc này Node 0 sẽ tạo gói tin **Route Request** mới để tìm đường đi đến Node 1.



Lúc này Node 2 nhận được gói tin **Route Request** từ Node 0 và Node 1 cũng là lân cận của Node 2.

Node 2 sẽ gửi gói tin Route Reply để thông báo đường đi đến Node 0 cho Node 1. Vì đây là lần đầu gửi gói tin đến Node 1, Node 2 sẽ gửi gói tin ARP Boardcast để tìm địa chỉ MAC của Node 1 phục vụ cho IEEE 802.11. Node 1 nhận được phản hồi lại địa chỉ MAC cho Node 2.

A screenshot of a computer

Description automatically generated

Đồng thời Node 2 sẽ phản hồi lại Route Request từ Node 1 bằng gói tin Route Replay để thông báo đường đi đến Node 1 cho Node 0. Và đây cũng là lần đầu gửi gòi tin đến Node 1, Node 2 sẽ gửi ARP Boardcast để tìm địa chỉ MAC của Node 0 phục vụ cho IEEE 802.11.

A screenshot of a computer

Description automatically generated

Node 0 sẽ phản hồi gòi tin **Route Replay** từ Node 2 bằng gói tin **Route Replay-ACK** để xác nhận với Node 2 rằng đã nhận được gói tin **Route Replay** và chấp nhận đường đi. Và đây cũng là lần đầu gửi gòi tin đến Node 2, Node 0 sẽ gửi **ARP Boardcast** để tìm địa chỉ MAC của Node 0 phục vụ cho IEEE 802.11.

A screenshot of a computer

Description automatically generated

# TÀI LIỆU THAM KHẢO