

2

HỆ THỐNG NHÚNG
MẠNG KHÔNG DÂY

Tìm hiểu các Routing Protocol trong mạng MANET

Biên soạn: Phan Trung Phát
Liên hệ: phatpt@uit.edu.vn

Lưu hành nội bộ

A. TỔNG QUAN

1. Mục tiêu

- Tìm hiểu cách vận hành của OLSR và cấu trúc gói tin.
- Tìm hiểu cách vận hành của AODV và cấu trúc gói tin.
- Hiểu về cách cài đặt giao thức định tuyến trên trong NS-3.

2. Môi trường thực hành

- Máy ảo Ubuntu 20.04 đã cài đặt NS-3.
- Phần mềm Wireshark.

B. GIỚI THIỆU

1. Optimized Link State Routing (OLSR)

Tìm hiểu về giao thức định tuyến **Optimized Link State Routing** (OLSR) sử dụng trong Mobile Ad-hoc Network (MANET). NS-3 cung cấp một loạt các phương thức và các tiện ích để mô phỏng được sự dịch chuyển của các node trong mô hình truyền thông mạng không dây (MANET) và các Routing Protocol Helper để cài đặt lên các chuẩn giao tiếp không dây IEEE 802.11 MAC.

2. Ad-hoc On-Demand Distance Vector (AODV)

Tương tự như OLSR, tại phần thực hành này sẽ tìm hiểu giao thức định tuyến **Ad-hoc On-Demand Distance Vector** (AODV) sử dụng trong Mobile Ad-hoc Network (MANET) và các dạng mạng không dây ad-hoc khác.

C. THỰC HÀNH

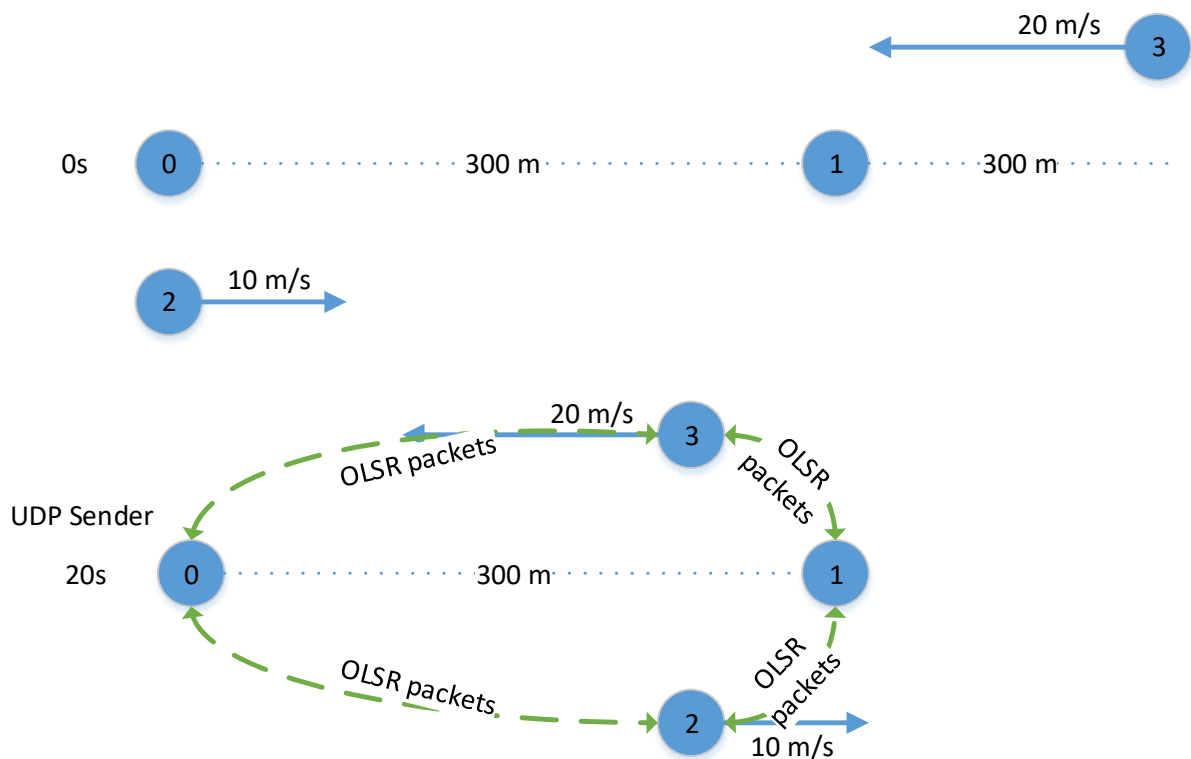
3. Cài đặt và mô phỏng giao thức OLSR

Mô tả mô hình

Trong mô hình sẽ có tổng cộng 4 Node, Node 0 và Node 1 sẽ đứng cố định một chỗ trong khi 2 Node 2 và Node 3 sẽ di chuyển lần lượt với các vận tốc 10 m/s và 20 m/s theo 2 hướng ngược nhau để đi vào vùng truyền thông của giao thức định tuyến OLSR.

Trên Node 1 sẽ được cài đặt một UDP Sink Application, Node 0 được cài đặt một UDP Sender liên tục gửi các gói tin đến địa chỉ IP của Node 1.

Sau khi Node 2 hoặc Node 3 đi vào giữa Node 0 và Node 1, giao thức định tuyến OLSR sẽ thiết lập một kênh truyền Mobile Ad-hoc Network (MANET) giữa Node 0 và Node 1, từ đó thiết lập kết nối UDP giữa Node 0 và Node 1.



Hình 1. Mô hình các node trong mô phỏng giao thức OLSR

Giải thích các bước cài đặt, xây dựng mô hình

Chuẩn bị mô hình di chuyển cho các node trong mô phỏng.

```
Ptr<ConstantVelocityMobilityModel> cvmm2;  
Ptr<ConstantVelocityMobilityModel> cvmm3;
```

Tạo kênh kết nối wifi (YansWifiPhyHelper trong NS-3).

```
YansWifiPhyHelper wifiPhy;  
wifiPhy.SetPcapDataLinkType (YansWifiPhyHelper::DLT_IEEE802_11);
```

Cài đặt thông số về công suất truyền và khoảng cách truyền.

```
wifiPhy.Set ("TxPowerStart", DoubleValue(13));  
wifiPhy.Set ("TxPowerEnd", DoubleValue(13));  
wifiPhy.Set ("TxPowerLevels", UIntegerValue(1));  
wifiPhy.Set ("TxGain", DoubleValue(0));  
wifiPhy.Set ("RxGain", DoubleValue(0));
```

Cấu hình chế độ ad-hoc cho kênh truyền cài đặt lên các NetDevice.

```
WifiMacHelper wifiMac;  
wifiMac.SetType ("ns3::AdhocWifiMac");  
  
// Set 802.11b standard  
wifi.SetStandard (WIFI_STANDARD_80211b);  
  
wifi.SetRemoteStationManager ("ns3::ConstantRateWifiManager",  
                               "DataMode",StringValue(phyMode),  
                               "ControlMode",StringValue(phyMode));  
  
NetDeviceContainer devices;
```

```
devices = wifi.Install (wifiPhy, wifiMac, c);
```

Cấu hình OSLR sử dụng trong quá trình truyền thông.

```
OlsrHelper olsr;  
  
// Install the routing protocol  
Ipv4ListRoutingHelper list;  
list.Add (olsr, 10);  
  
// Set up internet stack  
InternetStackHelper internet;  
internet.SetRoutingHelper (list);  
internet.Install (c);
```

Cấp địa chỉ IP cho các NetDevice thuộc các Node.

```
Ipv4AddressHelper ipv4;  
NS_LOG_INFO ("Assign IP Addresses.");  
ipv4.SetBase ("10.1.1.0", "255.255.255.0");  
Ipv4InterfaceContainer ifcont = ipv4.Assign (devices);
```

Các bước cài đặt UDP Sink Application và UDP Sender có thể tham khảo source code.

Cài đặt vị trí cho các Node.

```
MobilityHelper mobility;  
Ptr<ListPositionAllocator> positionAlloc = CreateObject  
<ListPositionAllocator>();  
positionAlloc ->Add(Vector(0, 0, 0)); // node0  
positionAlloc ->Add(Vector(300, 0, 0)); // node1  
positionAlloc ->Add(Vector(0, 110, 0)); // node2
```

```
positionAlloc ->Add(Vector(600, -150, 0)); // node3
```

Cài đặt tốc độ và hướng di chuyển cho các Node.

```
Vector vel2 (10, 0, 0);  
Vector vel3 (-20, 0, 0);  
  
mobility.SetMobilityModel("ns3::ConstantVelocityMobilityModel");  
mobility.Install(c);  
cvmm2 = mover2->GetObject<ConstantVelocityMobilityModel> ();  
cvmm2->SetVelocity(vel2);  
  
cvmm3 = mover3->GetObject<ConstantVelocityMobilityModel> ();  
cvmm3->SetVelocity(vel3);
```

4. Cài đặt và mô phỏng giao thức AODV

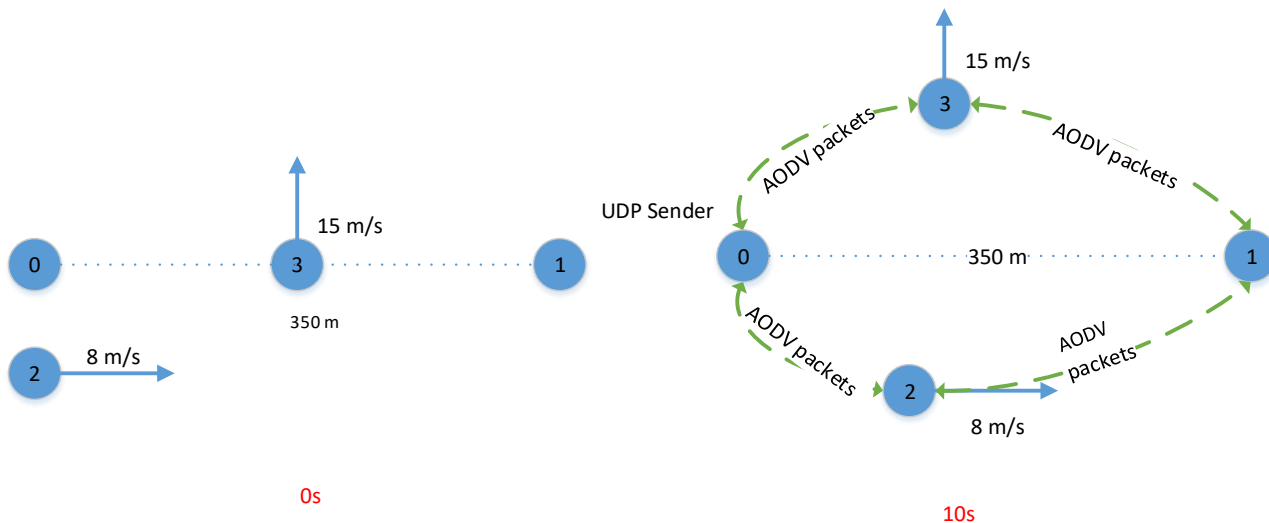
Mô tả mô hình

Trong mô hình sẽ có tổng cộng 4 Node, Node 0 và Node 1 sẽ đứng cố định một chỗ. Ban đầu Node 3 nằm trong vùng truyền thông của Node 0 và Node 1, trong khi Node 2 sẽ nằm ngoài vùng truyền thông này. Sau đó, Node 2 và Node 3 sẽ di chuyển lần lượt với các vận tốc 8 m/s và 15 m/s theo 2 hướng ngược nhau. Node 2 đi về bên phải vào vùng truyền thông giữa Node 0 và Node 1, trong khi đó Node 3 đi về phía trên ra khỏi vùng truyền thông giữa Node 0 và Node 1.

Trên Node 1 sẽ được cài đặt một UDP Sink Application, Node 0 được cài đặt một UDP Sender liên tục gửi các packet đến địa chỉ IP của Node 1.

Thực nghiệm và quan sát quá trình tái định tuyến khi Node 3 ra khỏi vùng truyền thông Node 0 và Node 1, và Node 2 đi vào vùng này của giao thức định tuyến AODV để thiết lập

một kênh truyền Mobile Ad-hoc Network (MANET) giữa Node 0 và Node 1, từ đó tái thiết lập kết nối UDP giữa Node 0 và Node 1.



Hình 2. Mô hình các node trong mô phỏng giao thức AODV

Giải thích các bước cài đặt, xây dựng mô hình

Cấu hình AODV sử dụng trong quá trình truyền thông.

```
AodvHelper aodv;
OlsrHelper olsr;
// Install the routing protocol
Ipv4ListRoutingHelper list;
list.Add (aodv, 10);

// Set up internet stack
InternetStackHelper internet;
internet.SetRoutingHelper (list);
internet.Install (c);
```

Cấp địa chỉ IP cho các NetDevice thuộc các Node.

```
Ipv4AddressHelper ipv4;  
NS_LOG_INFO ("Assign IP Addresses.");  
ipv4.SetBase ("10.1.1.0", "255.255.255.0");  
Ipv4InterfaceContainer ifcont = ipv4.Assign (devices);
```

Các bước cài đặt UDP Sink Application và UDP Sender có thể tham khảo source code.

Cài đặt vị trí cho các Node.

```
MobilityHelper mobility;  
Ptr<ListPositionAllocator> positionAlloc = CreateObject  
<ListPositionAllocator>();  
positionAlloc ->Add(Vector(0, 0, 0)); // node0  
positionAlloc ->Add(Vector(350, 0, 0)); // node1  
positionAlloc ->Add(Vector(0, 110, 0)); // node2  
positionAlloc ->Add(Vector(150, 0, 0)); // node3  
  
Ptr<Node> mover2 = c.Get(2);  
Ptr<Node> mover3 = c.Get(3);  
  
mobility.SetPositionAllocator(positionAlloc);
```

Cài đặt tốc độ và hướng di chuyển cho các Node.

```
Vector vel2 (8, 0, 0);  
Vector vel3 (0, -15, 0);  
  
mobility.SetMobilityModel("ns3::ConstantVelocityMobilityModel");  
mobility.Install(c);  
cvmm2 = mover2->GetObject<ConstantVelocityMobilityModel> ();  
cvmm2->SetVelocity(vel2);
```



```
cvmm3 = mover3->GetObject<ConstantVelocityMobilityModel> ();  
cvmm3->SetVelocity(v13);
```

D. YÊU CẦU & ĐÁNH GIÁ

1. Yêu cầu

Đối với nội dung OLSR, sinh viên thực hiện các yêu cầu sau:

1. Compile và chạy mô phỏng, quan sát quá trình chạy của kịch bản. Thu được các file pcap.
2. Địa chỉ IP của Node 0, Node 1, Node 2, Node 3 lần lượt là gì?

Dựa trên các file .pcap, sử dụng Wireshark để tìm và trả lời các câu hỏi sau:

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.
4. Địa chỉ IP đích của các gói tin OLSR từ các Node là gì? Chúng có ý nghĩa như thế nào? Giải thích.
5. Tìm kiếm trường thông tin TTL của các gói tin OLSR, cho biết chúng có điều gì đặc biệt? Từ đó rút ra ý nghĩa tại sao chúng có giá trị đó?
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?
7. Tìm kiếm thông tin Link Type: Symmetric Link và Asymmetric Link. Cho biết chúng hoạt động như thế nào?
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?
9. Tìm các thông điệp TC, cho biết thông điệp này đang cổ quảng bá điều gì?
10. Xác định các gói tin UDP từ Node 0 → Node 1 trước giây thứ 20.

Gợi ý: tìm port của UDP Sink App.

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?

Gợi ý: tìm và nêu rõ các Neighbor Node của Node 2.

Mở source code của kịch bản và chỉnh sửa theo yêu cầu:

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. Nộp lại file này.

Đối với nội dung AODV, sinh viên thực hiện các yêu cầu sau:

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

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

Dựa trên các file .pcap, sử dụng Wireshark để tìm và trả lời các câu hỏi sau:

15. 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. Chú ý liệt kê rõ các loại thông điệp AODV mà bạn tìm thấy dùng để định tuyến giữa Node 0, 1, 3.

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

17. 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?

18. 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.

19. 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?

20. 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.

Gợi ý: xem file pcap của Node 2 và Node 3.

2. Đánh giá

- Sinh viên tìm hiểu và thực hành theo hướng dẫn. Thực hiện theo **nhóm**.
- Sinh viên báo cáo kết quả thực hiện và nộp bài bằng file. Trong đó:
 - Trình bày chi tiết quá trình thực hành và trả lời các câu hỏi nếu có (kèm theo các ảnh chụp màn hình tương ứng).
 - Giải thích, tìm hiểu, lý giải các kết quả đạt được.
 - Tải mẫu báo cáo thực hành và trình bày theo mẫu được cung cấp.

Nén **.ZIP** tất cả các file và đặt tên file theo định dạng theo mẫu:

NhomY-LabX_MSSV1_MSSV2

Ví dụ: Nhom1-Lab02_25520001_25520002

- Nộp báo cáo trên theo thời gian đã thống nhất tại website môn học.
- Các bài nộp không tuân theo yêu cầu sẽ **KHÔNG** được chấm điểm.

HẾT

Chúc các bạn hoàn thành tốt bài thực hành!