

به نام خدا



Sharif University of Technology

Department of Electrical Engineering

Data Network

Instructor: Dr. Pakravan

Network Layer Simulation

Mohammad Javad Amin

401211193

Fall 2023

OSPF, or Open Shortest Path First, is a dynamic routing protocol commonly used in large-scale enterprise networks. It is an interior gateway protocol (IGP) designed to efficiently and dynamically distribute routing information within an Autonomous System (AS). OSPF operates on the link layer (Layer 2) and the network layer (Layer 3) of the OSI model and employs the Dijkstra algorithm to calculate the shortest path to reach destinations within the network.

Operational Principles of OSPF:

1. Neighbor Discovery:

- OSPF routers establish neighbor relationships by exchanging Hello packets. Routers must agree on certain parameters, such as area ID, to form a neighbor relationship.

2. Link-State Database (LSDB):

- OSPF routers maintain a complete and synchronized database of the network's topology called the Link-State Database (LSDB). This database contains information about routers, links, and their states.

3. Link-State Advertisements (LSAs):

- OSPF routers share information about their local networks by broadcasting Link-State Advertisements. These LSAs are used to update the LSDB and ensure that all routers have a consistent view of the network.

4. Dijkstra Shortest Path First Algorithm:

- OSPF routers use the Dijkstra algorithm to calculate the shortest path to reach each network in the AS. The result is a routing table that allows routers to make informed forwarding decisions.

5. Area Structure:

- OSPF networks are divided into areas, and routers within an area have detailed knowledge of that area's topology. This division into areas helps in scaling the network, reducing the size of the LSDB, and limiting the impact of changes within a specific area.

6. Hierarchical Design:

- OSPF employs a hierarchical design with backbone areas (Area 0) and non-backbone areas. This design enhances network scalability and reduces the amount of routing information that routers must process.

Advantages of OSPF:

1. Fast Convergence:

- OSPF converges quickly in response to changes in the network, making it suitable for dynamic environments.

2. Scalability:

- The hierarchical design of OSPF, with the use of areas, enables scalability. Each area can be managed independently, reducing the overall complexity of the network.

3. Load Balancing:

- OSPF supports equal-cost multipath (ECMP) routing, allowing for load balancing across multiple paths to a destination.

4. Support for VLSM and CIDR:

- OSPF supports Variable Length Subnet Masking (VLSM) and Classless Inter-Domain Routing (CIDR), providing flexibility in IP address allocation.

5. Security:

- OSPF supports authentication mechanisms to ensure the integrity of routing information.

Disadvantages of OSPF:

1. Resource Intensive:

- OSPF can be resource-intensive, especially in large networks, due to the frequent exchange of routing information and the maintenance of the LSDB.

2. Complex Configuration:

- Configuring OSPF can be complex, and misconfigurations may lead to suboptimal routing or network instability.

3. Limited Support for Policy-Based Routing:

- OSPF's focus on efficient routing may limit its support for complex policy-based routing requirements.

Importance of OSPF Areas:

1. Efficiency:

- Dividing the network into areas reduces the size of the LSDB within each area, making it more manageable. This reduces the burden on routers and improves overall network efficiency.

2. Scalability:

- Areas enable a hierarchical network structure, allowing for modular growth. Changes in one area have limited impact on routers in other areas, enhancing scalability.

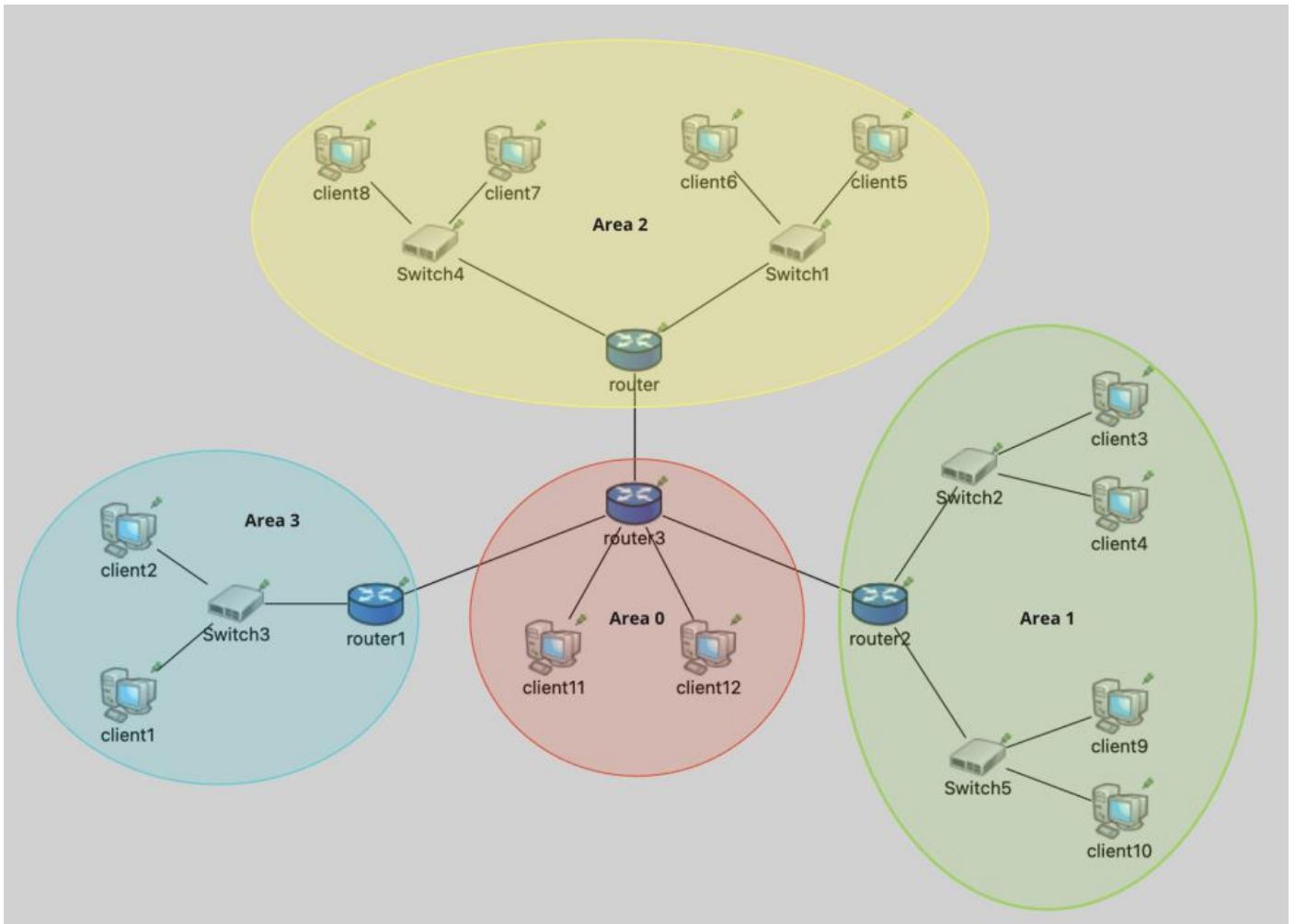
3. Reduced SPF Tree Calculation:

- By limiting the SPF tree calculation to a specific area, routers only need to consider the topology within that area, reducing the computational load.

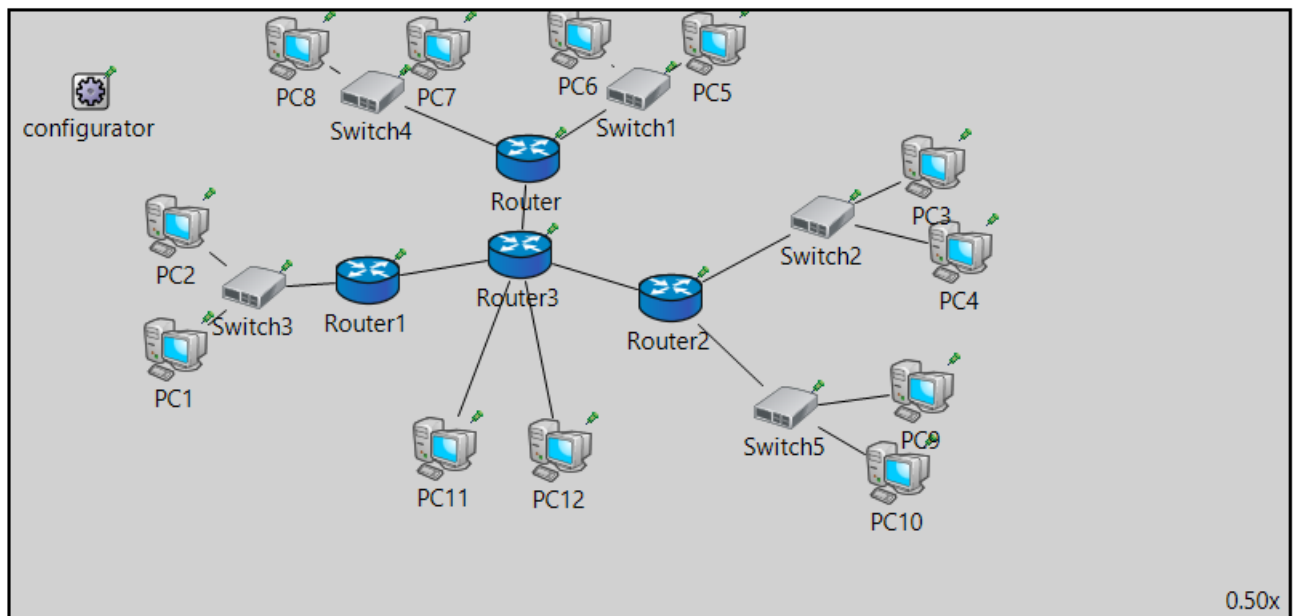
4. Control of Routing Information:

- Each area has its own link-state database, providing routers with detailed information about their local area without overwhelming them with information about the entire network.

شبکه را مطابق شکل زیر در فایل ned تعریف می کنیم.



Network



حال به توجه به صورت مسئله ناحیه بندی OSPF را در فایل XML را تعریف می کنیم.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<OSPFASConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="OSPF.xsd">
```

```

<!-- Areas -->
<Area id="0.0.0.0">
  <AddressRange address="192.168.4.0" mask="255.255.255.0" status="Advertise" />
</Area>

<Area id="0.0.0.1">
</Area>

<Area id="0.0.0.2">
</Area>

<Area id="0.0.0.3">
</Area>

<!-- Routers -->

<Router name="Router1" RFC1583Compatible="true">
  <BroadcastInterface ifName="eth0" areaID="0.0.0.3" />
  <PointToPointInterface ifName="eth1" areaID="0.0.0.0"/>
</Router>

<Router name="Router" RFC1583Compatible="true">
  <BroadcastInterface ifName="eth1" areaID="0.0.0.2" />
  <BroadcastInterface ifName="eth2" areaID="0.0.0.2" />
  <PointToPointInterface ifName="eth0" areaID="0.0.0.0" />
</Router>

<Router name="Router3" RFC1583Compatible="true">
  <PointToPointInterface ifName="eth0" areaID="0.0.0.0" />
  <BroadcastInterface ifName="eth1" areaID="0.0.0.0" />
  <BroadcastInterface ifName="eth2" areaID="0.0.0.0" />
  <PointToPointInterface ifName="eth3" areaID="0.0.0.0" />
  <PointToPointInterface ifName="eth4" areaID="0.0.0.0"/>
</Router>

<Router name="Router2" RFC1583Compatible="true">
  <PointToPointInterface ifName="eth2" areaID="0.0.0.0" />
  <BroadcastInterface ifName="eth1" areaID="0.0.0.1" />
  <BroadcastInterface ifName="eth0" areaID="0.0.0.1" />

</Router>

</OSPFASConfig>

```

حال با توجه به خواسته سوال که ping کردن PC ها است فایل INI را می نویسم.

```

[General]
network = Network

sim-time-limit = 200s

**.ospf.ospfConfig = xmldoc("ASConfig.xml")

*.PC*.numApps = 1
*.PC*.app[0].typename = "PingApp"

*.PC1.app[0].destAddr = "192.168.3.11"
*.PC1.app[0].sendInterval = 1s
*.PC1.app[0].startTime = 0.2s

```

```

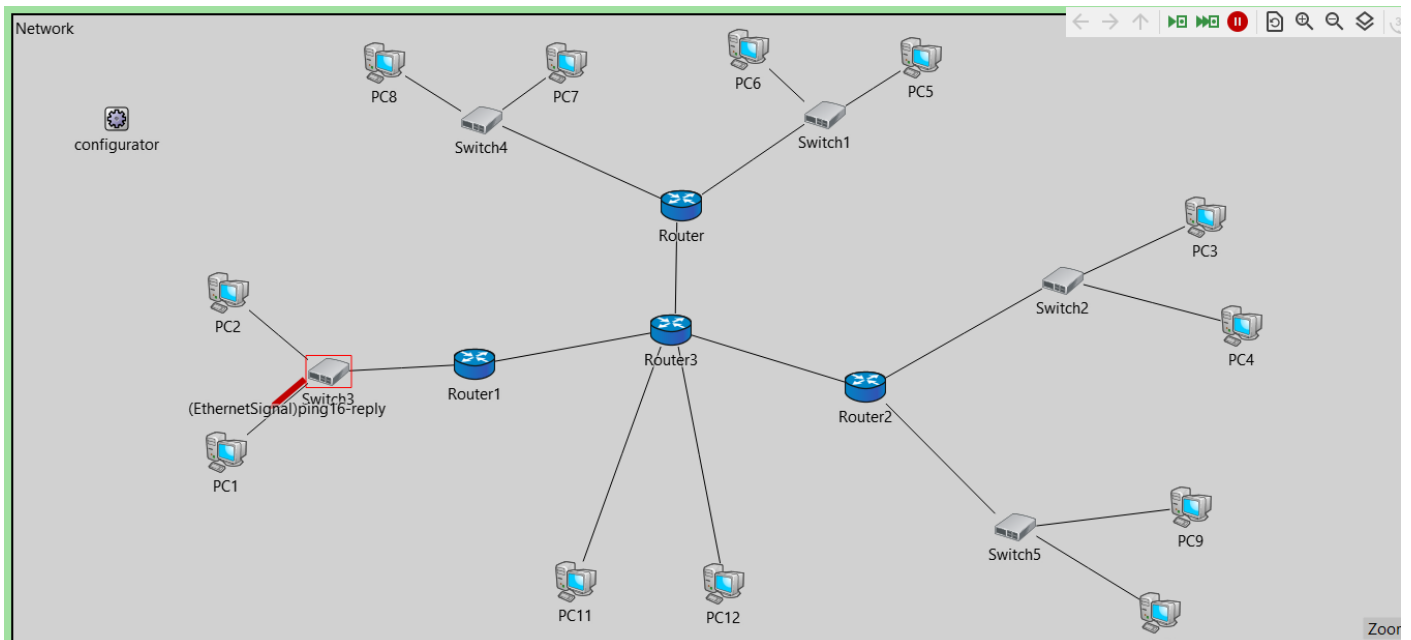
*.PC5.app[0].destAddr = "192.168.4.10"
*.PC5.app[0].sendInterval = 1s
*.PC5.app[0].startTime = 0.4s

*.PC2.app[0].destAddr = "192.168.2.2"
*.PC2.app[0].sendInterval = 1s
*.PC2.app[0].startTime = 0.6s

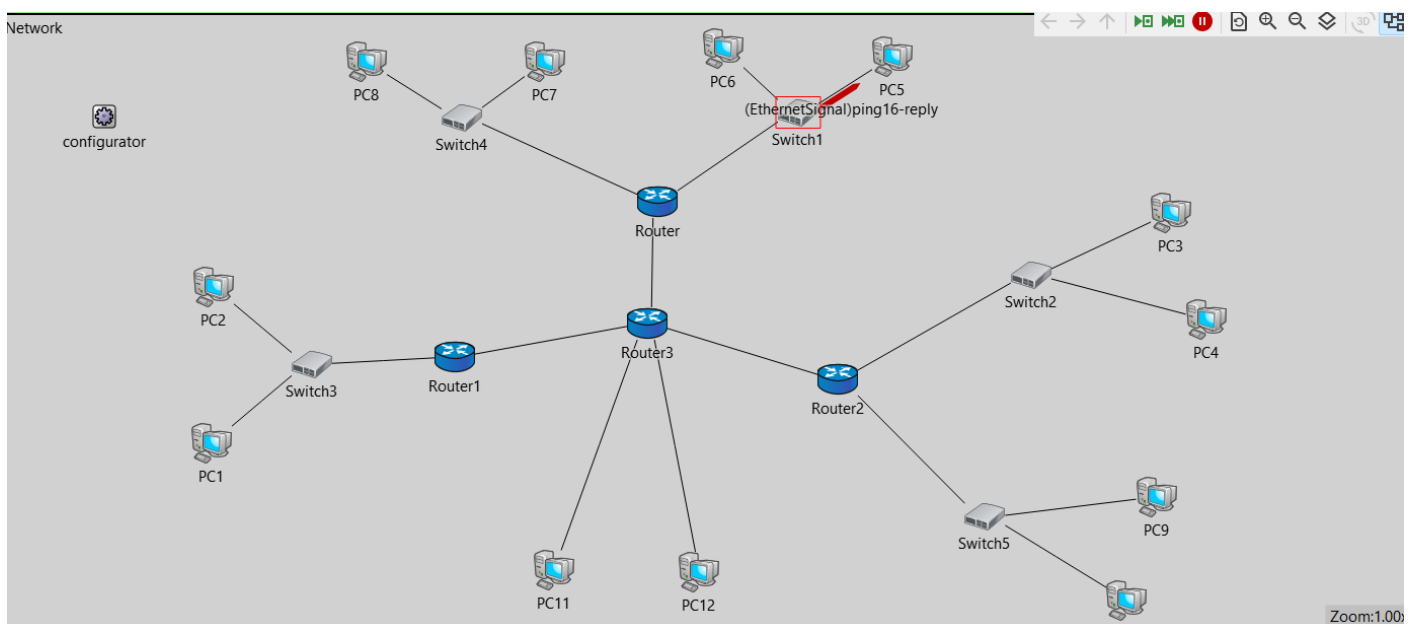
```

از اجرا فیلم تهیه شده است.

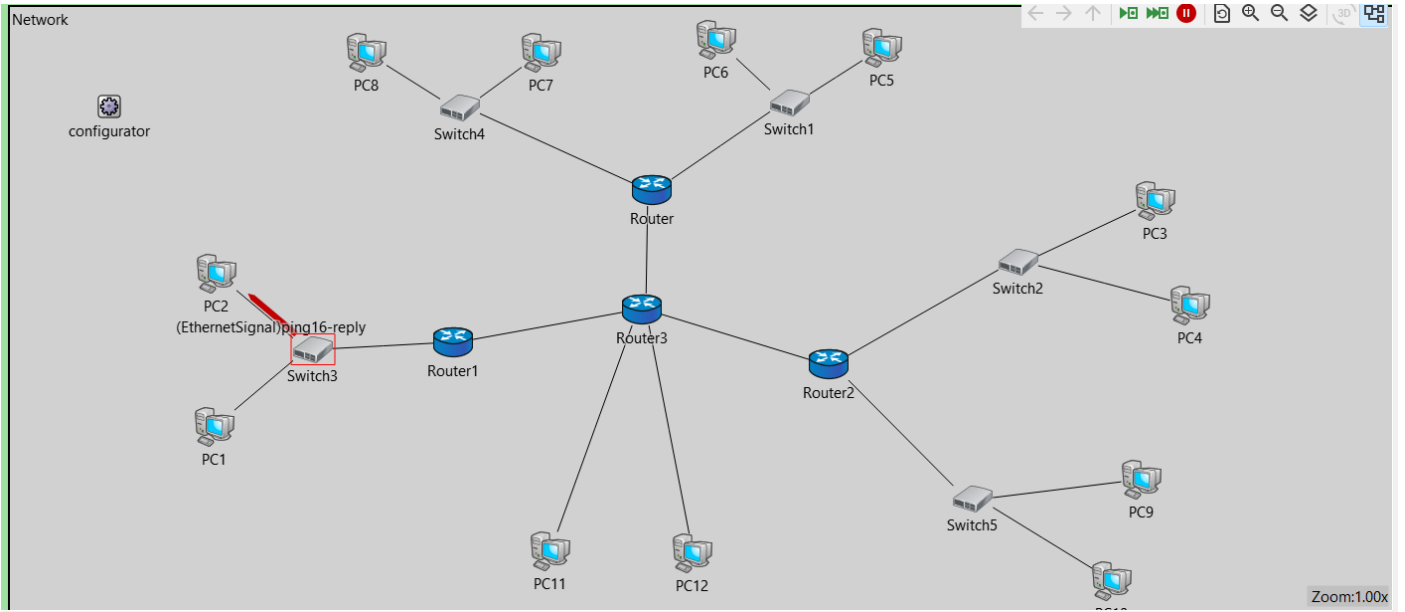
Ping:PC1->PC10



Ping:PC5->PC11

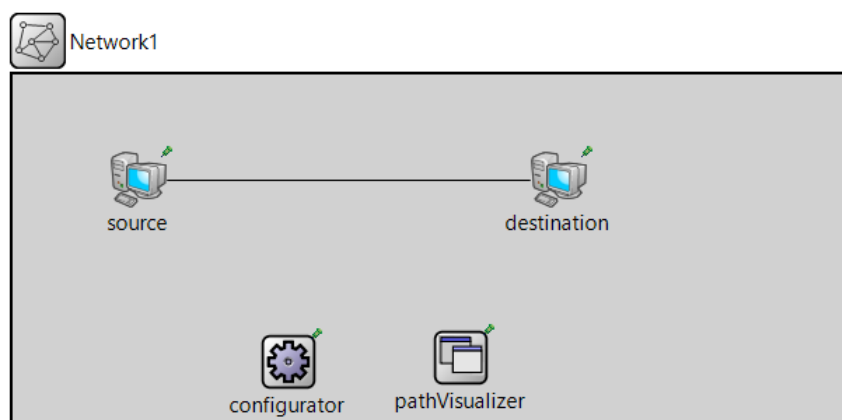


Ping:PC2->PC7



Enabling Visualization of Network Layer Activity:

مطابق صورت شبکه، فایل ned را می نویسیم.

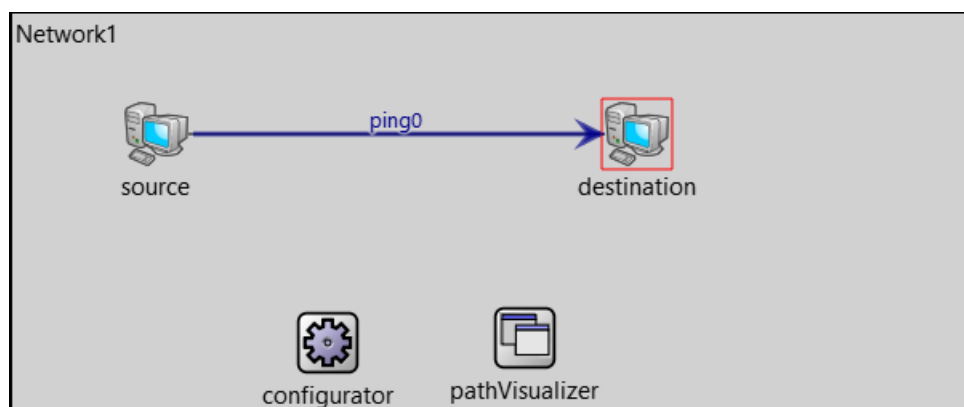


حال با توجه خواسته سوال source باید destination را ping کند. فایل INI را مطابق زیر می نویسیم.

```
[General1]
sim-time-limit = 5s
network = Network1

# Application parameters
*.source.numApps = 1
*.source.app[0].typename = "PingApp"
*.source.app[0].destAddr = "destination"

# Visualizer parameters
*.pathVisualizer.*.displayRoutes = true
*.pathVisualizer.*.fadeOutTime = 2s
```

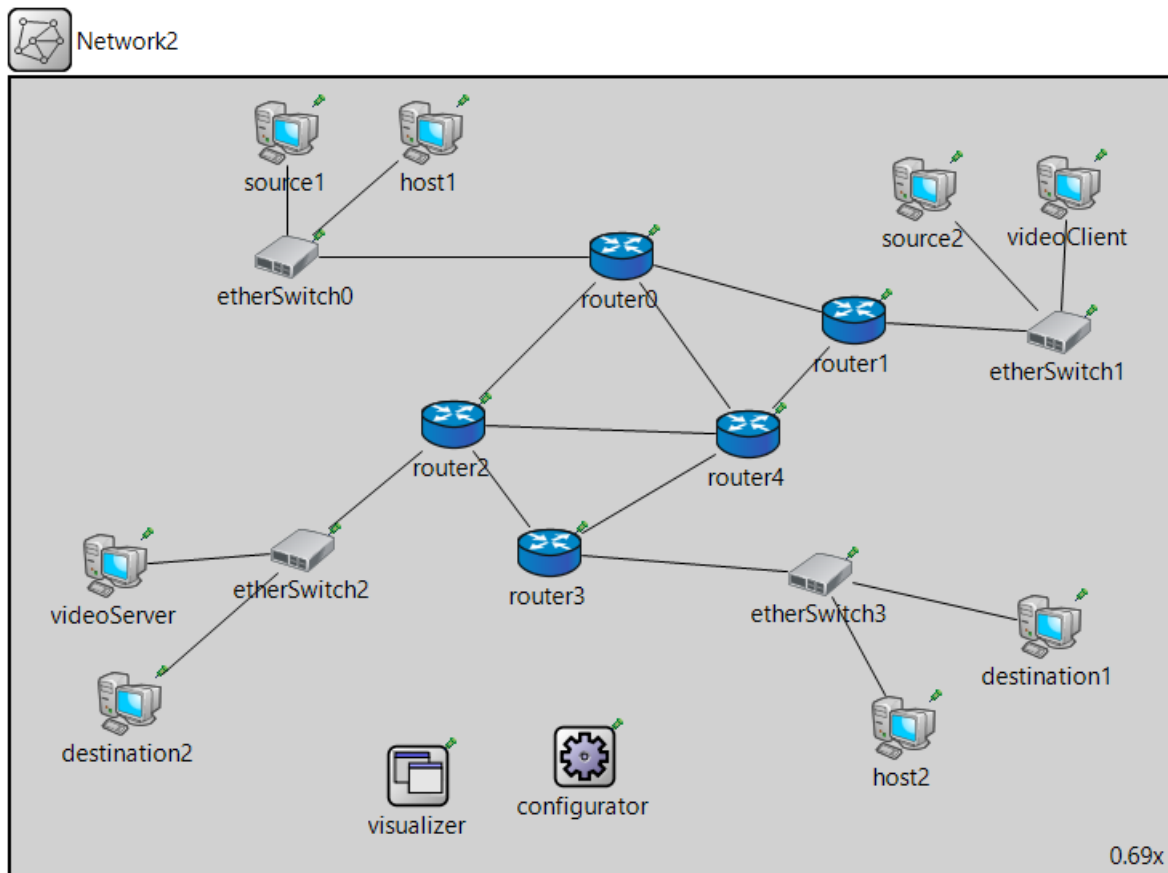


از اجرا فیلم تهیه شده است.

در ابتدا، بسته های ARP از منبع به مقصد حرکت می کند. ترافیک ARP ، NetworkRouteVisualizer را فعال نمی کند چرا که بسته های ARP از طریق لایه شبکه عبور نمی کنند. زمانی که بسته به صورت کامل توسط مقصد دریافت می شود ، یک پیکان به وسیله NetworkRouteVisualizer بین دو Host اضافه می شود که فعالیت مسیر شبکه را نشان می دهد. نام بسته همچنین روی پیکان نمایش داده می شود. پیکان به سرعت خاموش می شود زیرا پارامتر fadeOutTime از visualizer به 2s تنظیم شده است.

Filtering Network Path Activity

مطابق صورت شبکه، فایل ned را می نویسیم.



حال با توجه خواسته سوال source ها باید destination هل را ping کند و videoclient باید از سرور درخواست فایل و سرور باید برای آن ارسال کند.فایل INI را مطابق زیر می نویسم.(از showcase مربوط به این بخش در سایت INE کمک گرفتیم.)

در این قسمت NetworkRouteVisualizer فقط مسیر ترافیک ping را نمایش می دهد fadeoutTime را نیز 1.8s انتخاب کرده که تا ping بعدی از بین نرود.

```
[General]
sim-time-limit = 10s
network = Network2

**.arp.typename = "GlobalArp"

*.source*.numApps = 1

*.source*.app[0].typename = "PingApp"
*.source1.app[0].destAddr = "destination1"
*.source1.app[0].startTime = 0.2s

*.source2.app[0].destAddr = "destination2"
```

```

*.source2.app[0].startTime = 0.4s

*.videoClient.numApps = 1
*.videoClient.app[0].typename = "UdpVideoStreamClient"
*.videoClient.app[0].serverAddress = "videoServer"
*.videoClient.app[0].serverPort = 1234

*.videoServer.numApps = 1
*.videoServer.app[0].typename = "UdpVideoStreamServer"
*.videoServer.app[0].localPort = 1234
*.videoServer.app[0].sendInterval = 1s
*.videoServer.app[0].packetLen = 1500B
*.videoServer.app[0].videoSize = 6000B

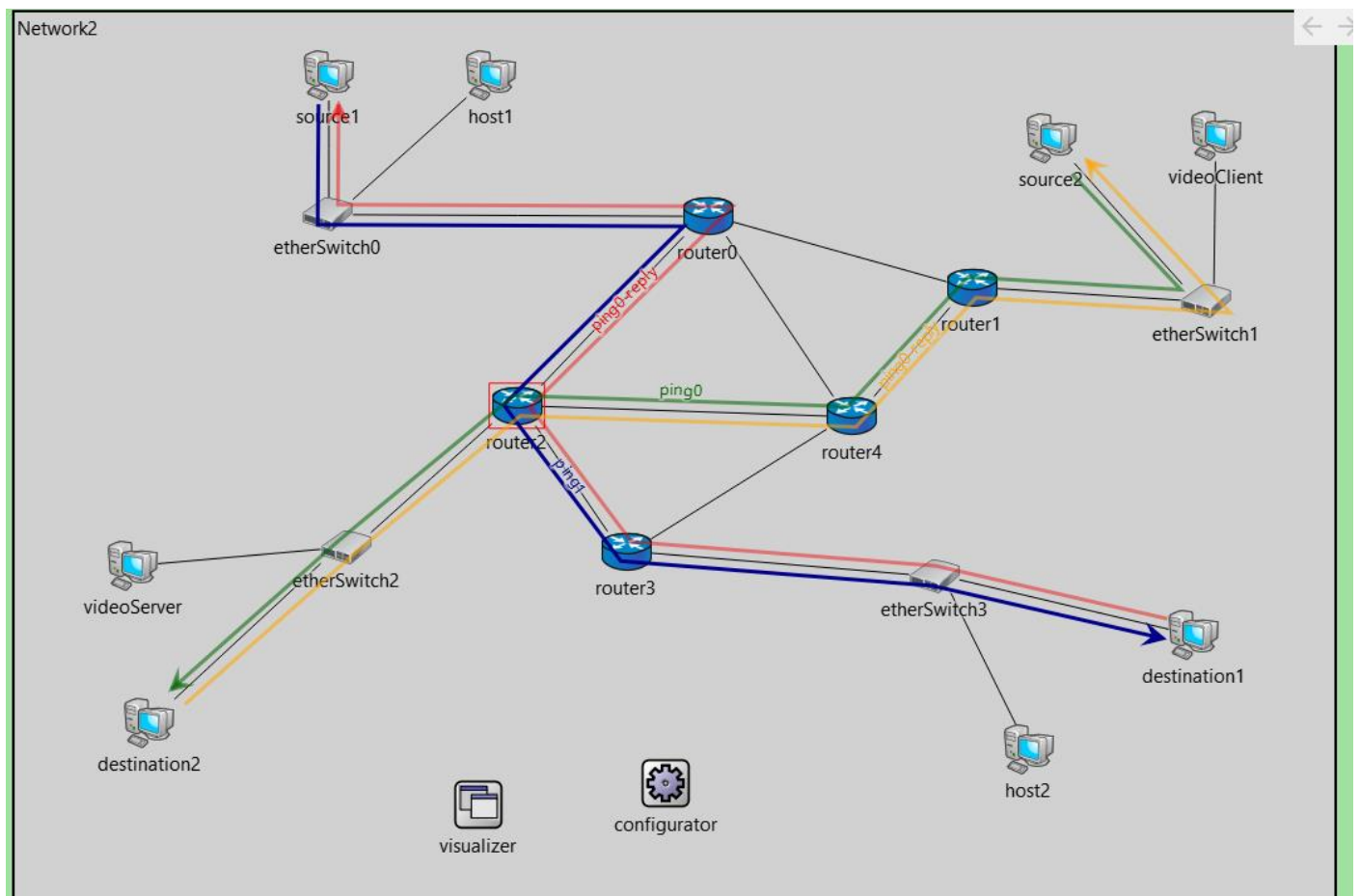
# Visualizer settings
*.visualizer.*.networkRouteVisualizer.displayRoutes = true
*.visualizer.*.networkRouteVisualizer.packetFilter = "ping*" #show ping trafic

*.visualizer.*.networkRouteVisualizer.fadeOutMode = "simulationTime"
*.visualizer.*.networkRouteVisualizer.fadeOutTime = 1.8s

```

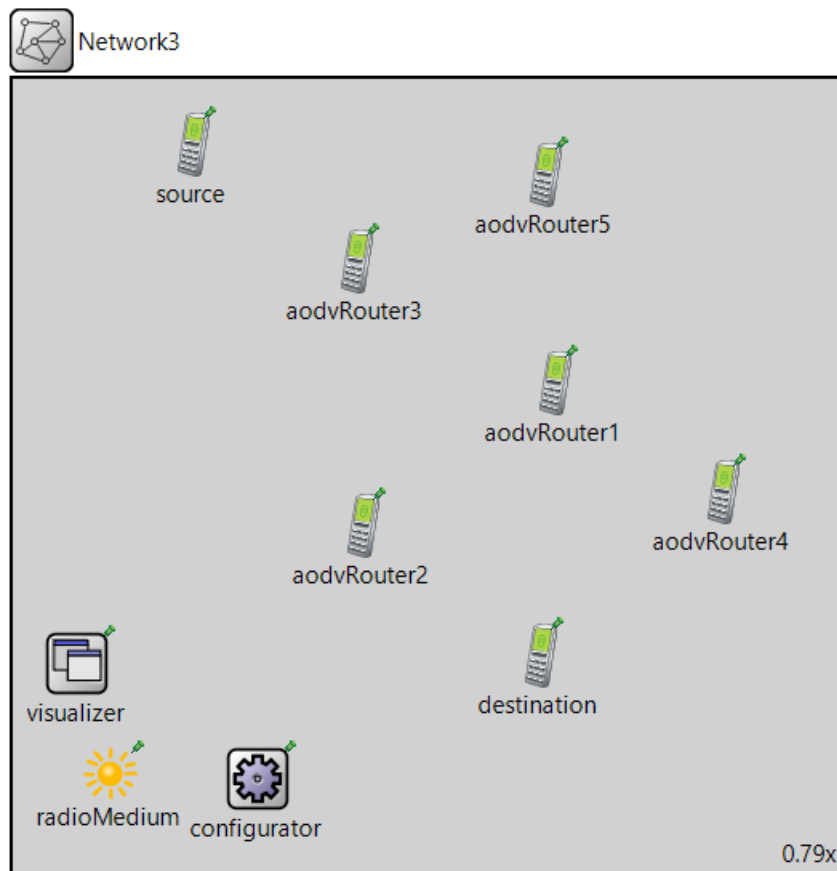
ابتدا sourceها destination ها را به ترتیب در زمان ۰.۲ و ۰.۴ ping کرده و مسیر ترافیک نیز به وسیله پیکان مشخص می شود و سپس videoclient از سرور مربوطه درخواست video stream کرده و سرور آنها به توجه به اندازه کل video در بسته هایی با اندازه مشخص شماره گذاری کرده و ارسال می کند. در این حالت با توجه به خواسته سوال مسیر ترافیک با پیکان مشخص نمی شود.

از اجرا فیلم تهیه شده است.



Visualizing Network Path Activity in a Mobile Ad-Hoc Network

مطابق صورت شبکه، فایل ned را می نویسیم.



حال با توجه خواسته سوال باید destination را ping کند. فایل INI را مطابق زیر می نویسم. (از showcase مربوط به این بخش در سایت INE کمک گرفتیم.)

```
[General]
sim-time-limit = 10s
network = Network3

**.arp.typename = "GlobalArp"

# Application parameters
*.source.numApps = 1
*.source.app[0].typename = "PingApp"
*.source.app[0].destAddr = "destination"

# mobility parameters
**.mobility.typename = "MassMobility"
**.mobility.speed = 10mps
**.mobility.angleDelta = uniform(-45deg, 45deg)
**.mobility.changeInterval = exponential(10s)
**.mobility.constraintAreaMinX = 200m
**.mobility.constraintAreaMinY = 40m
**.mobility.constraintAreaMinZ = 0m
**.mobility.constraintAreaMaxX = 630m
**.mobility.constraintAreaMaxY = 580m
**.mobility.constraintAreaMaxZ = 0m

# wlan parameters
**.wlan[*].radio.transmitter.power = 2mW

# Visualizer parameters
```

```

*.visualizer.*.mediumVisualizer.displayCommunicationRanges = true
*.visualizer.*.mediumVisualizer.nodeFilter = "source or destination"

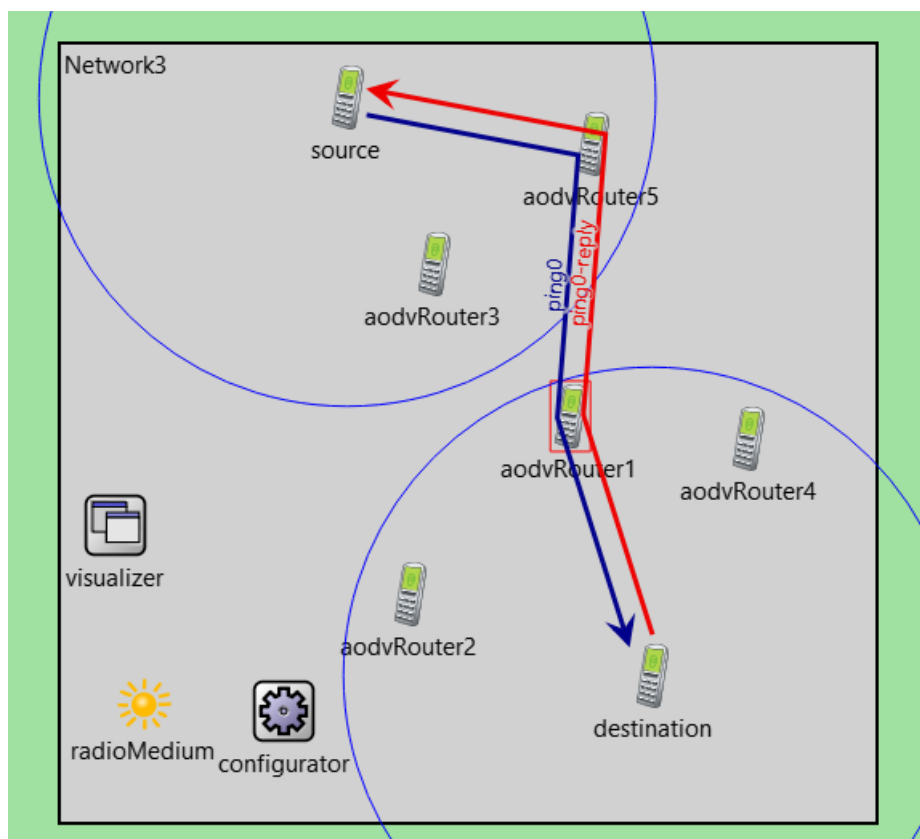
# Displaying network path activity
*.visualizer.*.networkRouteVisualizer.displayRoutes = true
*.visualizer.*.networkRouteVisualizer.packetFilter = "ping*"
*.visualizer.*.networkRouteVisualizer.fadeOutMode = "simulationTime"
*.visualizer.*.networkRouteVisualizer.fadeOutTime = 4s

```

در زیر توضیحاتی از بخش‌ها و پارامترهای کلیدی در فایل پیکربندی ارائه شده است:

پارامترهای حرکت:

- `mobility.typeName`: نوع حرکت برای تمام گره‌ها را به "MassMobility" مشخص می‌کند.
 - `mobility.speed`: سرعت ثابت تمام گره‌ها را به ۱۰ متر بر ثانیه تنظیم می‌کند.
 - `mobility.angleDelta`: تغییر زاویه حرکت گره‌ها را در بازه -۴۵ تا ۴۵ درجه تعیین می‌کند.
 - `mobility.changeInterval`: فاصله زمانی برای تغییر پارامترهای حرکت گره‌ها را بر اساس یک توزیع نمایی با میانگین ۱۰ ثانیه تنظیم می‌کند.
 - `mobility.constraintAreaMinX, mobility.constraintAreaMin, mobility.constraintAreaMinZ,`
- محدودیت‌های مستطیلی برای حرکت گره‌ها در داخل ناحیه شبیه‌سازی را مشخص می‌کند.



از اجرا فیلم تهیه شده است.

در این مدل شبیه‌سازی، گره‌ها توسط نوع AodvRouter نمایان می‌شوند و به صورت تصادفی در صحنه قرار می‌گیرند. از این گره‌ها، یکی به عنوان گره منبع تعیین می‌شود که مسئول شروع ارتباط با گره مقصد است. گره‌ها دارای برد ارتباطی هستند تا اطمینان حاصل شود که شبکه همواره متصل باقی می‌ماند. با این حال، ارتباط بین گره‌ها معمولاً از طریق مسیرهای چندگانه (multi-hop) صورت می‌گیرد، که نشان‌دهنده این است که پیام‌ها ممکن است برای رسیدن به مقصد خود از چندین گره واسط عبور کنند.

ویژگی‌های مدل شبیه‌سازی:

1. قرارگیری گره‌ها:

- گره‌ها نمونه‌های نوع AodvRouter هستند و به صورت تصادفی در داخل صحنه قرار می‌گیرند. این تصادفی بودن قرارگیری گره‌ها واقع‌گرایی رفتار شبکه‌های ad-hoc را شبیه‌سازی می‌کند.

2. گره‌های منبع و مقصد:

- یکی از گره‌ها به عنوان گره منبع شناخته می‌شود که ارتباط را با گره مقصد شروع می‌کند. این سناریوی ارتباطی به ارزیابی کارایی و کارایی پروتکل مسیریابی AODV در یک محیط متحرک و دینامیک کمک می‌کند.

3. برد ارتباطی:

- بردهای ارتباطی گره‌ها به گونه‌ای تنظیم شده‌اند که ارتباطات را ادامه دهند. با این حال، طراحی اطمینان حاصل می‌کند که ارتباط بین گره‌ها معمولاً از طریق مسیرهای چندگانه انجام می‌شود، که ویژگی شبکه‌های ad-hoc را نشان می‌دهد که ارتباط مستقیم همیشه ممکن نیست.

4. Random Node Roaming

- گره‌ها به صورت تصادفی در مرزهای تعیین شده حرکت می‌کنند. این حرکت پویا، واریابی در توپولوژی شبکه را شبیه‌سازی می‌کند و رفتار واقعی گره‌های متحرک در شبکه‌های ad-hoc را بازتاب می‌دهد.

5. AODV Routing Protocol:

- پروتکل مسیریابی انتخاب شده AODV (Ad-hoc On-Demand Distance Vector) است، که یک پروتکل مسیریابی MANET راکتیو (از نوع on-demand) است. AODV با پیام‌های Route Request (RREQ) و Route Response (RRES) عمل می‌کند. با این حال، در شبیه‌سازی مسیر این پیام‌ها در تصویرسازی ظاهر نمی‌شوند زیرا از طریق لایه شبکه عبور نمی‌کنند.

پیام‌های RREQ و RRES، بخش‌های از عملکرد AODV، در visualization ظاهر نمی‌شوند. این به این دلیل است که این پیام‌ها از طریق لایه شبکه عبور نمی‌کنند.