Data networks

DR.Pakravan

electrical engineering department

Ahmadreza Majlesara 400101861

assignment 4

June 15, 2024

Data networks

assignment 4

Ahmadreza Majlesara 400101861

OSPF Routing Protocol

OSPF is a well-known dynamic routing protocol. Below is a brief overview of some famous dynamic routing protocols:

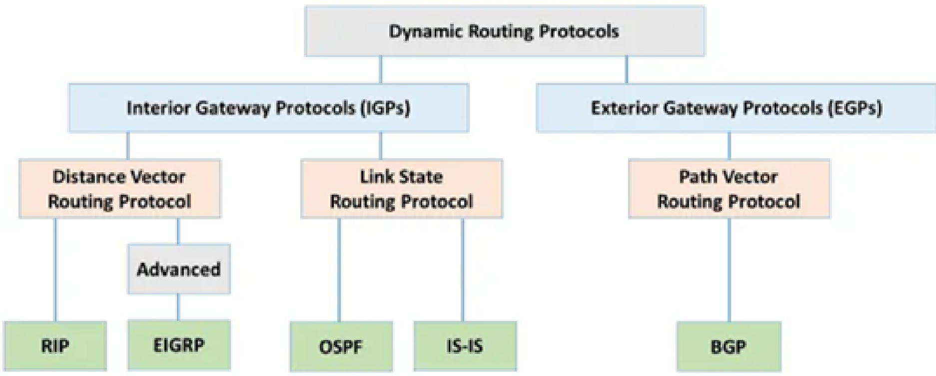


Figure 1. dynamic routing protocols

There are two versions of OSPF: OSPFv2 and OSPFv3. OSPFv2 is for IPv4, while OSPFv3 supports both IPv4 and IPv6. Before delving into the implementation of OSPF, please answer the following questions to grasp how this protocol operates:

(a) What are DR, BDR, and DROTHER, and how does OSPF choose them? What are their purposes?

(b) What is hello packet? What are hello interval and dead interval in OSPF protocol?

(c) What is a backbone area? Is it necessary for every OSPF configuration? Is there a solution if there's no physical link between a standard OSPF area and the backbone area?

(d) What are ABR and ASBR?

(e) What are LSA and LSDB? Explain about different types of LSA and their purposes.

(f) What is passive interface in OSPF and when should we use it?

Now, using the INET framework, implement the OSPF protocol in the network topology below. Confirm that your configuration works by demonstrating the success of the following pings: Host1 -> Host3 / Host4 -> Host3 / Host2 -> Host5

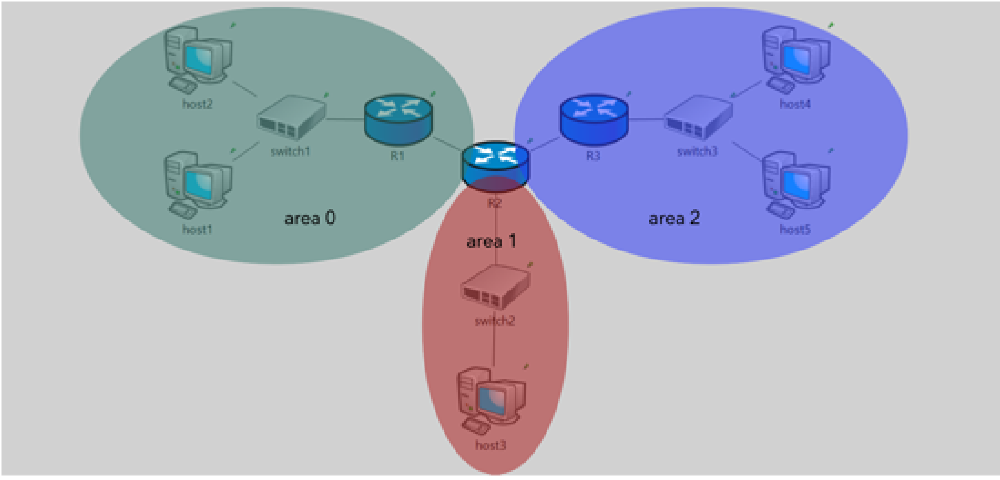


Figure 2. Network topology

solution

first lets answer the questions:

(a) :the Designated Router (DR) serves as the central point for routing updates within a network segment, significantly reducing the overhead associated with maintaining multiple adjacencies. The Backup Designated Router (BDR) acts as a standby to the DR, poised to take over its duties if the DR fails, thereby ensuring the stability and continuity of the network. Regular routers, known as DROTHERs, do not assume the roles of DR or BDR. Instead, they communicate through the DR and BDR to efficiently propagate routing information throughout the network, maintaining overall network efficiency.

(b) :Hello packets are used to establish and maintain neighbor relationships between routers, containing critical information such as router ID, area ID, Hello interval, Dead interval, router priority, and known neighbors. The Hello interval, typically 10 seconds on broadcast networks, defines how often these packets are sent, while the Dead interval, usually four times the Hello interval, specifies how long a router waits without receiving a Hello packet before declaring a neighbor dead. Consistent Hello and Dead intervals across all routers on a network segment are crucial for forming adjacencies, ensuring synchronized neighbor detection, and maintaining overall network stability. These intervals and the Hello packets themselves are essential for reliable and efficient OSPF operation.

(c) : the backbone area (Area 0) is essential as it acts as the central hub for inter-area routing, ensuring efficient exchange of routing information between different OSPF areas. Every OSPF configuration requires all areas to connect to Area 0, maintaining a hierarchical structure that enhances network scalability and simplifies routing. If there is no physical link between a standard OSPF area and the backbone area, virtual links can be used to create a logical connection, thereby preserving the necessary network hierarchy and ensuring continuous routing information flow. These aspects are crucial for the proper operation and efficiency of an OSPF network.

(d) :an Area Border Router (ABR) is crucial as it connects multiple OSPF areas, managing and summarizing routing information between them to enhance scalability and efficiency. It maintains separate link-state databases for each area it connects. An Autonomous System Boundary Router (ASBR) is essential for linking an OSPF network to external networks or different routing protocols, redistributing external routing information into the OSPF domain. This integration ensures comprehensive network connectivity and the exchange of routing information across various networks and protocols. Both ABRs and ASBRs play vital roles in optimizing and extending OSPF network functionality.

(e) :Link State Advertisements (LSAs) are crucial data packets that exchange routing and topology information between routers, forming the Link State Database (LSDB) that each router maintains to map the network topology. LSAs come in various types: Type 1 (Router LSA) details a router's interfaces within an area; Type 2 (Network LSA) is generated by the Designated Router to describe all routers on a network segment; Type 3 (Summary LSA) and Type 4 (ASBR Summary LSA) are generated by Area Border Routers to summarize inter-area routes and advertise ASBRs, respectively; Type 5 (AS External LSA) describes routes to external networks and is generated by ASBRs; Type 7 (NSSA LSA) is used within Not-So-Stubby Areas to describe external routes. These LSAs ensure OSPF routers have a consistent, updated view of the network, enabling efficient and accurate routing decisions.

(f) : A passive interface in OSPF is a network interface configured to suppress the sending of OSPF Hello packets, thereby preventing the formation of OSPF neighbor relationships and the exchange of routing updates on that interface. This feature is crucial for enhancing security by restricting OSPF information on untrusted or external networks, improving performance by reducing unnecessary OSPF traffic, and simplifying network management by excluding interfaces that connect to end devices (like PCs or servers) that do not require dynamic routing. Using passive interfaces helps maintain a secure, efficient, and manageable OSPF network.

now lets implement the OSPF protocol in Omnet++. first the network topology is shown in figure 3.

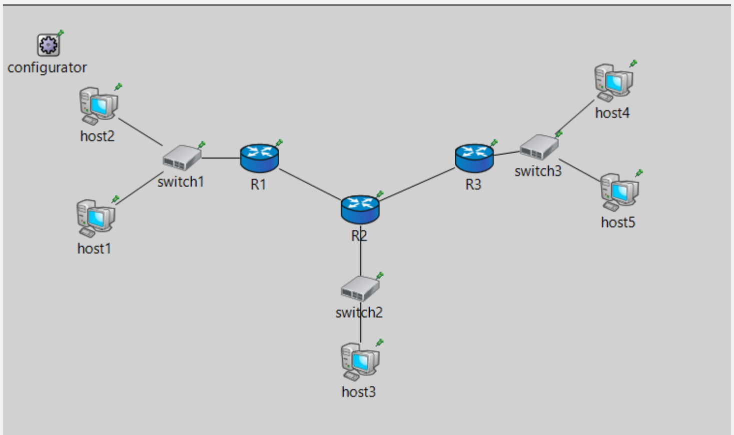


Figure 3. Network topology

as it can be seen this network has 5 hosts,3 switches and 3 routers. the connections between these and the definition of each submodule doesnt need explanation so i just show the xml configuration of .NED file in figure 4.

```
submodules:
    configurator: Ipv4NetworkConfigurator {
        parameters:

            config = xml("<config>"+
                "<interface among='host1 host2 R1' address='192.168.1.x' netmask='255.255.255.x' />"+
                "<interface among='host4 host5 R3' address='192.168.2.x' netmask='255.255.255.x' />"+
                "<interface among='host3 R1 R2 R3' address='192.168.3.x' netmask='255.255.255.x' />"+
                "<route hosts='host1 host3' destination='*' netmask='0.0.0.0' interface='eth0' />"+
                "<route hosts='host4 host3' destination='*' netmask='0.0.0.0' interface='eth0' />"+
                "<route hosts='host2 host5' destination='*' netmask='0.0.0.0' interface='eth0' />"+
                "</config>");
            addStaticRoutes = false;
            addDefaultRoutes = false;
            @display("p=47,41;is=s");
    }
```

Figure 4. IP configuration in .NED file

then we need a XML file to config all the IP addresses and net masks in routers and hosts to make them run OSPF protocol. the XML file is shown in figure 5.

```
<?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="192.168.3.0" 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="R1" RFC1583Compatible="true">
    <BroadcastInterface ifName="eth0" areaID="0.0.0.1" />
    <PointToPointInterface ifName="eth1" areaID="0.0.0.0" />
</Router>

<Router name="R3" RFC1583Compatible="true">
    <BroadcastInterface ifName="eth0" areaID="0.0.0.2" />
    <PointToPointInterface ifName="eth1" areaID="0.0.0.0" />
</Router>

<Router name="R2" RFC1583Compatible="true">
    <BroadcastInterface ifName="eth0" areaID="0.0.0.3" />
    <PointToPointInterface ifName="eth2" areaID="0.0.0.2" />
    <PointToPointInterface ifName="eth1" areaID="0.0.0.1" />
</Router>
</OSPFASConfig>
```

Figure 5. XML configuration file

as it can be seen in this figure the backbone area has areaID=0.0.0.0. the area associated with R1(left router) has areaID=0.0.0.1 and the area associated with R2(right router) has areaID=0.0.0.2. we also define the areaID for each ports of routers in this file. we use BroadcastInterface for hosts and PointToPointInterface for routers.
at last we difine the .ini file to start our simulation. in .ini file we have to import the .XML file to configure our network. not that for the IP addresses of destination i got their address from the simulation window which is shown in figure 7.

```
[General]
network = OSPF

sim-time-limit = 200s

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

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

*.host1.app[0].destAddr = "192.168.3.10"
*.host1.app[0].sendInterval = 1s
*.host1.app[0].startTime = 0.2s

*.host4.app[0].destAddr = "192.168.3.10"
*.host4.app[0].sendInterval = 1s
*.host4.app[0].startTime = 0.4s

*.host2.app[0].destAddr = "192.168.2.3"
*.host2.app[0].sendInterval = 1s
*.host2.app[0].startTime = 0.6s
```

Figure 6. INI configuration file

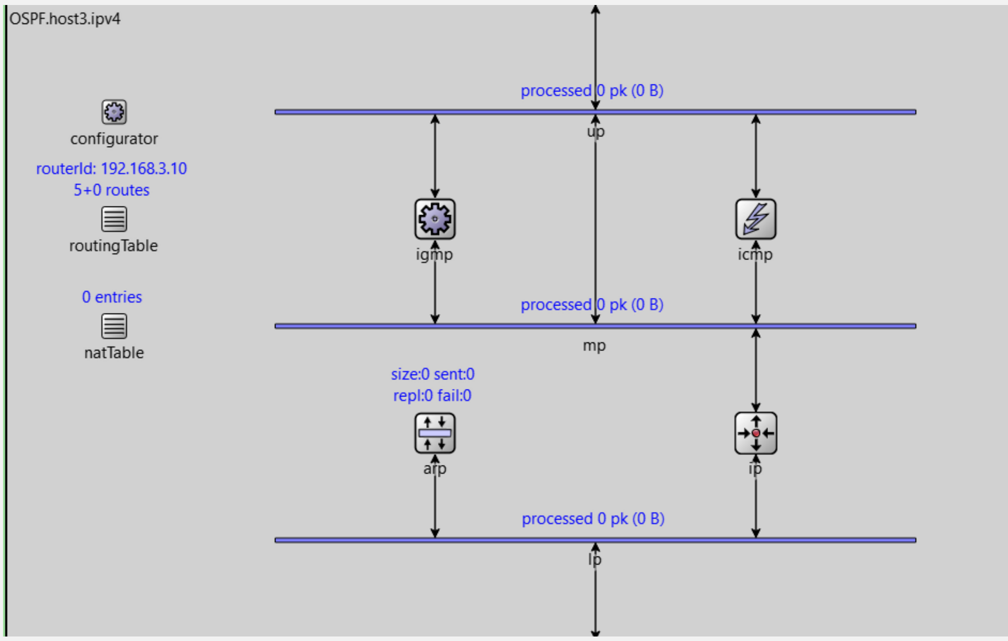


Figure 7. extracted IP address for host 3 in simulation window

at last i ran the simulation. the hellopackets were shown in the start of simulation which means that the OSPF protocol is working properly. and then i recoreded a video which i send you alongside this report. the video is called ping1.

ad-hoc Routing Protocols

Due to the dynamic nature of Mobile ad-hoc networks (MANETs), finding a routing protocol that suits the environment is a challenging problem. Here `omnet++` mobility and MANET routing packages can assist in simulating these networks. In this question, we aim to simulate a dynamic MANET and identify its suitable and robust routing protocols.

Ad-hoc On-Demand Distance Vector Routing (AODV)

Ad-hoc On-Demand Distance Vector (AODV) is a reactive routing protocol specifically designed for routing in mobile devices. It operates in mobile ad-hoc networks (MANETs) and other wireless ad-hoc networks. AODV was jointly developed by Charles Perkins and Elizabeth Royer (now Elizabeth Belding) and was first published in 1999. Now, answer the following questions:

1. Why is it important to determine the number of hops between the source and destination in each routing packet?
2. What purpose do the Source and Destination Sequence Numbers serve in each routing packet?
3. What happens when a link fails after successfully establishing an on-demand route?
4. What are the advantages and disadvantages of reactive routing protocols such as AODV?

solution

lets start with the answer to the questions:

1 In AODV, determining the number of hops between the source and destination in each routing packet is crucial for selecting the most efficient route, preventing routing loops, optimizing network resources, managing routing tables effectively, and ensuring quality of service. Fewer hops typically mean lower latency, reduced resource consumption, and better overall network performance.

2 Source and Destination Sequence Numbers in each routing packet serve to ensure the freshness and accuracy of the routing information. The Source Sequence Number is used to maintain the order of route requests from the source, preventing the use of outdated information. The Destination Sequence Number helps determine the most recent route to the destination, ensuring that the routing information is up-to-date and loop-free. By using these sequence numbers, AODV can effectively manage dynamic network changes and maintain reliable routes.

3 When a link fails after successfully establishing an on-demand route in AODV, the detecting node generates a Route Error (RERR) message, which is sent to upstream nodes that use the broken link. These nodes mark the affected routes as invalid in their routing tables. If the source still needs the route, it initiates a new Route Request (RREQ) to discover an alternative path, ensuring continuous and reliable communication.

4 Reactive routing protocols like AODV have the advantage of reduced overhead and better scalability since routes are established on-demand, making them adaptable to highly dynamic environments such as mobile ad-hoc networks. However, they can introduce initial latency during route discovery, and frequent route discoveries and repairs can lead to increased control message overhead and resource consumption, particularly in networks with high mobility or many nodes.

simulation

We follow the showcase of MANET routing protocols explained in here. The tutorials in the wireless section of the INET path are also helpful. Please read these tutorials and modify the code in `.ini` and `.ned` files to create the desired topology and conditions stated below.

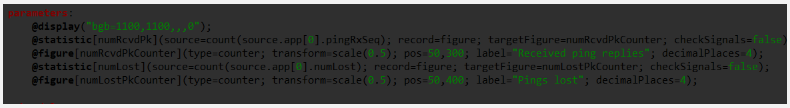
- Set the maximum network area to 1100×1100 and position the MANET mobile routers as shown in figure 8.
- Display the communication range of the devices in the simulation. Set the transmission power of the device to 2mW.
- Enable linear mobility for the devices (including the source and destination) and set the destination movement angle to 270 degrees.
- Increase the scale of the ping indicator and place it within the same frame as the network in the simulation.
- Turn off both routing timers.
- Run the Simulation.

Capture screenshots that show the ping route arrows and the ping indicator. Then, proceed with the simulation until you encounter events that indicate route failures. You can use the simulation recorder or other recording methods to capture these events.

solution

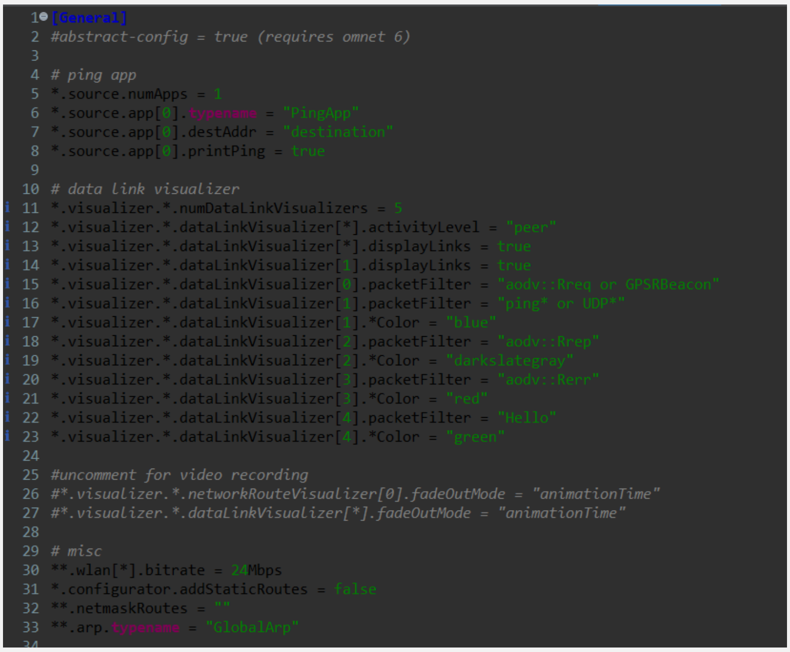
for the simulation in the OMNET++ environment, we need to follow the steps mentioned in the question. The following steps are taken to simulate the MANET routing protocols. first the network topology is shown in figure 8 . the network has 13 hosts which are MANETRouter. for the visualization we also imported a visualizer. the .NED file doesnt have any connection but the parameters of the network are shown in figure 9.for the .ini file the code for the visualization is shown in figure 10. we use the code in figure 11 to have mobility. set the transmission power. set the maximum area to 1100 × 1100. the output for this code is shown in figure 11. in this figure you can see the ping route arrows and the ping indicator. the ping indicator is placed within the same frame as the network. the routing timers are turned off. the event that indicates route failure is shown in figure 12.

Figure 8. network topology



```
parameters:
    @display("bgb=1100,1100,,,0");
    @statistic[numRcvdPk](source=count(source.app[0].pingRxSeq); record=figure; targetFigure=numRcvdPkCounter; checkSignals=false)
    @figure[numRcvdPkCounter](type=counter; transform=scale(0.5); pos=50,300; label="Received ping replies"; decimalPlaces=4);
    @statistic[numLost](source=count(source.app[0].numLost); record=figure; targetFigure=numLostPkCounter; checkSignals=false);
    @figure[numLostPkCounter](type=counter; transform=scale(0.5); pos=50,400; label="Pings lost"; decimalPlaces=4);
```

Figure 9. parameters of the network



```
[General]
#abstract-config = true (requires omnet 6)

# ping app
*.source.numApps = 1
*.source.app[0].typename = "PingApp"
*.source.app[0].destAddr = "destination"
*.source.app[0].printPing = true

# data link visualizer
*.visualizer.*.numDataLinkVisualizers = 5
*.visualizer.*.dataLinkVisualizer[*].activityLevel = "peer"
*.visualizer.*.dataLinkVisualizer[*].displayLinks = true
*.visualizer.*.dataLinkVisualizer[1].displayLinks = true
*.visualizer.*.dataLinkVisualizer[0].packetFilter = "aodv::Rreq or GPSRBeacon"
*.visualizer.*.dataLinkVisualizer[1].packetFilter = "ping* or UDP*"
*.visualizer.*.dataLinkVisualizer[1].*Color = "blue"
*.visualizer.*.dataLinkVisualizer[2].packetFilter = "aodv::Rrep"
*.visualizer.*.dataLinkVisualizer[2].*Color = "darkslategray"
*.visualizer.*.dataLinkVisualizer[3].packetFilter = "aodv::Rerr"
*.visualizer.*.dataLinkVisualizer[3].*Color = "red"
*.visualizer.*.dataLinkVisualizer[4].packetFilter = "Hello"
*.visualizer.*.dataLinkVisualizer[4].*Color = "green"

#uncomment for video recording
#*.visualizer.*.networkRouteVisualizer[0].fadeOutMode = "animationTime"
#*.visualizer.*.dataLinkVisualizer[*].fadeOutMode = "animationTime"

# misc
**.wlan[*].bitrate = 24Mbps
*.configurator.addStaticRoutes = false
**.netmaskRoutes = ""
**.arp.typename = "GlobalArp"
```

Figure 10. .ini file

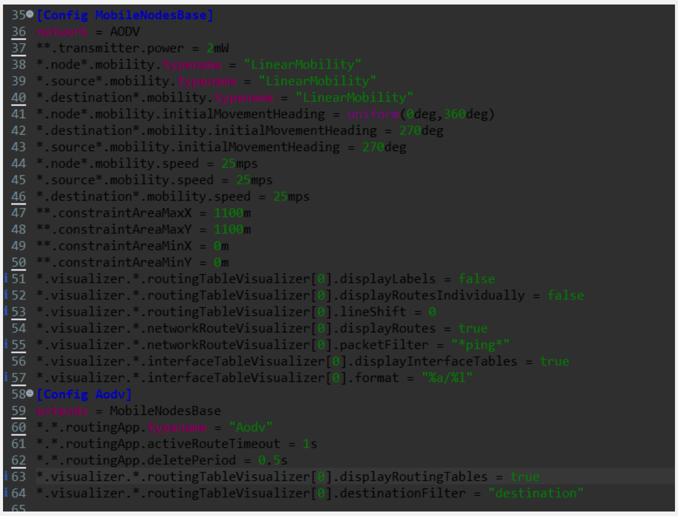


Figure 11. code for the subjects mentioned

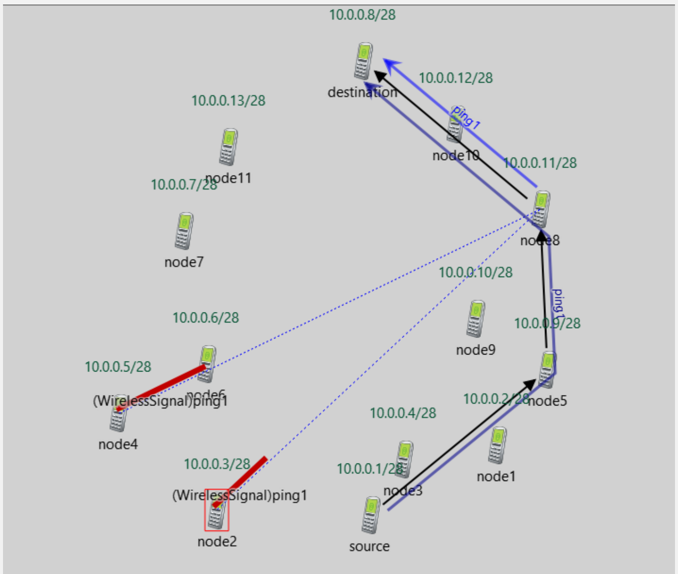


Figure 12. output of the code showing route and ping

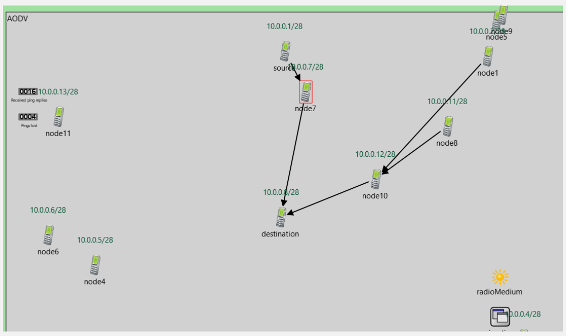


Figure 13. event that indicates route failure(new route is utilized)

Destination-Sequenced Distance-Vector (DSDV)

DSDV routing protocol is a table-driven and proactive routing protocol designed by Charles E. Perkins and Pravin Bhagwat in 1994. Read this and simulate the protocol using the last topology with the same settings and answer the following questions: DSDV routing protocol is a table-driven and proactive routing protocol designed by Charles E. Perkins and Pravin Bhagwat in 1994. Read this and simulate the protocol using the last topology with the same settings and answer the following questions:

1. What are the trade-offs associated with the hello packet interval time?
2. Provide a comparison between AODV and DSDV and explain the scenarios in which DSDV is a more suitable choice, as well as situations where the other protocol is preferable.
3. According to the last question, which one of the protocols suits sensor networks? Why?

Capture screenshots that represent the ping route arrows and the ping indicator. For skipping the hello events you can change ping and mobility time intervals.

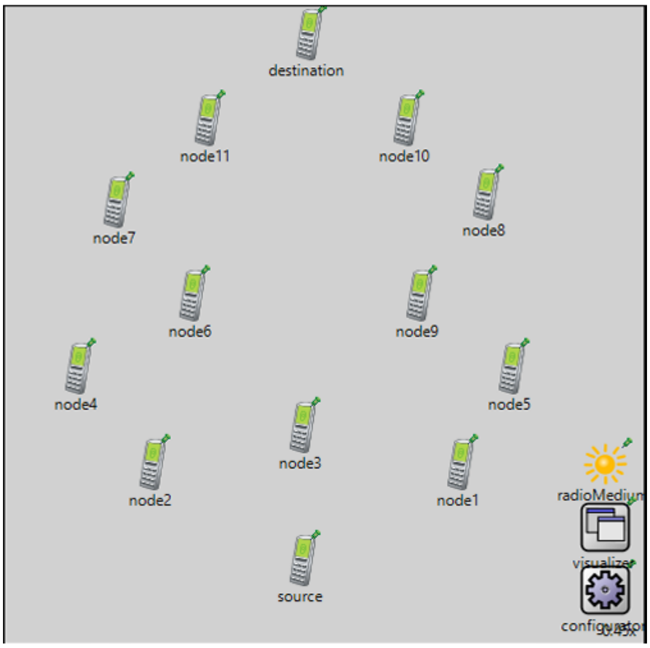


Figure 14. MANET topology

solution

lets start from questions:

1 In DSDV, the hello packet interval time involves trade-offs between network overhead and route accuracy. Shorter intervals increase the frequency of hello packets, which helps maintain more accurate and up-to-date route information but also leads to higher network overhead and increased battery consumption in mobile devices. Conversely, longer intervals reduce overhead and conserve energy but can result in less timely route updates, potentially leading to outdated routing information and decreased network performance.

2 AODV is a reactive routing protocol that establishes routes on-demand, making it more suitable for highly dynamic and large networks due to its lower overhead during idle periods and quick adaptability to changing topologies. In contrast, DSDV is a proactive routing protocol that maintains routes to all nodes at all times, making it more suitable for static or low-mobility networks, smaller networks, and applications requiring immediate route availability with low latency. DSDV's constant updates ensure up-to-date routing tables but can be less efficient in highly dynamic networks due to higher overhead.

3 In sensor networks, which typically have low mobility and require stable and consistent routes, DSDV is indeed more suitable.

the network topology and .NED file is as the same as befor. the .ini file is shown in figure 15. the output of the code is shown in figure 16.

```
[Config Dsdv]
extends = MobileNodesBase

# routing protocol
*.*.routing.typename = "Dsdv"
**.transmitter.power = 2mW

# node movement
*.node*.mobility.typename = "LinearMobility"
*.source*.mobility.typename = "LinearMobility"
*.destination*.mobility.typename = "LinearMobility"

*.node*.mobility.initialMovementHeading = uniform(0deg,360deg)
*.destination*.mobility.initialMovementHeading = 270deg
*.source*.mobility.initialMovementHeading = 270deg
*.node*.mobility.speed = 25mps
*.source*.mobility.speed = 25mps
*.destination*.mobility.speed = 25mps

**.constraintAreaMaxX = 1100m
**.constraintAreaMaxY = 1100m
**.constraintAreaMinX = 0m
**.constraintAreaMinY = 0m

*.*.routing.helloInterval = 1s
*.*.routing.routeLifetime = 2s

# ping app
*.source.app[0].sendInterval = 0.5s

# visualization
*.visualizer.*.routingTableVisualizer[0].displayRoutingTables = true

```

Figure 15. .ini file

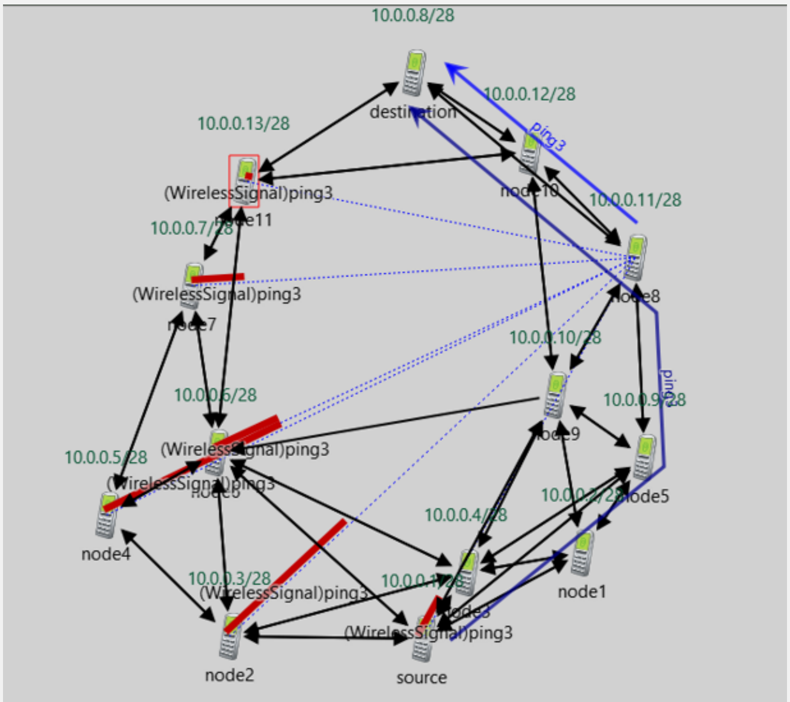


Figure 16. output of the code showing route and ping