Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam
(An Autonomous Institution, Affiliated to Anna University, Chennai)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Date: 24-07-2026

<table>
<tr><td>Course Code</td><td colspan="2">UCS3511</td><td>Course Name</td><td colspan="6">Networks Lab</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2" colspan="2">Lab</td><td rowspan="2">Course Category</td><td rowspan="2" colspan="2">Professional Core (PC)</td><td>L</td><td>T</td><td>P</td><td>TW & SL</td><td>TH</td><td>C</td></tr>
<tr><td>0</td><td>0</td><td>45</td><td>0</td><td>45</td><td>1.5</td></tr>
<tr><td>Regulation</td><td colspan="3">2024</td><td colspan="2">Academic Year & Batch</td><td colspan="6">2026-27 & 2024-28</td></tr>
<tr><td colspan="4">Degree and Branch</td><td colspan="8">B.E. Computer Science & Engineering</td></tr>
<tr><td colspan="4">Semester</td><td>V</td><td>Section</td><td colspan="2">A, B & C</td><td>K3, CO2</td><td colspan="2">PI: 1.1.2, 1.3.1, 1.4.1, 2.1.2, 2.1.3, 2.3.1, 3.1.1, 4.3.1, 5.1.1, 12.2.1, 12.3.1, 12.4.1 & 13.4.1</td></tr>
<tr><td>Assignment No.</td><td colspan="2">11</td><td>Assigned Date</td><td colspan="3">28-09-2026</td><td colspan="2">Due Date</td><td colspan="2">06-10-2026</td></tr>
</table>

Lab Assignment: Simulation of Common Network Topologies, Traffic Generation and Packet-Level Analysis Using Mininet and Wireshark

Application: Network Topology and Traffic Analysis

Scenario

A network administrator needs to study how different network topologies influence communication between hosts. Using **Mininet**, construct and simulate common network topologies, generate network traffic between hosts, and capture the exchanged packets using **Wireshark**. Analyze the packet-level behaviour by examining protocols, packet sequence, source and destination addresses, packet size, transmission timing, and communication flow.

The experiment should demonstrate how changes in topology and traffic patterns affect the observed packet-level behaviour.

1a. Create Network Topologies Using Mininet

Using Mininet, implement the following network topologies:

1. **Bus Topology**
2. **Star Topology**
3. **Ring Topology**
4. **Tree Topology**
5. **Linear Topology**

For each topology:

- Create the required hosts and switches.
- Assign IP addresses to the hosts.
- Establish the required links.
- Display the topology using Mininet commands.
- Verify connectivity using `ping`.
- Record the number of hosts, switches, and links.

1b. Generate Network Traffic

Generate traffic between selected hosts using Mininet commands and appropriate traffic-generation utilities.

Perform:

- ICMP traffic using `ping`.
- TCP traffic using `iperf`.
- UDP traffic using `iperf`.
- Multiple simultaneous traffic flows.

Record:

- Source host
- Destination host
- Protocol
- Packet/segment size
- Number of packets transmitted
- Transmission rate
- Packet loss
- Communication time

1c. Capture Packets Using Wireshark

Capture the generated traffic using **Wireshark** or an appropriate Mininet interface capture mechanism.

For each experiment, identify:

- Source IP address
- Destination IP address
- Source MAC address
- Destination MAC address
- Protocol
- Packet length
- Sequence/identifier information where applicable
- Arrival time
- Response time
- TCP flags for TCP traffic
- ICMP request/reply information

Use suitable Wireshark display filters such as:

```
icmp
tcp
udp
ip.addr == 10.0.0.1
tcp.flags
```

1d. Analyze Packet-Level Behaviour

For each topology, examine the captured packets and analyze:

1. Packet flow between source and destination.
2. Sequence of request and response packets.
3. Source and destination addressing.
4. Protocol-specific packet fields.
5. Packet transmission timing.
6. TCP connection establishment and termination.
7. UDP communication behaviour.
8. ICMP request and reply behaviour.
9. Packet loss or retransmission, if observed.
10. Differences in packet behaviour between topologies.

1e. Analyze Network Connectivity

Use Mininet commands such as:

```
nodes
net
links
dump
pingall
```

Determine:

- Number of active hosts.
- Number of switches.
- Available communication paths.
- Connectivity between hosts.
- Effect of topology on communication paths.

1f. Compare Different Topologies

For each topology, prepare a comparative observation based on:

Parameter	Observation
Number of Hosts	
Number of Switches	
Number of Links	

Parameter	Observation
Communication Path	
Packets Generated	
Packets Received	
Packet Loss	
Average RTT	
TCP Throughput	
UDP Throughput	
Retransmissions	

---

2. Sample Network Configuration

Use the following basic configuration for laboratory verification.

Topology	Hosts	Switches	Example Configuration
Bus	4	1	H1–S1–H2/H3/H4
Star	4	1	H1,H2,H3,H4 connected to S1
Ring	4	4	H1–S1–S2–S3–S4–S1
Tree	6	3	Root switch with two child switches
Linear	4	4	H1–S1–S2–S3–S4–H4

Host IP Addressing

Host	IP Address
H1	10.0.0.1/24
H2	10.0.0.2/24
H3	10.0.0.3/24
H4	10.0.0.4/24
H5	10.0.0.5/24
H6	10.0.0.6/24

---

3. Test Cases for Laboratory Verification

TC1 – ICMP Traffic in Star Topology

Topology: Star
Source: H1
Destination: H4
Packets: 20 ICMP packets
Packet size: 64 bytes

Execute:

```
ping -c 20 10.0.0.4
```

Expected observation:

- ICMP Echo Request packets should be observed.
- ICMP Echo Reply packets should be observed.
- Source and destination IP addresses should correspond to H1 and H4.
- RTT should be displayed by Mininet.
- ICMP packets should be visible in Wireshark.

TC2 – TCP Traffic in Linear Topology

Topology: Linear
Source: H1
Destination: H4
Protocol: TCP
Duration: 10 seconds
Traffic rate: 10 Mbps

Generate TCP traffic using:

```
iperf
```

Capture the packets in Wireshark.

Analyze:

- TCP three-way handshake.
- SYN, SYN-ACK and ACK packets.
- TCP data packets.
- Sequence and acknowledgement numbers.
- TCP throughput.
- Connection termination.

TC3 – UDP Traffic in Tree Topology

Topology: Tree
Source: H1
Destination: H6
Protocol: UDP
Duration: 10 seconds
Traffic rate: 5 Mbps

Generate UDP traffic using:

```
iperf
```

Expected observation:

- UDP packets should be captured continuously.
- No TCP handshake should be observed.
- Source and destination UDP port numbers should be identified.
- Packet size and transmission rate should be recorded.

TC4 – Simultaneous TCP and UDP Traffic

Topology: Ring

Generate simultaneous traffic:

```
H1 → H3 : TCP
H2 → H4 : UDP
```

Duration: 10 seconds.

Capture the traffic using Wireshark and identify:

- TCP packets.
- UDP packets.
- Source/destination pairs.
- Packet transmission sequence.
- Packet arrival timing.
- TCP and UDP traffic rate.

Expected observation:

Multiple traffic flows should be visible in the packet capture, allowing packet-level identification of the two protocols.

TC5 – Multi-Host ICMP Traffic

Topology: Star

Generate simultaneous communication:

```
H1 → H2
H1 → H3
H1 → H4
H2 → H4
```

Send 20 ICMP packets for each flow.

Analyze:

- Total packets transmitted.
- Total packets received.
- RTT for each communication pair.
- Packet sequence.
- Source and destination addresses.
- Packet timing in Wireshark.

TC6 – TCP Retransmission Observation

Topology: Linear

Generate TCP traffic between the end hosts and capture the traffic in Wireshark.

Apply the filter:

```
tcp.analysis.retransmission
```

Record whether retransmitted TCP segments are observed.

Analyze:

- Sequence number.
- Acknowledgement number.
- TCP flags.
- Retransmission occurrence.
- Time interval between original and retransmitted packets.

---

4. Sample Input

```
MININET NETWORK SIMULATION

Topology        : Star
Number of Hosts : 4
Number of Switches : 1

Hosts:
H1 : 10.0.0.1/24
H2 : 10.0.0.2/24
H3 : 10.0.0.3/24
H4 : 10.0.0.4/24

Traffic:
Source          : H1
Destination     : H4
Protocol        : ICMP
Packets         : 20
```

```
Packet Size     : 64 Bytes
```

Sample Mininet Output

```
*** Creating network
*** Adding hosts:
h1 h2 h3 h4

*** Adding switch:
s1

*** Creating links:
(h1, s1)
(h2, s1)
(h3, s1)
(h4, s1)

*** Starting network
*** Ping test

h1 -> h2 h3 h4
h2 -> h1 h3 h4
h3 -> h1 h2 h4
h4 -> h1 h2 h3

*** Connectivity test completed
```

---

5. Sample Wireshark Packet Output

For an ICMP communication between H1 and H4:

No.	Time	Source	Destination	Protocol	Length	Information
1	0.000	10.0.0.1	10.0.0.4	ICMP	98	Echo Request
2	0.001	10.0.0.4	10.0.0.1	ICMP	98	Echo Reply
3	1.002	10.0.0.1	10.0.0.4	ICMP	98	Echo Request
4	1.003	10.0.0.4	10.0.0.1	ICMP	98	Echo Reply

Sample TCP Packet Sequence

```
H1 → H4 : TCP SYN
H4 → H1 : TCP SYN, ACK
H1 → H4 : TCP ACK
H1 → H4 : TCP Data
H4 → H1 : TCP ACK
H1 → H4 : TCP Data
H4 → H1 : TCP ACK
...
H1 → H4 : TCP FIN
```

```
H4 → H1 : TCP ACK
```

---

6. Performance Observation Table – Sample Results

The following values are **illustrative laboratory reference values**; actual results depend on the Mininet configuration, host system, traffic rate, and packet-capture conditions.

TC	Topology	Protocol	Packets Sent	Packets Received	Packet Loss	Avg. RTT	Throughput
TC1	Star	ICMP	20	20	0%	0.42 ms	—
TC2	Linear	TCP	—	—	0%	1.18 ms	9.72 Mbps
TC3	Tree	UDP	6250	6245	0.08%	0.91 ms	4.96 Mbps
TC4	Ring	TCP + UDP	—	—	0%	1.35 ms	TCP: 9.45 Mbps / UDP: 4.82 Mbps
TC5	Star	ICMP	80	80	0%	0.48 ms	—
TC6	Linear	TCP	—	—	Varies	1.52 ms	8.91 Mbps

---

7. Wireshark Packet-Level Observation

Test Case	Protocol	Source → Destination	Packet Type	Packet Size	Important Field Observed
TC1	ICMP	H1 → H4	Echo Request	98 B	ICMP Type 8
TC1	ICMP	H4 → H1	Echo Reply	98 B	ICMP Type 0
TC2	TCP	H1 → H4	SYN	~74 B	SYN Flag
TC2	TCP	H4 → H1	SYN-ACK	~74 B	SYN + ACK
TC2	TCP	H1 → H4	ACK/Data	Variable	Sequence Number
TC3	UDP	H1 → H6	UDP Data	Variable	Source/Destination Port
TC4	TCP	H1 → H3	TCP Data	Variable	Sequence/ACK
TC4	UDP	H2 → H4	UDP Data	Variable	UDP Length
TC6	TCP	H1 → H4	Retransmission	Variable	Retransmission Flag

---

8. Analysis

Analyze the obtained Mininet and Wireshark results based on:

1. Effect of topology on the communication path.
2. Number of switches and links involved in each topology.
3. Difference between ICMP, TCP and UDP packet behaviour.

4. TCP three-way handshake observed in Wireshark.
5. Sequence and acknowledgement numbers in TCP packets.
6. Difference between TCP and UDP traffic generation.
7. Packet timing and RTT observed in Wireshark.
8. Occurrence of retransmissions, if any.
9. Effect of simultaneous traffic flows on packet transmission.
10. Relationship between topology, traffic pattern and packet-level behaviour.

Students should support the analysis using **Mininet command output and Wireshark packet captures**, rather than only reporting numerical values.

9. Laboratory Record

Students should include the following in the laboratory record:

1. Aim
2. Problem Analysis
3. Network Topology Diagrams
4. Mininet Topology Implementation
5. Traffic Generation Commands
6. Wireshark Packet Capture
7. Sample Input and Output
8. Wireshark Screenshots
9. Test Cases and Results
10. Performance Observation Table
11. Packet-Level Analysis
12. Comparative Analysis of Topologies
13. Conclusion

Learning Outcome:

Students will analyze the behaviour of common network topologies by implementing them in Mininet, generating ICMP/TCP/UDP traffic, capturing packets using Wireshark, and interpreting packet-level characteristics such as addressing, protocol fields, sequence numbers, timing, retransmissions, and communication flow.