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">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>Academic Year & Batch</td><td colspan="5">2026-27 & 2024-28</td></tr>
<tr><td colspan="4">Degree and Branch</td><td colspan="6">B.E. Computer Science & Engineering</td></tr>
<tr><td colspan="4">Semester</td><td>V</td><td>Section</td><td>A, B & C</td><td>K4, 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">6</td><td>Assigned Date</td><td colspan="2">10-08-2026</td><td colspan="2">Due Date</td><td colspan="2">18-08-2026</td></tr>
</table>

Assignment: Simulation of UDP Packet Transmission Using SimPy

Application: Smart-city Surveillance System

A smart-city surveillance system uses **UDP communication** to transmit video packets from multiple roadside cameras to a central monitoring server. Since UDP does not provide reliable delivery, packets may be lost due to network congestion, and packets may experience random transmission delays. As a network engineer, develop a **SimPy-based simulation** to model packet generation, transmission, random delay, and packet loss, and analyze the performance of UDP communication under different network conditions.

1a. UDP Client Simulation using SimPy

Develop a **UDP Client module using SimPy** with the following functionalities:

1. Accept the number of packets, packet size, packet generation interval, and packet-loss probability.
2. Generate packets periodically using a SimPy process.
3. Assign a unique sequence number and generation timestamp to each packet.
4. Introduce a random transmission delay for each packet.
5. Based on the specified packet-loss probability, randomly determine whether a packet is transmitted or lost.
6. Send successfully generated packets to the simulated UDP Server through a SimPy communication channel/resource.
7. Maintain statistics for:
    - Packets generated
    - Packets transmitted
    - Packets lost

- Packet generation time
- Transmission delay

1b. UDP Server Simulation using SimPy

Develop a **UDP Server module using SimPy** with the following functionalities:

1. Receive packets from the simulated communication channel.
2. Record the arrival time of each successfully received packet.
3. Identify the packet sequence number and source information.
4. Calculate the delay experienced by each received packet using its generation and arrival timestamps.
5. Detect missing packets based on their sequence numbers.
6. Maintain statistics for:
   - Packets received
   - Packets lost
   - Packet delivery ratio
   - Packet loss ratio
   - Minimum delay
   - Maximum delay
   - Average packet delay
7. Display the received packet information and final simulation statistics.

1c. Simulation and Analysis

Run the client-server simulation for the following network conditions:

1. **Error-free transmission:** 0% packet loss and no random delay.
2. **Low packet loss:** 10% packet loss with 10–50 ms random delay.
3. **Moderate packet loss:** 20% packet loss with 20–100 ms random delay.
4. **High packet loss:** 30% packet loss with 50–150 ms random delay.
5. Vary the packet generation rate and analyze its effect on packet loss and delay.

For each scenario, calculate and compare:

- Number of packets generated
- Number of packets transmitted
- Number of packets received
- Number of packets lost
- Packet delivery ratio
- Packet loss ratio
- Average packet delay
- Minimum and maximum packet delay

1d. Sample Data for Laboratory Verification

Test Case	Packets	Packet Size	Generation Interval	Loss Probability	Random Delay
TC1	10	100 bytes	200 ms	0%	0 ms
TC2	20	100 bytes	200 ms	10%	10–50 ms

Test Case	Packets	Packet Size	Generation Interval	Loss Probability	Random Delay
TC3	30	200 bytes	100 ms	20%	20–100 ms
TC4	50	200 bytes	67 ms	30%	50–150 ms

The exact number of lost packets and delay values will vary because packet loss and delay are randomly generated during simulation.

Expected Client Output

```
UDP CLIENT SIMULATION

Packets to Generate : 20
Packet Size         : 100 bytes
Generation Interval : 200 ms
Loss Probability    : 10%
Delay Range         : 10-50 ms

Packet 1 Generated at 0.20 s
Packet 1 Transmitted

Packet 2 Generated at 0.40 s
Packet 2 Transmitted

Packet 3 Generated at 0.60 s
Packet 3 Lost

...

Total Packets Generated : 20
Total Packets Transmitted : 18
Total Packets Lost : 2
```

Expected Server Output

```
UDP SERVER SIMULATION

Packet 1 Received
Generation Time : 0.20 s
Arrival Time    : 0.24 s
Delay           : 0.04 s

Packet 2 Received
Generation Time : 0.40 s
Arrival Time    : 0.43 s
Delay           : 0.03 s

Packet 4 Received
Generation Time : 0.80 s
Arrival Time    : 0.85 s
Delay           : 0.05 s

...

Packets Received : 18
Packets Lost     : 2
Packet Loss Ratio : 10%
Packet Delivery Ratio : 90%
```

```
Average Delay : 0.04 s
```

Performance Observation

Test Case	Generated	Transmitted	Received	Lost	Loss Ratio	Avg. Delay
TC1	10	10	10	0	0%	0 ms
TC2	20	Varies	Varies	Varies	Varies	Varies
TC3	30	Varies	Varies	Varies	Varies	Varies
TC4	50	Varies	Varies	Varies	Varies	Varies

Note: Students should record the **aim, problem analysis, SimPy-based client algorithm, SimPy-based server algorithm, source code, sample input/output, screenshots, test cases, performance observations, analysis, and conclusion** in the laboratory record.

Expected Learning Outcome – K4

Students will **analyze UDP packet transmission** by modelling packet generation, random transmission delay, packet loss, and packet reception using **SimPy**, and interpret the effect of varying network conditions on packet delivery and delay.