CHRIST
UNIVERSITY
BENGALURU, INDIA

Declared as Deemed to be University under Section 3 of UGC Act 1956

DEPARTMENT OF ADSE

CSE434P-Computer Networks

PROJECT REPORT

B. Tech - Computer Science and Engineering
(AIML)

Submitted By:

Abel Alexander - 2462004

Annmarie Vinish – 2462041

Darain Brit A – 2462060

4BTCSAIML A

Abstract

Natural disasters frequently damage conventional communication infrastructure, disrupting coordination between victims and rescue teams. This project presents a simulation of an Emergency Communication Network using a drone-based mesh architecture. The system is implemented in two layers:

1. Cisco Packet Tracer simulation demonstrating dynamic routing and network redundancy using OSPF.
2. Python-based distributed simulation implementing priority-based message transmission using priority queues.

The system ensures self-healing communication, multi-hop forwarding, and prioritization of life-critical emergency messages. This project integrates networking principles and distributed systems concepts to model disaster-resilient communication.

Background

During disasters, communication systems such as cellular towers may fail. Drone-based networks provide temporary communication infrastructure where mobile nodes form a mesh network to relay emergency messages. This project simulates such a system at:

- Network Layer (Cisco Packet Tracer)
- Application Layer (Python Distributed System)

Problem Statement

The project addresses:

- Maintaining communication when infrastructure fails.
- Dynamically rerouting traffic if a node fails.
- Prioritizing emergency messages.
- Simulating distributed multi-hop communication.

Objectives

- Design a drone mesh topology.
- Implement OSPF dynamic routing.
- Add redundant backup links.
- Simulate priority-based message transmission.
- Demonstrate self-healing network behaviour.

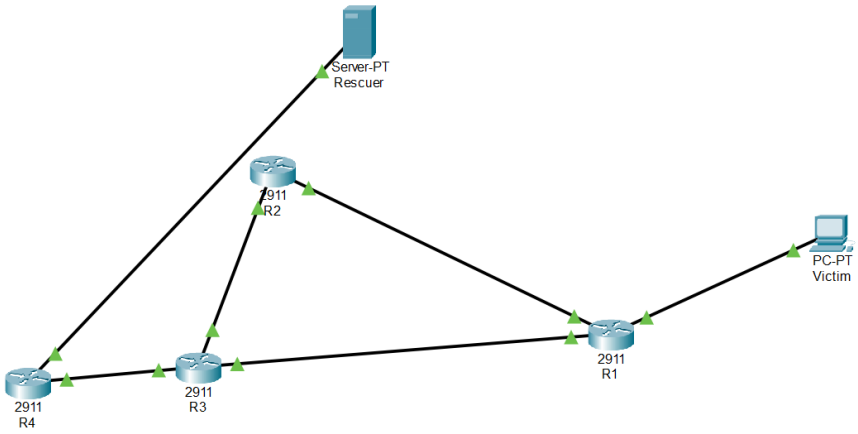
Cisco Packet Tracer Implementation

Network Topology

The drone nodes are represented by routers connected in a mesh topology with a backup link.

Topology:

PC → R1 → R2 → R3 → R4 → Server
Backup Link: R1 ↔ R3



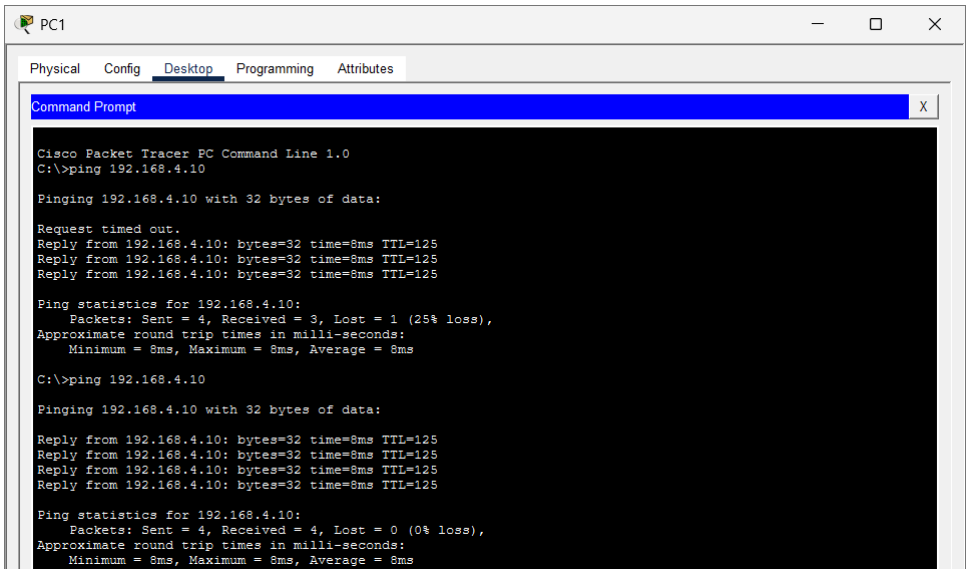
IP Addressing and OSPF Configuration

Each router interface is assigned a unique subnet. OSPF routing protocol is enabled on all routers to allow automatic path discovery and rerouting.

Dynamic Routing and Self-Healing

When Router R2 is shut down, OSPF recalculates the shortest path and reroutes traffic via the backup link.This demonstrates:

- Fault tolerance
- Redundancy
- Self-healing communication



Quality of Service (QoS)

QoS policies were configured using:

- Access Control Lists (ACL)
- Class Maps
- Policy Maps
- Service Policies

Traffic classes:

- HIGH – Medical emergency
- MEDIUM – Urgent assistance
- LOW – Default traffic

Due to limitations of Cisco Packet Tracer, full real-time queue scheduling under congestion cannot be visually demonstrated.

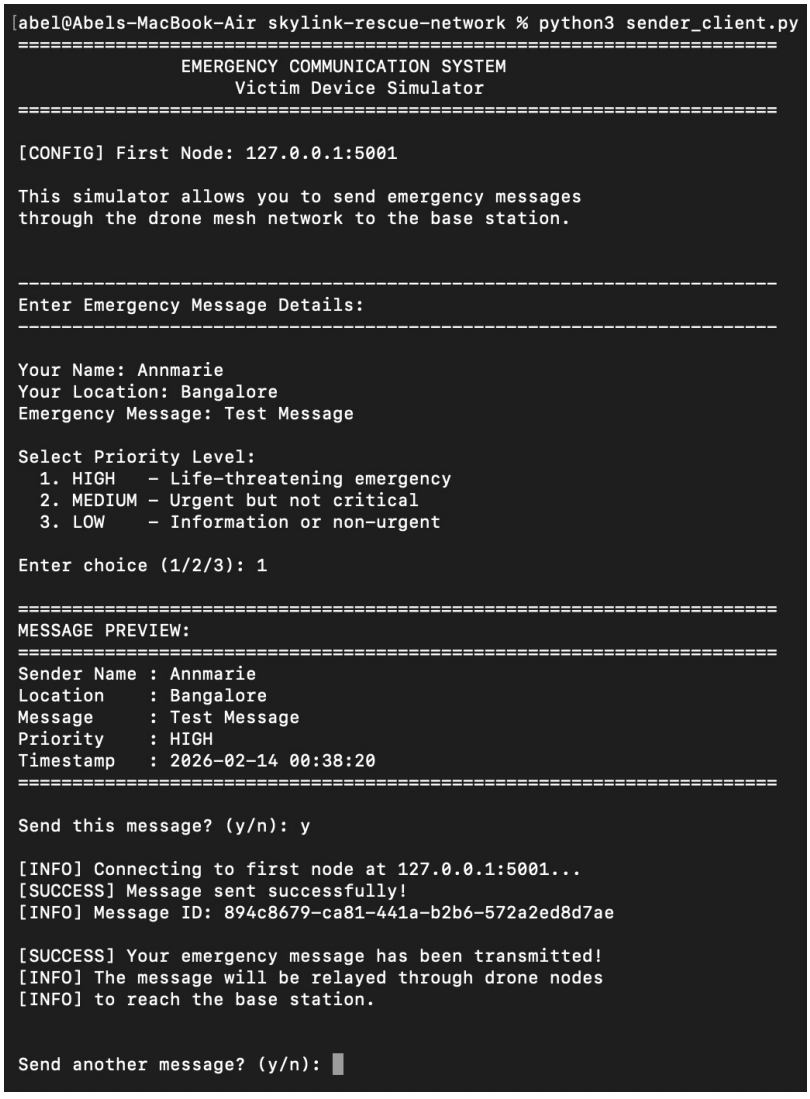
Python Distributed Mesh Simulation

System Architecture

The Python-based system consists of:

- sender_client.py – Victim device simulator
- node.py – Drone mesh node
- base_station.py – Rescue control center

Messages are forwarded dynamically between nodes until reaching the base station.



Priority Queue Implementation

Each drone node maintains a priority queue:

- HIGH → Priority value 1
- MEDIUM → Priority value 2
- LOW → Priority value 3

Messages are processed based on priority before forwarding.

```
[abel@Abels-MacBook-Air skylink-rescue-network % python3 node1.py
[00:37:54] [INFO] Message forwarding thread started

======================================================================
                    DRONE NODE - MESH NETWORK
======================================================================
[00:37:54] [INFO] Node listening on port 5001
[00:37:54] [INFO] Neighbors configured: 1
  Neighbor 1: 172.20.10.2:5002
======================================================================
[00:37:54] [SUCCESS] Node is ready to receive and forward messages!
======================================================================

[00:38:27] [INFO] Message received from 127.0.0.1:64466

======================================================================
NEW MESSAGE RECEIVED:
======================================================================
Message ID  : 894c8679-ca81-441a-b2b6-572a2ed8d7ae
Sender      : Annmarie
Location    : Bangalore
Message     : Test Message
Priority    : HIGH
Timestamp   : 2026-02-14 00:38:20
======================================================================

[00:38:27] [SUCCESS] Message queued with priority: HIGH (value: 1)
[00:38:27] [INFO] Queue size: 1
[00:38:27] [INFO] Processing message (Priority: HIGH, ID: 894c8679...)
[00:38:27] [INFO] Simulating transmission delay (1s)...
[00:38:28] [INFO] Attempting to forward to 172.20.10.2:5002...
[00:38:28] [SUCCESS] Forwarded to 172.20.10.2:5002
```

Dynamic Forwarding

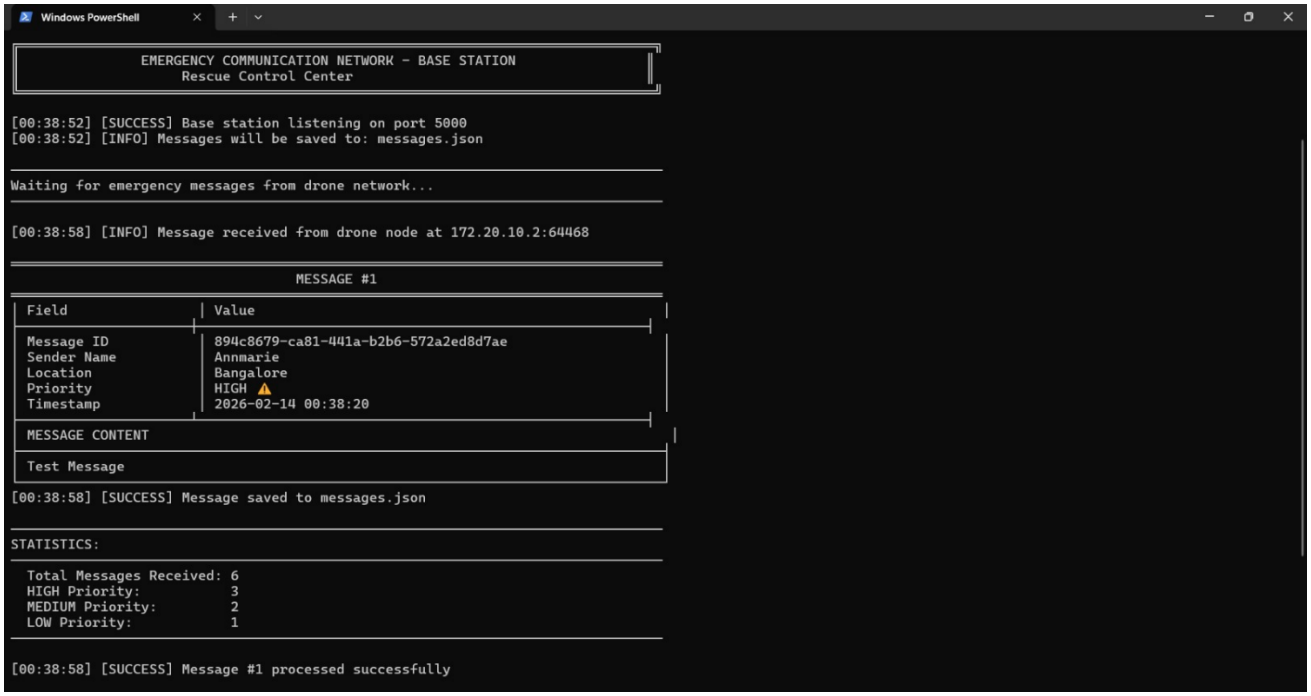
Each node maintains a neighbour list and attempts to forward messages to the next available node. If a node fails, alternate forwarding can occur.

```
[abel@Abels-MacBook-Air skylink-rescue-network % python3 node2.py
[00:37:53] [INFO] Message forwarding thread started

======================================================================
                    DRONE NODE - MESH NETWORK
======================================================================
[00:37:53] [INFO] Node listening on port 5002
[00:37:53] [INFO] Neighbors configured: 1
  Neighbor 1: 172.20.10.3:5000
======================================================================
[00:37:53] [SUCCESS] Node is ready to receive and forward messages!
======================================================================

[00:38:28] [INFO] Message received from 172.20.10.2:64467

======================================================================
NEW MESSAGE RECEIVED:
======================================================================
Message ID  : 894c8679-ca81-441a-b2b6-572a2ed8d7ae
Sender      : Annmarie
Location    : Bangalore
Message     : Test Message
Priority    : HIGH
Timestamp   : 2026-02-14 00:38:20
======================================================================

[00:38:28] [SUCCESS] Message queued with priority: HIGH (value: 1)
[00:38:28] [INFO] Queue size: 1
[00:38:28] [INFO] Processing message (Priority: HIGH, ID: 894c8679...)
[00:38:28] [INFO] Simulating transmission delay (1s)...
[00:38:29] [INFO] Attempting to forward to 172.20.10.3:5000...
[00:38:29] [SUCCESS] Forwarded to 172.20.10.3:5000
```

Base Station Processing

The base station:

- Receives messages
- Displays structured output
- Stores data in messages.json
- Tracks priority statistics



Results

The system successfully demonstrated:

- Multi-hop communication
- Dynamic routing with automatic rerouting
- Redundant backup path activation
- Priority-based message processing
- Successful end-to-end transmission

Cisco Packet Tracer validated routing and redundancy behavior.

Python simulation validated priority scheduling and distributed forwarding logic.

Conclusion

The project successfully models a drone-based emergency communication network at both networking and application layers.

Key achievements:

- Dynamic OSPF routing
- Redundant topology design
- Priority-based emergency transmission
- Self-healing network behavior
- Distributed system simulation

The integration of Cisco routing concepts and Python distributed systems creates a comprehensive disaster-resilient communication model.

Future Scope

- Real wireless mesh implementation
- Cloud-based monitoring dashboard
- AI-based route optimization
- Integration with IoT distress sensors
- SDN-based centralized control
- Real drone hardware deployment