THE NATIONAL INSTITUTE OF ENGINEERING

MYSURU

DEPARTMENT OF COMPUTER SCIENCE

AND ENGINEERING

SUBJECT: DISCRETE MATHEMATICAL STRUCTURES (BCS405A)

TOPIC: REPORT ON PRESENTATION ABOUT "APPLICATION OF GRAPH THEORY ON COMPUTER NETWORKS"

SUBMITTED BY:

NAMES	USN
1. PULKIT GARG	4NI23CS158
2. PURUSHOTHAM B J	4NI23CS159
3. RAHUL ANJAN H	4NI23CS161

SUBMMITED TO:

Mrs. R ANITHA

PROFESSOR AND HEAD

DEPARTMENT OF COMPUTER SCIENCE
AND ENGINEERING

NIE MYSURU

Contents

1. Introduction

Graph theory, a branch of discrete mathematics, studies structures used to model pairwise relations between objects. In its simplest form, a graph is a collection of **vertices (nodes)** and **edges (links)** that connect them.

In the context of computer networks, graph theory is a powerful tool to model and analyse the connectivity, routing, topology, and resilience of the network. Each device (e.g., routers, switches, computers) is represented as a vertex, and the communication links between them are represented as edges. This abstraction allows network engineers to apply mathematical principles to optimize and troubleshoot complex networking environments.

2. Why Use Graph Theory in Networking?

Graph theory offers various analytical and computational tools that simplify complex network analysis:

- **Analysing Connectivity**: Determines whether all nodes in a network can reach each other. This ensures no isolated segment exists in the network.
- **Shortest Path Routing**: Algorithms such as Dijkstra's or Bellman-Ford are used to compute the shortest (or least-cost) path from source to destination, improving data packet transmission efficiency.
- **Bottleneck Detection**: Nodes with high connectivity (high degree) may become bottlenecks. Identifying them helps mitigate performance degradation.

- **Simulation of Network Failures**: Network reliability can be tested by simulating the removal of critical nodes or links and observing the resulting connectivity.
- **Resource Optimization**: Helps in bandwidth management, traffic distribution, and routing cost minimization.

Example: Internet backbone providers use graph theory to simulate node/link failures and determine alternate paths to maintain uninterrupted service.

3. Modelling Network Topologies Using Graphs

3.1 Nodes and Links

- **Vertices (Nodes)** represent devices:
  - **Routers**: Forward packets across networks.
  - **Switches**: Connect devices in a LAN.
  - **End devices**: PCs, smartphones, IoT devices, servers.
- **Edges (Links)** represent communication channels:
  - **Wired**: Ethernet cables, fibre optics.
  - **Wireless**: Wi-Fi, Bluetooth, cellular.

3.2 Network Topology Types

Each network topology can be visualized using a graph:

- **Bus**: A single communication line (edge) connects multiple nodes.
- **Ring**: Nodes are connected in a circular pattern.
- **Star**: A central node connects to all others.
- **Mesh**: Every node is connected to every other node (full mesh) or selectively (partial mesh).

4. Vertex Degree and Network Analysis

Vertex Degree

In graph theory, the **degree** of a vertex is the number of edges incident to it. In networking:

- A **high-degree node** (e.g., a router with many links) handles more traffic and has higher importance.
- A **low-degree node** may be an endpoint with limited connectivity.

Applications in Networking

- **Identifying Central Hubs**: Nodes with a high degree serve as traffic aggregators and require load balancing and redundancy.
- **Load Distribution**: Knowing vertex degrees helps in redistributing traffic to avoid congestion.
- **Fault Analysis**: High-degree nodes are more likely to be single points of failure.

Example: A router connecting multiple subnets may be monitored for traffic spikes and failures.

5. Key Applications in Networking

5.1 Identifying Network Hubs

- High-degree nodes often serve as **gateways** or **central routers**.
- Their failure could disrupt connectivity to multiple nodes.
- Used in backbone networks, data centres, and ISPs.

5.2 Load Balancing

- Graphs help in evaluating alternative paths based on current node loads.
- Ensures fair distribution of network traffic.
- Essential in real-time applications and cloud computing.

5.3 Redundancy and Fault Tolerance

- Adds extra edges to create alternate paths.
- Prevents disconnection during node/link failures.
- Key in critical systems such as banking networks, air traffic control, and emergency response networks.

6. Graph Isomorphism in Computer Networks

6.1 Definition

Two graphs are **isomorphic** if there's a one-to-one correspondence between their vertices and edges, maintaining the connectivity structure.

6.2 Applications

a. Network Equivalence Verification

Used to verify whether two networks—possibly with different layouts—function the same.

Example: Before migrating to a new network infrastructure, engineers verify if the routing and structure remain consistent.

b. Simulating Topological Alternatives

Engineers can evaluate different physical layouts (e.g., ring vs. star) that offer the same logical functionality.

Example: Choosing a cost-effective yet resilient layout for a data center network.

c. Security and Forensics

Used to detect disguised malicious network patterns (e.g., botnets or P2P networks).

Example: Detecting a botnet by matching its traffic graph to known threat graphs, even if its node labels are obfuscated.

7. Euler Trails and Circuits in Network Applications

7.1 Euler Trail

A path that visits every **edge exactly once** but may start and end at different nodes.

- **Condition**: Graph must have exactly **0 or 2 odd-degree vertices**.

7.2 Euler Circuit

A special trail that starts and ends at the **same node**.

- **Condition**: All vertices have **even degree**, and the graph is connected.

8. Practical Applications of Euler Trails and Circuits

8.1 Network Maintenance and Monitoring

- Used to inspect all connections without repetition.
- Ideal for scheduling automated inspection bots in data centres.

Example: A maintenance bot inspects all fibre links in a network using an Euler trail.

8.2 Diagnostic Routing and Testing

- Loopback testing or routing diagnostics ensure all paths are checked.

Example: Cisco switches use loopback tests to diagnose internal paths, modelled via Euler circuits.

8.3 Cable Laying Optimization

- Design network cabling routes that minimize redundancy and cost.

Example: Laying fibre optics in a new office building with minimal overlap.

9. Case Studies and Examples

Data Center Routers

- A router with many direct connections has a high degree.
- Handles significant traffic and needs constant monitoring and redundancy.

Star vs. Ring Topology Simulation

- Isomorphic graphs with different physical layouts.
- Ring offers better fault tolerance; star is cheaper and easier to manage.

Botnet Detection

- A malicious network may alter node labels to evade detection.
- Graph isomorphism reveals structural similarity to known threat patterns.

10. Conclusion

Graph theory provides a rigorous and versatile framework for modelling, analysing, and optimizing computer networks. From topology design to routing, fault tolerance, and cybersecurity, graph-theoretic models guide engineers in building robust, efficient, and secure networks.

As networking evolves with software-defined networks (SDNs), Internet of Things (IoT), and 5G, the role of graph theory will only grow, enabling automation, dynamic reconfiguration, and intelligent decision-making.

11. References

1. **West, D. B.** *Introduction to Graph Theory*. Pearson Education, 2001.
2. **Kurose, J. F., & Ross, K. W.** *Computer Networking: A Top-Down Approach*. Pearson, 2021.
3. **Newman, M. E. J.** *Networks: An Introduction*. Oxford University Press, 2010.
4. IEEE Xplore: Articles on "Graph Theory in Network Routing and Security" — https://ieeexplore.ieee.org
5. ACM Digital Library: Search for "Applications of Graph Theory in Network Design" — https://dl.acm.org