NETWORK TOPOLOGY CHARACTERSTICS

ITCS 6114 - Course Project

Venkataraman Hegde
Abhay Marathe
Anushree Srivastava
Chandni Jivani
Shairal Neema

Table of Contents

1. Introduction:

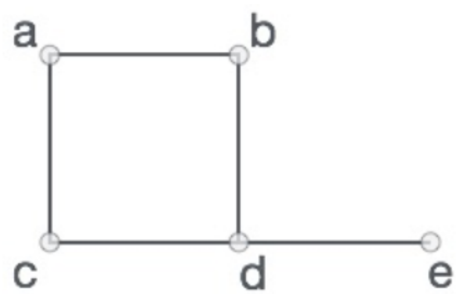
1) Graphs Data Structure

a) Definition

A Graph consists of a finite set of vertices(or nodes) and set of Edges which connect a pair of nodes. Generally, a graph **G** is represented as **G = (V , E)**, where **V is set of vertices** and **E is set of edges**.

b) Formal Definition

Formally, a graph is a pair of sets **(V, E)**, where **V** is the set of vertices and **E** is the set of edges, connecting the pairs of vertices. Take a look at the following graph



In the above graph,

V = {a, b, c, d, e}

E = {ab, ac, bd, cd, de}

c) Basic Operations

- Additions
  - addNode: adds vertices to your graph
  - addEdge: creates edges between two given vertices in your graph
- Removals
  - removeNode: removes vertices from your graph
  - removeEdge: removes edges between two given vertices in your graph
- Search
  - contains: checks if your graph contains a given value
  - hasEdge: checks if a connection exists between two given nodes in your graph.

2) Application of Graphs Data Structure

Some examples of graphs as data structures are:

- Airline Traffic (image above)
  - Node/vertex = Airport
  - Edges = direct flights between two airports

    - Weight = miles between two airports
- GPS Navigation
    - Node = road intersection
    - Edge = road
    - Weight = time required to go from one intersection to another
- Networks routing
    - Node = server
    - Edge = data link
    - Weight = connection speed

To name a few more real world applications of graphs we have,

- Electronic circuits
- Flight reservations
- Driving directions
- Telcom: Cell tower frequency planning
- Social networks. E.g., Facebook uses a graph for suggesting friends
- Recommendations: Amazon/Netflix uses graphs to make suggestions for products/movies
- Graphs help to plan the logistics of delivering goods

3) Why Graphs?

Here are a few points that help us to understand why we should use graphs in our day-to-day problems –

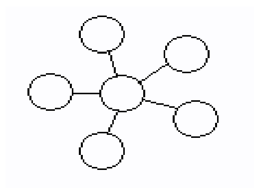
1. Graphs provide a better way of dealing with abstract concepts like relationships and interactions. They also offer an intuitively visual way of thinking about these concepts. Graphs also form a natural basis for analyzing relationships in a Social context
2. Graph Databases have become common computational tools and alternatives to SQL and NoSQL databases
3. Graphs are used to model analytics workflows in the form of DAGs (Directed acyclic graphs)
4. Some Neural Network Frameworks also use DAGs to model the various operations in different layers
5. Graph Theory concepts are used to study and model Social Networks, Fraud patterns, Power consumption patterns, Virality and Influence in Social Media. Social Network Analysis (SNA) is probably the best known application of graphs.
6. It is used in Clustering algorithms.
7. System Dynamics also uses some Graph Theory concepts.
8. Path Optimization is a subset of the Optimization problem that also uses Graph concepts

9. From a Computer Science perspective – Graphs offer computational efficiency. The Big O complexity for some algorithms is better for data arranged in the form of Graphs (compared to tabular data).

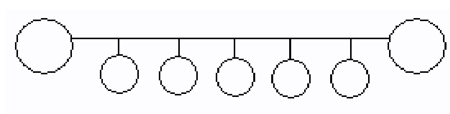
2. Basic Terminology of Network Topology

Network Topology is the topological structure used to depict the physical and logical structure. It is implemented using the graph theories where the nodes depict the nodes in the topology and the edges represent the connection between these nodes. The network topologies can be primarily classified as:

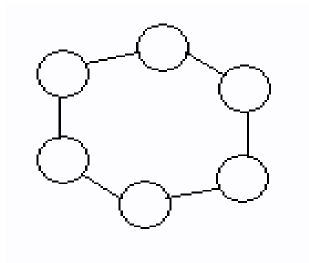
1. **Star Topology :** In this topology all the network devices are connected to one central computer. The information is passed via one central hub.



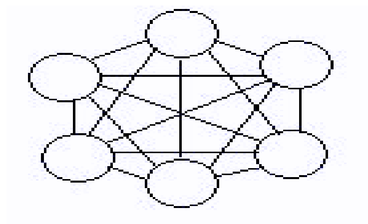
2. **Bus Topology**: In this network topology there is one central line and all the devices are connected to this main line.



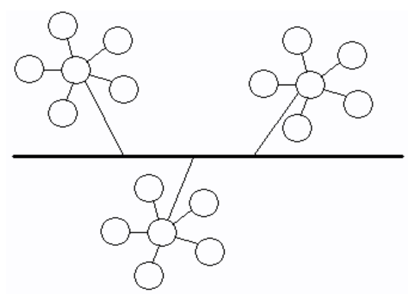
3. **Ring Topology:** In this topology all the nodes are connected in a closed loop and all the messages pass in a closed loop.



4. **Mesh Topology:** In this topology the circuit is designed such that all the devices are connected to all the other devices in the network. It provides the highest level of redundancy and robust against network failure



5. **Hybrid Topology:** This topology is also referred to as the Tree topology as it is combination of star and bus topology. The star network is connected with a central bus line.



Logical Topology: This term refers to the concept of networking which defines how the communication the communication happens in the network. Using switches and routers the logical aspects of the network can be implemented. The physical topology as opposed to this implements the physical connections in the network.

The network connection as also be classified as the following metrics: **Distance Class, Connection and Spectra class.**

Distance Class: The path are the entities that depict the connection between two separate nodes. Suppose we have two nodes A and B. The path from A -> B , PA -> PB can be represented as PA -> P1 -> P2 -> ….Pn+1 where Pn+1=PB. The metrics of Distance class are:

1. **Hop Count**: The number on nodes in between to specific nodes in the Hop count of the network. The shortest hop count of the network is the minimum number of nodes between two specific nodes.
2. **Closeness**: It is the average hop count between two specific nodes by the total number of nodes.
3. **Eccentricity**: It is the maximum distance between a specific node and any other node in the network.
4. **Diameter**: It the maximum eccentricity of a node over all the nodes in the network.
5. **Radius**: It the minimum eccentricity of a node over all the nodes in the network.
6. **Persistence**: It is the minimum number of links to be removed from the graph to make it disconnected.
7. **Girth**: It is the shortest hop count of closed loop in the graph.
8. **Expanion**: It is the average fraction of nodes in the graph that fall within a ball of radius h (in hops) centered at a random node in the topology.
9. **Betweenness**: Betweenness of a node (or a link) Bk is defined as the number of shortest paths between pairs of nodes that traverse a node or link k.
10. **Central Point of Dominance:** It the largest value of betweenness centrality if the network.
11. **Distortion**: It defines the number of hops required to go from one link of the network to another.

Connection Class: It represents the cohesive subgroups within a network.

1. **Degree**: The node degree distribution is the probability that the degree D of a randomly selected node equals k,

$$Pr[D = k] = dk$$

2. **Joint Degree Distribution**: The joint degree distribution (JDD) Pr[D1 = k1, D1 = k2] is the probability that a random pair of nodes possess a degree equal to k1 and k2, respectively.
3. **Assortativity**: It is a way to determine the correlation among the degrees is by considering the Pearson correlation coefficient of the degrees at either ends of each link.
4. **Coreness**: The node coreness ki of a given node ni is the maximum k such that this node is present in the k − core graph, but removed from the (k + 1) − core.
5. **Cliques and n-cliques**: A clique of a given graph G(N, L) is a subset of nodes such that all elements in the clique S(NS, LS), where NS ≤ N, LS ≤ L, are fully connected, hence, forming a full mesh.
6. **Clustering coefficient**: The local clustering coefficient of a node ni in a graph G measures the cliquishness of ni neighborhood

$$ci = yi / (di/2)$$

where yi is the number of links between neighbors of ni, and di is the degree of the node ni.

7. **Rich Club coefficient**: It the ratio of the number of links connecting the club members over the maximum number of allowable links in Sk, which measures how well the rich nodes know each other.
8. **Giant component**: A strongly connected component is a maximal subgraph GC of a directed graph such that for every pair of nodes nA, nB ∈ GC ,

there exists a directed path $P_{A\to B}(k)$ and a directed path $P_{B\to A}(k)$, for any hop k.

9. **Reliability**: Reliability metrics measure the number of removed elements that lead to disconnected components in a graph.
10. **Chromatic number**: It represents the adjacency of two nodes.

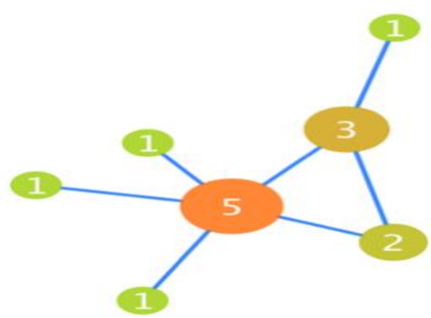
Spectra Class: The scalars and vectors are called eigenvalues and eigenvectors of A, respectively. The set of distinct eigenvalues λ is called the spectrum of A. Many network properties such as the vertex connectivity are related to the spectrum of a graph

3. **Network topology parameters**

1. Degree of a node and its distribution: The degree of a node is the number of edges connected to a graph. The degree of a node and the connectivity of the node are directly proportional to each other. It means that more is the degree of a node higher is its interconnectivity. The probability of Degree Distribution can be represented as:

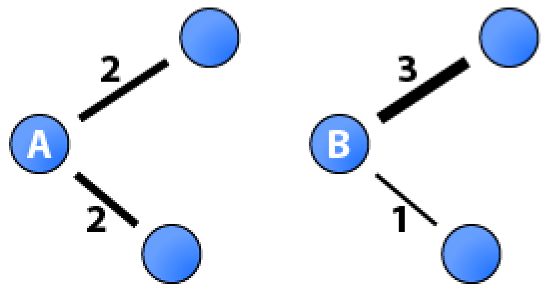
$$p(k)=N(k)/\sum N(k)$$

Here N(k) represents the nodes with degree k.



2. Strength of a Node and its distribution: Node Strength is the sum of the weights of the links connected to the node. If it is a directed graph there are mainly two types of strength:

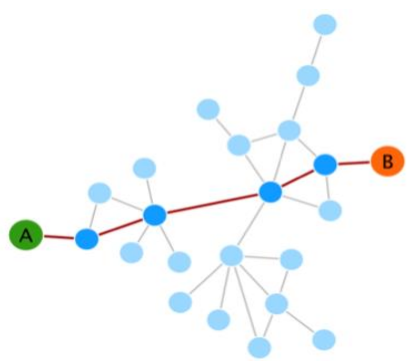
- Outward Strength : Outward links weights
- Inward Strength : Inward links weights



$S_i = \sum a_{ij} w_{ij}$

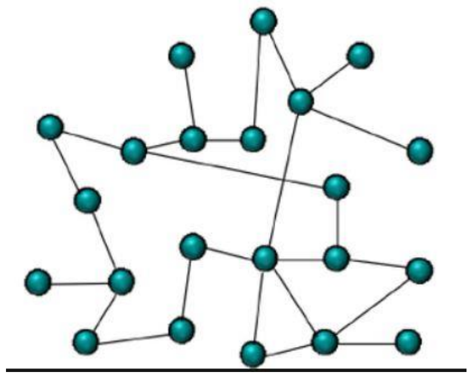
3. Characteristic Path Length: It represents the shortest distance between any two nodes. It is used to understand how the information flows in an application.

$L = \sum_i \sum_j L_{ij} / N(N-1)$, L_{ij} is the shortest path between i and jth node



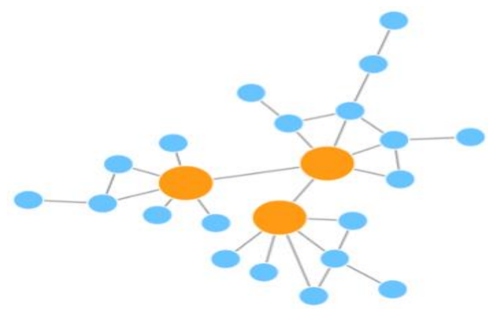
4. Clustering coefficient of a network: The clustering coefficient (C) is a measure of local cohesiveness. Traditionally the clustering coefficient C* of a node i is the ratio between the total number (ei)of the edges actually connecting its nearest neighbors to the t4* node and the total number of all possible edges between all these nearest neighbors

5. Random os Scale Free network : In random networks any node has the same probability to be connected with any other node of the network. On the other hand, in scale-free networks , some nodes act as “highly connected hubs” (high degree), although most nodes are of low degree.



Random Network

Scale Free Network



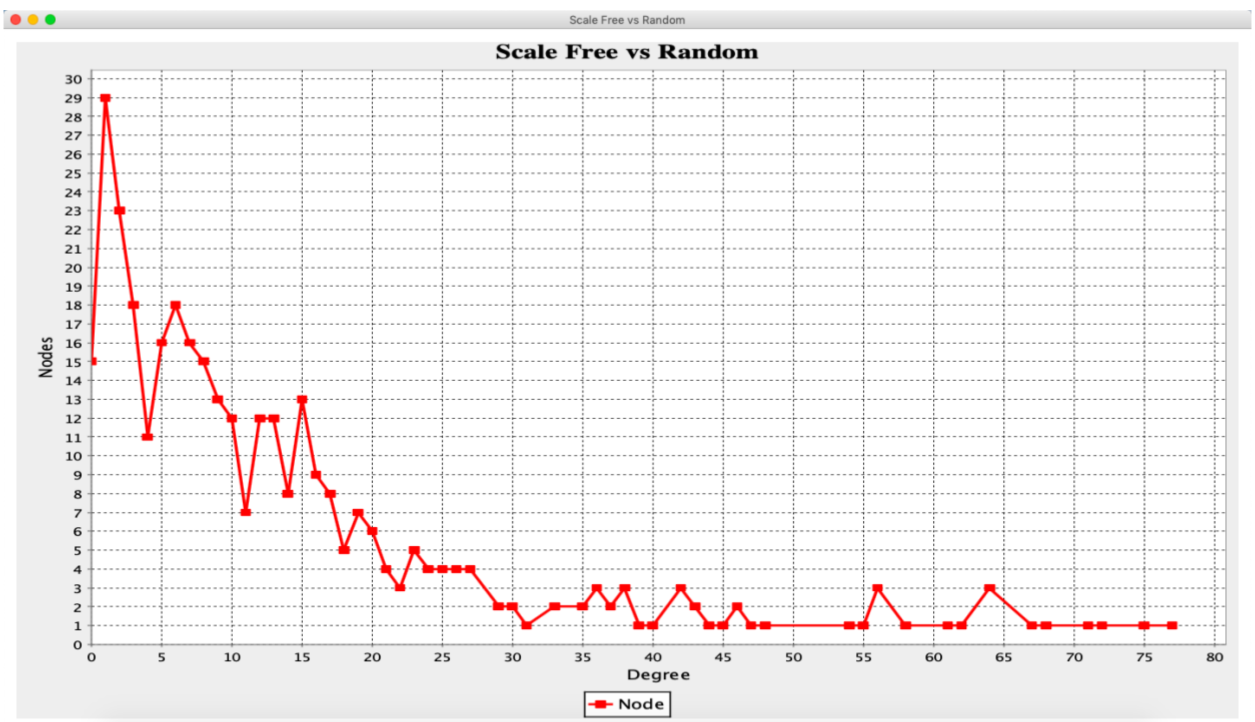
4. Output Screenshots::

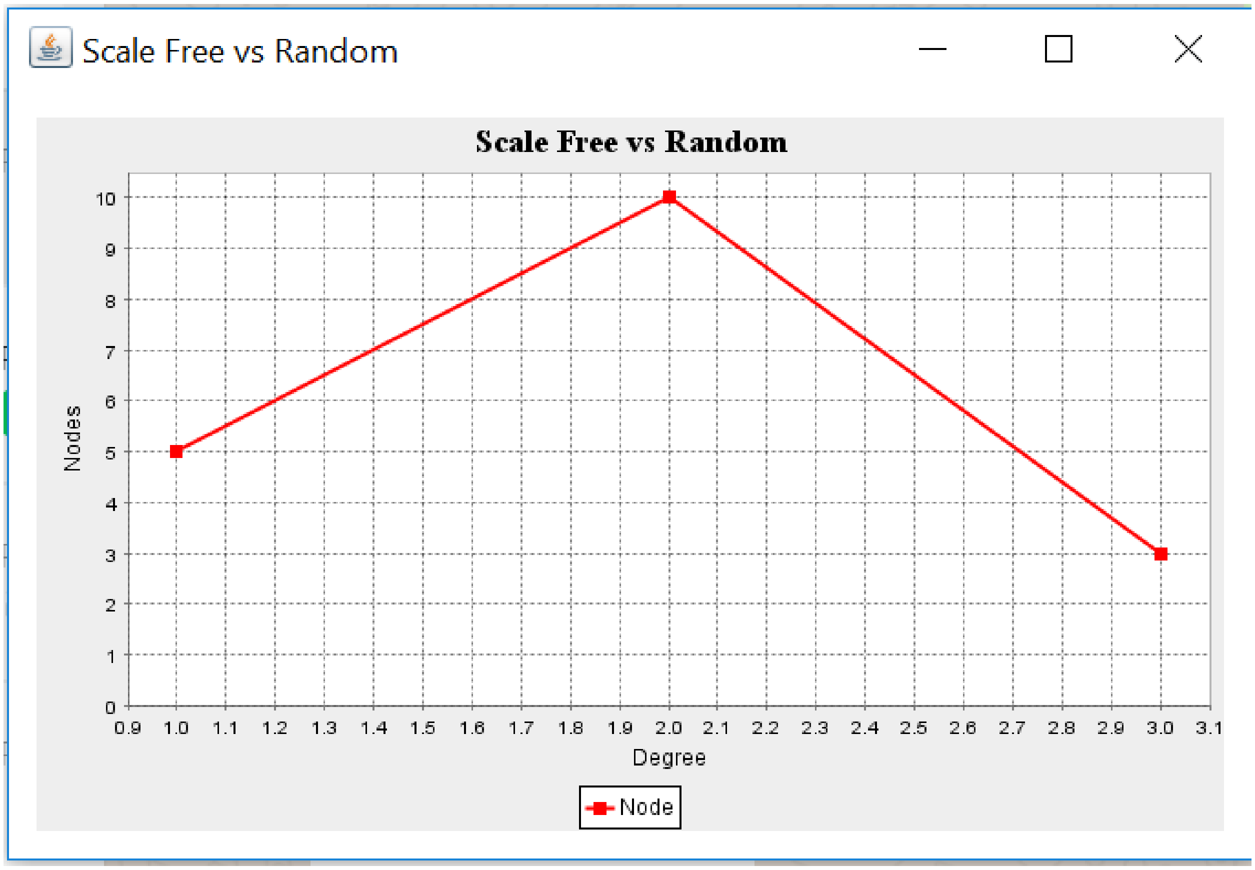
```
Adjacency list representation of graph 1
1 -> [[5|(3.0)], [2|(1.0)]]
2 -> [[1|(1.0)], [3|(3.0)], [5|(1.0)]]
3 -> [[2|(3.0)], [4|(1.0)]]
4 -> [[3|(1.0)], [5|(1.0)], [6|(2.0)]]
5 -> [[1|(3.0)], [2|(1.0)], [4|(1.0)]]
6 -> [[4|(2.0)]]
```

```
Property 1 – Degree of nodes and Distribution
Degree of Nodes
node    degree
-------------------
1       2
2       3
3       2
4       3
5       3
6       1
Degree Distribution of Nodes
Degree  Degree Dist
-------------------
1       0.17
2       0.33
3       0.5
Average Degree of this Network – 2.33
======================================================
```

```
Property 2 – Strength of nodes and Distribution
Strength of Nodes
node    strength
-------------------
1       4.0
2       5.0
3       4.0
4       4.0
5       5.0
6       2.0
Strength Distribution of Nodes
Strength    Strength Dist
-------------------
2.0     0.17
4.0     0.5
5.0     0.33
Average Degree of this Network – 4.0
```

```
Property 3 – Clustering Coefficient of Network
Clustering Coefficient for Nodes
node    Clustering Coefficient
------------------------------------
1       1.0
2       0.3333333333333333
3       0.0
4       0.0
5       0.3333333333333333
6       0.0
Average Degree of this Network – 0.28
======================================================
Property 4 – Characteristic path length
Characteristic Path Length of this Graph is – 1.23
```





5. References:

1. https://www.analyticsvidhya.com/blog/2018/04/introduction-to-graph-theory-network-analysis-python-codes/
2. https://www.britannica.com/topic/graph-theory
3. https://www.geeksforgeeks.org/applications-of-graph-data-structure/
4. https://towardsdatascience.com/graph-theory-history-overview-f89a3efc0478
5. https://medium.freecodecamp.org/a-gentle-introduction-to-data-structures-how-graphs-work-a223d9ef8837
6. https://adrianmejia.com/blog/2018/05/14/data-structures-for-beginners-graphs-time-complexity-tutorial/
7. Professors notes
8. Main Research Paper- Topological parameters of a network and relevant statistical tests