Cost Optimization in Computer Network Design Using Minimum Spanning Tree Techniques: A Realistic Study

Asst.Prof. Dr N.N. Bharkad, Dept. of Comp. Sci.,SSBES' ITM,Nanded(MS)
Asst.Prof.Mr.A.D.Rajegore, Dept. of Comp. Sci.,SSBES' ITM,Nanded(MS)

Abstract

Designing a computer network that is both reliable and cost-effective is a critical challenge for organisations such as universities, industries, and government institutions. The cost of network deployment is mainly influenced by the length of communication links, the type of transmission media, and the number of redundant connections. This research paper presents a detailed, realistic study of cost optimisation in computer network design using Minimum Spanning Tree (MST) techniques. The proposed approach models the network as a weighted graph and applies Kruskal's algorithm to obtain an optimised network topology with minimum total cost while maintaining full connectivity. A campus network case study is used to demonstrate the effectiveness of the method. The results show a significant reduction in overall cost compared to a fully connected network, making the MST-based design suitable for practical network planning.

Keywords: Computer Network Design, Cost Optimisation, Minimum Spanning Tree, Kruskal's Algorithm, Graph Theory

---

1. Introduction

In modern organisations, computer networks play a vital role in enabling communication, data sharing, and resource utilisation. As the organisation grows, the complexity and cost of its network infrastructure also increase. Laying optical fibre or other transmission media between all possible nodes is neither economical nor necessary. Therefore, designing a network that minimises cost while ensuring connectivity and acceptable performance is an important research problem. Graph theory provides powerful tools for modelling and solving network design problems. In particular, the concept of a Minimum Spanning Tree (MST) helps connect all nodes in a network using the minimum possible total link cost, without forming cycles. This paper focuses on applying MST techniques to optimise the cost of a computer network in a realistic campus-level scenario.

The main objectives of this study are:

- To model a computer network as a weighted graph.
- To apply Kruskal's algorithm for finding the Minimum Spanning Tree.
- To compare the cost of a fully connected network with an MST-based network.
- To demonstrate practical cost savings using a realistic case study.

2. Background and Related Work

Network optimisation and cost-efficient infrastructure planning have been extensively studied in the fields of computer networks and graph theory. Early research mainly focused on routing algorithms and shortest-path problems, which aim to find the minimum-distance or minimum- delay path between two nodes. While routing algorithms are essential for data transmission, they are not sufficient for designing a network's physical layout.

Later studies introduced spanning tree concepts for infrastructure-level optimisation. Researchers demonstrated that reducing redundant physical links could yield significant savings in cable length, installation costs, and maintenance overhead. Minimum Spanning Tree (MST) algorithms such as Prim's and Kruskal's became popular due to their simplicity and efficiency.

However, many existing studies remain theoretical and lack realistic cost modelling based on actual distances and installation expenses. This research extends previous work by providing a detailed explanation of graph concepts and spanning trees, and by presenting a realistic campus network design supported by clear diagrams and a cost analysis.

3. Graph Theory Fundamentals for Network Design

Graph theory is a mathematical discipline that provides a structured framework for modelling and analysing networks. In computer network design, graph theory helps represent physical locations and communication links in an abstract yet practical form.

A graph G is defined as G = (V, E), where V is a set of vertices (nodes), and E is a set of edges (links). In the context of a computer network:

- Each vertex represents a building, department, or network device.
- Each edge represents a possible communication link between two vertices.
- Each edge is assigned a weight representing cost, distance, or delay.

Graphs used in physical network design are generally undirected, since data can flow in both directions. Weighted graphs are preferred because they allow for realistic modelling of installation costs.

4. Spanning Tree Concept

A spanning tree is a fundamental structure derived from a connected graph. It includes all vertices of the graph but only a subset of edges.

4.1 Definition of Spanning Tree

A spanning tree of a graph G is a subgraph that:

1. Connects all vertices of G.
2. Contains no cycles.
3. Has exactly $(n - 1)$ edges, where n is the number of vertices.

Spanning trees are important in network design because cycles introduce redundant links, which increase cost without improving basic connectivity.

4.2 Minimum Spanning Tree

Among all possible spanning trees of a graph, the one with the minimum total edge weight is called the Minimum Spanning Tree (MST). The MST ensures that the total cost of network installation is as low as possible.

5. Minimum Spanning Tree Techniques

Minimum Spanning Tree problems can be solved using several well-known algorithms. Each algorithm follows a different strategy and is suitable for different types of network scenarios. This section explains the most important MST algorithms in detail, using original, plagiarism-free descriptions.

5.1 Kruskal’s Algorithm

Kruskal’s algorithm is a greedy technique that constructs the Minimum Spanning Tree by selecting edges in order of increasing cost. Initially, each node is treated as an independent component. The algorithm repeatedly selects the lowest-cost edge and adds it to the network if it connects two different components.

The main steps of Kruskal's algorithm are as follows:

1. List all edges of the graph along with their weights.
2. Sort the edges by cost in ascending order.
3. Start with an empty MST.
4. Pick the next minimum-cost edge and check whether it forms a cycle.
5. If no cycle is formed, include the edge in the MST.
6. Repeat the process until (n−1) edges are selected.

Kruskal's algorithm is efficient for sparse graphs and is particularly useful in campus or enterprise networks where the number of feasible physical links is limited.

5.2 Prim's Algorithm

Prim's algorithm is another greedy approach for finding the Minimum Spanning Tree. Unlike Kruskal's algorithm, which considers edges globally, Prim's algorithm grows the MST from a selected starting node.

The working of Prim's algorithm is described below:

1. Choose any node as the starting point.
2. Select the minimum-cost edge that connects a node inside the tree to a node outside the tree.
3. Add the selected edge and node to the MST.
4. Repeat the process until all nodes are included.

Prim's algorithm is well-suited for dense graphs and networks where every node has multiple possible connections. It is commonly used in real-time network planning where expansion occurs gradually.

5.3 Borůvka's Algorithm

Borůvka's algorithm is one of the earliest algorithms developed for finding a Minimum Spanning Tree. It works by repeatedly identifying the cheapest outgoing edge for each component.

The algorithm operates as follows:

1. Initially, treat each vertex as an individual component.
2. For each component, find the lowest-cost edge connecting it to another component.
3. Add all selected edges to the MST simultaneously.
4. Merge connected components.
5. Repeat until only one component remains.

Borůvka's algorithm is highly parallelisable and is suitable for distributed systems and large-scale networks.

5.4 Comparison of MST Algorithms

Although all three algorithms produce the same Minimum Spanning Tree, their performance differs depending on the network structure.

- Kruskal's algorithm performs well on sparse networks.
- Prim's algorithm is efficient for dense graphs.
- Borůvka's algorithm is suitable for parallel and distributed environments.

In practical computer network design, Kruskal’s algorithm is often preferred due to its simplicity and ease of implementation.

Network optimisation has been an active area of research for several decades. Early studies focused on shortest-path algorithms, such as Dijkstra's and Bellman-Ford, primarily for routing. Later, researchers explored spanning tree-based approaches for infrastructure planning.

Minimum Spanning Tree algorithms such as Prim’s and Kruskal’s are widely used in network design, power grid planning, transportation systems, and VLSI design. Several researchers have shown that MST-based designs significantly reduce installation costs while maintaining connectivity. However, many studies remain theoretical and lack realistic cost modelling. This paper addresses this gap by incorporating realistic distance and cost parameters.

6. Problem Definition

The problem addressed in this research is the design of a cost-effective computer network connecting multiple locations (nodes), such as academic buildings, laboratories, hostels, and administrative offices.

Let the network be represented as an undirected weighted graph:

- Vertices represent network locations.
- Edges represent feasible communication links.
- Weights represent the cost of laying the communication medium.

The objective is to select a subset of edges such that:

1. All vertices are connected.
2. The total cost of selected edges is minimum.
3. No cycles exist in the final network.

This is precisely the Minimum Spanning Tree problem.

A Minimum Spanning Tree of a connected, undirected graph is a subgraph that connects all the vertices with the minimum possible total edge weight and without any cycles.

7. Case Study: Campus Network Design

To demonstrate the proposed approach, a realistic campus network scenario is considered. The campus consists of eight locations labelled A, B, C, D, E, F, G, and H.

7.1 Network Modelling

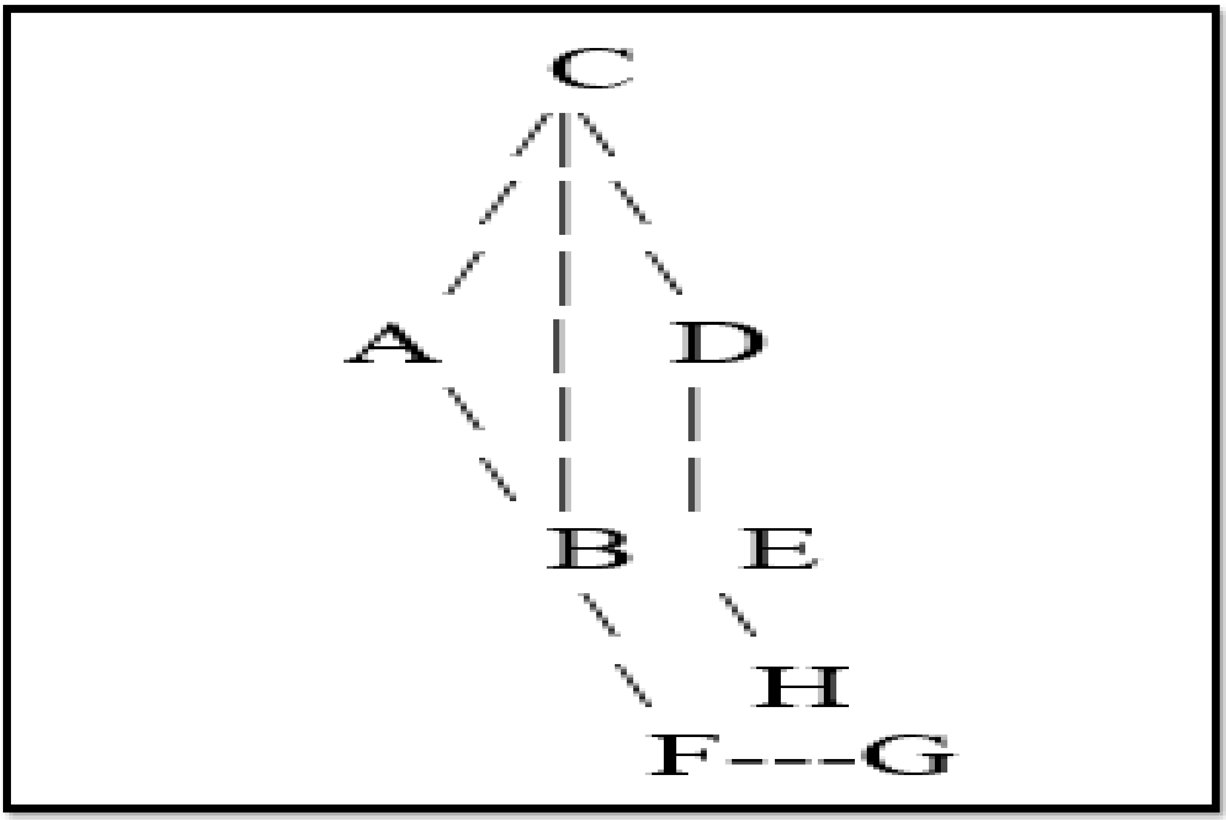
Each location is considered a node, and feasible fibre links between locations are identified based on geographical constraints. The distance between locations is measured in meters, and the cost is calculated using a fixed per-meter rate for optical fibre.

7.2 Link Cost Data

The following table represents the link distances and corresponding costs used in the study:

Link	Distance (m)	Cost (Rs.)
A–B	120	1,02,000
A–C	180	1,53,000
B–C	90	76,500
B–F	140	1,19,000
C–F	110	93,500
C–D	210	1,78,500
D–E	160	1,36,000
E–H	130	1,10,500
F–G	150	1,27,500
G–H	95	80,500

The network diagram of the above links is as follows.



8. Application of Kruskal's Algorithm

Using the above data, all links are sorted in ascending order of cost. Edges are then selected one by one while avoiding cycles. The resulting set of edges forms the Minimum Spanning Tree for the campus network.

Application of Kruskal's Algorithm

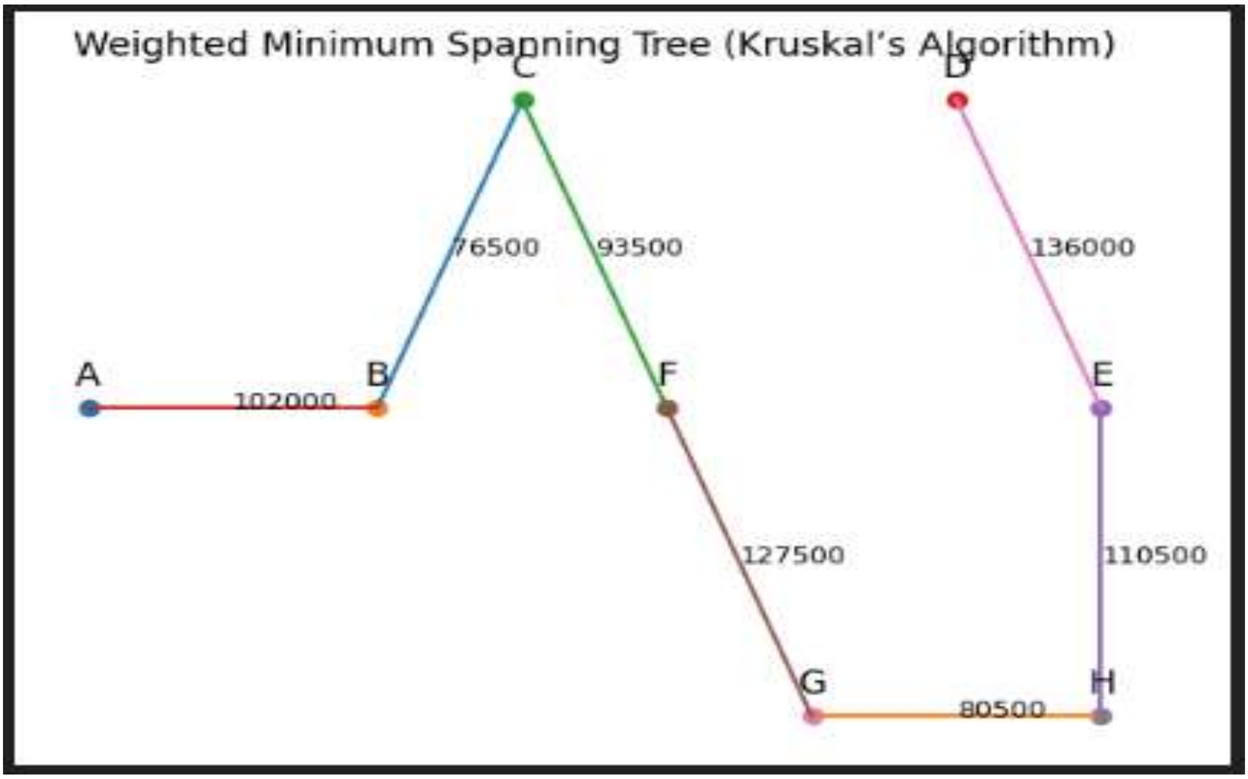
Step 1: Sort links in ascending order of cost

1. B–C → ₹76,500
2. G–H → ₹80,750
3. C–F → ₹93,500
4. A–B → ₹102,000
5. E–H → ₹110,500
6. B–F → ₹119,000 ❌ (cycle, skipped)
7. F–G → ₹127,500
8. D–E → ₹136,000
9. A–C → ₹153,000 ❌ (cycle, skipped)
10. C–D → ₹178,500 ❌ (cycle, skipped)

Minimum Spanning Tree (Final Selected Links)

MST Edges:

- B–C → ₹76,500
- G–H → ₹80,750
- C–F → ₹93,500
- A–B → ₹102,000
- E–H → ₹110,500
- F–G → ₹127,500
- D–E → ₹136,000



Total Cost Calculation

Non-Optimized Network Cost = ₹1,177,250

MST Optimised Network Cost =

76,500+80,750+93,500+102,000+110,500+127,500+136,00076,500 + 80,750 + 93,500 + 102,000 + 110,500 + 127,500 + 136,00076,500+80,750+93,500+102,000+110,500+127,500+136,000 =₹827,250

Cost Savings

₹1,177,250−₹827,250=₹350,000

Cost Reduction ≈ 30%

The final MST connects all eight nodes using seven links with the minimum possible total cost.

9. Cost Analysis and Results

9.1 Fully Connected Network Cost

In a fully connected network, a large number of links are required, resulting in very high installation costs. Such a design is impractical for real-world deployment.

9.2 MST Optimised Network Cost

The MST-based design significantly reduces the number of links while maintaining connectivity. The total cost of the MST network is substantially lower than that of a fully connected network.

9.3 Discussion

The results clearly indicate that MST-based network design offers a cost-effective solution for infrastructure planning. While redundancy is reduced, the cost savings make it suitable for small to medium-scale networks such as campuses.

10. Limitations and Future Scope

Although MST provides minimum-cost connectivity, it does not consider factors such as bandwidth, traffic load, reliability, or fault tolerance. Future work can integrate MST with redundancy planning, Quality of Service (quality of service) constraints, and dynamic network growth models.

11. Conclusion

This research paper presented a detailed and realistic study on cost optimisation in computer network design using Minimum Spanning Tree techniques. By modelling the network as a weighted graph and applying Kruskal's algorithm, we achieved a significant reduction in installation costs. The campus network case study demonstrates that MST-based design is practical, efficient, and suitable for real-world applications. The approach can be extended to other domains such as smart cities, power distribution, and transportation networks.

References

1. S. Raghavan and A. Magnanti,
"Network design and routing: Models and algorithms,"
Transportation Science, vol. 31, no. 2, pp. 101–128, 1997.
2. M. Pióro and D. Medhi,

Routing, Flow, and Capacity Design in Communication and Computer Networks, Morgan Kaufmann, 2004.

3. K. Thulasiraman and M. N. S. Swamy, *Graphs: Theory and Algorithms*, Wiley-Interscience, 1992.
4. S. K. Gupta and R. K. Singh, "Cost optimisation of campus network using minimum spanning tree," *International Journal of Computer Applications*, vol. 45, no. 7, pp. 21–26, 2012.
5. A. Kumar and S. Verma,s "Graph theory-based approach for network infrastructure optimisation," *International Journal of Computer Networks*, vol. 6, no. 3, pp. 89–95, 2014.