

# A Minimum Spanning Tree-based Routing Technique of FAT Tree for Efficient Data Center Networking

Prachi Gupta

CSE

Arya College of Engineering & IT

Jaipur

[guptaprachi005@gmail.com](mailto:guptaprachi005@gmail.com)

Vibhakar Pathak

Professor CS/IT

Arya College of Engineering & IT

Jaipur

[vibhakarp@rediffmail.com](mailto:vibhakarp@rediffmail.com)

## Article Info

**Page Number:** 79 – 85

**Publication Issue:**

**Vol 71 No. 1 (2022)**

## Abstract

Data centers (DCs) have become the critical and irreplaceable infrastructure to drive this ever-growing trend because of the exponential expansion of Internet services. In general, DC has several data & storage nodes connected by a network that is specially built, called Data Center Network (DCN). DCN functions as a communications backbone also play a significant role in maximizing the operation of DC. However, the infrastructure and operations are very difficult in comparison with the conventional network compared with unique DCN specifications, including a wide range of applications, high power density, and high reliability. In this paper, the Bellmanford MST algorithm is used for FAT-tree-based routing. The main characteristic of the FAT Tree is that the links which are used for connecting nodes from different levels may have different bandwidth. The Bellmanford algorithm used here is the shortest path algorithm which provides a reliable and short path for the route nodes to travel which provides fewer time complexities in data center networking.

## Article History

**Article Received:** 18 November 2021

**Revised:** 01 December 2021

**Accepted:** 15 December 2021

**Publication:** 27 January 2022

**Keywords:** - Cloud Computing, Data Center Network, Fat-Trees, Prim's Algorithm, Bellman-Ford Algorithm.

## I. INTRODUCTION

Cloud Computing (CC) is often defined as access storage and files, program, or application storage on the internet rather than on a PC. When the data is stored in the cloud, the data is not destroyed, since when data is processed in the cloud, the duplication of our data was generated and transmitted to various servers in the cloud. Protection that shifts from unauthorized person to data is the risk factor for cloud use. Cloud has two primary development model and application model models. The delivery model involves service infrastructure, service portal, and software service. The model deployment includes proprietary public clouds and hybrid clouds. It also includes four layers, which are the hardware layer and the infrastructure layer. It has also appealed to many companies by use of all these cloud providers. CC technology is a technology with the highest growth rate in the IT market, with a high growth rate in recent years. The number of CC providers is increasing exponentially and is one of the main mood words currently.

DC has recently gained considerable attention as an affordable platform for storing massive volumes of data and hosting major service applications. Big corporations including Amazon, Apple, Facebook, and Yahoo! are now using DC regularly for storage, Web service, and computation. With the advent of CC, service hosting is now a multi-billion-dollar company that is a major player in the future IT industry

The architectures of today's DC are far from ideal despite their importance. DC typically use dedicated servers to run apps, leading to poor server use and high operating costs. Through the development of server virtualization technology (e.g. VMware and Xen), multiple VM (virtual machines) could be co-located to a single physical computer, the situation was enhanced. These

technologies will provide efficient insulation between collocated VMs to improve application performance and avoid attacks from interfering. The only way of addressing any limitation of the existing DC architectures is to virtualize the server. [3].

DC is an irreplaceable and essential infrastructure to allow Internet services and applications to continually expand. These steadily expanding applications and services need a great deal of computational and storage capital. In response to this desperate need, DC is built in a consolidated and managed environment to house huge networked computer systems. A large number of computer and storage node interconnections are formed within a DC through a specifically built network known as a DCN network. The DC network, used in a data center, describes the networking networks used by network topologies, routing and switching equipment and protocols used (for example Ethernet and IP)

The DCN also enables operational optimizations to DC in addition to connectivity to achieve fundamental compromise among increased costs & essential business continuity. However, in comparison to networks and traditional networks (Ex., WAN (Wide Area Network) and LAN (Local Area Network)), DCN's architectural & operating principle is based on a set of specific issues, specifications to be overcome. It summarizes, in our view crucial, a selection of specific challenges and requirements for developing and operating DCN:

- Wide Variety of Applications
- Large Scale
- Strict Service Requirement
- High Energy Consumption [4].

DCN is a vital component of a data center architecture, which enables the communication between intra-data centers and a vast range of computer resources. The function and performance of the data center are largely defined by characteristics of DCN and its traffic control mechanisms. Also, the economic viability of DC relies heavily on service at reasonable (or high) levels of efficiency and usage of infrastructure and resources by users and applications. [5].

DCN offers a combination of electric, optical, and wireless technology. Several proposals are focused on the operation of DCN's system to accomplish a variety of targets, including high network use, low service latency, bandwidth assurance, high network robustness, and lower power usage. We are not completely and extensively protected in DCN studies. It is our humble objective to logically improve the recent attempts to study the DCN culture and to propose some potential opportunities for research in this vibrant field.

## II. LITERATURE REVIEW

**Wei, W., et al. [2020]** In this paper, they devote to how to balance multiple Use of resources to reduce resource fragmentation while maximizing virtual machine placement service rate, ensuring that physical resource waste and unnecessary output are avoided. They give a combined heuristic and genetic algorithm, which offers an approximate optimal solution with even less time difficulty, to solve this problem of bi-objective optimization [6].

**Namasudra, S., et al. [2020]** Companies have to rely on the untrusted in a cloud environment. Other problems with data access from the cloud environment include data protection, high search time for data users, high data access time, and overheard of high system. In this paper, a new, stable, and fast data access management paradigm is presented for the cloud environment to address all these problems. Here the table for quick and efficient data access is held by the cloud service provider. A long 1024-bit password or secret key based on Deoxyribonucleic Acid (DNA) is used for encrypting private or sensitive information for users of this proposed scheme. Experimental findings and analytical analysis show the efficacy of the proposed model against existing models [7].

**Zheng, G., et al. [2020]** Experimental findings suggest that dynamic cloud expansion algorithm, dependent upon dynamic programming concepts, can enhance performance and operational efficiency of VM groups more efficiently and quickly, enhancing Cloud dynamics based on heuristic functions. The idea is for an effective distribution of capital based on co-operative quality learning. For distributed resource block allocation, environment knowledge is gained from learning from the base station and information sharing. This allocation scheme can accomplish an optimum strategy for allocating personnel in a limited period and can complete learning in compliance with delaying needs of various services at any time. It can effectively boost system performance in comparison to traditional allocation schemes [8].

**Fernández-Cerero et al. [2020]** The dynamic shift of DC core information management structure to optimize DC performance: resource management. DISCERNER is suggested as a model for decision sciences to define current management patterns of resources to maximize overall performance for some time, instead of new resource management patterns once new workloads and templates appear. The model is used to evaluate current management models of resources. For real-time decisions dependent upon previous execution logs & the last DC situation, a classic classification machine-learning is used. A proven DC simulation

platform has simulated a series of extensive and commercial experiments. The findings show that in practical situations at least 20 percent of values of main performance metrics can be improved [9].

**K. Naseer Qureshi [2018]** In this text, they suggested an ad hoc vehicle CC platform. This model offers vehicle node computing resources and enhances data transmission and network performance. Furthermore, this paper addresses various services proposed including data aggregation, security, privacy, and utilization of funds. Finally, the experimental findings show that the proposed model works better in the network [10].

**S. Kim et al. [2018]** In this post, they propose the latest DC Architecture network design, which calls for highly efficient executions and successful data-sharing between applications. Data-based network architecture is a popular method for building new businesses and values in a hyper-connected society [11].

**M. Balanici and S. Pachnicke [2018]** In this article, the design and performance evaluation of the ICN are based on custom, long-term models for the generation of data traffic for many application groups. By applying accurate modeling to machine-produced data, and using presumed DC hardware accessible today and in the future, we evaluate efficient assessment and configured intra data center networks. Furthermore, we show how an optical circuit switching can be used to convert the Fat-Tre Network into its Hybrid optical equivalent, significant capital, and cost savings for respectable position connections [12].

**H. Yao [2017]** suggests an algorithm for a new TCP slow start for DCN known as the Slow Start algorithm (SDSS), which is based on the software-defined networking (SDN) technology.

software established. The proposed slow start thresh (Ssthresh) and congestion window (cwnd) can also be used with the proposed SDSS algorithm to exceed available bandwidth in any slot. Since the data transmission rate is not higher in each slot than the usable bandwidth, the risk of congestion reduces. The simulated results reveal that the SDSS algorithm being introduced would increase network congestion and decrease the capacity of cursor keys as compared to existing algorithms [13].

### III. RESEARCH METHODOLOGY

#### A. PROBLEM FORMULATION

The costs of the data center are concentrated in this order in servers, infrastructure, electricity requirements, and networking. Although prices are high, usage can be remarkably low. Resource spreading and fragmentation between data centers also result in low usage of resources and thus an increase in time consumption.

#### B. PROPOSED METHODOLOGY

To face these challenges, DCN algorithms are implemented and compared with each other for the result comparison. The algorithms implemented are FAT Tree using Prim's algorithm and FAT Tree algorithm with Bellmanford algorithm. In previous research work, the FAT Tree algorithm with Prim's algorithm was used which was later replaced by the Bellmanford algorithm. In sink source-based networks we use the Bellmanford minimum spanning tree (MST) algorithm to obtain the best routing path.

##### 1. FAT – TREES

Fat trees are innately tolerant because of the diversity of their paths, that is, several paths connecting to each destination each source. Fat Trees introduced by Charles E originally. Leiserson in as a choice of networks for parallel computation. The model is conceived as a set of processors, which share no common memory, which is interconnected via a routing network. The connected processors communicate using messages via the routing fabric it is connected with. The routing fabric is charged with the duty of delivering all the messages generated by the processors to reach their respective destinations. Although a Fat – tree takes the form of a binary tree, the said tree resembles more real-life trees rather than the common trees we generally study in Computer Science, because the links connecting the root becomes fatter as we move upwards towards the root. The leave nodes represent the IP cores viz. processors, memories, etc., and the non-leaves represent the routing nodes or commonly known as network nodes responsible for routing the packets from source to destination [14].

##### 2. SPANNING TREE

The Minimum Spanning Tree (MST) is a spanning tree with a reduced total edge weight. The minimal span is used for a series of edges, to reduce the path between each node and total edge weight. You can create a minimum tree using greedy algorithms, that is Algorithm of Kruskal and Prim. Prim's algorithm is based on visuals. In each passage, the graph selects a random node, checking all edges from the visited node to walking activities, choosing the highest edge, then adding the edge to MST.

**a) Prim's algorithm**

Prim is a gullible algorithm that solves connected and weighted undirected graph's MST problem. MST consists of a series of corners that connect each vertex in the original graph to minimize the overall weight of the tree's edges. The algorithm begins at an altered node of the graph, evaluating all of the edges accessible from the non-visited nodes so that the one at the lowest costs can be chosen. The selected rim will then be placed at the nodes visited and the edge will be added to that of the MST.

The following steps are taken to find MST using the Prim algorithm:

1. Where the graph has parallel edges and loops, remove loops and parallel edges.
2. Choose any node randomly, mark it 0, and label any other node as a  $\infty$ . The chosen node is known to be the new node. All other nodes are not visited.
3. Detect unvisited network nodes that are currently connected. Calculate distance to present node from unvisited nodes.
4. When the corresponding mark value is less than, weigh and replace each node with the node. The weights of the nodes are still marked to keep track of them as small as possible.
5. Mark the node as it is visited with color. Once we have a vertex, we don't have to look back.
6. Find the node of every uninvited node with the least weight, consider that node visited, and consider this node to be the new work node.
7. Repeat steps three, four, and five to all nodes.
8. All steps will be MST after completion. [15,16].

**3. Bellman-Ford Algorithm**

This is Richard Bellman's algorithm of 1958, Lester Ford's 1956. This is complex. It scans the structure and provides the best solution. This algorithm is for Shortest Path (SP), where the graph from the source node through all other nodes can be negatively weighted.

Bellman-Ford algorithm offers a dynamic source solution for all nodes in the diagram for minimum route issues in graphs with negative edges however not negative cycles for single-source shortest paths. The algorithm's main advantage is its simplicity which doesn't require complex data structures for implementation & also easily and with high accuracy, finds minimum weight.

Bellman-Ford algorithm utilizes relaxation to search shortest routes in single-source graphs. And there are negative edges there as well. The algorithm is often used to detect whether weight cycles are negative (no solution is provided). There is no negative distance when you ask about distances on a map. The basic form of the Dijkstra algorithm is identical. The  $|V| - 1$ -times relaxes per vertex, under which the  $|V|$  number is in the graph of vertices. The cost of a route is the total weighted edges of the path. Now SRV is a source that is the shortest path issue with the single source so you can find the distance from one single source to another spot. This algorithm returns the value of the negative cycle and gives the shortest path. This algorithm finds the shortest way down [17, 18].

**IV. RESULT AND DISCUSSION**

In our experiment, we had used switches, aggregate switches, virtual machines, routers, and Root nodes. The objective of our experiment is to compare Fat-tree-based routing using the aggregate switch as Fat-tree root node and the network beneath the root node is considered as sink source base network. In sink source-based networks we use Prim's minimum spanning tree (MST) and Bellmanford algorithm to obtain the best routing path. From the results obtained we can see that the Bellmanford algorithm can perform better on FAT Tree as compared to that of Prim's Algorithm.

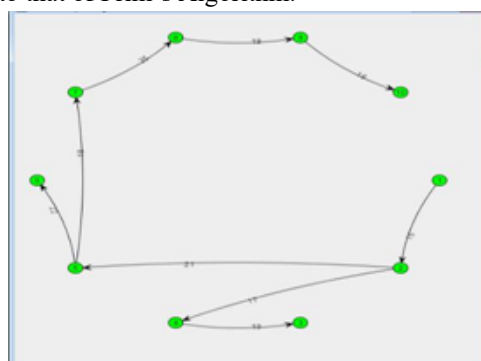


Figure 1: the shortest path through After Prim's

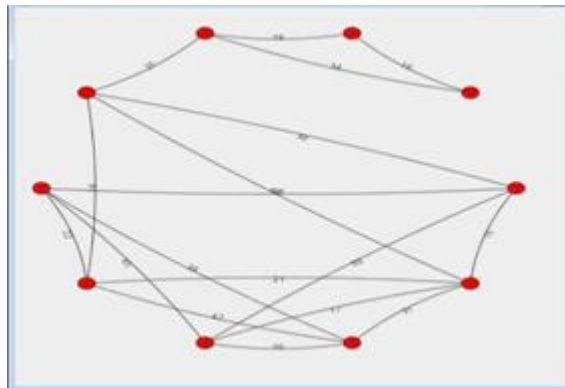


Figure 2: the shortest path through before Prims

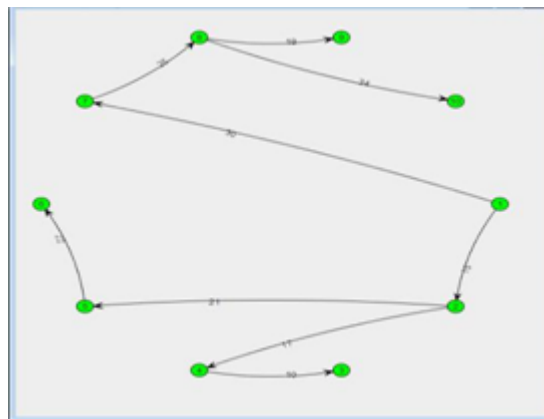


Figure 3: the shortest path through After Bellman-Ford

**Table 1: Comparison Table of Time Taken by Base & Propose Approach**

Route	Time Taken in ( $\mu$ sec)	
	Base (FatTree + Prims algorithm)	Propose (FatTree + Bellman-Ford algorithm)
e1	144	140
e2	149	143
e3	139	131
e4	133	129
e5	146	143
e6	143	140
e7	134	127
e8	132	130
e9	142	137
e10	218	212

```

===== OUTPUT =====
Route  Time taken (in µsec)  %age of Total Time
-----
e1     144                   9.72973%
e2     149                   10.067568%
e3     139                   9.3918915%
e4     133                   8.986486%
e5     146                   9.864865%
e6     143                   9.662162%
e7     134                   9.054054%
e8     132                   8.918919%
e9     142                   9.594595%
e10    218                   14.72973%
    
```

Figure 4: The comparison of time percentage for every route of Prim’s algorithm



Figure 4: Comparative line graph of Execution time between Base & proposed Approach

```

===== OUTPUT =====
Route  Time taken (in µsec)  %age of Total Time
-----
e1     140                   9.776536%
e2     143                   9.986033%
e3     131                   9.148045%
e4     129                   9.00838%
e5     143                   9.986033%
e6     140                   9.776536%
e7     127                   8.868715%
e8     130                   9.078213%
e9     137                   9.5670395%
e10    212                   14.804469%
    
```

Figure 5: The comparison of time percentage of every route by Bellmanford

### V. CONCLUSION

DCN (Data Center Network) acts like and plays a major role as a backbone for networking in new DC. Nevertheless, the specific characteristics of DCNs, like large scale applications, broad range, high reliability, and high energy density, represent major challenges in infrastructure and services architecture and operation. As a result, DCN has become one of the most popular study fields, and a lot has been done to fix different issues. Minimum Spanning Tree source-sink routing has the inbuilt property of resilience between a node to node communication. Various communication delays among nodes are found to be least in MST than traditional routing in both cases of the geographical mean area and far area.

## REFERENCES

1. N. Dhivya," A Survey Paper On Cloud Computing", IJARSE, ISSN:2319-8354, VOL-06, December 2017.
2. Mohamed Magdy Mosbah," Current Services In Cloud Computing: A Survey", International Journal of Computer Science, Engineering and Information Technology (IJCEIT), Vol.3, No.5, October 2013
3. Bari, M. F., Boutaba, R., Esteves, R., Granville, L. Z., Podlesny, M., Rabbani, M. G., ... Zhani, M. F. (2013). *Data Center Network Virtualization: A Survey. IEEE Communications Surveys & Tutorials, 15(2), 909–928.* doi:10.1109/surv.2012.090512.00043
4. Xia, W., Zhao, P., Wen, Y., & Xie, H. (2017). A Survey on Data Center Networking (DCN): Infrastructure and Operations. *IEEE Communications Surveys & Tutorials, 19(1), 640–656.* doi:10.1109/comst.2016.2626784
5. Rojas-Cessa, R., Kaymak, Y., & Dong, Z. (2015). *Schemes for Fast Transmission of Flows in Data Center Networks. IEEE Communications Surveys & Tutorials, 17(3), 1391–1422.* doi:10.1109/comst.2015.2427199
6. Wei, W., Wang, K., Wang, K., Gu, H., & Shen, H. (2020). *Multi-resource balance optimization for virtual machine placement in cloud data centers. Computers & Electrical Engineering, 88, 106866.* doi:10.1016/j.compeleceng.2020.106866
7. Namasudra, S., Sharma, S., Deka, G. C., & Lorenz, P. (2020). *DNA computing and table-based data accessing in the cloud environment. Journal of Network and Computer Applications, 102835.* doi:10.1016/j.jnca.2020.102835
8. Zheng, G., Zhang, H., Li, Y., & Xi, L. (2020). *5G network-oriented hierarchical distributed cloud computing system resource optimization scheduling and allocation. Computer Communications.* doi:10.1016/j.comcom.2020.10.005
9. Fernández-Cerero, D., Ortega, F. J., Jakóbič, A., & Fernández-Montes, A. (2020). *DISCERNER: Dynamic selection of resource manager in hyper-scale cloud-computing data centres. Future Generation Computer Systems.* doi:10.1016/j.future.2020.10.031
10. K. Naseer Qureshi, F. Bashir, and S. Iqbal, "Cloud Computing Model for Vehicular Ad hoc Networks," 2018 IEEE 7th International Conference on Cloud Networking (CloudNet), Tokyo, 2018, pp. 1-3, DOI: 10.1109/CloudNet.2018.8549536.
11. S. Kim, S. Kim, S. H. Park, and S. Kim, "A Data-centric Network Architecture for Service Optimization and Data Sharing," 2018 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, 2018, pp. 505-508, DOI: 10.1109/ICTC.2018.8539470.
12. M. Balanici and S. Pachnicke, "Hybrid electro-optical intra-data center networks tailored for different traffic classes," in *IEEE/OSA Journal of Optical Communications and Networking*, vol. 10, no. 11, pp. 889-901, Nov. 2018, DOI: 10.1364/JOCN.10.000889.
13. H. Yao, W. Muqing and L. Shen, "An SDN-based slow start algorithm for data center networks," 2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, 2017, pp. 687-691, DOI: 10.1109/ITNEC.2017.8284820.
14. Abhijit Biswas, Md. Anwar Hussain," A Survey of FAT – TREE Network – On-Chip Topology", International Journal Of Scientific & Technology Research Volume 8, Issue 11, November 2019
15. Rushita Acharya," A Survey of Prim's Algorithm", International Journal of Recent Advancement in Engineering & Research, Volume 1, Issue 1; December -2015
16. Jogamohan Medak," Review and Analysis of Minimum Spanning Tree Using Prim's Algorithm", International Journal of Computer Science Trends and Technology (IJCT) – Volume 6 Issue 2, Mar-Apr 2018
17. Jitendra Bahadur Singh," Investigation of Bellman-Ford Algorithm, Dijkstra's Algorithm for suitability of SPP", 2018 International Journal of Engineering Development and Research IJEDR | Volume 6, Issue 1 | ISSN: 2321-9939
18. Vaibhavi Patel, "A survey paper of Bellman-ford algorithm and Dijkstra algorithm for finding the shortest path in GIS application", International Journal of P2P Network Trends and Technology (IJPTT) – Volume 4 Issue 1 January to February 2014