A Refined Algorithm for Efficient Route Identification in Future Generation Networks

P. Calduwel Newton[1], Dr. L. Arockiam[2], Dr. E. George Dharma Prakash Raj[3]
R. Hari Prasath[4], and Tai-hoon Kim[5]
[1]*Lecturer, Bishop Heber College (Autonomous), Tiruchirappalli-620017*
[2]*Lecturer (SG), St.Joseph's College (Autonomous), Tiruchirappalli-620002*
[3] *Lecturer, Bharathidasan University, Tiruchirappalli-620023*
[4]*Programmer, FI Information Technology, Chennai, INDIA*
[5]*Dept. of Multimedia, Hannam University, Korea*
calduwel@yahoo.com

Abstract

The challenges faced by the today's network are significantly increased. They decrease the efficiency of the network. If enough measures are not taken, this will lead to severe problems in future generation networks. Though many parameters are involved in improving the efficiency, this paper focuses on distance and bandwidth. The proposed algorithm identifies the shortest route faster than Dijkstra's algorithm. It suggests new ideas in constructing source-to-destination route by refining the steps in existing algorithm. It also identifies the bandwidth-effective route using proposed algorithm. Ultimately, the outcome of this paper increases the efficiency of the network.

Keywords: *Future Generation Network, Bandwidth, Source-to-destination route, Dijkstra's algorithm, Network Efficiency*

1. Introduction

In the fields of networking and communications, there is a continual concern over poor bandwidth utilization. It is some times the case that host computers can employ applications or processes that are very bandwidth intensive, limit the amount of bandwidth that other hosts have at their disposal [1].

Nowadays the data to be sent through the network has different types. They are text, audio, video, animation, images, pictures, etc. Though the networking speed increases, the range of traffic rates which the network has to deal with widens. This mixture of different types of data makes the traffic pattern less predictable and its distribution more uneven.

The host employing applications or processes having high bandwidth requirement limit the bandwidth available to other hosts. Shortest route is a route with minimal length from source to destination. Some of these routes have higher bandwidth than others, some have lower propagation delay, and others see less congestion. These factors are responsible for the end-to-end performance achievable along any given route [2].

The existing routing algorithms have served remarkably well in the network environment where traffic load is light and network conditions change slowly. They are able to respond to topological changes automatically and adjust routing decisions when traffic changes. In the presence of congestion, shortest routing algorithms can reduce the traffic from the overloaded routes.

In a network environment where traffic approaches the capacity of routes and changes dynamically, shortest-path routing algorithms, particularly those that attempt to adapt to traffic changes, frequently exhibit instability, derive poor-quality routes and result in performance degradation [3, 4].

Our algorithm helps the network to be stable, to find quality route and to upgrade the performance. The above characteristics are possible when the routes are selected in minimum number of computations. In this paper, we have dealt with the temporal complexity. Though spatial complexity is also an another issue, it is out of the scope of this paper.

This paper is organized as follows. Section 2 highlights the motivations for this paper. Section 3 contributes to the proposed algorithm. Section 4 analyzes Dijkstra's algorithm and the proposed algorithm. Section 5 gives the conclusion and future enhancements. Section 6 lists the references.

2. Motivations

This section identifies various existing algorithms used to find the shortest route. It also gives the previous research works on route selection. Link state routing algorithm uses Dijkstra's algorithm to find shortest route. But, it takes long time for each router to compute its routing table [5]**.** Selecting a shortest route in a dynamically changing network environment is a challenge.

One of the major concerns in the Internet-based information society today is the tremendous demand for more and more bandwidth. There is great need for dynamically managing traffic demands and balance the network load [6].

Traffic engineering should be viewed as assistance to the routing and switching infrastructure that provides additional information used in forwarding traffic along alternate routes across network, trying to optimize service delivery throughout the network by improving its balanced usage and avoiding congestion caused by uneven traffic distribution. Traffic engineering signifies the ability to place traffic where the capacity exists to accommodate it. But the network engineering refers the ability to install capacity where the traffic exists.

When a traffic-engineering application implements the right set of features, it should provide precise control over placement of traffic flows within a routing and switching domain, gaining better network use and realizing a more manageable network [6].

Varadthan et al. present a simulation study on the effect of route changes on the performance of transport protocols. They show that small route changes during a TCP session can lead to significant reordering and a consequent reduction in performance [7].

Traffic engineering aims to makes more efficient use of network resources. Although it does not increase the bandwidth of the network, it enables us to transmit more data. With the growth of internet traffic approximately doubling every year, it is necessary to take measure to reduce traffic problems [8]. In the following [10], we will present a number of path selection method that are used today.

2.1. Widest-shortest path (WSP)

Widest-shortest path selects a path with minimum hop count among all feasible paths. In this case, higher bandwidth cannot be achieved. [9] If there are several such paths, the one with the maximum reserved bandwidth is selected.

2.2. Open shortest path first (OSPF)

In this method the total area is divided into number of sub-areas according to the backbones .Each router in a particular area that will compute each neighborhood routers and their cost. By using this information, it will find the shortest path of the graph.

2.3. Border gateway protocol (BGP)

The source will examine all the outgoing paths and finds the best path for the next neighborhood router .The same calculation is followed by all other routers If one path is down then it will choose the next best path.

2.4. Routing information protocol (RIP)

In this case each router that will store the information about neighborhood router that will be updated from time to time. According to this information it will be dynamically changed.

After analyzing the existing algorithms, it can be concluded as follows:

- Some algorithms are not finding the minimum distance path.
- The amount of information stored in each router is high.
- Repeated tracing is done.

The efficiency of an algorithm can be measured in terms of computational complexity, storage complexity and communication complexity [9]. This paper restricts only with computational complexity. That is, storage complexity and communication complexity are beyond the scope of this paper.

From the above discussions, we can conclude that reducing the computational complexity and increasing the efficiency of the network are noble contributions.

3. Proposed algorithm

This section proposes an algorithm in which the incoming routes to each node play an important role. When more than one incoming routes meet in a same node, only then comparison is made and route with minimum distance is selected. This is done repeatedly until it reaches the destination.

3.1. Algorithm

1 Determine all the intermediate route(s) with their distances from source to destination
2 Split the source route(s) and remaining route(s) (where source route is an intermediate route that starts from source node)
3 Using ending node of the source route(s) / intermediate route(s), build the end-to-end route with the remaining intermediate route(s)
4 If more than one route meets at the same node
    then
        Select a route with minimum distance
    End if
5 Repeat Step 3 and 4 until the destination is reached

As the network is growing exponentially, a route selected as an optimum may not be permanent. Assume that the route from the source to next intermediate node as the source route. Now compare each source route with all the intermediate routes [10, 11]. Then check whether the ending node of the source route and the starting node of the intermediate route are same or not. If both are same then add the distance of the two routes. The new source route is again compared with the other routes. While building the route, select the intermediate route with minimum distance. Do this until the final destination is reached.

4. Analysis

This section has two sub-sections. First section is used to compare the Dijkstra's shortest route algorithm with the proposed algorithm. Second section is used to find the bandwidth-effective route using proposed algorithm.

4.1. Dijkstra's algorithm vs. proposed algorithm

This section discusses the pros and cons of Dijkstra's algorithm with proposed algorithm and gives the best results. From Figure 1, source routes are identified. Then, various intermediate routes to the destination are identified. From these routes, next node to the destination is identified. Further route building process is done with minimum distance route. This process is done until it reaches the destination.

By keeping track of each flow, nodes that could not provide requirements and nodes that are misbehaving can be identified and corrected [12, 13]. Assume the source node as 'A' and destination node as 'D'. The following steps explain the proposed algorithm in detail with an example network (Figure 1).

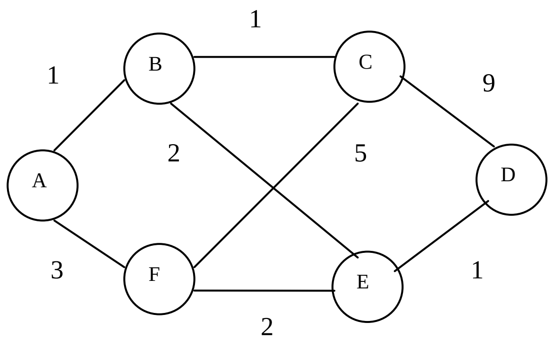

Figure 1. A Network with Distance

In Figure 1, the Intermediate Routes and their distances are,

A→B=1
B→C=1
C→D=9
B→E=2
A→F=3
F→E=2
E→D=1
F→C=5

Figure 1 is analyzed using Dijkstra's shortest route algorithm and then it is analyzed using the proposed algorithm. The various steps involved in the Dijkstra's shortest route algorithm are as follows. The asterisk (*) indicates that the shortest route computation is over.

Step 1:

The source routes are,
A→B=1
A→F=3

Step 2:
From step 1, select one route with minimum distance. Hence,
A→B=1
Step 3:
Move from node B to next node and compare
A→B→C=2
A→B→E=3
A→F=3
Step 4:
From step 3, select a route with minimum distance. Then,
A→B→C=2
Step 5:
Move from node C to next node and compare
A→B→C→D=11 *
A→B→E=3
A→F=3
Step 6:
Now consider the remaining routes, i.e. move from node E to next node. Then
A→B→C→D=11 *
A→B→E→D=4 *
A→F=3
Step 7:
Now, move from node F to next node
A→B→C→D=11 *
A→B→E→D=4 *
A→F→E=5
Step 8:
Then, move from node E to next node
A→B→C→D=11 *
A→B→E→D=4 *
A→F→E→D=6 *

From step 8, the ultimate shortest route is, A→B→E→D=4.

The following steps are derived from the proposed algorithm.

Step 1:
The source routes are,
A→B=1
A→F=3
Step 2:
Now, find the next intermediate route from the nodes B and F.
A→B→C=2
A→B→E=3
A→F→E=5
A→F→C=8
Step 3:

From step 2, select one route with minimum distance meeting at the same node. They are,

A→B→C=2

A→B→E=3

Step 4:

Then, finding of next intermediate route from C and E is done.

A→B→E→D=4 *

A→B→C→D=11*

From step 4, the ultimate shortest route is, A→B→E→D=4.

From the analysis, it is found that the Dijkstra's algorithm takes approximately eight iterations. But, the proposed algorithm takes only four iterations. This proves that the routes are computed quickly with minimum number of computations. Number of comparisons is also reduced.

The proposed algorithm is simulated using Java. Figures 2, 3 and 4 show that the results of the proposed algorithm.

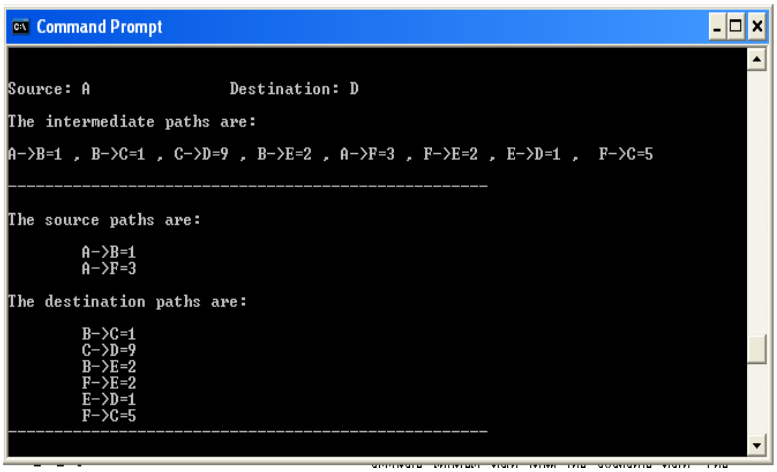

Figure 2. Intermediate Routes and their Distances

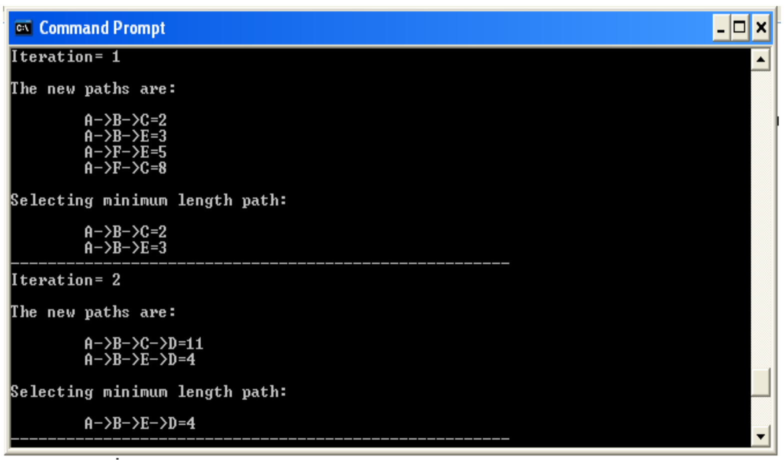

Figure 3. Route Selection in Progress

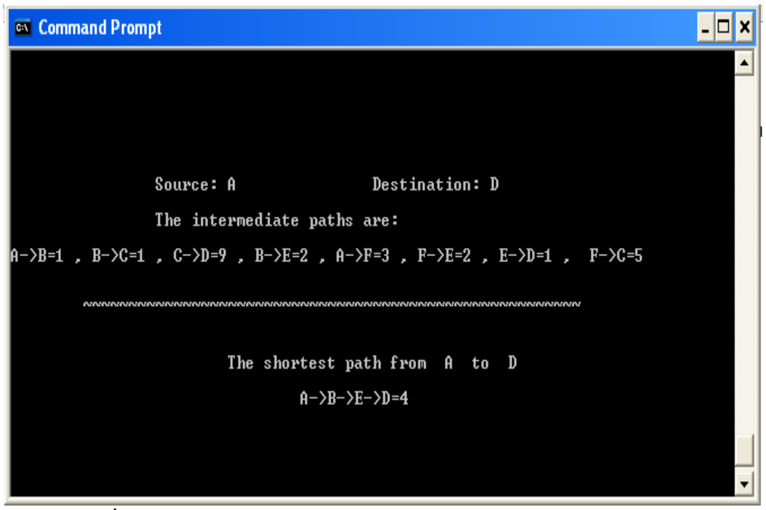

Figure 4. The Shortest Route

4.2. Identifying bandwidth-effective route

This section helps to identify the bandwidth-effective route from an example network. In Figure 5, the number on the link represents the unused bandwidth in percentage. For example, link from A to B has 20. It means 20% of the total bandwidth is unused. Assume all links have equal bandwidth, say 10 Mbps.

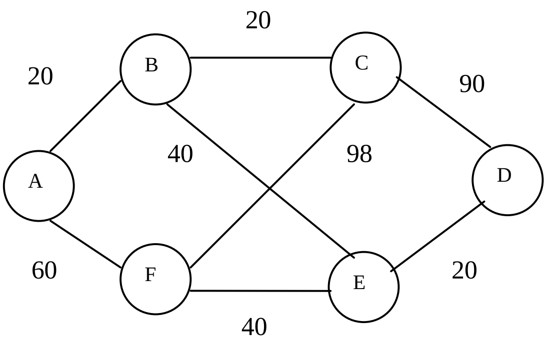

Figure 5. A Network with Unused Bandwidth

The proposed algorithm will identify the bandwidth-effective route as given below. It follows the virtual-circuit method to find a route. First, it will establish a route from source to destination. Second, it will send the data.

Step 1:

The Initial routes are,
A→B=20
A→F=60

Step 2:

Now, find the next intermediate route from the nodes B and F.
A→B→C=20
A→B→E=20

A→F→E=40
A→F→C=60
In this step, A to B has unused bandwidth 20%, B to E has 40%. But, A → B → E can avail only 20% because it concentrates on even flow of data from source to destination.

Step 3:
From step 2, select one route with maximum unused bandwidth meeting at the same node. They are,
A→F→E=40
A→F→C=60

Step 4:
The final routes are,
A→F→E→D=20
A→F→C→D=60

From step 4, the bandwidth-effective route is, A→F→C→D=60, since the link has 60% unused bandwidth.

5. Conclusion and future enhancements

This paper has proposed an algorithm and compared it with Dijkstra's shortest route algorithm. The result shows that our algorithm computes the routes very quickly. Almost it takes only half of the time compared to Dijkstra's algorithm. It greatly reduces the time complexity in finding the shortest route. It also finds the bandwidth-effective route. We are sure that, the outcome of this paper increases the efficiency of the network by identifying the path quickly. It also helps to increase bandwidth utilization and decreases the delay and jitter. Therefore, the efficiency of the network is increased.

The result comes from simulating the algorithm. It has been simulated using Java. In future, it may be implemented to exploit the advantages. Also, it may be extended to be used in multicasting networks and in future generation networks.

References

[1] H.Zhang, D. Ferrari, Connection Admission Control for Bandwidth Management of an Internet Access Link, Communications Magazine, IEEE, Vol. 38, Issue 5, May 2000, pp. 160-167.

[2] Stefan Savage, Andy Collins, Eric Hoffman, John Snell and Thomas Anderson, "The End-to-End Effects of Internet Route Selection", SIGCOMM 1999, pp.289-299.

[3] D. Bertsekas, ''Dynamic Behavior of Shortest Route Routing Algorithms for Communication Networks,'' *IEEE Transactions on Automatic Control*, vol. AC-27, no. 1, Feb. 1982.

[4] A. Khanna and J. Zinky, ''The Revised ARPANET Routing Metric,'' Proc. of ACM SIGCOMM'89, Sept. 1989, pp.45-56.

[5] Behrouz A. Forouzan, "Data Communications and Networking", Third Edition, 2004, pp.558.

[6] Francesco Palmieri, "GMPLS Control Plane Services in the Next-Generation Optical Internet", The Internet Protocol Journal, Vol.11, Number 3, September 2008.

[7] Kannan Varadhan, Deborah Estrin, and Sally Floyd, "Impact of Network Dynamics on End-to-End Protocols: Case Studies in TCP and Reliable Multicast". Technical Report USC-CS-TR 98-672, University of Southern California, Information Sciences Institute, April 1998.

[8] K.G. Coffman and A.M. Odlyzko, "Internet Growth : Is there a Moore's Law for data traffic?", In J. Abello, P.M. Pardalos and M.G.C. Resende, Editors, Handbook of Massive Data Sets, Kluwer Academic Publishers, 2001.

[9] Zheng Wang, "Routing and Congestion Control in Datagram Networks" Ph.D. thesis, University College London, January 1992, pp.32.

[10] E.George Dharma Prakash Raj, S.V.Kasmir Raja, Sinthu Janita Prakash, “QoS Routing for Communication Quality” Proceedings of the Asia Pacific Conference on Parallel and Distributed Computing Technologies, Vellore Institute of Technology, Vellore. South India, December 2004, pp.566 – 573.

[11] Calduwel Newton P , George Dharma Prakash Raj, Thomson Fredrick “An Adaptive Throughput-Aware Route Selection Algorithm”, Proceedings of the International Conference on Advanced Technologies in Telecommunication and Control Engineering, INTI College , Malaysia, August 2006, pp.66.

[12] Calduwel Newton P., “A Contemporary Technique to Guarantee Quality of Service (QoS) for Heterogeneous Data Traffic”, Proceedings of the International Conference on Information Security and Assurance (ISA 2008), IEEE CS, Korea, April 2008. pp. 210 – 213.

[13] Calduwel Newton P. et. al., “NPASA : A Noble Path Selection Algorithm”, Proceedings of International Symposium on Computer Science and its Applications (CSA 2008), IEEE CS, Australia, October 2008, pp. 52-55.

Authors

Authors did not want to publish their photos and bio-data.