Finding Shortest Path TCP Routing by Modified Dijkstra's Algorithm

Zin Me Me Win, Tin Win Maw
Computer University, Taung-Ngu, Myanmar
Zinmelay.88@gmail.com, tinwinmaw2009@gmail.com

Abstract

With the significant increment of today's network, there is a decrease in the efficiency of the network. The performance and reliability of the Internet depend, in large part, on the operation of the routing protocols. Routing protocols compute paths based on the network topology and configuration parameters. This paper presents the enhancing network routing by using modified Dijkstra's shortest path algorithm and focuses on distance. The proposed system identifies the shortest route faster than Dijkstra's algorithm. It provides optimized route by refining the steps in the Dijkstra's algorithm for computing and network efficiency. Simulation system is implemented for routers using java programming language.

1. Introduction

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.

Routing in computer network is an essential functionality, which influence both the network management as the quality of services in global networks. The routing algorithm is described by as network layer protocol that guides packets (information stored as small strings of bits) through the communication subset to their correct destinations. Routers forward IP datagram from one router to another on the path from source towards destination [7].

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 [9].

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 [1,3].

There are two ways to improve the efficiency of the route planning algorithms. One is to improve the traditional algorithms. The other is to put forward new algorithms [4]. This paper discusses the modified Dijkstra's shortest path algorithm to improve the network efficiency. The proposed 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.

2. Related Work

This section will present a number of path selection method that are used today.

Widest-Shortest-Path (WSP) selects a path with minimum hop count among all feasible paths. In this case, higher bandwidth cannot be achieved. [10] If there are several such paths, the one with the maximum reserved bandwidth is selected.

In Open-Shortest-Path-First (OSPF) 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 [5].

In Border-Gateway-Protocol (BGP) method, the source will examine all the outgoing paths and finds the best path for the next neighborhood routers. If one path is down then it will choose the next best path.

In Routing-Information-Protocol (RIP), each router that will store the information about neighborhood router will be updated from time to time. According to this information, it will be dynamically changed [8].

After analyzing the existing algorithm, it can be concluded as fellows: Some algorithms are not finding the minimum distance path. The amount of information stored in each router is high. Repeated tracing is done.

This section also 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 [2]. Selecting a shortest route in a dynamically changing network environment is a challenge.

The efficiency of an algorithm can be measured in terms of computation complexity, storage complexity and communication complexity [10]. This paper restricts only with computational complexity.

3. Network Routing Algorithm

Routing is a process of transmitting packets from its source node s to its destination node d on the network. Routing algorithms are classified as dynamic (adaptive) algorithm and static (non-adaptive) algorithm. In case of non-adaptive algorithms, the routing decision is based on local information. Shortest path routing algorithm also known as the Dijkstra's algorithm, is one of the most widely deployed non-adaptive routing algorithm in today's internet works.

3.1 Dijkstra's Algorithm

The Dijkstra's shortest path algorithm is the most commonly used to solve the single source shortest path problem today. The flow of Dijkstra's algorithm is a follow:

```
Initialization
S= {u}
for all nodes v
if v adjacent to u {
D (v) =c (u, v)
else D (v) = ∞

Loop
find w not in S with the smallest D (w)
add w to S
update d (v) for all v adjacent to w and not in S:
D (v) = min {D (v), D (w) +c (w, v)}
until all nodes in S
```

S = nodes whose least-cost path definitively known
u = source node, v = destination node
w = neighbor node
D (v) = current value of cost of path from source to destination v
c (u, v) = link cost from node u to node v
c (w,v) = the cost of the vertices conne*ct* to w
∞ = a constant larger than the greatest possible path length

3.2. Modified Dijkstra's Algorithm

This system uses 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. Processing step of this algorithm is as follows:

1. Determine all the intermediate route(s) with their distance 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.

Assume that the route from next intermediate node as the source route. Now compare the each source route with the entire intermediate route [6]. Then check whether the ending node of source route and starting node of intermediate route are same or not. If both are same, then add the distance of two routes. The new source route is again compared with the other routes. While building route, select the intermediate route with minimum distance.

4. Proposed System

This paper presents enhancing TCP routing by using modified Dijkstra's shortest path algorithm for network efficiency. It is based on distance. There will be multiple nodes implemented in the system and a source and destination for the data packed sending. When sending data packet from source to destination, modified Dijkstra's shortest path algorithm is used to find the best path. Refinement is performed in each step to adapt the network changes.

Routing protocols decide on routes to be taken. Routers must have idea of topology of internet in order to pick best route to take decisions based on some least cost criterion. It may depend on the current conditions of the network. This system helps the network to be stable to find quality route and upgrade the performance. Routes are selected in minimum number of computations. Figure 1 presents the architecture of the proposed system.

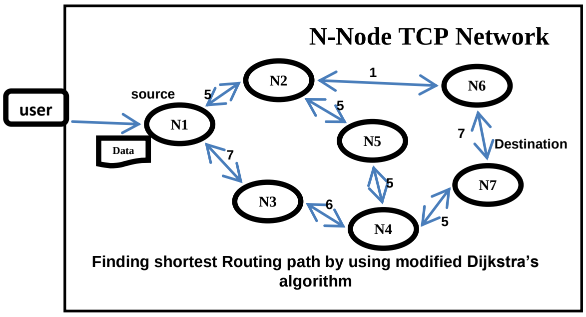

Figure 1: Proposed System Architecture

Source routes are identified in Figure 2. Then, various intermediate routes to the destination are identified. Further route building process is done with minimum distance route. This process is done until it reaches the destination. Assume the source node as 'A' and destination node as 'D'. The following steps explain the proposed algorithm in detail with an example network as in Figure2.

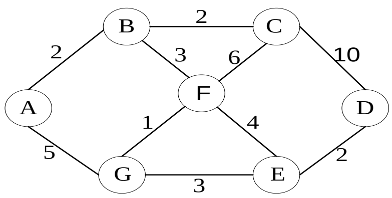

Figure 2: Sample Route Network Structure

In Figure 2, the Intermediate Routes and their distances are,

A → B = 2	D → E = 2
A → G = 5	E → F = 4
B → C = 2	E → G = 3
B → F = 3	F → G = 1
C → D = 10	
C → F = 6	

4.1Comparison of Dijkstra's algorithm and modified Dijkstra's algorithm

This section is used to compare the Dijkstra's shortest route algorithm with the modified Dijkstra's algorithm and gives the best result.

1. The source routes are:
A → B = 2
A → G = 5
2. From 1, the minimum distance is selected:
A → B = 2
3. Move from node B to next node and compare
A → B → C = 4
A → B → F = 5
A → G = 5
4. From 3, route with minimum distance is selected:
A → B → C = 4
5. Move from node C, and compute the next node, and compare
A → B → C → D = 14
A → B → F = 5
A → G = 5
6. Move from node F to next node and compare
A → B → C → D =14
A → B → F → C = 11
A → B → F → E= 9
A → G = 5
7. Compute the remaining node to get to D, (in this case compute C)
A → B → C → D = 14
A → B → F → C → D = 21
A → B → F → E = 9
A → G = 5
8. Compute the remaining node to get to D, (in this case Compute E)
A → B → C → D = 14
A → B → E → C → D = 21
A → B → F → E → D = 11
A → G = 5
9. Compute the remaining node G
A → B → C → D = 14
A → B → E → C → D = 21
A → B → F → E → D = 11
A → G → F = 6
A → G → E = 8
10. Move from node F to next node and compare
A → B → C → D = 14
A → B → E → C → D = 21
A → B → F → E → D = 11
A → G → F → C = 12
A → G → F → E = 10
A → G → E = 8
11. Compute the remaining node to get to D, (in this case compute C)
A → B → C → D = 14
A → B → E → C → D = 21
A → B → F → E → D = 11
A → G → F → C → D = 22
A → G → F → E = 10
A → G → E = 8
12. Move from E to D
A → B → C → D = 14
A → B → E → C → D = 21
A → B → F → E → D = 11
A → G → F → C → D = 22
A → G → F → E → D = 12
A → G → E → D = 10
13. Shortest path:
A → G → E → D = 10

Figure 3: Dijkstra's Algorithm

1. The source routes are:
A → B = 2
A → G = 5
2. Find next intermediate nodes from B and G
A → B → C = 4

A → B → F = 5
A → G → F = 6
A → G → E = 8

3. From 2, select the minimum distance for same node
F
A → B → F = 5
A → B → C = 4
A → G → E = 8

4. Move from node F to next node and compare
A → B → F → E = 9
A → B → F → C = 11
A → B → C = 4
A → G → E = 8

5. From 4, select the minimum distance for same node
(C and E)
A → B → C = 4
A → G → E = 8

6. Compute the distance for next intermediate node
A → B → C → D = 14
A → G → E → D = 10

7. From 6, the final shortest route is
A → G → E → D = 10

Figure 4: Modified Dijkstra's Algorithm

According to Dijistra's algorithm, it requires 13 steps and according to modified Dijkstra's algorithm, it requires only 7 steps.

The shortest path results obtained by computing Dijkstra's algorithm and modified Dijkstra's algorithm are same. However the computation steps are reduced by using modified Dijkstra's algorithm to find shortest path.

4.2 Simulation Results

The proposed algorithm is implemented as Windows based simulation system, developed using Java programming language.

From analysis, it is found that the Dijkstra's algorithm takes approximately thirteen iterations. But the modified Dijkstra's algorithm takes seven iterations. This proves that the routes are computed quickly with minimum number of computation Number of comparison is also reduced.

5. Conclusion

This paper presents the finding the best routing paths in sending data from computer to computer across the network. The best path is selected by computing the shortest path between source and destination. Dijkstra's shortest path algorithm is used to compute the shortest path. It is modified by refining in each step to get network efficiency.

Using this system, routing process can be optimized and sends data in the shortest path structure. Therefore it saves time as well as network usage. Moreover it modifies the Dijkstra's shortest algorithm, making less processing time in computing the best plan. Almost it takes only half of the time compared to Dijkstra's algorithm. It greatly reduces the time complexity in finding the shortest route.

The result comes from simulating the algorithm. We can conclude that reducing the computational complexity can increase the efficiency of the network.

6. References

[1] A.B. Forouzan, "*Data Communications and Networking*", Third Edition, pp.558, 2004.

[2] A. Khanna and J. Zinky, "*The Revised ARPANET Routing Metric*", Proc. of ACM SIGCOMM 89, pp.45-56, September 1989.

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

[4] F. Guerriero, Granat, Janusz, "The interactive analysis of the multicriteria shortest path problem by the reference point method", *European Journal of Operational Research 151*, 103-118, 2003.

[5] G. D. P.E. Raj, S.V.K. Raja and S. H. Prakash, *"QoS Routing for Communication Quality" Proceedings of the Asia Pacific Conference on Parallel and distributed Computing Technologies*, Vellore Institute of Technology, Vellore. South Idea, pp. 566-573, December 2004.

[6] J. Rexford, "*Route Optimization in IP Networks*", 2005.

[7] J. Wang, H. Steven, L. John and C. Doyle, "Can Shortest-path Routing and TCP Maximize Utility", California Institute of Technology, Pasadena, USA, *In the proceeding of IEEE INFOCOM 2003*.

[8] K. Varadhan, D. Estrin and S. 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.

[9] R. Monica, P. Klein, S. Rap, S. Subramanian and Henzinger,, "*Faster shortest-path algorithms for planar graphs*", Journal of Computer and System Sciences 55, 3-23, 1997.

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