A Multi-path Hybrid Routing Algorithm in Network Routing

Taehwan Cho* and Sangbang Choi

Department of Electronics Engineering, Inha University, Incheon, Korea
burujo@naver.com, sangbang@inha.ac.kr*

Abstract

The shortest path tree construction is essential in network routing, and the Dijkstra algorithm, which is a static routing algorithm, is widely used. When some links in a network have new weights, dynamic routing algorithms are more efficient than static routing algorithms. This is because the dynamic routing algorithms reduce redundancy by re-computing only the affected sections of the network with the changed links. However, the dynamic routing algorithms are not efficient in some situations and can increase computation time to create the shortest path tree. Therefore, hybrid routing algorithms, which reduce the total execution time of shortest path tree computation using the advantages of both static and dynamic routing algorithms, have been suggested recently. In this paper, a multi-path hybrid routing algorithm is presented that uses multi-path information to create the shortest path tree when some links have new weights. Comparisons with other routing algorithms, such as Dijkstra, Dynamic Dijkstra, and Hybrid Shortest Path Tree (HSPT), demonstrate that the multi-path hybrid routing algorithm provides better performance as the execution time decreases.

Keywords: *Hybrid routing, dynamic routing, multi-path, shortest path, network routing*

1. Introduction

In today's society, the demand for broadband Internet applications has grown rapidly. Therefore, high speed routing has become more important in Open Shortest Path First (OSPF), which is the most used intra-autonomous system routing protocol. When a topological change occurs due to an unexpected situation in the OSPF, the network routing algorithms are used to update the routing table. For example, if there is a link failure in a network, then the shortest paths must be recomputed. In this situation, the shortest paths computation is performed using the Dijkstra algorithm [1-3]. However, when some links have new weights in a network, using the Dijkstra algorithm can create redundancies through performing more computations and unnecessary corrections by repeating the operation for every node regardless of the location of the link weight change. Therefore, this can cause network instability because the overall routing table is frequently updated [4, 5].

These shortcomings have led to the introduction of dynamic routing algorithms in order to compute the shortest paths. Dynamic routing algorithms have been studied by Frigioni, Narvaez, and many others, and the Dynamic Dijkstra [6] and Reliable Dynamic Shortest Path (RDSP) [7] are two of the resulting dynamic routing algorithms. In dynamic routing algorithms, more computation time is required for one node than the static routing algorithms; however, the dynamic algorithms can reduce the computation time as they decrease the number of nodes that must be computed. That is, when some links have new weights in a network, the dynamic routing algorithms locate only the affected nodes using the changed links in order to re-compute the shortest paths. Therefore, the dynamic algorithms require less computation time for the shortest paths than that required by the static routing algorithms.

However, the dynamic routing algorithms require greater computation time depending on the location of the changed links.

Hybrid routing algorithms, which require less computation time, combine the advantages of the static routing algorithms with those of the dynamic routing algorithms; the hybrid routing algorithms have only recently been proposed. In order to efficiently compute the shortest paths using the Hybrid Shortest Path Tree (HSPT), the times when the static and dynamic algorithms are applied are very important.

In this paper, a multi-path hybrid routing algorithm is presented and it uses the multi-path information to create the shortest path tree when some links have new weights. In order to evaluate the proposed algorithm, it is compared with the well-known static Dijkstra, dynamic Dijkstra, and HSPT algorithms.

2. Problem

The Dijkstra algorithm, which is a static routing algorithm, is widely used in network routing. However, this well-studied static routing algorithm is very ineffective when the link status changes in a network and only a small section of the old shortest paths requires the update. This is because it re-computes the new shortest paths in their entirety, without using the old shortest paths information. This is the hardest operation for a router; it occupies requires significant CPU resources. It requires a very long time to compute the shortest paths using the Dijkstra algorithm; it also incurs a gap phase. In the gap phase, the routing table does not reflect the actual network topology until it completes the computation; thus, packet losses can occur, which is problematic [8-10].

For these reasons, dynamic routing algorithms are preferred over a static routing algorithms because they can reduce the computation time by removing the majority of the information from the old routing table and only re-computing the nodes affected by the updated shortest paths list. However, dynamic routing algorithms require greater computation times than static algorithms, and these times depend on the location of the changed links. That is, when some links have new weights near the root node, it requires a long computation time because the updated shortest paths affect a lot of nodes. Inversely, if some links near the end node have a new weight, then the computation time is shorter due to fewer nodes being affected. Therefore, dynamic routing algorithms do not have consistent results because they have wide variations for each computation time case. This is problematic in network routing.

Therefore, hybrid routing algorithms have been proposed that use the advantages of the static routing algorithms and dynamic routing algorithms. Hybrid routing algorithms are more efficient than static and dynamic algorithms; however, if hybrid routing algorithms use multi-path information, their performance improves more. That is, the multi-path hybrid routing algorithm has less computation time than that of hybrid routing algorithms.

3. Proposed Algorithm

In this paper, a multi-path hybrid routing algorithm is proposed that reduces the total shortest path tree (SPT) execution time using the multi-path information; the proposed algorithm is called the Multi-Path Hybrid Shortest Path Tree (MP-HSPT) and is based on the HSPT. In order to efficiently compute the shortest paths using the HSPT, the times when the static and dynamic algorithms are applied are very important. The static routing algorithms should be applied when computing the shortest paths where some links have new weights near the root node; static routing algorithms are applied in this situation because there are many nodes that must be computed near the root node. In this case, using static routing algorithms is a faster method for computing the shortest paths rather than using the dynamic routing algorithms, which require more computation time for each node. Alternatively, dynamic routing algorithms should be used to compute the shortest paths when some links have new weights near the end node; they are applied in this case because there are only a few nodes that must be computed near the end node. In this case, using dynamic routing algorithms to only re-compute the nodes affected by the old shortest paths is faster than using the static routing algorithms that re-compute every node. In addition, the MP-HSPT uses multi-path information. If other minimum weight paths are found, the algorithm uses the multi-path information to reduce the computation time. If other paths of minimum cost are not found, the algorithm reduces the number of affected nodes using the multi-path information.

```
Step 1: Use multi-path information
Find the multi-path by using the multi-path information
if multi-path found then
                Update the routing table
        else
                Go to Step 2
        endif
Step 2: Find the depth of whole network
Find the network depth by Depth First Search (DFS)
Determine D as 40% depth of whole network
for All nodes do
    Find the link whose link cost is changed
    Calculate the link depth by DFS
        if the link depth < D then
                Go to Step 3
        else
                Go to Step 4
        endif
endfor
Step 3: Static routing
Find shortest paths using static routing
Update the routing table
Step 4: Dynamic routing
Find shortest paths using dynamic routing
Update the routing table
```

Figure 1. Pseudo Code for the Proposed MP-HSPT

The multi-path hybrid routing algorithm is performed using the following procedures. First, the multi-path is found using the multi-path information. If a multi-path is found, it is included in the shortest path. If a multi-path is not found, the hybrid routing algorithm is applied. Second, in order to decide which algorithms should be applied, the depth of whole network is found using the Depth First Search (DFS) method. Third, when links with new weights and a depth of less than 40% are found, the static routing algorithms are applied to compute the shortest paths. Finally, when some links with new depth weights of more than 40% are found, the dynamic routing algorithms are applied to compute the shortest paths. The pseudo code for the proposed MP-HSPT is presented in Figure 1.

4. Performance Evaluation

The performance of the MP-HSPT is compared with previously published algorithms: the Dijkstra, Dynamic Dijkstra, and HSPT methods. The number of nodes, changed rate of link weights, and deviation of link weights were used as the input parameters in the simulations. The input parameters are presented in Table 1.

Table 1. Input Parameters

Parameters	Values
Number of nodes	50, 100, 150, 200
Changed rate of link weights (%)	100, 200, 300, 400
Deviation of link weights	5, 10, 15, 20

The number of nodes represents the number of nodes in the entire network, i.e. the network size. The changed rate for the link weights is the rate used to compare the new link weight to the old link weight. The deviation of the link weights is the difference between the link weights and the mean value of the link weights. In this study, the mean value is set to 10. The comparisons were undertaken based on the total computation time, which is the computation time of the shortest path in the given network.

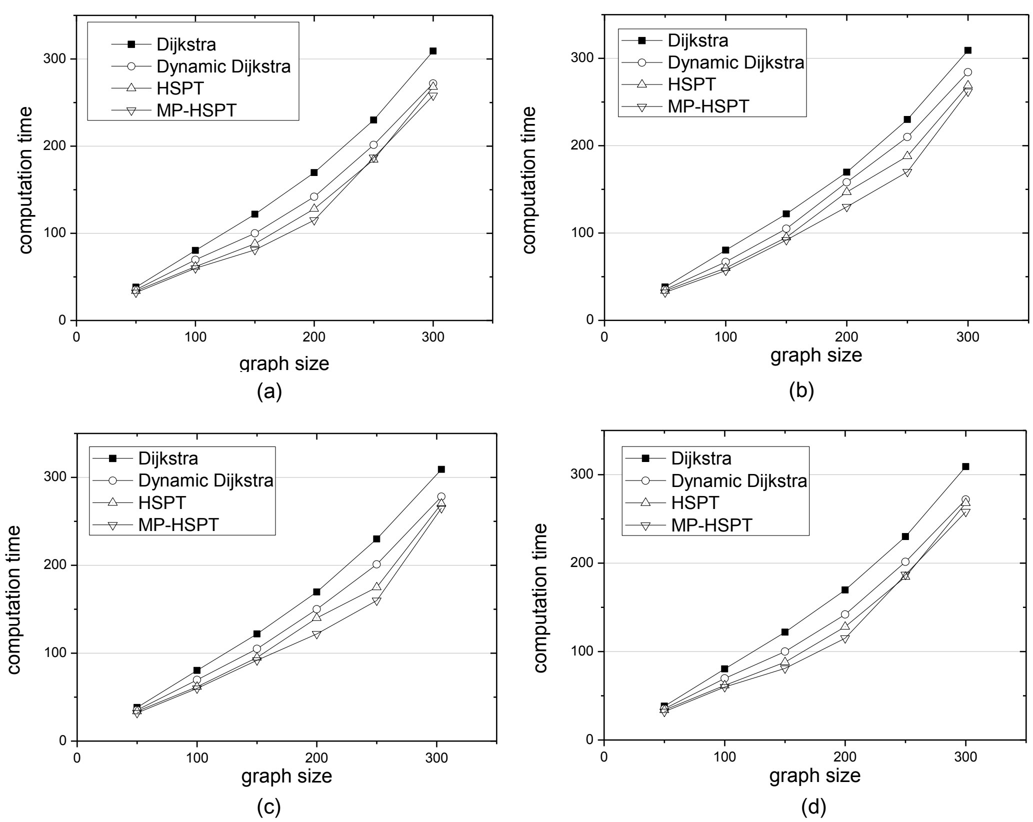

Figure 2. Total Computation Time with Respect to the Number of Nodes and the Changed Rate of Link Weights

Figure 2 shows the total computation time with respect to the number of nodes and changed rate of link weights. The changed rate of link weights of Figure 2(a) is 100%, Figure 2(b) is 200%, Figure 2(c) is 300%, and Figure 2(d) is 400%. As the number of nodes increased, the computation time for shortest paths required a longer time. If the changed rate of link weights was low, then the computation time for the shortest paths increased linearly. However, if the changed rate of link weights was high, then the computation time for shortest paths increased nonlinearly.

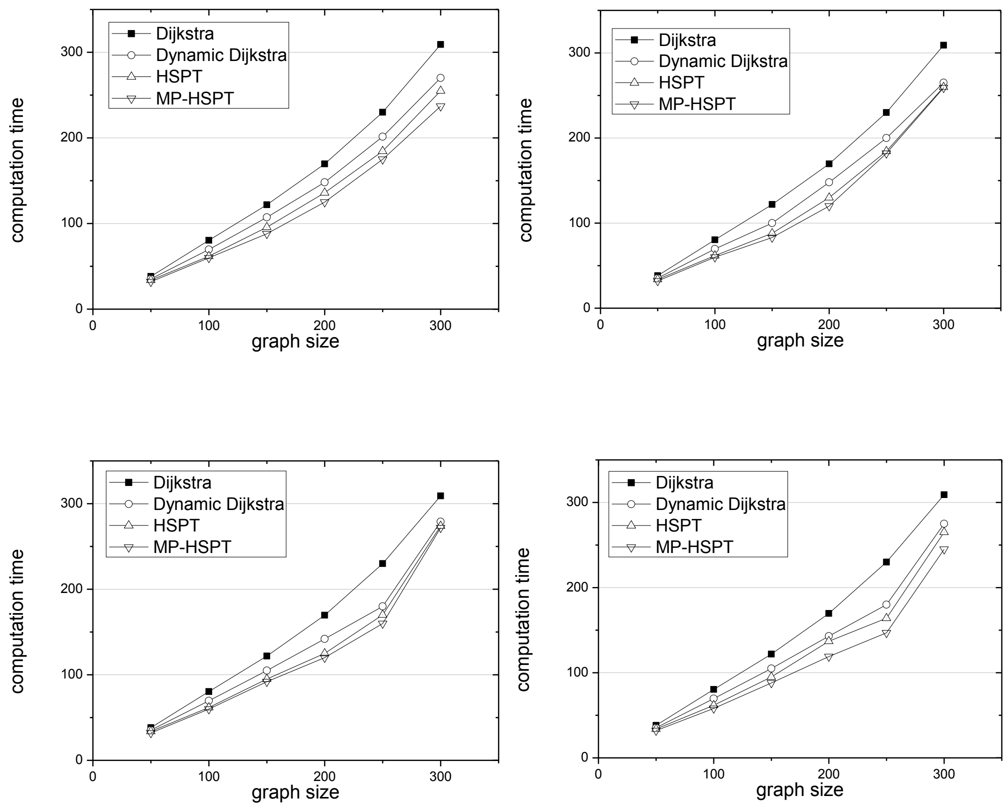

Figure 3. Total Computation Time with Respect to the Number of Nodes and Deviation of Link Weights

Figure 3 shows the total computation time with respect to the number of nodes and the deviation of link weights. The deviation of link weights of Figure 3(a) is 5, Figure 3(b) is 10, Figure 3(c) is 15, and Figure 3(d) is 20; the mean value was 10. As shown in the figures, the MP-HSPT outperforms the Dijkstra, Dynamic Dijkstra, and HSPT algorithms. As the number of nodes increased, the computation time for the shortest paths increased.

5. Conclusion

In this paper, the Multi-Path Hybrid Shortest Path Tree (MP-HSPT) algorithm was presented and it offers an efficient shortest path decision that can be used to reduce the total execution time using the multi-path information. The decreased total execution time also leads to reductions in packet losses. As shown in the comparison results, the proposed MP-HSPT algorithm provides better performance when compared with the Dijkstra, Dynamic Dijkstra, and HSPT methods in terms of computation time for the shortest path.

Acknowledgements

This work was supported by the Key Research Institute Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (grant no. 2011-0018394).

References

[1] E. Dijkstra, "A note two problems in connection with graphs", Numerical Math, vol. 1, **(1959)**, pp. 269-271.
[2] G. S. Cho and J. K. Ryeu, "An Efficient Method to Find a Shortest Path for a Car-Like Robot", International Journal of Multimedia and Ubiquitous Engineering, vol. 1, no. 1, **(2006)**, pp. 1-6.
[3] P. C. Newton, L. Arockiam and T. H. Kim, "An Efficient Hybrid Path Selection Algorithm for an Integrated Network Environment", International Journal of Database Theory and Application, vol. 2, no. 1, **(2009)**, pp. 31-38.
[4] B. Xiao, J. Cao, Z. Shao and E. H. M. Sha, "An Efficient Algorithm for Dynamic Shortest Path Tree Update in Network Routing", Journal of Communication and Networks, vol. 9, **(2007)**, pp. 499-510.
[5] S. Y. Ameen, A. A. Ahmed and I. A. Ibrahimi, "MANET Routing Protocols Performance Evaluation with TCP Taho, Reno and New-Reno", International Journal of u- and e-Service, Science and Technology, vol. 4, no. 1, **(2011)**, pp. 37-49.
[6] E. P. F. Chan and Y. Yang, "Shortest Path Tree Computation in Dynamic Graphs", IEEE Transactions on computers, vol. 58, **(2009)**, pp. 541-557.
[7] T. H. Cho, J. W. Kim, B. J. Kim, W. O. Yoon and S. B. Choi, "A Study on Shortest Path Decision Algorithm for Improving the Reliability of Dynamic Routing Algorithm", Journal of the Korean Institute of Information Scientists and Engineers, vol. 38, **(2011)**, pp. 450-459.
[8] V. Eramo, M. Listanti and A. Cianfrani, "Design and Evaluation of a New Multi-Path Incremental Routing Algorithm on Software Routers", IEEE Transactions on Network and service management, vol. 5, **(2008)**, pp. 188-203.
[9] S. K. Lee, J. W. Jang, S. J. Jang and J. Y. Shin, "Development And Performance Analysis of A BR-DBA Algorithm For Improve Network Performance", International Journal of Future Generation Communication and Networking, vol. 1, no. 1, **(2008)**, pp. 1-6.
[10] B. Jedari, R. Goudarzi and M. Dehghan, "Efficient Routing using Partitive Clustering Algorithms in Ferry-based Delay Tolerant Networks", International Journal of Future Generation Communication and Networking, vol. 2, no. 4, **(2009)**, pp. 1-14.

Authors

Taehwan Cho

He earned the B.S. degree in aerospace engineering from Inha University, Incheon, Korea, in 2001. He is currently working toward the Ph.D. degree in the computer architecture and networks Laboratory. His research interests include communication and security in computer networks, wireless mobile ad hoc and sensor networks.

Sangbang Choi

He earned the M.S. and Ph.D. in electrical engineering from the University of Washington, Seattle, in 1988 and 1990, respectively. He is currently a professor of electronic engineering at Inha University, Incheon, Korea. His research interests include computer architecture, computer networks, wireless communication, parallel and distributed systems.