Performance Improvement on MPLS On-line Routing Algorithm for Dynamic Unbalanced Traffic Load

Wisitsak Sa-Ngiamsak* Ekanun Sombatsakulkit** and Ruttikorn Varakulsiripunth*

* Department of Electrical Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand
(Tel : +66-2-737-3000 ext 3332; E-mail: wisitsaks@hotmail.com, kvruttik@kmitl.ac.th)
** Department of Electronic Engineering, North Eastern University, Khon Kaen, Thailand
(Tel : +66-4-322-2959 ext 219; E-mail: ekanun@neu.ac.th)

Abstract: This paper presents a constrained-based routing (CBR) algorithm called, Dynamic Possible Path per Link (D-PPL) routing algorithm, for MultiProtocol Label Switching (MPLS) networks. In MPLS on-line routing, future traffics are unknown and network resource is limited. Therefore many routing algorithms such as Minimum Hop Algorithm (MHA), Widest Shortest Path (WSP), Dynamic Link Weight (DLW), Minimum Interference Routing Algorithm (MIRA), Profiled-Based Routing (PBR), Possible Path per Link (PPL) and Residual bandwidth integrated - Possible Path per Link (R-PPL) are proposed in order to improve network throughput and reduce rejection probability. MIRA is the first algorithm that introduces interference level avoidance between source-destination node pairs by integrating topology information or address of source-destination node pairs into the routing calculation. From its results, MIRA improves lower rejection probability performance. Nevertheless, MIRA suffer from its high routing complexity which could be considered as NP-Complete problem. In PBR, complexity of on-line routing is reduced comparing to those of MIRA, because link weights are off-line calculated by statistical profile of history traffics. However, because of dynamic of traffic nature, PBR maybe unsuitable for MPLS on-line routing. Also, both PPL and R-PPL routing algorithm we formerly proposed, are algorithms that achieve reduction of interference level among source-destination node pairs, rejection probability and routing complexity. Again, those previously proposed algorithms do not take into account the dynamic nature of traffic load. In fact, future traffics are unknown, but, amount of previous traffic over link can be measured. Therefore, this is the motivation of our proposed algorithm, the D-PPL. The D-PPL algorithm is improved based on the R-PPL routing algorithm by integrating traffic-per-link parameters. The parameters are periodically updated and are dynamically changed depended on current incoming traffic. The D-PPL tries to reserve residual bandwidth to service future request by avoid routing through those high traffic-per-link parameters. We have developed extensive MATLAB simulator to evaluate performance of the D-PPL. From simulation results, the D-PPL improves performance of MPLS on-line routing in terms of rejection probability and total throughput.

Keywords: Constrained-Based Routing, On-line Routing, QoS Routing, Bandwidth Guaranteed Tunnel, and MPLS Network,

1. INTRODUCTION

MultiProtocol Label Switching (MPLS) network is an efficient solution for QoS-awareness network. It provides capabilities of constrained-based routing (CBR), therefore, QoS guaranteed Label Switching Paths (LSPs) are attained. This improves the Internet traffic which have only one traffic class, called best effort class. Furthermore, the MPLS network supports frameworks of Traffic Engineering (TE), QoS routing and Virtual Private Network (VPN). Hence, the MPLS network is getting more interesting and it is now standardized by the Internet Engineering Task Force (IETF) [1]. Furthermore, the MPLS technology is extended and modified to be Generalized MPLS (GMPLS) which can support more general switching types other than ATM switching such as Packet Switch Capable (PSC), Layer-2 Switch Capable (L2SC), Time-Division Multiplex Capable (TDM), Lambda (l) Switch Capable (LSC), and Fiber-Switch Capable (FSC) interfaces [2].

In this paper, constrained-based on-line routing of bandwidth guaranteed tunnels over MPLS networks is investigated. With on-line routing, future demands are unknown. Therefore, MPLS on-line routing should route an LSP which is able to achieves following objectives.

1. Select an LSP which is able to guarantee the demand bandwidth of the requesting traffic.
2. Use knowledge of ingress-egress points of LSPs in order to avoid interference to other ingress-egress pairs.
3. Utilize minimal network resources in order to have enough bandwidth to service future demands.
4. Reduce rejection probability and improve total throughput.

According to the first objective, a few works is done on routing in dynamic unbalanced traffic load. Hence, to achieve the objectives, we proposed a constrained-based routing algorithm called, Dynamic Possible Path per Link (D-PPL) routing algorithm. With the D-PPL algorithm, amount of dynamic incoming traffic over each link are periodically measured. The D-PPL tries to reserve residual bandwidth to service future request by avoid routing through those high traffic-per-link parameters. Therefore, from simulation results, the proposed algorithm can improve performance of MPLS on-line routing in terms of rejection probability and total throughput of traffic under dynamic traffic load.

The remainder of the paper is organized as followings. Section 2 presents related works for MPLS on-line routing algorithm. Section 3 describes details of the proposed routing algorithm. Simulation results are shown in Section 4. Finally, Section 5 concludes the paper.

2. RELATED WORKS

Much works have been done for the on-line routing problem in MPLS networks. Those works have their own pros and cons which will be described as following.

The simplest and most popular algorithm to route guaranteed demand is Min Hop Algorithm (MHA) [3]. In this scheme a minimum hop path which satisfies the traffic demand requirement is chosen. It is simple and uses the least network resource. However, it has the drawback of always using up some links, while other paths maybe under-used. This increases the chance of not being able to service new demands which may require traversing those bottleneck links.

Widest Shortest Path (WSP) algorithm [4] is an improvement over the MHA algorithm, where the shortest path with the largest available bandwidth is chosen. This will enable load balancing between the equal hop count paths for different traffic request. However, it still chooses to use up all available capacity on particular shortest paths before longer paths with more capacity are under-utilized.

Minimum Interference Routing Algorithm (MIRA) [5], uses the knowledge of ingress-egress label switching router that are potential traffic source-destination pairs. It makes the routing decision for a demand based on the "interference" level in would have on the future demand from other source-destination. This interference level is used as the link weight to calculate the shortest path for a new demand. The novel of this algorithm results the less chosen the critical links to other source-destination pairs. However, it has two major drawbacks. The first is complexity to calculate the maximum flow between any source-destination pairs and the link weight of all links. The second, MIRA considers only balance traffic flow of source-destination pairs, while the pair with much higher maximum traffic than other pairs is not considered [6].

Dynamic Link Weight (DLW) [6] algorithm improves two drawbacks of the MIRA algorithm. The algorithm is much simple comparing to the MIRA algorithm. With DLW algorithm, link weights are calculated based on residual bandwidth of corresponding links. After that, an LSP with minimum path weight is selected. So, an non shortest LSP with the highest residual bandwidth is utilized and load balancing is enabled. However, interference level between source-destination pairs are not taken into account.

Profile Based Routing (PBR) [7] is a routing algorithm that improves complexity of the MIRA algorithm. Link weights are off-line calculated by statistical profile of history traffics. Each source-destination pairs have their own profiles of link weights. So, the scheme is much less complexity than the MIRA algorithm. However, because of nature, PBR maybe unsuitable for MPLS on-line routing over dynamic unbalanced traffic load.

Possible Path per Link (PPL) routing algorithm [8] is developed based on link weights calculated by possible paths of all source-destination pairs. The link with high link weight value is considered as the link with high bottleneck (or interference) probability. this algorithm avoids selecting of those links with high link weights value. So, the rejection probability is reduced. Again, this algorithm does not take into account the dynamic traffic load.

Residual bandwidth integrated - Possible Path per Link (R-PPL) routing algorithm [9] is an extend version of the PPL algorithm. It combines the conventional PPL link weight with current residual bandwidth of the corresponding link. As a result, the R-PPL accomplishes better performance in terms of rejection probability and total throughput. Also, the algorithm does not take into account the dynamic nature of traffic load.

3. DYNAMIC – POSSIBLE PATH PER LINK (D-PPL) ROUTING ALGORITHM

3.1 Designing Objectives

As mention in previous the previously proposed algorithms do not take into account scenario of dynamic unbalanced traffic. So, the first designing objective is to improve performance of routing algorithm under dynamic unbalanced traffic load condition. The second objective is to maintain all basic objectives described in the section introduction.

In order to achieve the first objective, probability of traffic over links are measured and are periodically updated related to the update interval of the link state protocol. Let $G(N,L,R)$ is network graph. Let N, L, R denote to set of nodes, links and residual bandwidth of the links, respectively. Let P denotes to set of ingress-egress nodes pair. From Eq.(1), the probability is evaluated by total traffic over link l divided by total input traffic over all links. At initial state, total traffic over all links are set to 1 (see Eq.(2)). Further, Eq.(3) ensures that total probability is equal to 1.

$$p^{(l)} = \frac{T^{(l)}}{\sum T^{(j)}}, \quad l \in L,\ \forall(j) \in L \qquad (1)$$

$$T^{(l)} = 1, \qquad \forall(l) \in L,\ at\ initial\ state \qquad (2)$$

$$\sum p^{(a,b)} = 1, \qquad \forall(a,b) \in P \qquad (3)$$

where $p^{(l)}$: probability of traffic over link l

$T^{(l)}$: total traffic over link l in previous updated interval.

In order to achieve the second objective, the D-PPL is designed based on our previous routing algorithm, R-PPL [9]. Then, the input traffic probability calculated by Eq.(1) are integrated to the R-PPL algorithm which will be described in the next subsection.

3.2 Calculation of D-PPL Link Weight

In order to calculate the D-PPL link weight, PPL and R-PPL should be determined first. Fig.1 shows a procedure of PPL link weight calculation which is also based on Eq.(4). Note that the procedure is required to operate whenever the network topology has been changed because of administrator purposes or node/link failures. It is the off-line procedure. Therefore, the PPL weights are rarely updated. This shows that the algorithm complexity is in low level and the algorithm will require a short CPU calculation time over on-line routing. Moreover, the high PPL value indicates the high probability of bottleneck/congestion over the corresponding link. So, the routing algorithm should avoid selecting of those links with high PPL link weight value.

```
Procedure PPL_Calculation
{
1. Set $PPL^{(l)} = 1 \quad \forall l \in L$
2. For $a$ = 1 to Total_Edge_Node
    For $b$ = 1 to Total_Edge_Node
      If a is not equal b
        2.1 List all possible disjoint paths $DP$ routed
            from ingress node $a$ to egress node $b$
        2.2 $PPL^{(l)} = PPL^{(l)} + 1 \qquad \forall l \in DP$
      End If
    Next $b$
  Next $a$
3. Calculate $w_{PPL}^{(l)}$ according to Eq.(3).
}
```

Fig. 1 The PPL_Calculation procedure

On the other hand, R-PPL link weight is calculated on-line (see Eq.(5)). The weight value is varied by residual bandwidth of the corresponding link, $R^{(l)}$. Therefore, the R-PPL weight should be on-line updated or every time that a new request traffic arrives. Similar to the PPL weight, the high R-PPL weight signifies the high bottleneck probability and the low residual bandwidth of the corresponding link.

$$w_{PPL}^{(l)} = \frac{PPL^{(l)}}{\sum PPL^{(j)}}, \qquad l \in L, \ \forall (j) \in L \qquad (4)$$

$$w_{R-PPL}^{(l)} = \frac{w_{PPL}^{(l)}}{R^{(l)}}, \qquad l \in L \qquad (5)$$

$$w_{D-PPL}^{(l)} = w_{R-PPL}^{(l)} \times p^{(l)}, \qquad l \in L \qquad (6)$$

where $w_{PPL}^{(l)}$: *PPL* weight of link *l*

$w_{R-PPL}^{(l)}$: *R-PPL* weight of link *l*

$w_{D-PPL}^{(l)}$: *D-PPL* weight of link *l*

$PPL^{(l)}$: total possible path per link over link *l*

$R^{(l)}$: current residual bandwidth of link *l*

Based on the R-PPL link weight and the traffic-per-link probability, D-PPL link weights could be calculated by Eq.(6). This means that links with high traffic-per-link probability will be avoid. As a result, more resources on those links with high traffic-per-link probability are reserved for future request. Furthermore, the rejection probability of the network is then reduced.

3.3 D-PPL Routing Procedures

In this section, details of D-PPL on-line routing are presented. The procedure of the algorithm is shown in Fig.2. And, in Fig.3, the update procedure of traffic-over-link is demonstrated.

```
Procedure MPLS_Online_Routing
{
1. Reduce network graph by eliminate all links that have residual bandwidth less than request bandwidth.
2. Use Dijkstra algorithm to find a path from ingress node a to egress node b that has the minimum path weight according to Eq.(7).
3. Establish the selected path found in Step 3.
4. If no path is selected, the algorithm fails.
5. Update $T^{(l)}$ .
}
```

Fig. 2 The MPLS_Online_Routing procedure

$$Minimize \quad w_{Path}^{(a,b)}$$

subjected to

$$w_{Path}^{(a,b)} = \sum w_{D-PPL}^{(l)}, \qquad \forall (l) \in L^{(a,b)} \qquad (7)$$

where $w_{Path}^{(a,b)}$: Path weight from node *a* to *b*

$L^{(a,b)}$: set of links routed from node *a* to *b*

```
Procedure Traffic_Over_Link_Probability_Update
{
1. Update $p^{(l)}$ for all link $l \in L$ using Eq.(1).
2. Set $T^{(l)}$ back to initial state according to Eq.(2).
}
```

Fig. 3 The Traffic_Over_Link_Probability_Update procedure

4. SIMULATION RESULTS

4.1 Simulation Models

An experimental network, shown in Fig. 4, is chosen for performance evaluation in terms of rejection probability and total throughput. It composes of 15 nodes and 28 links. It has ten edge nodes which are node 1, 2, 4, 5, 8, 9, 12, 13, 14 and 15. According to link characteristics, all links are bi-directional. Capacity of links are 100 Mbps. Size of traffic requests are 1-5 Mbps varied by uniformly distribution. For more accuracy, we use the same orders of traffic requests to test on different routing algorithms. Ingress and egress nodes (or source- destination node pairs) of each traffic request are also populated by uniformly distribution. Furthermore, update interval of the traffic-over- link probability is set 1 second.

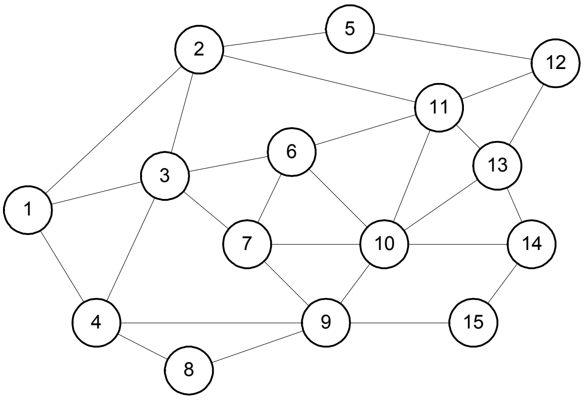
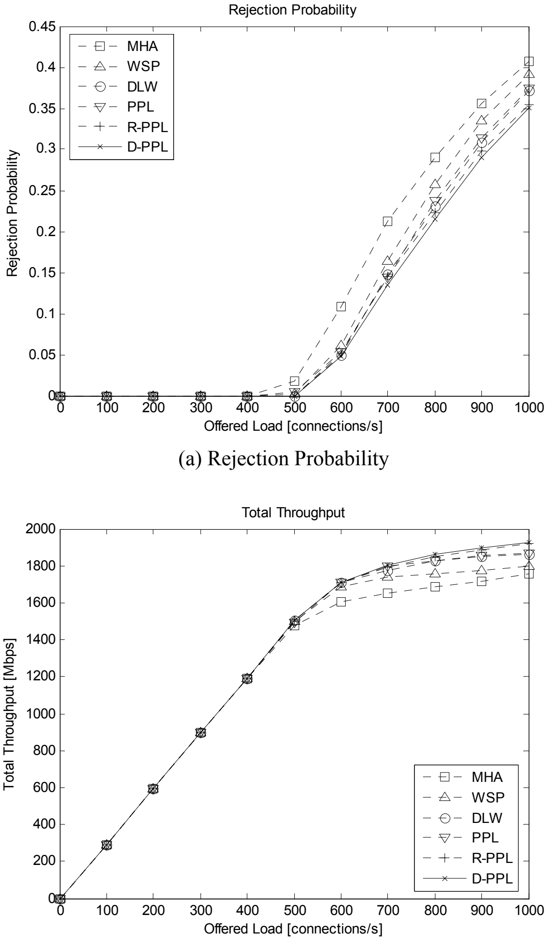


Fig. 4 Experimental network with 15 nodes.

Furthermore, simulation tests are composed of two traffic load scenarios: Balanced and Unbalanced traffic load. With the first scenario, all ten edged nodes. However, in unbalanced traffic load scenario, request traffics are inputted and outputted to only five edge nodes which are set of edge nodes {1, 2, 3, 5, 15}. Note that, for more accuracy, we repeat ten times simulation test for each scenarios.

4.2 Simulation Results of Balanced Load Scenarios

In balanced traffic load scenario, there are request traffics evenly coming to all edge nodes. From Fig.5, the rejection probability of the D-PPL algorithm performs the lowest level. It has a little bit lower than the R-PPL algorithm. Moreover, the total throughput of the D-PPL algorithm are also at the highest level. Again, it has a little bit higher than the R-PPL algorithm. Those results show that the D-PPL algorithm has the best performance under the balanced traffic load scenario. Besides, the MHA algorithm performs the worst results on both rejection probability and total throughput.



(a) Rejection Probability

(b) Total Throughput

Fig. 5 Simulation results under balanced traffic load.

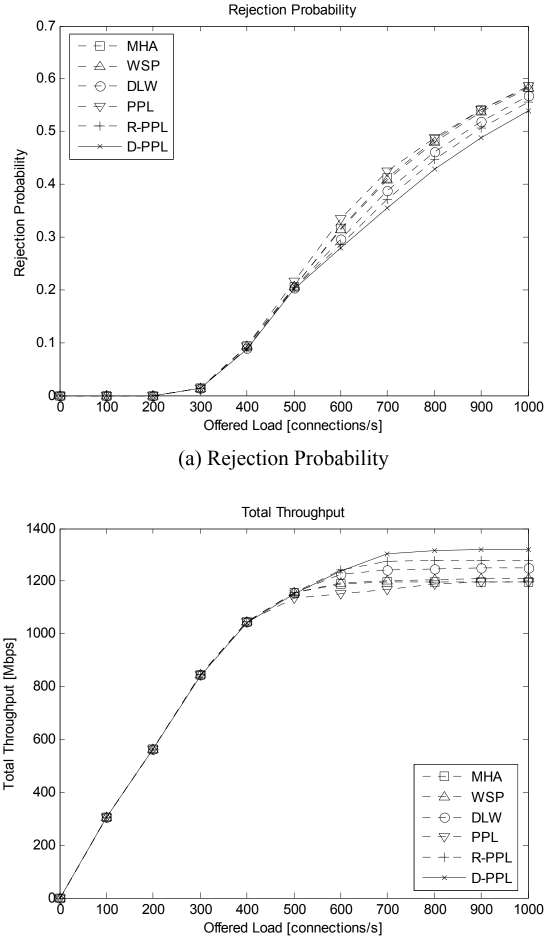
4.3 Simulation Results of Unbalanced Load Scenarios

In unbalanced traffic load scenario, incoming request traffics are inputted only on five selected edge nodes. In Fig.6, it is clearly shown that the D-PPL algorithm has the best performance on both rejection probability and total throughput. On the other hand, the PPL algorithm, our first version of routing algorithm, suffers from the worst performance. The reason is that the PPL algorithm is based link weights that evaluated from standard balanced traffic load. So, the possible-path-per-link probability of link will be unsuited for real incoming traffic.

5. CONCLUSIONS

This paper presents a on-line routing algorithm called, Dynamic Possible Path per Link (D-PPL) routing algorithm, for MultiProtocol Label Switching (MPLS) networks. In MPLS on-line routing, future traffics are unknown and network resource is limited. Therefore many routing algorithms such as Minimum Hop Algorithm (MHA), Widest Shortest Path (WSP), Dynamic Link Weight (DLW), Minimum Interference Routing Algorithm (MIRA), Profiled-Based Routing (PBR), Possible Path per Link (PPL) and Residual bandwidth integrated - Possible Path per Link (R-PPL) are proposed in order to improve network throughput and reduce rejection probability. Both PPL and R-PPL routing algorithm we formerly proposed, are algorithms that achieve reduction of interference level among source-destination node pairs, rejection probability and routing complexity. However, those previously proposed algorithms do not take into account the dynamic nature of traffic load. In fact, future traffics are unknown, but, amount of traffic over link can be measured. Therefore, this is the motivation of our proposed algorithm, the D-PPL. The D-PPL algorithm is improved based on the R-PPL routing algorithm by integrating traffic-over-link probabilities. The probabilities are periodically updated and are dynamically changed depended on current incoming traffic. The D-PPL algorithm tries to reserve residual bandwidth to service future request routed through those links with high traffic-per-link probability. Further, we have developed extensive MATLAB simulator to evaluate performance of the D-PPL algorithm. From simulation results, the D-PPL improves performance of MPLS on-line routing in terms of rejection probability and total throughput.

Our future works are pointed out to issues of supporting fairness and multiple traffic classes such as Differentiated Service model. Furthermore, an effect of variation of update interval of traffic-over-link probability will also be taken into account.



(a) Rejection Probability

(b) Total Throughput

Fig. 6 Simulation results under unbalanced traffic load.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the contribution of the Research Center for Communications and Information Technology (ReCCIT), Bangkok, Thailand and the Faculty of Engineering, Department of Electrical Engineering, North Eastern University, Khon Kaen, Thailand.

REFERENCES

[1] E. Rosen, A. Viswanathan, and R. Callon, Multiprotocol Label Switching Architecture," *RFC3031*, January 2001.

[2] E. Mannie et al., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture," *RFC3945*, 2004.

[3] D.O. Awduche, J. Malcom, J. Agogbua, M. O'D`ell, and J. McManus, "Requirements for traffic engineering over MPLS,"RFC2702, 1999.

[4] R. A. Guerin et al, "QoS routing mechanisms and OSPF extensions," *In Proceedings of IEEE GlobeCom*, 1997.

[5] M. Kodialam and T. V. Lakshman, "Minimum Interference Routing with Applications to MPLS Traffic Engineering," *In Proceedings of IEEE INFOCOM*, 2000.

[6] H. Pham and B. Lavery, "New Dynamic Link Weight Algorithm for On-Line Calculations of LSPs in MPLS Networks," *In Proceedings of IEEE ICCS*, 2002.

[7] Subhash Suri, Marcel Waldvogel, Daniel Bauer, and Priyank Ramesh Warkhede, "Profile-Based Routing and Traffic Engineering," *Elsevier Science Computer Communication*, Vol. 25, 2002.

[8] W. Sa-Ngiamsak, S. Thipchaksurat, and R. Varakul-siripunth, "A Bandwidth-Based Constraint Routing Algorithm for Multi-Protocol Label Switching Networks," *In Proceedings of ICACT*, 2004.

[9] W. Sa-Ngiamsak and R. Varakulsiripunth, "A Possible Path per Link CBR Algorithm for Interference Avoidance in MPLS Networks," *In Proceedings of ICCAS,* 2004.