Relay-Node Based Proactive Load Balancing Method in MPLS Network with Service Differentiation

Yukitaka Yoshida  
College of Media Arts, Science and Technology  
University of Tsukuba  
Tsukuba, Japan  
s0811559@u.tsukuba.ac.jp

Masatoshi Kawarasaki  
Graduate School of Library, Information and Media Studies  
University of Tsukuba  
Tsukuba, Japan  
mkawa@slis.tsukuba.ac.jp

Abstract—If IP network only uses conventional routing protocols such as open shortest path first (OSPF), some portions of the network become congested while other portions remain underutilized depending on the variations of traffic demand. MPLS that provides explicit routing along pre-determined path can improve load balancing throughout the network, thus achieves effective network resource utilization. In this paper, we introduce the overlay path that passes through strategically placed relay-node. By selecting a suitable relay-node dynamically, the network can proactively keep pace with changing traffic demand. We show that relay-node based load balancing can achieve almost equivalent network performance as global optimization by Linear Programming, while reallocating only 20% of total traffic. Furthermore, by selecting two kinds of path, namely primary path as the shortest path and overlay path as a detour path, in reference to VPN label, service differentiation is achieved, thus realize user-centric network.

Keywords—Load balancing; Relay-node; Linear Programming; Shortest path; MPLS; MPLS-VPN; Service differentiation

I. INTRODUCTION

Owing to wide spread broadband internet environment and emerging various applications, internet traffic is expanding rapidly. To meet increasing traffic demand while achieving required quality of service (QoS) levels, traffic engineering of IP network is actively studied. The objective is to maximize the network resource utilization and minimize the number of denied connections due to insufficient resource availability. Only with conventional routing protocols such as open shortest path first (OSPF), some portions of the network become congested while other portions remain underutilized depending on the variations of traffic demand. Effective traffic engineering is needed to optimize network resource utilization and performance.

For this purpose, Multi-Protocol Label Switching (MPLS) has been developed and deployed in many ISPs. Different from OSPF routing, MPLS network can provide explicit routing through Label Switched Paths (LSPs). Bandwidth of each LSP can also be pre-determined. Using these features, MPLS network can improve load balancing throughout the network, thus achieves effective network resource utilization.

Traffic engineering using MPLS has been actively studied ([1]-[6]). In [1], global optimization using Linear Programming is performed periodically to meet time varying traffic demand, under the constraint that only several large capacity paths are subject for optimization. Although frequent path rearrangement is avoided by this constraint, the algorithm cannot cope with sudden change in traffic demand or deviated traffic demand distribution. In [2], traffic is carried firstly by the primary path that is the shortest path. If the primary path is congested, the excessive traffic is carried by the secondary path that is a detour path. In this algorithm, however, as secondary path is selected randomly, it may be an ineffective path. Furthermore, as only one secondary path is prepared, if it is congested further traffic cannot be accepted. In [3], traffic engineering with priority scheme is discussed. By introducing the concept of TE-Class, when the network is congested, bandwidth of low-priority paths is reassigned to the high-priority paths. The traffic being carried by the preempted low-priority path is transported by other detour paths. However, the method of finding detour paths is not discussed. In [4], an adaptive load balancing between multiple paths is proposed focusing at TCP performance. By counting the number of active TCP flows at the source node, the algorithm gives the method of load balancing of the whole network. However, the way of establishing paths is not discussed and UDP traffic is not considered.

This paper proposes a proactive load balancing method in MPLS network with service differentiation. The proposed algorithm maintains maximum network performance in time-varying traffic demand while achieving multi-level QoS. Between a pair of source and destination nodes, we establish a primary path as the shortest path and multiple overlay paths as detour paths that pass through respective relay-node which are strategically determined by our algorithm. By sorting traffic adequately in multiple paths, effective load balancing is achieved. Furthermore, as MPLS by itself cannot provide service differentiation, we compliment it by using MPLS-VPN technology. We use the VPN-label for the segregation of delay sensitive traffic and throughput sensitive traffic, so that respective traffic is carried by appropriate path that meets each QoS requirement.

The rest of this paper is structured as follows: In Section II, we propose the relay-node based load balancing algorithm with mathematical formulation. Section III describes network model and traffic model for the evaluation of proposed algorithm.
Section IV describes the result and section V concludes this paper.

II. RELAY-NODE BASED LOAD BALANCING METHOD

A. Primary path and Overlay path

We define two kinds of path between every source and destination (SD) node pair. One path is the primary path that has the shortest route defined by IGP protocol such as OSPF. The other path is the overlay path that passes through a strategically placed relay-node (Fig. 1). All nodes except for source and destination nodes can be a candidate of a relay-node. In a given overlay path, path route from source node to relay-node and path route from relay-node to destination node are the shortest routes defined by IGP, respectively. By establishing multiple overlay paths and carrying packet streams both through the primary and overlay paths, we explore the optimization of bandwidth management against traffic variation. The main issue in our optimization algorithm is how to determine a relay-node for overlay path.

Primary path and overlay path are different not only in their routes from source to destination but also in their characteristics. As the route of overlay path may be changed at every optimization calculation, traffic transported by the path needs to change their route. As a result, packets may be lost and retransmitted. On the other hand, as the route of primary path is unchanged, the primary path can provide more stable packet transport than overlay path. Furthermore, as the primary path is the shortest path, packet transfer time is shorter in the primary path than in the overlay path. However, large traffic cannot be accepted owing to bandwidth limitation of the primary path. In our algorithm, we also manage the amount of traffic to be transferred over the primary path and overlay paths according to the change in user traffic demand between SD node pairs. The performance of the algorithm is directly influenced by the traffic ratio between the primary path and overlay paths.

By using these two kinds of path, service differentiation in MPLS network could be achieved. For example, primary path could be used for real-time traffic that is delay sensitive, and overlay path could be used for file transfer traffic that is throughput sensitive.

B. MPLS-VPN and Primary path

In order to allow MPLS to discriminate users (e.g., delay sensitive user and throughput sensitive user), we use MPLS-VPN (Virtual Private Network) [8].

Figure 1. Primary path and Overlay path

Figure 2. MPLS-VPN

MPLS-VPN is IP based VPN using MPLS technology. It is realized by expanding MPLS label stack to include VPN label. As shown in Fig.2, MPLS-VPN is structured by Customer Edge Router (CE), Provider Edge Router (PE) and Provider Router (P). PE and P correspond to LER (Label Edge Router) and LSR (Label Switching Router) in conventional MPLS, respectively. By using VPN label, ISP can provide logically separated VPN networks for respective users over an MPLS network. We use VPN label for user discrimination. PE adds VPN label ‘ds’ for delay sensitive traffic user (say VPN-A in Fig.2) so that MPLS network carry delay sensitive traffic over a primary path and VPN label ‘ts’ for a throughput sensitive traffic user (say VPN-B in Fig.2) so that throughput sensitive traffic is carried by overlay path. With this mechanism, MPLS network discriminates users, thus makes user-centric network.

C. Relay-Node Selection Algorithm

Variants required for relay-node selection are defined in Table I. The number of overlay paths is Omax. Adding the primary path, the total number of paths between a given SD pair is Omax + 1. We should determine the location of Omax relay-nodes for each SD pair. We assume that the location of some relay-nodes can be duplicated. To determine R(i,j) we should solve integer programming. In our case, it becomes 0-1 integer programming which is known to be NP-hard. In order to obtain approximate solution, we use greedy algorithm. As for R(i,j), although we start from arbitrary ‘m’ and determine the next, R(i,j) that was once determined does not change until next optimization calculation is performed. Every decision it makes is the one with the most obvious immediate advantage. Although it does not assure the optimum solution, an approximate solution can be obtained in a polynomial time. Small amount of calculation is an advantage of this greedy algorithm. Local search algorithm can also be envisaged, but we do not adopt it because of its large amount of calculation.

TABLE I. DEFINITIONS OF VARIANTS

<table>
<thead>
<tr>
<th>Variants</th>
<th>Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>C(i,j)</td>
<td>Capacity of the link between Node i and Node j</td>
</tr>
<tr>
<td>A(i,j)</td>
<td>Traffic demand from Node i to Node j</td>
</tr>
<tr>
<td>Ap(i,j)</td>
<td>Priority traffic demand from Node i to Node j</td>
</tr>
<tr>
<td>Omax</td>
<td>Maximum number of Overlay paths</td>
</tr>
<tr>
<td>Phop(i,j)</td>
<td>Number of hops along the primary path between Node i and Node j</td>
</tr>
<tr>
<td>Ohop(m,i,j)</td>
<td>Number of hops along m-th overlay path between Node i and Node j (1≤m≤Omax)</td>
</tr>
<tr>
<td>R(m,i,j)</td>
<td>Relay Node of m-th overlay path between Node i and Node j</td>
</tr>
<tr>
<td>Umax(m,i,j)</td>
<td>Maximum link utilization (MLU) of m-th overlay path between Node i and Node j that passes through R(m,i,j)</td>
</tr>
</tbody>
</table>
For each SD node pair, we should determine $O_{\text{max}}$ of overlap paths. As for heuristic procedure, firstly we determine the first overlap paths for all SD node pairs and then determine all the second overlap paths, and so on. As our algorithm uses greedy algorithm, overlap paths that are calculated at the later stage of heuristic procedure would get an advantage.

R(i,j) is determined to minimize following penalty function:

$$F(R(m,i,j)) = \alpha \cdot \exp(U_{\text{max}}(m,i,j)) + \beta \cdot \exp(\text{Ohop}(m,i,j) - \text{Phop}(i,j))$$  \hspace{1cm} (1)

The right side first term is a penalty about link utilization. R(i,j) is determined to minimize the maximum link utilization along the overlap path. That is, detour route is established so as to minimize the utilization of the bottleneck link. The right side second term is a penalty about the number of router hops that relates to the effectiveness in network resource utilization. If we try to lower the link utilization by taking a detour route, network resources are wasted. Thus, the first and second terms make a trade-off relationship. By the second term, we can restrict ineffective detouring.

D. Relay-Node Based Load Balancing Algorithm

By using the relay-node selection method as described in the previous section, we propose the load balancing algorithm as shown in Fig. 3.

III. EVALUATION MODEL

A. Network Model

For the evaluation of proposed algorithm, we used NSFNET topology as shown in Fig. 4. Node numbers and corresponding city names are shown in the figure. We assumed that all links are bidirectional and have the same capacity of 10Mbps. (C(i,j)=10,000). In this model, the average number of overlay paths is approximately six. Accordingly, $O_{\text{max}}=6$ in this topology.

B. Traffic Matrix

Traffic matrix $A(i,j)$ represents traffic intensity from the source node i to the destination node j. We assume homogenous traffic matrix; $A(i,j)=200$ ($0 \leq (i,j) \leq 13$, $i \neq j$).

For the evaluation of our proposed load balancing method, we use two types of traffic models; stationary traffic model and variable traffic model. In stationary traffic model, $A(i,j)=A_{\text{average}}(i,j)$.

C. Variable Traffic Model

As for variable traffic model, we assume two models; Sine wave model and Square wave model.

Sine wave model is expressed by following formula (2), where $A(i,j)(t)$ is the traffic demand between node i and j at time t, $A_{\text{average}}(i,j)$ is the average of traffic, T is the period of sine wave, and $\theta(i,j)$ is the initial phase at t=0. Variation function of a sine wave is shown in Fig. 5.

$$A(i,j)(t)=A_{\text{average}}(i,j)+0.5 \cdot A_{\text{average}}(i,j) \sin(2\pi t/T+\theta(i,j))$$  \hspace{1cm} (2)

Square wave model is expressed by following formula (3), where $M(i,j)$ is the increasing magnification, T is the starting time of a pulse and D is the duration of a pulse. Variation function of a square wave is shown in Fig. 6.

$$A(i,j)(t)=\begin{cases} A_{\text{average}}(i,j) & \text{if } (T+t \leq T+D) \\ M(i,j)-A_{\text{average}}(i,j) & \text{else} \end{cases}$$  \hspace{1cm} (3)

![Figure 3. Relay-node based load balancing algorithm](image)

![Figure 4. NSFNET Model](image)

![Figure 5. Variation function of a sine wave](image)

![Figure 6. Variation function of a square wave](image)
IV. EVALUATION RESULT

In this section, we evaluate our load balancing method for stationary traffic as well as variable traffic.

We compare three methods; (1) Shortest path based path-network, (2) Optimization by Linear Programming (LP), (3) Optimization by relay-node based load balancing algorithm. As link utilization is the main parameter to be balanced, link utilization characteristics are evaluated.

A. Load Balancing against Stationary Traffic

We assumed stationary traffic (A(i,j)=200). As for penalty function (Eq.1), $\alpha=1$ and $\beta=0.001$ were assumed (The reasoning of this assumption is described in section V-D). As for Linear Programming, we used GLPK solver v4.46[7].

The results are shown in Fig. 7, 8 and 9, respectively. In the shortest path based network (Fig. 7), large imbalance appear among link utilization. Particularly, traffic load concentrates on link 7-8 to become a bottleneck-link. On the other hand, in the optimization by LP (Fig. 8) and the optimization by relay-node algorithm (Fig. 9), traffic load is dispersed widely over the network and imbalance in link utilization is mitigated. As a result, bottleneck at link 7-8 is resolved. It should be noted that performance of relay-node algorithm optimization is close to that of LP optimization. Although LP optimization requires a large amount of calculations, it solves polynomials to minimize bottleneck link utilization. Accordingly, the result of LP optimization gives the theoretical lower limit. This means that relay-node based load balancing algorithm can achieve optimization that is close to the theoretical limit.

B. Behavior of Relay-Node Based Load Balancing

In this section, how the overlay paths are established by relay-node algorithm in NSFNET model is explained. For this purpose, we simulated traffic concentration to Node #4. That is, we quadruplicated traffic from every node to Node#4 (A(i,4)). Fig. 10 shows the primary path (dashed line) and newly established overlay paths (dotted line) to Node#4 from Node#8, #9 and #10. The figure on the side of overlay path shows the number of overlay paths that share the same route. As relay node can be selected in a duplicated manner, the same relay-node could be selected for multiple overlay paths. As a results, many overlay paths pass through the same route. It should be noted that the sum of figures on the side of overlay path does not reach Omax. This means that remaining overlay paths pass through the route of the primary path. If any node on the primary path was selected as a relay-node, the overlay path becomes identical to the primary path. Even in such situation, there is no overloading problem because the first term of penalty function suppresses link utilization. By this example, we can know that overlay paths are established to keep off from Link 7-8 that is likely to be a bottleneck link.

C. Load Balancing against Variable Traffic

In this section, we simulate the transit state by using variable traffic such as sine-wave and square-wave that are defined by section III-C. We assumed that $A_{average}(i,j)=200$ and $T=100$. Initial phase ($\theta(i,j)$) was randomly selected from $\{0, \pi/2, \pi, 3\pi/2\}$. In square wave model, traffic towards Node#4 is tripled from time $t=25$sec to $t=50$sec. That is $M(i,4)=3$, $T=25$ and $D=25$. Figure 11 and 12 show the temporal characteristics of bottleneck link utilization and horizontal axis represents time (t). From these results, we can calculate shortest path configuration cannot keep pace with traffic variation and cannot carry all of the demanded traffic. On the other hand, both optimization by LP and optimization by relay-node algorithm keep pace with traffic variation flexibly. That is, performance of the proposed algorithm is close to that of the optimization by LP.

D. Determination of Penalty Function Parameters

In section II-C, we introduced a penalty function for the selection of relay-node. In formula (1), $\alpha$ and $\beta$ need to be
Traffic that is influenced by optimization was only 20% of total traffic. That is, we can reduce the ratio of traffic switching by altering approximately 10% from that of LP optimization.

### E. Evaluation of Traffic Volume Subject for Re-optimization

In this section, traffic volume that is subject for re-optimization is evaluated. By reducing the number of paths that are changed at the time of re-optimization, influence of path switching can be mitigated. Fig. 14 compares three methods by the ratio of traffic that becomes subject for re-optimization, using the same traffic model as the section C above. From this result, we can know that optimization by relay-node algorithm can keep pace with traffic variation by altering approximately 20% of total traffic. That is, we can reduce the ratio of traffic that is influenced by path switching by 10% from that of LP optimization.

### V. CONCLUSION

We proposed relay-node based load balancing algorithm and evaluated its performance by two types of traffic in NSFNET model. It revealed that proposed algorithm can reduce bottleneck link utilization drastically compared to the conventional shortest path based path provisioning even under dynamically changing traffic demand. It achieved almost equivalent performance as the global optimization by Linear Programming that gives the theoretical limit. The amount of traffic that is influenced by optimization was only 20% of total traffic. We obtained the similar results with other topologies than NSFNET. Furthermore, primary path and overlay path determined. In order to determine these parameters for NSFNET model, we set $\alpha=1$ and observed the bottleneck link utilization for various values of $\beta$. The result is shown in Fig. 13. Vertical axis represents bottleneck link utilization and horizontal axis represents $\beta$. From this graph, we use $\beta=0.001$. In this simulation study in NSFNET model, we always set $\alpha=1$ and $\beta=0.001$.

**REFERENCES**


