Hierarchical dynamic traffic engineering considering the upper bounds of link utilizations

Yuichi Ohsita†, Takashi Miyamura§, Shin'ichi Arakawa‡, Kohei Shiomoto§, Masayuki Murata‡

† Graduate School of Economics, Osaka University
‡ Graduate School of Information Science and Technology, Osaka University
§ NTT Network Service Systems Laboratories, NTT Corporation

***Abstract*— Traffic Engineering (TE) is one efficient approach to handle traffic changes. To perform TE, a server called the Path Computation Element (PCE) collects the traffic information from all nodes within the network. Then, the PCE calculates the routes suitable to the current traffic. However, in a large-scale network, it is difficult for one PCE to collect all traffic information in a short period of time. Thus, it takes time to change the routes according to traffic changes. In this paper, we propose a method that changes the routes suitable to the current traffic soon after the traffic changes. In our method, we hierarchically divide the network into multiple ranges; the ranges of the lowest layer are constructed of a small number of nodes and the ranges of the upper layer are constructed from the multiple ranges of the lower layer. We deploy a PCE for each range. The PCEs in the lowest layer change the routes within a small range in a short interval according to the traffic information within the range to handle the traffic changes that occur in a short period of time. Against the traffic change that cannot be handled in the lower layer, the PCEs in the upper layer change the routes within the large ranges of the upper layer according to the aggregated traffic information collected from the PCEs of the lower layer. We also propose a method to aggregate traffic information and a method to calculate the new routes by using the aggregated traffic information considering the upper bounds of link utilizations. In this method, we aggregate traffic information so that we can calculate the upper bounds of the link utilizations after the route change only from the aggregated traffic information. Then, the PCE obtaining the aggregated traffic information calculates the new routes without causing any new congestion by checking the upper bounds of link utilizations calculated from the aggregated traffic information. In this paper, we evaluate our method by simulation and clarify that our method can mitigate the congestion soon after the traffic changes.**

***Index Terms*— Traffic Engineering, Hierarchization, Link Utilization, Traffic Matrix**

I. Introduction

In recent years, various new applications such as peer-to-peer, video on demand, SaaS and PaaS are deployed over the Internet. The traffic generated by such new applications causes unpredictable large changes in traffic demands. Though we can design the fixed routes so as to handle any possible traffic changes [1], this approach requires more than double the bandwidth required for the routes suitable to the traffic at each time.

Traffic Engineering (TE) [2–8] is one efficient way of handling such traffic changes. In the TE, we deploy a server called the Path Computation Element (PCE). The PCE collects the information of traffic amounts between all nodes periodically. Then, the PCE recalculates the routes within the network so as to accommodate all the current traffic without any congestion according to the collected traffic information.

However, in a large-scale network, the information of traffic amounts between all nodes is hard to collect in a short interval because the number of nodes the PCE has to query and the amount of information to be collected are large. Thus, the TE using the information of traffic amounts between all nodes is hard to recalculate the routes against traffic changes that occur in a short period of time.

To reduce the amount of information required by the TE, several TE methods using only the information of traffic amount on each link have been proposed [3–6]. The methods proposed by Refs. [3–5] perform the TE by using the traffic amounts between all nodes estimated by the traffic amount on each link. However, there are the cases that the routes calculated by these approaches cannot mitigate the congestions due to estimation errors.

The method that is not affected by estimation errors has also been proposed by Ref. [6]. In this method, we calculate the range of possible traffic amount by using the traffic amount on each link. Then, we calculate the routes so as to minimize the upper bounds of the link utilizations for the range of the possible traffic amount. However, this approach requires large calculation time in a large-scale network.

One approach to perform the TE against the traffic changes that occur in a short period of time is to divide the network into multiple small ranges. By performing the TE within the small ranges, the amount of traffic information required by the TE and the calculation time of the TE are reduced. As a result, we can perform the TE in a short interval. However, the TE only within the small ranges cannot handle the large traffic changes.

In this paper, we propose a TE method that can handle both traffic changes in a short period of time and large traffic changes. In our method, we hierarchically divide the network into multiple ranges; the ranges of the lowest layer are constructed of a small number of nodes and the ranges of the upper layer are constructed from the multiple ranges of the lower layer. We deploy a PCE for each range. Each PCE obtains the traffic information within the range by collecting traffic information from the node within the range or exchanging the aggregated traffic information with PCEs of the lower or upper layers. Then, the PCE calculates the routes so as to mitigate the congestion without causing any congestion according to the obtained traffic information.

In our approach, the PCE of the lowest layer changes the routes in a short interval because the amount of required

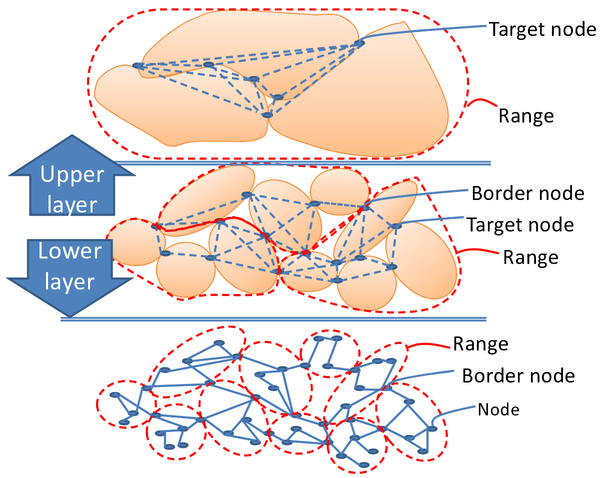

Fig. 1. Overview of Hierarchical Dynamic Traffic Engineering

information and the fraction of the traffic affected by the route change are small. Thus, we can change the routes soon after the traffic changes occur. In addition, even if the PCE of the lowest layer cannot mitigate the congestion, we can mitigate the congestion by the route change of the upper layers.

Our approach is different from the existing hierarchical routing approaches [9–11] at the following points. 1) The hierarchically constructed range in our method is used only for monitoring the traffic, detecting the congestion and calculating the new routes for the flow passing the congested link. Thus, in our method, the network topology is not required to be constructed hierarchically. In addition, the hierarchically constructed ranges in our method do not limit the possible routes and do not cause the concentration of traffic to a certain node. 2) In this paper, we also propose a method to aggregate traffic information so that we can detect the congestion and calculate the upper bounds of the link utilizations after the route change only from the aggregated traffic information. By avoiding the calculated upper bounds of the link utilizations exceeding a threshold, we calculate a new route for a flow without causing any new congestion. 3) Our method uses the aggregated information not only from the lower layer but also from the upper layer. By using the aggregated information from the upper layer, the PCE changes the routes to mitigate the congestion detected within the range without causing any congestion outside the range.

The rest of this paper is organized as follows. In Section II, we explain the overview of our hierarchical dynamic traffic engineering. In Section III, we propose a method to aggregate traffic information and a method to recalculate the routes according to the aggregated traffic information. Then, we evaluate our method in Section IV. Finally, Section V provides a conclusion.

II. Overview of Hierarchical Dynamic Traffic Engineering

In our method, we hierarchically divide the network into multiple ranges. Fig. 1 shows the overview of the hierarchically constructed ranges. As shown in Fig. 1, in the lowest layer, we divide the network into multiple ranges so that each link belongs to one of the ranges. Some of the nodes become the border of the ranges. We call the nodes at the border of the ranges the *border nodes*. We deploy a PCE for each range. Each PCE collects traffic information on all links within the range by querying all nodes within the range.

The ranges of the upper layers are constructed from the multiple ranges of the lower layer. Similar to the range of the lowest layer, we deploy a PCE for each range in the upper layer. Each PCE of the range of upper layer maintains the topology constructed of the nodes that are the border node in the lower layer. We call the node included in the topology maintained by the PCE the *target node*. Each PCE obtains the aggregated traffic information between the target nodes that belong to the same range in the lower layer from the PCE of the lower layer.

Each PCE of each layer checks whether the congestion occurs within the range periodically by using the obtained traffic information. If the congestion is detected, the PCE changes the routes within the range so as to mitigate the detected congestion.

In our method, the interval to check the congestion is set to be shorter in the lower layer. By checking the congestion in a short interval, the PCE of the lower layer detects the congestion soon after the congestion occurs. Then, the PCE changes the routes within the small range. The PCE of the upper layer changes the routes only when the congestion cannot be mitigated sufficiently by the route changes in the lower layers because the route changes within a large range in the upper layer affect large amount of traffic.

Figure 2 shows the routes changed by the PCE after the detection of the congestion. As shown in Fig. 2, each PCE changes the routes of the traffic passing nodes within the range and the border nodes passed by the traffic from/to the nodes within the range. The new routes for the traffic passing the nodes within the range are calculated over the topology constructed of the target nodes. When calculating the routes, we avoid the new congestion by checking the upper bounds of the link utilizations calculated from the obtained traffic information. Then, the PCE changes the routes of the traffic to the calculated routes by sending the message to the nodes within the range. When the PCE changes the border nodes traversed by the traffic from/to the nodes within the range, the PCE also obtains the traffic information between border nodes within the range and the border nodes outside the range from the PCE of the upper layer. Then, the PCE selects the border nodes so as to mitigate the detected congestion without causing any new congestion both within the range and outside the range.

In our method, each PCE uses only local traffic information or aggregated traffic information. Thus, each PCE can collect or exchange the traffic information in a short interval since the amount of traffic information required by each PCE is small. By using the collected or exchanged traffic information, the PCEs of the lower layers change the routes in a short interval againt the traffic changes that occur in a short period of time. In addition, by using the exchanged aggregated traffic information, the PCEs of the upper layer also detect the congestion that cannot be mitigated by the PCEs in the lower layers and change the routes to mitigate the detected congestion.

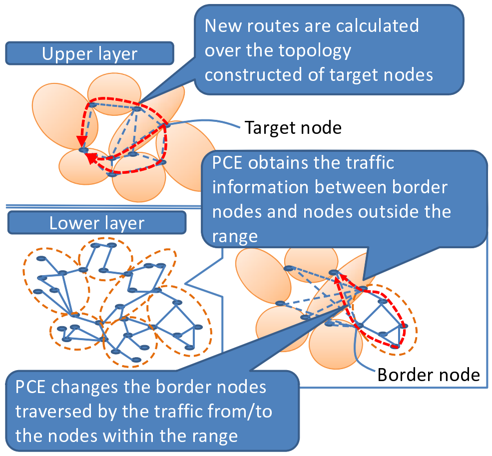

Fig. 2. Route changes by a PCE of the lowest layer and a PCE of the upper layer

TABLE I
AGGREGATED INFORMATION FOR LINK l

Notation	Description
b_l	Bandwidth of link l
x_l^{all}	Total Traffic Amount on Link l
$x_l^{\max}$	Upper Bound for Traffic Amount of Flows on Link l Whose Routes can be Changed in the Upper Layer
$x_l^{\min}$	Lower Bound for Traffic Amount of Flows on Link l Whose Routes can be Changed in the Upper Layer
$f_{a,b,l}^{\mathrm{lower}}$	Fraction of Traffic between Border Nodes a and b passing Link l

The details of the method to aggregate traffic information and the method to change the routes according to the aggregated information are described in Section III.

III. TRAFFIC INFORMATION AGGREGATION AND ROUTE CHANGES BASED ON AGGREGATED TRAFFIC INFORMATION

In this section, we propose a method to aggregate traffic information and method to change the routes according to the aggregated information.

In our method, each PCE knows the set of the target nodes which is denoted as N. Each PCE also knows the current routes of the flows whose routes can be changed by the PCE. We denote the set of the flows whose routes can be changed as P. The route of the flow p is represented as $f_{p,a,b}^{\mathrm{in}}$ where a and b are the nodes included in N and $f_{p,a,b}^{\mathrm{in}}$ denotes the fraction of traffic of flow p passing nodes a and b.

A. Traffic Information Aggregation

1) Information to the PCE of the upper layer: In our method, each PCE selects the link whose utilization is the largest among the link passed by each flow between border nodes. Then, the PCE generates the aggregated traffic information shown in Table I for the selected links and sends the generated traffic information to the PCE of the upper layer. Among the information shown in Table I, $f_{a,b,l}^{\mathrm{lower}}$ is calculated from $f_{p,a,b}^{\mathrm{in}}$. $x_l^{\max}$ and $x_l^{\min}$ are calculated by the linear programming described in Appendix I.

By obtaining the above aggregated traffic information, the PCE of the upper layer can detect the congestion between the border nodes. In addition, by using $f_{a,b,l}^{\mathrm{lower}}$, $x_l^{\max}$ and $x_l^{\min}$, the PCE of the upper layer can identify the flow passing the congested link and calculate the upper bound for the link utilization after the change of routes. The detail of the calculation of routes using the aggregated traffic information is explained in Section III-B.

2) Information to the PCE of the lower layer: The PCE uses the aggregated traffic information between border nodes within the range and the border nodes outside the range when changing the border nodes traversed by the traffic from/to the nodes within the range. This aggregated traffic information is generated by the PCE of the upper layer by the following steps. First, the PCE of the upper layer selects the flows from/to the range maintained by the PCE that is the destination of the aggregated traffic information. Then, the PCE of the upper layer selects the link whose utilization is the largest among the links passed by each selected flow. Finally, the PCE calculate the information described in Table I for the selected links in the same manner as Section III-A.1. By using the aggregated traffic information, the PCE of the lower layer can check whether the change of the border nodes passed by the traffic causes new congestion outside the range.

B. Route Change Based on the Aggregated Traffic Information

In this subsection, we propose the method to change the routes so as to mitigate the congestion according to the aggregated traffic information. In our method, we first identify the congested links by using the aggregated traffic information obtained from the PCE of the lower layer. In this paper, we identify the links whose utilizations are larger than the threshold T_H as the congested links. We denote the set of identified congested links as L^{target}.

Next, we identify the flows passing the congested links. Each PCE knows the set of the flows whose routes can be changed by the PCE denoted as P^{current} and the routes of the flows denoted as $f_{p,a,b}^{\mathrm{in}}$. In addition, each PCE also knows the target node pairs whose flows pass the congested link $l \in L^{\mathrm{target}}$ from the aggregated traffic information of link l. Thus, by using $f_{p,a,b}^{\mathrm{in}}$ and the aggregated traffic information of link l, we can identify the flows passing the congested links. We denote the set of the identified flow as P^{target}.

After the identification of the flow passing the congested links, we calculate the new routes for the identified flows $p \in P^{\mathrm{target}}$. When calculating the new routes, we avoid the new congestion by checking the upper bounds of link utilizations calculated by using the aggregated traffic information. The new route for flow p is calculated by the following steps.

Step 1: Construct the topology G of the target nodes. In the lowest layer, G is constructed by adding links between the target nodes where a link exists. In other layers, G is constructed by adding links between the target nodes that belong to the same range in the lower layer.

Step 2: Calculate the route for p on the topology G by the shortest path first algorithm.

Step 3: Check whether the upper bounds for the utilizations of the links on the route calculated at Step 2 are less than the threshold T_H. If the upper bounds of the utilizations of all links are less than the threshold T_H, designate the routes calculated at Step 2 as the new routes for the flow p. Otherwise go to Step 4.

Step 4: Remove links whose upper bounds of utilizations are larger than T_H from the topology G and go to Step 2.

The upper bounds for the link utilizations required at Step 3 of the above steps are calculated by the linear programming described in Appendix III. However, in our method, we reduce the number of calculation of the linear programming described in Appendix III by the following steps to reduce the calculation time.

In our method, we calculate the upper bound for the traffic amount $t_p^{\max}$ of each flow p in P^{target} in advance by calculating the linear programming described in Appendix II. Then we calculate the roughly forecasted link utilization $\hat{x}_l$ by the following equation,

$$\hat{x}_l = \frac{1}{b_l}\left(x_l^{\text{before}} + \sum_{p \in P_l^{\text{after}}} t_p^{\max}\right), \quad (1)$$

where x_l^{before} is the traffic amount on link l before the route change and P_l^{after} is the set of flows newly passing the link l after the route change. $\hat{x}_l$ is larger than the actual link utilization after the route change. Therefore, if $\hat{x}_l$ is less than T_H, we recognize that the link l is not congested after the route change without calculating the linear programming described in Appendix III. When $\hat{x}_l$ is larger than T_H, we check the upper bounds for the link utilizations after obtaining the accurate upper bound for the utilization of the link l by calculating the linear programming described in Appendix III.

After the routes for all flows in P^{target} is determined, the PCE configure the nodes within the range to set the routes. The above steps calculate the routes on the topology G. That is, the routes calculated by the above steps indicates the set of the target nodes passed by each flow. When the PCE of the upper layer calculates the new routes, the routes between the target nodes connected in topology G are set according to the routes between the nodes calculated by the PCE of the lower layers.

The above steps are used both for the calculation of the routes within the range and for the calculation of the border nodes passed by the flow from/to the nodes within the range. In our method, we first try to mitigate the congestion by changing routes within the range. Then if changing the routes only within the range cannot mitigate the congestion, we change the border nodes passed by the flow from/to the nodes within the range.

IV. Evaluation

In this section, we evaluate our method by simulation. We use the US topology (46 nodes, 70 links) shown in Fig. 3. We generate the initial traffic amount between each node pair randomly so as to follow the lognormal distribution according to the results of Ref. [12] and set the initial routes so as to minimize the link utilization. Then, we newly generate the current traffic amount between each node pair randomly so as to follow the lognormal distribution.

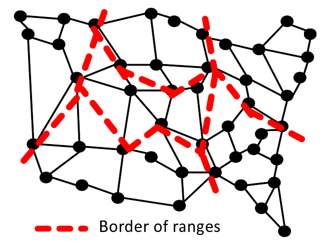

Fig. 3. US topology (2 layers)

TABLE II
Comparison of Amount of traffic information and Calculation time

	Number of elements of traffic information (max)	Calculation Time (max) [sec]
Our method (lowest layer)	77	0.29
Our method (top layer)	108	3.68
Method using the information of the whole network	242	875.25

In our evaluation, we divide the network into two layers. In the lowest layer, we divide the network into 6 ranges as shown in Fig. 3. In each range of the lowest layer, we change the routes once per minute. In the top layer, we maintain the whole network by using the aggregated traffic information obtained from the PCE of the lowest layer. In the top layer, we change the routes once per 13 minutes. We assume the traffic between each node pair can be splittable. When changing the routes, we regard the one-tenth of the traffic between each node pair as a flow.

A. Amount of traffic information and calculation time

We compare our method with the method using the utilizations of all links of the whole network. In this comparison, we use two metrics, the number of elements of traffic information required by a PCE and the calculation time required to calculate the routes once. In both methods, we calculate the routes by the method described in Section III-B with $T_H = 0.4$.

Table II shows the results. According to Table II, the number of elements of traffic information required by each PCE of our method is much smaller than the method using the information of the whole network. In addition, the calculation time of our method is also much smaller than the method using the information of the whole network. This is because our method calculates the routes only within the small ranges or the routes on the topology constructed of the nodes that are the border nodes in the lowest layer.

B. Achievable link utilization

We investigate the link utilization achieved by our method in 30 minutes by changing the threshold T_H. In this evaluation, we generate 70 patterns of traffic amount between each node pairs. Fig. 4 shows the cumulative distribution of the link utilizations achieved by changing the routes. In Fig. 4, we compare the three cases; the case of changing routes in all layers by our method, the case of changing routes only in the lowest layer by our method, and the case of changing routes by the method using the traffic information of the whole network. In addition, we also plot the cumulative distribution of the link utilizations before the route change.

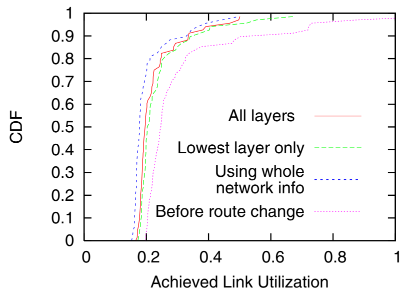

Fig. 4. CDF of the link utilization achieved by TE methods

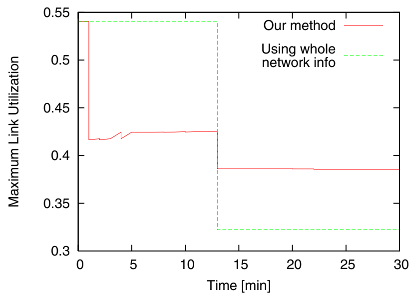

Fig. 5. Time series of maximum link utilization

Figure 4 shows that the changing routes only in the lowest layers can reduce the link utilization significantly in many cases. That is, we can mitigate the congestion by changing the routes only in the lowest layer in many cases.

Figure 4 also shows that we can reduce the link utilization by changing the routes in the top layer even when changing routes only in the lowest layer cannot mitigate the congestion sufficiently. As a result, our method can achieve the similar link utilization to the method using the traffic information of the whole network.

C. Time to mitigate the congestion

Figure 5 shows the time series of the maximum link utilization after the traffic changes. In this figure, we plot two cases that routes are changed by our method and the method using the utilizations of all links of the whole network. In this simulation both methods calculate the routes by the method described in Section III-B with $T_H = 0.4$. The method using the utilizations of all links of the whole network changes the routes in the same interval of the top layer of our method (i.e., once per 13 minutes).

Figure 5 shows that our method reduces the maximum link utilization soon after the traffic change by changing the routes in the lowest layer. Then, we can make the maximum link utilization less than 0.4, 13 minutes after the traffic change by changing the routes in the top layer.

Figure 6 shows the cumulative distribution of the time required to make the maximum link utilization less than 0.4. In this figure, we investigate the time required to make the maximum link utilization less than 0.4 for 37 cases that the maximum link utilizations before the route change become more than 0.4. Fig. 6 shows that our method can make the maximum link utilization less than 0.4 faster than the method using the utilization in all links in more than 80% of cases.

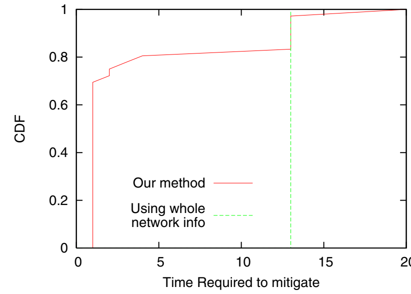

Fig. 6. Time required to make the maximum link utilization less than 0.4

This is because we can mitigate the congestion by changing the routes in the lowest layer in many cases as shown in Fig. 4.

V. Conclusion and Future work

In this paper, we proposed a method that changes the routes suitable to the current traffic soon after the traffic changes. In our method, we hierarchically divide the network into multiple ranges. Then we perform the TE within the small ranges in a short period of time to handle traffic changes that occur in a short period of time. In addition, we also perform the TE within the large ranges constructed from multiple small ranges to handle significant traffic changes. In this paper, we evaluated our method by simulation and clarified that our method can mitigate the congestion soon after the traffic changes.

One of our future research topics is to evaluate our method in other topologies or in the cases of more than 3 layers. Another future research topic is a method to divide the networks suitable to our hierarchical dynamic traffic engineering.

Acknowledgments

This work was partly supported by Grant-in-Aid for Scientific Research (B) 22300023 and Grant-in-Aid for Young Scientists (B) 2370077 of the Ministry of Education, Culture, Sports, Science and Technology in Japan.

Appendix I
Linear programming to calculate the Upper/Lower bounds of traffic included in the aggregated traffic information

Inputs

- P^{current} Set of flows at the current layer.
- P^{upper} Set of flows whose routes can be changed in the upper layer.
- L Set of links whose traffic information is maintained by the PCE.
- $f_{p,l}$ Fraction of traffic amount of flow $p \in P^{\mathrm{current}}$ passing the link $l \in L$. This is calculated by the following equation according to the routing information $f^{\mathrm{in}}_{p,a,b}$ maintained by the PCE.

$$f_{p,l} = \begin{cases} f^{\mathrm{lower}}_{a,b,l} f^{\mathrm{in}}_{p,a,b} & \left(\begin{array}{l} l \text{ is included in} \\ \text{the aggregated} \\ \text{information} \end{array}\right) \\ f^{\mathrm{in}}_{p,l_a,l_b} & \left(\begin{array}{l} l \text{ is directly} \\ \text{monitored link} \end{array}\right) \end{cases}$$

where l_a and l_b is the node connected to link$l$.
- $x^{\min}_l$,$x^{\max}_l$ Aggregated traffic information of link l included in L.

Variables

t_p Traffic amount of flow p.

Objective

To obtain the upper bound of traffic amount included in the aggregated traffic information, we maximize the sum of traffic amount of flows p included in P^{upper} passing the link l. To obtain the lower bound of traffic amount included in the aggregated traffic information, we minimize the sum of traffic amount of flows p included in P^{upper} passing the link l.

$$\text{maximize(minimize)} \sum_{p \in P^{\text{upper}}} f_{p,l} t_p$$

Constraints

$$\forall l \in L\colon x_l^{\min} \leq \sum_{p \in P^{\text{current}}} f_{p,l} t_p \leq x_l^{\max}$$

Appendix II
Linear Programming to obtain the upper bound of the traffic amount of the flow

Inputs

P^{current}	Set of flows at the current layer.
L	Set of links whose traffic information is maintained by the PCE.
$f_{p,l}$	Fraction of traffic amount of flow $p \in P^{\text{current}}$ passing the link $l \in L$.
$x_l^{\min}$,$x_l^{\max}$,x_l^{all}	Aggregated traffic information of link l included in L.

Variables

t_p Traffic amount of flow p.

Objective

Maximize the traffic amount of flow p.

$$\text{maximize } t_p$$

Constratins

$$\forall l \in L\colon x_l^{\min} \leq \sum_{p \in P^{\text{current}}} f_{p,l} t_p \leq x_l^{\max}$$

$$\forall l \in L\colon \sum_{p \in P^{\text{current}}} f_{p,l} t_p + t_l^{\text{loewer}} = x_l^{\text{all}}$$

Appendix III
Linear Programming to obtain the upper bound of link utilization

Inputs

P^{current}	Set of flows at the current layer.
L	Set of links whose traffic information is maintained by the PCE.
$f_{p,l}$	Fraction of traffic amount of flow $p \in P^{\text{current}}$ passing the link $l \in L$ before the route change.
$f_{p,l}^{\text{new}}$	Fraction of traffic amount of flow $p \in P^{\text{current}}$ passing the link $l \in L$ after the route change. This is defined by using the routes calculated by the PCE $f_{p,a,b}^{\text{new}}$ in the same way as $f_{p,l}$.
$x_l^{\min}$,$x_l^{\max}$,x_l^{all},b_l	Aggregated traffic information of link l included in L.

Variables

t_p	Traffic amount of flow p.
t_l^{loewer}	Sum of traffic amounts on link l whose routes cannot be changed by the PCE of the current layer.

Objective

Maximize the link utilization of link l.

$$\text{maximize } \frac{1}{b_l}\left(\sum_{p \in P^{\text{current}}} f_{p,l}^{\text{new}} t_p + t_l^{\text{loewer}}\right)$$

Constraints

$$\forall l \in L\colon x_l^{\min} \leq \sum_{p \in P^{\text{current}}} f_{p,l} t_p \leq x_l^{\max}$$

$$\forall l \in L\colon \sum_{p \in P^{\text{current}}} f_{p,l} t_p + t_l^{\text{loewer}} = x_l^{\text{all}}$$

References

[1] D. Applegate and E. Cohen, “Making routing robust to changing traffic demands: algorithms and evaluation,” *IEEE/ACM Transactions on Networking*, vol. 14, pp. 1193–1206, Dec. 2006.

[2] B. Fortz and M. Thorup, “Internet traffic engineering by optimizing OSPF weights,” in *Proceedings of IEEE INFOCOM*, vol. 2, pp. 519–528, Mar. 2000.

[3] M. Roughan, M. Thorup, and Y. Zhang, “Traffic engineering with estimated traffic matrices,” in *Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement*, pp. 248–258, Nov. 2003.

[4] Y. Ohsita, T. Miyamura, S. Arakawa, S. Ata, E. Oki, K. Shiomoto, and M. Murata, “Gradually reconfiguring virtual network topologies based on estimated traffic matrices,” *IEEE/ACM Transactions on Networking*, vol. 18, pp. 177–189, Feb. 2010.

[5] Y. Ohsita, T. Miyamura, S. Awakawa, E. Oki, K. Shiomoto, and M. Murata, “Estimation of current traffic matrices from long-term traffic variations,” *IEICE Transactions on Communications*, vol. E92-B, pp. 171–183, Jan. 2009.

[6] I. Juva, “Robust load balancing,” in *Proceedings of GLOBECOM*, pp. 2708–2713, Nov. 2007.

[7] G. Retvari and G. Nemeth, “Demand-oblivious routing: distributed vs. centralized approaches,” in *Proceedings of IEEE INFOCOM*, pp. 1217–1225, Mar. 2010.

[8] P. Casasa, F. Larrocab, J.-L. Rougierc, and S. Vatona, “Taming traffic dynamics: Analysis and improvements,” *Computer Communications*, July 2010.

[9] L. Kleinrock and F. Kamoun, “Hierarchical routing for large networks performance evaluation and optimization,” *Computer Networks*, vol. 1, pp. 155–173, May 1976.

[10] S. Uludag, K.-S. Lui, K. Nahrstedt, and G. Brewster, “Analysis of topology aggregation techniques for QoS routing,” *ACM Computing Surveys*, vol. 39, pp. 7/1–7/31, Aug. 2007.

[11] W. S. Lai and Y. Lai, “Hierarchical routing by PNNI and OSPF/BGP,” in *Proceedings of IEEE PACRIM*, pp. 951–956, Aug. 2009.

[12] A. Nucci, A. Sridharan, and N. Taft, “The problem of synthetically generating IP traffic matrices: Initial recommendations,” *ACM SIGCOMM Computer Communication Review*, vol. 35, pp. 19–32, July 2005.