Aggregation of Traffic Information for Hierarchical Routing Reconfiguration

Yuichi OHSITA[a,*], Takashi MIYAMURA[b], Shin'ichi ARAKAWA[a],
Shohei KAMAMURA[b], Daisaku SHIMAZAKI[b], Kohei SHIOMOTO[c],
Atsushi HIRAMATSU[d], Masayuki MURATA[a]

[a] *Graduate School of Information Science and Technology, Osaka University,*
1-5, Yamadaoka, Suita, Osaka, 565-0871, Japan
[b] *NTT Network Service Systems Laboratories, NTT Corporation,*
3-9-11 Midori-cho, Musashino, Tokyo 180-8585, Japan.
[c] *NTT Network Technology Laboratories, NTT Corporation,*
3-9-11 Midori-cho, Musashino, Tokyo 180-8585, Japan.
[d] *NTT Advanced Technology Corporation, Japan*

Abstract

One approach to accommodate a large and time-varying traffic is dynamical routing reconfiguration based on the traffic matrix (TM), which is obtained by monitoring the amounts of traffic between all node pairs. However, it is difficult to monitor and collect the amounts of traffic between all node pairs in a large network. Though reconfiguration methods only based on the amount of traffic on each link have been proposed to overcome this problem, these methods, require a large calculation time and cannot be applied to large networks. This paper discusses a dynamic routing reconfiguration method that can adapt routes to changes in traffic within a short period only based on the amount of traffic on each link. We introduce a hierarchical routing reconfiguration based on the monitored amount of traffic on each link to reduce the calculation time. Moreover, we also propose a method of aggregating traffic information that is suitable for hierarchical routing reconfiguration based on the monitored amount of traffic on each link. Our method aggregates traffic information so that the upper bounds of link utilization after route changes can be calculated by using the aggregated traffic information. Thus, the routing controller using the aggregated traffic information calculates the suitable routes without large link utilization by taking into consideration the upper bounds of the link utilization. This paper evaluates our method through simulations, where we demonstrated that the routing reconfiguration of each layer calculated suitable routes with

---

[*] The corresponding author. (Phone: +81-6-6879-4542)

Email addresses: `y-ohsita@ist.osaka-u.ac.jp` (Yuichi OHSITA), `arakawa@ist.osaka-u.ac.jp` (Shin'ichi ARAKAWA), `kamamura.shohei@lab.ntt.co.jp` (Shohei KAMAMURA), `shimazaki.daisaku@lab.ntt.co.jp` (Daisaku SHIMAZAKI), `shiomoto.kohei@lab.ntt.co.jp` (Kohei SHIOMOTO), `atsushi.hiramatsu@ntt-at.co.jp` (Atsushi HIRAMATSU), `murata@ist.osaka-u.ac.jp` (Masayuki MURATA)

[1] A part of this paper was presented at IEEE GLOBECOM, Houston, Texas, Dec. 2011 [1].

short calculation times. Then, we reduced the link utilization immediately after traffic had changed by combining the routing reconfiguration of each layer.

Keywords: Traffic engineering; routing reconfiguration; aggregation; link utilization; traffic matrix

---

1. Introduction

Various new applications such as cloud storage services have been deployed over the Internet. Such applications increase the amount of traffic and cause unpredictable changes in traffic. A network must accommodate such time-varying traffic efficiently. However, if routing optimal to a current traffic is configured, the routing configuration becomes no longer suitable after traffic changes.

One approach to accommodate such a large and time-varying traffic is dynamical routing reconfiguration [2–13]. In these methods, a central server dynamically calculates the routes periodically based on the traffic matrix (TM), which is obtained by monitoring the amounts of traffic between all node pairs. Then, the nodes within a network are configured based on the calculated routes. In this paper, we call the central server *routing controller*. However, it is difficult to monitor and collect the amounts of traffic between all node pairs in large networks, because the amount of traffic required to be monitored is $\mathrm{O}(N^2)$ where N is the number of nodes. Thus, the routing reconfiguration method using the monitored TM is hard to control the routes in real time to follow changes in traffic that occur in short periods.

Routing reconfiguration methods only based on the amount of traffic on each link have been proposed to overcome this problem [8, 9]. The amount of traffic on each link can easily be monitored and collected, because the amount of traffic required to be monitored is $\mathrm{O}(L)$ where L is the number of links. The amount of traffic on each link in these methods is used as the constraint on the TM. They then calculate the routes to avoid large utilization of links for all TMs that satisfy the constraints. These methods, however, cannot be applied to large networks, because the calculation time to obtain the largest link utilization for all TMs satisfying the constraints is $\mathrm{O}(N^8L^2)$, because the L linear programmings which have N^2 variables and L constraints are required to be solved, and the linear programming with n variables and l constraints can be solved at $O(n^4l)$ by using the method by Renegar [14].

One approach to reducing the calculation time is to hierarchically divide the network into areas; the area with the lowest layer is constructed from a small number of nodes, and the area with the upper layer is constructed from multiple areas of the lower layer. The multiple layers are constructed by continuing to construct the area of the upper layer from multiple areas of the lower layer. The traffic information for each area is aggregated and exchanged between the layers. Then, the routing controller of each area of each layer calculates the routes by only using aggregated traffic information. The time to calculate the routes is significantly reduced by hierarchically dividing the network and aggregating

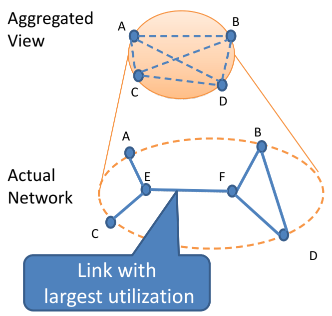


Figure 1: Example of simple aggregation of traffic information

traffic information. We will call this method *hierarchical routing reconfiguration* after this.

The routing controller of each layer is required to predict link utilization after the route has changed only from aggregated traffic information to effectively carry out the hierarchical routing reconfiguration. If we use aggregated traffic information that only includes the amount of traffic in each area and the degree of congestion, we cannot accurately predict link utilization after the route has changed. This is because we cannot obtain the amount of traffic in flows whose routes can be changed from the traffic information aggregated at the lower layer. As a result, the routing reconfiguration using aggregated traffic information may even increase link utilization. For example, the link between E–F is a bottleneck link, and all flows between A–B, A–D, C–B, and C–D pass the link between E–F in Figure 1. However, we cannot know that the traffic between A–B, A–D, C–B, and C–E are concentrated on link E–F from the aggregated traffic information without knowledge of the detailed information of the topology within the area. As a result, the routing controller using aggregated traffic information may increase the traffic for all these flows, and cause large utilization of link E–F.

This paper discusses the traffic information aggregation method for the hierarchical routing reconfiguration, and the routing reconfiguration method using the aggregated traffic information. As describe above, one of the most important issues in this approach is to calculate routes that can effectively reduce link utilization only from aggregated traffic information. In our traffic information aggregation method, the aggregated traffic information is generated so as to include the information of the links whose utilization may become large and the constraints on TMs. The upper bound of the link utilization after the route change is calculated by using the constraints on TMs obtained from the aggregated traffic information. Thus, we reconfigure the routes based on the aggregated traffic information to reduce the link utilization by taking into consideration the upper bounds of the link utilization.

The routing controller of the lower layer in the hierarchical routing reconfiguration uses detailed traffic information on narrower areas to change routes locally to mitigate large utilization of links. The routing controller of the upper layer uses aggregated traffic information on wider areas to mitigate large utilization of links that cannot be mitigated by changes in local routes.

The traffic information used by the routing controller of each area at each layer in the hierarchical routing reconfiguration only includes a small number of links. Thus, the calculation time for routing reconfiguration method that takes into consideration all TMs under the constraints defined by aggregated traffic information is short. The number of nodes handled by the routing controller of each area and the number of links included in aggregated traffic information are reduced by hierarchically dividing the network and using aggregated traffic information. The calculation time to obtain the largest link utilization for all TMs required by the routing controller of each area is $\mathrm{O}(N^8 L^2 / R^{10})$ by reducing the number of nodes to N/R and the number of links to L/R, where R is the number of areas, and this is significantly smaller than the routing reconfiguration method without hierarchically divided areas.

The main contributions of this paper are summarized below.

1. This paper proposes a traffic information aggregation suitable to the hierarchical routing reconfiguration. In this aggregation method, we first select links whose utilization may become large after the routes have changed at each layer. Then, we calculate the ratio of the amount of traffic of flow between source and destination nodes passing the link for each of the selected links, and the upper and lower bounds for the traffic of flows passing the link whose routes can be changed by the routing controller using aggregated traffic information. Finally, we generate aggregated traffic information including the calculated ratio of traffic and the upper and lower bounds for traffic whose routes can be changed for selected links. We can obtain constraints on TM by using this aggregated traffic information and identify links whose utilizations may become large after the route has changed. Thus, the routing controller using aggregated traffic information can calculate the routes to reduce link utilization by taking into consideration the upper bounds for link utilization.

2. This paper introduces a method of calculating routes using the aggregated traffic information. In this method, the routing controller calculates the routes so as to avoid the large link utilization by considering the all TMs satisfying the constraints obtained from the aggregated traffic information. In addition, the routing controller using the aggregated traffic information cannot change the routes of the traffic whose source and destination nodes are within the same area of the lower layer, but the traffic whose routes cannot be changed by the routing reconfiguration using the aggregated traffic information may affect the link utilization. Thus, our method considers such traffic. This method focuses the calculation of new routes instead of setting the configured routes, and the routes calculated by our method can be configured by any routing technologies such as Multiprotocol Label Switching (MPLS) or OpenFlow.

3. This paper demonstrates that the hierarchical routing reconfiguration using our aggregated traffic information reduces the link utilization immediately after the traffic changes without causing the large link utilization. The routing reconfiguration method introduced in this paper avoids the large link utilization by considering all TMs satisfying the constraints obtained from the aggregated traffic information. The calculation time of the routing reconfiguration is reduced in our method by reducing the number of nodes handled by the routing reconfiguration of each area and the number of links included in aggregated traffic information. These advantages of our method are demonstrated through numerical evaluation using various topologies.

The rest of this paper is organized as follows. Section 2 explains related work. Section 3 explains the routing reconfiguration method that takes into account all TMs under the constraint defined by traffic information in detail. We propose hierarchical routing reconfiguration and a method of aggregating traffic information suitable for hierarchical routing reconfiguration in Section 4. Then, Section 5 discusses our evaluation of the approach and Section 6 provides a conclusion.

2. Related Work

2.1. Routing reconfiguration

Many methods of routing reconfiguration have been proposed [2–13]. They have accommodated time-varying traffic by dynamically setting the open shortest path first (OSPF) weights or reconfiguring the label switched paths (LSP) of Multiprotocol Label Switching (MPLS).

In these methods, a central server, called *routing controller* in this paper, calculates the routes based on the current traffic, and configure the routers in the network based on the calculated routes. The routing controller requires a TM that is obtained by monitoring the amounts of traffic between all node pairs. The routes suitable for current traffic are calculated by methods of optimization using TM as input. Fortz et al. solved the problem to optimize the OSFP weights by local search method[2]. The optimization problems of the routes using TMs considering the failures have also been discussed [3]. However, information about the amount of traffic between all nodes is difficult to collect in short time intervals in large-scale networks because the amount of traffic to be monitored and collected is $\mathrm{O}(N^2)$ where N is the number of nodes. Methods of routing reconfiguration using estimated TM have also been proposed [4–7]. They only require information on traffic on each link, which can be collected much more easily than directly monitoring the traffic between all nodes because the amount of traffic that needs to be monitored is $\mathrm{O}(L)$ where L is the number of links. Roughan et al. first proposed the routing reconfiguration using the estimated TMs [4]. We also proposed a method where the routing reconfiguration and TM estimation cooperates with each other to reduce the estimation errors [15], and improved the method to consider the long-term traffic changes [6], or to consider the sudden traffic changes [5]. To reduce the traffic information to be collected, we also proposed a method to estimate the TMs from a small number of links [7]. However, the estimated TM includes estimation errors even when the method to reduce the estimation error is used, and link utilization may not be reduced because of these estimation errors.

Methods of routing reconfiguration that take into consideration all TMs under constraints have also been proposed [10–13]. They calculate routes to minimize the worst-case utilization of links for all TMs. The constraints on TMs can be set without monitoring traffic by using the bandwidths of the ingress and egress links; the total traffic from a node should be less than the bandwidth of the ingress link of the node, and the total traffic to a node should be less than the bandwidth of the egress link of the node. Belotti et al. [12] proposed a method to obtain the optimal routes so as to consider all TMs within certain constraints. Wang et al. proposed a method that considers both of the worst-case utilization and the link utilization calculated from the currently monitored TM [10]. Altin et al. set the OSPF weights considering all TMs [13]. Retvari et al. proposed a method to set the optimal routing strategy as the function of the TM considering all TMs [11]. However, if the constraints on TMs are set without traffic being monitored, many TMs satisfy the constraints, and routing configuration methods cannot find the routes that can accommodate all possible TMs.

Other methods that take into account all TMs under constraints obtained from current traffic information have also been proposed, where the constraints on current TMs are obtained from the amount of traffic on each link [8, 9]. Juva proposed a method to optimize the routes considering all TMs under the constraints on TMs obtained from the traffic amount on each link [8]. Kitahara et al. have proposed a method to set the constraints on TMs by using the bounds of link traffic and the bandwidth of the ingress and egress links [9]. By taking into consideration all TMs under the constraints, they calculate routes suitable for current traffic by only using the amount of traffic on each link without the impact of estimation errors. However, it takes a long time to calculate routes that take into account all TMs in large networks. We introduce hierarchical routing reconfiguration in this paper using aggregated traffic information to reduce the calculation time based on these methods.

Multi-path adaptive traffic engineering (MATE) [16] and traffic engineering with the XCP like protocol (TeXCP) [17] are other types of routing reconfiguration methods, where multiple paths are set from the source node to the destination node in advance. Then, the source node sets the ratio of traffic sent to each path based on current traffic. These approaches can immediately adapt the ratio of traffic when changes in traffic occur because the source nodes only consider their own ratio and do not require the optimization of the whole of the network. However, the source node is required to know the current situation in all paths, which entails a high monitoring overhead. The overhead to collect the required traffic data in our approach is significantly smaller than that with these methods because our approach only requires the amount of traffic on each link to be known, which can easily be collected by a central routing controller. In addition, these approaches may cause the temporal congestion, because the ratio of traffic sent to each path is set by each source node independently and traffic from different source node may concentrate on a certain link. On the other hand, our approach avoids the temporal concentration of traffic by considering upper bounds of link utilization based on the constraints on TMs obtained from the aggregated traffic information.

2.2. Hierarchical abstraction of networks

Methods of hierarchically abstracting networks were introduced by Kleinrock in 1976 [18], and are widely used (e.g., for QoS routing [19], ATM and Internet[20], and so on). They reduce the sizes or amounts of messages, the sizes of routing tables, and the route calculation time by hierarchically abstracting the network.

In these methods, the network is abstracted by transforming the network topology. Then, the information is attached to each node or each link in the transformed network topology. However, as discussed in Section 1, simply attaching information to each node or each link is insufficient to be used as inputs of the routing reconfiguration; the routing controllers cannot predict the utilization of links after route change.

We hierarchically divided the network into multiple areas to reduce the calculation time similarly to that achieved with these methods. However, our method is different from existing methods in that the traffic information aggregated by our method includes the constraints on TMs and the relation between the link utilization and the amount of traffic passing the area, instead of simply attaching information to nodes or links. By using this information, the routing controller obtains the possible TMs and the upper bounds of link utilization after the route change only from the aggregated traffic information.

3. Method of Routing Reconfiguration Taking into Consideration All Traffic Matrices

In this paper, the routes are calculated and reconfigured by the routing controllers based on the monitored or aggregated traffic information. Then, the routing controllers configures the nodes in the network according to the calculated routes.

In this paper, we focus on the method of calculating the suitable routes instead of the method of configuring the nodes within the calculated routes. The calculated routes can be set by configuring the nodes in the network with the technologies such as MPLS or OpenFlow. By using MPLS, the route of each lightpath is configured independently based on the calculated routes. Similarly, by using OpenFlow, the route of each flow is configured by setting the rules for each flow.

3.1. Methods of routing reconfiguration taking into consideration worst case traffic

This subsection explains the method of routing reconfiguration taking into consideration worst case traffic, which is proposed by Juva [8]. This method calculates the routes of the *flow*, which is defined as traffic from a certain source node to a certain destination node, taking into account all TMs under the constraints obtained from the amount of traffic on each link.

The set of the possible amounts of traffic for flow p is defined in this method as traffic v_p satisfying the following constraints for all links.

$$x_l^{\min} \leq \sum_{p \in P} f_{p,l} v_p \leq x_l^{\max} \tag{1}$$

where $x_l^{\min}$ and $x_l^{\max}$ indicate the lower and upper bounds for traffic on link l obtained by monitoring the amounts of traffic. P is the set of flows in the network, and v_p is the traffic rate for flow p. Here, $f_{p,l}$ is the portion of traffic for flow p passing link l when monitoring traffic.

Then, the routes are set to minimize the worst link utilization for all TMs under the constraints. That is, this method calculates $f_{p,l}^{\text{new}}$, which is the portion of traffic for flow p passing link l after the route change, with

$$\text{minimize} \max_{l \in L, t \in T} \sum_{p \in P} f_{p,l}^{\text{new}} v_p \tag{2}$$

where t is the TM whose element corresponding to flow p is v_p, T is the set of TMs satisfying the constraints, and L is the set of links.

Even though single linear programming for routing reconfiguration that takes into consideration all TMs has been proposed [21], it takes a long time to solve linear programming. Therefore, a heuristic method is used to obtain suitable routes.

3.2. Method of routing reconfiguration used in this research

In this paper, the routes were calculated by a heuristic method taking into account all TMs, which was based on the method proposed by Juva [8], but was different from the existing method in two ways. (1) The main goal of the method was to make the maximum link utilization less than threshold T_H instead of minimizing link utilization, because link utilization does not have a large impact on the quality of communication unless link utilization becomes larger than a certain threshold and congestion occurs. (2) We also introduced traffic whose routes could not be changed so as to be applied to the hierarchical routing reconfiguration explained in Section 4. In the hierarchical routing reconfiguration, the routing controllers of the upper layer cannot change the routes of the traffic whose source and destination nodes are within the same area of the lower layer. However, such traffic may also affect the link utilization. Thus, the routing controller should consider such traffic to avoid a large link utilization.

We assumed that flows could be split in the heuristic method and we split the flows equally into *subflows*, and calculated the routes for subflows. We also assumed that the routing reconfiguration was carried out before congestion occurred, and the traffic rate did not change significantly during the routing reconfiguration.

In this method, we first select flows that pass links whose utilizations are larger than threshold T_h. We denote the set of selected flows as P^{target}. Then, we calculate the routes of subflows for the selected flows. The routes of subflow p^{sub}, which is the subflow for flow p, are calculated in five steps.

Step 1: Construct topology G from the network where the routes are calculated.

Step 2: Calculate the route for p^{sub} on topology G with the Dijkstra algorithm. If there is no route for p^{sub}, keep the previous route for p^{sub} and go to Step 5. Otherwise go to Step 3.

Step 3: Check whether the upper bounds for the utilizations of links on the route calculated in Step 2 are less than threshold T_H. If yes, designate the routes calculated in Step 2 as the new routes for p^{sub} and go to Step 5. Otherwise, go to Step 4.

Step 4: Remove links having upper bounds for utilizations larger than T_H from topology G, and go to Step 2.

Step 5: End.

These steps consider the upper bounds for link utilizations, and reconfigure the routes only when the reconfiguration does not cause the links whose upper bounds of the utilization are larger than T_H. Thus, there is a case that no suitable routes for the flow can be found, and the route of the flow cannot be changed. Even in this case, the largest link utilization may be reduced by changing the routes of the other flows whose routes can be changed without causing the link utilization larger than T_H.

In the above steps, the large number of hops can be avoided by keeping the previous route if the number of hops of the found route is larger than the predefined threshold (e.g., γ times the number of hops of the shortest path of the flow), though our evaluation does not limit the number of hops.

The upper bounds for the link utilizations required in Step 3 mentioned above are calculated with the linear programming described in Appendix A, which allows for the routes of some flows on each link to be changed. We use the following steps to reduce the calculation time by reducing the number of calculations in linear programming in our method of calculating routes.

We calculate the upper bound for the amount of traffic $v_p^{\max}$ for each flow p in P^{target} in advance by calculating the linear programming described in Appendix B in our method of route calculation. We then calculate $\hat{x}_l$ defined by

$$\hat{x}_l = \frac{1}{b_l}\left(x_l^{\mathrm{before}} + \sum_{p \in P_l^{\mathrm{after}}} f_{p,l}^{\mathrm{new}} v_p^{\max}\right) \tag{3}$$

where x_l^{before} is the amount of traffic on link l before the route changes, P_l^{after} is the set of flows newly passing link l after the route changes and $f_{p,l}^{\mathrm{new}}$ is the fraction of traffic for flow p newly passing link l. $\hat{x}_l$ is larger than the actual link utilization after the route changes unless the traffic changes significantly

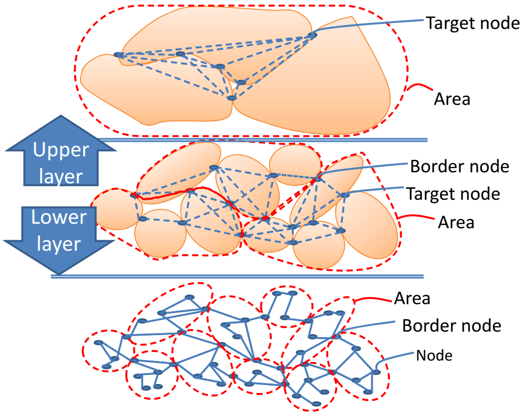


Figure 2: Hierarchically constructed areas

during the change in route, because $\hat{x}_l$ is the total traffic rate passing the link before the route changes and the upper bound for the amount of traffic for flows that newly pass the link. Therefore, if $\hat{x}_l$ is less than T_H, we can recognize that the utilization of link l becomes less than T_H without calculating the linear programming described in Appendix A. When $\hat{x}_l$ is larger than T_H, we check the upper bounds for link utilizations after we obtain accurate upper bounds for the utilization of link l by calculating the linear programming described in Appendix A.

4. Hierarchical Routing Reconfiguration and Aggregation of Traffic Information

4.1. Overview of hierarchical routing reconfiguration

We hierarchically divide the network into multiple areas in our approach. Figure 2 overviews the hierarchically constructed areas, where we divide the network into multiple areas in the lowest layer so that each link belongs to one of the areas. Some of the nodes become the border between the areas. We call the nodes at the border of the areas *border nodes* for the areas. Each area has multiple nodes including border nodes and non-border nodes. We call nodes included in each area *target nodes* for the area.

The areas of the upper layers are constructed from the multiple areas of the lower layer, and they include only nodes that are border nodes in the lower layer. That is, only the border nodes for the areas of the lower layer become the target nodes for the areas of the upper layer.

The routing controller is deployed for each area of each layer, and changes the routes within the area based on the traffic information. Figure 3 outlines the traffic information used by the routing controller of each area, where we can see our approach uses two kinds of traffic information. The first is traffic information within the area. The routing controller identifies links with utilizations larger than the threshold by using this kind of information, which is obtained in the lowest layer by monitoring the amounts of traffic on the links. This kind of information is obtained in the upper layer as aggregated traffic information generated from traffic information within the areas of the lower layer.

The second kind of information is traffic information from *nearby areas* that belong to the same area at the upper layer. The routing controller checks whether there are links with large utilization in nearby areas by using this kind of information, which is generated from traffic information within the areas of the upper layer.

The routing controller of each area changes three routes, i.e., (1) the routes for subflows passing links within the area, (2) border nodes passed by subflows whose source nodes are within the area, and (3) border nodes passed by subflows whose destination nodes are within the area. The routing controller uses monitored or aggregated traffic information to avoid link utilizations larger than threshold T_H when changing routes, and it calculates new routes over the topology constructed of the target nodes in the area maintained by the controller and the target nodes of the upper layer. The details on the routing reconfiguration method using aggregated traffic information are described in Subsection 4.3.

The routing controller of the lower layer in our approach changes routes within a small area by using detailed traffic information on the area, and the routing controller of the upper layer changes routes of a large area by using aggregated traffic information. The routing controller can be implemented on ordinary computers such as workstations or server computers. We may deploy a server acting as a routing controller for each area or deploy a single server that hosts the routing controllers of all areas.

In our approach, each routing controller of each layer reconfigures the routes only when a link whose utilization is larger than the threshold is found. When such a link is found, the routing controller reconfigures the routes considering the worst link utilization calculated from all TMs satisfying the constraints on TMs included in traffic information, and avoids the reconfiguration which causes the worst link utilization larger than the threshold. Therefore, the reconfiguration of each layer does never cause the link utilization larger than the threshold, and no additional reconfiguration is required after the reconfiguration of routes in some layers.

None of the routing controller for each layer requires a large amount of traffic information or a long calculation time. If we divide the network into areas so that each area of each layer includes N/R nodes and the aggregated traffic information includes L/R links, the calculation time to obtain the largest link utilization for all TMs required by the routing reconfiguration method is $\mathrm{O}(N^8L^2/R^{10})$, since the largest link utilization is obtained by the L/R linear programings that have N^2/R^2 variables and L/R constraints. Thus, the routing controller of each layer of each area calculates routes within a short time, and we reduce link utilization immediately after traffic changes by combining the routing reconfiguration of each layer.e

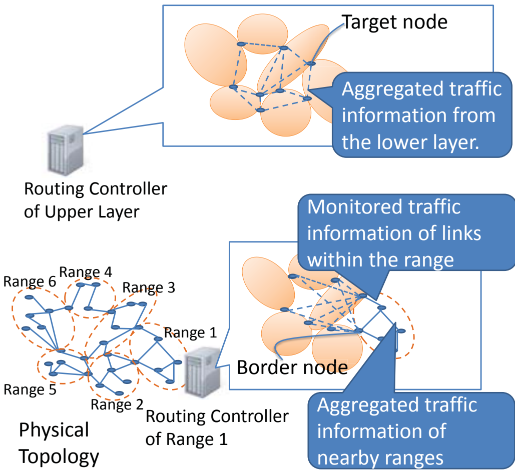


Figure 3: Traffic information used by routing controller

4.2. Traffic information aggregation

We propose a method of aggregating traffic information in this subsection that generates two kinds of aggregated traffic information, i.e., that from the area of the lower layer and that from nearby areas.

Both kinds of aggregated traffic information are generated in the following steps. First, our method of aggregating traffic information selects links whose utilization may become larger than threshold T_H after the routing controller using the aggregated traffic information changes the routes. Then, we generate the aggregated traffic information for each of the selected links. The generated traffic information includes constraints on traffic whose routes can be changed by the routing controller using aggregated traffic information. Information to calculate the upper bounds for utilizations of the selected links is also included in the generated traffic information.

The rest of this subsection explains how to select links whose information is included in aggregated traffic information, and how to generate aggregated traffic information for each of the selected links.

4.2.1. Selection of links

Case of aggregation of information to upper layer. We select the link with the largest utilization from links passed by each subflow between each border node pair. This is because the routing reconfiguration in the upper layer only increases or decreases the amount of traffic between each border node pair in the lower layer. The routing reconfiguration of the upper layer does not have a large impact on the order of link utilization between links passed by each subflow. Thus, the link with the largest utilization is the most likely to have utilization greater than threshold T_H of the links passed by a subflow.

Case of aggregation of information to lower layer. The aggregated traffic information from nearby areas is generated from the traffic information of the area of the upper layer, and is used by the routing controller of the lower layer. The main impact of the routing reconfiguration of the lower layer on the areas of the upper layer is only on the changes in the amount of traffic from or to areas where the routes are changed. Thus, the routing controller of the lower layer only requires traffic information on links passed by subflows from or to the areas. When generating aggregated traffic information used by the routing controller of an area of the lower layer, we select the link with the largest utilization from links passed by each subflow from or to the areas of the lower layer, similar to the case of aggregated information to the upper layer.

A common link is selected for multiple subflows in the above steps, and the number of selected links is much smaller than the number of links within the area in most cases.

4.2.2. Generation of aggregated traffic information

We generate the following aggregated traffic information for the selected links.

b_l Bandwidth of link l

$x_l^{\mathrm{max\_all}}$ Upper bound for total amount of traffic on link l

$x_l^{\mathrm{min\_all}}$ Lower bound for total amount of traffic on link l

x_l^{max} Upper bound for amount of traffic of flows on link l whose routes can be changed by the routing controller using aggregated information

x_l^{min} Lower bound for amount of traffic of flows on link l whose routes can be changed by the routing controller using aggregated information

P_l Set of subflows passing link l

$f_{p,l}$ Portion of traffic of subflow $p \in P_l$ passing link l

$f_{a,b,l}^{\mathrm{in}}$ Portion of traffic between nodes a and b passing link l, where nodes a and b are included in set of node pairs specified by the routing controller using aggregated traffic information.

We obtain the following constraints from this aggregated traffic information on the amount of traffic v_p for flow p whose routes can be changed by the routing controller using aggregated traffic information.

$$x_l^{\mathrm{min}} \leq \sum_{p \in P_l} f_{p,l} v_p \leq x_l^{\mathrm{max}}$$

The upper bound for the utilization of link l after the route changes is also calculated with the linear programming described in Appendix A by using this aggregated information. The details on the routing reconfiguration method using aggregated traffic information are explained in Subsection 4.3.

Of the above information, $x_l^{\mathrm{max\_all}}$ and $x_l^{\mathrm{min\_all}}$ are obtained from the monitored traffic rates on link l. x_l^{max} and x_l^{min} indicate the upper and lower bounds for the traffic of subflows whose routes can be changed by the routing controller using aggregated traffic information. The routing controller of the upper layer only changes the routes of subflows that pass multiple areas of the lower layer, and it cannot change the routes of subflows whose source and destination nodes are within the same area of the lower layer. The routing controller of the lower layer also cannot change the routes of subflows that do not traverse the areas maintained by the controller. We establish a set of subflows whose routes can be changed by the routing controller using aggregated traffic information by considering these points. We then calculate x_l^{max} and x_l^{min} with the linear programming described in Appendix C.

$f_{p,l}$ is obtained from the information on routes within the area. Similarly, $f_{a,b,l}^{\mathrm{in}}$ for nodes a and b, which are included in the set of node pairs specified by the routing controller using aggregated traffic information, is also obtained from the information on routes.

4.3. Routing reconfiguration using aggregated traffic information

This subsection explains how to carry out the routing reconfiguration method described in Section 3 when using aggregated traffic information by constructing a suitable topology G to use this information. We should then check whether the upper bounds for utilizing links are less than T_H based on the aggregated traffic information.

4.3.1. Abstracted topology used by routing controller

The routing controller using aggregated traffic information calculates routes on the abstracted topology constructed of target nodes of the area maintained by the controller, which are the border nodes of the lower layer, and target nodes of the upper layer as shown in Figure 4. The method in Section 3 calculates the routes of subflows by setting the abstracted topology as G, which indicates the set of target nodes passed by each subflow. The routes between the target nodes connected in the abstracted topology are set according to the routes calculated in the lower layer.

As our method to aggregate traffic information does not depend on the abstracted topology used to calculate routes, any method to abstract the topology can be used. In our evaluation described in Section 5, we construct the abstracted topology by adding links between all the target node pairs that belong to the same area in the lower layer. We also add a link between the border nodes of the areas maintained by the controller and target nodes of the upper layer.

When calculating the routes of subflows passing the area, we calculate the path from the source target node, which is the first node within the area passed by the subflow, to the destination target node, which is the last node within

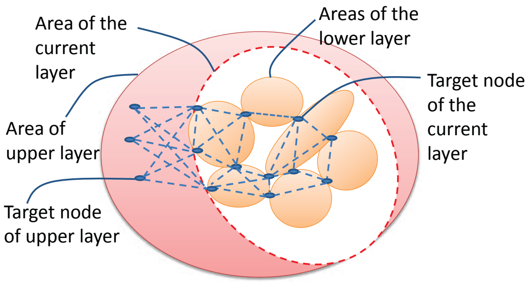


Figure 4: Abstracted topology used by routing controller

the area passed by the subflow. When we change the border node passed by the subflow from the area, we calculate the path from the source target node to the destination target node of the upper layer. Similarly, when we change the border node passed by the subflow to the area, we calculate the path from the source target node of the upper layer to the destination target node. We can select the border node passed by the subflow by taking into consideration the link utilization both within and outside the area by calculating the routes on the topology including the target nodes of the upper layer.

4.3.2. Checking link utilization based on aggregated traffic information

The routing controller calculates the upper bound for utilization of each link with the linear programming described in Appendix A when calculating routes, where we need to set $f^{\mathrm{new}}_{p,l}$, which indicates the portion of the amount of traffic for flow p passing link l after the routing reconfiguration.

The routing controller using aggregated traffic information sets the routes between target nodes according to the routes calculated in the lower layer. Thus, if the subflow of flow p newly passes the link between nodes a and b on the abstracted topology, the portion of traffic for flow p passing link l is

$$f^{\mathrm{new}}_{p,l} = f^{\mathrm{new}}_{p,a,b} f^{\mathrm{in}}_{a,b,l}$$

where $f^{\mathrm{new}}_{p,a,b}$ is the portion of traffic for flow p passing nodes a and b, and $f^{\mathrm{in}}_{a,b,l}$ is the portion of traffic between nodes a and b passing link l. $f^{\mathrm{new}}_{p,a,b}$ is the variable determined by the routing reconfiguration method. $f^{\mathrm{in}}_{a,b,l}$ is obtained from aggregated traffic information by setting nodes a and b as a node pair whose routing information should be included in the aggregated traffic information. If no routes of any subflows for flow p are changed, we set

$$f^{\mathrm{new}}_{p,l} = f_{p,l}$$

where $f_{p,l}$ is the value included in aggregated traffic information.

Our method aggregates traffic information so that the aggregated information includes information on links whose utilizations have a high possibility of becoming larger than threshold T_H. Therefore, we avoid link utilizations larger than threshold T_H by only checking the upper bounds for utilization of links included in aggregated traffic information. The calculation time is significantly reduced by using aggregated traffic information, because aggregation reduces the number of links whose utilization needs to be checked and the number of constraints for TMs. The impact of using aggregated traffic information is discussed in Section 5.

5. Evaluation

We evaluated the hierarchical routing optimization using aggregated traffic information, which is discussed in this section.

First, we used the small network topologies, whose number of nodes is from 46 to 161, to evaluate the hierarchical routing optimization in the cases of various traffic or various parameters. By using such topologies, we nvestigated the necessity for dynamic routing reconfiguration taking into consideration all TMs under the constraint obtained from the current amount of traffic on each link in our evaluation. We then demonstrated that the hierarchical routing reconfiguration can calculate the suitable routes without congestion immediately after traffic changes. We also illustrated that the hierarchical routing reconfiguration using aggregated traffic information could work regardless of the size of the areas or the topology. Then, we demonstrated that our hierarchical routing optimization can calculate the suitable routes immediately even in a larger network.

5.1. Method to divide areas used in our evaluations

In the hierarchical routing reconfiguration, any method to divide the network into rages can be used. In our evaluation, we used a simple method that divides the network by the following steps.

Construction of the areas of the lowest layer. We first selected the node having the largest number of links from the nodes that did not belong to the other areas. We then chose n nodes that did not belong to the other areas and that were the nearest to the selected nodes, where n is a predefined value indicating the number of nodes in an area. All areas of the lowest layer are constructed by continuing the above steps until all nodes belong to one of the areas.

Construction of the areas of the other layers. The areas of the other layers were constructed by combining the adjacent areas of the lower layer until the number of target nodes reached more than n.

Even though there may be more sophisticated methods (e.g., those using hierarchical structures of networks), this simple approach was sufficient for our evaluation, because our aim was to discuss the effectiveness of hierarchical routing reconfiguration using aggregated traffic information and did not focus on the effectiveness of hierarchically constructed areas.

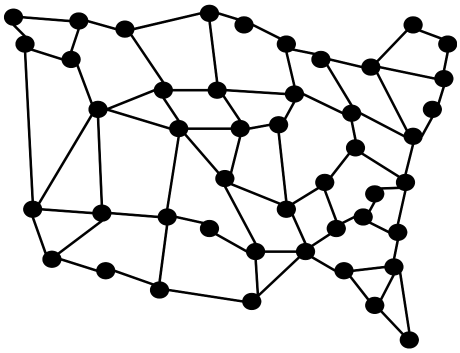


Figure 5: USANet Topology

5.2. Necessity for dynamic routing reconfiguration taking into consideration link utilization

5.2.1. Environments

Topology. We used the USANet Topology (46 nodes and 76 links) [22] shown in Figure 5 in this evaluation, which is the one of the models for ISP topologies. We divide the USANet Topology into two layers for hierarchical routing reconfiguration by setting n to 6.

Traffic and initial routes. The initial TM in this evaluation was generated randomly to follow the lognormal distribution according to the results obtained by Nucci et al. [23]. We used the same parameters for the lognormal distribution as Nucci et al. did [23] and scaled the TM so that the generated traffic could be accommodated in the topology used in our evaluation. The initial routes were set to minimize the link utilization for initial traffic. Then, we generated sudden changes by randomly regenerating the current TM to follow the lognormal distribution in the same way as the initial TM. We assumed the traffic would be constant during the routing reconfiguration. We generated 55 patterns of traffic changes in this evaluation.

Routing reconfiguration methods used in this evaluation. We compared three methods to demonstrate the necessity of dynamic routing reconfiguration by taking into consideration all TMs under the constraint obtained by the current amount of traffic on each link.

Hierarchical routing reconfiguration In this method, we dynamically changed the routes taking into account all TMs under the constraints obtained from monitored or aggregated traffic information, where the routes were calculated with the method described in Section 3. We set the threshold T_H so that the maximum utilization could be minimized by the routing reconfiguration. We reconfigured routing for the upper layer once every four times for the routing reconfiguration of the lower layer.

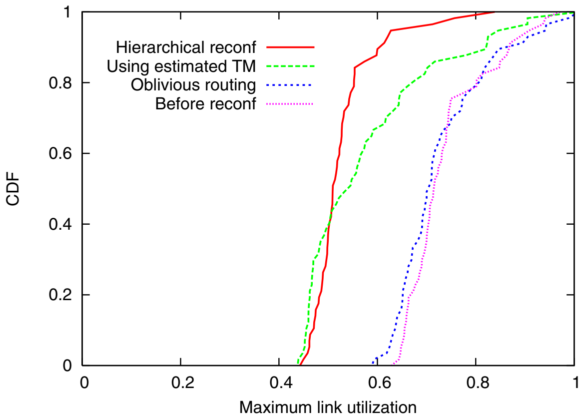


Figure 6: Maximum link utilization after 30 Steps (USANet Topology and number of nodes in each area = 6)

Route optimization using estimated amount of traffic In this method, we optimized the routes by linear programming that minimized the maximum link utilization based on TM estimated with the tomogravity method [24]. Using the estimated TM is one approach used to reconfigure the routes when the amount of traffic cannot be directly measured because of the large scale of the network. We demonstrated the effectiveness of considering all TMs by comparing the hierarchical routing reconfiguration with this method.

Oblivious routing In this method, we set the static routes that were optimized to minimize the upper bound for maximum link utilization by taking into consideration all possible TMs. The set of possible TMs in this evaluation was set by using the hose model, which is similar to the method proposed by Applegate and Cohen [21], where the possible amount of traffic is constrained by the bandwidths of the ingress and egress links; the total traffic from a node should be less than the bandwidth of the ingress link of the node, and the total traffic to a node should be less than the bandwidth of the egress link of the node. The bandwidths of the ingress and egress links in our evaluation for each node were set to the maximum amount of traffic from or to the node in all patterns of traffic we generated.

Metric. We compared the maximum link utilization at each step to investigate the necessity for dynamic routing reconfiguration by taking into account all TMs under the constraints obtained from the monitored amount of traffic on each link.

5.2.2. Results

Figure 6 plots the cumulative distribution for maximum link utilization achieved by the routing reconfiguration methods at the 30th step, where we have also plotted the cumulative distributions for maximum link utilizations before the routing reconfiguration and that of static oblivious routing. According to this figure, static oblivious routing cannot achieve sufficiently low link utilizations when traffic changes significantly. That is, we need to dynamically change routes to accommodate traffic that changes unpredictably.

The figure also indicates that route optimization using the estimated amount of traffic cannot reduce maximum link utilization sufficiently in some cases. There are even cases where maximum link utilization becomes larger than that before routing reconfiguration. This is caused by the estimation errors included in the amount of traffic.

Figure 7 plots maximum link utilizations at each step for a certain case. The figure indicates that route optimization using the estimated traffic makes maximum link utilization significantly larger than that in many previous steps because of estimation errors in the amount of traffic. However, the routing reconfiguration method that takes into account all TMs does not make maximum link utilization significantly larger than that in previous steps and reduces the maximum link utilization as steps are completed even when we use aggregated traffic information. This is because this method takes into consideration the upper bounds for link utilizations calculated from traffic information. We can avoid route changes that cause large link utilization by taking into account the upper bounds for link utilizations.

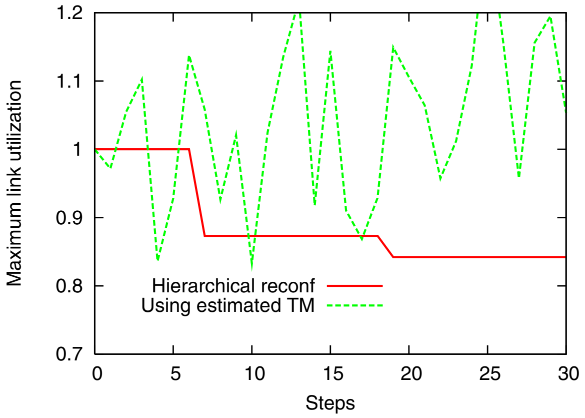


Figure 7: Maximum link utilizations after each step (USANet Topology, number of nodes in each area=6, and normalized by initial utilization)

5.3. *Effectiveness of hierarchical routing reconfiguration with aggregated traffic information*

5.3.1. *Environments*

Routing reconfiguration methods used in this evaluation. We compared three methods to demonstrate the effectiveness of hierarchical routing reconfiguration with aggregated traffic information.

Hierarchical routing reconfiguration using aggregated traffic information We hierarchically divided the network into areas and the routing controller carried out routing reconfiguration for each area of each layer. The routing controller of each area of each layer used monitored or aggregated traffic information and calculated suitable routes by taking into consideration all TMs that satisfied the constraints obtained from traffic information. We carried out routing reconfiguration for the upper layer once every four times for the routing reconfiguration of the lower layer in this evaluation, which was similar to that described in Subsection 5.2.

Routing reconfiguration using traffic information only within area We divided the network into areas and carried out routing reconfiguration for each area similarly to the hierarchical routing reconfiguration. However, in this method the routing controller only used link utilization within the area maintained by the controller, and changed routes within the area to mitigate congestion within the area. We clarified whether aggregated traffic information was required by comparing the hierarchical routing reconfiguration with this method.

Routing reconfiguration using all traffic information We calculated routes by using the method described in Section 3 without dividing the network into areas. We demonstrated the effectiveness of hierarchically dividing the network into areas and using aggregated information by comparing hierarchical routing reconfiguration with this method.

Metrics.

Calculation time We compared the time required to calculate the routes in each step. When we ran each routing reconfiguration method, we measured the calculation time by using a computer with a 2.53 GHz Intel Xeon Processor (E5540) and 64 GB of RAM. Linear programming was solved with CPLEX 12.1 and the other processes were implemented in C++ and compiled by gcc 4.4.2 in each method.

Link utilization achieved within certain time We compared link utilization achieved within a certain time to clarify the time required to reduce link utilization to less than the threshold. We assumed that each step of routing reconfiguration would take $T^{\mathrm{calc}} + T^{\mathrm{monitor}}$ s where T^{calc} is the time required to calculate the routes and T^{monitor} is the time required to monitor the amount of traffic on each link. We set T^{calc} to the measured maximum calculation time for each step and T^{monitor} to 30 s. Our approach reduces the amount of traffic information required by each routing controller, and may reduce the time interval required to monitor the amount of traffic T^{monitor}. In our evaluation, we use the same value of T^{monitor} in all methods to show the advantages of our method even when the time interval required to monitor the amount of traffic is not reduced. Link utilization achieved within a certain time was calculated as the minimum of maximum link utilization achieved with the routing reconfiguration method at various settings for threshold T_H.

Topology. We used the USANet Topology (46 nodes and 76 links), which was similar to that in Subsection 5.2. We divided the USANet Topology into two layers by setting n to 6 for hierarchical routing reconfiguration and the routing reconfiguration only using traffic information within an area, so that each range of each layer has alternative routes for the flows.

Traffic and initial routes. We generated similar traffic and initial routes to those described in Subsection 5.2.

5.3.2. Results

Table 1 summarizes the maximum time required to calculate the new routes and indicates that the calculation times required by routing reconfiguration using aggregated traffic information or routing reconfiguration only using traffic information within an area are significantly shorter than routing reconfiguration that takes into consideration all link utilizations. This is because aggregating or dividing the network reduces the number of links whose utilizations need to be checked and the number of constraints for TMs.

Table 1: Route calculation time (USANet Topology and number of nodes in each area=6)

		Maximum Calculation Time [sec]
Hierarchical reconf using aggregated information	Lowest layer	3.25
	Top layer	8.64
Reconf only using information within area		0.23
Reconf using all information		664.54

Figure 8 compares the cumulative distributions of maximum link utilizations achieved with each routing reconfiguration method within 20 min, where we have also plotted the cumulative distributions of maximum link utilizations achieved with hierarchical routing reconfiguration within 10 min and that of maximum link utilizations before routing reconfiguration.

The figure indicates that the routing reconfiguration only using traffic information within an area cannot sufficiently reduce link utilization. This is because the utilization of all links within a certain area increases because of large traffic changes. The hierarchical routing reconfiguration, on the other hand, can significantly reduce maximum link utilization within 10 min, because routing controller for the upper layer changes the routes of flows passing multiple areas in this method. As a result, we can reduce link utilization that cannot be reduced by route changes only within an area. Moreover, the figure indicates that the hierarchical routing reconfiguration can reduce the maximum link utilization faster than the routing reconfiguration using all traffic information, because calculating new routes with routing reconfiguration taking into account all link utilizations takes more than 10 min, as shown in Table 1.

Figure 9 plots the average for maximum link utilizations achieved within a certain time normalized by initial link utilization. As shown in this figure, the routing reconfiguration using aggregated traffic information reduces maximum link utilizations immediately after traffic changes, while routing figuration that takes into account all link utilizations takes more time to reduce the maximum link utilizations because of the long calculation time.

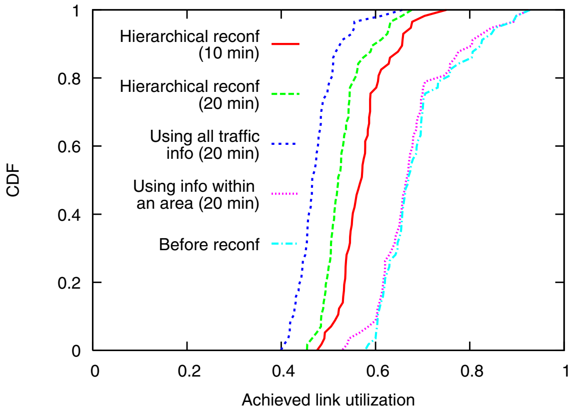


Figure 8: Achieved link utilizations (USANet Topology and number of nodes in each area=6)

5.4. Impact of sizes of areas

This subsection discusses our evaluation of the impact of the sizes of areas on the performance of the hierarchical routing reconfiguration by changing the number of nodes within each area. Even though we used a simple method of dividing the network into areas, the discussion described in this subsection can be applied to cases where more sophisticated methods of dividing networks into areas are used, because the discussion is not based on methods of dividing networks but on the sizes of the areas.

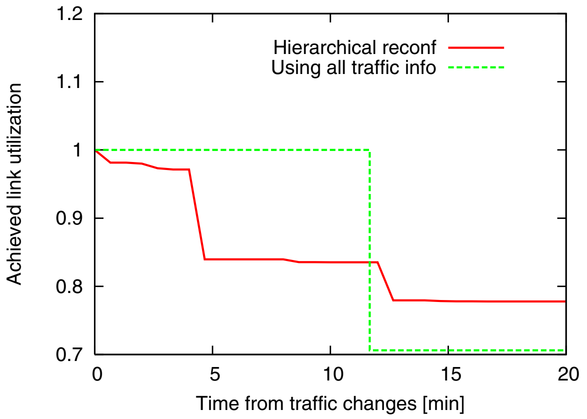


Figure 9: Average for maximum link utilization achieved by certain time (USANet Topology, number of nodes in each area=6, and normalized by initial utilization)

5.4.1. Environments

Compared hierarchical routing methods used in this evaluation. We compared hierarchical routing reconfiguration using aggregated traffic information with areas of different sizes, where we changed the number of nodes in each area of the lowest layer from 6 to 8. We divided the network into two layers in all cases.

Topology. We used the USANet Topology (46 nodes and 76 links), which was similar to that described in Subsection 5.2, in this evaluation.

Traffic and initial routes. We generated similar traffic and initial routes to those described in Subsection 5.2.

Metrics. This subsection discusses our comparison of the calculation time and link utilization achieved within a certain time similar to that in Subsection 5.3.

5.4.2. Results

Figure 10 summarizes the average for normalized maximum link utilizations achieved by a certain time when the number of nodes in each area of the lowest layer is changed. As we can see in this figure, hierarchical routing reconfiguration reduces maximum link utilization immediately after traffic changes regardless of the sizes of areas.

Table 2 summarizes the maximum time required to calculate routes once for 55 patterns in generated traffic changes. The table indicates that as the number of nodes in each area of the lowest layer increases, the time for the lowest layer to calculate the route increases, while the calculation time for the top layer decreases. This is because the calculation time of linear programming to obtain the upper bounds for link utilization increases as the number of target nodes and the number of links whose upper bounds for utilizations need to be calculated

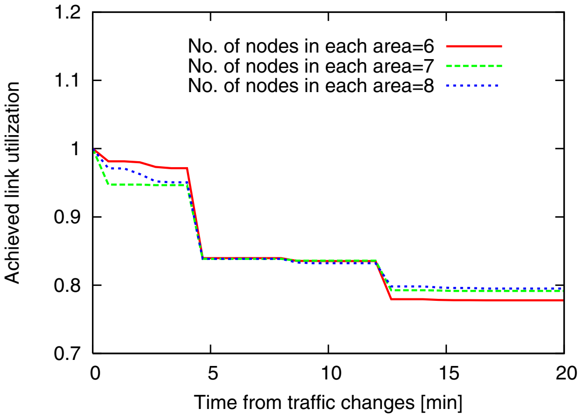


Figure 10: Average for maximum link utilization achieved by certain time (USANet Topology and number of nodes in each area is changed)

increases. When the sizes of areas at the lowest layer reduce, the number of target nodes at the upper layer increases. The increase in the number of target nodes increases the number of links included in aggregated traffic information used at the top layer, and leads to increased calculation time for the top layer.

Even in the case that the number of nodes in each area is set to 8, we can calculate the routes within 17 s. Moreover, the calculation time depends on the spec of the computer, and can be reduced by using more powerful computers. However, even if more powerful computer is used, the above tendency that the increase in the number of target nodes leads to increased calculation time is true. Therefore, we should set the number of nodes in each area such that no areas of any layers include many target nodes to avoid long calculation times.

Table 2: Route calculation time (s) (USANet Topology and different number of nodes included in each area)

	No. of nodes in each area of lowest layer		
	6 nodes	7 nodes	8 nodes
Lowest Layer	3.25	7.21	16.96
Top Layer	8.64	3.86	3.41

5.5. Impact of topology

We used USANet Topology in all the previous evaluations. We also used other topologies to demonstrate that our approach works regardless of the topology.

5.5.1. Environments

Topology. We used two ISP topologies in this evaluation, i.e., the Ebone topology (87 nodes and 161 links) and the Tinet topology (161 nodes and 328 links), which are router-level ISP topologies measured by Rocketfuel [25]. The Ebone topology was divided into two layers by setting n to 10 and the Tinet topology was divided into three layers by setting n to 11 for the routing reconfiguration using aggregated traffic information so that each range of each layer has alternative routes for flows.

Traffic and initial routes. We generated similar traffic and initial routes to those described in Subsection 5.2. We used 55 traffic patterns in this evaluation for the Ebone topology and 28 for the Tinet topology.

Compared routing reconfiguration methods used in this evaluation. We used the hierarchical routing reconfiguration and the routing reconfiguration using all traffic information to demonstrate the effectiveness of hierarchically dividing the network and using aggregated information.

Metrics. We used two metrics that were similar to those in Subsection 5.3, i.e., calculation time and link utilization achieved within a certain time. When we ran each routing reconfiguration method, we measured the calculation time by using a computer with a 2.53 GHz Intel Xeon Processor (E5540) and 64 GB of RAM.

5.5.2. Results

We compared the maximum times required to calculate new routes in the Ebone and Tinet topologies as listed in Table 3, which were similar to those in Table 1. The results indicate that hierarchical routing reconfiguration reduces the calculation time both in the Ebone and the Tinet topologies.

Table 3: Maximum calculation time [s]

		Ebone	Tinet
Hierarchical reconf using aggregated information	Lowest layer	88.56	18.10
	Middle layer	-	97.94
	Top layer	28.80	380.76
Routing reconf using all information		12421.20	80649.20

Figure 11 compares the cumulative distributions of maximum link utilizations achieved by the hierarchical routing reconfiguration, where we have also plotted the cumulative distributions of maximum link utilizations before routing reconfiguration. As we can see from Fig.11, the routing reconfiguration using aggregated traffic information can reduce maximum link utilization significantly within 20 min in both topologies, similar to the results in Fig. 8.

We have also plotted the average for maximum link utilizations achieved by a certain time in Fig. 12, which is similar to that in Fig. 9. The figure indicates that the routing reconfiguration using aggregated traffic information can reduce maximum link utilization within 20 min in both topologies, while calculating new routes with the routing reconfiguration using all traffic information once takes more than 200 min in the Ebone topology and more than 22 h in the Tinet topology, as has been summarized in Table 3.

5.6. Demonstration of the hierarchical routing optimization in a large network

Finally, we demonstrate that our hierarchical routing optimization using aggregated traffic information works even in a large network.

5.6.1. Environments

Topology. We generated the network topology with 500 nodes and 1250 bidirectional links by using the Watts-Strogatz model [26]. In this model, the network topology is generated by adding links between nodes a and b with the following probability.

$$\alpha exp\left(\frac{d(a,b)}{\beta d^{\max}}\right)$$

where α and β is parameters, $d(a,b)$ is a distance between nodes a and b, $d^{\max}$ is the largest distance among all node pairs. The parameter α controls the number of links. The parameter β controls the tendency to connect the nearby nodes, and setting β to a small value makes the nearby nodes connected closely, while setting β to a large value adds the connections between far nodes. Because setting α so that the required number of links are connected is difficult, we generated the network topology by placing the nodes randomly and continuing to add one link with the above probability by setting α to 1 until the sufficient number of links are added. β is set to 0.1 so that the nearby nodes are closely connected, since the nearby nodes are closely connected and construct the modules in the actual ISP topologies[27]. Then we divide the generated network topology into four layers by setting n to 18.

Traffic and initial routes. We set the initial routes as the shortest path between the nodes. Then, we generated the similar traffic to those described in Subsection 5.2. We used 13 traffic patterns.

Metrics. The aim of this subsection is to demonstrate that our hierarchical routing optimization using the aggregated traffic information calculate the suitable routes immediately. Thus, we used link utilization achieved within a certain time defined in Subsection 5.3. When we ran each routing reconfiguration method, we used a computer with 2.40 GHz Intel Xeon Processor (E7-4870) and 512 GB of RAM.

5.6.2. Results

Figure 13 shows the cumulative distributions of maximum link utilizations achieved by the hierarchical routing reconfiguration, where we have also plotted the cumulative distributions of maximum link utilizations before routing reconfiguration. We have also plotted the average for maximum link utilizations achieved by a certain time in Figure 14. As we can see from these figures, the routing reconfiguration using aggregated traffic information can reduce the maximum link utilization significantly within 20 min in most cases, and within 40 min even in all cases in the network topology with 500 nodes.

This figure also indicates that the maximum link utilizations cannot be reduced significantly after 40 minutes. We change the routes so as to make the upper bounds of link utilization small. However, the routing reconfiguration within 40 minutes already reduces the link utilizations of the bottleneck links, and increases the link utilizations of the other links. As a result, any alternative routes for any flows can no longer reduce the upper bounds of link utilization, and no routes are changed after 40 minutes.

6. Conclusion

We introduced hierarchical routing reconfiguration that reduced the calculation time, where we hierarchically divided the network into areas; the areas of the lowest layer were constructed from a small number of nodes and those of the upper layer were constructed from multiple areas of the lower layer.

Then, we proposed a traffic information aggregation suitable to the hierarchical routing reconfiguration. In this aggregation method, we first select links whose utilization may become large after the routes have changed at each layer. Then, we calculate the ratio of the amount of traffic of flow between source and destination nodes passing the link for each of the selected links, and the upper and lower bounds for the traffic of flows passing the link whose routes can be changed by the routing controller using aggregated traffic information. Finally, we generate aggregated traffic information including the calculated ratio of traffic and the upper and lower bounds for traffic whose routes can be changed for selected links. We can obtain constraints on TM by using this aggregated traffic information and identify links whose utilizations may become large after the route has changed.

We also introduced a routing reconfiguration method using the aggregated traffic information, focusing on the calculation of the routes. In this method, each routing controller reconfigures the routes within its area based on the monitored or aggregated traffic information. The routing controller calculates the routes so as to avoid the large link utilization by considering the all TMs satisfying the constraints obtained from the aggregated traffic information. In addition, this method considers the traffic whose routes cannot be changed by the routing controller because the routing controller cannot change the routes of the traffic whose source and destination nodes are within the same area of the lower layer.

We demonstrated that the hierarchical routing reconfiguration using our aggregated traffic information reduces the link utilization immediately after the traffic changes without causing the large link utilization through numerical evaluation using various topologies.

In this paper, we divided the network into areas by using a simple approach, which designated n nearest nodes as those that constructed an area in our evaluation. However, there may be more sophisticated methods that we could use to divide the network into areas, which we intend to address in future research projects.

Appendix A. Linear Programming to Obtain Upper Bounds for Link Utilization

Inputs

P Set of flows whose routes can be changed.

L Set of links.

$f_{p,l}$ Portion of amount of traffic for flow $p \in P$ passing link $l \in L$ before routing reconfiguration.

$f_{p,l}^{\text{new}}$ Portion of amount of traffic for flow $p \in P$ passing link $l \in L$ after routing reconfiguration.

x_l^{min},x_l^{max} Lower and upper bounds for amount of traffic passing link l whose routes can be changed.

$x_l^{\text{min\_all}}$,$x_l^{\text{max\_all}}$ Lower and upper bounds for amount of traffic passing link l.

Variables

v_p Amount of traffic for flow p.

v_l^{stat} Total amount of traffic on link l whose routes cannot be changed.

Objective

Maximize link utilization of link l.

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

Constraints

$$\forall l \in L\text{: } x_l^{\text{min}} \leq \sum_{p \in P} f_{p,l} v_p \leq x_l^{\text{max}}$$

$$\forall l \in L\text{: } x_l^{\text{max\_all}} \leq \sum_{p \in P} f_{p,l} v_p + v_l^{\text{stat}} \leq x_l^{\text{max\_all}}$$

Appendix B. Linear Programming to Obtain Upper Bounds for Amount of Traffic of Flows

Inputs

P Set of flows whose routes can be changed.

L Set of links.

$f_{p,l}$ Portion of amount of traffic for flow $p \in P^{\text{current}}$ passing link $l \in L$.

$x_l^{\min}$,$x_l^{\max}$ Lower and upper bounds for amount of traffic passing link l whose routes can be changed.

$x_l^{\text{min\_all}}$,$x_l^{\text{max\_all}}$ Lower and upper bounds for amount of traffic passing link l

Variables

v_p Amount of traffic for flow p.

v_l^{stat} Total amount of traffic on link l whose routes cannot be changed.

Objective

Maximize amount of traffic for flow p.

$$\text{maximize } v_p$$

Constraints

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

$$\forall l \in L\colon\ x_l^{\text{max\_all}} \leq \sum_{p \in P} f_{p,l} v_p + v_l^{\text{stat}} \leq x_l^{\text{max\_all}}$$

Appendix C. Linear Programming to Calculate Upper and Lower Bounds for Traffic Included in Aggregated Traffic Information

Inputs

P Set of flows maintained in current area.

P^{agg} Set of flows whose routes can be changed by routing controller using aggregated traffic information.

L Set of links whose traffic information is maintained in current area.

$f_{p,l}$ Portion of amount of traffic for subflow $p \in P^{\text{current}}$ passing link $l \in L$.

$x_l^{\min}$,$x_l^{\max}$ Lower and upper bounds for amount of traffic passing link l whose routes can be changed by routing controller of current area

Variables

v_p Amount of traffic for flow p.

Objective

We maximized the total amounts of traffic for flows p included in P^{agg} passing link l to obtain the upper bounds for the amount of traffic included in aggregated traffic information. We minimized the total amount of traffic for flows p included in P^{agg} passing link l to obtain the lower bounds for the amount of traffic included in aggregated traffic information.

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

Constraints

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

Acknowledgements

This work was supported in part by the Strategic Information and Communications R&D Promotion Programme of the Ministry of Internal Affairs and Communications, Japan.

References

[1] Y. Ohsita, T. Miyamura, S. Arakawa, K. Shiomoto, M. Murata, Hierarchical dynamic traffic engineering considering the upper bounds of link utilizations, in: Proceedings of IEEE GLOBECOM, 2011, pp. 1–6.

[2] B. Fortz, M. Thorup, Internet traffic engineering by optimizing OSPF weights, in: Proceedings of IEEE INFOCOM, Vol. 2, 2000, pp. 519–528.

[3] B. Fortz, Applications of meta-heuristics to traffic engineering in IP networks, International Transactions in Operational Research 18 (2) (2011) 131–147.

[4] M. Roughan, M. Thorup, Y. Zhang, Traffic engineering with estimated traffic matrices, in: Proceedings of ACM SIGCOMM IMC, 2003, pp. 248–258.

[5] Y. Ohsita, T. Miyamura, S. Arakawa, S. Ata, E. Oki, K. Shiomoto, M. Murata, Gradually reconfiguring virtual network topologies based on estimated traffic matrices, IEEE/ACM Transactions on Networking 18 (1) (2010) 177–189.

[6] Y. Ohsita, T. Miyamura, S. Awakawa, E. Oki, K. Shiomoto, M. Murata, Estimation of current traffic matrices from long-term traffic variations, IEICE Transactions on Communications E92-B (1) (2009) 171–183.

[7] Y. Tarutani, Y. Ohsita, S. Arakawa, M. Murata, Optical-layer traffic engineering with link load estimation for large-scale optical networks, IEEE/OSA Journal of Optical Communications and Networking 4 (1) (2012) 38–52.

[8] I. Juva, Robust load balancing, in: Proceedings of IEEE GLOBECOM, 2007, pp. 2708–2713.

[9] Y. Kitahara, E. Oki, Optimal routing strategy by hose model with link-traffic bounds, in: Proceedings of IEEE GLOBECOM, 2011, pp. 1–5.

[10] H. Wang, H. Xie, L. Qiu, Y. R. Yang, Y. Zhang, A. Greenberg, COPE: Traffic engineering in dynamic networks, in: Proceedings of ACM SIGCOMM, 2006, pp. 99–110.

[11] G. Retvari, G. Nemeth, Demand-oblivious routing: distributed vs. centralized approaches, in: Proceedings of IEEE INFOCOM, 2010, pp. 1217–1225.

[12] P. Belotti, M. C. Pinar, Optimal oblivious routing under linear and ellipsoidal uncertainty, Optimization and Engineering 9 (3) (2008) 257–271.

[13] A. Altin, B. Fortz, H. Umit, Oblivious OSPF routing with weight optimization under polyhedral demand uncertainty, Networks 60 (2) (2012) 132–139.

[14] J. Renegar, A polynomial-time algorithm based on newton 's method for linear programming, Mathematical Programming 40 (1988) 59–93.

[15] Y. Ohsita, T. Miyamura, S. Arakawa, S. Ata, E. Oki, K. Shiomoto, M. Murata, Gradually reconfiguring virtual network topologies based on estimated traffic matrices, in: Proceedings of IEEE INFOCOM, 2007, pp. 2511–2515.

[16] A. Elwalid, C. Jin, S. H. Low, I. Widjaja, MATE: MPLS adaptive traffic engineering, in: Proceedings of IEEE INFOCOM, 2001, pp. 1300–1309.

[17] S. Kandula, D. Katabi, B. Davie, A. Charny, Walking the tightrope: Responsive yet stable traffic engineering, ACM SIGCOMM Computer Communication Review 35 (4) (2005) 253–264.

[18] L. Kleinrock, F. Kamoun, Hierarchical routing for large networks performance evaluation and optimization, Computer Networks 1 (1976) 155–173.

[19] S. Uludag, K.-S. Lui, K. Nahrstedt, G. Brewster, Analysis of topology aggregation techniques for QoS routing, ACM Computing Surveys 39 (3) (2007) 7/1–7/31.

[20] W. S. Lai, Y. Lai, Hierarchical routing by PNNI and OSPF/BGP, in: Proceedings of IEEE PACRIM, 2009, pp. 951–956.

[21] D. Applegate, E. Cohen, Making routing robust to changing traffic demands: algorithms and evaluation, IEEE/ACM Transactions on Networking 14 (2006) 1193–1206.

[22] G. Suna, H. Yua, V. A. andLemin Lia, A cost efficient framework and algorithm for embedding dynamic virtual network requests, Future Generation Computer Systems 29 (5) (2013) 1265–1277.

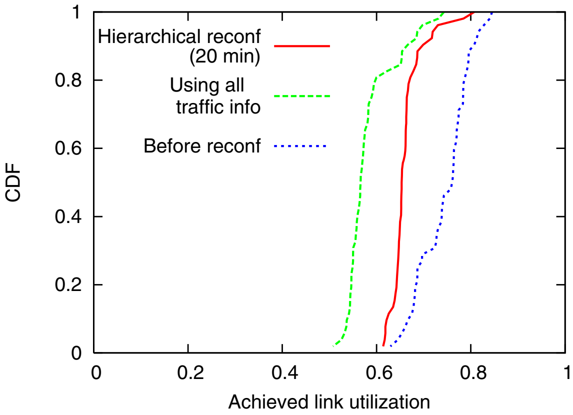
[23] A. Nucci, A. Sridharan, N. Taft, The problem of synthetically generating IP traffic matrices: Initial recommendations, ACM SIGCOMM Computer Communication Review 35 (2) (2005) 19–32.

[24] Y. Zhang, M. Roughan, N. Duffield, A. Greenberg, Fast accurate computation of large-scale IP traffic matrices from link loads, in: Proceedings of ACM SIGMETRICS, 2003, pp. 206–217.

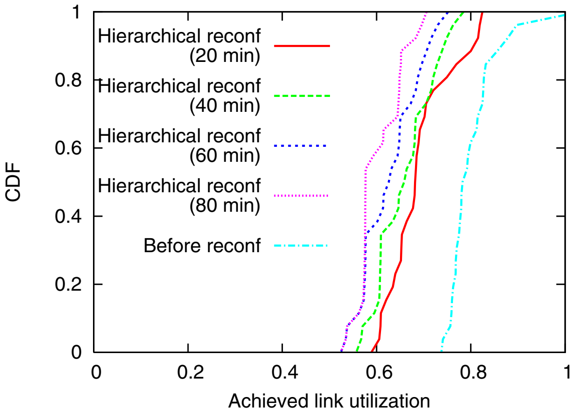
[25] N. Spring, R. Mahajan, D. Wetherall, Measuring ISP topologies with rocketfuel, IEEE/ACM Transactions on Networking 12 (1) (2004) 2–16.

[26] D. Watts, S. Strogatz, Collective dynamics of 'small-world' networks, Nature 393 (1998) 440–442.

[27] T. Hirayama, S. Arakawa, K. Arai, M. Murata, Modularity structure and traffic dynamics of isp router-level topologies, in: Proceedings of NOLTA, 2012.

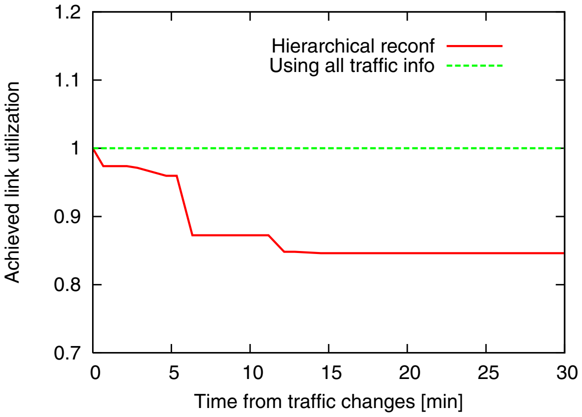


(a) Ebone

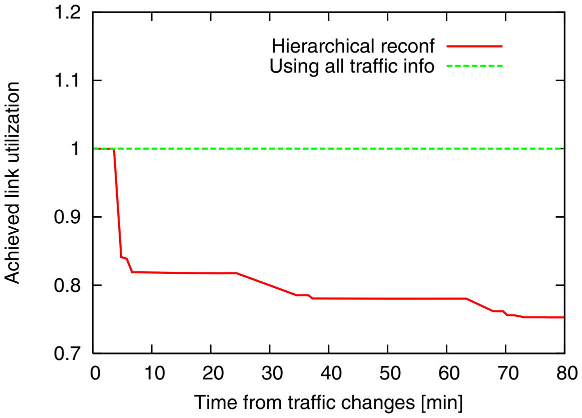


(b) Tinet

Figure 11: Achieved link utilizations



(a) Ebone



(b) Tinet

Figure 12: Average for maximum link utilization achieved by certain time (normalized by initial utilization)

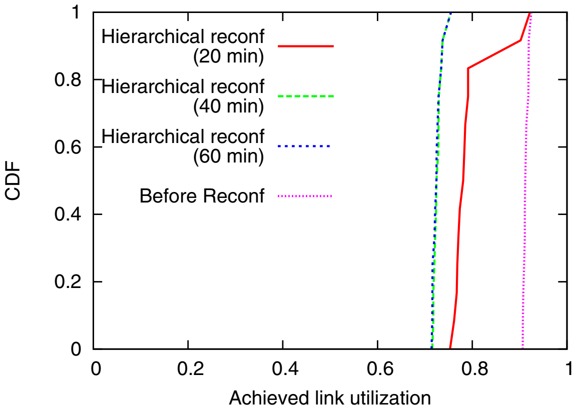


Figure 13: Achieved link utilization (network with 500 nodes)

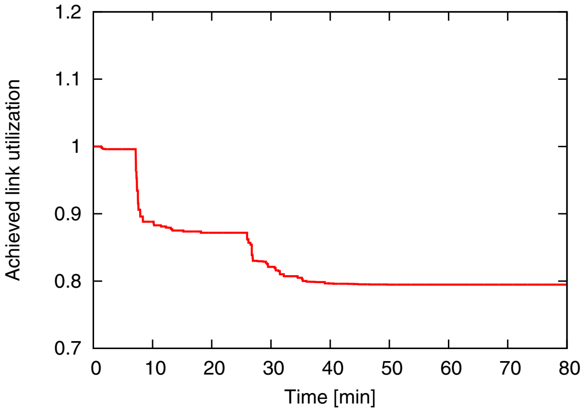


Figure 14: Average for maximum link utilization achieved by certain time (normalized by initial utilization, network with 500 nodes)