*Partial Optimization Method of Topology Aggregation for Hierarchical QoS Routing

Wooguil Pak
Samsung Secui.com Corporation
Soonhwa-dong, Jung-gu
Seoul, Korea

Saewoong Bahk
School of Electrical Engineering
Seoul National University
Seoul, Korea

***Abstract*-To support various services in the current Internet, QoS routing was proposed. QoS routing finds a path to meet the requested QoS specification for a user, reserves network resource, thereby guaranteeing the QoS for the user. Many works have been done for QoS routing in an autonomous system (AS) to make it feasible in large networks. For a large network, hierarchical QoS routing is a most promising candidate because it is scalable. It divides the network into several levels and routing is performed at each level. The most important factor in hierarchical QoS routing is topology aggregation, which make lower level nodes send simplified and aggregated network topology information to upper level nodes. Therefore the topology aggregation enables QoS routing to be run in large networks while it causes some error during the aggregation process. In this paper, we introduce a way to optimize the topology aggregation to improve the performance of QoS routing in terms of exactness. Our scheme uses partial optimization technique instead of whole topology optimization, which is general enough to be used with other existing schemes.**

I. INTRODUCTION

Internet traffic is doubled every three months. As network traffic increases, demand for various services with different service requirements also increases. However the current Internet supports best-effort service only. Many researchers tried to support various service types to meet this demand, and consequently differentiated service and integrated service were proposed. However, integrated service has a problem in scalability and differentiated service has difficulty in guaranteeing the QoS of traffic. Researchers have come to the conclusion that they need to find fundamental solutions. One of the strong proposals is QoS routing. QoS routing finds a best path that satisfies the requested QoS specification of a user with high probability, reserves network resource for the user. As a result, it guarantees the QoS for the user [7]. In this way, the Internet can support various service types with different requirements that are usually represented by bandwidth, end-to-end delay, jitter and so on.

Until now many researchers have tried to make efficient algorithms to find the appropriate path that satisfies the given QoS specifications in a small network such as an AS. Recently, they begin to try to use QoS routing in large networks [6]. Existing works for QoS routing in small networks have some problem in being applied for large networks because of their high computational cost and message complexity. To overcome this scalability problem hierarchical QoS routing has been proposed. In hierarchical QoS routing, nodes are clustered into level groups. At least a link connects two groups and the two connecting nodes are called the border nodes. Each border node has a mapping table for the set of nodes within the same group. The set of nodes is called logical nodes and stores the higher level information. A link connecting two logical nodes is called a logical link. The set of logical nodes is further clustered to form higher level groups. This briefly explains the topology aggregation process to achieve scalability by reducing the size of the global states. In this way the hierarchical QoS routing can reduce computational cost and message complexity for large networks. So the topology aggregation is the most important component in hierarchical QoS routing.

In this paper, we introduce a partial optimization method of topology aggregation. This improves the overall exactness of topology aggregation and minimizes the maximum error. This can be used with other topology aggregation schemes and can improve exactness of aggregation. This paper is organized as follows. Section II reviews the previous work and Section III proposes a partial optimization method. Section IV considers how to solve partial optimization problem. In Section V we analyze our method through simple analysis and simulations, followed by conclusion in Section VI.

II. EXISTING ALGORITHM

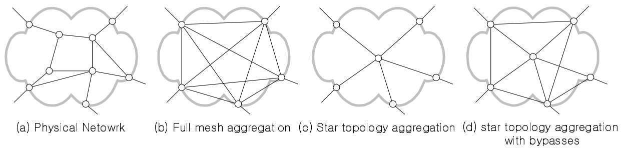
Topology aggregation is composed of link aggregation and node aggregation. Good aggregation means it needs low computational power and the information complexity is small and the error of the logical parameters is small in comparison with parameters of physical links.

A. Link aggregation

Link aggregation is to represent links connected between two nodes with logical parameters[4]. Some schemes simplify parameters of all physical links as only one virtual link parameter for each QoS specification. They can be classified with single-path-parameter approach, multiple-path-parameters-best-case approach and multiple-path-parameters-worst-case approach[2]. Others represent them with line or curve with virtual parameters. These can be classified with approximated line approach and approximated curve approach[1][2]. They have larger information data than the former but they have less error.

* This work was supported by the KOSEF under grant 1999-1-30200-005-3.

0-7803-7400-2/02/$17.00 © 2002 IEEE


(a) Physical Netowrk

(b) Full mesh aggregation

(c) Star topology aggregation

(d) star topology aggregation with bypasses

Fig. 1. Topology aggregation

B. *Node aggregation*

Node aggregation is used to represent the lower level network as a single logical node in the upper level network. This logical node has one parameter called diameter for all links and nodes in the aggregated network for each QoS specification. Node aggregation is often called *simple node aggregation*

C. *Topology aggregation*

Topology aggregation represents the lower level network as a simplified network in the upper level network [1-4]. It uses link aggregation and node aggregation to aggregate sub networks. Most exact aggregation is full mesh aggregation in Fig. 1 (a). As all node pairs have the aggregated links in full mesh aggregation, the complexity of information is $O(b^2)$, where b is the number of border node. This value is too high to achieve good scalability. Star topology aggregation uses border nodes, virtual nodes called nucleus nodes, and links connecting border nodes and nucleus nodes. It has the information complexity of $O(b)$ which is very low in comparison with full mesh. However the difference between the aggregated link and the real physical link is high. So to achieve good scalability and exactness, star topology aggregation with bypasses has been proposed. Instead of using an aggregated link, a bypass link is used when the difference is so high. The bypass has the same parameter values as those in full mesh links. For scalability, the maximum number of bypasses is restricted to b.

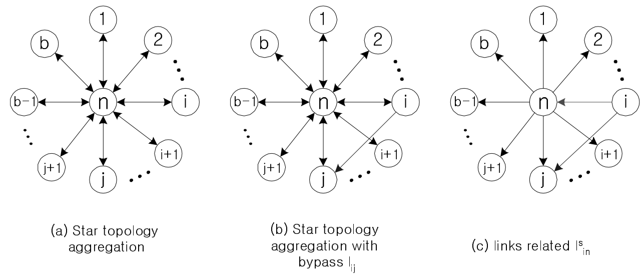
D. *Problem of star topology aggregation with bypasses*

There are two ways to get star topology aggregation with bypasses. One is to make star topology aggregation directly with b bypasses without the intention of forming star topology aggregation. The other is to create optimized star topology first, and compare a link from optimized star topology with a link from full mesh topology, pick b links in order of big differences, and replace them with bypasses [1], [2]. The former can have lower error than the latter, but it is not a feasible solution because its computational complexity is too high to be used in real network. The latter is very simple and has low computational complexity. However the error can be so high, resulting in waste of network resources. In most works, the latter is preferred to the former because it can be implemented in real networks.

III. Proposed algorithm

A. *System model and notation*

Autonomous system is modeled as ***G(V, E)***, where ***V*** is the set of nodes and ***E*** is the set of bi-directional links. ***B*** is the set of border nodes. l_{ij} represents the aggregated link that connects start node i and end node j . Superscript f is used to indicate a link in full mesh and s to indicate a link in star topology.



(a) Star topology aggregation

(b) Star topology aggregation with bypass l_{ij}

(c) links related l^s_{in}

Fig. 2. Star topology aggregation with one bypass

To extend this definition in star topology aggregation with bypasses, we redefine $l^s{}_{ij}$ as $l^s{}_{in} + l^s{}_{nj}$ where node i and node j are not connected directly or $l^f{}_{ij}$ where node i and node j are connected with a bypass. In this paper, we consider only bandwidth and delay as the QoS parameters for simplicity and they can be extended to n QoS parameters easily

For convenience, we use the approximated line aggregation scheme for link aggregation and write $l_{ij} = [(D_1,W_1), (D_2,W_2)]$. We use this notation because it can be used for other schemes such as the single-path-parameter approach, the multiple-path-parameters-best-case approach, and the multiple-path-parameters-worst-case approach. D represents a function to quantify how much different a link is from the other link[1]. For example, $D(l^s{}_{ij},\ l^f{}_{ij})$ indicates the error between star topology link $l^s{}_{ij}$ and full mesh link $l^f{}_{ij}$

B. *Definition of optimized topology aggregation*

We suppose that star topology aggregation with or without bypasses is optimized when the topology is formed with links in the set ***S***.

S is defined as follows.

$$S = \{ l^s_{ij} \mid \sum_{p \neq q,\, p \in B, q \in B} D(l^f_{pq}, l^s_{pq}) \leq \sum_{p \neq q,\, p \in B, q \in B} D(l^f_{pq}, l_{pq}), \forall l_{pq} \} \quad (1)$$

From (1), we can derive the following equations that are called *partial optimization conditions*.

$$\sum_{k \neq i, k \in B} D(l^f_{ik}, l^s_{ik}) = \sum_{k \neq i, k \in B} D(l^f_{ik}, l^s_{in} + l^s_{nk}) \leq \sum_{k \neq i, k \in B} D(l^f_{ik}, l + l^s_{nk}), \forall l \quad (2)$$

$$\sum_{h \neq j, h \in B} D(l^f_{hj,} l^s_{hj}) = \sum_{h \neq j, h \in B} D(l^f_{hj}, l^s_{hn} + l^s_{nj}) \leq \sum_{h \neq j, h \in B} D(l^f_{hj}, l^s_{hn} + l'), \forall l' \quad (3)$$

IV. Partial optimization to minimize the error

It is a difficult problem to obtain the optimized topology with b bypasses. In general it has very high computational complexity. To solve this problem, we propose a partial optimization method for topology aggregation. Partial optimization adds a bypass link to optimized star topology, selects two links that can affect the exactness of aggregated topology very much but are independent of each other, and change the values of links appropriately to achieve better exactness. We explain this for a single bypass addition case and extend it to a general case later.

A. *Single bypass case*

If a bypass is used for star topology aggregation like in Fig. 2(b), we can write eq. (2) as follows.

$$\sum_{k\neq i,k\neq j,k\in B} D(l_{ik}^{f},l_{ik}^{s}) \le \sum_{k\neq i,k\neq j,k\in B} D(l_{ik}^{f},l+l_{nk}^{s})$$

$$(\because \sum_{k\neq i,k\in B} D(l_{ik}^{f},l_{ik}^{s}) = \sum_{k\neq i,k\neq j,k\in B} D(l_{ik}^{f},l_{ik}^{s}) + D(l_{ij}^{f},l_{ij}^{s}) \qquad (2)'$$

$$= \sum_{k\neq i,k\neq j,k\in B} D(l_{ik}^{f},l_{ik}^{s}) + D(l_{ij}^{f},l_{ij}^{f})$$

$$= \sum_{k\neq i,k\neq j,k\in B} D(l_{ik}^{f},l_{ik}^{s}))$$

(2)’ does not hold because a+b≤c+d →a≤c (where a, b, c, d >0) is not true.

To satisfy (2), we need to recalculate all l^{s}_{ij}’s but it takes long time because of its computational complexity. To avoid this problem and reduce the error, we assume that (2)’ can be satisfied by recalculating only a few links that need to satisfy some conditions. First, the number of links to be changed is very small. Second, it needs very low computational power to recalculate. Last, the improvement is very high with the change of a few links’ values. These conditions are necessary for the algorithm to be adopted in real networks.

Let’s choose $l^{s}_{in}{}'$ to satisfy (2)’ in Fig. 2(c),

$$\sum_{k\neq i,k\neq j,k\in B} D(l_{ik}^{f},l_{in}^{s}{}'+l_{nk}^{s}) \le \sum_{k\neq i,k\neq j,k\in B} D(l_{ik}^{f},l+l_{nk}^{s}),\forall l \qquad (4)$$

In the same way we can get the following.

$$\sum_{h\neq i,h\neq j,h\in B} D(l_{hj}^{f},l_{hn}^{s}+l_{nj}^{s}{}') \le \sum_{h\neq i,h\neq j,h\in B} D(l_{hj}^{f},l_{hn}^{s}+l),\forall l \qquad (5)$$

Because most links are dependent of each other, so we can’t guarantee the increase of overall exactness. Now we verify that the topology aggregation improves exactness by selecting $l^{s}_{in}{}'$ and $l^{s}_{nj}{}'$ to satisfy (4) and (5).

$$\sum_{h\in B,k\in B,h\neq k} D(l_{hk}^{f},l_{hk}^{s}{}^{*})$$

$$= \sum_{\substack{h\in B,k\in B,h\neq k\\ and\ except(h=i,k=j)}} D(l_{hk}^{f},l_{hk}^{s}{}^{*}) + \sum_{h=i,k=j} D(l_{hk}^{f},l_{hk}^{s}{}^{*})$$

$$= \sum_{\substack{h\in B,k\in B,h\neq k\\ and\ except(h=i,k=j)}} D(l_{hk}^{f},l_{hk}^{s}{}^{*}) + D(l_{ij}^{f},l_{ij}^{s*})$$

$$= \sum_{\substack{h\in B,k\in B,h\neq k\\ and\ except(h=i,k=j)}} D(l_{hk}^{f},l_{hn}^{s}{}^{*}+l_{nk}^{s}{}^{*})$$

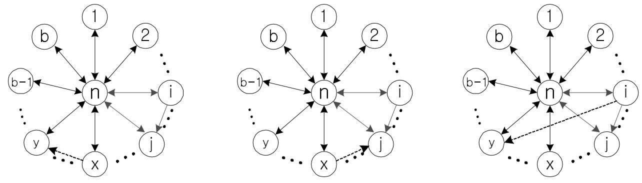
$$= \sum_{h\in B,k\in B,h\neq k,h\neq i,k\neq j} D(l_{hk}^{f},l_{hn}^{s}{}^{*}+l_{nk}^{s}{}^{*}) + \sum_{h=i,k\in B,k\neq j,h\neq k} D(l_{hk}^{f},l_{hn}^{s}{}^{*}+l_{nk}^{s}{}^{*}) + \sum_{k=j,h\in B,h\neq k,h\neq i} D(l_{hk}^{f},l_{hn}^{s}{}^{*}+l_{nk}^{s}{}^{*})$$

$$= \sum_{h\in B,k\in B,h\neq k,h\neq i,k\neq j} D(l_{hk}^{f},l_{hn}^{s}{}^{*}+l_{nk}^{s}{}^{*}) + \sum_{k\in B,k\neq i,k\neq j} D(l_{ik}^{f},l_{in}^{s}{}^{*}+l_{nk}^{s}{}^{*}) + \sum_{h\in B,h\neq k,h\neq i} D(l_{hj}^{f},l_{hn}^{s}{}^{*}+l_{nj}^{s}{}^{*})$$

$$\ge \sum_{h\in B,k\in B,h\neq k,h\neq i,k\neq j} D(l_{hk}^{f},l_{hn}^{s}{}^{*}+l_{nk}^{s}{}^{*}) + \sum_{k\in B,k\neq i,k\neq j} D(l_{ik}^{f},l_{in}^{s}{}'+l_{nk}^{s}{}^{*}) + \sum_{h\in B,h\neq k,h\neq i} D(l_{hj}^{f},l_{hn}^{s}{}^{*}+l_{nj}^{s}{}') \qquad (6)$$

In (6), we guarantee that we make the error of topology aggregation small by recalculating $l^{s}_{in}{}'$ and $l^{s}_{nj}{}'$ to satisfy (4) and (5). Moreover, the calculation complexity is very low. The second and third parts of (6) contain $l^{s}_{in}{}'$ and $l^{s}_{nj}{}'$ and they are independent of each other. So it is possible to calculate $l^{s}_{in}{}'$ and $l^{s}_{nj}{}'$ independently. And $l^{s}_{in}{}'$ and $l^{s}_{nj}{}'$ can improve the exactness very much because they are heavily affected by bypass.

B. Multi bypasses Case



(a) Case Ⅰ (b) Case Ⅱ (c) Case Ⅲ

Fig. 3. Multi bypasses

We extend the case by adding multi bypasses to the star topology aggregation. Bypass link l_{ij} is added first and additional bypasses will be added. According to the formation style, there exist three cases as shown in Fig. 3 where the dotted lines indicate additional bypasses. We will show only the results without showing detailed calculation procedures. To simplify the expression, we define ***E***(x) as the set of end nodes of bypasses that start from node x, and define ***S***(y) as the set of start nodes of bypasses that end at node y.

$$\sum_{h\in B,h\neq x} D(l_{xh}^{f},l_{xh}^{s}{}^{*})$$

$$= \sum_{h\in E(x)} D(l_{xh}^{f},l_{xh}^{s}{}^{*}) + \sum_{h\in B-E(x)} D(l_{xh}^{f},l_{xh}^{s}{}^{*}) \qquad (7)$$

$$= \sum_{h\in E(x)} D(l_{xh}^{f},l_{xh}^{f}) + \sum_{h\in B-E(x)} D(l_{xh}^{f},l_{xh}^{s}{}^{*})$$

$$= \sum_{h\in B-E(x)} D(l_{xh}^{f},l_{xh}^{s}{}^{*})$$

$$\sum_{h\in B-E(x)} D(l_{xh}^{f},l_{xh}^{s}{}') = \sum_{h\in B-E(x)} D(l_{xh}^{f},l_{xn}^{s}{}'+l_{nh}^{s}{}^{*}) \le \sum_{h\in B-E(x)} D(l_{xh}^{f},l+l_{nh}^{s}{}^{*}),\forall l \qquad (8)$$

In the same way,

$$\sum_{k\in B-S(y)} D(l_{ky}^{f},l_{ky}^{s}{}') = \sum_{k\in B-S(y)} D(l_{ky}^{f},l_{kn}^{s}{}^{*}+l_{ny}^{s}{}') \le \sum_{k\in B-S(y)} D(l_{ky}^{f},l_{kn}^{s}{}^{*}+l'),\forall l \qquad (9)$$

we can reduce the gap in inequality by choosing $l^{s}_{xn}{}'$ and $l^{s}_{ny}{}'$ to satisfy (8) and (9).

V. SIMULATION AND ANALYSIS

A. Simulation results

To measure the exactness enhancement in simulation, we should know how we make optimized star topology and how we measure the error of star topology link from full mesh link. Generally, because these need very high computational costs, people use heuristic method to get them.

We choose the way described in [1] to make star topology and to define function *D*. But *D* in [1] is irrational equation and it needs high computational cost. To make computational cost low and the equation easy to solve, we redefine and use d that is approximated and rational *D*.

$d(l^{f}_{ij}, l^{s}_{ij})$

$$= (\sqrt{(a_1-a_2)^2+(b_1-b_2)^2})^2 + (\sqrt{(c_1-c_2)^2+(d_1-d_2)^2})^2 \qquad (10)$$
$$= (a_1-a_2)^2+(b_1-b_2)^2+(c_1-c_2)^2+(d_1-d_2)^2$$

where $l^{f}_{ij} = [(a_1,b_1), (c_1,d_1)]$, $l^{s}_{ij} = [(a_2,b_2), (c_2,d_2)]$

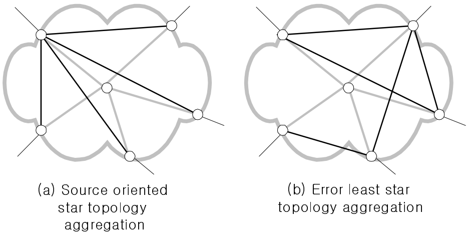
While we use *d* in computation, we use *D* to measure the degree of exactness of our scheme when compared with the old scheme. We can solve (8) and (9) by using (10), and write the results as follows.

$$l_{xn}^{s\,'} = \frac{1}{N(B-E(x))}\sum_{h\in B-E(x)}(l_{xh}^{f} - l_{nh}^{s\,*}) \tag{11}$$

$$l_{ny}^{s\,'} = \frac{1}{N(B-S(y))}\sum_{h\in B-S(y)}(l_{ky}^{f} - l_{kn}^{s\,*}) \tag{12}$$

To simulate various schemes, we select two star topology aggregation schemes as shown in Fig. 4: *source oriented star topology aggregation* and *error least star topology aggregation*. We call the general star topology aggregation scheme that has virtual node named nucleus as the least error star topology aggregation scheme in [1] to distinguish from the source oriented star topology aggregation scheme in [2]. The least error star aggregation scheme fits well into source routing schemes, and the source oriented star aggregation scheme does for distributed routing schemes though it also can be used for source routing schemes[5]. We test various link aggregation schemes such as single-path-parameter approach, multiple-path-parameters-best-case approach multiple-path-parameters-worst-case approach, and approximated line approach[2].

We consider 10 network topologies that have different sizes of border nodes. Each link has random parameter values for star topology aggregation with bypasses. We measure the total sum of *D* of all pairs of nodes and the maximum value of *D*. The total sum of *D* explains the overall exactness, and the maximum *D* indicates large error occurred at a particular path. We represent them as the ratio of the value after adding bypasses to the one before adding bypasses. Our scheme is compared with the widely adopted schemes that were previously explained.


(a) Source oriented star topology aggregation

(b) Error least star topology aggregation

Fig. 4. Source oriented star topology aggregation and error least star topology aggregation

TABLE I
RESULTS IN ERROR LEAST STAR TOPOLOGY AGGREGATION

Border node size	Metric	Scheme	Line	One	Best	Worst
5	Decrement ratio of Total D	Old	0.49	0.49	0.47	0.5
		New	0.55	0.71	0.57	0.83
	Decrement ratio of Max D	Old	0.4	0.4	0.24	0.47
		New	0.53	0.6	0.43	0.6
10	Decrement ratio of Total D	Old	0.28	0.25	0.27	0.34
		New	0.36	0.32	0.34	0.46
	Decrement ratio of Max D	Old	0.34	0.37	0.36	0.3
		New	0.42	0.37	0.43	0.39

TABLE II
RESULTS IN SOURCE ORIENTED STAR TOPOLOGY AGGREGATION

Border node size	Metric	Scheme	Line	One	Best	Worst
5	Decrement ratio of Total D	Old	0.19	0.17	0.2	0.16
		New	0.42	0.40	0.38	0.41
	Decrement ratio of Max D	Old	0.03	0.01	0.06	0.02
		New	0.29	0.23	0.27	0.27
10	Decrement ratio of Total D	Old	0.1	0.29	0.21	0.2
		New	0.15	0.43	0.43	0.47
	Decrement ratio of Max D	Old	0.01	0.08	0.02	0.04
		New	0.21	0.1	0.12	0.19

B. Analysis

In Tables I and II, the total sum of *D* and the max *D* of our schemes are reduced in comparison with the old scheme for all link aggregation methods. Figures 5 and 6 show how our partial optimization scheme influences the error reduction. Existing star topology aggregation algorithms use the follow steps:

(a) Make full mesh topology aggregation

(b) Create a star topology network by using the result from (a)

(c) Select *b* node pairs and replaces corresponding links with bypasses.

In least error star topology aggregation, the computational complexity of the step (b) is $O(b^2log(b))$ while it is $O(b)$ in source oriented star topology aggregation. The computational complexity of our algorithm is $O(b^2log(b)+ b^2)=O(b^2log(b))$ because those of (11) and (12) are $O(b)$. The increasing computational cost is very low in least error star aggregation but it is very high in source oriented star aggregation. The enhancements in the total *D* and the Max *D* are greater than the existing schemes. If source oriented star aggregation is used with a distributed routing algorithm, it takes long time to get sufficient information to reconstruct the entire network topology and it causes to use of network resources very inefficiently. Our scheme makes almost exact aggregation topology and allows network resource to be used efficiently.

TABLE III
EXACTNESS COMPARISON

	No bypass	Old scheme	Optimum	Approximated D (d)
Average D	222	197	153	170

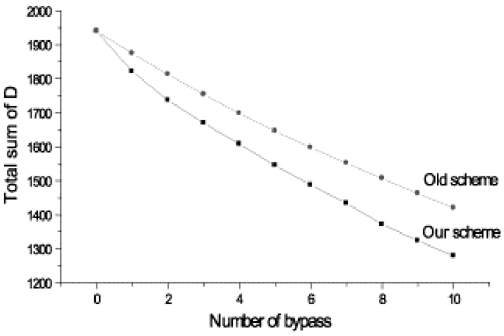


Fig. 5. Total sum of *D* for approximation by line link aggregation (border node size = 10).

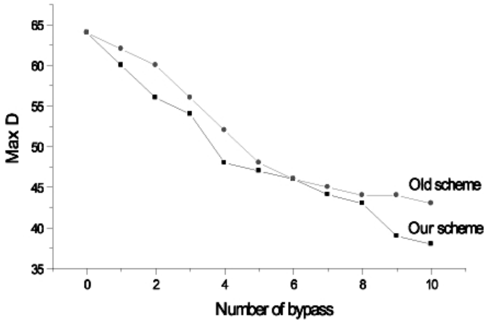


Fig. 6. Max *D* for approximation by line link aggregation (border node size = 10).

Previously we used *d* instead of *D* because it requires a lot of computational power to obtain *D*. Calculation error occurs due to the use of *d* but it will be compensated by the improvement. By using *D*, we can get more improved topology aggregation. It is a trade-off between complexity and exactness improvement. Table III shows the error that incurs by using *d*. To measure the exactness of *d* and *D*, we simulated star topology aggregation in a large network with 10 border nodes. In this simulation we tried 50 networks by using 5 types of networks and assigned bandwidth and delay randomly to each link.

VI. CONCLUSION

Topology aggregation is a major subject in QoS routing and has been researched actively. In this paper, we suggested the partial optimization scheme to improve the exactness of star topology aggregation with bypasses. The existing algorithms simply added bypasses to optimized star topology and they could not get topology aggregation with low error. To obtain more exact topology, we suggested a new scheme called *partial optimization* and verified that it increased the exactness level of aggregation. The simulation results were shown for various network topologies. Our scheme is general enough to be used for various existing aggregation schemes including link aggregation and topology aggregation schemes to make topology aggregation more exact.

REFERENCES

[1] K. -S. Lui and K. Nahrstedt, "Topology Aggregation of Bandwidth-Delay Sensitive Networks," *IEEE Globecom 2000*, November 27 , December 1, 2000.

[2] T. Kormaz and M. Krunz, “Source-Oriented Topology Aggregation with Multiple QoS Parameters in Hierarchical ATM Networks, *IWQoS ’99*, pp. 137-146, 1999.

[3] W. C. Lee, “Spanning Tree Method For Link State Aggregation in Large Communication Networks,” *IEEE INFOCOM*, vol. 1, pp. 297-302, 1995.

[4] B. Awerbuch and Y. Shavitt, “Topology Aggregation for Directed Graph,” *in IEEE Proc. ISCC' 98*, pp. 47-52, 1998.

[5] J. Behrens and J.J. Garcia-Luna-Aceves, “Hierarchical Routing Using Link Vectors,” *IEEE INFOCOM ‘98*, April 1998.

[6] F. Hao and E. W. Zegura, “On Scalable QoS Routing: Performance Evaluation of Topology Aggregation,” *INFOCOM '00*, March 2000.

[7] R. A. Guerin and A. Orda, “ QoS Routing in Networks with Inaccurate Information: Theory and Algorithms,” *IEEE/ACM Trans. Net.* June *'99*