Pre-Computation Based Selective Probing (PCSP) Scheme for Distributed Quality of Service (QoS) Routing with Imprecise State Information

Won-Ick Lee and Byeong Gi Lee

Abstract: **We propose a new distributed QoS routing scheme called pre-computation based selective probing (PCSP). The PCSP scheme is designed to provide an exact solution to the constrained optimization problem with moderate overhead, considering the practical environment where the state information available for the routing decision is not exact. It does not limit the number of probe messages, instead, employs a qualitative (or conditional) selective probing approach. It considers both the cost and QoS metrics of the least-cost and the best-QoS paths to calculate the end-to-end cost of the found feasible paths and find QoS-satisfying least-cost paths. It defines strict probing condition that excludes not only the non-feasible paths but also the non-optimal paths. It additionally pre-computes the QoS variation taking into account the impreciseness of the state information and applies two modified QoS-satisfying conditions to the selection rules. This strict probing condition and carefully designed probing approaches enable to strictly limit the set of neighbor nodes involved in the probing process, thereby reducing the message overhead without sacrificing the optimal properties. However, the PCSP scheme may suffer from high message overhead due to its conservative search process in the worst case. In order to bound such message overhead, we extend the PCSP algorithm by applying additional quantitative heuristics. Computer simulations reveal that the PCSP scheme reduces message overhead and possesses ideal success ratio with guaranteed optimal search. In addition, the quantitative extensions of the PCSP scheme turn out to bound the worst-case message overhead with slight performance degradation.**

Index Terms: **Distributed routing, quality of service (QoS), routing.**

I. INTRODUCTION

Broadband integrated networks are evolving to support various applications with complicate and diverse quality of service (QoS) requirements. There are two essential elements: One is QoS routing that determines how to find the available resources and how to use them most effectively and the other is QoS-guaranteing mechanism that supports how to reserve and allocate resources or guarantee requirements.

The goal of QoS routing is not only in finding the QoS-satisfying paths but also in enhancing network efficiency by determining the least-cost path that has sufficient resources to satisfy the given QoS requirements. This procedure is a heavily complex task because such constrained optimization problem is NP-complete and it should be executed for every request [1]–[3]. Furthermore, the state information is inherently imprecise in the dynamic networks [4], [5]. This impreciseness seriously affects the performance of the applied QoS routing scheme: The selected path may fail to guarantee the requirements, or the routing scheme may fail to find any feasible path even if such one exists.

There has been proposed a large number of routing algorithms and related models to support QoS. Source routing schemes [6]–[16] and distributed routing schemes [17]–[20] were proposed as the basic QoS routing schemes; and imprecise state information models [4], [5] and QoS partitioning methods [22]–[24] were introduced to additionally handle the imprecise state information and resource allocation issues.

Source routing schemes intend to determine the optimal path at the source node [1]–[3], so the computational complexity at the source node is very high and consequently, developing effective heuristics that reduce the computational complexity has been the main interest of research. Though many outstanding proposals such as ϵ-approximation method [12] and Lagrange relaxation method [13] were made, there still remain several important issues: The proposed algorithms still have heavy-polynomial computational complexity, are hard to guarantee an exact solution except for the A* PRUNE [15] algorithm,[1] and have no effective handling method for the practical environments with imprecise state information.

Distributed schemes eliminate the computational complexity problem employing the limited flooding to the whole network, namely, *selective probing*. It searches *feasible paths*[2] by forwarding *probe messages* in a designed parallel and hop-by-hop manner and then selects the best path at the destination node, so the *message overhead*[3] becomes the major overhead to reduce [1]–[3]. The recently proposed schemes [17]–[20] reduce the message overhead by employing quantitative methods that implicitly or explicitly control the number of probe messages. However, the quantitative method inherently degrades the optimal properties and requires to optimize the control parameters. It basically finds two different types of "good" paths—small-cost paths and good-QoS paths—but they are selected independently of each other. Therefore, it will fail to find the optimal path if the optimal path is not selected either as a small-cost path or as a good-QoS path due to having moderate values of cost and QoS metrics.

In this paper, we are going to present a new distributed QoS

Manuscript received September 11, 2002; approved for publication by Danny Raz, Division III Editor, July 15, 2005.

The authors are with the School of Electrical Engineering, Seoul National University, Seoul, 151-742, Korea, email: lwi@tsp.snu.ac.kr, blee@snu.ac.kr.

---

[1] Though the ϵ-approximation methods can guarantee a solution ϵ away from optimal, the computational complexity increases when small ϵ is required.

[2] A feasible path refers to the path that meets the QoS requirements.

[3] Message overhead refers to the number of messages generated during the probing process.

1229-2370/06/$10.00 © 2006 KICS

routing scheme named pre-computation based selective probing (PCSP). The PCSP scheme is designed to reduce message overhead while maintaining optimal properties—ideal success ratio and cost optimality. In contrast to the other selective probing approaches, the PCSP scheme does not limit the number of probe messages but, instead, employs a qualitative (or conditional) selective probing approach that excludes "bad" paths—that is, non-feasible paths and non-optimal paths—to maintain ideal properties. Especially, it excludes non-optimal paths by using both the end-to-end cost and the end-to-end QoS metrics in conjunction, thereby reducing the message overhead to be comparable with or less than the existing distributed schemes without using any auxiliary parameters. It selects the least-cost path (or LC path) and the best-QoS path (or BQ path) as the candidate paths and computes both the cost and QoS metrics for them as pre-computed information. The PCSP scheme is also designed to operate in the practical environments with imprecise state information. It computes the maximum possible variation between the real and the advertised QoS metrics as additional pre-computed information and defines two QoS-satisfying conditions to handle the imprecise environment.

However, due to its conservative selective probing approach, the PCSP scheme may suffer from high message overhead when the state information is extremely imprecise and the network size is large. Therefore, we extend the PCSP scheme by adopting additional quantitative heuristics in order to bound the worst-case message overhead. The qualitative nature of the PCSP scheme helps to employ any additional heuristics with minimal overhead. Furthermore, the performance degradation in optimal properties induced by such extensions is small as the strict probing condition already reduces the set of candidate paths.

The paper is organized as follows: First, we introduce the system model in Section II, focusing on the imprecise state information model. Then, we describe the framework of the PCSP scheme in Section III, with the detailed algorithm presented in Section IV and the properties discussed in Section V. In Section VI, we present quantitative extensions of PCSP scheme. Finally, in Section VI, we discuss simulation results on the PCSP scheme and its extended versions in terms of effectiveness and search optimality.

II. SYSTEM MODEL

We consider a network composed of a set of nodes V and a set of full-duplex, directed communication links E that interconnect the nodes. The state information of a network consists of some QoS metrics and a cost metric. The QoS metrics can be grouped into two categories, additive metric such as delay and non-additive metric such as residual bandwidth [1]. A state information is composed of the information about the packet processing at the node and the information about the packet transmission at the link. Usually, the former is the dominant part and is regarded as the allocatable resource that varies continuously with the change of traffic and call admission condition. On the other hand, the latter may be treated as the quasi-static and unavoidable cost. In this paper, we assume that the state information of each link includes the information of its parent node. The state information of a link (i,j) is composed of a delay $D(i,j)$, a residual bandwidth $B(i,j)$, and a cost $C(i,j)$. For a given path p, the end-to-end metrics are represented by such state information as follows

$$D(p) = \sum_{(i,j)\in p} D(i,j) \tag{1}$$

$$B(p) = \min_{(i,j)\in p} B(i,j) \tag{2}$$

$$C(p) = \sum_{(i,j)\in p} C(i,j). \tag{3}$$

We denote by $Q_R(s,t)$ the QoS requirement of the source and destination node pair (s,t).

A. Imprecise State Information Model

QoS routing requires the mechanisms that estimate, allocate, and reserve resources and the mechanisms that collect and advertise the state information. While the former can be performed instantaneously using exact values when the change or request occurs, the latter is normally done not immediately but in periodic or threshold-based manner so as to reduce the update overhead. In addition, the transmission and processing delay in the links is not negligible either. As a consequence, the advertised information is inherently imprecise [4], [5] even though the state information estimated in each router is exact.

For handling imprecise state information, several models have been proposed [4], [5], [19]. Among them, we apply the basic concept of Chen and Nahstedt's model [19] in building our model. We assume that each router is able to measure the exact state information of all its outgoing links, but knows only the imprecise state information about all other links. We ignore the impreciseness caused by cost metric and topology change, as their impact is not significant when compared with that caused by QoS metric change [19].

We define by the QoS variation $\Delta Q(i,j)$ of link (i,j), the maximum difference between the advertised value $Q_{ad}(i,j)$, and the real value $Q_{real}(i,j)$, i.e.,

$$\Delta Q(i,j) = \sup\{|Q_{ad}(i,j) - Q_{real}(i,j)|\}. \tag{4}$$

It is important to estimate the QoS variation considering the way how the state information is advertised. If we adopt the periodic update, then the well-known weighted sum methods used for the RTT calculation of TCP will be one of the most acceptable candidates. In this case, the QoS variation can be calculated by [19]

$$\Delta Q^{new} = \alpha \cdot \Delta Q^{old} + (1-\alpha)\cdot\beta\cdot|Q_{ad}^{new} - Q_{ad}^{old}| \tag{5}$$

where $\alpha < 1$ and $\beta > 1$. On the other hand, if we adopt the threshold-based update, in which the new state information is advertised when the QoS metric goes out of the range $[\delta_l \cdot Q_{ad}, \delta_u \cdot Q_{ad}]$ for the lower and upper update threshold factors $\delta_l < 1$ and $\delta_u > 1$, then the QoS variation can be calculated by

$$\Delta Q = \max\{(\delta_u - 1)\cdot Q_{ad},\ (1-\delta_l)\cdot Q_{ad}\}. \tag{6}$$

In the two methods above, there always exists some possibility of mis-estimating the QoS variation. To minimize the effect of such mis-estimation, we introduce a compensation factor γ

and the compensated QoS variation $\Delta^C Q \equiv \gamma \cdot \Delta Q$. We adapt γ in a self-configurable manner as follows: If the newly advertised state information Q_{ad}^{new} exceeds the QoS variation range $[\max(Q_{ad}^{old} - \Delta Q,\ Q_{\min}),\ \min(Q_{ad}^{old} + \Delta Q,\ Q_{\max})]$,[4] then γ is adjusted to $\alpha_\gamma \cdot \gamma$ for a pre-defined increasing factor $\alpha_\gamma > 1$. If Q_{ad}^{new} does not exceed the range by n_γ times, then γ is adjusted to $\beta_\gamma \cdot \gamma$ for a pre-defined decreasing factor $\beta_\gamma < 1$. In the remaining part of this paper, we assume that ΔQ is well compensated and use it as the compensated QoS variation value.

For a path p, we calculate the additive and non-additive end-to-end QoS variations in different ways. As the representative metrics, we define the end-to-end delay variation $\Delta D(p)$ and the end-to-end residual bandwidth variation $\Delta B(p)$ as follows

$$\Delta D(p) = \sum_{(i,j)\in p} \Delta D(i,j) \tag{7}$$

$$\Delta B(p) = \Delta\{\min_{(i,j)\in p} B(i,j)\}. \tag{8}$$

We define two extreme QoS-satisfying conditions by considering the QoS variation—a strict condition and a loose condition. The strict condition is that the given metric always satisfies the requirements even if the QoS variation is considered. In contrast, the loose condition is that the given metric possibly satisfies the requirements if the QoS variation is considered. These two conditions should be adequately determined according to the nature of the QoS metrics. For the delay requirement D_R and bandwidth requirement B_R, the strict condition Q_S and the loose condition Q_L are represented as follows

$$Q_S \ :\ D + \Delta D \le D_R \text{ or } B - \Delta B \ \ge B_R \tag{9}$$

$$Q_L \ :\ D - \Delta D \le D_R \text{ or } B + \Delta B \ \ge B_R. \tag{10}$$

B. Pre-Computed Information

In the PCSP scheme, each node calculates a pre-computed information based on the collected state information. Considering the QoS variation, each node i determines the LC path $LC_i(t)$ that has the least cost and the BQ path $BQ_i(t)$ that guarantees best QoS (such as the least delay path or the maximum residual bandwidth path) for every possible destination node t among the paths connecting itself to the destination node t. For both paths, it pre-computes cost metrics $C(LC_i(t))$, $C(BQ_i(t))$, QoS metrics $Q(LC_i(t))$, $Q(BQ_i(t))$, QoS variations $\Delta Q(LC_i(t))$, $\Delta Q(BQ_i(t))$, and the next hop nodes $N(LC_i(t))$, $N(BQ_i(t))$. Note that this contrasts to the existing distributed schemes that pre-compute only the QoS metric for the BQ path and only the cost metric for the LC path. This extension enables the PCSP scheme to compute the end-to-end cost of the found feasible paths and find the QoS-satisfying least-cost path (QLCP), thereby bringing forth strict selective probing procedures and meritorious properties that were impossible before.

The state information may be collected through existing protocols such as link state protocol and distance vector protocol. Unlike the source routing schemes, the PCSP scheme as well as the other distributed schemes can use the distance vector protocol to achieve better scalability and lower update overhead as it does not require the whole state information. However, due to the slow convergence time and possible instability, the distance vector protocol may degrade the routing performance especially when the state information is highly imprecise or changes fast. In this case, PCSP scheme gains more benefits by using the link state protocol.

C. Messages

The PCSP scheme uses four types of messages: *set_msg*, *find_msg*, *probe_msg*, and *ack_msg*. *set_msg* and *find_msg* are used to notify that a possible optimal path is found. *probe_msg* is used to find feasible paths. *ack_msg* is generated during the acknowledgement phase to announce the routing results.

Each message m contains the basic information such as source node s, destination node t, parent node k, and message type ID. For the QoS routing, it also contains the QoS requirement $Q_R(s,t)$; the QoS and the cost metrics of the path $p(s,i)$ that it has traversed, $Q(p(s,i))$, $C(p(s,i))$; and the least end-to-end (e2e) cost value of the determined feasible paths, $C_{\min}^{e2e}(s,i,t)$. Note that the terms $Q(p(s,i))$ and $C(p(s,i))$ are exact as they are updated using the exact state information at each node, not the advertised information.

III. THE PCSP SCHEME—A FRAMEWORK

The PCSP scheme aims to achieve low message overhead without sacrificing ideal success ratio or cost optimality by employing a qualitative (or conditional) approach. It defines a strict probing condition and forwards the probe messages to a set of neighbor nodes satisfying this condition, namely, *probing set*.

The probing condition is designed to exclude two kinds of nodes—the nodes that do not have any QoS-satisfying paths and the nodes that do not have any less-cost paths than the readily determined feasible paths. For the former, the PCSP scheme checks the end-to-end QoS and the QoS variation of the BQ paths. If the end-to-end QoS of a neighbor node's BQ path does not satisfy the QoS requirement even considering the end-to-end QoS variation of the path, the PCSP scheme judges that the neighbor has no QoS-satisfying path and excludes it from the probing set. For the latter, the PCSP scheme uses the end-to-end cost of LC paths for judgement. The key design issue is how to find the feasible paths having a low cost. Using the end-to-end cost and QoS metric in combination, the PCSP scheme can calculate the end-to-end cost of the found feasible paths and find the QLCPs, thereby excluding the non-optimal paths effectively.

We describe in the next two subsections two selective probing algorithms of the PCSP scheme. They are both formulated using the extended pre-computed information of the current node and its neighbor nodes,[5] the information in the received message, and the calculated end-to-end metrics of the candidate paths. During the probing process, an intermediate node i that receives a *probe_msg* knows the link metrics; the QoS and the cost metrics from the source node s to itself out of the information in the *probe_msg*; the QoS metric, the cost metric, and the QoS variation of the LC and the BQ paths out of the pre-computed

---

[4] $Q_{\min}$ and $Q_{\max}$ mean the minimum and maximum possible Q values.

[5] The information of the neighbor nodes can be acquired during the state information gathering procedure or by requesting to the neighbor nodes.

information. Using these metrics, the node can calculate the end-to-end cost and QoS metrics from the source node s to the destination node t via the LC or BQ path of itself or its neighbor node j as follows.[6]

$$C_{e2e,i}(p_i(t)) = C(p(s,i)) + C(p_i(t)) \tag{11a}$$

$$D_{e2e,i}(p_i(t)) = D(p(s,i)) + D(p_i(t)) \tag{11b}$$

$$B_{e2e,i}(p_i(t)) = \min(B(p(s,i)), B(p_i(t))) \tag{11c}$$

$$C_{e2e,i}(p_j(t)) = C(p(s,i)) + C(i,j) + C(p_j(t)) \tag{12a}$$

$$D_{e2e,i}(p_j(t)) = D(p(s,i)) + D(i,j) + D(p_j(t)) \tag{12b}$$

$$B_{e2e,i}(p_j(t)) = \min(B(p(s,i)), B(i,j), B(p_j(t))) \tag{12c}$$

where $p_k(t) \in \{LC_k(t), BQ_k(t)\},\ k = i, j$. The QoS variation of the pre-computed information represents the end-to-end value because the metric from the source node to the current node and the link metric are both exact information gathered during the probing process (see Section II-A).

A. Selective Probing Using the Least-Cost Value of the Determined Feasible Paths

We develop a new selective probing approach using the least end-to-end cost value of the determined feasible paths, $C^{e2e}_{\min}(s,i,t)$. This value is determined from the feasible paths found by a *probe_msg* while probing up to node i. An intermediate node i receiving a *probe_msg* updates $C^{e2e}_{\min}(s,i,t)$ if any of its LC and BQ paths satisfies the end-to-end QoS requirement and has less end-to-end cost than the received $C^{e2e}_{\min}(s,i,t)$ value.[7] As each node knows the pre-computed information of its neighbor nodes, this process can be extended by considering the LC and BQ paths of its neighbor nodes.

The PCSP scheme uses $C^{e2e}_{\min}(s,i,t)$ value to select the probing set $R_i(t)$ of node i among its neighbor node set N_i. If the end-to-end cost of the LC path of a neighbor node j, $C_{e2e,i}(LC_j(t))$, is greater than $C^{e2e}_{\min}(s,i,t)$, then all the paths passing through node j have a greater end-to-end cost than the found feasible path has, as it is the least-cost value among them. Therefore, the PCSP scheme adds a neighbor node j to the probing set only if $C_{e2e,i}(LC_j(t)) \leq C^{e2e}_{\min}(s,i,t)$ and $Q_{e2e,i}(BQ_j(t))$ satisfies the QoS requirement.[8]

B. Selective Probing Using the QoS-Satisfying LC Paths

At the same time, we can use the LC paths that satisfy the end-to-end QoS requirement (i.e., QLCPs). If the LC path of an intermediate node satisfies the QoS requirement, it is determined to be the optimal path among all the paths passing through this node.[9] Once such an LC path is determined, it is not necessary to continue any additional probing from this node. So, the probing process of this node (not the whole process) is terminated and a *find_msg* is sent to the destination node through this LC path.

In the PCSP scheme, the second procedure is performed at the parent node in conjunction with the first one. An intermediate node i that received a *probe_msg* examines the neighbor node as follows: First, node i updates $C^{e2e}_{\min}(s,i,t)$. It tries to find a neighbor node j having a QLCP from the neighbor node set $N_i(t)$. If the found path $LC_j(t)$ has an end-to-end cost that is less than or equal to $C^{e2e}_{\min}(s,i,t)$, a *find_msg* is forwarded to the neighbor node j and this node is excluded from the probing set $R_i(t)$.

C. Illustration of the Selective Probing Procedure

Table 1 and Fig. 1 illustrate the selective probing procedure at an intermediate node i for the delay-constrained QoS routing case. Fig. 1 shows only a part of the whole network—an intermediate node i, its neighbors j_n, node k, and the destination node t. The two numbers at each link indicate delay and cost metrics, respectively with the delay metric given in ms. The BQ paths are represented in dashed lines, and the LC paths in dotted lines in the figure.

Table 1 lists all the information available at node i. A *probe_msg* arrives at node i with the path information $D(p(s,i)) = 80$, $C(p(s,i)) = 60$, the delay requirement $D_R(s,t) = 170$, and the least cost of the determined feasible paths $C^{e2e}_{\min}(s,i,t) = 120$. Node i can calculate the end-to-end metrics of the neighbors' BQ and LC paths based on the path information, the link metrics, and the pre-computed information. Among the paths, the LC path of node j_1 satisfies the delay requirement and has a smaller end-to-end cost than $C^{e2e}_{\min}(s,i,t)$. So $C^{e2e}_{\min}(s,i,t)$ is updated to the end-to-end cost of the j_1's LC path, 100 ms.

Fig. 1(a) shows the selective probing procedure using the QLCP. In this case, the end-to-end LC path delays of the neighbor nodes j_1 (i.e., 160) and j_2 (i.e., 160) satisfy the QoS requirement (i.e., 170). So, the PCSP scheme forwards a *find_msg* to node j_1 and then excludes node j_1 from the probing process, as only the end-to-end LC path cost of node j_1 (i.e., $C_{e2e,i}(LC_{j1}(t)) = 100$) is less than or equal to the updated $C^{e2e}_{\min}(s,i,t)$ value.

Fig. 1(b) shows the probing set selection process at node i. As node j_1 is excluded from the probing set in the above procedure, node i performs the probing condition test for nodes j_2 and j_3. Only node j_3 is included in the probing set, as $D_{e2e,i}(LC_{j3}(t)) \leq D_R(s,t)$ and $C_{e2e,i}(LC_{j3}(t)) \leq C^{e2e}_{\min}(s,i,t)$.[10] At node j_3, a *probe_msg* is forwarded to the destination node t directly and the optimal path $p(s\cdots i, j_3, t)$ (with $D_{e2e} = 130, C_{e2e} = 95$) is determined out of the two found feasible paths, $p(s\cdots i, j_3, t)$ and $p(s\cdots i, j_1, t)$.

By employing the two approaches above, we can exclude from the probing process the paths whose end-to-end cost is higher than that of a readily determined feasible path or whose end-to-end QoS does not satisfy the QoS requirement. It helps to reduce the message overhead especially when the QoS requirement is not tight as there exist a large number of QoS-satisfying

---

[6] $\{e2e, i\}$ indicates that the corresponding end-to-end value is estimated at node i.

[7] That is, there exists a path $p \in \{LC_i(t), BQ_i(t)\}$ for which $Q_{e2e,i}(p)$ satisfies $Q_R(s,t)$ and $C_{e2e,i}(p) < C^{e2e}_{\min}(s,i,t)$.

[8] The second condition is a necessary condition for the existence of QoS-satisfying paths through node j.

[9] Note that it is not necessarily optimal for the paths that do not pass through this node.

[10] For node j_2, $D_{e2e,i}(LC_{j2}(t)) \leq D_R(s,t)$ but $C_{e2e,i}(LC_{j2}(t)) \geq C^{e2e}_{\min}(s,i,t)$.

Table 1. Information available at node i.

Node	Information in *probe_msg*	Link metric form node i		Pre-computed information				End-to-end metrics		QoS
		Delay	cost	Path	Next hop	Delay	Cost	Delay	Cost	
i	$D(p(s,i))$: 80 $C(p(s,i))$: 60 $D_R(s,t)$: 170 $C_{\min}(s,i,t)$: 120	-	-	BQ	j_2	20	70	100	130	O
				LC	j_3	100	30	180	90	X
j_1		40	20	BQ	j_2	30	50	150	130	O
				LC	t	40	20	160	100	O
j_2		10	40	BQ	t	10	30	100	130	O
				LC	k	70	25	160	125	O
j_3		20	20	BQ	j_2	20	50	120	130	O
				LC	k	80	10	180	90	X

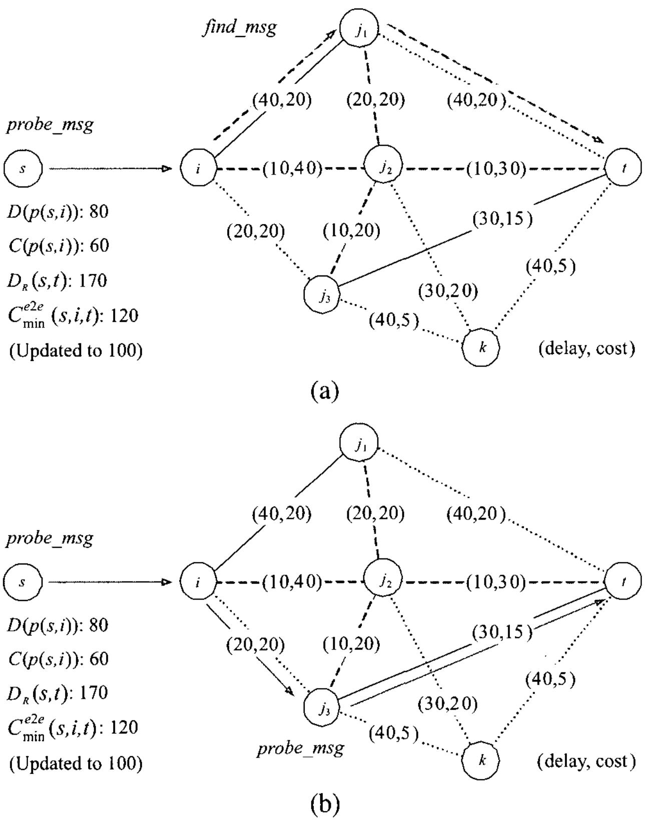


Fig. 1. Illustration of selective probing procedure (a) using a QoS-satisfying LC path and (b)using the least-cost value of the determined feasible paths.

LC or BQ paths. Since a path is excluded only if a better path exists, the optimal path is never excluded during the probing process. Therefore, the path selected among all the received feasible paths at the destination node thus becomes the optimal one. In this way, the PCSP scheme can always guarantee the cost optimality.

D. Considerations for Imprecise State Information

The major change required for the imprecise state information model is the selection rule. The PCSP scheme applies two different types of QoS-satisfying conditions—the strict QoS-satisfying condition as described in (9) and the loose QoS-satisfying condition as described in (10).

The PCSP scheme uses the strict QoS-satisfying condition when the result may exclude some paths during the probing process. There are two such cases: The first is when calculating the least-cost value of the determined feasible path, $C^{e2e}_{\min}(s,i,t)$. In this case, every path whose end-to-end cost is higher than this value is excluded from the probing process. The second is when selecting QoS-satisfying LC paths, as the selected LC path excludes the other paths passing through the current node. If the strict QoS-satisfying condition is not applied to these two cases, a possible optimal path (i.e., the path whose end-to-end cost is less than that of the selected LC path or the least-cost value) may be left out during the probing process due to the impreciseness of state information.

The loose QoS-satisfying condition is applied to the case when it is used to select the paths to probe. A neighbor node can be a member of the probing set if its BQ path satisfies the QoS requirement loosely, with other requirements satisfied as well. This adaptation of the QoS-satisfying conditions enables the PCSP scheme to maintain the optimal properties even with imprecise state information.

IV. ROUTING PROCESS OF THE PCSP SCHEME

The routing process of the PCSP scheme consists of two phases: One is the *probing phase* to find feasible paths and select the optimal path, and the other is the *acknowledgement phase* to set up routing tables and reserve resources.

The *probing phase* is subdivided into three processes—the *pre-process* at the source node, the *probing process* at the intermediate nodes, and the *post-process* at the destination node. The operation at each of the three processes is detailed in the following subsections.

```
1:  Pre_Process(source node $s$, destination node $t$,
            qos_requirement $D_R(s, t)$, time_out_value $T\_to$)
2:      If ($D(LC_s(t)) + \Delta D(LC_s(t)) <= D_R(s,t)$) then
3:          Forward_msg(set_msg, $s$, $N(LC_s(t))$);
4:      else if ($D(BQ_s(t)) - \Delta D(BQ_s(t)) <= D_R(s, t)$) then
5:          $C_{\min}$ = MAX;    Probing_Process(probe_msg, $s$);
6:      else Reject_request;
```

(a)

```
1:  Probing_Process(message probe_msg, node $i$)
2:      $n = 0$;
3:      For all the neighbor node $j$ except the parent node,
4:          If ($D_{e2e,i}(LC_j(t)) + \Delta D(LC_j(t)) <= D_R(s, t)$
                && $C_{e2e,i}(LC_j(t)) <= C^{e2e}_{\min}(s, i, t)$) then
5:              $n = j$;  $C^{e2e}_{\min}(s, i, t) = C_{e2e,i}(LC_j(t))$;
6:          If ($D_{e2e,i}(BQ_j(t)) + \Delta D(BQ_j(t)) <= D_R(s, t)$
                && $C_{e2e,i}(BQ_j(t)) <= C^{e2e}_{\min}(s, i, t)$) then
7:              $n = 0$;  $C^{e2e}_{\min}(s, i, t) = C_{e2e,i}(BQ_j(t))$;
8:      If ($n >= 0$) then Forward_msg(find_msg, $i$, $n$);
9:      For all the neighbor node $j$ except $n$ and the parent node,
10:         If ($D_{e2e,i}(BQ_j(t)) - \Delta D(BQ_j(t)) <= D_R(s, t)$
                && $C_{e2e,i}(LC(t)) <= C^{e2e}_{\min}(s, i, t)$) then
11:             Forward_msg(probe_msg, $i$, $j$);
```

(b)

```
1:  Post_Process(message $m$, destination node $t$)
2:      If (time_out) then
3:          If ($P_{sel}$ != NULL ) then
4:              Forward_msg(ack_msg, $t$, parent_node($P_{sel}$));
5:              return success;
6:          else return failure;
7:      If ($m$ == set_msg) then
8:          Forward_msg(ack_msg, $t$, parent_node(path($m$)));
9:          return success;
10:     If ($m$ == probe_msg or $m$ == find_msg) then
11:         If ($C$(path($m$)) < $C_{sel}$) then
12:             $P_{sel}$ = path ($m$);  $C_{sel}$ = $C$(path($m$));
13:             return waiting;
```

(c)

Fig. 2. Pseudo code of the PCSP scheme: (a) Pre-process at the source node, (b) probing process at an intermediate node i, (c) post-process at the destination node.

A. Pre-Process at the Source Node

As the *pre-process* at the source node s, the PCSP scheme prepares the *probing process*. The source node determines whether the probing process is needed or not by checking the end-to-end QoS metrics of the LC and BQ paths. Fig. 2(a) shows the pseudo code of the pre-process in the case of the delay-constrained least-cost routing.[11]

The PCSP scheme checks the end-to-end QoS metric of the LC path (line 2 and 3): If the LC path satisfies the QoS requirement $Q_R(s, t)$ strictly (i.e., strict condition), it is determined as the optimal path. In this case, probing-process is not needed. The source node generates and forwards a *set_msg* to the destination node through the LC path to notify that the particular path is optimal.

Otherwise, the PCSP scheme checks the end-to-end QoS metric of the BQ path (lines 4 and 5): If the BQ path satisfies $Q_R(s, t)$ loosely (i.e., loose condition), the PCSP scheme prepares the probing process. The least-cost value of the determined feasible paths, $C^{e2e}_{\min}(s, i, t)$, is initialized to infinite (maximum) value. After the initialization, the source node performs the probing process.

If the BQ path does not satisfy the requirement loosely, the call is rejected as there exists no feasible path (line 6).

The forward_msg(message m, transmit_node i, receive_node j) function delivers a message m from node i to its neighbor node j. During the delivery, this function updates the cost and QoS metrics of the message $C(p(s, i)$ and $Q(p(s, i))$, using the real values of the link metrics $C(i, j)$ and $Q(i, j)$. Therefore, as mentioned in Section II-A, $Q(p(s, i))$ is an exact information gathered during the probing process.

B. Probing Process at an Intermediate Node

In the *probing process*, the PCSP scheme finds the feasible paths in parallel and hop-by-hop manner. An intermediate node i that receives a *probe_msg* p selects the neighbor nodes to send a *find_msg* or a *probe_msg*. Fig. 2(b) shows the pseudo code.

First, the PCSP scheme checks the end-to-end QoS metric of the BQ and LC paths of its neighbor nodes and updates $C^{e2e}_{\min}(s, i, t)$ value (lines 3–7): If any BQ path or LC path satisfies $Q_R(s, t)$ strictly and has a smaller value of end-to-end cost, $C^{e2e}_{\min}(s, i, t)$ is updated. During this process, the QLCP of its neighbor node with the end-to-end cost equal to the updated $C^{e2e}_{\min}(s, i, t)$ value is found.

Secondly, the PCSP scheme forwards a *find_msg* to the neighbor node n having the selected QLCP if one exists (line 8): This message is then forwarded to the destination node through the selected QLCP.

Third, the PCSP scheme selects the probing set $R_i(t)$ among the neighbor node set N_i and sends *probe_msg*s to those nodes (lines 9–11): The node n to which a *find_msg* is already sent and the parent node are excluded. The condition of the probing set is that the BQ path should satisfy $Q_R(s, t)$ loosely and the end-to-end cost of the LC path should not be greater than $C^{e2e}_{\min}(s, i, t)$. Consequently, in the case of delay-constrained QoS routing, it becomes

$$\begin{aligned} R_i(t) = \{j; j \in (N_i - \{k, n\}) \mid \\ C_{e2e,i}(LC_j(t)) \le C^{e2e}_{\min}(s, i, t) \text{ and} \\ D_{e2e,i}(BQ_j(t)) - \Delta D(BQ_j(t)) \le D_R(s, t)\}. \end{aligned} \tag{13}$$

---

[11] The procedure for the bandwidth-constrained routing is identical except for the QoS-satisfying conditions.

C. Handling of Multiple Messages

During the probing process, a node may receive two or more messages (*probe_msg*s or *find_msg*s). This may happen frequently, causing additional message overhead, especially when the state information is highly imprecise. In the case of the TBP scheme, it records the links through which a probe message was forwarded during the routing process and does not forward additional messages through these links [19]. The ETBR scheme applies color-based ticket distribution scheme and uses historical results [20]. But these algorithms suffer from performance degradation caused by the message drop.

In the PCSP scheme, the handling mechanism of multiple messages is designed to maintain the optimal properties. When a node receives multiple messages during the routing process, it checks the usefulness of the new message. The newly arrived message is discarded if there exists readily processed message m which has less cost and better additive QoS metric (i.e., $C(p(s,i)_m) \leq C(p(s,i)_{new})$ and $Q_{additive}(p(s,i)_m)$ better than $Q_{additive}(p(s,i)_{new})$). The comparison of non-additive QoS metric is meaningless as the non-additive metric does not take effect on the routing process any further if it satisfied the requirement already.

In this handling mechanism, a node can receive exponential amount of messages. We can apply a distributive mechanism reducing the message overhead at each intermediate node such as the logarithmic algorithm [25].[12] However, this kind of algorithm processes the messages selectively, resulting in the sacrifice of the cost optimality and delays in routing process. Furthermore, as the message processing mechanism is not coordinated between intermediate nodes, the usual performance can be degraded than that of other coordinated algorithm such as TBP [19]. We will discuss the extended PCSP schemes that reduce the worst-case message complexity using additional techniques such as TBP in Section VI.

D. Post-Process at the Destination Node

At the destination node t, the PCSP scheme handles the forwarded messages: *set_msg*, *find_msg*, and *probe_msg* (Fig. 2(c)). If a *set_msg* received (line 7–9), the path through which the message is forwarded is selected as the optimal path. Then, the *acknowledgement phase* is initiated immediately. When a *probe_msg* or a *find_msg* received (lines 10–13), the received path is newly selected if it has a smaller cost value than that of the previously selected path. If the message is the first one, the PCSP scheme starts a timer. This timer is used to limit the waiting time for the messages. The time-out value can be calculated using some given equations, such as $k \cdot RTT(s,t)$ (in the bandwidth constrained QoS routing case) or $k \cdot D_R(s,t)$ (in the delay constrained QoS routing case), for the scaling factor k, the round trip time $RTT(s,t)$ and the delay requirement $D_R(s,t)$. The scaling factor is determined at the source node and transported by the messages.

When time-out occurs (lines 2–6) and there exists a selected path, the *acknowledgement phase* is initiated with the selected path.

In the *acknowledgement phase*, an *ack_msg* with success flag is sent backward to the source node through the selected optimal path. Each node receiving the *ack_msg* sets up the routing table and reserves the resources. If there is no enough resource to support this call, a *ack_msg* with failure flag is generated and forwarded to both the source and the destination nodes through the selected path. A node receiving such *ack_msg* clears the routing table and releases the resource.

V. PROPERTIES OF THE PCSP SCHEME

As the PCSP scheme uses carefully designed selective probing approaches and routing algorithms, it retains a number of desirable properties as follows:

1) The selective probing algorithm of PCSP scheme guarantees a loop-free operation: If a *probe_msg* arrives at an already-visited node, it is discarded by the handling mechanism of multiple messages (see Section IV-C), thereby blocking the possibility of forming loops.
2) The PCSP scheme always finds a feasible path if one exists: It checks the BQ path of the source node with the loose QoS-satisfying condition. So, if this path happens not to satisfy the loose condition, it means that there exists no path satisfying the requirements.
3) If a path is excluded from the probing process, it means that there exists a feasible path readily found with a smaller end-to-end cost: A path is excluded from the probing process only if it has a higher end-to-end cost than the least-cost value of the determined feasible paths.
4) The selected path is always cost-optimal: According to the third property, the optimal path is never excluded from the probing process. So, it is supposed to be selected at the destination node unless any transmission error occurs.
5) The PCSP scheme has no auxiliary parameter to optimize: The only required parameter is the time-out parameter that may be set in a reasonable range considering the call setup time.

VI. EXTENSIONS OF THE PCSP SCHEME

Despite all the meritorious properties, the PCSP scheme may suffer from high message overhead due to its conservative search process. When the state information is extremely imprecise and the network size is large, the PCSP scheme may generate a large number of messages as it does not limit the number of messages explicitly. The message overhead of PCSP scheme still retains exponential behavior in the worst case.

In such a case, it is desirable to use other heuristics, in addition to the PCSP scheme, that define additional probing conditions or message control mechanisms limiting the number of messages explicitly at the sacrifice of the optimal properties. As the PCSP scheme defines only qualitative (or conditional) probing conditions, the PCSP scheme can be easily extended by employing any quantitative heuristics along with its original algorithm. In order to bound the worst-case message overhead, we extend the original PCSP scheme by adopting the basic idea of

---

[12] In the logarithmic algorithm, each node counts the number of arrival of useful messages and handles a message when the counter value reaches the number of a natural power of two, i.e., only the first, the second, the forth, etc.

the TBP algorithm as an additional quantitative heuristic, naming it PCSP scheme with ticket (PCSP_T).

The PCSP_T algorithm determines a certain number N_0 of tickets per connection request and distributes the tickets during the probing process as the TBP scheme [20]. It has two kinds of tickets, *yellow tickets* and *green tickets*. The purpose of yellow tickets is to maximize the probability of finding feasible paths. So, they prefer the paths having better QoS. In contrast, the purpose of green tickets is to maximize the probability of finding low-cost paths. They prefer the paths having smaller costs. As an example, Fig. 3 [20] shows the initial number of the both tickets, Y_0 for the yellow tickets and G_0 for the green tickets determined at the source node considering the QoS variation. In the figure, Φ defines the maximum number of yellow tickets, θ the threshold value specifying the sufficiently-large range for the QoS requirement, and Ω the maximum number of green tickets. During the probing process, each node receiving the probe message p with $Y(p)$ yellow tickets and $G(p)$ green tickets, redistributes the tickets to the neighbors as follows

$$Y(p_j) = \frac{(D(i,j) + D(p_j(t))^{-1}}{\Sigma_{j' \in R_i(t)}(D(i,j') + D(p'_j(t))^{-1}} Y(p)$$
$$G(p_j) = \frac{(C(i,j) + C(p_j(t))^{-1}}{\Sigma_{j' \in R_i(t)}(C(i,j') + C(p'_j(t))^{-1}} G(p). \quad (14)$$

In contrast to the original TBP scheme, the PCSP_T scheme should consider the found feasible LC or BQ paths which determine the $C^{e2e}_{\min}(s,i,t)$ value. As they exclude other paths from the probing process, a probe message or a find message with at least one ticket should be sent to them. But, in many cases, they gain no ticket since they have moderate cost and moderate QoS metrics, compared with other paths. Therefore, the PCSP_T scheme assigns one ticket to the path that updates the $C^{e2e}_{\min}(s,i,t)$ value lastly and decreases the total ticket number before distributing the tickets.[13] The found path also participates in the ticket distribution process to gain additional tickets according to (14), thereby reducing the message overhead further.

Though the PCSP_T scheme bounds the number of messages existing in the network at a time, the total number of messages is not still bounded as a probe message with a ticket can traverse the longest path between the source and the destination nodes. An efficient method may be found in limiting the hop count: We put an additional condition that the distance (i.e., hop) of the selected paths should not exceed a pre-determined value above the minimum source-to-destination distance. In fact, it makes sense to limit the hop count of the selected path as an excessively long-distance path can block other calls. This hop-limited scheme, named PCSP scheme with ticket and hop-limit (PCSP_TH) helps to avoid setting up extremely long-distance paths. As a result, this PCSP_TH scheme bounds the worst-case message overhead below a controllable value, $N_0 \cdot H$, where N_0 is the initial ticket number and H is the hop limit.

We will confirm the performance of the PCSP_T and the PCSP_TH schemes in comparison with the PCSP and other existing schemes through simulations in the next section.

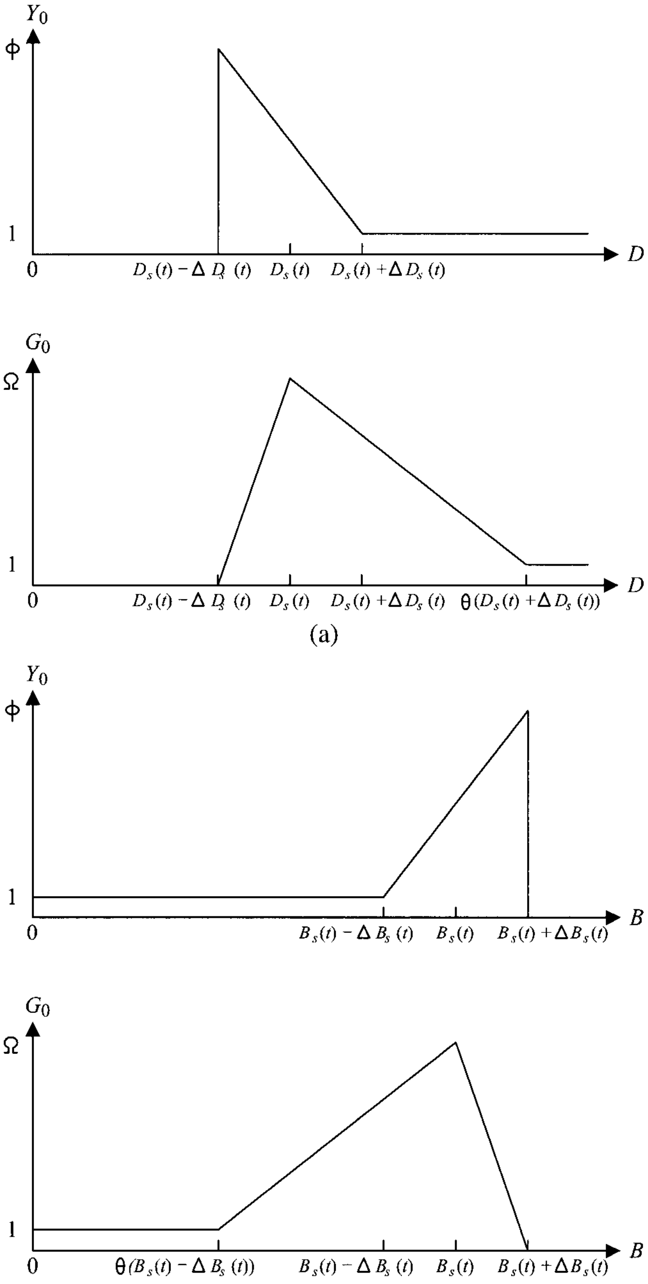


Fig. 3. Initial ticket numbers (Y_0: Yellow ticket, G_0: Green ticket): (a) Delay-constrained routing case, (b) bandwidth-constrained routing case.

VII. SIMULATION RESULTS

Now, we examine the performance of the PCSP scheme and its extensions through computer simulations. We consider the delay-constrained least-cost routing and the bandwidth-constrained least-cost routing as the representative additive and non-additive QoS routing problems. For the performance estimates, we employ the following three measures—the *success ratio* and the *average path cost* as the optimality metrics and the *average message overhead* as the efficiency metric, which are

[13] If the found path is an LC path, a green ticket is assigned. Otherwise, the found path is a BQ path, so a yellow ticket is assigned.

respectively defined as follows[14]

$$success\ ratio = \frac{number\ of\ accepted\ connections}{total\ number\ of\ connection\ requests} \tag{15}$$

$$average\ path\ cost = \frac{total\ cost\ of\ all\ established\ routing\ paths}{number\ of\ established\ routing\ paths} \tag{16}$$

$$average\ message\ overhead = \frac{total\ number\ of\ messages\ sent}{total\ number\ of\ connection\ requests}. \tag{17}$$

For the network topology, we generated power-law topologies randomly using the INET topology generator [28]. We varied the network size among 30, 50, 100, 300, 500, and 1,000 nodes and generated 100 topologies per each network size. As the INET topology generator requires a minimum node number of 3,038, we modified the INET topology generator to support small networks without violating the power-law property.

We set the bandwidth of each link to 155 Mbps. We assumed that the link cost C, the advertised delay D_{ad}, and the advertised residual bandwidth B_{ad} are uniformly distributed in the ranges of [1, 200], [0, 50 ms], and [1, 100 Mbps], respectively, with the real values D_{real} and B_{real} uniformly distributed in the ranges of $[(1-\xi)D_{ad}, (1+\xi)D_{ad}]$ and $[(1-\xi)B_{ad}, (1+\xi)B_{ad}]$, respectively. We varied the *imprecision ratio* ξ among the values 0, 10%, 20%, 30%, and 50%.

We set up 100 different configurations (in terms of the advertised and the real state information) per each imprecision ratio value for each topology and carried out 1,000 simulation runs for each configuration. At each simulation run, we randomly selected the source and the destination nodes. For the delay-constrained routing, we used the delay requirements of the common real-time services 50, 80, 100, 120, 150, 200, 400, and 500 ms. For the bandwidth-constrained routing, we used the bandwidth requirements of 1, 2, 5, 10, 20, 50, 70, and 100 Mbps.

We carried out computer simulations for four different schemes—the PCSP scheme, the flooding based (FB) scheme [18], the ticket based probing (TBP) scheme [19], and the source routing (SR) scheme [15]—and their extensions. We selected the FB scheme as the reference of optimality that guarantees the ideal success ratio and the cost-optimal search; and the TBP scheme as the reference of effectiveness that achieves low message overhead with acceptable level of degradation in optimal properties. For the TBP scheme, we employed the ETBR algorithm [20] and set Φ (maximum allowable number of yellow tickets) $= 4$, θ (scaling factor) $= 1.5$, Ω_l (minimum number of green tickets) $= 2$, and Ω_h (maximum allowable number of green tickets) $= 6$. For the SR scheme, we applied an exhaustive search technique to find QoS-satisfying cost-optimal path. We did not apply any heuristics to reduce computational complexity because computational complexity is of no concern to the distributed schemes. In the route setup procedure, we assumed that the setup-request message is forwarded to the destination node and then the acknowledgement message is returned backward to the source node.[15] When simulating the extended algorithms PCSP_T and PCSP_TH, we set the ticket related parameters to be the same as for the TBP scheme. For the hop-limiting extensions, PCSP_TH, TBP_H, and SR_H, we set the maximum allowable hop limit to 2 plus the minimum source-to-destination distance.

There also exist outstanding QoS multicast routing algorithms. However, the main issue of the QoS multicast routing algorithms is how to merge the QoS-satisfying route of each destination node to a QoS-satisfying tree. Consequently, if there is only one destination (unicast routing case), the QoS multicast routing algorithms act like a simple source routing (QDMR algorithm [26]) or limited flooding algorithms (QMRP-k algorithm [27]). Though the QMRP-k algorithm uses limited flooding, the algorithm is much simpler than that of the TBP algorithm [19] proposed by the same authors. We simulated QDMR for delay-constraint least-cost routing problem and QMRP-k for bandwidth-constraint least-cost routing problem. But, due to the reasons mentioned above, the performances are worse than those of the unicast routing algorithms. Therefore, we do not include the performance curve of the QoS multicast routing algorithm in the simulation graphs.

Figs. 4–6 and 8–10 show the simulation results of the delay-constrained least-cost routing problem and the bandwidth-constrained least-cost routing problem, respectively for 100-node topology. Figs. 7 and 11 show the simulation results of the 50% imprecision ratio for various different network sizes. Each simulation result consists of the *success ratio*, *relative success ratio*, *average path cost*, *relative average path cost*, *probability of non-optimal search*, and *average message overhead*, each of which is composed of three graphs with three different imprecision ratios, 0%, 20%, and 50%. The "relative" values above are those computed as the ratio to the corresponding optimal references (i.e., the FB scheme). The probability of non-optimal search indicates the fraction that the cost of the selected path is greater than the cost of the optimally-selected path by the FB scheme.

In the case of the delay-constrained least-cost routing, Figs. 4(a) and 4(b) plot success ratio and relative success ratio, respectively. We observe that the PCSP scheme exhibits an optimal performance, with negligibly small variance over different imprecision ratios. This implies that the PCSP scheme always finds out a feasible path if one exists, even for highly imprecise information. And even the extended PCSP schemes exhibit only about 1% failure. In contrast, the performance of the SR schemes degrades rather severely as the imprecision ratio increases—when the imprecision ratio is 50% and the delay requirement is tight as 50, the SR schemes fail about 25%.

Fig. 5 show the cost-optimal performances. It is shown as if the SR scheme performs best but it is misleading as the SR scheme fails to find all the optimal paths (by about 10% in average) and the unfound paths usually have high cost (Fig. 5(b)). The TBP scheme has an additional cost of about 15% (Fig. 5(a)), and about 30% of the selected paths are not cost-optimal

---

[14] For a fair comparison, in (17), we count a message as l messages if it traverses a path consisting of l hops.

[15] The SR scheme shows a minimum bound of the message overhead.

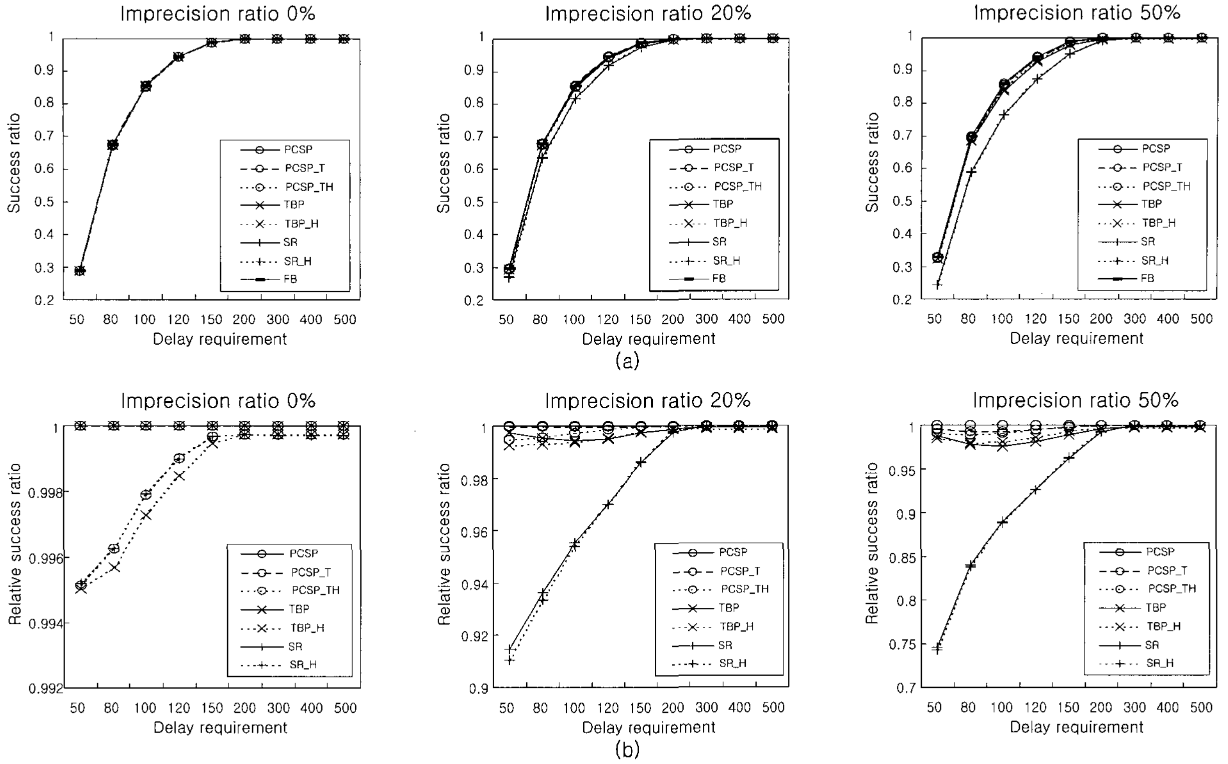


Fig. 4. Simulation results for the delay-constrained least-cost routing: (a) Success ratio, (b) relative success ratio.

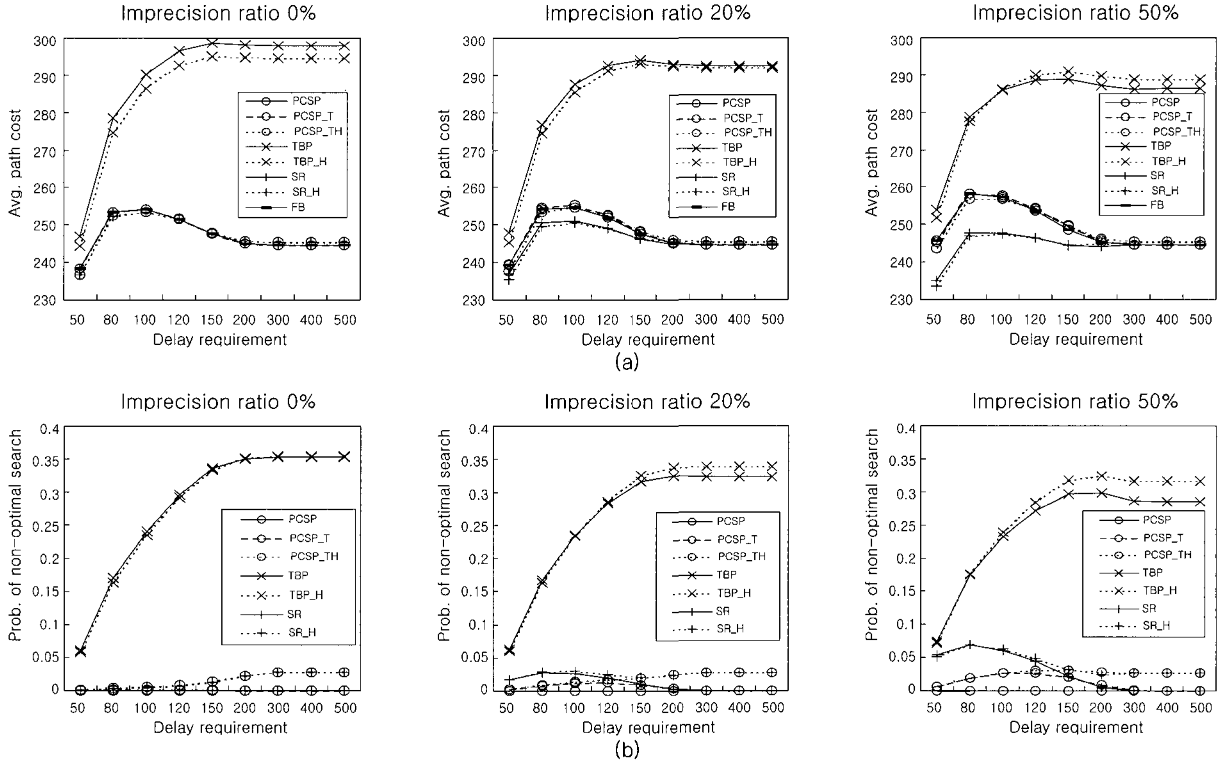


Fig. 5. Simulation results for the delay-constrained least-cost routing: (a) Average path cost, (b) probability of non-optimal search.

(Fig. 5(b)). In contrast, the PCSP scheme exhibits optimal performance in the average path cost—the same as that of the FB scheme, which indicates that the PCSP scheme always determines the optimal path from the given information, even at high imprecision ratio.

Fig. 6 plots the average message overheads. The FB scheme is

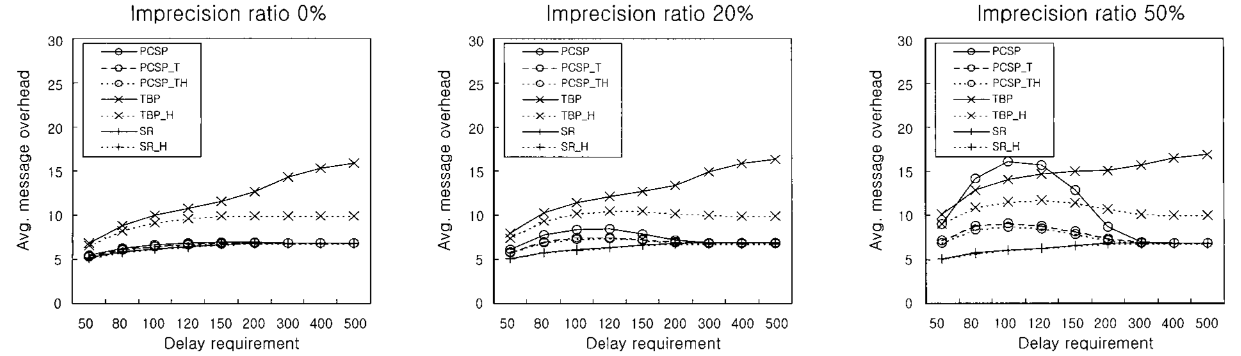


Fig. 6. Simulation results for the delay-constrained least-cost routing: Average message overhead.

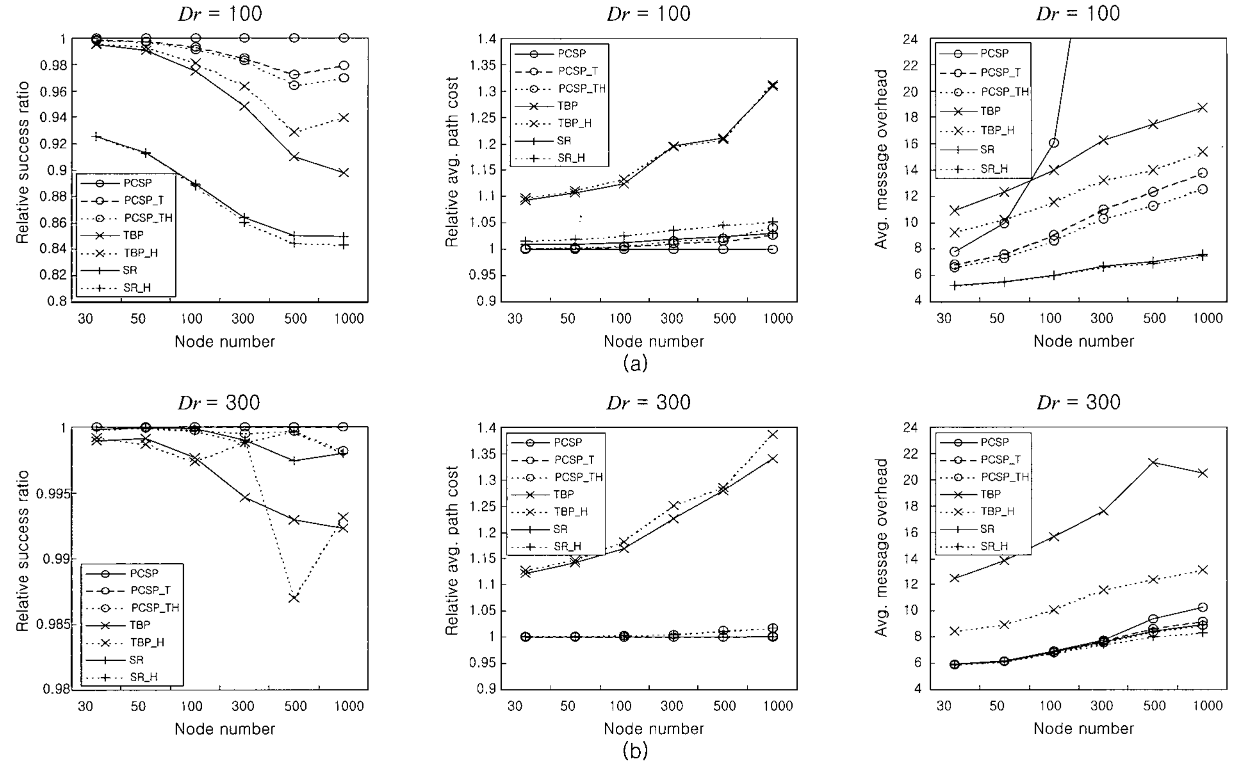


Fig. 7. Simulation results for the delay-constrained least-cost routing (50% imprecision ratio): (a) Tight requirement case (100 ms), (b) loose requirement case (300 ms).

excluded because its overhead is too high to put in the same figure. We observe that the PCSP scheme reduces the average message overhead of the existing distributed QoS routing schemes to a manageable level. The average message overhead of the PCSP scheme is slightly larger than that of the SR scheme for the cases of small delay requirement but becomes almost the same when the delay requirement increases over 300 ms. This phenomenon may be interpreted to happen because the LC and BQ paths of the source node or the intermediate nodes are likely to satisfy the QoS requirement and the PCSP scheme uses such paths effectively. However, when the QoS requirement is moderate and the impreciseness is high (e.g., 100–200 ms requirement at the 50% imprecision ratio), the message overhead of the PCSP scheme exceeds that of the TBP scheme. It is the drawback of using conditional probing approach, which can be mitigated by adopting the quantitative extensions of the PCSP scheme, PCSP_T and PCSP_TH. We observe that the PCSP_T and PCSP_TH schemes bound the message overheads much below those of the TBP schemes at only small performance degradation—about 3% decrease in relative success ratio (see Fig. 4(b)) and about 3% increase in non-optimal search probability (see Fig. 5(b)).

Fig. 7 shows the simulation results with 50% imprecision ratio for various different network sizes for a tight delay requirement (100 ms) and a loose delay requirement (300 ms). This tight delay requirement is selected among the values for which PCSP schemes exhibit a worst performance (i.e., 80, 100, and 120 ms). Both cases show that the PCSP scheme always maintains its optimal properties but fails to bound the message overhead in the worst case (Fig. 7(a)). However, the PCSP_T and PCSP_TH schemes always out-perform the TBP and TBP_H

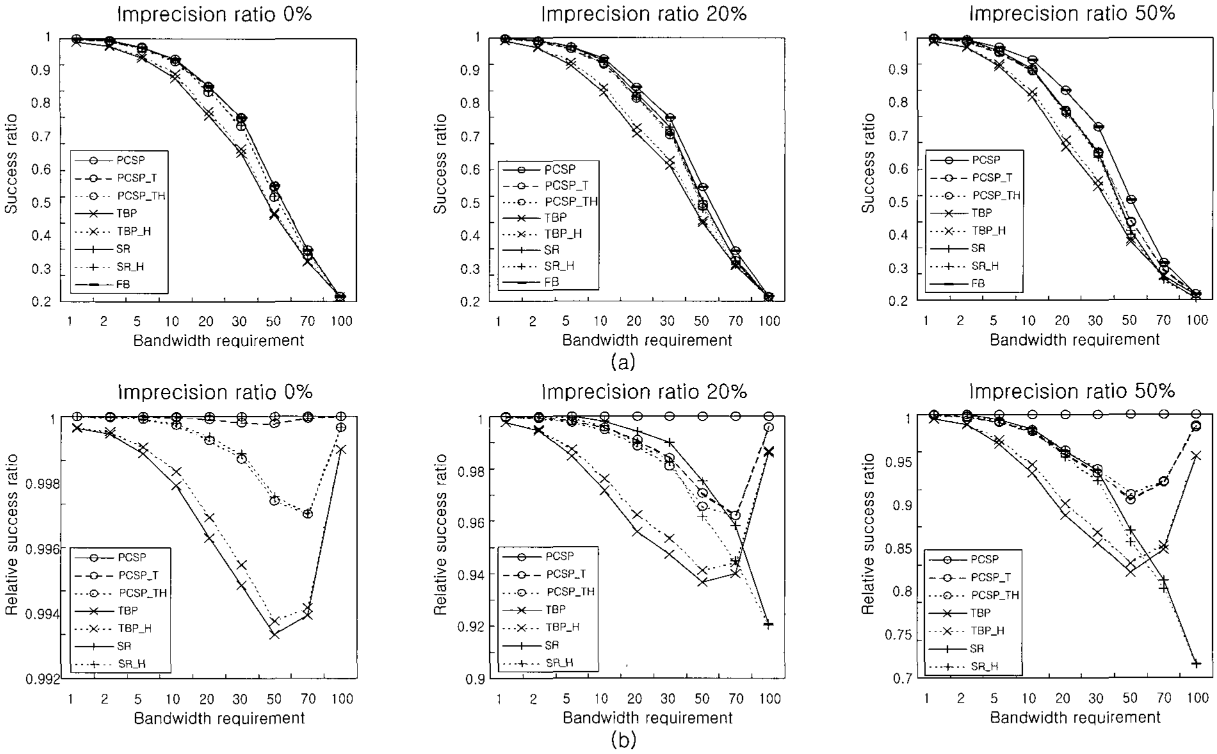


Fig. 8. Simulation results for the bandwidth-constrained least-cost routing: (a) Success ratio, (b) relative success ratio.

schemes regardless of the network size even in the worst case. Furthermore, even is the worst-case (Fig. 7(a)), the performance degradations are quite small: PCSP_TH failed to find about 3% feasible paths that were found by the optimal scheme and the found paths have more cost by about 3%.

In the case of bandwidth-constrained least-cost routing, Figs. 8(a) and 8(b) show the success ratio and the relative success ratio, respectively. We observe that the SR and the TBP schemes frequently fail to find some feasible paths. The SR scheme suffers from performance degradation because the non-additive QoS metric is more sensitive to the impreciseness than the additive metric.[16] In the TBP scheme case, the performance degrades due to the ineffectiveness of the quantitative probing scheme that uses non-additive QoS metric of LC paths as the guideline. However, we observe that the PCSP scheme that uses qualitative probing scheme maintains an ideal success ratio for the non-additive QoS requirement even at high imprecision ratio.

Fig. 9(a) shows the average path cost, and Figs. 9(b) and 9(c) respectively show the average path cost and the probability of non-optimal search only for the cases that all the three schemes (i.e., SR, TBP, and PCSP schemes) succeed to find a feasible path. We observe that the performances of the SR and TBP schemes appear to outperform the PCSP scheme but, in reality, they fail to find some optimal paths with high cost which the PCSP schemes successfully find. This is well demonstrated in the relative success ratio in Fig. 8(b) and the average path cost for the all-success cases in Fig. 9(b). In those figures, the PCSP scheme turns out to perform best and determines the optimal paths without failure, even with highly imprecise state information.

Fig. 10 plots the average message overhead for the all-success cases. The FB scheme is not shown in the figure as its overhead is too high. We observe that the average message overhead of the PCSP scheme increases substantially as the imprecision ratio increases. This is an inevitable phenomenon of the qualitative probing approaches, which happens as the price for maintaining optimal properties. It can be relieved by adopting quantitative extensions of the PCSP scheme. As shown in Fig. 10, the extended schemes bound the message overhead at the sacrifice of the relative success ratio by about 15% (see Fig. 8(b)) and the probability of non-optimal search by about 8% (see Fig. 9(c)).

Fig. 11 shows the simulation results with 50% imprecision ratio for various different network size for a tight bandwidth requirement (50 Mbps) and a loose bandwidth requirement (5Mbps). This tight delay requirement is selected among the values for which PCSP schemes exhibit a worst performance (i.e., 30, 50, and 70 Mbps). Both cases show that the PCSP scheme always maintains its optimal properties but fails to bound the message overhead in the worst case (Fig. 11(a)). However, the PCSP_T and PCSP_TH schemes always outperform the TBP and TBP_H schemes regardless of the network size even in the worst case: The PCSP extensions find more optimal paths with less message overheads than the TBP and TBP extension.

The PCSP schemes with additional heuristics, PCSP_T and PCPS_TH, try to reduce the worst-case message overhead with the sacrifice of the other performance metrics such as success ratio, and cost-optimality. Therefore, the small effects on the

---

[16] Note that the variance of the summation of random numbers is less than the variance of the minimum or the maximum of them.

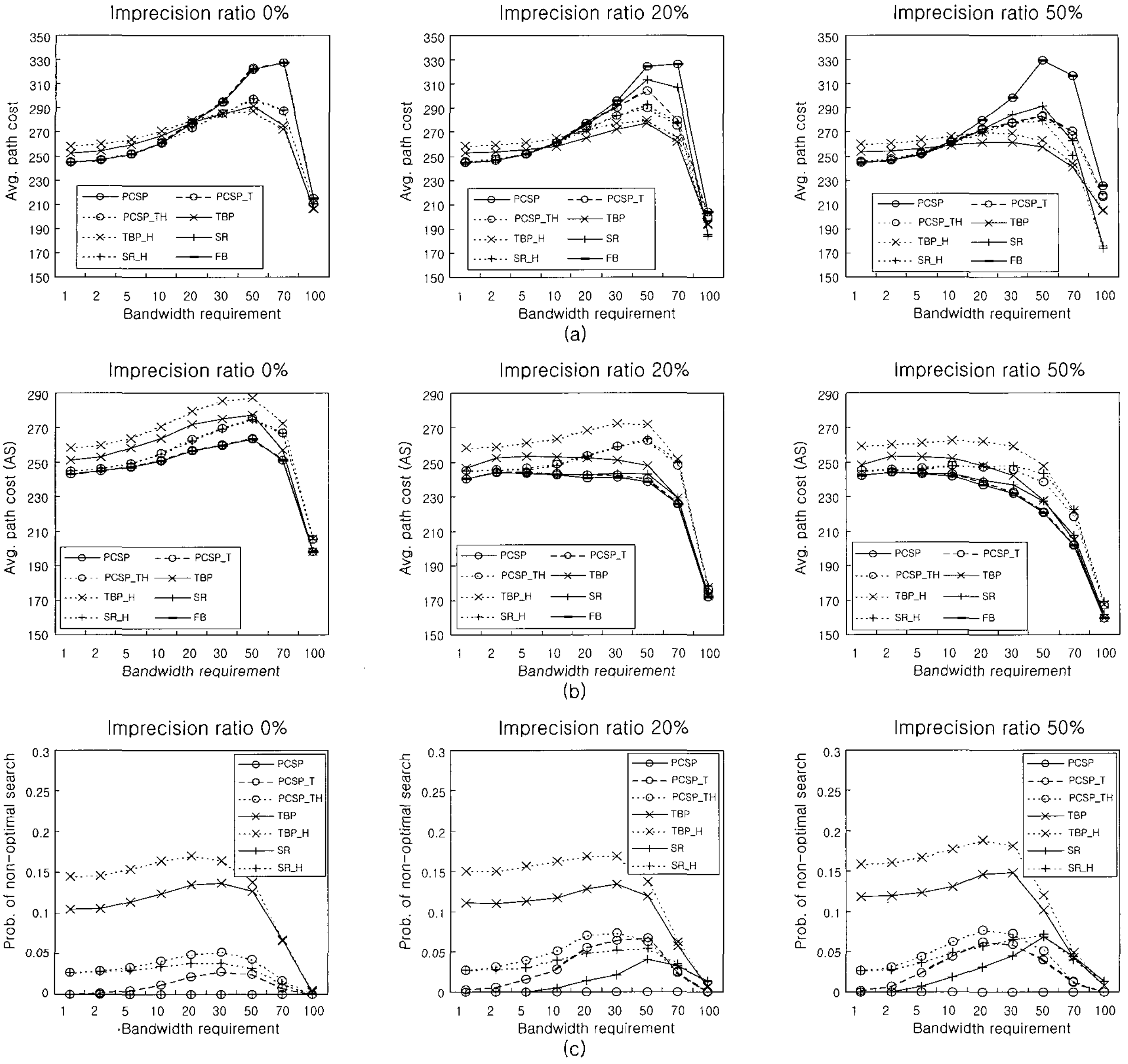


Fig. 9. Simulation results for the bandwidth-constrained least-cost routing: (a) Average path cost, (b) average path cost (all success), (c) probability of non-optimal search (all success).

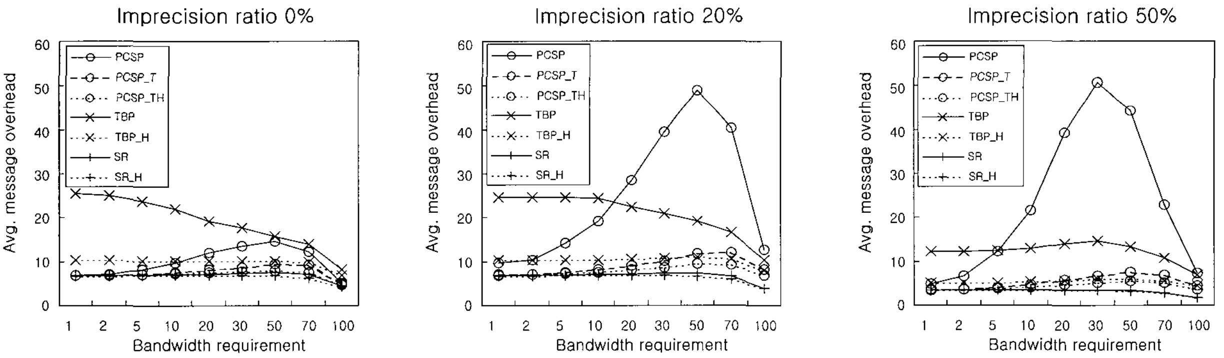


Fig. 10. Simulation results for the bandwidth-constrained least-cost routing: Average message overhead (all success).

success ratio and cost-optimality are desired, which means that the additional heuristics slightly degrade the optimal properties of the PCSP scheme. The significant effects of using additional heuristics are shown in the efficiency metric, average message overhead. In the Figs. 6 and 10, the extended PCSP schemes reduce the worst-case—when the state information is highly

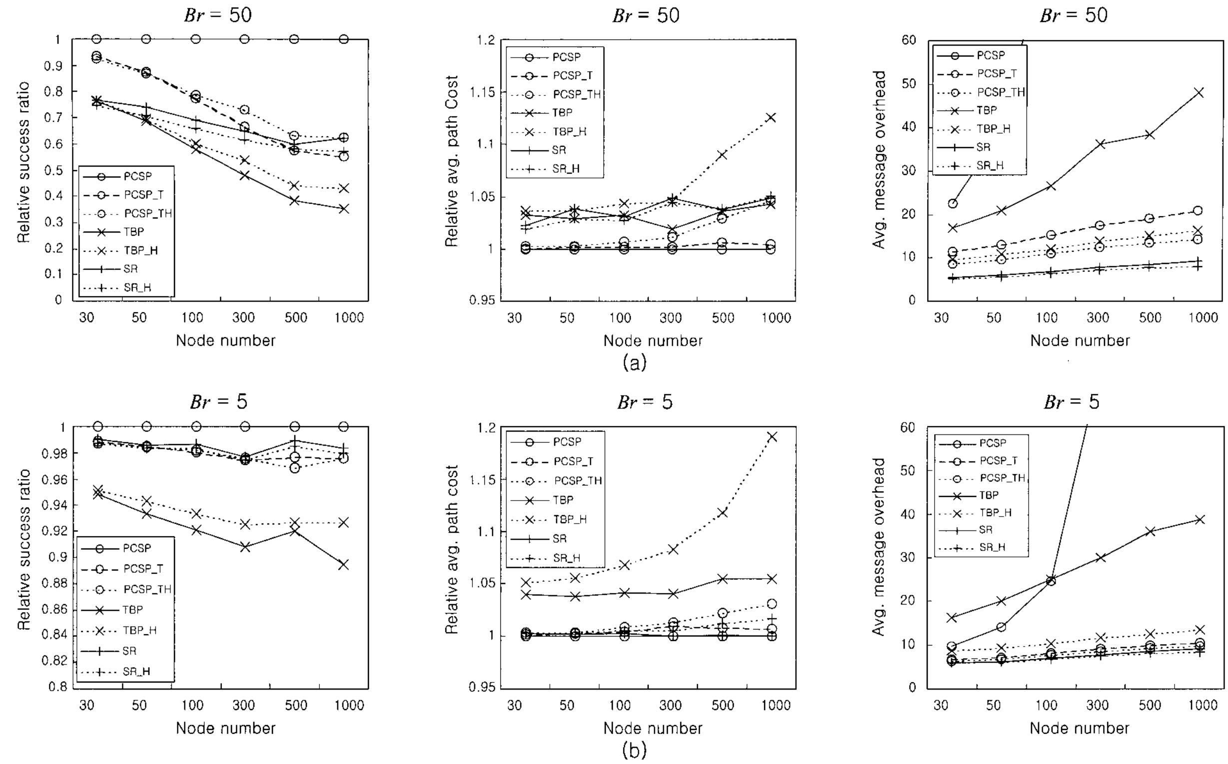


Fig. 11. Simulation results for the bandwidth-constrained least-cost routing (50% imprecision ratio): (a) Tight requirement case (50 Mbps), (b) loose requirement case (5 Mbps).

imprecise—message overhead significantly.[17]

VIII. CONCLUSION

In this paper, we have presented a new distributed QoS routing scheme, named PCSP, that can enhance the performance of the existing schemes significantly by applying qualitative selective approach. It is designed to provide an exact solution with moderate overhead to the constrained optimization problem, taking into account the practical environment where the state information available for the routing decision is not exact.

The PCSP algorithm does not limit the number of probe messages, instead, employs a qualitative, or conditional, selective probing approach. We have extended the pre-computed information to include both the cost and the QoS metrics of the LC and BQ paths. Using this extended pre-computed information and the information collected by the probe messages, the PCSP scheme can calculate the end-to-end cost and the end-to-end QoS metrics of the candidate paths. So, the PCSP scheme is able to find the QoS-satisfying LC and BQ paths and acquire their end-to-end cost metrics. It uses them in conditional manner to determine strict probing condition that excludes not only non-feasible paths but also non-optimal paths. This strict probing condition and the carefully designed probing approaches enable to strictly limit the set of neighbor nodes involved in the probing process, thereby reducing the message overhead comparable to or less than the existing distributed schemes without any auxiliary parameters. At the same time, the qualitative nature guarantees the PCSP scheme to maintain ideal success ratio and cost-optimal search.

In practical dynamic network environments, it is essential to consider the impreciseness of state information when implementing QoS routing. For the imprecise state information modelling, we have employed the variation model as it is simple but adequate for practical implementation. In this model, the QoS variation was considered as an additional pre-computed information and two QoS-satisfying conditions were defined carefully in order to maintain the desirable features of the PCSP scheme—that is, low message overhead, ideal success ratio, and cost-optimal search.

However, the PCSP scheme may suffer from high message overhead due to its conservative probing approach when the network size is large and the state information is extremely imprecise. But, the qualitative nature of the PCSP scheme helps to extend the original algorithm by employing additional heuristics. In combination with a quantitative approach such as TBP, the extended PCSP scheme (PCSP_TH) is able to bound the worst-case message overhead below a controllable level, with slight performance degradation in optimal properties of the original algorithm.

According to the simulation results, the PCSP scheme turned out to exhibit ideal success ratio and cost optimal search at the same time, regardless of the imprecise state information. In addition, the quantitative extensions of PCSP scheme, PCSP_T and PCSP_TH, turned out to bound the worst-case message overhead with negligible performance degradation in optimal properties

---

[17] In the third figure of Fig. 6, PCSP_T and PCSP_TH reduce about 50% of the average message overhead of the original PCSP scheme and in the third figure of Fig. 10, they reduce about 85% of the average message overhead of the original scheme.

even for large networks and for highly imprecise state information. In this way, the PCSP scheme, especially in its extended versions, can guarantee a cost-effective search with predetermined message overhead bound and can maintain such performance merits even in practical, imprecise state information environments.

REFERENCES

[1] S. Chen and K. Nahrstedt, "An overview of quality of service routing for the next generation high-speed networks: Problems and solutions," *IEEE Network Mag.*, vol. 12, no. 6, Nov./Dec. 1998.

[2] G. Apostolopoulos, R. Guérin, S. Kamat, A. Orda, and S. K. Tripathi, "Intra-domain QoS routing in IP networks: A feasibility and cost/benefit analysis," *IEEE Network Mag.*, vol. 13, no. 5, Sept./Oct. 1999.

[3] B. Wang and J. C. Hou, "Multicast routing and its QoS extension: Problems, algorithms, and protocols," *IEEE Commun. Mag.*, Jan. 2000.

[4] D. H. Lorenz and A. Orda, "QoS routing in networks with uncertain parameter," *IEEE/ACM Trans. Networking*, vol. 6, no. 6, Dec. 1998.

[5] R. Guérin and A. Orda. "QoS-based routing in networks with inaccurate information: Theory and algorithms," *IEEE/ACM Trans. Networking*, vol. 7, no. 3, June 1999.

[6] Z. Wang and J. Crowcroft, "Quality of service routing for supporting multimedia applications," *IEEE J. Select. Areas Commun.*, Sept. 1996.

[7] Q. Ma and P. Steenkiste, "Quality of service routing with performance guarantees," in *Proc. Int. IFIP Workshop QoS*, May 1997.

[8] Q. Ma and P. Steenkiste, "A multi-class QoS routing algorithm," in *Proc. IEEE INFOCOM'99*, Mar. 1999.

[9] G. Apostolopoulos, R. Guérin, and S. Kamat, "Implementation and performance measurements of QoS routing extensions to OSPF," in *Proc. IEEE INFOCOM'99*, Mar. 1999.

[10] A. Orda and A. Sprintson, "QoS routing: The precomputation perspective," *IEEE/ACM Trans. Networking*, vol. 11, no. 4, Aug. 2003.

[11] F. Ergun, R. Sinha, and L. Zhang, "QoS routing with performance-dependent costs," in *Proc. IEEE INFOCOM 2000*, Mar. 2000.

[12] A. Goel, K. G. Ramakrishnan, D. Kataria, and D. Logithetis, "Efficient computation of delay-sensitive routes from one source to all destinations," *Proc. IEEE INFOCOM 2001*, Apr. 2001.

[13] A. Juttner, B. Szviatovszki, and Z. Rajko, "Lagrange relaxation based method for the QoS routing problem," in *Proc. IEEE INFOCOM 2001*, Apr. 2001

[14] T. Korkmaz and M. Krunz, "Multi-constrained optimal path selection," in *Proc. IEEE INFOCOM 2001*, Apr. 2001.

[15] G. Liu and K. G. Ramakrishnan, "A*Prune: An algorithm for finding K shortest paths subject to multiple constraints," in *Proc. IEEE INFOCOM 2001*, Mar. 2001.

[16] R. Guérin and A. Orda, "Computing shortest paths for any number of hops," *IEEE/ACM Trans. Networking*, vol. 10, no. 1, 2002.

[17] D. S. Reeves and H. F. Salama, "A distributed algorithm for delay-constrained unicast routing," *IEEE/ACM Trans. Networking*, vol. 8, no. 2, 2000

[18] S. Chen and K. Nahrstedt, "Distributed quality of service routing in high-speed networks based on selective probing," *Tech. rep., Univ. IL at Urnana-Champaign*, Dept. Comp. Sci., 1998.

[19] S. Chen and K. Nahrstedt, "Distributed quality of service routing in ad-hoc networks," *IEEE J. Select. Areas Commun.*, vol. 17, no. 8, 1999.

[20] L. Xiao, J. Wang, and K. Nahrstedt, "The enhanced ticket-based routing algorithm," in *Proc. IEEE ICC 2002*, Apr. 2002.

[21] J. L. Sobrinho, "Algebra and algorithms for QoS path computation and hop-by-hop routing in the Internet," *IEEE/ACM Trans. Networking*, vol. 10, no. 4, 2002

[22] D. Raz and Y. Shavitt. "Optimal partition of QoS requirements with discrete cost functions," in *Proc. IEEE INFOCOM 2000*, Mar. 2000.

[23] D. H. Lorenz and A. Orda, "Optimal partition of QoS requirements on unicast paths and multicast trees," *IEEE/ACM Trans. Networking*, vol. 10, no. 1, 2002.

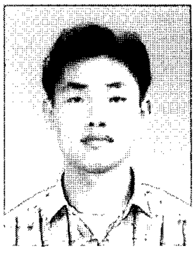
[24] A. Orda and A. Sprintson, "A scalable approach to the partition of QoS requirements in unicat and multicast," in *Proc. IEEE INFOCOM 2002*, June 2002.

[25] I. Cidon, R. Rom, and Y. Shavitt, "Multi-path routing combined with resource reservation," in *Proc. IEEE INFOCOM'97*, Apr. 1997.

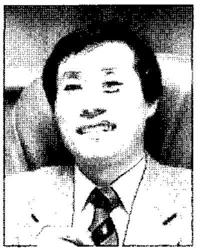
[26] I. Matta and L. Guo, "QDMR: An efficienntt QoS dependent multicast routing algorithm," *J. Commun. Networks*, vol. 2, no. 2, 2000.

[27] S. Chen, K Nahrstedy, and Y. Shabitt, "A QoS-aware multicast routing protocol," *IEEE J. Select. Areas Commun.* vol. 18, no. 12, 2000.

[28] J. Winick, C. Jin, Q. Chen, and S. Jamin, "Inet: Internet topology generator," available at http://www.topology.eecs.umich.edu/inet.



Won-Ick Lee received the B.S. and M.E. degrees in electrical engineering from Seoul National University, Seoul, Korea, in 1998 and 2000, respectively. He spent the summer of 1998 as an intern at Electronics and Telecommunications Research Institute (ETRI). He is currently working toward the Ph.D. degree in electrical engineering and computer science at Seoul National University. His current research interests include QoS, routing algorithm, wireless MAC protocol, wireless sensor networks, and resource allocation in wireless/wired networks.



Byeong Gi Lee received the B.S. and M.E. degrees from Seoul National University, Seoul, Korea, and Kyungpook National University, Daegu, Korea, both in electronics engineering, and the Ph.D. degree in electrical engineering from the University of California, Los Angeles. He was with Electronics Engineering Department of ROK Naval Academy as an instructor and Naval Officer in active service from 1974 to 1979, and worked for Granger Associates, Santa Clara, CA, as a senior engineer responsible for applications of digital signal processing to digital transmission from 1982 to 1984, and for AT&T Bell Laboratories, North Andover, MA, as a member of Technical Staff responsible for optical transmission system development along with related the standards works from 1984 to 1986. He joined the faculty of Seoul National University in 1986 and served as the Director of the Institute of New Media and Communications in 2000 and the Vice Chancellor for Research Affairs from 2000 to 2002.

Dr. Lee was the founding chair of the Joint Conference of Communications and Information (JCCI), the chair of the Steering Committee of the Asia Pacific Conference on Communications (APCC), and the chair of the founding committee of the Accreditation Board for Engineering Education of Korea (ABEEK). He served as the President of Korea Society of Engineering Education (KSEE) and as a Vice President of Korea Institute of Communication Sciences (KICS). He was the editor of the *IEEE Global Communications Newsletter*, an associate editor of the *IEEE Transactions on Circuits and Systems for Video Technology*, and the founding Associate Editor-in-Chief and the EIC of the *Journal of Communications and Networks* (JCN). He served for the IEEE Communications Society as the Director of Membership Programs Development, as the Director of Asia Pacific Region, as the Director of Magazines and as a member-at-Large to the Board of Governors. He currently serves a Vice President of the ABEEK and as the TPC Chair of IEEE International Conference on Communications (ICC) 2005.

Dr. Lee is a co-author of *Broadband Telecommunication Technology*, 1st & 2nd editions, (Artech House, 1993 & 1996), *Scrambling Techniques for Digital Transmission* (New York: Springer Verlag, 1994), *Scrambling Techniques for CDMA Communications* (Norwell, MA: Kluwer Publisher, 2001), and *Integrated Broadband Networks* (Norwood, MA: Artech House, April 2002). He holds seven U.S. patents with four more patents pending. His current fields of interest include broadband networks, wireless networks, communication systems, and signal processing.

He received the 1984 Myril B. Reed Best Paper Award from the Midwest Symposium on Circuits and Systems, Exceptional Contribution Awards from AT&T Bell Laboratories, a Distinguished Achievement Award from Korea Institute of Communication Sciences (KICS), and the 2001 National Academy of Science (of Korea) Award. He is a member of the National Academy of Engineering of Korea, a member of Sigma Xi, and a Fellow of the IEEE.