

Clustering-Based Distributed Precomputation for Quality-of-Service Routing*

Yong Cui and Jianping Wu

Department of Computer Science, Tsinghua University,
Beijing, P.R.China, 100084
cy@csnet1.cs.tsinghua.edu.cn, jianping@cernet.edu.cn

Abstract. As a potential solution to provide quality of service (QoS) for next-generation IP networks, QoS routing (QoSR) seeks to find a multi-constrained path, where the scalability and routing performance are still open problems. We propose a novel Clustering-based Distributed Precomputation algorithm (CDP) for multi-constrained QoSR. After dominating path selection is analyzed to omitting numerous dominated paths, a clustering technique is further presented for dominating path aggregation in routing computation. These two techniques in turn achieve efficient aggregation of the QoS routing table. CDP greatly decreases the computational complexity on a single node by utilizing the distributed computation on each node in the network. Simulation results confirm that CDP not only has low computational complexity, but also achieves high routing performance with good scalability on both QoS parameters and the network scale.

1 Introduction

The next-generation Internet based on IP networks is expected to support applications with diverse quality-of-service (QoS) requirements [1][2]. As an important method to provide QoS, QoS routing (QoSR) seeks to find a feasible path satisfying multiple constraints for each QoS application, and thus performs QoS control on the level of path selection from numerous paths in the network [3][4][5][6][7][8]. Since the bandwidth is heavily limited and transmission delay is relatively large in wireless networks, QoSR becomes an important potential solution for QoS control in the next-generation wireless networks.

This paper proposes a novel solution to the general QoSR problem with diverse QoS parameters: Clustering-based Distributed Precomputation (CDP). In CDP, each node, maintaining a QoS routing table, broadcasts its routing table to all of its neighbors, while receiving the routing information sent by its neighbors. In order to reduce the QoS routing table, we introduce dominating path selection and propose the

* Supported by: (1) the National Natural Science Foundation of China (No. 60403035); (2) the National Major Basic Research Program of China (No. 2003CB314801).

clustering-based aggregation of dominating paths, which achieve high routing performance for CDP with low computational complexity and good scalability.

The rest of this paper is organized as follows. The problem formulation is given in Section II. We analyze the dominating paths and propose the algorithm in Section III. In Section IV, CDP is evaluated by extensive simulations. Finally, conclusions appear in Section V.

2 Problem Formulation

A directed graph $G(V, E)$ presents a network. V is the node set and the element $v \in V$ is called a node representing a router in the network. E is the set of edges representing links that connect the routers. The element $e_{ij} \in E$ represents the edge $e = v_i \rightarrow v_j$ in G . In QoS, each link has a group of independent weights $(w_0(e), w_1(e), \Lambda, w_{k-1}(e))$, which is also called QoS weight $w(e)$, where $w_l(e) \in \mathfrak{R}^+$ for $l = 0, 1, \Lambda, k-1$. QoS weights can be divided into three classes: additive (e.g. cost, delay), multiplicative (e.g. loss rate) and concave (e.g. available bandwidth) [3].

In this paper $w_l(e)$ ($l = 0, 1, \Lambda, k-1$) can be any kind of QoS parameters. Since multiplicative parameters can be transformed into additive constraints, we only consider additive and concave constraints. Accordingly, for a path $p = v_0 \rightarrow v_1 \rightarrow \Lambda \rightarrow v_j$, $w_l(e) \in \mathfrak{R}^+$ and $0 \leq l \leq k-1$, the path weight $w_l(p) = \sum_{i=1}^j w_l(v_{i-1} \rightarrow v_i)$ if $w_l(e)$ satisfies the additive characteristic, or $w_l(p) = \prod_{i=1}^j w_l(v_{i-1} \rightarrow v_i)$ if $w_l(e)$ is multiplicative, or $w_l(p) = \max_{i=1}^j w_l(v_{i-1} \rightarrow v_i)$ if $w_l(e)$ is concave.

Definition 1. Feasible path

For a given graph $G(V, E)$, source node s , destination node t , $k \geq 2$ and a constraint vector $c = (c_0, c_1, \Lambda, c_{k-1})$, the path p from s to t is called a feasible path, if $w_l(p) \leq c_l$ for any $l = 0, 1, \Lambda, k-1$ (if $w_l(p)$ represents the available bandwidth of path p , it should be $w_l(p) \geq c_l$). We write $w(p) \leq c$ in brief. \square

Note: $w(e)$ and c are both k -dimensional vectors. For a given QoS request with its constraint c , QoS seeks to find a feasible path p satisfying $w(p) \leq c$ based on the network state information.

In addition to the traditional destination and the next hop, QoS routing tables need to maintain the QoS weight $w(p)$ of each path. When a QoS flow (packet) arrives at a node, the node only seeks to find a feasible path in the table and forwards the flow (packet) to the next hop accordingly.

Definition 2. Extended distance vector

For a given path p from source s to destination t , $(s, t, w(p))$ is called the extended distance vector of path p . \square

Each node in the network converts the items in the routing table it maintains into extended distance vectors, and then sends them to its neighbors. Based on these vectors received by each node, a node computes its routing table with CDP.

3 Dominating Path Aggregation

Since there are different paths between two nodes in an ordinary network, a lot of routes may exist for a certain destination in the QoS routing table. Multi-constrained QoSR is an NPC problem [9] [5], so the number of routes may increase exponentially with the network scale. In order to increase the scalability of the QoSR, it is necessary to restrict the number of routes to a certain destination. Some recent research results show that a few distinctive routes can represent the numerous paths and satisfy most QoS requests [2] [10].

3.1 Dominating Path

Path set $P_{\langle s,t \rangle}$ denotes a set of paths with the same pair of source and destination $\langle s,t \rangle$, i.e. $p \in P_{\langle s,t \rangle}$ is a path from s to t .

Definition 3. *Dominating path*

For a given non-empty path set $P_{\langle s,t \rangle}$ and $p \in P_{\langle s,t \rangle}$, if there is no path $p' \in P_{\langle s,t \rangle}$ satisfying $w(p') < w(p)$, path p is called a dominating path on $P_{\langle s,t \rangle}$. \square

The “dominating” relationship between the paths in path set $P_{\langle s,t \rangle}$ is a partial order, resulting in that $P_{\langle s,t \rangle}$ is a partially ordered set. Therefore, multiple minimum elements (dominating paths) may exist in $P_{\langle s,t \rangle}$.

Theorem 1. Routing performance will not be decreased by omitting non-dominating paths from non-empty path set $P_{\langle s,t \rangle}$.

Proof: For a non-dominating path $p \in P_{\langle s,t \rangle}$, there must be a dominating path $p' \in P_{\langle s,t \rangle}$ with $w(p') < w(p)$. For any QoS request from s to t with constraint c , if p is a feasible path, i.e. $w(p) \leq c$, then $w(p') < c$, i.e. p' is also a feasible path for c . After the omission of p from $P_{\langle s,t \rangle}$, $P_{\langle s,t \rangle}$ still has the element $p' \in P_{\langle s,t \rangle}$ that satisfies the request. Thus, the omission does not decrease the routing performance. \square

Therefore, each node in the network may omit a lot of non-dominating paths from $P_{\langle s,t \rangle}$ in the distributed routing process, and only maintains dominating paths for routing computation and communication.

Definition 4. *Dominating path set*

For a given non-empty path set $P_{\langle s,t \rangle}$, if $\forall p \in P_{\langle s,t \rangle}$ is a dominating path on $P_{\langle s,t \rangle}$, $P_{\langle s,t \rangle}$ is called a dominating path set. \square

Definition 5. *Maximum dominating path set*

For a given non-empty path set $P_{\langle s,t \rangle}$ and a dominating path set $A \subset P_{\langle s,t \rangle}$, if $\forall p \in P_{\langle s,t \rangle} - A, \exists p' \in A$ that $w(p') < w(p)$. A is called the maximum dominating path set on $P_{\langle s,t \rangle}$, where A is denoted by $D_{\langle s,t \rangle}$. \square

Since the maximum dominating path set $D_{\langle s,t \rangle}$ is independent to the method by which $D_{\langle s,t \rangle}$ is computed or selected from $P_{\langle s,t \rangle}$, we omit the detailed method to calculate $D_{\langle s,t \rangle}$ in the paper.

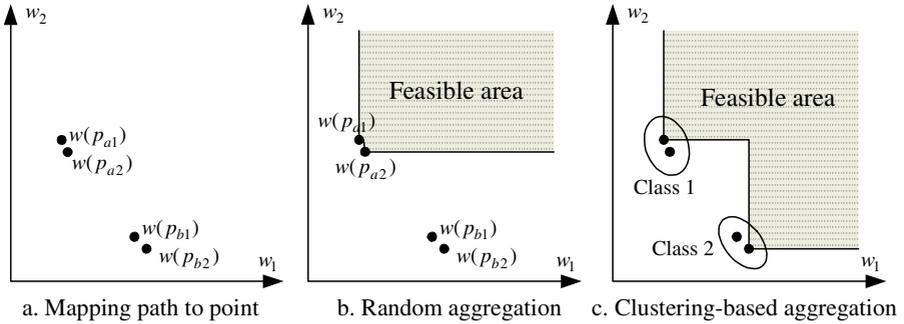


Fig. 1. Aggregation of dominating paths ($R=2$)

3.2 Clustering-Based Dominating Path Aggregation

Ordinarily speaking, the selection of dominating path can eliminate a lot of paths to improve the scalability without reducing the routing performance. However, in some large-scale networks, dominating paths for a certain $\langle s,t \rangle$ may still be numerous, or even be exponential to the network scale [11]. In order to improve the scalability, it needs to restrict the maximum number of dominating paths for a certain $\langle s,t \rangle$.

Definition 6. *Maximum reduplication R*

The maximum number of dominating paths for a given $\langle s,t \rangle$ is called the maximum reduplication, denoted as R . □

According to the maximum reduplication R , each node needs to guarantee that the routes to a certain destination should be no more than R . We then analyze how to select R representative paths to satisfy QoS requests as much as possible in the QoS weight space. In a network with k QoS weights, the weight $w(p)$ of path p can be regarded as a point (or vector) in the k -dimensional Euclidean Space. Thus, the maximum dominating path set $D_{\langle s,t \rangle}$ represents a point set in the k -dimensional Euclidean space.

As an example shown in Fig. 1.a, the set $D_{\langle s,t \rangle}$ includes 4 dominating paths in the QoS weight space with $k=2$. We choose $R=2$, i.e. we need to select 2 paths into the QoS routing table from the 4 paths to improve the scalability. One possible method is random selection, where path p_{a1} and p_{a2} may be selected as shown in Fig. 1.b. For any QoS request with constraint c , if a selected path p satisfies $w(p) \leq c$, p can be taken as the feasible path for the request. Therefore, the shadowed area in the figure represents the feasible area for requests. In order to select a more representative path from $D_{\langle s,t \rangle}$ to enlarge the feasible area, another possible method is to aggregate dominating paths to R classes by clustering, and then select one path from each class. Fig. 1.c shows a possible result of the clustering-based aggregation, where p_{a1} and p_{a2} aggregate to class 1 while p_{b1} and p_{b2} compose class 2. Thus, a path is then selected from each class to

construct the feasible area. The clustering-based aggregation is generally easier to satisfy a QoS request than the random process.

```

Aggregation_Program (D<s,t>, R, T)
1) times=0 /* iteration times */
2) AR(t)=? /* aggregated paths */
3) Select R paths to PR randomly
4) D<selected> = w(p) of R paths in PR
5) Label R points in D<selected>
6) DO
7)   FOR EACH path p in D<s,t>
8)     Find nearest point q in D<selected>
9)     Label p as q's label
10)  FOR EACH label
11)   IF path exists for label in D<s,t>
12)     q = average w(p) of p in D<s,t> with label
13)     Replace original point by q in D<selected>
14)   ELSE /* i.e. path does NOT exist for label */
15)     find the w(p) farthest to the point with label
16)     Replace original point by w(p) in D<selected>
17)  times=times+1
18) WHILE times<T AND D<selected> changed
19) FOR EACH point in D<selected>
20)  Add nearest path p in D<s,t> to AR(t)
21) RETURN AR(t)

```

The above Aggregation_Program shows the pseudo-code for clustering-based aggregation of dominating paths. The input of the algorithm includes the dominating path set $D_{\langle s,t \rangle}$ computed by node s with the destination t , the maximum reduplication R and the maximum number of iterations T for clustering. P_R is the initial path set with R paths randomly selected from $D_{\langle s,t \rangle}$. $D_{\langle \text{selected} \rangle}$ is a set of R points in the Euclidean Space rather than a set of paths, where each point represents a clustered class. The output of the algorithm $A_R(t)$ is R paths to destination t after clustering, i.e. the routes from s to t .

The algorithm initializes the maximum iteration number, the path set A_R , the dependency of routing table to dominating paths, $D_{\langle \text{selected} \rangle}$ as the random R paths from $D_{\langle s,t \rangle}$ and the label of each point in $D_{\langle \text{selected} \rangle}$ where $D_{\langle \text{selected} \rangle}$ is a point set in the Euclidean Space rather than a path set. (Line 1-5). It then performs the iteration of labeling procedure, which includes the following 3 steps: (1) Labeling each path in $D_{\langle s,t \rangle}$ (Line 7-9): for each path in $D_{\langle s,t \rangle}$, find in $D_{\langle \text{selected} \rangle}$ the point nearest in Euclidean distance to the path and label the path as the label of the point. (2) Generating the new point set $D_{\langle \text{selected} \rangle}$ (Line 10-16): for each label, compute the average QoS weight of all of the paths with the label, and replace the original point with the label in $D_{\langle \text{selected} \rangle}$ by this average weight (Line 11-13). If no paths exist for a certain label, the point is then replace by the weight of the farthest path to the original point with the label (Line 14-16). (3) The labeling procedure iterates until the maximum iteration number reaches or $D_{\langle \text{selected} \rangle}$ is not changed in the iteration (Line 17-18). After the above iteration procedure, for each point in $D_{\langle \text{selected} \rangle}$, the algorithm finds the path in $D_{\langle s,t \rangle}$ nearest in Euclidean distance to the point. These paths compose the routes from s to t .

4 Performance Evaluation

Simulation results in this section show the performance evaluations of the following different algorithms. (1) CDP (i.e. the proposal in the paper). (2) CDP without clustering-based aggregation, i.e. to randomly select R paths from the dominating path set as the routes. This algorithm can be taken as the distributed extension to the path-limited heuristic proposed by Yuan, where Yuan’s heuristic is a centralized algorithm [11], this algorithm is denoted as D-Yuan. The comparison between CDP and D-Yuan will show the efficiency of clustering-based aggregation. (3) CDP without both clustering aggregation and dominating path selection, i.e. to randomly select R route from the extended distance vectors maintained by a node, denoted as DPRnd. This comparison between D-Yuan and DPRnd will show the efficiency of dominating path selection. (4) MEFPA, a well performed centralized precomputation algorithm in literatures [2]. We set the number of weight divisions $b=5$ and $R \leq C_{b+k-2}^{k-1}$ in MEFPA, and keep the number of routes consistent in the four algorithms in each simulation.

We propose the spindle network topology, as shown in Fig. 2, for the comparison of different routing algorithms. Src is the source node, where the final routing tables are compared in different algorithms. In the network, a single link exists between Src and each NB node, while multiple duplicate links exists between each NB and Dst. The number of links between an NB and Dst is denoted $DupNum$, and $DupNum$ links represent that there are numerous different paths from NB to Dst.

In the simulations, we set (1) degree=10 for Src, i.e. 10 neighbors, (2) $DupNum = 40, 200, 1000, 5000$, respectively, (3) QoS weight number $k = 2, 3, 4, 5$, respectively, (4) $w_l(e) \sim \text{uniform}[1,1000]$ for $l = 0,1,\Lambda, k - 1$, where $w_l(e)$ have no correlation for different e or l . (5) QoS request from Src to Dst with the constraint c . Since the average weight of path p from Src to Dst is $w_l(p) = 1000$ for $l = 0,1,\Lambda, k - 1$, we select c_l obeying Gaussian probability distribution with mean $\mu = 1000$ and deviation $\sigma = 1000/3$. The routing performance is evaluated by the routing success ratio [2][6][7]. Computational complexity is then evaluated by running time of algorithms.

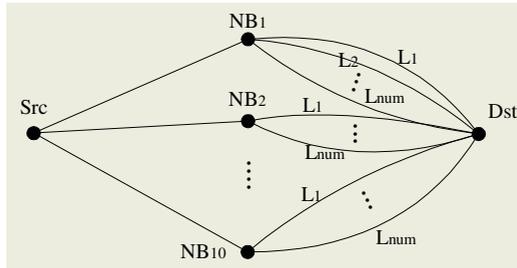


Fig. 2. Spindle network topology

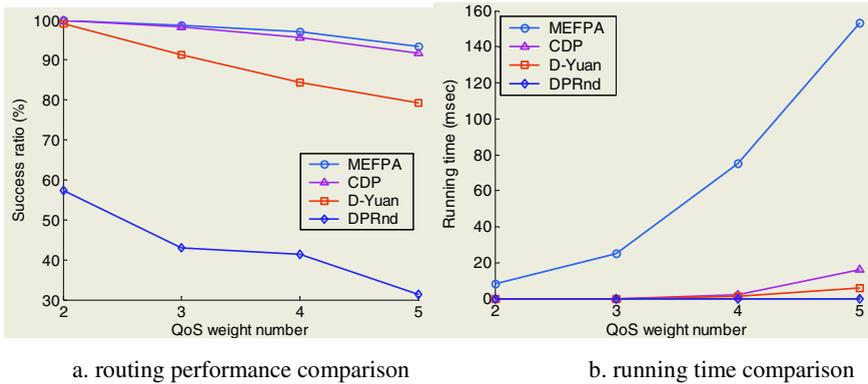


Fig. 3. Performance evaluation of four algorithms with different weight number

Fig.3 show the performance of these four algorithms with $DupNum=1000$ for large-scale networks and weight number $k=2,3,4,5$. The success ratio in Fig. 3a shows $DPRnd < D\text{-Yuan} < CDP \approx MEFPA$ in routing performance, which confirms that both dominating path selection (D-Yuan) and clustering-based aggregation (CDP) improve the routing performance effectively. Considering the QoS weight number, DPRnd and D-Yuan degrade their performance heavily with increasing k , while MEFPA and CDP sustain their performance. Although centralized MEFPA achieves a high performance by making overall plans and taking all factors into consideration on Src, the routing performance of distributed CDP is still comparable with that of MEFPA.

From the running time comparison shown in Fig. 3b, the distributed algorithms, DPRnd, D-Yuan and CDP, greatly excel the centralized MEFPA, especially in large-scale networks. In fact, the computation of centralized MEFPA is based on the whole network state information, while the input information is only $O(dR)$ for a given destination in distributed algorithms, e.g. CDP. In a real large network with complex topology, MEFPA may need more computational resources to create a shortest path tree, while CDP is independent of the network topology. Although MEFPA compute the whole routing table every time, the frequent changes of network state overload each node in the large-scale QoS network. If a long hold timer is used to reduce the computation frequency, a large staleness may lower the routing performance [12]. Therefore, considering both routing performance and running time shown in Fig.3, the advantages are apparent to deploy CDP rather than MEFPA in real IP networks.

5 Conclusion

QoSR algorithms in literature often have some limitations including connection orientation, centralized computation and limited types of QoS weights. In this paper, we introduced the extended distance vectors to support diverse QoS parameters and proposed a clustering-based distributed precomputation QoSR algorithm, namely CDP. We analyzed selection of dominating paths in the QoS weight space and gave the algorithm for clustering-based dominating path aggregation. Since we proved that

omitting none-dominating paths does not decrease the routing performance, the dominating path selection performs a lossless reduction of routes. Clustering-based dominating paths aggregation further reduces the number of routes with representative paths. Hence, both the dominating path selection and the clustering-based aggregation improve the routing performance and scalability effectively. Moreover, CDP adopts distributed computation, which radically reduces the computational complexity. Simulation results further confirm that CDP performs well in terms of both routing performance and scalability.

References

1. X. Xiao and L. M. Ni, Internet QoS: A big picture, *IEEE Network*, vol. 13, no. 2, pp. 8-18, March-April 1999.
2. Y. Cui, K. Xu, J.P. Wu. Precomputation for Multi-constrained QoS Routing in High-speed Networks. In *Proceedings of the IEEE INFOCOM 2003*, San Francisco, CA, April 2003.
3. Y. Cui, J. P. Wu, K. Xu, et al. Research on internetwork QoS routing algorithms: a survey. *Chinese Journal of Software*, vol. 13, no. 11, 2065-2076, 2002.
4. S. Chen and K. Nahrstedt, An overview of quality-of-service routing for next-generation high-speed networks: Problems and solutions, *IEEE Network*, vol. 12, no. 6, pp. 64-79, Nov. 1998.
5. Z. Wang and J. Crowcroft. Quality-of-service routing for supporting multimedia applications. *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 7, pp. 1228-1234, Sep. 1996.
6. Yong Cui, Ke Xu, Jianping Wu, and et al. Multi-constrained routing based on simulated annealing. *IEEE ICC 2003*.
7. T. Korkmaz, M. Krunz, Multi-Constrained Optimal Path Selection. *IEEE INFOCOM'01*, vol.2, pp. 834 -843, 2001.
8. L. Xiao, K. S. Lui, J. Wang and K. Nahrstedt. QoS Extension to BGP. *IEEE ICNP*, 2003.
9. M. S. Garey, D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, New York, 1979.
10. A. Orda and A. Sprintson, QoS routing: the precomputation perspective. *IEEE INFOCOM'00*, vol. 1, pp. 128-136, 2000.
11. X. Yuan, X. Liu, Heuristic Algorithms for Multi-Constrained Quality of Service Routing, *IEEE INFOCOM'01*, vol.2, pp. 844 -853, 2001.
12. A. Shaikh, J. Rexford, K. G. Shin, Evaluating the impact of stale link state on quality-of-service routing. *IEEE/ACM Transactions on Networking*, vol. 9, no. 2, Apr. 2001