SCIENCE CHINA
Information Sciences

• **RESEARCH PAPERS** •

October 2010 Vol. 53 No. 10: 1947–1962
doi: 10.1007/s11432-010-4066-x

Supporting multiple metrics in QoS-aware BGP

CUI Yong[1]*, ZHAO YouJian[1], KORKMAZ Turgay[2] & ZHANG TieLei[2]

[1]*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084, *China;*
[2]*Department of Computer Science, University of Texas at San Antonio, USA*

Received January 25, 2008; accepted August 20, 2009

Abstract Being an important and yet a challenging problem, the QoS-based routing in the converging Internet has received significant attention from the research community. However, most of the QoS-based routing research is conducted in the context of intra-domain routing, leaving QoS-based inter-domain routing relatively open. In this paper, we focus on QoS-based inter-domain routing and specifically investigate how to enhance the current inter-domain routing protocol (BGP) with QoS extensions. To support multiple QoS metrics, BGP speakers have to advertise multiple routes for each destination. However, this will increase the routing message overhead and make QoS-aware BGP unscalable. To provide scalability without significantly sacrificing the performance, we propose path reduction algorithms called contribution based reduction (CBR) algorithms. Extensive simulations show that the proposed schemes achieve high performance in finding feasible paths with low complexity in terms of message overhead and computation, making the QoS extension to BGP scalable.

Keywords BGP, QoS, routing, multiple constraints

Citation Cui Y, Zhao Y J, Korkmaz T, et al. Supporting multiple metrics in QoS-aware BGP. Sci China Inf Sci, 2010, 53: 1947–1962, doi: 10.1007/s11432-010-4066-x

1 Introduction

The continuous growth in both commercial and public network traffic with quality-of-service (QoS) requirements is calling for the evolution of the Internet from a simple best-effort network towards a converged one that can provide various levels of QoS guarantees and/or differentiations to voice, video, and data applications. To achieve this long sought goal, researchers have been considering a wide spectrum of approaches ranging from highly revolutionary ones (e.g., ATM) to laissez-fair ones (e.g., no major changes, use over-provisioning). Since both ends of the spectrum have their serious disadvantages in practice, it is deemed necessary to develop evolutionary solutions that can incrementally be deployed in the Internet.

The Internet is a collection of several autonomous systems (ASs) that are owned and operated by different organizations [1, 2]. This naturally leads to the division of the Internet routing (be it QoS-based or not) into two different, yet complementary parts, namely intra-domain routing within a single AS, and inter-domain routing between ASs. Actually, this partitioning is necessary to adequately address varying design objectives. For example, while intra-domain routing focuses more on the efficient use of network resources, inter-domain routing is more concerned with scalability. Accordingly, today's intra-domain and inter-domain routing protocols (e.g., OSPF [3] and BGP [4]) use two different approaches,

*Corresponding author (email: cy@csnetl.cs.tsinghua.edu.cn) namely link-state routing and path/distance-vector routing, respectively. However, the current versions of these protocols have adopted relatively simple solutions without taking any knowledge about the QoS requirements of applications or the dynamics of network state information into account. Supporting QoS connections requires the existence of a routing mechanism that computes the QoS paths [5]. To facilitate the aforementioned evolution of the Internet, it is necessary to improve both intra-domain and inter-domain routing protocols with QoS extensions [6, 7].

Traditional approach to QoS-based routing is to first acquire the dynamic network state information using link-state routing and then to compute QoS-based paths using this information along with given QoS requirements. Naturally, this approach is a good fit in the context of intra-domain routing, and widely studied in the literature and resulted in QoS-extensions to OSPF [8, 9]. However, this approach cannot cope with the scalability and policy issues in the context of inter-domain routing. A natural choice seems to extend BGP with QoS metrics. However, the studies in this direction are still in their infancies, calling for further research. In [10] the authors propose a new BGP attribute, named the QOS_NLRI, to encode QoS metrics associated with the underlying path. However, this work does not consider how to determine or use such information. In [11] the authors concentrate on extending BGP with one metric, namely bandwidth. However, more than 2 QoS metrics are common in the QoS context and need to be considered. In this paper, we focus on QoS-based inter-domain routing and particularly consider extending BGP with multiple QoS metrics.

In essence, BGP uses path/distance-vector approach and disseminates an AS-level path to convey just the reachability information for each destination network. However, for QoS extension, BGP needs to include the QoS metrics associated with the advertised AS-level path. Note that these QoS metrics can be either direct metrics (such as the available bandwidth, delay, jitter) or well-defined metrics (such as the ABI in [11]). Inside an AS, we can assume that each border router pre-computes the QoS metrics of several paths from that border router to every destination network within that AS. If the intra-domain protocol is not QoS-aware, the border routers could use measurements to obtain QoS metrics of the paths within their domain. Border routers then advertise QoS metrics and other reachability information to their neighbors. Receiving border routers can concatenate their own paths with the received ones and create several new paths. After enforcing some predefined path selection policies to reject some routes, BGP advertises remaining AS-level paths and the QoS metrics associated with these to their inside and outside peers using I-BGP and E-BGP. Since the number of QoS metrics is more than 2, there will be several routes to one single destination. Advertising all these paths and the associated QoS metrics would cause significant message overhead. For the scalability reasons, it is necessary to reduce this routing message overhead while maximizing the success ratio in identifying paths that meet the given QoS requirements.

In response to this, we propose multi-constrained "path reduction" algorithms called contribution based reduction (CBR) algorithms. In essence, the proposed CBR algorithms try to minimize the number of QoS-based routes advertised by BGP while maximizing the feasibility region covered by the advertised paths. Specifically, the proposed algorithms select L multi-constrained paths from N ones ($L < N$) such that the selected L paths are able to maximize the probability of satisfying various QoS routing requests that would be supported if all the underlying paths were advertised.

When developing and evaluating the proposed CBR algorithms, we consider an Optimal algorithm that tries all the possible combinations of L out of N paths and selects the best combination that maximizes the feasibility region. Therefore, we first present the Optimal algorithm. The key disadvantage of this algorithm is its high computational complexity. To reduce this complexity while providing the comparable performance to the Optimal algorithm, we propose two algorithms: 1) the incremental contribution algorithm (denoted by ContriInc); and 2) the improved incremental contribution algorithm (denoted by ImprovedInc). In essence, both algorithms try to select the least number of paths that make the greatest contribution in terms of covering the underlying QoS feasibility region of all paths. Using simulations, we show that our algorithms provide good performance compared to the Optimal while significantly reducing the message overhead and computational complexity.

The rest of this paper is organized as follows. We present related work in section 2. In section 3, we discuss a general framework that can be used to extend BGP with QoS. In section 4, we formulate the path reduction problem and describe an upper-bounded optimal algorithm, the proposed CBR algorithms, and the related concepts. In section 5, we evaluate our schemes using extensive simulations. Finally, we conclude this paper in section 6.

2 Related work

With respect to the QoS extension to BGP, some related work has been done. Bonaventure [12] proposes a flexible QoS attribute that can be used to distribute QoS information with BGP. In [10] the authors propose a new BGP attribute, named the QOS_NLRI, to encode QoS metrics associated with the underlying path. In [13], a set of BGP attributes, i.e., Traffic Engineering Weights, are defined to allow an ISP to indicate its traffic engineering preferences when announcing routes. However, none of these works consider how to determine or use such information. Fei and Gerla [14] extend MBGP (multiprotocol extension to BGP4) for inter-domain QoS Multicast. However, this work does not consider the problem of routing message overhead and the scalability. Based on statistical property in QoS routing, the authors [11] concentrate on extending BGP with one metric, namely bandwidth. However, single metric does not contain sufficient information to assess whether user QoS requirements can be met or not [15]. Therefore, more than 2 QoS metrics are common in the QoS context and need to be considered.

On the other hand, in the area of intra-domain routing, QoS-extensions to OSPF [8, 9] have been well studied. Furthermore, many multi-constrained QoS routing algorithms based on intra-domain routing are proposed to find a feasible path that satisfies multiple constraints. For 2-constrained problems, Jaffe [16] proposes a distributed pseudo polynomial algorithm with a time complexity of $O(n^5 b \log nb)$, where n is the number of nodes and b is the maximum weight of a link. Jaffe also proposes a polynomial time heuristic based on the linear function $g(p) = a_1 w_1(p) + a_2 w_2(p)$. In order to solve the delay-constraint least cost (DCLC) problem, some approximate algorithms are proposed in [17, 18]. For an arbitrary $\varepsilon > 0$, these algorithms can find a path in polynomial time. Not only is the delay constraint satisfied in the path, but also its cost is less than $(1 + \varepsilon)$ times the optimal cost. In addition, based on some specific scheduling schemes, dependencies among QoS parameters exist. Then the original NP-complete multi-constrained problem can be reduced to the standard shortest path problem [19, 20]. As for the general multi-constrained problem, the tunable accuracy multiple constrains routing algorithm (TAMCRA) [21] and its reformation the self adaptive multiple constraints routing algorithm (SAMCRA) [22] try to minimize the nonlinear function $g_\lambda(p) = \sum_{l=1}^{k} (w_l(p)/c_l)^\lambda$. Particularly, the former is designed for the PNNI protocol and the latter for IP networks. Aiming to solve the multi-constrained optimal path problem, Korkmaz [23] proposes an online heuristic approach called H_MCOP, which optimizes both the nonlinear function $g_\lambda(p)$ (for feasibility) and the primary function $c(p)$ (for optimality). Recently, off-line/precomputation-based methods have been proposed as an instrument to facilitate scalability, improve response time and reduce computation load on network elements [24]. For the off-line/precomputation solutions, Yuan [25] proposes a limited granularity heuristic and a limited path heuristic for solving the multi-constrained path (MCP) problem, using the extended Bellman-Ford algorithm. Additionally, the multi-constrained energy function-based precomputation algorithm (MEFPA) by Cui [26] is specially designed for routing precomputation. By precomputation, this algorithm figures out a QoS routing table that contains multiple feasible paths for every possible multi-constrained QoS requests. However, without inter-domain QoS routing support, all these intra-domain based schemes/algorithms cannot be deployed in a large scale, since the routing message overhead and path computation complexity are quite high when the network size becomes large.

In this paper, we propose a new scalable extension to BGP with multiple metrics and give the big picture that well integrates the inter-domain and intra-domain routing together.

3 Scalable QoS extension framework

In this section, we describe a general framework to introduce multiple QoS metrics into BGP. Consider

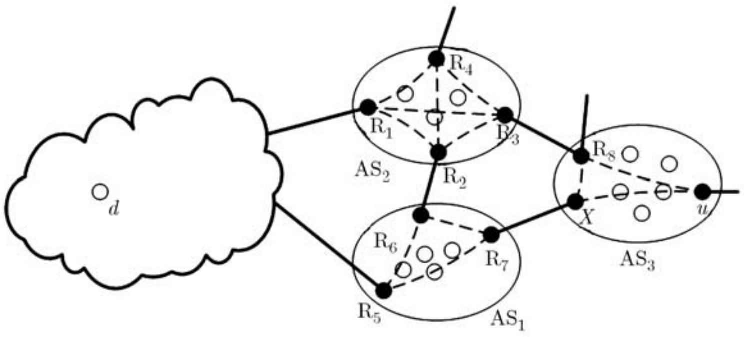

Figure 1 BGP scenario.

Figure 1 for a typical network architecture at BGP level. The current Internet consists of ASs in which the routers are classified as the border routers (the black dots in Figure 1) and the internal routers (the white dots in Figure 1). Currently, no metric is associated with the links. Only the reachability information is exchanged between ASs. In the conventional BGP, the advertisements propagated between BGP routers are formatted in the UPDATE message, which mainly contains the destination addresses, the list of ASs traversed by the UPDATE message (AS_PATH), and the next hop address (NEXT_HOP).

When extended with multiple metrics, we envision that each link (say e) will be associated with K independent general weights, $w_1(e), w_2(e), \ldots, w_K(e)$, where $w_l(e) \in \mathbb{R}^+$ $(1 \leqslant l \leqslant K)$. Accordingly, each path (say p) will also be associated with K weights, of which the lth $(1 \leqslant l \leqslant K)$ is denoted as $w_l(p)$. Each weight represents a practical QoS metric, either a direct metric (such as the available bandwidth, delay, cost, etc.) or a well-defined metric (such as the ABI [11]). To inform other nodes about the multiple QoS metrics associate with a path, QoS-extended BGP must include multiple (say K) QoS metrics into the UPDATE message. Then these QoS metrics are advertised along with the path information in both intra-AS and inter-AS manners using I-BGP and E-BGP with QoS extensions. During these advertisements, every router figures out its paths leading to every destination. Although we focus on BGP extension only, BGP will definitely cooperate with some interior gateway protocol (IGP) used inside an AS. For integrity, we describe the big picture in three aspects: IGP, E-BGP and I-BGP.

In each AS, we run an IGP which may or may not be QoS-aware. In a QoS-aware AS, due to multiple metrics, there may be multiple good paths from a source node to one single destination node. Consider two paths: p_1 with 1 ms delay and 0.1% loss rate, p_2 with 0.5 ms delay and 0.5% loss rate. p_2 is better than p_1 in terms of delay, but p_1 is better than p_2 in terms of loss rate. We cannot simply tell which path is better, so both are good paths. Therefore, if the IGP inside an AS is QoS-aware, we allow each node to precompute multiple internal paths to every node within that AS. In order to reduce the routing message overhead, we must restrict the number of the internal paths to every node to an acceptable level, say L $(L \geqslant 1)$. Then IGP communicates the information about the L paths and their related QoS metrics to BGP. So each border router will know L paths to each internal node within that AS. Likewise, each internal node will know L paths to each border router within that AS, as well. If the IGP is not QoS-aware, we could use measurements to obtain QoS metrics about the paths between every two nodes within the same AS. For example, we can use CapProbe [27] or Pathrate [28] to obtain the bandwidth capacity, and Ping for delay or delay jitter. Then each border router knows only one path to each internal node within that AS and each internal node knows only one path to each border router.

Among different ASs, the external BGP (E-BGP) is used to exchange information about paths and related QoS metrics. Border routers (using E-BGP with our QoS extension) within an AS will advertise L precomputed paths for each destination to their neighbors that are the border routers in other ASs. Note that a particular case is that, if an AS is not QoS-aware, border routers within that AS will advertise only one path for each destination node which is within the same AS to their peers in different ASs.

The internal BGP (I-BGP) runs among routers within the same AS. Using I-BGP with our QoS extension, a border router (say X) will send the externally-learned paths to internal nodes (e.g., X may receive L paths for a destination (d) from another AS and sends them to internal nodes). Each node (say

u) knows L paths from itself to that border router (X) and now it learned L paths from X to a given destination (d). So each node (u) can concatenate L ($u - X$) and L ($X - d$) paths and have $L * L = L^2$ ($u - d$) paths. If a node receives paths from other border routers as well, then a node will have at most $B * L^2$ paths, where B is the number of border routers within the same AS. Note such a particular case, if that AS is not QoS-aware, each internal node (u) still obtains only one path to each destination (d) after this routing message exchange. Anyway, the number of obtained paths in border routers is large. We denote this number as M ($1 \leqslant M \leqslant BL^2$). After a node obtains M paths, it will first enforce some predefined path selection policies to reject some paths and thus reduce M to N ($1 \leqslant N \leqslant M$). These policies may be the commercial relationships between ASs, the number of hops in terms of AS, the multiple exit discriminator, etc. [11]. If all the N paths are propagated to other ASs, the increased routing message overhead may affect the scalability of BGP. To reduce the number of routes advertised by BGP routers, we propose two path reduction algorithms in section 4.

Our proposed path reduction algorithms are executed to further reduce the number of paths to L ($1 \leqslant L \leqslant N$). At the same time, our algorithms also consider the routing success ratio. Among multiple paths, our algorithms select L of them that maximize the routing success ratio. Afterwards, each node installs the L selected paths into the local forwarding table, and each border router further propagates such routing results to neighboring ASs.

4 Path reduction algorithms

4.1 Problem formulation

The problem is to select L out of N paths such that the selected paths will maximize the probability of satisfying various QoS routing requests that would be supported if all the underlying paths were advertised. Before formally defining this problem, we introduce some notations and key concepts.

The notation $\boldsymbol{w}(p) = (w_1(p), w_2(p), \ldots, w_K(p))$ ($K \geqslant 2$) denotes the weight vector of path p and the notation $\boldsymbol{c} = (c_1, c_2, \ldots, c_K)$ denotes the QoS constraint vector. Notation $\boldsymbol{w}(p) \leqslant \boldsymbol{w}(q)$ denotes $w_l(p) \leqslant w_l(q)$ for all $1 \leqslant l \leqslant K$. Other relational operators $<$, $=$, $>$ and $\geqslant$ are defined similarly.

Definition 1 (Multi-constrained path p). Given a directed graph $G(V, E)$, a source node s, a destination node t and K QoS constraints represented by a constraint vector $\boldsymbol{c} = (c_1, c_2, \ldots, c_K)$, the path $p = s \rightarrow v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow t$ is called a multi-constrained path (MCP), if $\boldsymbol{w}(p) \leqslant \boldsymbol{c}$.

Definition 2 (QoS request space S). Given a source node s, a destination node t, a constraint vector $\boldsymbol{c} = (c_1, c_2, \ldots, c_K)$ and K maximum values $C_1, C_2, \ldots, C_K$ for $c_1, c_2, \ldots, c_K$, the set $S(C_1, C_2, \ldots, C_K) = \{(c_1, c_2, \ldots, c_K) \mid 0 \leqslant c_1 \leqslant C_1, 0 \leqslant c_2 \leqslant C_2, \ldots, 0 \leqslant c_K \leqslant C_K\}$ is called the QoS request space from s to t. We write S in brief. From now on, all concepts in this paper are discussed in the QoS request space S. For convenience, we could write $C_l(p) = C_l$ for all $1 \leqslant l \leqslant K$ when point $p \in S(C_1, C_2, \ldots, C_K)$.

Definition 3 (Feasible region of a path $F(p)$). Given the QoS request space S and a path p from source s to destination t with a weight vector $\boldsymbol{w}(p) = (w_1(p), w_2(p), \ldots, w_K(p))$, the set $F(p) = \{\boldsymbol{c} \mid \boldsymbol{c} \in S, \boldsymbol{c} \geqslant \boldsymbol{w}(p)\}$ is called the feasible region of path p in S.

Definition 4 (Feasible region of a path set $F(P_{\langle s,t \rangle})$). Given the QoS request space S from source s to destination t and a path set $P_{\langle s,t \rangle} = \{p_1, p_2, \ldots, p_N\}$ from s to t, the feasible region of path set $P_{\langle s,t \rangle}$ in S is defined as

$$F(P_{\langle s,t \rangle}) = \begin{cases} \phi, & P_{\langle s,t \rangle} = \phi, \\ \displaystyle\bigcup_{i=1}^{N} F(p_i), & \text{else}. \end{cases} \tag{1}$$

According to Definitions 2, 3 and 4, the QoS request space S is a finite region of the K-dimensional space. Any path p can be represented by a point in S and the weight vector $\boldsymbol{w}(p)$ of p is the coordinate of the point. Likewise, any QoS request $\boldsymbol{c}$ can also be represented by a point in S. If $\boldsymbol{c} \in F(p)$, path p is

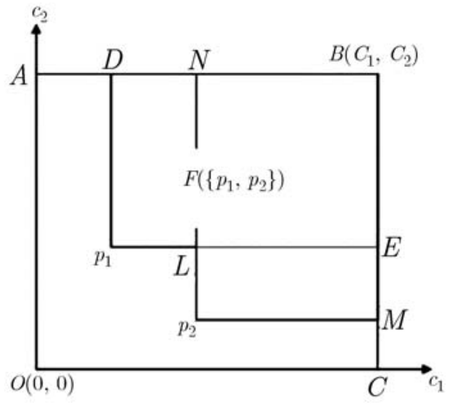

Figure 2 QoS request space S.

a feasible multi-constrained path that satisfies the request $\boldsymbol{c}$. If $\boldsymbol{c} \in F(P_{\langle s,t\rangle})$, there must exist a path in $P_{\langle s,t\rangle}$ that satisfies the request $\boldsymbol{c}$.

To illustrate the above concepts when $K = 2$, consider Figure 2. Rectangular region $OABC$ represents a QoS request regtion $S(C_1, C_2)$, and paths p_1 and p_2 are respectively represented by a point in S. Region BEp_1D represents the feasible region of path p_1, i.e. $F(p_1)$; while region BMp_2Lp_1D represents the feasible region of path set $\{p_1, p_2\}$, i.e. $F(\{p_1, p_2\})$.

Although any path can be represented by a point in the QoS request space, there may not be any real path that corresponds to a point in the QoS request space. However, similar properties hold for both a point and a path. For instance, the concept of feasible region can be defined with respect to either a path or a point. Therefore, in this paper, a path is regarded as equivalent to a point and all path-related concepts also applies to a point in the QoS request space. We shall use paths and points interchangeably in the rest of this paper.

Definition 5 (Intersection of paths $p_1 \wedge p_2$). Given two paths p_1 and p_2 with weight vectors $\boldsymbol{w}(p_1)$ and $\boldsymbol{w}(p_2)$, a path (or a point in S) p' is called the intersection of p_1 and p_2 if $w_l(p') = \max\{w_l(p_1), w_l(p_2)\}$ for all $1 \leqslant l \leqslant K$, where $w_l(p')$ is the lth element of the weight vector (or coordinate) of path p'. We write as $p' = p_1 \wedge p_2$.

Definition 6 (Intersection of path and path set $p \wedge P_{\langle s,t\rangle}$). Given a path p and a path set $P_{\langle s,t\rangle} = \{p_1, p_2, \ldots, p_N\}$, the intersection of p and $P_{\langle s,t\rangle}$ is defined as

$$p \wedge P_{\langle s,t\rangle} = \begin{cases} \phi, & P_{\langle s,t\rangle} = \phi, \\ \{p \wedge p_1, p \wedge p_2, \ldots, p \wedge p_N\}, & \text{else.} \end{cases} \tag{2}$$

According to Definitions 5 and 6, the intersection of two paths is also a path (or a point) while the intersection of a path and a path set is a set of paths (or points), so the concept of feasible region can still apply to the intersection of two paths or of a path and a path set. In Figure 2, the intersection of p_1 and p_2 is the point marked by L, whose feasible region is $BELN$.

Definition 7 (Capacity of feasible region $\Psi(F(P_{\langle s,t\rangle}))$). Given the QoS request space $S(C_1, C_2, \ldots, C_K)$ from source s to destination t and a path set $P_{\langle s,t\rangle} = \{p_1, p_2, \ldots, p_N\}$, the capacity of feasible region for $P_{\langle s,t\rangle}$ is defined as follows:

(i) If $P_{\langle s,t\rangle} = \phi$,

$$\Psi(F(P_{\langle s,t\rangle})) = 0. \tag{3}$$

(ii) If $P_{\langle s,t\rangle}$ contains only one path p, i.e. $P_{\langle s,t\rangle} = \{p\}$,

$$\Psi(F(P_{\langle s,t\rangle})) = \Psi(F(\{p\})) = \prod_{l=1}^{K}(C_l - w_l(p)). \tag{4}$$

(iii) If $P_{\langle s,t\rangle}$ contains more than one path, we arbitrarily select a path $p' \in P_{\langle s,t\rangle}$ and then $P'_{\langle s,t\rangle} = P_{\langle s,t\rangle} - \{p'\}$ is obtained. In this case, we define

$$\Psi(F(P_{\langle s,t\rangle})) = \Psi(F(P'_{\langle s,t\rangle})) + \Psi(F(\{p'\})) - \Psi(F(p' \wedge P'_{\langle s,t\rangle})). \tag{5}$$

Definition 8 (Optimal path reduction problem (OPR))**.** Given the QoS request space S from source s to destination t and a path set $P_{\langle s,t\rangle} = \{p_1, p_2, \ldots, p_N\}$, the problem is to find a subset $\pi \subseteq P_{\langle s,t\rangle}$ with L ($L \leqslant N$) elements such that

$$\pi = \arg \max_{\pi \subseteq P_{\langle s,t\rangle}} \Psi(F(\pi)). \tag{6}$$

In Definition 7, the concept of capacity intuitively implies a measure of space occupied by the feasible region of a path set in the QoS request space. Given a path set, if the multi-constrained QoS requests are uniformly distributed in the QoS request space, the routing success ratio can be represented as the capacity of the feasible region of the path set divided by the capacity of the whole QoS request space. In fact, the capacity is exactly the area in the case of $K = 2$. As displayed in Figure 2, given the path set $\{p_1, p_2\}$, the routing success ratio can be represented as the area of region BMp_2Lp_1D (i.e. $F(\{p_1, p_2\})$) divided by the area of the QoS request space $OABC$. Therefore, the greater the capacity of feasible region is, the more likely the corresponding path set satisfies the QoS constraints that would be satisfied by all the underlying paths. Thus, the goal in the OPR problem is to select L out of N given paths in order to maximize the capacity of feasible region for the path set composed of the selected L ones. In order to solve the OPR problem, we will present an optimal algorithm in subsection 4.3 and propose two contribution based reduction (CBR) algorithms in subsection 4.4. Because all three algorithms need to compute the capacity of feasible region of a path set as a sub-procedure, we first discuss how to figure out this capacity in subsection 4.2.

4.2 Computation of capacity of feasible region (Capacity and ImprovedCapacity)

In this subsection, we prepare two algorithms to compute the capacity of feasible region: Capacity and ImprovedCapacity.

4.2.1 *Capacity algorithm*

As shown in Figure 3, the algorithm Capacity is used to calculate the capacity of feasible region for the give path set (say $PATH0$). It is a recursive procedure and can be roughly divided into three parts:

(i) A path p_0 is arbitrarily selected from $PATH0$ and its capacity of feasible region is figured out (lines 1–5).

(ii) The entire QoS request space is cut into 2^K small request spaces along the K dimensions of p_0 and at the same time any path is cut into different small spaces (lines 6–25).

(iii) In all small spaces except the one p_0 belongs to, the capacity of feasible region for every path set is computed recursively, and then the sum of these capacities plus the capacity of feasible region of p_0 is just the expected result (lines 26–30).

Note that variable $nSeq(p)$ in Figure 3 represents the serial number of the small request space to which p belongs.

Let us now consider the computational complexity of the Capacity algorithm.

Theorem 1. The worst-case time complexity of Capacity algorithm is $O(\frac{(2^K-1)^{N+1}}{(2^K-2)^2})$.

Proof. Let $C(N)$ represent the number of comparison times needed by the algorithm $Capacity(PATH0)$ when set $PATH0$ contains N paths. Obviously $C(0) = 0$. One path can be cut into 2^K new paths at worst, so we get $2^K(N-1)$ new paths after one path is removed from the original N paths and used to cut the left $N-1$ paths. These $2^K(N-1)$ paths are then distributed into 2^K small QoS request spaces, so there are $(N-1)$ paths in each small space. At the same time, we consider the loop from lines 9 to 22 in Figure 3. Whenever the outer loop is finished, the number of paths doubles. Therefore, the total

```
Capacity($PATH0$)
1) IF($PATH0 == \phi$)
2)   RETURN 0
3) Arbitrarily select a path $p_0$ from $PATH0$
4) Delete $p_0$ from $PATH0$
5) $p0\_Capacity = \prod_{l=1}^{K}(C_l(p_0) - w_l(p_0))$
6) FOR EACH path $p$ in $PATH0$
7)   $nSeq(p) = 0$
8) $PATH1 = \phi$
9) FOR($l = 1; l \leqslant K; l++$)
10)   FOR EACH path $p$ in $PATH0$
11)     Delete $p$ from $PATH0$
12)     IF($w_l(p) < w_l(p_0)$)
13)       $p\_copy1 = p$
14)       $p\_copy2 = p$
15)       $w_l(p\_copy2) = w_l(p_0)$
16)       $nSeq(p\_copy2) += 2^{l-1}$
17)       $C_l(p\_copy1) = w_l(p_0)$
18)       Add $p\_copy1$ into $PATH1$
19)       Add $p\_copy2$ into $PATH1$
20)     ELSE
21)       $nSeq(p) += 2^{l-1}$
22)       Add $p$ into $PATH1$
23)   Swap $PATH0$ and $PATH1$
24) FOR EACH path $p$ in $PATH0$
25)   Add $p$ into $PATH[nSeq(p)]$
26) $cpty = p0\_Capacity$
27) FOR($n = 0; n < 2^K - 1; n++$)
28)   $capacityV[n] = Capacity(PATH[n])$
29)   $cpty += capacityV[n]$
30) RETURN cpty
```

Figure 3 Computation of $\Psi(F(PATH0))$.

number of comparison times here is $(N-1) + 2^1(N-1) + \cdots + 2^{K-1}(N-1) = (2^K-1)(N-1)$. Now, the recurrence relation (i.e., difference equation) can be derived as follows:

$$\begin{cases} C(N) = (2^K-1)(N-1) + (2^K-1)C(N-1), \\ C(0) = 0, \end{cases}$$

$$C(N) = \frac{(2^K-1)^{N+1}}{(2^K-2)^2} - \frac{(2^K-1)}{2^K-2}N - \frac{2^K-1}{(2^K-2)^2}. \tag{7}$$

In (7), $C(N)$ is dominated by the first term $\frac{(2^K-1)^{N+1}}{(2^K-2)^2}$, so the worst-case time complexity of Capacity is $O(\frac{(2^K-1)^{N+1}}{(2^K-2)^2})$.

According to Theorem 1, the Capacity algorithm has exponential complexity in the worst case. In order to reduce the complexity, we propose the ImprovedCapacity algorithm in the next subsection.

4.2.2 *ImprovedCapacity algorithm*

This ImprovedCapacity algorithm is based on the following theorem:

Theorem 2. Suppose that $P_{\langle s,t\rangle,N} = \{p_1, p_2, \ldots, p_N\}$ is a path set from source s to destination t, subset $SUB_l^i(P_{\langle s,t\rangle,N}) = \{p_{r1}, p_{r2}, \ldots, p_{rl}\} \subseteq P_{\langle s,t\rangle,N}$ exactly contains l paths $(1 \leqslant l \leqslant N, 1 \leqslant i \leqslant C_N^l)$, and notation $INS(SUB_l^i(P_{\langle s,t\rangle,N})) = p_{r1} \wedge p_{r2} \wedge, \ldots, \wedge p_{rl}$. Then

$$\Psi(F(P_{\langle s,t\rangle,N})) = \sum_{l=1}^{N}(-1)^{l-1}\sum_{i=1}^{C_N^l}\Psi(F(INS(SUB_l^i(P_{\langle s,t\rangle,N})))). \tag{8}$$

Proof. We use mathematical induction with respect to N. When $N = 1$, the LHS of (8) $= \Psi(F(P_{\langle s,t\rangle,1}))$ $= \Psi(F(p_1))$; while the RHS of (8) $= (-1)^{1-1}\Psi(F(INS(SUB_1^1(P_{\langle s,t\rangle,1})))) = \Psi(F(p_1))$. Therefore, eq. (8) is satisfied by $N = 1$.

Suppose (8) is satisfied by N, then we show (8) is also satisfied by $(N+1)$ as follows:

$$\begin{aligned} \Psi(F(P_{\langle s,t\rangle,N+1})) &= \Psi(F(P_{\langle s,t\rangle,N})) + \Psi(F\{p_{N+1}\}) - \Psi(F(p_{N+1} \wedge P_{\langle s,t\rangle,N})) \quad \text{(according to (5))} \\ &= \sum_{l=1}^{N}(-1)^{l-1}\sum_{i=1}^{C_N^l}\Psi(F(INS(SUB_l^i(P_{\langle s,t\rangle,N})))) + \Psi(F\{p_{N+1}\}) \\ &\quad + \sum_{l=1}^{N}(-1)^{l}\sum_{i=1}^{C_N^l}\Psi(F(INS(SUB_l^i(p_{N+1} \wedge P_{\langle s,t\rangle,N})))) \end{aligned}$$

(according to supposed statement).

On the other hand,

$$\begin{aligned}
&\sum_{l=1}^{N+1}(-1)^{l-1}\sum_{i=1}^{C_{N+1}^l}\Psi(F(INS(SUB_l^i(P_{\langle s,t\rangle,N+1}))))\\
&=\sum_{l=1}^{N}(-1)^{l-1}\sum_{i=1}^{C_{N+1}^l}\Psi(F(INS(SUB_l^i(P_{\langle s,t\rangle,N+1}))))\\
&\quad+(-1)^N\Psi(F(INS(SUB_{N+1}^1(P_{\langle s,t\rangle,N+1}))))\\
&=\sum_{l=1}^{N+1}\Psi(F(INS(SUB_l^i(P_{\langle s,t\rangle,N+1}))))+\sum_{l=2}^{N}(-1)^{l-1}\sum_{i=1}^{C_{N+1}^l}\Psi(F(INS(SUB_l^i(P_{\langle s,t\rangle,N+1}))))\\
&\quad+(-1)^N\Psi(F(INS(SUB_{N+1}^1(P_{\langle s,t\rangle,N+1}))))\\
&=\sum_{i=1}^{N}\Psi(F(INS(SUB_1^i(P_{\langle s,t\rangle,N}))))+\Psi(F(\{p_{N+1}\}))\\
&\quad+\sum_{l=2}^{N}(-1)^{l-1}\sum_{i=1}^{C_N^l}\Psi(F(INS(SUB_l^i(P_{\langle s,t\rangle,N+1}))))\\
&\quad+\sum_{l=2}^{N}(-1)^{l-1}\sum_{i=1}^{C_N^{l-1}}\Psi(F(INS(SUB_{l-1}^i(p_{N+1}\wedge P_{\langle s,t\rangle,N}))))+(-1)^N\Psi(F(INS(P_{\langle s,t\rangle,N+1})))\\
&=\sum_{l=1}^{N}(-1)^{l-1}\sum_{i=1}^{C_N^l}\Psi(F(INS(SUB_l^i(P_{\langle s,t\rangle,N}))))\\
&\quad+\Psi(F(\{p_{N+1}\}))+\sum_{l=2}^{N+1}(-1)^{l-1}\sum_{i=1}^{C_N^{l-1}}\Psi(F(INS(SUB_{l-1}^i(p_{N+1}\wedge P_{\langle s,t\rangle,N}))))\\
&=\sum_{l=1}^{N}(-1)^{l-1}\sum_{i=1}^{C_N^l}\Psi(F(INS(SUB_l^i(P_{\langle s,t\rangle,N}))))\\
&\quad+\Psi(F(\{p_{N+1}\}))+\sum_{l=1}^{N}(-1)^{l}\sum_{i=1}^{C_N^l}\Psi(F(INS(SUB_l^i(p_{N+1}\wedge P_{\langle s,t\rangle,N})))).
\end{aligned}$$

Therefore,

$$\Psi(F(P_{\langle s,t\rangle,N+1}))=\sum_{l=1}^{N+1}(-1)^{l-1}\sum_{i=1}^{C_{N+1}^l}\Psi(F(INS(SUB_l^i(P_{\langle s,t\rangle,N+1})))),$$

that is, (8) is satisfied by $(N+1)$.

Eq. (8) in Theorem 2 provides another method to compute the capacity of feasible region. It is derived as the sum of N items, so its computation complexity is still high. However, we find that an item has a decreasing impact on the total sum as l increases in (8). Therefore, we consider an approximation to (8). Specifically, we define

$$M(T)=\sum_{l=1}^{T}(-1)^{l-1}\sum_{i=1}^{C_N^l}\Psi(F(INS(SUB_l^i(P_{\langle s,t\rangle,N})))),\quad(1\leqslant T\leqslant N).\tag{9}$$

Apparently, $M(T)$ is an approximation to $\Psi(F(P_{\langle s,t\rangle,N}))$. Furthermore, the greater T is, the closer they are. Especially, $M(T)=\Psi(F(P_{\langle s,t\rangle,N}))$, when $T=N$.

The algorithm to compute the capacity of feasible region according to eq. (9) is called Improved-Capacity. It has an approximation parameter T. When T is small, this algorithm can achieve $O(KN^T)$ complexity.

4.3 The Optimal algorithm (Optimal)

In order to solve the OPR problem, a common solution is the Optimal algorithm which considers all possible combinations of L out of N paths (totally C_N^L combinations). When the Optimal considers a certain combination, it will call the sub-procedure $Capacity(PATH0)$ that appears in subsection 4.2.1, to calculate the capacity of feasible region of $PATH0$, where $PATH0$ is the set of L paths determined by the current combination. As the Optimal considers all combinations, it is able to exactly select the one with maximum capacity of feasible region. Therefore, the Optimal is regarded as a reference point (upper bound) in our evaluation. The key disadvantage of the Optimal is its high computational complexity, i.e., $O(C_N^L \mid Capaticy \mid)$, where $\mid Capacity \mid$ is the complexity of the sub-procedure *Capaticy*. This makes it impractical.

4.4 Contribution based reduction algorithms (ContriInc and ImprovedInc)

In this subsection, we propose two contribution based reduction (CBR) algorithms: the ContriInc and ImprovedInc algorithms, both of which are based on the following two key concepts.

Definition 9 (Contribution region $C(p; P_{\langle s,t\rangle})$)**.** Given the QoS request space S from source s to destination t, a path set $P_{\langle s,t\rangle} = \{p_1, p_2, \ldots, p_N\}$ and a path $p \notin P_{\langle s,t\rangle}$, the set $C(p; P_{\langle s,t\rangle}) = F(p) - F(P_{\langle s,t\rangle})$ is called the contribution region of p towards $P_{\langle s,t\rangle}$.

Definition 10 (Capacity of contribution region)**.** Given the QoS request space S from source s to destination t, a path set $P_{\langle s,t\rangle} = \{p_1, p_2, \ldots, p_N\}$ and a path $p \notin P_{\langle s,t\rangle}$, the capacity of contribution region of p towards $P_{\langle s,t\rangle}$ is defined as $\delta(C(p; P_{\langle s,t\rangle})) = \Psi(F(p)) - \Psi(F(p \wedge P_{\langle s,t\rangle}))$.

Definition 9 describes how to determine the contribution of path p towards the feasible region of path set $P_{\langle s,t\rangle}$. Definition 10 gives a measure to quantify such a contribution. According to Definition 10, the capacity of contribution region of p is just its capacity of feasible region when $P_{\langle s,t\rangle}$ is empty. Moreover, from Definitions 7 and 10, it comes to the conclusion that $\Psi(F(\{p\} \cup P_{\langle s,t\rangle})) = \Psi(F(P_{\langle s,t\rangle})) + \delta(C(p; P_{\langle s,t\rangle}))$ when $p \notin P_{\langle s,t\rangle}$.

We now describe one of the proposed contribution based reduction (CBR) algorithms, namely the incremental contribution algorithm (ContriInc). The pseudo code of ContriInc is shown in Figure 4(a). In essence, ContriInc selects L paths from set $PATH0$ and stores them into $PATH1$ such that each of the selected paths makes the maximum capacity of contribution to so far selected paths (i.e., $PATH1$). To compute the contribution increment of each path p in $PATH0$ towards $PATH1$ (i.e., $\delta(C(p; PATH1))$), ContriInc calls the sub-procedure ContriCapacity (line 6). As shown in Figure 4(b), ContriCapacity computes $PATH1 = p \wedge PATH0$ (lines 1–5), $\Psi(F(p_0))$ (line 6), and $\Psi(F(PATH1))$ (line 7) by calling the sub-procedure Capacity.

Due to its iterative call to the Capacity algorithm proposed in subsection 4.2.1, the worst-case complexity of the ContriInc algorithm is also exponential with respect to the problem scale, according to Theorem 1. The bottleneck in the complexity of the ContriInc algorithm lies in the sub-procedure Capacity. For the reason above, we propose an improved CBR algorithm ImprovedInc(T) with polynomial complexity.

If the algorithm in Figure 4(b) calls ImprovedCapacity (subsection 4.2.2) instead of Capacity (line 7) and the other parts in Figure 4 remain unchanged, the improved version of ContriInc is obtained, denoted as ImprovedInc(T), where T is a parameter representing the computational approximation of the algorithm. In fact, ImprovedCapacity reduces computation complexity by introducing approximation.

We now analyze the computation complexity of the ImprovedInc algorithm when T is small. In Figure 4(a), the bottleneck in complexity lies in calling ContriCapacity (line 6). In Figure 4(b), the complexity of ContriCapacity can be restricted to $O(Kn^T)$ when $PATH1$ contains n paths and T is small. Therefore, the complexity of ImprovedInc is $O(\sum_{i=1}^{L-1} Ki^T) = O(KL^{(T+1)})$. Extensive simulations in section 5 show that the ImprovedInc algorithm achieves very high performance when $T = 2$. Thus, the ImprovedInc can be restricted to $O(KL^3)$ complexity in practice.

```
ContriInc()
1) $PATH0 = \{p_1, p_2, \ldots, p_N\}$
2) $PATH1 = \phi$
3) FOR($i = 0; i < L; i++$)
4)   $maxCpty = -1$
5)   FOR EACH $p$ in PATH0
6)     $cpty$=ContriCapacity($p, PATH1$)
7)     IF($cpty > maxCpty$)
8)       $maxCpty = cpty$
9)       $maxp = p$
10)  Add $maxp$ into $PATH1$
11)  Delete $maxp$ from $PATH0$
12) RETURN $PATH1$
```

(a)

```
ContriCapacity($p_0, PATH0$)
1) FOR EACH $p$ in $PATH0$
2)   FOR($l = 1; l \leqslant K; l++$)
3)     IF($w_l(p) < w_l(p_0)$)
4)       $w_l(p) = w_l(p_0)$
5)     Add $p$ into $PATH1$
6) $cpty1 = \prod_{l=1}^{K}(C_l - w_l(p_0))$
7) $cpty2$=Capacity($PATH1$)
8) RETURN $cpty1 - cpty2$
```

(b)

Figure 4 The incremental contribution algorithm (ContriInc). (a) Main routine; (b) Sub-procedure: computation of $\delta(C(p_0; PATH0))$.

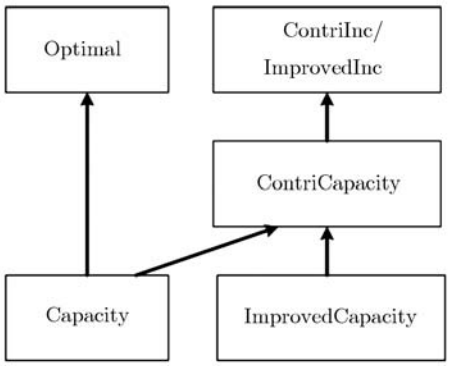

Figure 5 Relationship of function call among algorithms.

4.5 Brief summary of algorithms

Now we make a summary of the relationship of function calls among algorithms, as shown in Figure 5. The following facts can be found in Figure 5: When ContriCapacity calls Capacity, we get the incremental contribution algorithm (ContriInc); When ContriCapacity calls ImprovedCapacity, we get the improved incremental contribution algorithm (ImprovedInc).

5 Performance evaluation

Simulations in this section include two parts: In part 1, we evaluate the performance and running time of the proposed CBR algorithms; in part 2, we evaluate the success rate and message overhead of the proposed QoS extension schemes to BGP.

5.1 Evaluation of path reduction algorithms

We compare two CBR algorithms, i.e., the incremental contribution algorithm (ContriInc) and its improved version (ImprovedInc), with the Optimal algorithm, in terms of performance and running time.

We randomly generate N paths, i.e., $w_l(p_i) \sim Uniform(1, C_l)$ for all $1 \leqslant l \leqslant K$ and $1 \leqslant i \leqslant N$. In this section of simulations, we set $C_l = 1000$ for all $1 \leqslant l \leqslant K$. In order to reduce errors, each simulation is repeated 100 times and the average results are used. When evaluating running times, we run each algorithm in a PC with a Pentium (R) 4 CPU of 3.00 GHz, 1.00 GB RAM and Windows XP system.

In order to evaluate the performance of each algorithm, we define capacity ratio as the capacity of feasible region of the selected L paths divided by the capacity of feasible region of all N paths. The capa-

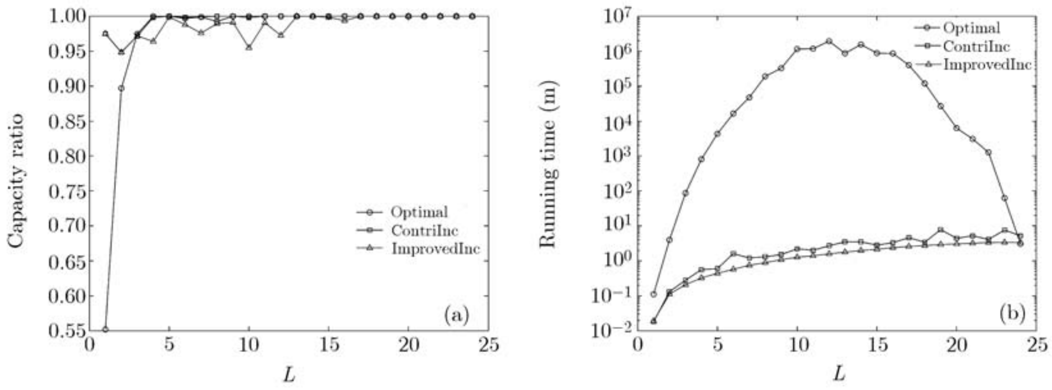

Figure 6 Comparison when L is varied. (a) Capacity ratio; (b) running time.

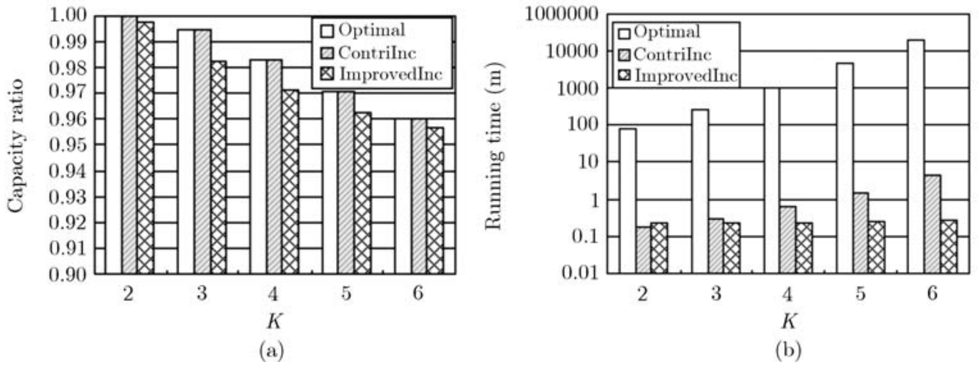

Figure 7 Comparison when K is varied. (a) Capacity ratio; (b) running time.

city ratio changes in $[0, 1]$ and that of the Optimal algorithm should be always the greatest of all algorithms (upper bound).

In Figure 6, we compare the capacity ratio and running time of all the three algorithms when L changes. $N = 24$, $K = 3$ and $T = 2$ in simulations. Figure 6(a) shows that the ContriInc algorithm almost achieves the same capacity ratio with Optimal and the capacity ratio of ImprovedInc is slightly smaller. When L increases from 1, the capacity ratio of each algorithm rapidly rises and when $L = 5$ or 6, it has been close to one. In Figure 6(b), the y-axis represents the running time, in logarithmic scale. The running time of Optimal is much longer than the other two algorithms in the mass, especially when L approximates to $N/2$. On the contrary, ImprovedInc has the shortest running time.

In Figure 7, we compare the capacity ratio and running time of all algorithms when K changes. Note that the y-axis in Figure 7(b) is in logarithmic scale. $N = 16$, $L = 5$ and $T = 2$ in simulations. From Figure 7(a), we find that the capacity ratio of each algorithm just decreases slightly as K increases. While Figure 7(b) shows that it takes every algorithm widely different running time. ImprovedInc runs much faster than the other two algorithms. Furthermore, the running time of Optimal or ContriInc is exponential with respect to K, but ImprovedInc is insensitive to K.

In Figure 8, we compare the capacity ratio and running time of all algorithms when T changes. Note that the y-axis in Figure 8(b) is in logarithmic scale. $N = 16$, $L = 10$ and $K = 3$ in simulations. Because Optimal and ContriInc have nothing to do with T, the capacity ratio and running time of them basically remains constant when T changes. From Figure 8(a), we find that the capacity ratio of ImprovedInc has an upward trend when T increases, but it is quite high when $T = 2$. From Figure 8(b), it can be found that the running time of ImprovedInc rises relatively fast as T increases and eventually exceeds that of ContriInc, but ImprovedInc has the lowest time complexity when $T \leqslant 2$. Therefore, it is ideal to let $T = 2$ in practice.

5.2 Evaluation of the QoS extension schemes to BGP

We implement a simulator to evaluate the QoS routing performance and routing message overhead for

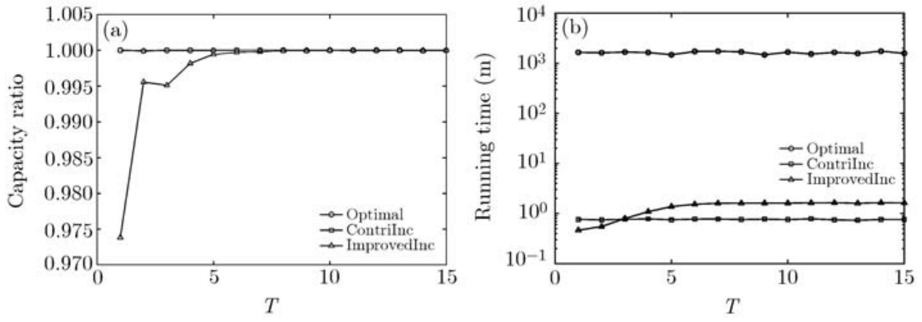

Figure 8 Comparison when T is varied. (a) Capacity ratio. (b) running time.

inter-domain and intra-domain routing systems. The simulator is based on a 2-level hierarchical topology model. In the simulations, the Internet topology generator BRITE [29] is used to generate 2-level top-down topologies. Both the inter-AS and the intra-AS topologies are generated using Waxman [30] model.

For comparison, we simulate four BGP-based routing systems, including our proposed extension to BGP. The four routing systems are classified according to their different intra-AS routing algorithms and inter-AS path selection policies:

1. Shortest AS_PATH routing (SASP): The standard shortest path algorithm runs within an AS and the path with the shortest AS_PATH is selected in inter-AS routing. This SASP routing is just similar to the real BGP system.

2. Smallest average of weights routing (SAW): The shortest path algorithm runs with respect to the average of K ($K \geqslant 2$) weights within an AS and the path with the smallest average of weights is selected in inter-AS routing.

3. L random paths routing (LRAND): Any multi-constrained QoS routing algorithm can be used to precompute L ($L \geqslant 1$) paths within an AS and L paths are randomly selected in inter-AS routing.

4. All paths routing (APR): Any multi-constrained QoS routing algorithm can be used to precompute paths within an AS and all paths are selected in inter-AS routing.

5. Our proposed CBR routing (CBR): Any multi-constrained QoS routing algorithm can be used to precompute L paths within an AS and L good paths are selected using CBR algorithms in inter-AS routing.

We first generate a topology with 50 ASs and 50 routers (totally 2500 nodes) within each AS using BRITE. And then for the topology 10 instances of link weights are generated. Finally, for each instance of link weights, 1000 routing requests are generated. When link weights are generated, each link weight $w_l(e)$ is uniformly distributed in $[0, \max W]$. As for the routing constraints from source s to destination t, each constraint is randomly selected from $[\alpha * \max W * \min D(P_{\langle s,t\rangle}), \beta * \max W * \min D(P_{\langle s,t\rangle})]$, where $\alpha \in [0,1]$, $\beta \geqslant 1$ and $\min D(P_{\langle s,t\rangle})$ is the shortest distance from s to t. Here α and β are called constraint factors. For any node (say u), when the CBR simulation is conducted, we choose the ImprovedInc as the path reduction algorithm for its low complexity. The parameters of the algorithm is as follows: $T = 2$ and $C_l = \beta * \max W * \min D(P_{\langle u,t\rangle})$ $(1 \leqslant l \leqslant K)$, where $\min D(P_{\langle u,t\rangle})$ is the shortest distance from u to t. Note that the parameter C_l should be set according to the statistical measurement results of each node in real networks and here it is only a simple estimate for the state of routing requests. Therefore, our proposed BGP QoS extension and related algorithms may be more powerful in practice than the simulation results will show. In addition, for LRAND and CBR routing systems, we choose the limited path heuristic [25] and MEFPA [26] as the intra-AS routing algorithms. For space limit, the results displayed in this section are only based on the limited path heuristic. However, this choice of intra-AS QoSR algorithms has little effect on the overall results.

In order to evaluate the performance of each routing system, the success ratio (SR) is defined as the number of satisfied QoS routing requests divided by the number of all the generated routing requests. Figure 9(a) shows the success ratio from source s to destination t after the routing message exchange becomes convergent. It can be found that the CBR routing achieves the biggest success ratio. This is because the CBR routing can provide multiple paths when $L > 1$. As L increases, the success ratio of

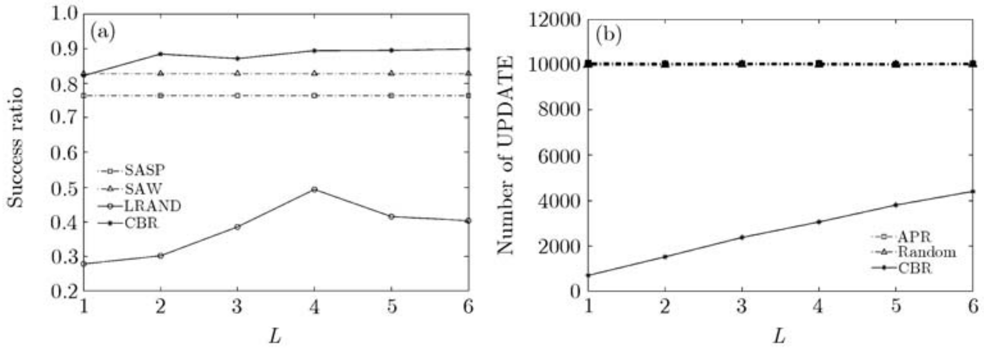

Figure 9 Comparison when L is varied. (a) Success ratio; (b) number of UPDATE.

the CBR routing becomes higher but more routing messages are required to be exchanged between BGP routers. Especially when a new AS is added to an existing network, in order to obtain routing information about each node in the added AS, UPDATE messages are required to be advertised all over the whole network until convergence. As shown in Figure 9(b), the y-axis represents the number of UPDATE messages that are needed to advertise the routing information about one certain destination node. We found that in simulation the LRAND routing system is difficult to converge when the network becomes big, so we make the system stop advertising UPDATE messages when the number of total UPDATE messages reaches 100000. From simulation we found that the number of UPDATE messages is linear to L in CBR routing and this brings about good scalability of the protocol design. Because the LRAND routing system is difficult to converge, it requires quite a lot of UPDATE advertisement, much more than the CBR routing system. From the result we also see that the number of UPDATE messages which APR routing system advertises also exceeds the maximum limit(100000). Note that the LRAND routing shows low and unpredictable success ratio in Figure 9(a). And the success ratio of LRAN in Figure 9(a) is not sampled in the convergence state. Therefore, when multiple metrics/paths are added to BGP, careful design is required to insure the convergence and scalability. Our path reduction algorithms are apt to select the same path set every time the path selection process is performed. So the convergence can be achieved by the CBR system.

We have also studied the success ratio and the number of UPDATEs when K is varied by simulations. We find that the success ratios of all routing systems slightly decrease as K increases its value. We also find that the number of UPDATE messages is relatively insensitive to K. The related figures are omitted here for space limit.

6 Conclusions

In this paper, we studied the problem of inter-domain QoS-based routing and specifically investigated how to extend BGP to support QoS under multiple metrics. One of the key challenges is how to make BGP aware of multiple QoS metrics and advertise the underlying routes with different QoS metrics. In BGP, it is essential to reduce the routing message overhead without significantly sacrificing the performance. In response to this, we proposed two contribution based reduction (CBR) algorithms, namely, the incremental contribution algorithm and its improved version, to perform path selection and reduce the number of UPDATE messages in BGP.

In the extended BGP, every node within one AS obtains multiple paths to each single destination through some IGP and the intra-domain QoS routing computation. Afterwards, each border router within the AS advertises the routing messages about these multiple paths to its peer within a different AS, using E-BGP with QoS extension. When a border router receives multiple paths from another AS, it propagates this routing information within its own AS, using I-BGP with QoS extension. After a node within the same AS receives all alternate paths from every border router, it calls one of the proposed CBR algorithms to reduce the total number of the paths to an acceptable level. Finally, each node installs the selected paths into the local forwarding table and further sends such routing results to neighboring ASs.

In essence, our proposed CBR algorithms try to select the least number of paths that make the greatest contribution in terms of covering the underlying QoS feasibility region of all paths. Therefore, they can reduce the number of routes advertised by BGP speakers while maximizing the routing success ratio, making the QoS extension to BGP scalable.

Under this scalable BGP QoS extension framework, we expect that the traditional protocol systems be well integrated with QoS schemes to provide new techniques supporting new QoS-oriented service in the next generation network.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant Nos. 60911130511, 60873252, 60403035), the National Basic Research Program of China (Grant Nos. 2009CB320501, 2009CB320503), the National High-Tech Research & Development Program of China (Grant No. 2008AA01A324).

References

1 Huitema C. Routing in the Internet. Upper Saddle River, NJ: Prentice Hall, Inc., 1995
2 Halabi B. Internet Routing Architectures. Las Vegas, NV: Cisco Press, 1997
3 Moy J. OSPF version 2, Standards Track RFC 2328, IETF, April 1998
4 Rekhter Y, Li T. A border gateway protocol 4 (BGP-4). Standards Track RFC 1771, IETF, March 1995
5 Bejerano Y, Breitbart Y, Orda A. Algorithms for computing qos paths with restoration. IEEE/ACM Trans Netw, 2005, 13: 648–661
6 Crawley E, Nair R, Rajagopalan B, et al. A framework for QoS-based routing in the Internet. Technical Report, RFC 2386, IETF, August 1998
7 Xiao X, Ni L M. Internet QoS: A big picture. IEEE Netw, 1999, 13: 8–18
8 Van Mieghem P E, Kuipers F, Korkmaz T, et al. Quality of service routing. In: Smirnov M, Crowcroft J, Roberts J, et al., eds. Quality of Future Internet Services, LNCS 2856. Berlin: Springer, 2003. Ch. 3: 80–117
9 Guerin R, Orda A, Williams D. QoS routing mechanisms and OSPF extensions. In: GLOBECOM '97, Vol. 3, IEEE, Phoenix, Arizona, 1997. 1903–1908
10 Jacquene C. Providing quality of service indication by the BGP-4 protocol: the QOS NLRI attribute. Technical Report, (draft-jacquenet-qos-nlri-00.txt), IETF, July 2000
11 Xiao L, Lui K S, Wang J, et al. QoS extension to BGP. In: Proceedings 10th IEEE International Conference on Network Protocols, Paris, France, 2002. 100–109
12 Bonaventure O. Using BGP to distribute exible QoS information. Technical Report, (draft- bonaventure-bgp-qos-00.txt), IETF, February 2001
13 Abarbanel B, Venkatachalam S. BGP-4 support for traffic engineering. Technical Report (draft-abarbanel-idr-bgp4-te-00.txt), IETF, September 2000
14 Fei A, Gerla M. Extending BGMP for shared-tree interdomain qos multicast. In: Proceedings of IWQoS 2001, Karlsruhe, Germany, 2001. 123–139
15 Wang Z, Crowcroft J. Quality-of-service routing for supporting multimedia applications. IEEE J Select Areas Commun, 1996, 14: 1228–1234
16 Jaffe J M. Algorithms for finding paths with multiple constraints. Networks, 1984, 14: 95–116
17 Raz D, Shavitt Y. Optimal partition of QoS requirements with discrete cost functions. In: INFOCOM, Vol. 2, Tel-Aviv, Israel, 2000. 613–622
18 Lorenz D H, Orda A, Raz D, et al. Efficient QoS partition and routing of unicast and multicast. In: IWQoS 2000, Karlsruhe, 2000. 75–83
19 Pornavalai C, Chakraborty G, Shiratori N. QoS based routing algorithm in integrated services packet networks. J High Speed Netw, 1998, 7: 99–112
20 Ma Q, Steenkiste P. Quality-of-service routing for traffic with performance guarantees. In: Proceedings of the Fifth International IFIP Workshop on Quality of Service, Columbia University, 1997. 115–126
21 De Neve H, Van Mieghem P. A multiple quality of service routing algorithm for PNNI. In: Proceedings of the ATM Workshop, IEEE, Seattle, Washington, USA, 1998. 324–328
22 Van Mieghem P, Kuipers F. Hop-by-hop quality of service routing. Comput Netw, 2001, 37: 407–423
23 Korkmaz T, Krunz M. Multi-constrained optimal path selection. In: Proceedings of the INFOCOM 2001 Conference, Vol. 2, IEEE, Anchorage, Alaska, 2001. 834–843

24 Orda A. Precomputation schemes for qos routing. IEEE/ACM Trans Netw, 2003, 11: 578–591

25 Yuan X, Liu X. Heuristic algorithms for multi-constrained quality of service routing. In: Proceedings of the INFOCOM 2001 Conference, Vol. 2, IEEE, Anchorage, Alaska, 2001. 844–853

26 Cui Y, Xu K, Wu J. Precomputation for multi-constrained QoS routing in high-speed Networks. In: Proceedings of the INFOCOM 2003 Conference, San Francisco, 2003, 2: 1414–1424

27 Kapoor R, Chen L J, Lao L, et al. Capprobe: a simple and accurate capacity estimation technique. SIGCOMM Comput Commun Rev, 2004. 67–78

28 Dovrolis C, Ramanathan P, Moore D. What do packet dispersion techniques measure? In: Proceedings of IEEE INFOCOM. Anchorage: IEEE Press, 2001. 905–914

29 Boston University representative Internet topology generator, http://www.cs.bu.edu/brite/.

30 Waxman B M. Routing of multipoint connections. IEEE JSAC, 1988, 6: 1617–1622