多约束服务质量路由中的路径压缩算法

赵有健 张铁蕾 崔 勇
(清华大学计算机科学与技术系 北京 100084)

摘 要 多约束服务质量路由是一种能够支持灵活的服务质量控制的有效方案.然而在多约束的环境下,从一个源节点到一个目的节点可能存在多条路径,因而必须相应地增大路由表容量.由于当前路由表的规模已相当庞大,尤其是在高速核心网中,因此,为了在 QoS 路由表中存储更少的路径信息,需要首先进行路径压缩.文章以解决最优路径压缩问题(OPR)为目标,力图在尽量减小路由表存储规模的同时使路由成功率最大化.为了实现这个目标,文中提出了两个基于贡献区域的算法:增量贡献算法和改进的增量贡献算法.这两个算法从一个大的多约束路径集合中依次计算出具有最大贡献区域的积的路径,最后得到一个小的结果路径集合.大量模拟实验表明,这两个算法能够以较低的运算复杂度获得令人满意的路由成功率.

关键词 服务质量路由;覆盖网络;多约束;路径选择;预计算
中图法分类号 TP393

Path Reduction for Multi-Constrained Quality-of-Service Routing

ZHAO You-Jian ZHANG Tie-Lei CUI Yong
(*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084)

Abstract Multi-constrained quality-of-service routing (QoSR) is regarded as a promising solution to support flexible QoS-oriented services. However, there may exist multiple paths from a source node to a destination node in the context of multi-constrained routing and thus the routing table has to be enlarged accordingly. Since the current routing table size is quite large, especially in the high-speed core networks, path reduction needs to be performed in order to store less paths into the QoS routing table. In this paper the authors try to solve the Optimal Path Reduction Problem (OPR) which aims to reduce the storage space of the QoS routing table as much as possible and at the same time to maximize routing success ratio. To achieve this goal, the authors propose two contribution region based algorithms: the incremental contribution algorithm and its improved version. These two algorithms figure out the path with maximum capacity of contribution region one by one from a large set of multi-constrained paths and finally obtain a small set of selected paths. Extensive simulations show that these two algorithms achieve satisfying performance in terms of the routing success ratio with low time complexity.

Keywords QoS routing; overlay; multiple constraints; path selection; precomputation

1 引 言

当前的 Internet 只能提供尽力而为(best-effort)的服务.然而,下一代网络要求能够支持多种不同的应用,提供面向 QoS 的服务[1].为了提供服务质量,可以借助 Overlay 技术,在现有的 Internet 之上支持诸如语音、视频等多种 QoS 应用.现在这方面已

收稿日期:2006-01-10;最终修改稿收到日期:2007-09-18.本课题得到国家自然科学基金(90604029,60403035)和国家"九七三"重点基础研究发展规划项目基金(2003CB314801)资助.**赵有健**,男,1969 年生,博士,副研究员,主要研究方向为计算机网络体系结构、高速计算机网络设备.E-mail: zhaoyj@csnet1.cs.tsinghua.edu.cn. **张铁蕾**,男,1981 年生,硕士研究生,主要研究方向为服务质量路由和算法. **崔 勇**,男,1976 年生,博士,助理研究员,主要研究方向为计算机网络体系结构、服务质量控制、路由算法和性能评价.

存在大量研究(如文献[2-3]等).然而,Overlay 网络需要解决本身的路由问题,特别是在面向 QoS 应用的 Overlay 网络中,服务质量路由(QoSR)仍然是关键的问题之一[4].

服务质量路由计算可分为预计算和在线计算两种[5].预计算是一个后台进程根据网络状态信息构造路由表,而在 QoS 请求时,通过快速查找路由表确定可行路径的方式.在线计算是 QoS 请求到达时,根据状态信息计算可行路径的方式.由于在 QoS 连接请求高速到达情况下在线计算所需要的计算开销很大,因此在高速核心网络中路由预计算方式更为可取[6].

然而,服务质量路由具有多约束(如同时满足带宽、延迟约束等)的特性,因此在一对源和目的节点之间存在多条路径,甚至路径数目可能随网络规模的增大成指数增长[7].这样,当采用路由预计算时,为了提供对不同服务质量的支持,必须在 QoS 路由表中为每个目的节点保存多个路由信息,然后根据不同的 QoS 约束从中选择一个或多个.由于核心网中核心节点的路由表规模已相当庞大,现在为了提供服务质量将使得路由表规模成倍地提高,因此路由表的容量成为瓶颈,不可能为每个目的节点保存所有可能的路由信息.这就要求在众多路径中进行选择,使它们能够以较大的可能性满足各种 QoS 约束,本文称之为路径压缩(path reduction).

多约束路径选择问题[8]是服务质量路由所要研究的重要课题.由于它是一个 NPC 问题[9],许多启发式算法被设计出来.然而,其中有些启发式算法只支持在线计算,因而不能适应核心网高速路由计算的需求,如基于非线性函数[1]的一类算法.而对于另一些支持预计算的算法来说,针对每个目的节点的路径数目是它们的一个可调参数(如 MEFPA 算法[4]的 B 和路径受限算法[7]的 X),自然可以通过将此参数设小以减小路由表容量,但同时也会造成路由成功率的显著降低.而本文的算法在实施路径压缩的同时,力图获得最优的路由成功率.

本文针对最优路径压缩问题(Optimal Path Reduction Problem,OPR),力图从一对源和目的节点之间存在的 N 条路径中选出 $L(L\leq N)$ 条路径,使得这 L 条路径能够以最大的可能性满足各种 QoS 约束.本文首先给出最优算法,并以此作为其它各种算法的性能上界.由于最优算法的计算复杂度很高,因此本文基于贡献区域的概念,提出复杂度更低的两个算法:增量贡献算法和改进的增量贡献算法,并对各算法进行了对比和分析.这两个算法根据一条路径相对于一个路径集合的贡献大小,逐次选择出需要的路径.它们能够以较低的计算复杂度提供接近最优算法的路径压缩结果,具有很高的性能和良好的可扩展性.

2　相关工作

在 QoSR 领域,许多算法被设计出来用于寻找一条满足多个约束的可行路径.对于二重约束问题,文献[8]提出了一种基于线性函数 $g(p)=a_1w_1(p)+a_2w_2(p)$ 的分布式算法.另有一些算法[10-11]专门用于解决延迟受限最小花费(Delay-Constraint Least Cost,DCLC)问题.另外,基于某些特定的调度机制,QoS 参数之间存在相关性.这样,原来 NP 完全的多约束问题就能够简化为标准的最短路径问题[12-13].

对于更一般的多约束路径选择问题,TAMCRA[14]和它的改良版本 SAMCRA[15]力图使得非线性函数 $g_\lambda(p)=\sum_{l=1}^{k}(w_l(p)/c_l)^\lambda$ 最小化.更进一步,H_MCOP[1]的目标是解决多重约束路径优化问题,该算法的设计原则也是基于非线性函数 $g_\lambda(p)$.然而,这些算法只能支持在线路由计算,这使得它们在高速核心网络中无法胜任.文献[7]基于扩展 Bellman-Ford 算法(EBFA)提出了粒度受限(limited granularity)和路径受限(limited path)两个启发式算法,它们可以支持路由预计算.除此之外,MEFPA 是专门为路由预计算而设计的,它为每个可能的多重服务质量约束预先计算出多条可行路径[4].

然而,一方面,高速核心网络需要支持路由预计算的算法;另一方面,即使是能够支持预计算的现有算法,也没有考虑对于路径的最优压缩问题.本文提出的两个算法,目标正是解决最优路径压缩问题,在尽量减小路由表存储规模的同时使路由成功率最大化.而且,鉴于已经存在很多 QoSR 路由算法(如 EBFA、粒度受限、路径受限、MEFPA 等)能够计算出从源节点到目的节点的一个路径集合,因此,本文的路径压缩算法以这样的一个路径集合作为输入,压缩后得到一个路径子集.

3　最优路径压缩问题

有向图 $G(V,E)$ 表示一个网络,其中 V 是节点

集合，代表路由器；E 是边集合，代表链路. 每条边 $e\in E$ 具有 K 个权值 $w_1(e),w_2(e),\cdots,w_K(e)$，并且 $w_l(e)\in R^+(1\leq l\leq K)$. 路径 $p=v_0\rightarrow v_1\rightarrow v_2\rightarrow\cdots\rightarrow v_n$ 同样也有 K 个权值，其中第 $l(1\leq l\leq K)$ 个权值记为 $w_l(p)$. 根据权值类型的不同，路径权值与边的权值存在不同的关系. 这些权值可分成三种类型：可加性权值(如花费、延迟)、可乘性权值(如链路丢失率)和最小性权值(如带宽). 对于可加性权值，$w_l(p)=\sum_{e\in p}w_l(e)$；对于可乘性权值，$w_l(p)=\prod_{e\in p}w_l(e)$；对于最小性权值，$w_l(p)=\min_{e\in p}w_l(e)$. 实际上，可乘性权值能够通过取对数转换成可加性权值[1]. 因此，我们只需要考虑可加性权值和最小性权值.

在本文下面的描述中，$\boldsymbol{w}(p)=(w_1(p),w_2(p),\cdots,w_K(p))$ 表示路径 p 的权值向量. 对于路径 p 和 q，$\boldsymbol{w}(p)\leq\boldsymbol{w}(q)$ 表示的意义是：对于所有的 $1\leq l\leq K$ 满足 $w_l(p)\leq w_l(q)$. 对于其它的操作符(如$<$，$=$，$>$，$\geq$或$+$)和权值向量之间的比较和操作，都具有与此类似的定义.

定义 1. 多约束路径 p(Multi-Constrained Path). 给定一个有向图 $G(V,E)$，源节点 s，目的节点 t，向量 $\boldsymbol{c}=(c_1,c_2,\cdots,c_K)$ 表示 K 个 QoS 约束，如果路径 $p=s\rightarrow v_1\rightarrow v_2\rightarrow\cdots\rightarrow t$ 满足 $\boldsymbol{w}(p)\leq\boldsymbol{c}$，即对于所有的 $1\leq l\leq K$ 都有 $w_l(p)\leq c_l$，则路径 p 称为多约束路径(MCP).

注意，在定义 1 中，对于 w_l 和 c_l 是最小性权值(例如 w_l 表示带宽，c_l 表示用户对带宽的需求)的情况，我们可以将 w_l 和 c_l 取倒数，则转化成最大性权值，即令 $\frac{1}{w_l(p)}=\max_{e\in p}\frac{1}{w_l(e)}$. 这样，定义 1 中的不等式就可表示成 $\frac{1}{w_l(p)}\leq\frac{1}{c_l}$，因此，定义 1 的表述形式对各种权值类型都可适用. 本文所讨论的路径都是指多约束路径.

图 1 是 $K=2$ 时的一个例子. 有向图包含 4 个节点和 5 条边，每条边各有 2 个权值，已在图中标出. 从源节点 s 到目的节点 t，共有 3 条路径，分别是 sat，sbt 和 st. 它们的权值向量分别是 $\boldsymbol{w}(sat)=(1,2)+(1,2)=(2,4)$；$\boldsymbol{w}(sbt)=(2,1)+(2,1)=(4,2)$；$\boldsymbol{w}(st)=(5,3)$. 对于约束向量 $\boldsymbol{c}=(3,5)$，只有路径 sat 满足 $\boldsymbol{w}(sat)\leq\boldsymbol{c}$，是满足约束 $\boldsymbol{c}$ 的多约束路径.

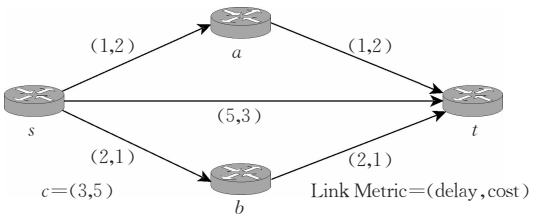


图 1 有向图和多约束路径示意图

定义 2. QoS 请求空间 S(QoS Request Space). 设 $\boldsymbol{c}=(c_1,c_2,\cdots,c_K)$ 表示从源节点 s 到目的节点 t 的路由请求对于路径权值的 K 个 QoS 约束，C_1，C_2，$\cdots$，C_K 分别是 $c_1,c_2,\cdots,c_K$ 的最大可能取值，那么集合 $S(C_1,C_2,\cdots,C_K)=\{(c_1,c_2,\cdots,c_K)\mid 0\leq c_1\leq C_1,0\leq c_2\leq C_2,\cdots,0\leq c_K\leq C_K\}$ 称为从 s 到 t 上的 QoS 请求空间，简记为 S. 本文下面的所有概念都在 QoS 请求空间之内定义. 为描述方便，当一个点 $p\in S(C_1,C_2,\cdots,C_K)$ 时，可记 $C_l(p)=C_l(1\leq l\leq K)$.

定义 3. 一条路径的可行区域 $F(p)$(Feasible Region of a Path). 设 S 为从源节点 s 到目的节点 t 上的 QoS 请求空间，路径 p 的权值向量为 $\boldsymbol{w}(p)=(w_1(p),w_2(p),\cdots,w_K(p))$，则集合 $F(p)=\{(c_1,c_2,\cdots,c_K)\mid(c_1,c_2,\cdots,c_K)\in S,(c_1,c_2,\cdots,c_K)\geq\boldsymbol{w}(p)\}$ 称为路径 p 在请求空间 S 中的可行区域.

定义 4. 路径集合的可行区域 $F(P_{\langle s,t\rangle})$(Feasible Region of a Path Set). 设 S 为从源节点 s 到目的节点 t 上的 QoS 请求空间，$P_{\langle s,t\rangle}=\{p_1,p_2,\cdots,p_N\}$ 是从 s 到 t 的一个路径集合，$F(p_1)$，$F(p_2)$，$\cdots$，$F(p_N)$ 分别是路径 $p_1,p_2,\cdots,p_N$ 在 S 中的可行区域，则路径集合 $P_{\langle s,t\rangle}$ 在请求空间 S 中的可行区域定义为

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

由定义 2～4 可知，QoS 请求空间是 K 维空间中的一个有限区域，任意一条路径 p 都可用 QoS 请求空间中的一个点来表示，其权值向量 $\boldsymbol{w}(p)$ 就是该点的坐标. 同样，在 QoS 请求空间中，任意一个 QoS 请求 $\boldsymbol{c}$ 也可以用一个点来表示，其约束向量 $(c_1,c_2,\cdots,c_K)$ 表示该点坐标. 如果存在路径 p 使得 $\boldsymbol{c}\in F(p)$，则 p 就是能够满足请求 $\boldsymbol{c}$ 的一条多约束路径；如果存在路径集合 $P_{\langle s,t\rangle}$ 使得 $\boldsymbol{c}\in F(P_{\langle s,t\rangle})$，则路径集合 $P_{\langle s,t\rangle}$ 中必定存在一条能够满足请求 $\boldsymbol{c}$ 的多约束路径.

图 2 表示了 $K=2$ 的情况，区域 $OABC$ 表示的是一个 QoS 请求空间，路径 p_1，p_2 在这个请求空间中分别用一个点表示. 区域 BEp_1D 表示路径 p_1 的可行区域 $F(p_1)$，BMp_2N 表示路径 p_2 的可行区域 $F(p_2)$；而区域 BMp_2Lp_1D 表示的是路径集合 $\{p_1,$

$p_2\}$的可行区域 $F(\{p_1,p_2\})$.

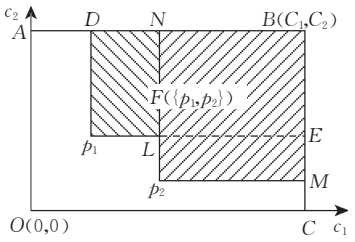


图 2　QoS 请求空间示意图

虽然任意一条路径可以用 QoS 请求空间中的一个点来表示，但 QoS 请求空间中的任意一个点不一定存在与其相对应的实际路径. 然而，在本文所讨论的 QoS 请求空间中，一个点也具有与一条路径相同的性质(例如可行区域等概念也同样适用于一个点)，因此，本文将路径和 QoS 请求空间中的点看作是等价的，有关路径的所有概念对于 QoS 请求空间中的点也同样适用.

定义 5.　路径的交 $p_1 \wedge p_2$ (Intersection of Paths). 设两条路径 p_1 和 p_2 的权值向量分别为 $\boldsymbol{w}(p_1)$ 和 $\boldsymbol{w}(p_2)$，假设 p' 是一条路径(或 QoS 请求空间中的一个点)，其对应的权值向量(或坐标)为 $\boldsymbol{w}(p')$，且 $\boldsymbol{w}(p')$ 的每个分量 $w_l(p')=\max\{w_l(p_1),w_l(p_2)\}$ $(1\leq l\leq K)$，则路径 p' 称为路径 p_1 和 p_2 的交，记为 $p'=p_1\wedge p_2$.

定义 6.　路径与路径集合的交 $p\wedge P_{\langle s,t\rangle}$ (Intersection of a Path and a Path Set). 设 p 是一条路径，$P_{\langle s,t\rangle}=\{p_1,p_2,\cdots,p_N\}$ 是一个路径集合，则路径 p 与路径集合 $P_{\langle s,t\rangle}$ 的交定义为

$$p\wedge P_{\langle s,t\rangle}=\begin{cases}\varnothing, & P_{\langle s,t\rangle}=\varnothing\\ \{p\wedge p_1,p\wedge p_2,\cdots,p\wedge p_N\}, & \text{其它}\end{cases} \tag{2}$$

由定义 5 和定义 6 可知，两条路径的交仍是一条路径(点)，路径与路径集合的交是一个路径(点)集合，因此，对于路径的交或路径与路径集合的交而言，仍然存在可行区域的概念. 在图 2 中，路径 p_1 和 p_2 的交就是点 L，它的可行区域是 $BELN$.

定义 7.　路径集合的可行区域的积 $\Psi(F(P_{\langle s,t\rangle}))$ (Capacity of Feasible Region of a Path Set). 设 $S(C_1,C_2,\cdots,C_K)$ 为从源节点 s 到目的节点 t 上的 QoS 请求空间，$P_{\langle s,t\rangle}=\{p_1,p_2,\cdots,p_N\}$ 是从 s 到 t 的一个路径集合，$F(P_{\langle s,t\rangle})$ 是路径集合 $P_{\langle s,t\rangle}$ 在 S 中的可行区域，则可行区域 $F(P_{\langle s,t\rangle})$ 在 S 中的积记为 $\psi(F(P_{\langle s,t\rangle}))$，定义如下：

(i) 若 $P_{\langle s,t\rangle}=\varnothing$ (空集)，则

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

(ii) 若 $P_{\langle s,t\rangle}$ 只包含一条路径，即 $P_{\langle s,t\rangle}=\{p\}$，则

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

(iii) 若 $P_{\langle s,t\rangle}$ 包含多条路径，任取 $p'\in P_{\langle s,t\rangle}$，记 $P'_{\langle s,t\rangle}=P_{\langle s,t\rangle}-\{p'\}$，则

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

定理 1.　一个路径集合的可行区域的积是唯一的(证明见附录).

定义 8.　最优路径压缩问题 (Optimal Path Reduction Problem, OPR). 设 S 为从源节点 s 到目的节点 t 上的 QoS 请求空间，给定从 s 到 t 的一个路径集合 $P_{\langle s,t\rangle}=\{p_1,p_2,\cdots,p_N\}$，最优路径压缩问题定义为：求得 $P_{\langle s,t\rangle}$ 的一个元素个数为 $L(L\leq N)$ 的子集 $\pi=\{p_{r1},p_{r2},\cdots,p_{rL}\}$，使得

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

定义 7 的直观意义是：积的概念表示了对一个路径集合的可行区域在 QoS 请求空间中所占空间大小的一种度量，而定理 1 说明了这种度量的唯一性. 积越大，则对应的路径集合能够满足不同 QoS 约束的概率越大. 定义 8 指出了最优路径压缩问题的目标就是从给定的 N 条路径中选出 L 条路径，使得该路径子集所对应的可行区域的积最大化，从而能够以最大的概率满足各种不同的 QoS 约束.

4　多约束路径压缩算法

本章针对最优路径压缩(OPR)问题，提出了一系列算法，并对它们进行了分析和对比.

4.1　最优算法 Optimal

最优路径压缩问题的任务是从 N 条路径中选出 L 条，选择方法共有 C_N^L 种. 最优算法穷举所有 C_N^L 种可能的情况，因此可以得到精确最优的结果. 该算法可作为其它近似算法的性能上界.

图 3 是最优算法 Optimal 的主程序框架. 它的主程序是一个递归过程，Optimal($PATH0,M,R$) 的意义是从包含 M 条路径的集合 $PATH0$ 中选出 R 条路径，选择结果保存在 $PATH2$ 中. 进行路径压缩时应首先令 $PATH0=\{p_1,p_2,\cdots,p_N\}$，然后调用 Optimal($PATH0,N,L$)，最后会在集合 $PATH2$

中得到选出的 L 条路径. 图 3 中的算法使用 $a[L]$保存选出的 L 条路径的下标编号. 算法首先从$\{R-1, R, R+1, \cdots, M-1\}$中选择第$(R-1)$条路径的下标编号(第 1,2 行),然后判断 R 是否等于 1(第 3 行). 如果 R 等于 1,说明 $a[L]$已经保存好 L 条路径的下标编号. 这时先将得到的 L 条路径暂存在集合 $PATH1$ 中,然后调用 $Capacity$ 函数计算这 L 条路径的可行区域的积并据此与原先已经选出的结果相比较,更新相应变量,将可行区域的积较大的 L 条路径存入集合 $PATH2$(第 4～12 行). 如果 R 不等于 1,继续递归在余下的 n 条路径中选择 $R-1$ 条(第 13,14 行).

```
Optimal(PATH0,M,R)
1) FOR(n=M-1; n>=R-1; n--)
2)     a[R-1]=n
3)     IF(R==1)
4)         PATH1=∅
5)         FOR(i=0; i<L; i++)
6)             Add PATH0(a[i]) into PATH1
7)         cpty=Capacity(PATH1)
8)         IF(maxCpty<cpty)
9)             maxCpty=cpty
10)            PATH2=∅
11)            FOR(i=0; i<L; i++)
12)                Add PATH0(a[i]) into PATH2
13)    ELSE
14)        Optimal(PATH0,n,R-1)
```

图 3 最优算法 Optimal

Capacity 函数用于计算路径集合 $PATH0$ 的可行区域的积,如图 4 所示. 它也是一个递归过程,大体由 3 部分组成:(1)在 $PATH0$ 中任选一条路径 p_0并计算它的可行区域的积(第 1～5 行);(2)沿着 p_0的各个维度将整个 QoS 请求空间分割成 2^K个小的请求空间,同时将每一条路径也分割到各个小的请求空间内(第 6～25 行);(3)在各个小的请求空间(除了包含 p_0本身的请求空间)内递归计算各路径集合的可行区域的积,将这些积加上 p_0可行区域的积得到最后结果(第 26～30 行). 注意,该算法中的变量 $nSeq(p)$表示对路径 p 所在的小请求空间的编号.

定理 2. 算法 Capacity 在最坏情况下的时间复杂度为 $O\left(\frac{(2^K-1)^{L+1}}{(2^K-2)^2}\right)$(证明见附录).

结合定理 2 可以得到,最优算法 Optimal 在最坏情况下的时间复杂度是 $O\left(C_N^L\frac{(2^K-1)^{L+1}}{(2^K-2)^2}\right)$.

4.2 增量贡献算法 ContriInc

最优算法 Optimal 虽然可以达到最佳性能,但算法复杂度很高(组合数的复杂度). 为了设计复杂度更低的算法,下面提出贡献区域的概念.

```
Capacity(PATH0)
1)  IF(PATH0==∅)
2)      RETURN 0
3)  Arbitrarily select a path p0 from PATH0
4)  Delete p0 from PATH0
5)  p0_Capacity=∏_{l=1}^{K}(C_l(p0)-w_l(p0))
6)  FOR EACH path p in PATH0
7)      nSeq(p)=0
8)  PATH1=∅
9)  FOR(l=1; l<=K; l++)
10)     FOR EACH path p in PATH0
11)         Delete p from PATH0
12)         IF(w_l(p)<w_l(p0))
13)             p_copy1=p
14)             p_copy2=p
15)             w_l(p_copy2)=w_l(p0)
16)             nSeq(p_copy2)+=2^{l-1}
17)             C_l(p_copy1)=w_l(p0)
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
```

图 4 计算路径集合 $PATH0$ 的可行区域的积

定义 9. 贡献区域 $C(p;P_{\langle s,t\rangle})$(Contribution Region). 设 S 为从源节点 s 到目的节点 t 上的 QoS 请求空间,$P_{\langle s,t\rangle}=\{p_1,p_2,\cdots,p_N\}$是从 s 到 t 的一个路径集合,路径 p 也是从 s 到 t 的一条路径,且 $p\notin P_{\langle s,t\rangle}$,$F(p)$和 $F(P_{\langle s,t\rangle})$分别是路径 p 和路径集合 $P_{\langle s,t\rangle}$在 S 中的可行区域,则集合 $C(p;P_{\langle s,t\rangle})=F(p)-F(P_{\langle s,t\rangle})$称为路径 p 相对于路径集合 $P_{\langle s,t\rangle}$的贡献区域.

定义 10. 贡献区域的积 $\delta(C(p;P_{\langle s,t\rangle}))$(Capacity of Contribution Region). 设 $P_{\langle s,t\rangle}=\{p_1,p_2,\cdots,p_N\}$是从 s 到 t 的一个路径集合,路径 p 是从 s 到 t 的一条路径,且 $p\notin P_{\langle s,t\rangle}$,则路径 p 相对于路径集合 $P_{\langle s,t\rangle}$的贡献区域的积定义为 $\delta(C(p;P_{\langle s,t\rangle}))=\psi(F(p))-\psi(F(p\wedge P_{\langle s,t\rangle}))$.

由定义 9 可知,一条路径的贡献区域表示了该路径相对于某个路径集合所起的作用,而定义 10 提出的贡献区域的积正是对这一作用的一种度量. 由定义 10 可以看出,当路径集合 $P_{\langle s,t\rangle}$为空时,路径 p 的贡献区域的积就是它的可行区域的积.

而且，结合定义 7 容易得到如下结论：当 $p \notin P_{\langle s,t\rangle}$ 时，$\psi(F(\{p\} \cup P_{\langle s,t\rangle})) = \psi(F(P_{\langle s,t\rangle})) + \delta(C(p; P_{\langle s,t\rangle}))$. 本文提出一个基于贡献区域的算法：增量贡献算法 ContriInc，如图 5 所示.

由图 5(a)可以看出，算法 ContriInc 的思想是从集合 $PATH0$ 的 N 条路径中依次选出具有最大贡献区域的路径 p，然后将 p 从 $PATH0$ 中删除并加入到 $PATH1$ 中. 这样循环 L 次，就在 $PATH1$ 中得到了选出的 L 条路径. 算法调用了子过程 ContriCapacity，如图 5(b)所示，该子过程的目的在于计算贡献区域的积 $\delta(C(p_0; PATH0))$. 1～5 行实际上是计算 $PATH1 = p \wedge PATH0$，第 6 行计算 $\psi(F(p_0))$，第 7 行计算 $\psi(F(PATH1))$.

```
ContriInc()
1)  PATH0={p1,p2,…,pN}
2)  PATH1=∅
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) 增量贡献算法主程序

```
ContriCapacity(p0,PATH0)
1)  FOR EACH path p in PATH0
2)      FOR(l=1; l<=K; l++)
3)          IF(wl(p)<wl(p0))
4)              wl(p)=wl(p0)
5)      Add p into PATH1
6)  cpty1=∏_{l=1}^{K}(Cl−wl(p0))
7)  cpty2=Capacity(PATH1)
8)  RETURN cpty1−cpty2
```

(b) 子过程：计算贡献区域的积

图 5　增量贡献算法 ContriInc 和子过程 ContriCapacity

在图 5(b)中，ContriCapacity 子过程调用了 Capacity过程(第 7 行). 根据定理 2 可知，增量贡献算法在最坏情况下的复杂度仍与问题规模成指数关系.

4.3　改进的增量贡献算法 ImprovedInc

使得增量贡献算法 ContriInc 复杂度仍然较高的关键是 ContriCapacity 子过程，这是由于它调用了 Capacity 过程. 本节继续对算法进行改进，使它们的计算复杂度控制在多项式范围内.

定理 3.　设 $P_{\langle s,t\rangle,N} = \{p_1, p_2, \cdots, p_N\}$ 是从源节点 s 到目的节点 t 的一个路径集合，$F(p)$ 和 $F(P_{\langle s,t\rangle,N})$ 分别表示路径 p 和路径集合 $P_{\langle s,t\rangle,N}$ 的可行区域，$SUB_l^i(P_{\langle s,t\rangle,N}) \subseteq P_{\langle s,t\rangle,N}$ 且 $SUB_l^i(P_{\langle s,t\rangle,N})$ 中只包含 l 条路径($1 \le l \le N, 1 \le i \le C_N^l$)，则

$$\psi(F(P_{\langle s,t\rangle,N})) = \sum_{l=1}^{N}(-1)^{l-1}\sum_{i=1}^{C_N^l}\psi(F(INTSEC(SUB_l^i(P_{\langle s,t\rangle,N})))) \tag{7}$$

其中 $INTSEC(SUB_l^i(P_{\langle s,t\rangle,N}))$ 表示集合 $SUB_l^i(P_{\langle s,t\rangle,N})$ 中所有 l 条路径的交(证明见附录).

定理 3 给出了计算路径集合的可行区域的积的另一种方案. 式(7)是一个 N 项和的形式，完全按该式进行计算复杂度仍然较高，但我们发现它的每一项对总和的影响依次递减. 因此我们对这一结论做一个近似，记

$$M(S) = \sum_{l=1}^{S}(-1)^{l-1}\sum_{i=1}^{C_N^l}\psi(F(INTSEC(SUB_l^i(P_{\langle s,t\rangle,N})))) \tag{8}$$

很明显，$M(S)$ 是对 $\psi(F(P_{\langle s,t\rangle,N}))$ 的一个近似，S 越大两个值就越接近. 特别是当 $S=N$ 时，有 $M(N) = \psi(F(P_{\langle s,t\rangle,N}))$.

现在我们改写图 4 中的算法 Capacity，使用式(8)来计算可行区域的积，改写后的算法记为 $ImprovedCapacity(PATH0, S)$. 如果在图 5(b)的第 7 行改为调用 ImprovedCapacity，而图 5 中所有算法的其余部分均不变，便得到了增量贡献算法的改进版本，记为 $ImprovedInc(S)$，其中 S 是表示算法精确度的一个参数. 大量的模拟实验表明，当 $S=2$ 时，算法已经能够获得很高的性能.

现在分析当 S 取 2 时改进算法的复杂度. 图 5(a)中的算法在第 6 行调用 ContriCapacity，是复杂度的关键. 在图 5(b)中，当 $PATH1$ 保存有 n 条路径并限制 $S=2$ 时，ContriCapacity 的复杂度可限制在 $O(Kn^2)$. 因此，ImprovedInc 的复杂度为

$$O\left(\sum_{i=0}^{L-1}Ki^2\right) = O(KL(L-1)(2L-1)/6)) = O(KL^3).$$

4.4　算法小结

现在将本文提出的一系列算法子过程之间的调

用关系做一总结，如图 6 所示，图中的箭头表示被调用关系. 值得注意的是，当 ContriCapacity 调用 Capacity 时，得到的是未改进的增量贡献算法，即 4.2 节介绍的 ContriInc 算法；当 ContriCapacity 调用 ImprovedCapacity 时，得到的就是改进后的增量贡献算法，即 4.3 节介绍的 ImprovedInc 算法.

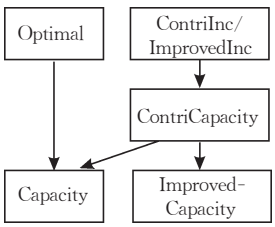


图 6 各算法子过程的调用关系

5 性能模拟

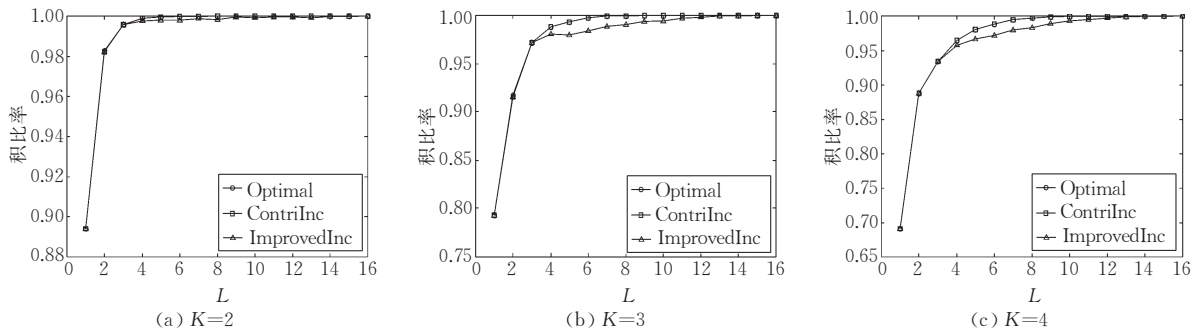
这一部分通过模拟实验对算法的性能进行比较和分析. 模拟实验分为两部分，第一部分对本文提出的几个算法进行比较，第二部分结合现有的 QoSR 算法进行性能评价.

5.1 各路径压缩算法的比较

本节的模拟实验对本文提出的几个算法在性能和运算时间上进行比较，这些算法包括：最优算法 Optimal，增量贡献算法 ContriInc 和改进的增量贡献算法 ImprovedInc.

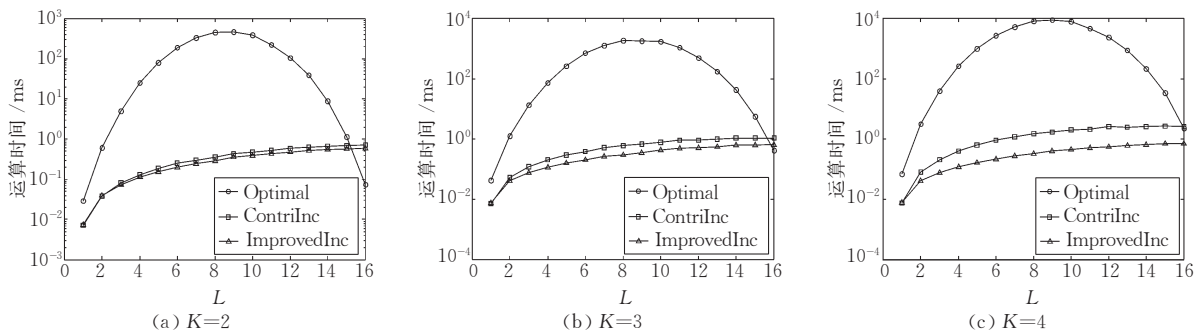
在本节的模拟实验中，我们随机生成 N 条路径，即 $w_l(p_i)\sim Uniform(1,C_l)$，其中 $1\leq l\leq K$，$1\leq i\leq N$. 在本节实验中，我们令 $C_l=1000$，$1\leq l\leq K$. 为减小误差，每个数据都计算 100 次并取平均值.

在图 7 中，我们在 L 变化的情况下比较了各算法的性能. 实验中令 $N=16$，$S=2$，并模拟了 K 分别取 2，3 和 4 三种情况. 为了对各算法的性能进行度量，我们定义了积比率(Capacity Ratio)的概念，它表示某个算法得到的 L 条路径的可行区域的积与所有 N 条路径的可行区域的积的比值. 它应该在[0,1]之间变化，并且算法 Optimal 的积比率始终是所有算法中最高的(性能上界). 从整体来看，算法 ContrInc 几乎可以达到与 Optimal 相同的积比率，而 ImprovedInc 的积比率略低于 ContrInc 和 Optimal. 当 L 从 1 开始增加时，各算法的积比率都急剧增加，到 $L=5$ 或 6 时，各算法的积比率已经非常接近于 1. 图 8 在 L 变化的情况下比较了各算法的运算时间，注意图中 Y 轴使用的是对数坐标. 从总体来看，算法 Optimal 具有最长的运算时间，而且远远高于其它两个算法(特别是当 L 接近 $N/2$ 时)；而 ImprovedInc 具有最短的运算时间.



(a) $K=2$ (b) $K=3$ (c) $K=4$

图 7 各算法的性能比随 L 的变化情况



(a) $K=2$ (b) $K=3$ (c) $K=4$

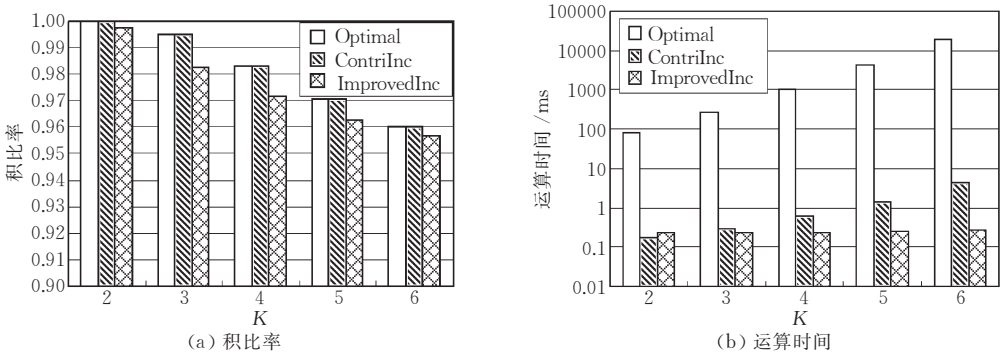
图 8 各算法的运算时间随 L 的变化情况

综合图 7 和图 8 的结果，我们可以看出，ContriInc 和 ImprovedInc 的运算复杂度远远小于最优算法 Optimal，却能够获得接近最优的性能. 当 $L=5$ 或 6 时，各算法的积比率已经非常接近于 1，因此，在

QoS 路由表中为每个目的节点保存 5 或 6 个路径信息就可获得很好的路由性能，而不必保存所有路径.

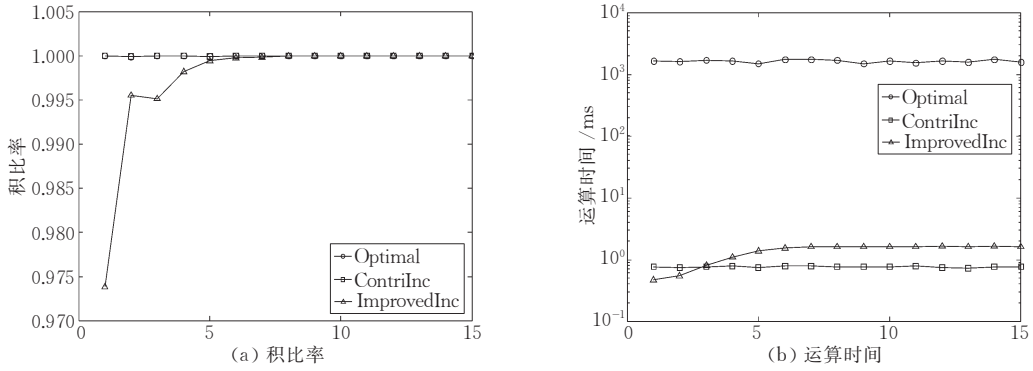
图 9 在 K 变化的情况下比较了各算法的积比率和运算时间，图 9(b)中 Y 轴使用的仍是对数坐标. 实验中令 $N=16, L=5, S=2$. 从图 9(a)中可以看出，各算法的性能随 K 的增大只有小幅度的减低. 而从图 9(b)中可以看出，各算法花费的运算时间却存在巨大的差异，ImprovedInc 的运算时间远远低于另外两个算法. 而且，Optimal 和 ContriInc 的运算时间都随 K 的增大呈指数增长，而 ImprovedInc 的运算时间对 K 的增大并不敏感，几乎维持不变.

图 10 表示了各算法的积比率和运算时间受 S 变化的影响，图 10(b)中 Y 轴使用的仍是对数坐标. 实验中令 $N=16, L=10, K=3$. 由于 Optimal 和 ContriInc 两个算法与 S 无关，因此它们的性能和运算时间基本不随 S 的变化而变化. 从图 10(a)中可以看出，ImprovedInc 的性能随 S 的增大呈上升趋势，但 $S=2$ 时已能获得较高的性能. 从图 10(b)中可以看出，ImprovedInc 的运算时间随 S 的增大存在较快的增长，并最终超过 ContriInc，但当 $S\leq 2$ 时算法 ImprovedInc 具有最低的运算复杂度. 因此，在实际计算中，S 取 2 比较理想.



(a) 积比率　　(b) 运算时间

图 9　各算法的性能比和运算时间随 K 的变化情况



(a) 积比率　　(b) 运算时间

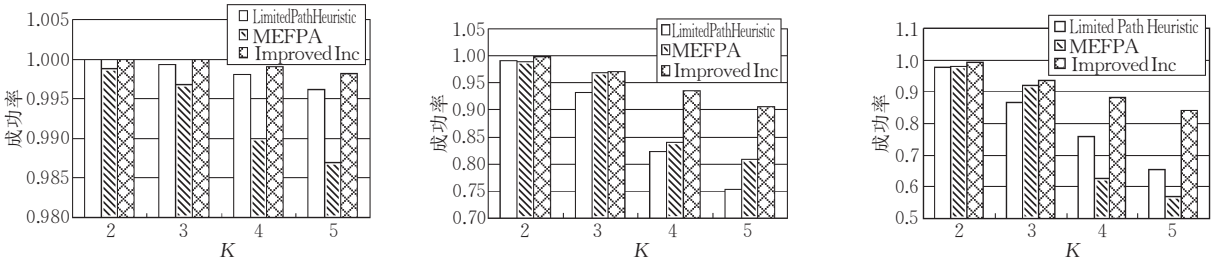
图 10　各算法的性能比和运算时间随 S 的变化情况

5.2　与现有 QoSR 算法的比较

本节将算法 ImprovedInc 同两个现有的 QoSR 算法(即 Limited Path Heuristic[7] 和 MEFPA[4])进行性能上的对比. 在实验中，我们使用 GT-ITM 拓扑生成器[16]分别生成了具有 50，100 和 200 个节点的随机拓扑，链路数分别是 116，374 和 1404. 对于每个生成的拓扑，我们选择所有节点对中距离(跳数)最远的节点对作为源和目的节点，这样选出的源和目的节点之间更可能存在较多的可选路径. 实验时我们先调用 EBFA 得到最优路径(Optimal QoS Path[7])集合，然后使用 ImprovedInc 完成路径压缩. 当然，也可以先调用其它的预计算路由算法(如粒度受限、路径受限、MEFPA 等)计算最优路径集合，本文的路径压缩算法并不依赖于任何一种具体算法. 调用每个算法得到路径选择结果以后，我们随机生成 1000 个 QoSR 请求，然后计算各算法能够满足的请求数目. 我们使用成功率(Success Ratio)来度量每个算法的性能，它定义为某算法能够满足的 QoSR 请求数目与总共生成的 QoSR 请求数目的比值.

图 11 将 ImprovedInc 与现有的 QoSR 算法 Limited Path Heuristic 和 MEFPA 进行了性能比较. 实验中令 ImprovedInc 的参数 $L=5, S=2$，Limited Path Heuristic 的参数 $X=L=5$，而 MEFPA 的参数 b 当 $K=2,3,4,5$ 时分别取 5，3，2，2，以使得

三个算法获得相当的路径选择数目.在图 11(a)所示的 50 个节点的情况中,三个算法性能上没有太大差异,这是因为网络规模还比较小,源和目的节点之间存在的路径数目较少,还没有体现出路径压缩的优势.但在 100 和 200 个节点的情况下,Improved-Inc 的性能明显高于两个传统的 QoSR 算法.



(a) 50 个节点 (b) 100 个节点 (c) 200 个节点

图 11 与现有 QoSR 算法的性能比较

6 结 论

为支持 QoS 路由预计算并尽量减小路由表空间,本文明确定义了最优路径压缩问题(Optimal Path Reduction Problem,OPR),并使用最优算法给出了该问题的性能上界.

为了解决 OPR 问题并克服最优算法复杂度高的缺点,本文提出了基于贡献区域的增量贡献算法,并结合问题的实质对算法进行了改进,进一步降低了计算复杂度.大量的模拟实验表明,这类基于贡献区域的算法能够获得接近最优算法的性能,但极大地降低了问题求解的复杂度.

在实际的网络应用中,本文提出的路径压缩算法可在下列两种场合下使用:(1)当源和目的节点之间存在的最优路径较少时,可以首先计算出所有最优路径,然后针对所有最优路径组成的集合进行路径压缩;(2)当源和目的节点之间存在的最优路径数目过于庞大时,如最优路径的数目与网络规模成指数关系的情况,这时可以首先使用多约束 QoSR 启发式算法计算出最优路径集合的一个子集,然后针对这个子集再进行路径压缩.

参 考 文 献

[1] Korkmaz T, Krunz M. Multi-constrained optimal path selection//Proceedings of the IEEE INFOCOM'01. Alaska, 2001, 2: 834-843

[2] Zhang Xin-Yan, Liu Jiang-Chuan, Li Bo, Yum Tak-Shing Peter. CoolStreaming/DONet: A data-driven overlay network for peer-to-peer live media streaming//Proceedings of the IEEE INFOCOM. Barcelona, 2005, 3: 2102-2111

[3] Li Zhi, Mohapatra Prasant. QRON: QoS-aware routing in overlay networks. IEEE Journal on Selected Areas in Communications, 2004, 22(1): 29-40

[4] Cui Y, Xu K, Wu J P. Precomputation for multi-constrained QoS routing in high-speed networks//Proceedings of the IEEE INFOCOM 2003. San Francisco, CA, 2003

[5] Cui Yong, Wu Jian-Ping, Xu Ke, Xu Ming-Wei. Research on internetwork QoS routing algorithms: A survey. Journal of Software, 2002, 13(11): 2065-2075(in Chinese)

[6] Orda A, Sprintson A. QoS routing: The precomputation perspective//Proceedings of the IEEE INFOCOM'00. Israel, 2000, 1: 128-136

[7] Yuan X, Liu X. Heuristic algorithms for multi-constrained quality of service routing//Proceedings of the IEEE INFOCOM'01. Alaska, 2001, 2: 844-853

[8] Jaffe J M. Algorithms for finding paths with multiple constraints. Networks, 1984, 14(1): 95-116

[9] Wang Z, Crowcroft J. Quality-of-service routing for supporting multimedia applications. IEEE JSAC, 1996, 14(7): 1288-1234

[10] Lorenz H, Orda Ariel, Raz Danny, Shavitt Yuval. Efficient QoS partition and routing of unicast and multicast//Proceedings of the IWQoS 2000. Pittsburgh, 2000: 75-83

[11] Raz D, Shavitt Y, Raz Danny, Shavitt Yuval. Optimal partition of QoS requirements with discrete cost functions//Proceedings of the IEEE INFOCOM'00. Israel, 2000, 2: 613-622

[12] Ma Q, Steenkiste P. Quality-of-service routing with performance guarantees//Proceedings of the 4th International IFIP Workshop on Quality of Service. New York, 1997: 115-126

[13] Pornavalai C, Chakraborty G, Shiratori N. QoS based routing algorithm in integrated services packet networks//Proceedings of the IEEE ICNP'97. Atlanta, GA, 1997

[14] de Neve H, Van Mieghem P. A multiple quality of service routing algorithm for PNNI//Proceedings of the IEEE ATM Workshop Proceedings. Virginia, 1998: 324-328

[15] Van Mieghem P, de Neve H, Kuipers F. Hop-by-hop quality of service routing. Computer Networks, 2001, 37(3): 407-423

[16] Thomas Megan, Zegura Ellen W. Generation and analysis of random graphs to model internetworks. College of Computing, Georgia Tech.: Technical Report: GIT-CC-94-46, 1994

附　录.

定理 1.　一个路径集合的可行区域的积是唯一的.

证明.　路径集合记为 $P_{\langle s,t\rangle}$，设元素个数为 N，下面对 N 用归纳法证明.

当 $N=0$ 时，由式(3)知 $\psi(F(P_{\langle s,t\rangle}))=0$，显然结果是唯一的.

当 $N=1$ 时，由式(4)知 $\psi(F(P_{\langle s,t\rangle}))=\prod_{l=1}^{K}(C_l-w_l(p))$，显然结果是唯一的.

现假设 $N\leq n(n\geq 1)$ 时积是唯一的，下面证明 $N=n+1$ 时积也是唯一的.

这时 $N\geq 2$，$\psi(F(P_{\langle s,t\rangle}))$ 的值要根据式(5)来计算. 任取两条路径 p_{s1}，$p_{s2}\in P_{\langle s,t\rangle}$，在式(5)中，当 $p'=p_{s1}$ 时计算结果记为 $\psi^1(F(P_{\langle s,t\rangle}))$，当 $p'=p_{s2}$ 时，计算结果记为 $\psi^2(F(P_{\langle s,t\rangle}))$. 并记 $P'_{\langle s,t\rangle,s1}=P_{\langle s,t\rangle}-\{p_{s1}\}$，$P'_{\langle s,t\rangle,s2}=P_{\langle s,t\rangle}-\{p_{s2}\}$，则

$$\psi^1(F(P_{\langle s,t\rangle}))=\psi(F(P'_{\langle s,t\rangle,s1}))+\psi(F(\{p_{s1}\}))-\psi(F(p_{s1}\wedge P'_{\langle s,t\rangle,s1})) \quad (附\ 1)$$

$$\psi^2(F(P_{\langle s,t\rangle}))=\psi(F(P'_{\langle s,t\rangle,s2}))+\psi(F(\{p_{s2}\}))-\psi(F(p_{s2}\wedge P'_{\langle s,t\rangle,s2})) \quad (附\ 2)$$

如果 $p_{s1}=p_{s2}$，则 $P'_{\langle s,t\rangle,s1}=P'_{\langle s,t\rangle,s2}$，$p_{s1}\wedge P'_{\langle s,t\rangle,s1}=p_{s2}\wedge P'_{\langle s,t\rangle,s2}$，且 $P'_{\langle s,t\rangle,s1}$，$P'_{\langle s,t\rangle,s2}$，$p_{s1}\wedge P'_{\langle s,t\rangle,s1}$ 和 $p_{s2}\wedge P'_{\langle s,t\rangle,s2}$ 的元素个数为 n，由归纳假设知它们的可行区域的积是唯一的，所以 $\psi^1(F(P_{\langle s,t\rangle}))=\psi^2(F(P_{\langle s,t\rangle}))$.

如果 $p_{s1}\neq p_{s2}$，记 $P'_{\langle s,t\rangle,s1,s2}=P'_{\langle s,t\rangle,s1}-\{p_{s2}\}=P'_{\langle s,t\rangle,s2}-\{p_{s1}\}$，则有

$$\psi(F(P'_{\langle s,t\rangle,s1}))=\psi(F(P'_{\langle s,t\rangle,s1,s2}))+\psi(F(\{p_{s2}\}))-\psi(F(p_{s2}\wedge P'_{\langle s,t\rangle,s1,s2})) \quad (附\ 3)$$

$$\psi(F(P'_{\langle s,t\rangle,s2}))=\psi(F(P'_{\langle s,t\rangle,s1,s2}))+\psi(F(\{p_{s1}\}))-\psi(F(p_{s1}\wedge P'_{\langle s,t\rangle,s1,s2})) \quad (附\ 4)$$

$$\psi(F(p_{s1}\wedge P'_{\langle s,t\rangle,s1}))=\psi(F(p_{s1}\wedge P'_{\langle s,t\rangle,s1,s2}))+\psi(F(\{p_{s1}\wedge p_{s2}\}))-\psi(F(p_{s1}\wedge p_{s2}\wedge P'_{\langle s,t\rangle,s1,s2})) \quad (附\ 5)$$

$$\psi(F(p_{s2}\wedge P'_{\langle s,t\rangle,s2}))=\psi(F(p_{s2}\wedge P'_{\langle s,t\rangle,s1,s2}))+\psi(F(\{p_{s1}\wedge p_{s2}\}))-\psi(F(p_{s1}\wedge p_{s2}\wedge P'_{\langle s,t\rangle,s1,s2})) \quad (附\ 6)$$

将式(附 3)～(附 6)代入式(附 1)，(附 2)，得

$$\psi^1(F(P_{\langle s,t\rangle}))=\psi(F(P'_{\langle s,t\rangle,s1,s2}))+\psi(F(\{p_{s1}\}))+\psi(F(\{p_{s2}\}))-\psi(F(p_{s1}\wedge P'_{\langle s,t\rangle,s1,s2}))-\psi(F(p_{s2}\wedge P'_{\langle s,t\rangle,s1,s2}))-\psi(F(\{p_{s1}\wedge p_{s2}\}))+\psi(F(p_{s1}\wedge p_{s2}\wedge P'_{\langle s,t\rangle,s1,s2})) \quad (附\ 7)$$

$$\psi^2(F(P_{\langle s,t\rangle}))=\psi(F(P'_{\langle s,t\rangle,s1,s2}))+\psi(F(\{p_{s1}\}))+\psi(F(\{p_{s2}\}))-\psi(F(p_{s1}\wedge P'_{\langle s,t\rangle,s1,s2}))-\psi(F(p_{s2}\wedge P'_{\langle s,t\rangle,s1,s2}))-\psi(F(\{p_{s1}\wedge p_{s2}\}))+\psi(F(p_{s1}\wedge p_{s2}\wedge P'_{\langle s,t\rangle,s1,s2})) \quad (附\ 8)$$

由式(附 7)，(附 8)可知，两表达式完全相同，且由归纳假设可知 $\psi(F(P'_{\langle s,t\rangle,s1,s2}))$，$\psi(F(p_{s1}\wedge P'_{\langle s,t\rangle,s1,s2}))$，$\psi(F(p_{s2}\wedge P'_{\langle s,t\rangle,s1,s2}))$ 和 $\psi(F(p_{s1}\wedge p_{s2}\wedge P'_{\langle s,t\rangle,s1,s2}))$ 均有唯一值，所以 $\psi^1(F(P_{\langle s,t\rangle}))=\psi^2(F(P_{\langle s,t\rangle}))$.

综上，对于任意的 N，一个路径集合的可行区域的积都是唯一的.

定理 2.　算法 Capacity($PATH0$)在最坏情况下的时间复杂度为 $O\left(\frac{(2^K-1)^{L+1}}{(2^K-2)^2}\right)$.

证明.　设 $C(L)$ 为 $PATH0$ 包含 L 条路径时的所需的比较次数，则 $C(0)=0$. 在最坏情况下，每条路径都被分割成 2^K 条新路径，所以从 L 条路径中选出一条再进行分割后得到 $2^K(L-1)$ 条新路径. 这 $2^K(L-1)$ 条新路径被分到 2^K 个小的请求空间中，每个请求空间分到 $(L-1)$ 条路径. 同时，考虑算法 9～22 行的循环，在最坏情况下每经过一次外层循环，路径数量变为原来的两倍，因此总共比较次数为 $(L-1)+2^1(L-1)+2^2(L-1)+\cdots+2^{K-1}(L-1)=(2^K-1)(L-1)$. 至此，我们可以列出递推公式如下

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

所以

$$C(L)=\frac{(2^K-1)^{L+1}}{(2^K-2)^2}-\frac{2^K-1}{2^K-2}L-\frac{2^K-1}{(2^K-2)^2} \quad (附\ 9)$$

在式(附 9)中，第 1 项 $\frac{(2^K-1)^{L+1}}{(2^K-2)^2}$ 对于比较次数 $C(L)$ 起决定作用，因此算法 Capacity($PATH0$)在最坏情况下的时间复杂度为 $O\left(\frac{(2^K-1)^{L+1}}{(2^K-2)^2}\right)$.

定理 3.　设 $P_{\langle s,t\rangle,N}=\{p_1,p_2,\cdots,p_N\}$ 是从源节点 s 到目的节点 t 的一个路径集合，$F(p)$ 和 $F(P_{\langle s,t\rangle,N})$ 分别表示路径 p 和路径集合 $P_{\langle s,t\rangle,N}$ 的可行区域，$SUB_l^i(P_{\langle s,t\rangle,N})\subseteq P_{\langle s,t\rangle,N}$ 且 $SUB_l^i(P_{\langle s,t\rangle,N})$ 中只包含 l 条路径 $(1\leq l\leq N,1\leq i\leq C_N^l)$，则

$$\psi(F(P_{\langle s,t\rangle,N}))=\sum_{l=1}^{N}(-1)^{l-1}\sum_{i=1}^{C_N^l}\psi(F(INTSEC(SUB_l^i(P_{\langle s,t\rangle,N})))) \quad (附\ 10)$$

其中 $INTSEC(SUB_l^i(P_{\langle s,t\rangle,N}))$ 表示集合 $SUB_l^i(P_{\langle s,t\rangle,N})$ 中所有 l 条路径的交.

证明.　对 N 使用数学归纳法. 首先，当 $N=1$ 时，式(附 10)左边 $=\psi(F(P_{\langle s,t\rangle,1}))=\psi(F(p_1))$；而式(附 10)右边 $=(-1)^{1-1}\psi(F(INTSEC(SUB_1^1(P_{\langle s,t\rangle,1}))))=\psi(F(p_1))$. 左边=右边，式(附 10)成立.

现假设式(附 10)对 N 成立，下面证明它对于 $(N+1)$ 也成立.

$$\psi(F(P_{\langle s,t\rangle,N+1}))=$$
$$\psi(F(P_{\langle s,t\rangle,N}))+\psi(F(\{p_{N+1}\}))-\psi(F(p_{N+1}\wedge P_{\langle s,t\rangle,N}))$$
（根据定义 7）
$$=\sum_{l=1}^{N}(-1)^{l-1}\sum_{i=1}^{C_N^l}\psi(F(INTSEC(SUB_l^i(P_{\langle s,t\rangle,N}))))+$$
$$\psi(F(\{p_{N+1}\}))+$$
$$\sum_{l=1}^{N}(-1)^{l}\sum_{i=1}^{C_N^l}\psi(F(INTSEC(SUB_l^i(p_{N+1}\wedge P_{\langle s,t\rangle,N})))),$$
（根据归纳假设）

而

$$\sum_{l=1}^{N+1}(-1)^{l-1}\sum_{i=1}^{C_{N+1}^l}\psi(F(INTSEC(SUB_l^i(P_{\langle s,t\rangle,N+1}))))$$
$$=\sum_{l=1}^{N}(-1)^{l-1}\sum_{i=1}^{C_{N+1}^l}\psi(F(INTSEC(SUB_l^i(P_{\langle s,t\rangle,N+1}))))+$$
$$(-1)^N\psi(F(INTSEC(SUB_{N+1}^1(P_{\langle s,t\rangle,N+1}))))$$
$$=\sum_{i=1}^{N+1}\psi(F(INTSEC(SUB_1^i(P_{\langle s,t\rangle,N+1}))))+$$
$$\sum_{l=2}^{N}(-1)^{l-1}\sum_{i=1}^{C_{N+1}^l}\psi(F(INTSEC(SUB_l^i(P_{\langle s,t\rangle,N+1}))))+$$
$$(-1)^N\psi(F(INTSEC(P_{\langle s,t\rangle,N+1})))$$
$$=\sum_{i=1}^{N}\psi(F(INTSEC(SUB_1^i(P_{\langle s,t\rangle,N}))))+\psi(F(\{p_{N+1}\}))+$$
$$\sum_{l=2}^{N}(-1)^{l-1}\sum_{i=1}^{C_N^l}\psi(F(INTSEC(SUB_l^i(P_{\langle s,t\rangle,N}))))+$$
$$\sum_{l=2}^{N}(-1)^{l-1}\sum_{i=1}^{C_N^{l-1}}\psi(F(INTSEC(SUB_{l-1}^i(p_{N+1}\wedge P_{\langle s,t\rangle,N}))))+$$
$$(-1)^N\psi(F(INTSEC(P_{\langle s,t\rangle,N+1})))$$
$$=\sum_{l=1}^{N}(-1)^{l-1}\sum_{i=1}^{C_N^l}\psi(F(INTSEC(SUB_l^i(P_{\langle s,t\rangle,N}))))+$$
$$\psi(F(\{p_{N+1}\}))+$$
$$\sum_{l=2}^{N+1}(-1)^{l-1}\sum_{i=1}^{C_N^{l-1}}\psi(F(INTSEC(SUB_{l-1}^i(p_{N+1}\wedge P_{\langle s,t\rangle,N}))))$$
$$=\sum_{l=1}^{N}(-1)^{l-1}\sum_{i=1}^{C_N^l}\psi(F(INTSEC(SUB_l^i(P_{\langle s,t\rangle,N}))))+$$
$$\psi(F(\{p_{N+1}\}))+$$
$$\sum_{l=1}^{N}(-1)^{l}\sum_{i=1}^{C_N^l}\psi(F(INTSEC(SUB_l^i(p_{N+1}\wedge P_{\langle s,t\rangle,N})))),$$

所以

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

即式(附 10)对$(N+1)$也成立. 证毕.

ZHAO You-Jian, born in 1969, Ph. D., associate professor. His research interests include computer network architecture, high-speed computer network devices, etc.

ZHANG Tie-Lei, born in 1981, M. S. candidate. His research interests include quality-of-service routing and algorithms.

CUI Yong, born in 1976, Ph. D., assistant professor. His research interests include computer network architecture, quality-of-service control, routing algorithms and evaluation.

Background

Quality-of-service routing (QoSR) is regarded as a promising solution to support flexible QoS-oriented services. Unlike the traditional routing, QoSR tries to meet multiple constraints. Due to the NP-completeness of the multi-constrained routing, researchers in this field have designed many effective heuristics to find a feasible path or several feasible ones in a given network. This paper steps further and proposes an algorithm to reduce the number of the selected paths to an acceptable level and at the same time to maximize routing success ratio.

This work is supported by the National Natural Science Foundation of China (Nos. 90604029 and 60403035) and the National Basic Research Program (973 Program) of China (No. 2003CB314801). These research programs aim to design and establish a novel network architecture. This architecture should be provided with plenty of attracting features, such as scalability, tolerance of failure, self-configuration, supporting QoS-oriented services, etc. This paper focuses on the problem of QoS control and routing. The work presented in this paper can be used to provide theoretical support for the large-scale deployment of QoS routing in the high-speed core networks.