性能可调的启发式多约束路由算法

崔 勇,徐 恪,吴建平
(清华大学计算机科学与技术系,北京 100084)

摘 要: 作为下一代互联网的核心问题之一,多约束的服务质量路由(QoSR)用来寻找一条同时满足多个约束条件的可行路径.由于 QoSR 具有 NPC 的复杂度,为此我们结合线性、非线性能量函数将多个 QoS 度量转化成单一能量值,设计了可调节的启发式算法 BFS_MCP.该算法将深度可调的广度优先搜索策略引入传统 Dijkstra 算法中,使它能够随路由器 CPU 负载和实际网络规模而实时调节算法的运行时间,因而 BFS_MCP 算法具有广泛的适应性.此外,广泛深入的实验结果表明,广度优先的搜索策略能够极大地提高算法性能.

关键词: 能量函数; QoS 路由; 可调算法; 多约束; 性能评价
中图分类号: TP393.01 **文献标识码:** A **文章编号:** 0372-2112 (2002) 12A-1968-05

An Adjustable Heuristic Algorithm for Multi-Constrained Routing Problem

CUI Yong, XU Ke, WU Jian-ping
(*Department of Computer Science, Tsinghua University, Beijing* 100084, *China*)

Abstract: As a challenging problem of the upcoming next-generation networks, multi-constrained quality-of-service routing (QoSR) is to find a feasible path satisfying multiple constraints simultaneously. For the NP complete complexity of QoSR, we propose an adjustable heuristic BFS_MCP based on converting multiple weights to a single metric with linear and non-linear energy functions. Bringing the breadth-first search with the adjustable depth to the standard Dijkstra's algorithm, BFS_MCP can adjust its computation complexity according to the CPU load on a router in real time. Therefore, it has an extensive adaptability. Furthermore, extensive simulations show that the breadth-first search increases the performance greatly.

Key words: energy function; QoS routing; adjustable heuristic; multiple constraints; performance evaluation

1 引言

如何为应用提供不同的服务质量(QoS)保证是互联网络面临的一个重要难题[1,2],而服务质量路由(QoSR)则是其中的一个核心技术和热点问题[3~5].通常 QoS 约束可分为链路约束和路径约束.其中,链路约束可以转化为对整个路径上瓶颈链路的约束,如带宽;而路径约束则是对组成端到端路径上的所有链路的约束,如延迟.由于对链路约束,可以通过剪枝预先除去不符合要求的链路,从而保证在剩余子图中求得的路径满足链路约束,因此本文主要考虑多重(k > =2)路径约束的情况.由于多约束的 QoSR 是 NP 完全问题,为此研究人员设计了很多启发式算法.但这些算法往往具有很大的局限性[1]:(1)计算复杂度过高、无法应用到实际环境中;(2)算法性能太低,找不到实际存在的可行路径;(3)算法只能针对某些特殊情况,不具有普适性.本文提出一种基于路径评价的可调算法,首先对每个节点标号,然后再根据对这些标号的评估来计算可行路径.计算过程中,算法的性能能够随 CPU 负载和网络规模来实时调节,从而达到满意的效果.

2 算法理论基础

用有向图 $G(V,E)$表示一个网络.其中 V 为节点集,元素 $v\in V$ 称为图 G 的一个顶点,代表网络中具有路由能力的节点,如路由器;E 为弧集,元素 $e_{ij}\in E$ 记为 $e=v_i\rightarrow v_j$ 称为图 G 的一条边(弧),代表网络中的一条链路.在 QoSR 中给每个链路 e 关联上一组相互无关的权值$(w_1(e),w_2(e),\cdots,w_k(e))$称为链路 e 的 QoS 度量,简写为 $w(e)$.其中对 $1\leqslant l\leqslant k$,路径约束类型的度量 $w_l(e)\in R^+$ 满足可加性[6],即对路径 $p=v_0\rightarrow v_1\rightarrow\cdots\rightarrow v_n$,有 $w_l(p)=\sum_{i=1}^{n}w_l(v_{i-1}\rightarrow v_i)$.

定义 1 多约束路径 MCP

对于给定的有向图 $G(V,E)$,包含源节点 s、目标节点 t 和 $k\geqslant 2$ 重权值 $w_k(e)\in R^+$,以及约束向量 $c=(c_1,c_2,\cdots,c_k)$,从 s 到 t 的路径 p 称为多约束路径,若 $w_l(p)\leqslant c_l(1\leqslant l\leqslant k)$,简写为 $w(p)\leqslant c$.

对于给定的 QoS 请求,QoSR 的主要任务就是在当前的网络状态下寻找满足要求的路径.Dijkstra 给出了单一度量下计

收稿日期:2002-05-15;修回日期:2002-08-19
基金项目:国家"八六三"项目(No.2002AA103067);国家自然科学基金(No.69725003,90104002)

算最短路径树(SPT)的算法[7],具有较低的算法复杂度.然而对于 MCP 问题涉及到同时考虑多种度量,因此导致问题的复杂度为 NPC 而无法直接使用原有算法.一种可能的思路是将多种度量转化为单一度量,以便使用原有算法.

定义 2　称 $g_\lambda(p)=\sum_{l=1}^{k}(w_l(p)/c_l)^\lambda$ 为路径 p 的能量函数,表示该路径的耗费值,其中 c 为特定 QoS 请求的约束.

下面,我们分别从线性、非线性能量函数两个角度,讨论能量函数与求解最小能量路径的关系,以及最小能量路径与求解 MCP 问题的关系.

定理 1　以能量函数 $g_1(p)=\sum_{l=1}^{k}(w_l(p)/c_l)$ 为关键字,可以使用 Dijkstra 算法建立以节点 s 为根的最小能量树 $T(g)$,满足沿着 $T(g)$ 从 s 到任意节点 t 的路径 p_T 有 $g_1(p_T)=\min_{p(s,t)\in G}g_1(p(s,t))$.

证明　因为 $g_1(p)$ 为线性函数,满足

$$\begin{aligned}g_1(p)&=g_1(e_1+\cdots+e_n)\\&=\sum_{l=1}^{k}w_l(e_1+\cdots+e_n)/c_l\\&=\sum_{l=1}^{k}w_l(e_1)/c_l+\cdots+\sum_{l=1}^{k}w_l(e_n)/c_l\\&=g_1(e_1)+\cdots+g_1(e_n).\end{aligned}$$

所以可以首先算出每个链路的能量值 $g_1(e)$,然后以 $g_1(e)$ 为关键字代替原 Dijkstra 算法中的链路花费,使用原有算法以 s 为源建立 SPT.由于原算法能够保证从 s 到任意节点沿着该 SPT 的路径具有最小的花费,因此以 $g_1(e)$ 为关键字的算法保证 p_T 为从 s 到 t 的具有最小能量(关于 g_1)的路径.

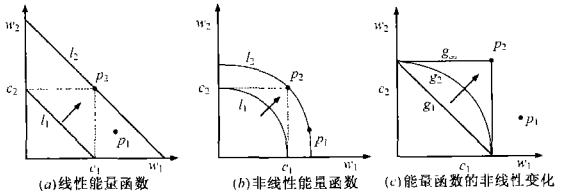


图 1　最小能量路径满足 *MCP* 的可能性

对于非线性的能量函数 $g_{\lambda\neq1}(p)=\sum_{l=1}^{k}(w_l(p)/c_l)^\lambda$ 来说,并不能保证在多项式时间内找到一条具有最小能量值的路径.因此,虽然在寻找最小能量路径时,线性函数具有很好的性质;然而在使用最小能量路径求解 MCP 问题时,非线性函数却有很大优势.如图 1 所示,给定约束条件 (c_1,c_2),并存在可行路径 p_2 满足约束.如图 1(a)所示,当使用线性函数 g_1 计算的最小能量路径时,p_1 成为最小能量路径,然而 p_1 并不满足约束.而如图 1(b)所示,当使用非线性函数 g_2 时,这种最小能量路径不满足约束的可能性降低了.图 1(c)表明了随着能量函数的非线性变化,最小能量路径与可行路径之间的关系,表明 λ 越大,最小能量路径满足约束的可能性也越大.如果进一步使用 g_∞,则可以保证最小能量路径一定能够满足约束.我们所设计的广度优先算法就是基于这种思想,使用启发式算法寻找具有最小能量 g_∞ 的路径.

3　广度优先的启发式算法

3.1　算法思想

一种很自然的想法是使用原有 Dijkstra 算法在每次将一个节点加入到部分建好的 SPT 时,不仅考虑这一步如何处理最优,在处理这一步之前也同时更深层次的考虑后续几步,从而获得一个综合评价较好的方案,然后再根据这个综合方案来进行这一步的处理.这样,我们在每次加入新节点时,采用广度优先的搜索方式,结合对该节点的部分子孙节点的考虑进行判优,从而决定哪个节点是这个新加入节点的父节点.

3.2　算法描述

基于上述广度优先搜索的思想和 Dijkstra 算法,我们针对多约束的 QoSR 问题设计了启发式算法 BFS_MCP,如图 2 所示.算法输入图 G 和给定的 QoS 请求,其源节点为 s、目标节点为 t 和约束向量为 c,其中 $c=(c_1,c_2,\cdots,c_k)$ 为 k 重约束.BFS_MCP 算法首先从目标节点 t 到源节点 s 对网络中的所有节点反向标号,标号过程使用线性函数 g_1 建立小能量树 $SPT_{\lambda=1}$,从而保证每个节点 u 沿着该树到达节点 t 具有最小能量 $Gr[u]$(关于 g_1).每个节点 u 的标号记录了沿着这条路径到达目标节点 t 的 k 重权值,从而在正向广度搜索时,可以参考这些权值.标号过程完成后,BFS_MCP 算法使用非线性函数 g_∞ 计算从 s 到 t 的正向最小能量路径.其中,我们对 Dijkstra 算法的扩展集中体现在 Relax 松弛部分:使用广度优先的搜索方式并结合对反向标号的考虑.如果从 s 到 t 的正向最小能量路径具有的 K 维度量 $d[t]$ 满足约束 c,则返回该路径;否则拒绝该请求或者进行 QoS 协商.算法中,各个符号所表示的意义如表 1 所示.

```
Reverse_Relax(u,v)
IF Gr[u] > Σ_{k=1}^{K} r_R[v] + w_k(u,v)/c_k THEN
   Gr[u]: = Σ_{k=1}^{K} r_R[v] + w_k(u,v)/c_k
   r[u]: = r[v] + w(u,v)
   ∏r[v]: = u
ENDIF
BFS_MCP(G=(V,E),s,t,c,H)
   Reverse_Dijkstra(G,t);
   IF Gr[s] > K THEN
      RETURN failure
   ENDIF
   BFS_Dijkstra(G,s,h);
   IF d[t] < = c THEN
      RETURN this path
   ENDIF
   RETURN failure
```

图 2　BFS_MCP 算法伪码描述

原有 Dijkstra 算法包括以下几个步骤:初始化、寻找最优节点 u(Extract)、将最优节点 u 加入到部分建成的 SPT 中、松弛节点 u(Relax).其中,松弛节点 u 是依次检查 u 的所有不在 SPT 中的邻居节点 v,并根据花费的大小,有选择的更改 v 的父节点以及到达节点 v 的花费值[7],从而使 v 沿着其父节点到达 SPT 的根节点路径最短.在 BFS_MCP 算法中分别使用了

Reverse_Dijkstra 函数和 BFS_Dijkstra 函数.这两个函数都是在原有 Dijkstra 算法基础上的扩展,并将原来的花费改成了能量值.Reverse_Dijkstra 使用 Reverse_Relax 函数取代原来 Dijkstra 算法中松弛节点 u 的函数;BFS_Dijkstra 使用 BFS_Relax 函数来取代原来的松弛节点 u 的函数,并引入了广度优先的搜索策略.其中在 Reverse_Relax 函数中,使用 g_1 函数算出 Gr[u]作为原有 Dijkstra 算法的关键字,计算反向最小能量树 $SPT_{\lambda=1}$.

表 1 算法中符号的意义

算法中使用的符号	符号所代表的意义
u	任一中间节点
Gr[u]	*Reverse_Dijkstra* 对节点 u 的能量值(关于 g_1)
$r[u]=(r_1[u],\cdots,r_K[u])$	沿着反向小能量树 $SPT_{\lambda=1}$从 t 到 u 的度量
v 节点	节点 u 的子节点
||r[v]	节点 v 的前驱节点
d[u]	沿着正向小能量树 $SPT_{\lambda=\infty}$从 s 到 u 的度量
x	队列中的一个元素,包括该元素对应的当前节点 x.v、节点 v 的前驱节点 x.u、v 所在的深度 x.h 和 v 所对应的度量 x.w

```
BFS_Relax(u, v, H)
minVal = INFINITY
x.u = u      /* precedent node */
x.v = v      /* present node */
x.h = 0      /* depth */
x.w = d(u) + w(u, v)
  initialize(stack)
  push(stack, x)
  WHILE(stack not empty)
    x: = pop(stack)
    value: = max_{k=1}^{K}(x.w_k + r_k(x.v))c_k
    IF minVal > value THEN
      minVal: = value
    IF x.w_k ≤ c_k for each k THEN
      RETURN minVal
    IF x.h < H THEN
      FOR each n in neighbor_set of x
       IF n is not in SPT THEN
         y.u: = x.v
         y.v: = n
         y.h: = x.h + 1
         y.w: = x.w + w(y.u, y.v)
         push(stack, y)
```

图 3 Width_First_Relax 伪码描述

BFS_Relax 函数的伪码描述如图 3 所示.该函数首先初始化(第 1-6 行),然后将 x 放入堆栈(第 7 行),进而当堆栈不空时,通过广度优先的循环来寻找节点 v 的子孙节点所具有的最小能量值(g_∞)(第 8-22 行).WHILE 循环中,首先取出堆栈中的栈顶元素 x(第 9 行),然后计算元素 x 对应的节点 $x.v$ 所具有的能量值(第 10 行),而将本次广度优先搜索中找到的最小能量值记录在 min*Val* 中(第 11-12 行).如果发现 $x.v$ 所对应的路径能够满足约束 c,则返回目前所找到的最小能量(第 13-14 行);否则在满足深度小于 H 的情况下(第 15 行),将 $x.v$ 的不在当前 SPT 中的所有子节点放入堆栈(第 16-22 行).

3.3 算法分析

BFS_MCP 算法使用基于线性能量函数 g_1 的反向标号(Reverse_Dijkstra)和基于非线性能量函数 g_∞ 的正向计算(BFS_Dijkstra),并在正向计算中的 Dijkstra 算法松弛节点过程中,引入了可调节的广度优先搜索策略.用 H 表示广度优先所搜索的深度,则当 $H=0$ 时,BFS_MCP 算法退化为 H_MCOP 算法(不考虑优化花费的情况).由于基于能量函数 g_1 和 g_∞ 的操作都满足保序性,因此只要网络状态信息一致,各节点所计算的路由表不会构成回路[8].

现计算求解 k 约束问题时 BFS_MCP 算法的时间复杂度.设具有 k 种度量的网络 $G(V,E)$中,$m=|E|$为网络的边数、$n=|V|$为节点数、$B=2m/n$ 为节点的平均度数.由于使用堆排序的 Dijkstra 算法其时间复杂度为 $O(m+n\log n)$,因此我们可以通过预处理计算每个链路的能量值(见定理 1 及其证明),得到 Reverse_Dijkstra 函数的时间复杂度为 $O((k+1)m+n\log n)$.由于 BFS_Dijkstra 函数需要每一次寻找最优节点和松弛节点的过程中计算.因此 BFS_Dijkstra 函数的时间复杂度为 $O(k(mB^H+n\log n))$.这样,BFS_MCP 总的时间复杂度为 $O(kmB^H+(k+1)m+kn\log n+n\log n)=O(kmB^H+kn\log n)$.当 $H=0$ 时,算法复杂度为 $O(km+kn\log n)$,为原有 Dijkstra 算法的 k 倍.

4 算法性能评价

性能评价部分主要包括两部分内容:(1)QoS 请求中约束条件的分布与算法性能之间的关系;(2)算法性能与搜索深度 H 的关系.其中,当 $H=0$ 时,本文所设计的 BFS_MCP 算法将退化为 H_MCOP 算法[6].在评价上述关系时,我们基于个节点的完全随机拓扑图[9],为每个链路产生了在[1,1000]区间内均匀分布的 K 种度量 $w_k(e)$,且相互无关.我们分别模拟了网络节点数为 50、100、200 和 500 的情况,并对每种情况产生了 10 个拓扑图,在每个拓扑图上随机选取 100 次 QoS 请求的源-目的节点对(一个节点可能被选取多次),每个源节点使用 BFS_MCP 算法计算最小能量路径.算法评价中,首先对这 100 个路由请求计算其成功率 SR(Success Ratio),然后再通过对 10 个同类拓扑图所对应的 10 个成功率的统计,得到其均值 SR.

4.1 两约束算法比较

在 QoSR 算法性能评价中,通常使用路由成功率作为评价算法性能的标准,即算法能够找到可行路径的请求数目与所模拟的总请求数目的比值.我们注意到即便是对同一个网络和 QoS 请求的源目的对集$\{(s,t)\}$,只要其约束条件的产生遵从不同的分布或不同的范围,算法的性能也大相径庭,这也是导致不同文献中的算法评价所给出的数据无法直接比较的

主要原因.然而由于 Internet 缺少典型结构[10],对将来 QoS 请求的约束条件更缺乏足够的认识,因此很难给出合理的模型和分布.实际中很多 QoS 业务对路径上多种度量的关心程度是不同的,例如为了提高文件传输的性能,通常认为丢失率的重要性比延迟大若干倍.

为此,我们在归一化链路度量的基础上,对给定源目的对(s,t)的 QoS 请求,设计了权重比例仿真法来产生其约束.首先令每个 QoS 业务对路径度量有一个关心系数 $a=(a_1,\cdots,a_K)$,其归一化的表示 $a_l/(a_1+a_2)$代表该 QoS 业务对 w_l 的关心程度.基于这种思想,我们使用上面所述的线性能量函数 $g'_1(p)=\sum_{l=1}^{k}(a_l w_l(p))$来构造 QoS 请求:对给定的源目的对$(s,t)$,使用 Dijkstra 算法计算出的具有最小能量值的路径 $p(s,t)$(定理 1),然后令该路径所对应的度量 $w(p(s,t))$为相应的 QoS 约束.由于这样构造的 QoS 请求必然具有可行路径,因此可以将"路由成功率"扩展到对算法绝对性能的评价中.

在二维情况下,我们可以简化为对给定的 $a_1\in[0,1]$,$a_2=1-a_1$.图 4 给出了 $K=2$ 不同的约束对算法性能的影响,其中横坐标为特定的 a_1,而纵坐标 SR 为路由成功率表示路由算法的性能,H 代表 BFS_MCP 算法搜索的深度.从图示数据可见,当约束分布条件 a_1 与 a_2 的值很接近时,非广度优先搜索算法($H=0$)能够获得很高的性能.这实际上是因为在反向计算 $SPT_{\lambda=1}$时,认为两个权值具有相同的重要性造成的.而当约束分布条件 a_1 与 a_2 的值不接近时,非广度优先搜索算法($H=0$)就无法获得很高的性能了.这是因为由于应用业务对两个权值具有不同重要程度的约束,而反向计算 $SPT_{\lambda=1}$时却将两个权值同等看待造成了.在这种情况下,广度优先的搜索方式就起到了很大作用.此外,随着网络规模 N 的增大,非广度优先搜索算法($H=0$)的性能明显降低;而随着 H 的增加,网络规模对算法性能的影响逐步降低.

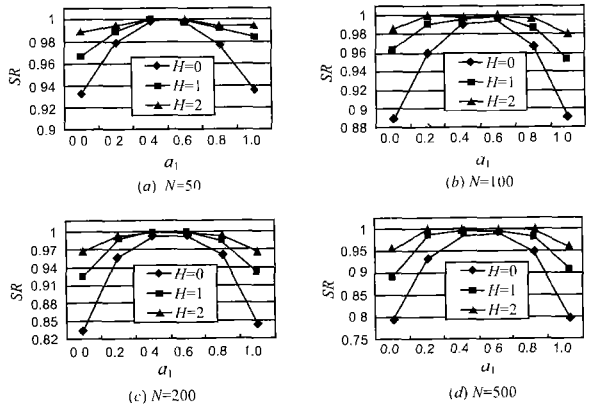


图 4　QoS 约束的分布对算法性能的影响

4.2　多约束算法比较

为了在多约束($k>2$)的情况下研究搜索深度与算法性能之间的关系,我们针对每个源目的对(s,t)产生 QoS 约束条件时,首先取随机数 $b_l\sim\text{uniform}(0,1)(l=1,2,\cdots,k)$,然后令 $a_l=b_l/\sum_{l=1}^{k}b_l$,从而再以 $g'_1(p)=\sum_{l=1}^{k}(a_l w_l(p))$为关键字构造最小能量路径 p,然后以度量 $w(p)$为 QoS 请求的约束条件,即 $c(s,t)=w(p_i)$.图 5 给出了多约束条件对算法性能的影响,横坐标为 QoS 度量的个数 K,H 代表 BFS_MCP 算法搜索的深度.随着约束个数的增加,非广度优先搜索算法($H=0$)性能下降很快;而 BFS_MCP 依然能够保持很高的性能,并且对 QoS 约束个数的敏感性小.此外,BFS_MCP 在多约束情况下对网络规模 N 具有良好的可扩展性.因此 BFS_MCP 能够适应具有多重 QoS 度量的下一代互联网络.

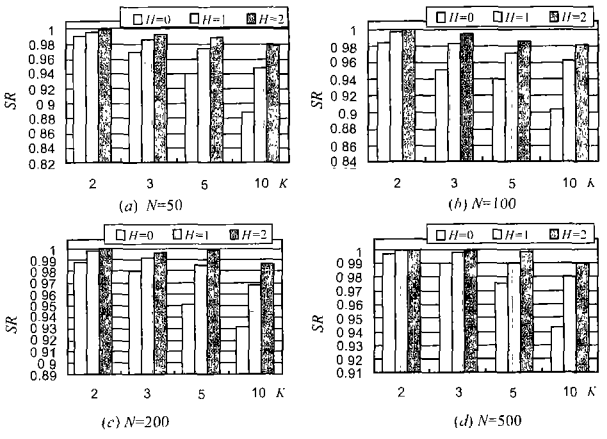


图 5　K 约束下的算法性能

5　结论

由于 QoSR 是 NPC 问题,至今尚无有效算法.本文提出了一种可调节的广度优先启发式算法 BFS_MCP.该算法首先使用线性能量函数对每个节点标号,然后使用非线性函数以广度优先的启发式算法寻找可行路径.在算法性能评价中,我们注意到即便是对同一个网络,只要产生的 QoS 请求其约束条件遵从不同的分布,算法的性能将受到很大影响.为此,我们设计了权重比例仿真法来产生 QoS 请求,不但所仿真的 QoS 请求在一定意义下能够代表实际应用业务的要求,而且能够保证所产生的 QoS 请求必然具有可行路径.基于权重比例仿真法,我们将原来只能用于算法相对性能比较的路由成功率推广到算法的绝对性能评价中.实验结果表明,BFS_MCP 算法对网络规模和约束个数都具有很好的可扩展性.此外,该算法还可根据路由器负载情况进行实时调节:在轻负载情况下能够进一步提高路由性能,而在重负载下能够减小算法的执行时间.

参考文献:

[1] 崔勇,吴建平,徐恪,等.互联网络服务质量路由算法研究综述[J].软件学报,2002,13(11):2065-2076.

[2] Xiao X, Ni L M. Internet QoS: A big picture[J]. IEEE Network, 1999, 13(2):8-18.

[3] Oliveira J C, Akyildiz I F, Uhl G. A new preemption policy for diffserv-aware traffic engineering to minimize rerouting[A]. IEEE INFOCOM'02[C]. New York: IEEE Communication Society, 2002.695-704.

[4] Wang J, Nahrstedt K. Hop-by-hop routing algorithms for premium-class traffic in diffserv networks[A]. IEEE INFOCOM'02[C]. New York:

IEEE Communication Society, 2002. 704 – 714.

[5] Feng G, Douligeris C, Makki K., Pissinou N. Performance evaluation of delay-constrained least-cost QoS routing algorithms based on linear and nonlinear Lagrange relaxation [A]. IEEE INFOCOM' 02 [C]. New York: IEEE Communication Society, 2002. 2273 – 2279.

[6] Korkmaz T, Krunz M. Multi-constrained optimal path selection [A]. IEEE INFOCOM'01 [C]. Alaska: IEEE Computer and Communications Societies, 2001. 834 – 843.

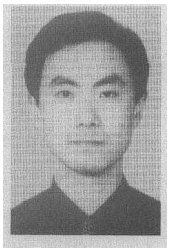
[7] Dijkstra E. A note on two problems in connection with graphs [J]. Numerische Mathematik, 1959, 1: 269 – 271.

[8] Sobrinho J L. Algebra and algorithms for QoS path computation and hop-by-hop routing in the Internet [A]. IEEE INFOCOM'01 [C]. Alaska: IEEE Computer and Communications Societies, 2001. 727 – 735.

[9] Zegura E W, Calvert K L, Donahoo M J. A quantitative comparison of graph-based models for Internet topology [J]. IEEE/ACM Transactions on Networking, 1997, 5(6): 770 – 783.

[10] Floyd S, Paxson V. Difficulties in simulating the Internet [J]. IEEE/ACM Transactions on Networking, 2001, 9(4): 392 – 403.

作者简介:



崔 勇 男, 1976 年 8 月生于新疆乌鲁木齐, 1995 年保送到清华大学计算机系读本科, 1999 年本科毕业后在该系直读博士学位, 他的主要研究兴趣包括计算机网络体系结构, 高速互联网路由体系结构, 性能评价, 协议仿真系统和测试, 以及多目标优化的路由算法. Email: cy@csnet1.cs.tsinghua.edu.cn.

徐 恪 男, 1974 年 12 月生于江苏洪泽, 2001 年 6 月于清华大学计算机系获得工学博士学位, IEEE 会员, 现任清华大学计算机系助理研究员, 主要研究兴趣是高速网络、交换机和路由器体系结构、服务质量路由算法和拥塞控制算法, 近年来已经在国内外主要学术刊物和学术会议上发表了 50 多篇科研论文.