Qos 路由问题 路由器 启发式算法

66-68

QoS 中的路由问题*)

The Problems in QoS Routing

彭　孜　曾家智　周明天

（电子科技大学计算机学院　成都 610054）

TP393　TN915.05

Abstract The next-greneration high-speed networks are expected to suppot a wide range of delay-sensitive multimedia applications. They need a different routing algorithm from the conventional one. The goal of the new algorithm is twofold:(1)satisfying the QoS requirements for every admitted connections, and(2)achieving global efficiency in resource utilization. Thus, most of problems in QoS routing area have multiple constraints which make them NP-Hard. Until now, the generally effcient algorithm has not been found. In this paper, we pose two algorithms for PCPO problems common in QoS communication, and analyze respective characteristics and uses.

Keywords QoS, Routing algorithm, NP-Hard

一、引言

QoS 中的路由问题可以形式地表述成在一个带权的简单无向图 G(V,E,W)中寻找适合条件的一条路径或一棵树。其中,V 是节点集,代表路由器或交换机;E 是边集,代表节点之间的线路;W 是赋给边的权值集,与传统的网络不同,这里的权往往是多元偶,代表 QoS 限制,举个例子,见图1。对单点传输(Unicast)而言,就是要在两个端用户之间找一条合乎要求的路径;对多点路由(Multicast)而言,就是要找一棵树,这棵树起于发送端,覆盖了所有的接收端,并且从发端到收端的每条路径都满足要求。这棵树又叫受限的斯坦纳树(Constrained Steiner Tree)。

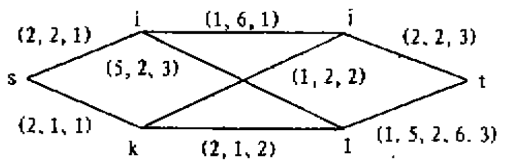


图1　链接状态(带宽、时延、代价)

QoS 中路由问题多种多样,详细的分类见文[1]。针对这些问题涌现了大量的算法,其中不乏好的。如,Sun-Landgendorfer[2]算法和 Kompella[3]算法,分别用于求单点路由和多点路由中时延受限-代价最优路径。我们重点讨论如何在单点路由中寻找时延受限-代价最优路径,这在 QoS 通信中很常见,比如,两个用户通过 IP 网打可视电话,就需要建立这样一条连接。它属于路径受限-路径优化问题(PCPO),是 NP-完全问题[4]。但是,如果所有的限制条件都相互关联,即是某一个量的函数,或者,这些条件除一个以外其余都取整数,问题就可在多项式时间内求解(在第二节算法1中给出了解释)。

二、我们的算法

下面的算法主要针对路径受限-路径最优(PCPO)问题。先约定在本节中使用的一些符号,U 为时延上限,d(u,v)为边(u,v)上的时延,c(u,v)为其上的代价,M 为边数,N 为节点数,D(P)和 C(P)分别为路径 P 的时延和代价。假定每个节点都保存一张距离向量表,分别记录到其它节点的最小时延和最小代价路径。首先证明一个简化定理,把所有的网络图都用此定理进行简化,可以删去一些不合条件的边,减少计算量。

定理1　令 $Ds=\min\{d(s,j)\}$(即以为 s 起点且时延最小的边),$Dt=\min\{d(j,t)\}$(即以 t 为终点且时延最小的边),任意一条边(u,v),如果 $d(u,v)>U-(Ds+Dt)$ 且它不与 s,t 相邻,则这条边可删去。

证明:设 s,t 间经过边(u,v)的路径的最小时延为 D,显然,$D\geq d(u,v)+Ds+Dt>U$,所以它不可能存

*)本课题受信息产业部电科院预研基金资助。彭　孜　硕士生,主攻方向为网络与通信。曾家智　教授,主要从事高性能网络研究。周明天　教授,博士导师,主要从事分布计算和网络研究。

在。在后面我们还将引入一个更强的定理。(u,v 与 s,t 点相异)

2. **算法描述**

算法1(最佳算法):实际上,Kompella 算法加以修正就可以用来建造源目的点间 PCPO 路,它属于源路由算法。假定时延取整数,设 U 为时延上限,B(s,v)为 s,v 间的最小时延值(按最短路径法算出),$C_d(s,v)$为点 s,v 间时延等于 d 的最小代价路径,P(s,v)为其间时延小于 U 的最小代价路径。显然,

$$C_d(s,v)=\min\{C_{d-d(u,v)}(s,u)+c(u,v)\} \qquad (1)$$

$$P(s,v)=\min\{C_d(s,v)\}(d<U) \qquad (2)$$

C(s,v)的初始值等于 $C_d(s,v)(d=B(s,v))$。令 $B^*=\min\{B(s,v)\}$,依次令 $d=B^*,B^*+1,\cdots,U$,(原算法从 d+1开始算,其实没必要,从 B^* 开始可以节约计算量)算出所有的 $C_d(s,t)$就能求出 p(s,t),这是最优而非近似解。算法的时间复杂度为 $O(N^2U)$,多项式时间(见后面证明),其实,这个方法也能用来求解多限制路径问题(MPC)。比如,求时延受限-抖动受限路径,显然,若 P(s,t)满足抖动限制,则它就是最后的解。下面以伪码来形式地描述算法。

```
for(d=B*;d<=U;d++)do
  for 任意 v 属于 V-s do
  {Cd(s,v)=Cd-d(u0,v)(s,u0)+c(u0,v);//赋初始值
  for 任意 u 属于 V-s-v-u0 do
  if Cd(s,v)>Cd-d(u,v)(s,u)+c(u,v)then Cd(s,v)=
  Cd-d(u,v)(s,u)+c(u,v)//找 Cd(s,v)的最小值
```

可以用下面的定理说明算法1的正确性。

定理2 算法1可以求出最优解,即算法是正确的。

证明:下面给出一个非形式化的证明。从给出的关系式(1)、(2)可以看出,算法把时延取各种值的情况都计算出来,再从这些值中选出最小值(即最优路径),因此本质上是穷举,当然可以求得最优解。算法的正确性依赖于公式(1)的正确。时延为 d 的路径可以按如下方法求出:从 s 沿时延为 d-d(u,v)的路径到 u(u 与 v 相邻),再从 u 到 v。算出所有时延为 d 的路径,从中选择代价最小的作为 $C_d(s,v)$,这正是公式(1)表达的含义。算法从一个初始值开始以每次递加1的方式迭代计算,时延小于 d 的路径已计算出,所以时延 d 的路径也可算出,这一步与 Floyd 算法[5]中从 A^K 计算 A^{K+1} 实质是一样的。补充说明一点,因为时延是整数,而整数是可枚举的,所以算法可以靠穷举求得最优解。

定理3 算法1的时间复杂度为 $O(N^2U)$,最坏情况不超过 $O(N^3)$(未考虑计算最短路径的时间,边的权值独立于 N)。

证明:对任意一点 v 计算 $C_d(s,v)$时,除去 s,v 外还有 N-2个点(u),按照要求应计算 N-2次,以便算出 s,v 间的所有路径进行比较。除 s 外,有 N-1个 v,所以对任意 d,得出所有的 $C_d(s,v)$需要 $O(N^2)$次计算,因为,d 可取 $U-B^*$ 个值,所以总的复杂度为 $O(N^2U)$。U 最多为 s,v 间最大时延路径的值,这条路径最多包含 N-1条边,设这些边中时延的最大值为 L(L 与 N 无关),有 $U<=L(N-1)$,可见 U 最多是 N 的一次多项式,所以最坏情况下,算法的复杂度为 $O(N^3)$

定理3说明了为什么在限制条件中除去一个外,其余的都取整数时,时间复杂度是多项式的。但是,也该看到随着限制条件增加,复杂度以乘积的形式增加。当条件较多时,运算量很可观。例如,时延和时延抖动都取整数,抖动的上限为 K,求合乎条件的最小代价路,按照上述思路,除了算出时延取各种值的情况外,对时延抖动也要作同样的计算,所以,复杂度变为 $O(N^2UK)$,的确以乘积的形式增长。对这样的组合优化问题寻求最优解,付出一点时间代价是正常的。不过,实际中很难出现条件多到难以计算的地步,算法还是有价值的。

算法适用于对路径限制很严格,网络变化轻微,路径一经建立不再轻易变动的场合。对于小规模的网络,如一个小的自治系统,算法也是可行的,实践中,广泛使用的是一些在结果的最优化和时间耗费之间作出折衷的近似算法,它们的效率比较高,得到的结果往往也令人满意。

算法2(近似算法):如果权值是一维的,就很容易求解,这启发我们将二维的权值变为一维。新的权值应能反映边的好坏(约定权值越小越好,表示为 w(u,v))。此法关键在于设计好变换函数,对于任意两条边,设时延和代价分别为 d_1,c_1,d_2,c_2,我们希望此函数能满足如表1的关系。即当 $d_1<d_2,c_1<c_2$时,w_1 比 w_2小,第一条边比第二条边好;$d_1<d_2,c_1>c_2$时,w_1,w_2大小由二者之和的大小决定,表中其余各项可类似理解。显然,函数 d+c 满足表1中关系。

表1

D_1 d_2	c_1 c_2	w_1 w_2
<	<	<
<	=	<
<	>	由 d+c 定
=	<	<
=	=	=
=	>	>
>	<	由 d+c 定
>	=	>
>	>	>

算法可表述如下:

(1)权值变换,把二维变为一维,图2表示这一过程;

(2)在新的权值图中,利用最短路算法找一条最短路 P,P 的时延和代价都应该是比较优的。

(3)把新的权值变为原来的,看 P 是否满足要求,

如不违背，则结束，否则转(4)；

(4)依次将 P 上的点 j(包括 t)作如下替换：$P_d{}^*(s,j)$ 替换 $P(s,j)$，再由 s 判断哪一次的替换最好，最坏情况是替换成 $P_d{}^*(s,t)$。

因为 Q_j 可由点 j 计算，最后由源点 s 选择最好的 Q_j，因此算法是分布式的。寻找最短路径的时间为 $O(N^2)$，因为 s，t 间的路径最多包含 N－2个节点，最坏情况下每个点进行一次替换，加上每条边的变换，所以总的复杂度为 $O(M+N+N^2)$。

上面的函数忽略了一条边与其它边的联系。某条边尽管 d，c 都比较大，但全局地看可能比较好，比如，与它相邻的边的权值很小。因此，变换函数应是全局性的。令

$$\begin{cases} D^*(u,v)=\min\{P_d{}^*(s,u)+d(u,v)+P_d{}^*(v,t),\\ \qquad P_d{}^*(s,v)+d(u,v)+P_d{}^*(u,t)\} \\ C^*(u,v)=\min\{P_c{}^*(s,u)+c(u,v)+P_c{}^*(v,t),\\ \qquad P_c{}^*(s,v)+c(u,v)+P_c{}^*(u,t)\} \end{cases}$$

因为经过边(u，v)的路径可能先经过 u，也可能先经过 v，所以应考虑两种情况，取其中的较小值。显然，$D^*(u,v)$，$C^*(u,v)$ 分别是经过边(u，v)的最小时延路径和最小代价路径的下限。如果 $D^*(u,v)$ 大于上限 U，那么最后符合条件的路径不可能包含边(u，v)，可以删去(这实际上又引入一个更强的简化定理)。令

$$w(u,v)=\begin{cases} C(D^*(u,v))-C^*(u,v)+D^*(u,v)-U \\ \qquad \text{if } U-D^*(u,v)\geqslant 0 & (1)\\ +\infty \quad \text{if } U-D^*(u,v)<0 & (2) \end{cases}$$

解释一下这个式子，当(2)为真时，经过边(u，v)的最短时延路都超过限制，所以令权值为无穷大。当(1)为真时，$D^*(u,v)-U$ 为负数，负得越多(即越小)，表明可扩展性越好。$C(D^*(u,v))-C^*(u,v)$ 表示最小时延路上的代价与最小代价路的差异，越小表明两种路径的重合度越好。完成权值变换后，其余的过程同上。

算法实际上是一个特殊的启发式算法，两种变权策略就是启发策略。许多启发式算法采用了类似贪婪法的策略，扩展到哪，就从哪里按照启发公式计算出最好的一点继续扩展，这在某些情况下会导致很差的结果，即所谓的“局部优化”。我们的算法把变权与选径分开进行，寻径前已经知道边的优劣，寻径时就可以充分利用这些信息。并且，定义全局变权法，使用不同于贪婪法的 Dijkstra 算法，也是意图避免这一情形。

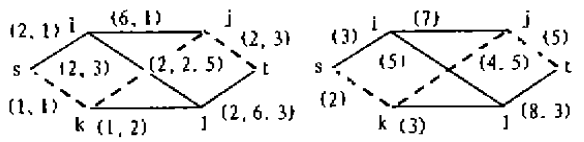
与前一个算法正好相反，这个算法属于近似算法，它吸引人的地方在于效率，能够在较短的时间内找到可行解。因此，适用于网络变化剧烈，状态更新快，需要经常调整路径(重路由)的场合。

定理4 如果 $U-D^*(u,v)<0$，则边(u，v)可删去。

证明：参考文中叙述(略)。

3. 实例分析

对算法1，我们主要关心其能否求出最优解，这已经从理论上得到证明(见文中叙述)，算法的时间复杂度也分析得很清楚，关键是看近似算法的性能，我们在 UNIX 环境下利用 C 语言实现了该算法，用进程模拟路由器进行信息交互。用来模拟的网络拓扑如图2，权值为(时延，代价)，找一条时延上限为5的最小代价路，图2(1)代表变换前的带权图，其中粗虚线表示用穷举法找出的最优路径；图2(2)表示变换后的图，这里使用局部变权法，变换函数为 d＋c。算法首先作权值变换，然后在(2)中利用 Dijkstra 算法找出最短路径(图中粗虚线表示)，可以看出它与(1)中路径重合。无须替换就找到可行解，而且是最优的。虽然这是最好的情况，但也说明算法的有效性。



(1)旧的带权图　　(2)新的带权图

图2　权值的变换

其实，路径计算可以更简单，利用定理1进行简化，这里，$D_i=1$，$D_j=2$，$U=5$，$U-D_i-D_j=2$，显然，边(i，j)满足定理的条件，所以可将它删去，删去一条边，可以节约计算量。

结语　QoS 中的许多路由问题是 NP 难的，计算开销比较大，因此急需寻找新的简单有效的算法，尤其是带有通用性和可扩展性的能满足一类 QoS 要求的算法。网络中状态更新总有一定的滞后，利用不精确信息进行路由也是值得研究的[6,7]。多路由和重路由以及在这个领域使用人工智能和神经网络技术也是今后发展的方向。

参考文献

1　Chen Shigang，Nahrstedt. An Overview of Quality of Service Routing for Next-Generation High-speed Networks：Problems and Solutions. IEEE Network，1998(Nov/Dec.)
2　Sun Q，Langendorfer H. A New Distributed Routing Algorithm with End-to-End Delay Guarantee，1997
3　Kompella V P. Multicasting Routing for Multimedia Communication. IEEE Trans. Networking，1993(June)
4　卢开澄. 组合数学-算法与分析(下册). 清华大学出版社
5　严蔚敏. 数据结构(第二版). 清华大学出版社
6　Guerin R，Orda A. QoS-based Routing in Networks with Inaccurate Information：Theory and Algorithms，IEEE INFOCOM97，Japan，Apr. 1997
7　Orda A. QoS Routing in Networking with Uncertain Parameters. IEEE Trans. Networking，1998(Dec.)