

基于信源路由的时延受限点到点路由算法

张宝贤, 刘 越, 陈常嘉

(北方交通大学通信与信息工程系, 北京 100044)

摘 要: 本文研究了网络路由中的一个 NPC 问题: 时延受限最小代价路由问题. 文中提出了一个理论框架, 并给出了多个简单有效的启发式算法, 在满足给定时延约束条件可行路径存在时, 算法总能找到满足约束条件的代价优化路径. 文中提出的启发式算法复杂性为 $O(|V|^2)$ 且在线复杂性为 $O(|V|)$. 仿真显示算法取得了良好的平均代价值. 最后将模型扩展到多 QoS 限制条件下的路由问题.

关键词: 时延受限; 服务质量路由; 信源路由

中图分类号: TN919.1 **文献标识码:** A **文章编号:** 0372-2112(2001)04-0516-05

Source-Based Delay-Constrained Unicast Routing Algorithms

ZHANG Bao-xian, LIU Yue, CHEN Chang-jia

(Department of Communication and Information Engineering, Northern Jiaotong University, Beijing 100044, China)

Abstract: In this paper, we study the NP hard delay constrained least cost routing problem. We propose a new framework to solve the problem and provide simple and efficient source routing heuristics from the model, through which one can always find a delay constrained path if such a path exists. Computation complexity of our heuristics is $O(|V|^2)$ and on line complexity is $O(|V|)$. Simulation results show that our heuristics achieve good cost performance. Finally, we extend the model to routing problem under multiple QoS constraints.

Key words: delay constrained; QoS based routing; source routing

1 引言

许多新的实时应用的出现, 如: 多媒体业务, 对网络路由算法产生了很大的影响. 这些应用对时延敏感, 并且占用大量网络资源. 因此对路由机制提出了如下要求: (1) 满足实时应用的端到端时延要求; (2) 有效地管理网络资源. 当前 Internet 上的路由机制主要根据最短路径算法, 它只能在给定的单个代价准则下找到基于最短时延或最多某种可用资源的路径. 寻找满足时延约束的最小代价路径是一个 NPC 问题^[1]. 因此设计简单、有效、可扩展的启发式算法, 不仅具有重要意义, 而且有很大的挑战性.

2 问题的形式化表示

给定有向简单图 $G=(V, E)$, 其中 V 是顶点的集合, E 是有向边的集合. 对于任意有向边 $e=(u, v) \in E$, 定义两个重量函数 $C(e)$ 和 $D(e)$. 其中 $D(e)$ 表示分组经过链路 e 时所经历的时延的测度. $C(e)$ 可以是链路使用价格或链路某些网络资源使用状态的测度. 给定信源 s 和信宿 d , 链路集合 $e_1=(s, v_2), e_2=(v_2, v_3), \dots, e_k=(v_k, d)$ 构成了从 s 到 d 的有向路径 $P(s, d)$. 路径 $P(s, d)$ 的代价定义如下:

$$C(P(s, d)) = \sum_{e \in P(s, d)} C(e) \quad (1)$$

类似, 定义路径 $P(s, d)$ 的端到端时延:

$$D(P(s, d)) = \sum_{e \in P(s, d)} D(e) \quad (2)$$

时延受限最小代价路由 (DCLC) 问题可以表述为^[2]: 对于有向图 $G=(V, E)$, 给定链路时延函数, $D: E \rightarrow R^+$ (R^+ 是除 0 以外的正实数集合), 链路代价函数, $C: E \rightarrow R^+$, 给定信源 s , 信宿 d 和时延约束 Δ , 寻找有向路径 $P(s, d)$, 满足:

$$D(P(s, d)) < \Delta \quad (3)$$

且最小化 $C(P(s, d))$. 这是一个 NP 问题.

3 理论框架

本文提出的模型在更高的层面上将网络看成由超级链路组成的超图, 超级链路指网络中两点间的最短时延(LD)路径或最小代价(LC)路径. 超图上路径 (v_1, \dots, v_m) 是由所有从 v_{i-1} 到 v_i 的超级链路组成的. 对于给定 s 和 d , 称由 s, d 确定的 n 阶超图 $SG_n(s, d)$ 由顶点集合 $\{s, d\} \cup V^n$ 组成(图 1(d)). 超级链路的使用和仅考虑两个出行方向(最小代价和最短时延方向)分析问题均不是全新的概念. 本文的主要贡献是将两

个概念用超图有效地组合起来, 给出一种系统可任意调节算法复杂性的处理框架, 较为深入地探讨了基于头两阶超图的启发式算法的复杂性、性能和应用, 并仿真评估算法性能, 与已知算法比较, 本文算法优势如下:

- ④ 若满足要求的解存在, 算法总能给出一条可行路径;
- ⑤ 能够较好地与现有的各种路由协议适配;
- ⑥ 算法得到的解自然无环, 使得算法的实现程序代码易写、易读、便于调测和可靠性高;
- ⑦ 复杂性低, 特别是在线算法复杂性只有 $O(1 \text{ V})$;
- ⑧ 仿真表明算法平均代价值性能优于已知算法。

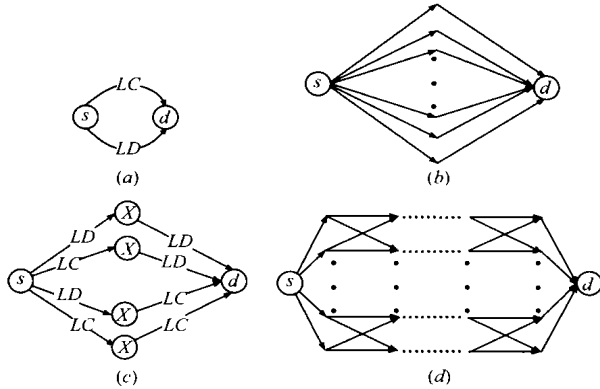


图 1 理论模型示意图

n 阶超图上的求解复杂性为 $O(V^n)$, 但仅考虑两个出行方向求解问题本身隐含了对简单性的追求要重于对精确性的追求, 低阶超图就显得更有意义, 本文着重研究 0 和 1 阶超图, 以及半阶超图. 0 阶超图(图 1(a)), s, d 之间仅有所示两条单跳路径. 1 阶超图如图 1(b) 所示, s, d 之间存在 $4V$ 条有向路径, 每条路径由两段超级链路组合而成, 相应表述成 $(s \rightarrow x \rightarrow d)$, $x \in V$, 通过给定结点 $x, x \in V$ 的组合路径如图 1(c) 所示. 连接两段超级链路的结点称中继结点. 实际上在一阶超图上, 可以通过恰当设计中继结点的选取空间降低求解复杂性、提高算法实用性, 如 0 阶超图中超级链路上的所有结点(指从 s 到 d 的最小代价和最短时延路径上结点)就是一个好的选择, 称之为半阶超图. 下面主要给出基于半阶和 1 阶超图的启发式算法.

4 信源路由点到点时延受限路由算法

首先, 定义文中用到的符号: $P_{ul}(u, v)$: 从结点 u 到 v 的最短时延路径; $P_{lc}(u, v)$: 从 u 到 v 的最小代价路径; $P_{ld}(u, v)$: 从 u 到 v 的最小代价路径; $D(u, v)$: 链路 (u, v) 上的时延; $C(u, v)$: 链路 (u, v) 上的代价; $relay_node$: 选择的中继结点, 即连接两段超级链路的结点; $PATH$: 最终求得的时延受限路径, 它记录如下信息: 采用的中继结点 $relay_node$, 从 s 到 $relay_node$ 的路径方向 up (最短时延和最小代价方向之一), 从 $relay_node$ 到 d 的路径方向 $down$ (最短时延和最小代价方向之一).

4.1 基于低价超图的启发式算法

算法执行过程中, 采用信源路由方式构造时延受限路径,

即: 信源具备网络拓扑和所有链路上的状态信息, 可行路径的计算由信源本地运算完成.

算法 1(0 阶超图上的启发式算法, 记作 **SDCR 0**): 如图 1(a) 所示, 从 s 到 d 的最小代价和最短时延路径之间选择满足约束条件代价小的一个作为问题的解.

算法 2(半阶超图上的启发式算法, 记作 **SDCR 1**): 显而易见, 把 $P_{ul}(s, d)$ 的一部分替换成最小代价路径可以降低代价, 而把 $P_{lc}(s, d)$ 的一部分替换成最短时延路径可以降低时延. 组合路径的构造方式如图 2 所示, 两条粗线代表 s, d 之间的最短时延路径和最小代价路径; 候选组合路径是细线代表的替代路径和相应粗线上的下游部分合成的组合路径. SDCR 1 可以分成两个阶段, 分别如下:

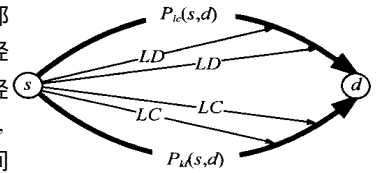


图 2 SDCR-1 选路示意图

第一阶段: 计算路径过程

步骤 1 计算从信源 s 到所有结点 $v \in G$ 的最短时延路径^{*}; 计算过程中, 检查:

$$D(P_{ld}(s, d)) < \Delta \quad (4)$$

如果不等式(4)不成立, 则 s 和 d 之间不存在满足给定约束的路径, SDCR 1 停止.

步骤 2 计算从信源 s 到任意结点 $v \in G$ 的最小代价路径^{**}. 计算过程中检查:

$$D(P_{lc}(s, d)) < \Delta \quad (5)$$

如果不等式(5)成立, 那么 $P_{lc}(s, d)$ 是问题的解, 算法结束.

步骤 3 在从 s 到 d 的最短时延路径上逆向寻找一个结点 v , 使得从 s 到 v 的最小代价路径上时延加上从 v 到 d 的最短时延满足时延约束, 将上述组合路径作为一个候选解保留. 相应计算如下:

初始, $relay_node = d$; /* d 是第一个检测结点 */
 $PATH = P_{ld}(s, d)$;

/* $P_{ld}(s, d)$ 是第一个已知可行路径 */

对于 $relay_node(rely_node(s), 执行循环\{$

$$D(P_{ul}(rely_node, d)) = D(P_{ul}(s, d)) - D(P_{ul}(s, rely_node)) \quad (6)^{***}$$

判断 $D(P_{lc}(s, rely_node)) + D(P_{ul}(rely_node, d)) < \Delta$ (7)

如果不等式(7)成立, 那么令: $PATH = P_{lc}(s, rely_node)$

* (I) 如无特殊声明, 文中计算最短路由采用 Dijkstra 算法^[3]; (II) 计算过程中, 当遇到距离信源最短时延大于 Δ 的结点时, 步骤 1 运算可以结束, 这是因为时延大于 Δ 的结点不可能成为从信源到信宿时延受限路径的一部分.

** 计算过程中, 当遇到距离信源最小代价大于 $C(P_{ul}(s, d))$ 的结点时, 运算可以停止, 这是因为将这样的结点包含在最终受限路径上将导致最终路径大于已知可行路径 $P_{ul}(s, d)$ 的代价.

*** 等式(6)的成立因为是全局最短路径, 那么一定局部最短; 同理说明等式(8).

$\cup P_{id}(relays_node, d)$; 转步骤 4;

否则选择 $relay_node$ 在 $P_{id}(s, relay_node)$ 的上一跳结点作为下一个检测结点, 继续循环;

}

步骤 4 在从 s 到 d 的最小代价路径上逆向寻找结点 w , 使得从 s 到 w 的最短时延加上从 w 到信宿的最小代价路径上的时延满足时延约束, 并将相应组合路径作为一个候选解保留, 当多个候选解存在时, 选择代价最小的一个. 最坏情况下, 仍保持步骤 3 中的解. 相应计算如下:

初始, $rdelay_node := d$;

对于 $relay_node (relay_node \neq s)$, 执行循环{

$D(P_{lc}(relay_node, d))$

$$= D(P_{lc}(s, d)) - D(P_{lc}(s, relay_node)) \quad (8)$$

$$\text{判断: } D(P_{id}(s, relay_node)) + D(P_{lc}(relay_node, d)) < \Delta \quad (9)$$

$$C(P_{id}(s, relay_node)) + C(P_{lc}(relay_node, d)) < C(\text{PATH}) \quad (10)$$

如果式(9)、(10)同时成立, 那么 $\text{PATH} := P_{id}(s, relay_node) \cup P_{lc}(relay_node, d)$ 并选择 $relay_node$ 在 $P_{lc}(s, relay_node)$ 的上一跳结点作为下一个检测结点, 继续循环;

}

return PATH;

其中 $C(\text{PATH})$ 是已知最好可行路径上的代价.

第二阶段: 路径建立过程

这一阶段根据第一阶段计算得到的路径建立一条从信源到信宿的时延受限路径.

方法 1: 沿第一阶段计算的路径, 发送一个控制分组通知沿途结点: 该结点在时延受限路径上的上下游关系并在有关路由器上建立一个表项从而为会话建立固定连接.

方法 2: 采用隧道的方式*, 即: IP 中的 IP, 信源在分组封装时, 第一层 IP 报头明确指出该分组须沿 up 方向发送到 $relay_node$, 第二层 IP 报头指示分组沿 down 方向从 $rdelay_node$ 到 d . 方法 2 的优点在于, 只需信源为给定 (s, d) 之间时延约束为 Δ 的所有会话按照前述隧道式分组封装. 沿途路由器上不再需要建立与该会话有关的路由表, 因而大大降低了路由表开销.

最近的研究显示, QoS 路由较传统路由提高了网络利用率, 然而实现 QoS 路由引入的代价也是关心的一个重点, 如算法的复杂性, 计算频度和协议开销. 这样在低处理代价下的良好路由性能更可取.

SDCR-1 的步骤 1、2 和 OSPF^[4] 的路由表计算方法一致. 因而可直接利用底层点到点路由表. 附加开销主要体现在对路由表保存内容的修改, 具体如下:

每个结点保持一个时延向量和一个代价向量. 每个结点 v_i 的时延向量由 $|V| - 1$ 项组成, 每项对应一个其他网络结点 $v_j (v_j \neq v_i)$ 包含如下信息: 信宿结点 ID; 从 v_i 到 v_j 的最短时延路径上的端到端时延; 从 v_i 到 v_j 的最短时延路径上的端到端代价; 对于 v_i , 从 v_i 到 v_j 的最短时延路径的下一跳结点; 对于

v_j , 从 v_i 到 v_j 的最短时延路径的上一跳结点.

每个结点 v_i 的代价向量由 $|V| - 1$ 项组成, 每项对应一个网络结点 $v_j (v_j \neq v_i)$ 包含如下信息: 信宿 ID; 从 v_i 到 v_j 的最小代价路径上的端到端时延; 从 v_i 到 v_j 的最小代价路径上的端到端代价. 对于 v_i , 从 v_i 到 v_j 的最小代价路径的下一跳结点; 对于 v_j , 从 v_i 到 v_j 的最小代价路径的上一跳结点.

在路由表中保存内容作出上述修改之后, SDCR-1 中只有步骤 3、4 需要在线进行, 算法执行时间为 $O(|V|)$.

4.2 算法 2(1 阶超图上的启发式算法, 记做 SDCR 2)

与 SDCR-1 类似, 算法执行分为两个阶段.

第一阶段: 路径计算过程

步骤 1-2 (同算法 SDCR-1).

步骤 3 计算图中所有结点到信宿 d 的最短时延路径;

步骤 4 计算图中所有结点到信宿 d 的最小代价路径;

步骤 5 根据步骤 1~4 的计算结果构造一阶超图(如图 1(b)) 从所有有向组合路径中选择满足时延约束代价最小的一个作为问题的解.

第二阶段: 路径建立过程

同样可以采用 SDCR-1 中采用的两种方法. 步骤 3~4 需要计算图中各点到信宿的最小代价和最短时延路径, 这可以通过修改 Dijkstra 算法进行反向最短路径计算完成. 修改后的算法复杂性仍为 $O(|V|^2)$. 因此, 步骤 1~4 执行时间均为 $O(|V|^2)$, 步骤 5 为 $O(|V|)$, 总计算时间为 $O(|V|^2)$. 若信源具有图中各点间的最小代价和最短时延信息, 那么只有步骤 5 需在线运算, 执行时间降为 $O(|V|)$.

4.3 算法的正确性

由于半阶超图是一阶超图的一种特例. 因而 SDCR-2 的正确性能够保证 SDCR-1 的正确性.

定理 1: SDCR 2 选择的最终组合路径是无环的.

证明详见附录.

4.4 最优解和 SDCR 2 获得的可行解之间的一些关系

下面是给出 SDCR-2 算法的解与最优解间定量关系的一些研究结果, 希望能为进一步的深入研究工作提供有意义的帮助. 规定 $\text{PATH}_{\text{opt}}(\Delta)$ 和 $\text{PATH}_{\text{SDCR-2}}(\Delta)$ 分别是时延限制为 Δ 时的最优解和 SDCR 2 算法找到的解, 有:

性质 1: 对于任意 $v \in \text{PATH}_{\text{opt}}(\Delta)$, $P_{lc}(s, v) \cup P_{lc}(v, d)$ 是一个可行路径组合与 SDCR 2 算法的解是最优解等价, 即: (1)

如果 $\text{PATH}_{\text{opt}}(\Delta) \neq \text{PATH}_{\text{SDCR-2}}(\Delta)$, 则 $D(P_{lc}(s, v)) + D(P_{lc}(v, d)) < \Delta$ 一定不成立; (2) 如果 $D(P_{lc}(s, v)) + D(P_{lc}(v, d)) < \Delta$, 那么 $\text{PATH}_{\text{opt}}(\Delta) = \text{PATH}_{\text{SDCR-2}}(\Delta)$;

性质 2: 对于任意 $v \in V$, 如果 $P_{lc}(s, v) \cup P_{ld}(v, d)$ 和 $P_{ld}(s, v) \cup P_{lc}(v, d)$ 同时可行, 那么对于 $P_{lc}(s, v)$ 和 $P_{lc}(v, d)$ 的路径组合 $LC(s, v, d)$ 有, $D(LC(s, v, d)) \leq 2\Delta - D(P_{id}(s, d))$.

推论: (1) 若 $P_{lc}(s, v) \cup P_{lc}(v, d)$ 是一个可行路径组合但

* 这种方式适用于 IP 网络环境. 一般采用的方式是松散信源路由, 信包头中包含分组必须依次穿越的沿途路由器地址清单, 这里只需要包含一个中继结点的地址.

SDCR-2 算法的解优于 $LC(s, v, d)$, 则 $v \notin PATH_{opt}(\Delta)$; (2) 若结点 $v \in PATH_{opt}(\Delta)$ 且 $P_{lc}(s, v) \cup P_{ld}(v, d)$ 和 $P_{ld}(s, v) \cup P_{lc}(v, d)$ 同时可行, 则 $C(PATH_{SDCR-2}(2\Delta - D(P_{ld}(s, d)))) \leq C(PATH_{opt}(\Delta))$, 即: 在这种条件下, SDCR-2 算法对时延约束 $2\Delta - D(P_{ld}(s, d))$ 找到的解不会劣于时延约束为 Δ 时的最优解。

推论的一个可能应用是在计算过程中一直记录一条代价最小的满足性质 2 的路径 $LC(s, v, d)$, 若最终解与这条路径的代价有较明显劣化, 则可以回问用户是否接受 $2\Delta - D(P_{ld}(s, d))$ 的时延, 当应用对时延的要求较松散时, 有可能接受这样的要求, 特别是当 $D(P_{ld}(s, d))$ 接近 Δ 时。

5 性能仿真

修改文[5]中仿真模型进行仿真实验来评估本文提出算法的性能。仿真中采用不同规模的双工, 简单连通 ATM 网络, 链路容量 155Mbps。网络结点的位置随机散布在 $4000 \times 2400\text{km}^2$ 范围之内。采用文[6]中的随机图产生机制来构造随机图, 同时保证随机图的输出总是一个连通网络。

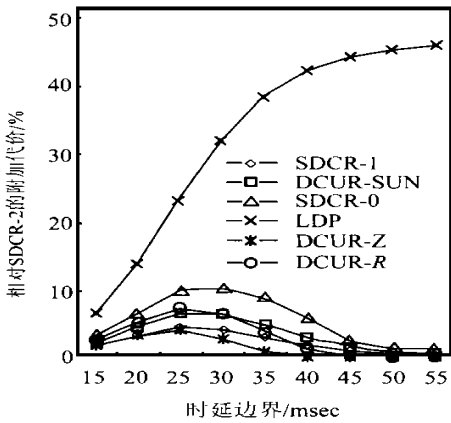


图 3 150 节点, 平均度数 4, 可变时延约束条件下代价比较

链路上传播时延采用光速的 $2/3$, 在这一假定下, 覆盖网络的矩形尺寸是 $20 \times 12\text{msec}^2$ 。假定高速网络, 分组尺寸很小, 每个结点有限缓存。此时链路时延主要由传播时延决定, 排队时延可以忽略。定义链路代价是 1 到 10 之间的一个随机

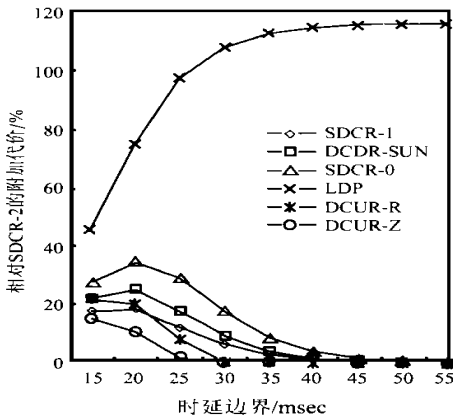


图 4 150 节点, 平均度数 10, 可变时延约束条件下代价比较

整数。仿真试验在 1000 组随机网络中进行, 对于生成的每个随机网络, 在所有可能的(信源, 信宿)对间构造受限路径, 再对所有成功连接进行代价平均。仿真比较了 SDCR-0, SDCR-1, DCUR-SUN^[7], DCUR-R^[2], DCUR-Z^[8], LDP 相对于 SDCR-2 的代价性能。其中 LDP 指最短时延路由。算法 x 的额外开销为:

$$Inefficiency_x = \frac{(\text{cost}_x - \text{cost}_{SDCR-2})}{\text{cost}_{SDCR-2}}$$

图 3 显示在 150 节点, 平均度数为 4, LDP, SDCR-0, DCUR-SUN, SDCR-1, DCUR-R 和 DCUR-Z 相对 SDCR-2 的代价额外开销最大比率分别为: 46%, 11%, 7%, 5%, 7%, 4%。而当平均度数为 10 时(图 4), 代价额外开销最大比率分别上升到: 117%, 35%, 26%, 19%, 22% 和 11%。

6 多 QoS 限制条件下的路由问题

多限制条件下的 QoS 路由问题: 给定有向简单图 $G=(V, E)$, 其中 V 是顶点的集合, E 是有向边的集合。对于任意有向边 $e=(u, v) \in E$, 定义 n 个 QoS 测度, $L_i(e), i=1, \dots, n$ 。传统最短路径算法假定 $n=1$, 但对于 QoS 路由问题, $n>1$ 。给定信源 s 和信宿 d , 链路集合 $e_1=(s, v_2), e_2=(v_2, v_3), \dots, e_k=(v_k, d)$ 构成了从 s 到 d 的有向路径 $P(s, d)$ 。对于路径 $P(s, d)$ 的第 i 个测度定义如下: $C_i(P(s, d)) = \sum_{e \in P(s, d)} L_i(e)$ 。那么 QoS 路由可以表述如下: 给定图 $G=(V, E)$, 信源 $s \in V$ 和信宿 $d \in V$, 一组最大允许测度值 $L_i > 0, i=1, \dots, n$, 寻找 s 和 d 之间一条路径满足 $L_i(p) < L_i, i=1, \dots, n$ 。这样的路径称为是可行的。本节提出的算法是对第三节模型的扩展。为了限制复杂性, 集中讨论对一阶和半阶超图的扩展, 具体如下:

步骤 1 计算信源到网络中各点和网络中各点到信宿基于测度 $L_i(e), i=1, \dots, n$ 的最短路由;

步骤 2 构造类似于图 1(b) 式的网络, 其中从 s 到 d 经过网络中某个点 x 的组合路径数为 n^2 。相应路径组合为 $least_cost_i(s, x), i=1, \dots, n$ 路径之一和 $least_cost_i(x, d), i=1, \dots, n$ 之一的组合。

步骤 3 从上述路径中选取同时满足: (a) 所有约束条件; (b) 最小化 $\sum_{i=1}^n L_i(p)$ 或最小化所有测度中最重要的一个。

步骤 3 中的条件 (b) 对所选路径进行了一定程度的优化, 同时避免了环路的发生。步骤 1 的计算时间为 $O(n|V|^2)$ 。步骤 2, 3 的执行

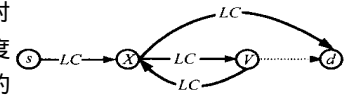


图 5 可能的环路例子

时间为 $O(n^2|V|)$, 因而总的计算复杂性为 $O(n^2|V| + n|V|^2)$ 。同样算法可以采用类似 SDCR-1(或 SDCR-0)的简化方法, 即: 将中继节点的选择限定在从 s 到 d 的第 i 最短路径上, 然后对 i 从 1 到 n 循环。这样没有必要计算图中各点到信宿的第 i 测度最短路由。

7 结论

本文提出的算法的核心思想是通过设计结点间最短时延和最小代价路径组合策略, 从组合路径中选择满足时延约束

条件代价最小的一个作为问题的解.文中给出了基于链路状态网络环境的多种启发式算法.与前人工作比较,本文的主要贡献在于:(1)构造了一个适合求解各种类型 QoS 路由问题的一般性框架;(2)给出了多种简单而有效启发式算法;(3)提出的算法可以通过引入传统的隧道机制,在降低路由表规模的同时实现 QoS 路由.仿真显示提出的算法简单而有效,易于在链路状态网络环境中实现.未来的工作是将本文提出的算法扩展到组播应用中.

参考文献:

- [1] Z. Wang et al. Quality of service routing for supporting multimedia applications [J]. IEEE JSAC, 1996, 14(7): 1228- 1234.
- [2] D. S. Reeves and H. F. Salama. A distributed algorithm for delay constrained unicast routing [J]. IEEE/ ACM Transaction on Networking, Apr. 2000, 8(2): 239- 250.
- [3] D. Bertsekas and R. Gallager. Data Networks [M]. Prentice Hall, 1987.
- [4] J. Moy. OSPF Version 2- RFC No. 1583 [M]. INTERNET- RFC, March 1994.
- [5] H. F. Salama. Multicast routing for real time communication on high speed networks [D]. PhD thesis, North Carolina State University, Department of Electrical and Computer Engineering, Nov. 1996.
- [6] B. M. Waxman. Routing of multipoint connections [J]. IEEE JSAC, 1988, 6(9), 1617- 1722.
- [7] Q. Sun et al. A new distributed routing algorithm for supporting delay sensitive applications [R]. Internal Report. Institute of Operating Systems and Computer Networks, TU Braunschweig, Bueltenweg 74/ 75, 38106 Braunschweig, Germany, March 1997.
- [8] 张宝贤. 实用可扩展 QoS 路由的研究 [D]. 博士学位论文. 北京: 北方交通大学, 2000, 4.

附录:

定理 1 SDCR 2 选择的最终组合路径是无环的.

证明 证明分为以下几种情况来说明.

(1) 若 $P_k(s, d)$ 满足时延约束, 则它是问题的解, 显然无环;

(2) 若以结点 $v(v \neq s, d)$ 作为中继结点, 经过 v 的某种路径组合是 SDCR 2 得到的解, 那么存在四种组合情况, 即: $P_{ld}(s, v)$ 和 $P_k(s, v)$ 两者之一, 和 $P_{ld}(v, d)$ 和 $P_{lc}(v, d)$ 两者之一的组合路径, 下面以其中一种为代表进行分析.

假定 $P_k(s, v)$ 和 $P_{lc}(v, d)$ 是组合路径中, 满足时延约束的代价最小的. 假定 $P_{lc}(s, v)$ 和 $P_{lc}(v, d)$ 具有公共结点, 不失一般性, 假定相交于结点 x (如图 5 所示). 由 $D(P_{lc}(s, v)) + D(P_k(v, d)) < \Delta$, 有 $D(P_k(s, x)) + D(P_{lc}(x, d)) < \Delta$. 再 $C(P_k(s, x)) + C(P_k(x, d)) < C(P_k(s, v)) + C(P_{lc}(v, d))$, 从而以结点 x 作为中继结点, $P_{lc}(s, x)$ 和 $P_{lc}(x, d)$ 的路径组合满足时延约束且代价优于 $P_k(s, v)$ 和 $P_k(v, d)$ 的路径组合, 与假定 $P_{lc}(s, v)$ 和 $P_{lc}(v, d)$ 是满足时延约束条件代价最小的组合路径相矛盾, 因此 $P_{lc}(s, v)$ 和 $P_{lc}(v, d)$ 不存在环路. 同理证明其他组合情况. (3) 若以 $x, x \neq s, d$ 为中继的路径组合均不满足时延约束条件或代价高于 $C(P_{ld}(s, d))$, 则 $P_{ld}(s, d)$ 是问题的解, 显然无环; 证毕.

作者简介:

张宝贤 生于 1972 年, 现于北方交大通信与信息工程系攻读博士学位. 主要研究方向是: QoS 路由, 组播通信, 计算机网络.

刘 越 生于 1973 年, 现于北方交大通信与信息工程系攻读博士学位. 主要研究方向: 组播通信, Internet 路由.