

一种分类预计算 QoS 路由算法*

冯 径^{1,2,3}, 周润芳^{1,3}, 顾冠群^{1,3}

¹(东南大学 计算机科学与工程系,江苏 南京 210096);

²(解放军理工大学 气象学院,江苏 南京 210007);

³(计算机网络和信息集成教育部重点实验室,江苏 南京 210096)

E-mail: jfeng@seu.edu.cn; rfzhou11@yahoo.com.cn; ggu@seu.edu.cn

http://www.seu.edu.cn

摘要: 为了满足 Internet 上各种网络应用的传输服务质量的需求,Internet 工程任务组(IETF)先后提出了集成服务/资源预留模型(integrated service/resource reservation protocol,简称 IntServ/RSVP)、区分服务模型(differentiated service,简称 DiffServ)以及从流量工程角度提出的多协议标记交换(multi protocol label switching,简称 MPLS)。这些服务模型都需要一个与之相适应的 QoS 路由机制和算法,为不同的服务质量请求分配不同的路由。在研究各种现有的 QoS 路由算法的基础上,提出了一种以带宽为基本尺度的分类的最小代价路径优先算法(classified minimum cost path first,简称 CCPF)。并通过计算机仿真,与其他相关算法进行了比较。结果证明,CCPF 算法能在较小的时间复杂度内有效地计算并找出不同带宽区间的路由,可供多种服务模式作为候选路由使用。

关键词: 服务模式;QoS 路由;算法;多目标规划

中图分类号: TP391 文献标识码: A

目前,在基于 IP 技术的互联网上,网络服务质量的矛盾表现为:发送方希望能随意发送高负载、高突发性的通信量;接收方期望低延迟、高吞吐率地接收信息;服务提供方希望以最小的基础设施代价换取最大的投资回报率,这使得网络服务质量研究成为热点问题。

首先,增加带宽是适应实时网络应用的第 1 步,但这仍然不能在突发流量时有效地避免抖动等问题,所以必须增加 IP 的服务能力,以使网络能对有严格时间要求的应用加以区分,并提供适应性服务。就我们目前对网络技术的理解,认为 QoS 体系结构的可能组件应包括:服务定义(提供不同级别 QoS 的服务)、用户通信需求的交互方法(信令、接纳控制和策略管理)、服务提供者保证用户委托的方法(策略控制和整形)、服务提供者寻径的方法(基于 QoS 的路由选择)以及基于 QoS 的转发机制(缓冲区分配和丢弃策略、排队规范和服务策略、流量管理)^[1]。

作为网络传输 QoS 体系的重要组件之一,QoS 路由选择有两类应用背景,一类是为流量工程;另一类是为动态请求。对于前者来说,其操作主要以长程的通信量变化为基础,以聚集流为处理对象,考虑粗粒度的性能需求,此时的 QoS 路由选择的目标是在缓慢改变通信量模式的情形下获得最大的网络整体性能(如普遍地减小延迟)。这要通过持续地测量通信量的图谱来计算流量聚集的路径、优化各种性能的测度。而对于后者来说,QoS 路由是为每个请求而计算的,这些请求被显式地表达成资源的需求,路由计算更加频繁,资源分配的粒度更小,因此,

* 收稿日期: 2000-07-05; 修改日期: 2000-10-16

基金项目: 国家自然科学基金资助项目(69896249);国家 863 高科技发展计划资助项目(863-300-02-03-99);国家重点基础研究发展规划 973 资助项目(G1999032700);江苏省自然科学基金资助项目(BK9909)

作者简介: 冯径(1962 -),女,江苏南京人,博士,讲师,主要研究领域为计算机网络协议、路由机制和算法;周润芳(1974 -),女,江苏丹阳人,硕士生,主要研究领域为计算机网络协议工程;顾冠群(1940 -),男,江苏常州人,教授,博士生导师,中国工程院院士,主要研究领域为计算机网络、分布处理、CIMS。

这种背景下的 QoS 路由选择的目标是满足单个请求的性能约束.换句话说,为了提供有力的保证,不仅在更新网络状态和路由计算上有更大的开销,而且还要为此付出附加的信令开销^[2].

针对第 2 类应用背景,同时考虑到网络状态的不精确性,为了适当地减少路由计算的频度并提供一定使用范围的路由,我们提出了采用预计算的方法,事先对应用流量分几个区间,计算满足这几个区间的带宽请求的可行路径,然后根据应用请求服务模式,选择合适的转发路由.也就是说,将路由问题分成与实际请求无关的可行路径计算和与实际请求相关的路由优化选择两部分.

本文第 1 节描述 QoS 路由中多目标规划问题.第 2 节提出基于 CCPF 算法的 QoS 路由机制.第 3 节给出了对 CCPF 算法本身的性能进行仿真比较的结果.第 4 节对全文进行总结.

1 用混合法解决 QoS 路由的多目标规划问题

与单目标决策不同,多目标决策最显著的特点是目标之间的不可公度性和目标产生的矛盾性.所谓目标之间的不可公度性,是指各个目标没有统一的度量标准.如在 QoS 路由的选择目标中,通常希望端到端的延迟最小,丢失率最小,瓶颈带宽最大,所占用的网络资源最少,若各种服务所需的费用不同,用户还希望使用网络的费用最小等.在上述 5 个目标中,延迟是用时间单位(秒或毫秒)度量的,丢失率一般无量纲(也可以用比特),瓶颈带宽用比特/秒度量,所占用的网络资源目前还没有一定的度量标准,有的算法认为路径的跳数在一定程度上表现了占用的网络资源,使用网络的费用当然用钱来衡量.这样一来,从物理意义上讲,不能把多个目标直接归并为单个目标;而目标之间的矛盾性使得强调改善某个目标,却使该方案中另一个目标变坏(如强调服务质量指标,可能会使网络的使用费用增大).

因此,在多目标规划中,一般不存在所有目标函数共同的极大点,需要引进非劣解(noninferior solution)的概念,非劣解又称为有效解或 Pareto 最优解(Pareto optimal solution).一般来说,一个多目标规划有无穷多个非劣解,使决策者满意的非劣解叫最终解(final solution).

多目标规划的标准形式是:

$$\begin{aligned} (\text{VOP}) \max & [f_1(x), \dots, f_p(x)]; \\ \text{s.t. } & x \in X \end{aligned}$$

其中 $x=(x_1, \dots, x_n)^T$ 是 n 维向量, x 所在的空间叫决策空间, $f_1(x), \dots, f_p(x)$ 称为目标函数, p 维向量 $(f_1(x), \dots, f_p(x))$ 所在的空间称为目标空间, X 是决策空间上的可行集.多目标规划问题又称为向量最优化问题(vector optimization problem, 简称 VOP),常用的方法有加权法、约束法和混合法^[3].

混合法依据 x^* 是多目标规划(VOP)的非劣解的充要条件:对于任意给定的一组 $w_j^0 > 0, j=1, \dots, p$, 存在一组实数 $\varepsilon_j (j=1, \dots, p)$, 使 x^* 是:

$$\begin{aligned} \max & \sum_{i=1}^p w_i^0 f_i(x); \\ \text{s.t. } & f_j(x) \geq \varepsilon_j, j=1, \dots, p \quad x \in X \end{aligned}$$

的一个最优解.

像约束法那样,有规律地变动 $(\varepsilon_1, \dots, \varepsilon_p)$ 的值,求解一系列上述问题,可以得到近似的非劣解集.因为此条件较弱,混合法比加权法或约束法要方便.

假设在 QoS 路由的选择目标中,决策空间 $x=(x_1, \dots, x_n)^T$ 分别对应(链路长度、媒体类型、租用费率、端口吞吐率、剩余带宽、端口缓冲能力、路径的跳数(链路数)),则目标函数 $f_1(x), \dots, f_p(x)$ 分别代表路径 P 上的端到端传输延迟函数、丢失率函数、可利用带宽函数、网络资源占用函数和费用函数.

通常,我们希望在可行路径集 (P_1, P_2, \dots, P_m) 上得到满足以下目标的优化路径:

$$\min f_1(x), \min f_2(x), \max f_3(x), \min f_4(x), \min f_5(x).$$

根据网络模型^[4],我们可以得到相应的目标函数表达式,为了便于对应到多目标规划的求解过程和借用“最短路优先”的路由算法,当给定一组约束值 $\varepsilon=(D, L, B, R, C)$ 时(它们分别代表路径上的允许传输延迟、丢失率、瓶颈带宽、网络资源占用限制和费用),不妨将可利用带宽函数 $f_3(x)$ 进行加权处理,这样就将取极大值转换成取

极小值,相应地 B 转换成 B' .所以,约束值变成 $\epsilon'=(D,L,B',R,C)$.因此,上述 QoS 路由可形式化描述成:

$$(\text{VOP})\min[f_1(x),\dots,f_p(x)];$$

$$\text{s.t. } x \in X$$

采用混合法,可将该(VOP)问题转化成以下形式:

$$\min \sum_{i=1}^p w_i^0 f_i(x);$$

$$\text{s.t. } f_j(x) \leq \epsilon_j', j=1, \dots, p, x \in X$$

其中 $p=5, X$ 为包括 x 在内的结点和链路属性, w_i^0 为根据应用需求和决策者的偏好给定的一组权值,它反映了决策者对每个决策目标的偏重程度.当只考虑其中一个目标时,如占用网络资源最少, $f_4(x)$ 简单地表示成路径的跳数,则此时 $W=(0,0,0,1,0)$, $f_4(x)=x_7$.问题转化成一般的求最小跳数的路径问题.当限定两个目标时,如最小瓶颈带宽和传输延迟,则 $W=(1,0,1,0,0)$,可以在限制带宽的前提下(即在满足带宽的链路中),以传输延迟为尺度,利用最短路径算法求解路径.

因此,当需要满足更多的目标时,可以按照每次给定不同的权值(一般选取两个尺度,尽量避免相同操作的尺度),求得一组路径.在这些路径集合中,首先标定共同的链路,再根据偏好(如第一考虑占用较少网络资源的目标,第二考虑带宽等),选取满足该目标的其他链路.

2 CCPF 算法

2.1 路由尺度的选择

目前的集成服务用 RSVP 来充当 QoS 请求的信令协议,它只在确保服务(guaranteed service,简称 GS)中对带宽提出了定量的要求,对排队和处理的延迟控制是通过令牌桶的整形和调度来完成的,虽然可以利用 ADSPEC(advertisement specification)对象的相关字段计算出从源点 s 到当前结点 v_i 的延迟 d_i ,但其前提是每个中间结点必须执行令牌桶调度策略,否则,计算将无法进行.而 C-LS(control-loaded service)并没有定量地确定服务质量,预留消息的主要内容就是 TSPEC(traffic specification)和服务类型说明,完全可以从预请求和 Path 消息中得到,无须等到 Path 消息到达后再决定,因此预留状态可以在传递 Path 消息时一并完成,省略 Resev 消息的开销.

扩大一个尺度(metric),就会增加很多计算开销,甚至导致路由计算的 NP-完全问题^[5].若考虑到在高速链路的网络中,延迟主要来自处理,则可以将带宽看成影响性能的主要矛盾,而将链路的媒体类型作为路由选择的参考因素,对于时间敏感的应用来说,要过滤掉像卫星、无线等高延迟链路.

定理 2.1. 若链路代价取 $Cost_i=(1-Re_i/C_i)$, Re_i 为链路 i 的可利用带宽, C_i 为链路 i 的容量.当且仅当 $Re_i \geq$ 请求带宽 W , 路径 P 有最小的 $Cost_i$, 链路 $i \in P$, 则 P 是满足 W 的有最大可利用率的路径.当 $C_i=C$ 时, P 是满足 W 的有最大瓶颈带宽的路径.

证明:当 $Re_i \geq W, Cost_i=(1-Re_i/C_i)=(C_i-Re_i)/C_i$.

因为 Re_i 是链路 i 的可利用带宽(即剩余带宽),所以, $(C_i-Re_i)/C_i$ 为链路 i 的使用率, $Cost_i$ 表达了链路 i 已被使用的情况;

若路径 P 满足 $Cost(P)=\min Cost_i$, 链路 $i \in P$, 每条链路目前的使用率都最小,即相对而言空余的比率最大,故 P 是满足 W 的有最大可利用率的路径.

当 $C_i=C$ 时, $Re_i \geq W, Re_i/C$ 越大,说明 $(Re_i-W)/C$ 越大,即满足请求带宽 W 的余度越大.

显然, $0 \leq Re_i/C \leq 1, \min Cost_i=(1-Re_i/C)$ 表示了 $\max Re_i/C$, 所以, P 是满足 W 的有最大瓶颈带宽的路径.

从最优化理论的角度来看,定理 2.1 的 $Cost$ 可以理解成阶段指标.

上述 $Cost$ 可理解为在网络 (V,E) 上寻找一条从源点 s 到目的结点 t 的路径 $P(s,t \in V)$, 该路径具有满足带宽 W 要求的最大可利用性.

可将其转化为最小代价问题,指标函数为阶段指标之和,最优值函数 $f(v_j)$ 是由 s 出发,到终点 t 的最小费用.

即求 $\text{Min}_j(f_j)$.

基本方程为

$$f(v_j) = \text{Min}[Cost(v_i, v_j) + f(v_i)]$$

$$v_0 = s, f(v_0) = 0$$

$$\text{s.t. } v_i \in V, i=0, 1, 2, \dots, n$$

$$W > 0$$

$$0 \leq Re_i \leq C_i$$

2.2 CCPF算法的基本思想

我们的路由选择由两个组件构成,一是计算从某一结点到达任意结点满足约束条件的可行路径算法;二是从可行路径中选择一条路径以平衡网络负载的优化算法.

为了在多项式时间内用尽量少的系统开销找出可行路径,根据对网上通信量的统计分析,可以把带宽请求分成3个区间, $R_1=64\text{Kbps}$, $R_2=1.5\text{Mbps}$, $R_3=4\text{Mbps}$.就是说,预先计算瓶颈带宽 B 满足 $R_1 \leq B < R_2$, $R_2 \leq B < R_3$ 和 $B \geq R_3$ 的可行路径.为此我们设计了一个基于传统 Dijkstra 最短路径算法的分类最小代价路径优先算法 CCPF(classify minimum cost path first),以带宽作为主要的路由尺度(链路约束),根据定理 2.1,引入加权代价函数进行全局路径优化($Cost=1-R/C \cdot R$ 为链路剩余带宽, C 为链路容量).该算法以满足请求 GS 对带宽的要求为目标.

定义 2.1. 从源点 S 到目的结点 Z 的路径 P 可以用一个链表来表示,它包括源结点 S 和目的结点 Z 的标识、路径 P 的瓶颈带宽和所经过的跳数(链路数),并有一个指向下一跳路由器的指针.

```
struct node{
    int vi; //结点标识
    node *adjacent; //指向某个结点(结点数组中的任意一个元素)的邻接点
    int band; //从该结点到 adjacent 的链路的可利用带宽
    int capac; //从该结点到 adjacent 的链路的容量
    enum {UTP,CAX,OPF,SAT}media; //媒体类型,分别为双绞线、同轴电缆、光纤和卫星
}net_node[MAXNODE]; //网络中的结点数组,指示相应结点元素的邻接表

struct state{
    int vi; //结点标识
    float cost_si; //从 S 到 i 的
    int hop_si; //该路径的跳数
    int bottleband; //该路径的瓶颈带宽
    int perpoint; //结点 i 的前驱
    enum {perm,temp}lab; //结点 i 的状态,分别为“永久”和“临时”,以免循环路由
}link_state[3][MAXNODE]; //从源到所有可能结点的 3 类路由
```

选择的 R_1, R_2, R_3 这 3 个区间,基本上对应于目前因特网上开展的普通字符文件传输、压缩视频和音频信号传输以及其他实时多媒体通信.

基于 Dijkstra 最短路径算法的 CCPF 算法,将 R_1, R_2, R_3 分别作为请求带宽 W , 目前网络链路的可利用带宽为 R , 按照广度搜索,寻找满足条件的最优路径.实现以最大带宽 R_3 为判断条件,然后依次以 R_2 和 R_1 为判断条件,在满足条件的链路中将 $Cost$ 最小的链路加入路径.显然,满足高带宽要求的链路一定满足低带宽的要求,这样,3 条路径将会有很多共同链路.为了在网络中尽可能地找出满足条件的、由不同链路组合的路径,用控制条件过滤掉一些高带宽链路,然后用一个辅助程序根据网络结点的邻接关系和所选的 3 条路径构造路由表.

下面给出 CCPF 算法的伪代码,假设结点用 0 到 n 的整数来标识,求出从源点 s 到任意其他结点的满足带宽要求的最小代价路径,通过其他结点的前驱来标识.

(1) 初始化状态结构

```
Initial(state*p)
{
  for (p=&link_state[0][0];p<&link_state[3][n];p++)
  {
    p->vi=-1;
    p->perpoint=-1;
    p->cost_si=MAXVALUE;
    p->lab=temp;
    p->bottleband= MAXVALUE;
    p->hop_si= MAXVALUE;
  }
}
```

(2) 计算从源点 s 到其他各结点的最小代价路径(结点编号从 0 到 n)

```
Minnum_Cost_Path(int s)
{
  int i,j,hop_num,node_num;
  float c,min_c;
  int k[3]; //记录每类带宽区间的在当前循环中的起始点
  int pk[3]; //记录每类带宽区间的在前一次循环中的起始点
  node*q; //工作结点指针
  for(i=0;i<3;i++){
    link_state[i][s].vi=s;
    link_state[i][s].cost_si=0;
    link_state[i][s].lab=perm;
    link_state[i][s].hop_si=0; //将源点的代价、跳数置为 0,状态置为永久
  }
  k[0]=s;k[1]=s;k[2]=s; //从源点开始搜索
  node_num=1;
  do{
    i=2;
    q=& net_node[k[i]];
    do{
      if (q!=NULL){//对每个点 k[i],取其邻接点为 j,计算从 k[i]到 j 的代价
        j=q->vi; //指向 k[i]的一个邻接点
        if (q->band>=R3)AND(link_state[i][j]. lab==temp){
          c=1-q->band/q->capac; //≥R3 带宽区间的链路的代价
          Compute_Cost; //计算满足≥R3 带宽的路径代价
        }
      }
    }
    else{
      i=1;
      if (q->band>=R2)AND (link_state[i][j]. lab==temp){
        c=1-q->band/q->capac; //[R2,R3]带宽区间的链路的代价
      }
    }
  }
}
```

```

        Compute_Cost; //计算满足 $[R_2, R_3]$ 带宽的路径代价
    }
    else{
        i=0;
        c=1-q→band/q→capac; // $[R_1, R_2]$ 带宽区间的链路的代价
        Compute_Cost; //计算满足 $[R_1, R_2]$ 带宽的路径代价
    }
}
q=q→adjacen;
} while(q!=NUL); //当 q=NUL 时,对所有  $k[i]$  的邻接点搜索完毕
for(i=0;i<3;i++){
    min_c=MAXVALU;
    for (j=0;j<n;j++){//在所有临时状态的结点中寻找距离源最小代价的结点为起点
        if (link_state[i][j].lab==temp AND link_state[i][j].cost_si<min_c){
            min_c=link_state[i][j].cost_si;
            k[i]=j; //取最小代价的结点为新一轮循环起点
        }
    }
}
link_state[i][k[i]].lab=perm; //从源到  $k[i]$  的第  $i$  类路径上的下一跳为  $k[i]$ 
node_num++; //实际网络结点加 1,又完成通往一个结点的路径
}
} while(node_num!=n); //当所有结点寻找完毕,就结束
}

void Compute_Cost() //对每类带宽区间的链路,计算从  $k[i]$  到  $j$  的最小代价
{
    if (link_state[i][k[i]].cost_si+c<link_state[i][j].cost_si){
        link_state[i][j].cost_si=link_state [i][k[i]].cost_si+c; //更新代价
        link_state[i][j].hop_si=link_state[i][k[i]].hop_si+1; //更新跳数
        link_state[i][j].perpoint=k[i]; //记录前驱结点
        if (q→band<link_state[i][k[i]].bottleband){
            link_state[i][j].bottleband=q→band; //记录路径上的瓶颈带宽
        }
    }
}
}

```

2.3 与服务模式相关的路由选择机制

以集成服务模式为例,可以在 RSVP 应用程序接口中发送一个预请求消息 Pre_Resv,告诉发送方(server)自己所需的服务类型.

定义 2.2. Pre_Resv::=<Server 地址/端口,Client 地址/端口,服务类型,[URL],[价格]>.

与集成服务给定的标识相一致,GS = 5,C-LS = 2.若不发预请求消息,则认为无须特殊的 QoS 支持,RSVP 守护进程不作任何处理,按传统的尽量服务处理.URL 用于 Client/Server 模式下,由接收方指定所要访问的文件路径;“价格”即为用户愿意支付的服务代价,这两项都设为可选项.

当得知一个应用请求时,我们采用服务类型与“最大适应”相结合的方式,在可行路由中选择合适的下一跳.若没有合适的路由,则触发路由计算程序,按新的请求执行路由选择程序.我们设计了一个最大适应路由选择算

法 RBAR(most redundant bandwidth adaptive route selecting),从如图 1 所示的扩展的 OSPF 路由表中选取合适的路由(表中的可行路径按带宽的降序排列,即 R_3, R_2, R_1).

Type	Dest	Area	Path type	Candidate hop	Adv
				Cost BottB Router	LinkBand Madium Next
					...

类型, 目的结点, 域, 路径类型, 备选路径, 邻居结点, Cost 为到目的结点的代价;BottB 为路径的瓶颈带宽;Router 为下一跳邻居结点标识, LinkBand 为下一跳链路的带宽;Madium 为下一跳链路的媒体类型;Next 为下一类候选结点.

Fig.1 Extended OSPF routing table

图 1 扩展的 OSPF 路由表

对于不同的服务类型采用不同的选择策略:

- 对于 GS,考虑尽量减少链路拥塞的可能性,所以只要 $W \leq R_3$,不考虑 R_2, R_1 的下一跳是否可能满足要求,就将其作为下一跳的路由.此时要附加地判断一下应用对时延的要求,若要求最小时延,则要避免卫星等长延迟链路.
- 对于 C-LS,由于它不会在所选路径上预留指定的带宽值,我们考虑以路径带宽的总体性能为度量,即通过 $Cost$,即使通信量峰值有可能突破路径的瓶颈带宽,但只要其路径的 $Cost$ 较小,就说明整个路径有较大的可利用率,服务质量以某个概率 P_c 得到保证.
- 对于 B-ES,RARS 不去进行任何判断,总是将排在最后的候选路由作为默认的下一跳路由,即存在满足 R_1 瓶颈带宽的路由,则选之,否则 R_2, R_3 依次类推.这样尽量保证网络中所有链路均衡使用.

3 CCPF 算法性能分析

影响路径计算的开销有两个因素,一个是算法本身,另一个是算法被实际调用时的情况.前者是指算法本身的时间和空间复杂度;后者是由于算法在实际调用时所需的链路状态更新和计算触发条件所引起的占用处理器的时间和带宽的程度.

我们先来考虑算法本身的复杂度.

可以证明,CCPF 算法在最坏情况下,可以在 $O(N(L+3M))$ 时间复杂度内找出从某一结点到其他任意结点的满足一定带宽区间的所有可行路由,其中 N 为网络中的结点数, L 为网络结点的最大度数.

对 CCPF 算法的仿真测试,在处理器为奔腾的 PC 机上,主频 450MHz,128MB 内存,10GB 硬盘,在 FreeBSD3.4-RELEASE 环境下运行.随机产生 20~200 个结点的网络拓扑,各条链路的可利用带宽取 $[0,100\text{Mbps}]$ 的随机数.在每种拓扑结构下任选 10 个源结点,按 CCPF 算法计算从源到其他任意结点的路径,统计每种拓扑结构下的平均延迟.

RFC2676 也给出了满足多个级别的预计算 QoS 路由算法,但它是以带宽为限制条件,以 hop 数为 Dijkstra 最短路径操作尺度(metric),在带宽 \geq 某个级别带宽的链路中计算路径.这种处理对于最高级别的带宽是没有问题的,但其他较低级别带宽,有可能大多数链路会选择与高级别带宽路径相同的链路,从而造成大多数流聚集到高可利用带宽链路上的情况,导致网络资源使用的不均衡.同时,RFC2676 建议的算法也没有考虑对链路传输媒体的考虑,这在高速传输时还是很重要的,因为当距离足够长时,带宽造成的延迟与媒体传播延迟相比可能不占主要地位.

通过计算机仿真,我们将 CCPF 算法、RFC2676 给出的满足多个级别的预计算 QoS 路由算法^[6](图 2 中标注为 OSPF)和 Chotipat 的 QoS_{BF} ^[7]进行了比较,图 2 中分别显示了计算时间(a)、成功率(b)和瓶颈带宽(c)等项指标.

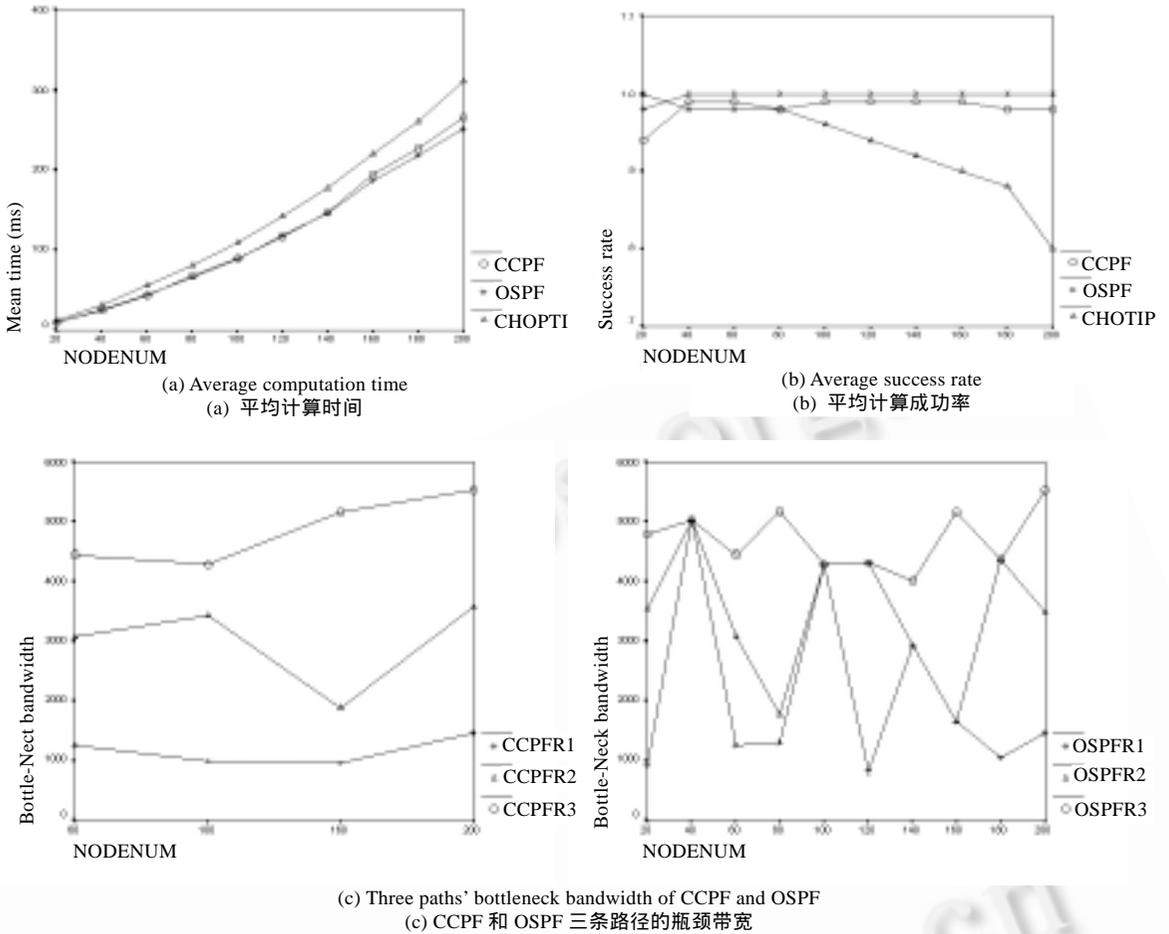


Fig.2
图 2

因为 CCPF 一次计算满足 3 个带宽期间的路由,为了使算法具有可比性,我们在实现 RFC2676 给出的算法时,也设置同样的 3 个带宽,同时让 Chotipat 的算法运行 3 次,每次满足与上述算法相同的带宽.需要说明的是,在测试平均计算时间和平均成功率时,3 个算法对于不同的结点集合,是在不同的随机网络上运行的,这样考虑主要是为了更能反映平均情况.CCPF 和 OSPF 在 20~200 个结点的随机网络上,每次任选一对(源,目的)对,分别找到的路径的瓶颈带宽见表 1(表中数字 *xx-yy* 意为瓶颈带宽-跳数).

从图 2(a)和图 2(b)中可以看出,RFC2676 扩展的 OSPF 算法平均执行时间最短、成功率最高,CCPF 次之,Chotipat 的算法最弱.这至少反映以下两个事实:

(1) 路由约束越松,成功率和执行时间越小.

Chotipat 的算法是采用 Bellman-Ford 方法,以带宽为链路约束,延迟为优化路径条件,同时限制最大跳数,所以,当网络范围越大(结点数目越大)时,其成功率越低;而 CCPF 和 OSPF 算法均是基于 Dijkstra 最短路径法,前者是用带宽的函数作为路径优化尺度,在满足带宽的区间中选择链路;后者用 hop 数作为路径优化尺度,在大于等于某带宽的链路中选择,所以当结点数目增大时,判断和计算时间较少.

(2) 一次选择 3 条(有限条)路径所花的开销,显然比运行 3(多)次选择 3(多)条路径要小,因为路由程序除了执行计算路径的代码以外,还要执行其他辅助工作,尤其是可以将计算满足有限个带宽区间的判断放在一个循环中时,寻径的效率会提高.

Table 1 Comparison of a group of bandwidths between CCPF and OSPF algorithms**表 1** CCPF 和 OSPF 算法的一组路径的瓶颈带宽比较

CCPF	20	40	60	80	100	120	140	160	180	200
R_1-R_2	837-5	985-6	1256-4		985-4			934-6	1055-5	
R_2-R_3			3065-4		3425-6	3795-6	2929-2	1875-8		3578-6
$\geq R_3$	4794-6	5030-3	4463-6	5184-7	4293-1	4322-5	4006-5	5166-5	4374-6	5542-11
OSPF										
$\geq R_1$	904-2	5030-3	1256-4	1292-3	4293-1	798-3	2929-2	1651-4	1055-5	1472-4
$\geq R_2$	3555-5	4030-3	3065-4	1777-4	4293-1	4322-5	2929-2	1651-4	4374-6	3491-4
$\geq R_3$	4794-6	5030-3	4463-6	5184-7	4293-1	4322-5	4006-5	5166-5	4374-6	5542-11

从表 1 和图 2 中可以看出,CCPF 的分带宽区间计算路由,由于路径带宽限制上限,所以 3 类路径是经过完全不同的链路,当随机网络的可利用带宽超过上限时,有可能找不到合适的路径,但第 3 个区间是 $\geq R_3$,与 OSPF 算法给出的结果一致.而 OSPF 算法表面上看来每次都能找到路由,但 3 类路由有较大的重复.因此,用 CCPF 算法至少可以找到满足最大带宽(R_3)的路由,这条路径是可以满足需要 R_1 和 R_2 带宽的请求;所选路径在规定的带宽区间里具有最大可能利用率,但不一定是最小的跳数,适合对要求不同服务质量保证的应用请求分配不同的路径,可以均衡使用网络.OSPF 算法能找到大于某个带宽的路径,且具有最小跳数,剩余带宽较大的链路会被重复选择,在下一个计算周期里,有可能变得拥塞.

同时也可以看出,任何一个算法都不能覆盖所有的优点,但用 CCPF 算法的区间限制和 $cost$ 函数,可以方便地更改路由策略,如将每条链路的 $cost$ 设置成 1,带宽上限取无穷大,就变换成了 OSPF 的算法.

4 结束语

我们给出的 CCPF 算法,由于大部分计算发生在后台计算,其算法本身的复杂度不像请求计算那样严格.对于适应性 QoS 路由来说,有可替代路由是很重要的,用 CCPF 算法,一次计算可提供 3 个带宽区间的可行路由,便于在拒绝接纳和发生链路故障的情况下很快地找到替代路由.我们把该算法在 NS 环境中加以扩展,以不同的应用流量为背景,也得到了肯定的结果.同时,在 MPLS 上将获得的不同路径分配给不同的转发等价类,其仿真实验正在进行之中.

References:

- [1] White Paper—QoS Protocols & Architectures. <http://www.qosforum.com>.
- [2] Decasper, D., Dittia, Z., Parulkar, G., *et al.* Router plugins: a software architecture for next-generation routers. *IEEE/ACM Transactions on Networking*, 2000, 8(1):2~15.
- [3] Ma, Zhen-hua. *Modern Applying Mathematics Manual—Operational Research and Optimization Theory*. Beijing: Tsinghua University Press, 1998. 307~369 (in Chinese).
- [4] Feng, Jing, Ma, Xiao-jun, Gu, Guan-qun. Network model research adapted to QoS routing mechanism. *Chinese Journal of Computers*, 2000, 23(8):799~805 (in Chinese).
- [5] Wang, Z., Crowcroft, J. Quality of service routing for supporting multimedia application. *IEEE Journal on Selected Areas in Communications*, 1996, 14(7):1288~1234.
- [6] Apostolopoulos, G., Williams, D., Kamat, S., *et al.* Routing mechanisms and OSPF extensions. In: *Internet Engineering Task Force Request for Comments 2676*, 1999.
- [7] Chotipat, Pornavalai, Goutam, Chakraborty, Noria, Shirator. QoS based routing algorithm in integrated services packet networks. *Journal of High Speed Networks*, 1998, 3(7):99~112.

附中文参考文献:

- [3] 马振华. 现代应用数学手册——运筹学与最优化理论卷. 北京:清华大学出版社, 1998. 307~369.
- [4] 冯径, 马小骏, 顾冠群. 适应 QoS 路由机制的网络模型研究. *计算机学报*, 2000, 23(8):799~805.

A Classified Pre-Computed QoS Routing Algorithm*

FENG Jing^{1,2,3}, ZHOU Run-fang^{1,3}, GU Guan-qun^{1,3}

¹(Department of Computer Science and Engineering, Southeast University, Nanjing 210096, China);

²(Institute of Meteorology, PLA University of Science and Technology, Nanjing 210007, China);

³(Key Laboratory of Computer Network and Information Integration, Ministry of Education, Nanjing 210096, China)

E-mail: jfeng@seu.edu.cn; rfzhou11@yahoo.com.cn; ggu@seu.edu.cn

http://www.seu.edu.cn

Abstract: In order to satisfy the quality of service (QoS) of the application on Internet, Internet engineering task force (IETF) has presented a set of network service mode such as IntServ/RSVP (Integrated Service/Resource Reservation Protocol), DiffServ (differentiated service) and MPLS (multi protocol label switching). All of these service modes need a correlative QoS routing mechanism and algorithm to assign an adapted route. Based on the research of existing routing algorithms, a new pre-computed QoS routing algorithm CCPF (classified minimum cost path first) is put forward, which uses bandwidth as a basic metric and gives a weighted cost function as a path optimal criterion. By simulation, this algorithm is compared with the others. The results show that it can get different routes among different bandwidth-ranges to support multiple service modes.

Key words: service mode; QoS routing; algorithm; multiple object programming

* Received July 5, 2000; accepted October 16, 2000

Supported by the National Natural Science Foundation of China under Grant No.69896249; the National High Technology Development 863 Program of China under Grant No.863-300-02-03-99; the National Grand Fundamental Research 973 Program of China under Grant No.G1999032700; the Natural Science Foundation of Jiangsu Province of China under Grant No.BK9909

第 2 届中国 Rough 集与软计算学术研讨会(CRSSC 2002)

征文通知

由中国计算机学会人工智能与模式识别专业委员会主办、苏州大学承办的第 2 届中国 Rough 集与软计算学术研讨会(CRSSC 2002)拟定于 2002 年 10 月下旬在苏州召开.欢迎高等院校教师、科研院所和企业的科技人员以及博士生、硕士生踊跃参加.

一、征文范围

- (1) Rough 集理论及其应用;Rough 逻辑与 Rough 推理;Rough 代数理论及其应用;
- (2) Fuzzy 集理论与 Fuzzy 逻辑;Dynamic Fuzzy 集与 Dynamic Fuzzy 逻辑;
- (3) 分布智能、群体智能、网络智能;神经网络模型与新算法;
- (4) 人工生命;演化计算;计算智能;
- (5) 数据挖掘;知识发现;机器学习;知识工程;知识理论;
- (6) 信息 Granular 与 Granulation 计算;
- (7) 自然语言理解与机器翻译;
- (8) 智能控制、智能管理;
- (9) 软计算理论及其应用;
- (10) 可拓工程理论及其应用;
- (11) 其他.

二、征文要求

论文必须未公开发表过,一般不超过 6000 字;论文格式参照《计算机研究与发展》排版;论文一律用 A4 打印稿,一式 3 份,用 Word 7.0 以上排版,欢迎电子投稿;大会录用论文将在《计算机科学》杂志上发表.

三、重要日期

截稿日期: 2002 年 6 月 10 日 录用日期: 2002 年 7 月 10 日

四、联系方式

投稿方式: 215006,江苏省苏州市苏州大学计算机工程系 李凡长 教授 电子投稿请送: lfzh@suda.edu.cn

联系电话: 0512-5113205(O)(石明芳,王根荣) 李凡长教授电话: 013962116494;0512-5112737(办);0512-7489202(宅)