

一种可选择最佳带宽的多路路由算法*

王洪波¹, 张尧学¹, 郭国强², 顾钧³

¹(清华大学 计算机科学与技术系 网络互联实验室,北京 100084);

²(常德师范学院 计算机科学系,湖南 常德 415003);

³(香港科技大学 计算机科学系,香港)

E-mail: wuanghb@yahoo.com; zyx@moe.edu.cn

http://sun475.cs.tsinghua.edu.cn

摘要: 实现服务质量保证,在多用户、多请求接入的网络中都是采取为相应请求预留资源的方式,而多请求预留资源使得网络局部出现资源“碎块”.单路路由算法不能利用这些资源接入用户请求.为提高网络的资源利用率,采用多路路由方法,在单一通路不能满足连接请求要求时,寻找多条能联合满足要求的一组虚拟不相交并路,并预留资源共同满足用户的服务质量要求.为验证算法的有效性,在 ns2 模拟器上实现了所提出的算法,并与其他算法进行了性能比较.实验结果表明,提出的算法能很好地适应网络运行状况.在网络轻负载情况下,通信开销小.在网络重负载情况下,使用并行多路增加连接请求的接通率.将此算法应用于支持服务质量保证的多媒体数据网,能充分利用有限的网络资源,提高传输多媒体信息的能力.

关键词: 多路路由;积极出弧;服务质量;虚拟不相交通路;可行并路

中图分类号: TP391 文献标识码: A

在包交换网络中,按多个参数选择路由,实现资源预留,保证服务质量^[1-5]是网络研究的热点和难点.多路径路由就是在相同的源节点和目的节点之间不存在满足要求的单路径时,选择多条综合能力满足要求的路径.研究多路路由的目的是通过分摊负载的方式解决 QoS 保证、负载平衡和提高资源利用率等问题.多路路由问题是流量规划基础研究问题之一.

多参数约束下的路由问题多数为 NPC(NP-complete)问题^[5],Wang 在文献[5]中分析得出带宽和延迟、代价、延迟抖动、丢失率之一结合的两参数路由问题是多参数路由中仅有的非 NPC 问题,这类问题可以在多项式时间内求解.对于多媒体业务来说,带宽和延迟无疑是最重要的参数指标.目前已有很多带宽和延迟相结合的单路路由算法,参见文献[1~4].

J.J. Garcia-Luns-Aceves, Israel Cidon 等人对多路径路由问题进行了分析研究,并提出了一些算法^[6-9].J.J. Garcia-Luns-Aceves 在文献[6~8]中提出的算法都是支持多路路由的路由表维护算法,并没有具体阐述连接建立时选路的策略和预留资源的过程.Cidon 等人在文献[9]中给出了 3 种多路选路算法,但仅是多路探测,最后选择其中的一条单路.其中的快速算法主要考虑的找路速度,最优路径可能并没有选中,并且通信开销呈指数增长;慢速算法为了保证选择最优的通路需要等收到所有输入链路上的路由探测包之后才前向探测,延迟最长,连接响应时间性能最坏;超快速算法采用泛洪式探测策略先建立源和目的对的多播连接,然后在传送数据阶段逐渐减少为单播连接,通信开销大,并且长时间预留不会使用到的资源,这样反而降低资源利用率.

* 收稿日期: 2000-11-23; 修改日期: 2001-05-09

基金项目: 国家自然科学基金资助项目(69873024);国家重点基础研究发展规划 973 资助项目(G1998030406)

作者简介: 王洪波(1977 -),男,江西湖口人,硕士生,主要研究领域为路由算法,服务质量控制;张尧学(1956 -),男,湖南常德人,博士,教授,博士生导师,主要研究领域为网络互联,服务质量控制;郭国强(1964 -),男,湖南常德人,副教授,主要研究领域为服务质量控制,流量规划;顾钧(1956 -),江苏苏州人,博士,教授,博士生导师,主要研究领域为多媒体通信,VLSI 电路和系统设计,高性能软件系统,并行分布式处理器,联合优化.

由于源路由方式不能解决算法的扩展性问题,本文在文献[6~8]中算法的基础上提出一种新的分布式路由^[1~4,9]选择算法,算法主要应用在要求服务质量保证的连接请求环境.当网络运行在重负载情形且不存在独立满足连接要求(主要考虑延迟和带宽)的单路径时,本文提出的算法同时在多条路径预留资源,建立源和目的之间一组通路,共同满足连接的要求.由于算法能充分使用在单路路由算法建立通路情况下无法使用的资源为用户提供服务,无论是从平衡负载、提高网络资源利用率来看,还是从增加连接请求的接通率的角度来看,我们提出的多路路由算法都具有良好的应用前景.

1 模型与定义

我们把计算机网络抽象为连通图 $G(N, L)$, N 是路由器的集合, L 是传输线路的集合. 每条链路 l 上传播延迟 d^l 是常量, 可用带宽 b^l 时常变化, 且都是非负数. 由一条链路直接相连的两节点称为邻居节点. 用 a^i_m 表示与邻居 m 相连的链路, 节点 i 的邻居节点集用 N^i 表示. 假定邻居节点间无差错通信, 通信数据在有限时间内按序到达. 接收节点在一个单位时间内处理接收的信息, 与链路直接相连的节点在有限时间检测到链路失效、恢复和代价变化.

我们提出的分布式路由算法是在连接请求包到达的节点 i 时, 根据邻居节点提供信息选择到达目的节点 j 的优先下一跳节点集 $N^i_j(t) \subseteq S^i_j(t)$, 其中 $S^i_j(t) \subseteq N^i$ 为文献[7]中计算的去往目的 j 的无环多路上的下一节点集. 下面我们给出本文中常用的其他定义和缩写.

概念 1. 虚拟不相交路径. 从路径的传输能力上看, 若两条路径上的公共节点和公共链路都不是两条路径的瓶颈节点和瓶颈链路, 则称这两条路径为虚拟不相交路径.

概念 2. 可行并路. 能满足连接请求 r 的服务质量要求的一组并行虚拟不相交路径, 记为 fp^r ; fp^r_i 表示节点 i 与目的节点 j (也用 t^r 表示) 之间连接请求 r 的可行并路, $fp^r_i(l)$ 表示沿节点 i 的出弧 l 的子并路, $fp^r_i(l) \subseteq fp^r_i$, $N^i_{fp^r}$ 和 $L^i_{fp^r}$ 分别表示可行并路 fp^r 在节点 i 的下一节点集和出弧集, 并记 $FP^r = \{fp^r_1, fp^r_2, fp^r_3, \dots\}$.

概念 3. 积极下一跳. 节点 i 与去往目的节点 j 的满足条件的下一跳. 记优先下一跳的集合为 N^i_j .

概念 4. 积极出弧. 与优先下一跳相连链路相称, 记为

$$a^i_{jm} = (i, m), \quad m \in N^i_j,$$

并记

$$A^i_j = \{a^i_{jm} \mid m \in N^i_j\}.$$

概念 5. 积极可用带宽. 它是节点 i 到目的节点 j 的所有积极出弧上的可用带宽之和, 记为

$$AB^i_j = \sum_{l \in A^i_j} b^l.$$

概念 6. 积极并路. 从节点 s 出发, 所有访问到节点 i 都沿 A^i_j 到 $N^i_j(t)$, 直到目的节点 j 的有向连通路由图, 记为

$$AMP^s_j = \{(s, m) \mid m \in N^s_i(t)\} \cup \{(m, n) \mid n \in A^m_i(t), m \in A^s_i(t)\} \cup \dots \cup \{(u, j) \mid j \in N^u_j(t)\}.$$

概念 7. 最大并路. 从节点 s 出发, 所有访问到节点 i 的都向 S^i_j 转发请求节点, 直到目的节点 j 的有向连通路由图.

$$MMP^s_j = \{(s, m) \mid m \in S^s_i(t)\} \cup \{(m, n) \mid n \in S^m_i(t), m \in S^s_i(t)\} \cup \dots \cup \{(u, j) \mid j \in S^u_j(t)\}.$$

文献[6~8]已证明 $\{(m, n) \mid n \in S^m_i(t), m \in N\}$ 组成的路由图 $SG_i(t)$ 即使在链路状态时常变化的情况下的任意时刻都无环路, 故 MMP^s_j 中无环路. 又因为 $N^i_j(t) \subseteq S^i_j(t), m \in N$, 所以 AMP^s_j 中也无环路.

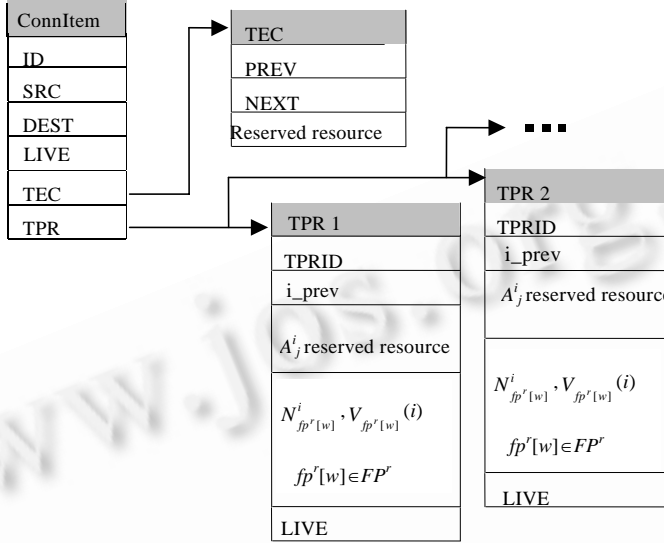
2 分布式多路路由算法(DMPRA)

2.1 数据结构

文献[6~8]都没有考虑如何为需要服务质量保证的应用提供保证服务, DMPRA(distributed multipath parallel routing algorithm)为了能按应用的要求建立通路并保证服务质量, 除了需要维护^[6]的一些路由信息外,

还要在连接所经历的节点上维护相应的连接信息.下面介绍本算法特殊需要的数据结构,其他表项见文献[6].

(1) 连接表.存储节点 i 上现有的连接信息,每个连接项 ConnItem(如图 1 所示),包括连接标识 ID、临时请求连接项 TPR(temporary processing request)和已确定连接项 TEC(table of established connection).TPR 项包括 TPR 标识符、TPR 存活活期、请求包经历的前一节点 i_prev 、连接请求到达时计算的 N^i_j 在 A^i_j 及节点 i 上预留的资源,其他可行路集 FP^r .TEC 项记录连接的前一节点集(输入邻居集)、下一节点集和预留的资源.



连接表项, 预留的资源.

Fig.1 Structure of item of route connection table

图 1 路由表连接项结构图

(2) 主距离表.除文献[6]中主距离表的所有信息外,记录到达每个节点的最小多路延迟.路由信息包的结构如下:

- ID^r :连接标识符
- s^r, t^r :为连接的源和目的节点
- σ^r :连接的最大突发速率
- $path$:路由信息包经历的路径向量
- D^r :最大端到端延迟限制
- b^r :带宽要求
- c^r :最大数据包的大小要求

2.2 计算 $A^i_j(N^i_j)$

在数据结构部分,我们定义路由包结构的同时也相应定义了连接的特性,如果以 $n(p)$ 表示路径 p 的跳数,路径 p 上的最大有效带宽为

$$r(p) = \min_{l \in p} b_l . \tag{1}$$

同时假定网络每个节点基于速率调度策略,因此可以估算在路径 p 上连接 r 的端到端的延迟为

$$D^r(p, r) = \frac{\sigma^r + n(p) * c^r}{r(p)} + \sum_{l \in p} d^l . \tag{2}$$

我们用 $D^r(p)$ 表示 $D^r(p, r)$ 的最小可能值,也称作路径 p 的保证延迟.可见,在单路连接中可行路径的条件是

$$D^r(p) \leq D^r, \text{ 且 } r(p) \geq b^r . \tag{3}$$

并行多路径连接可行路的条件是

$$\forall p \in fp^r \quad D^r(p) \leq D^r, \text{ 且 } \sum_{p \in fp^r} r(p) \geq b^r . \tag{4}$$

因此,增加并路径上的预留带宽或减少路径的跳数能减少端到端的延迟.但由于带宽的凹特性^[2],在逐跳分布式

预留带宽算法中,每个节点在下游积极出弧集上最多只能有效使用当前请求的带宽.

在支持并行多路径连接网络中,我们假定每个节点基于速率调度策略根据每条路径的传输能力转发数据包,则可以计算 fp_{ij}^r 到达目的节点的延迟,

$$V_{fp_i^r}(i) = \begin{cases} 0 & \text{if } i = j, \\ W_i + \sum_{l=(i,m) \in E_{fp_i^r}^l} p(l) * (V^*(m) + d^l) & \text{if } i \neq j, \end{cases} \quad (5)$$

其中

$$W_i = \frac{\sigma^i + c^i}{b^r} \quad (6)$$

是一个数据包在路由器 i 的发送延迟,忽略排队延迟 W_i 即为数据包待时间的期望值.

$$p(l) = \frac{r(fp_i^r(l))}{b^r}, \quad (7)$$

式(7)为在可行并路中使用子可行并路的传输数据的占总到达数据的比例.所以

$$V_{AMP_j^i}(i) = V_r^*(i) = \min\{V_{fp_i^r}(i) \mid \forall fp_i^r \in FP_i^r\}, \quad (8)$$

即 $V_r^*(i)$ 是从 i 点到目的点 j 连接 r 的最小延迟期望.

下面我们介绍计算 A_j^i 的过程,并给出示例.当节点 i 收到连接请求时,向邻居节点 $m \in S_j^i$ 查询邻居节点到达目的节点的最小延迟期望,在收到所有查询节点的 $V_j^*(m)$ 后,先去掉 $V_j^*(m_u) + d_{m_u}^i \geq D^i$ 节点 m_u ,剩下的节点集以 b^i 为界划分为数个集合,每个集合中的节点唯一地确定一条从 i 到目的节点 j 的可行并路, $b_{m_u}^i \geq b^i$ 为单节点集合 $\{m_u\}$,在 $b_{m_u}^i < b^i$ 的所有节点集中遵循组合的原则组成集合,每个子集合确定的并路上可用带宽不小于 b^i ,并且若去掉子集中任意节点的新并路可用带宽小于 b^i .然后在优先使用最短路径的原则下,根据式(5)计算由每个子集中节点所确定的并路的最小延迟期望.选择最小延迟期望值最小的集合为 N_j^i .

算法 1. 计算积极出弧集.

Procedure calcu_AOS

Begin

For $m \in S_j^i$ Do Request $V_j^*(m)$; End for

Do wait while received all m replying $V_j^*(m)$;

$S = S_j^i$; $m \in S_j^i$

while $m \in S$ and $V_j^*(m) + d_m^i > D^i$ do $S = S \setminus \{m\}$; End while

$t = 0$;

while $m \in S$ and $b_m^i \geq b^i$ do $N_{fp_i^r[t]}^i = \{m\}$; $S = S \setminus \{m\}$; $t++$; End while

$B = \sum_{m \in S} b_m^i$;

If $B > b^i$ then divide_set(S , $N_{fp_i^r[t]}^i$, t); End if

If $t = 0$ then $A_j^i = \phi$;

Else

$N_j^i = \phi$; $V_{AMP_j^i}^* = \infty$;

For all $N_{fp_i^r[w]}^i$ do calculate $V_{fp_i^r[w]}(i)$;

If $V_{fp_i^r[w]}(i) > D^r$ then remove $V_{fp_i^r[w]}(i)$ from $N_{fp_i^r[w]}^i$; End if

If $V_{fp_i^r[w]}(i) < V_{AMP_j^i}^*$ then $N_j^i = N_{fp_i^r[w]}^i$; $V_{AMP_j^i}^* = V_{fp_i^r[w]}(i)$; End if

End for

Remove $N_{fp^r[w]}^i = N_j^i$ from $N_{fp^r[]}$

End if

End procedure

例 1:如图 2 所示,请求 r 达到节点 n , (参数如图), 起始 $S = S_j^n = \{m_1, m_2, m_3, m_4, m_5\}$, n 查询 $m_1 \sim m_5$ 得到 $V_j^*(m_1) \sim V_j^*(m_5)$, 因 $V_j^*(m_5) \geq D^r$, 先从 S 中移去 $V_j^*(m_5)$ 得到 $S = \{m_1, m_2, m_3, m_4\}$, 然后划分 S , 得到 $N_{fp^r[0]}^n = \{m_1, m_2, m_4\}$, $N_{fp^r[1]}^n = \{m_1, m_3\}$, $N_{fp^r[2]}^n = \{m_2, m_3\}$, $N_{fp^r[3]}^n = \{m_3, m_4\}$, 根据式 (5) 计算得 $V_{fp^r[0]}^n(n) = 49.0384\text{ms}$, $V_{fp^r[1]}^n(n) = 46.8384\text{ms}$, $V_{fp^r[2]}^n(n) = 47.2384\text{ms}$, $V_{fp^r[3]}^n(n) = 47.0384\text{ms}$, 得 $N_j^n = N_{fp^r[1]}^n = \{m_1, m_3\}$.

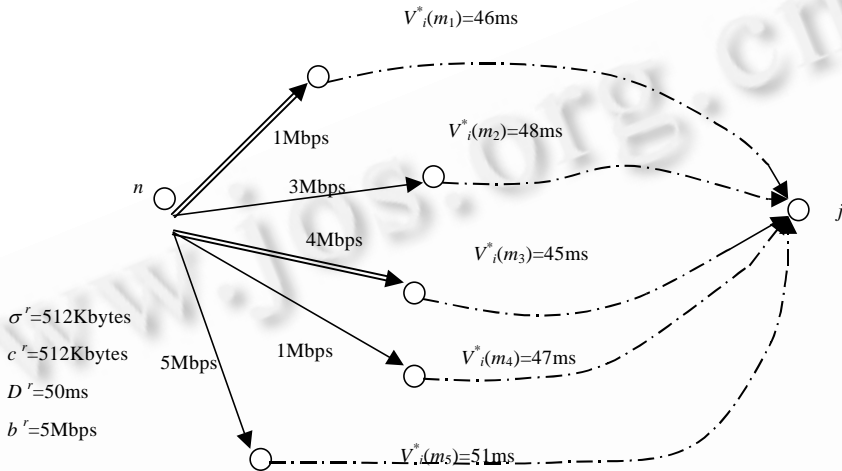


Fig.2 Calculating N_j^i

图 2 计算 N_j^i

2.3 多路选路算法

ITC 研究表明,现有网络中骨干网上业务量占总业务量的 80%.也就是说,大部分业务的连接需要经过边界路由器.在建立从源到目的端的多路径连接时(如文献[9]中的算法),为了消除回路,中间节点如果收到一个以上相同连接的请求包,中间节点只能转发一个请求包,其他的连接请求包将被丢弃.这样,在到达边界路由器之前域内的多路路由将变得没什么意义,连接失败率相应增加,因为最后的结果是只有一条通路穿过域边界.

本文提出的多路选择算法采用文献[6]中所提算法计算 S_j^i 和维护路由表状态,文献[6~8]已证明从 S_j^i 出发的多路在任何时候都无环路,又 $N_j^i \subseteq S_j^i$,所以从 A_j^i 出发的多路径在任何时候无环路.本算法对所有到达的连接建立请求包计算 A_j^i ,在准许接入的条件下,预留相应的资源,并沿 A_j^i 转发请求到 N_j^i 中所有的节点,所以我们提出的多路路由选择算法支持并行通路汇集,也即能建立虚拟不相交多通路.

下面详述建立连接的过程,然后给出过程的伪代码.为了支持路径合并,对某一连接请求 R ,节点 i 在第 1 次收到连接请求时,初始化连接 R 的连接数据结构(具体连接信息的意义见第 2.1 节的数据结构部分),然后计算当前连接请求的所有可行并路 FP^r 及 $V_{fp^r[]}(i)$,若可行并路存在,选择最小的 $V_{fp^r[]}(i)$ 为 $V_{AMP_j^i}(i)$,相应的 $N_{fp^r[w]}^i$ 为 N_j^i ,之后根据最小延迟路径优先使用的原则,预留节点 i 和 A_j^i 上的资源,修改延迟、带宽要求后,新建一个临时连接项 TPR 记录这些信息(包括 A_j^i 、在每条积极出弧上预留的带宽及 FP^r),最后沿 A_j^i 转发请求 R 到 N_j^i 中的所有节点.以后收到连接请求后,先计算是否存在可行并路,在可行并路存在的情况下,与第 1 次收到连接请求后面的处理相同.为了区分不同的临时连接项.我们使用连接请求包中的路径向量作为标识.若不存在可行并路,且不存在其他的 TPR 和 TEC 项,则释放初始的连接结构,并沿连接请求包中的路径向量后退到前一节点.当节点收到后退请求包后,先释放在原来 A_j^i 上预留的资源,然后检查此 TPR 的 FP^r 是否为空,在非空的情况下,选择 $V_{fp^r[w]}(i)$ 最小的 $fp^r[w]$ 为新的 N_j^i ,并从 FP^r 中移去 $fp^r[w]$,此后更新 TPR 并转发请求到 N_j^i ;若 FP^r 为空,则释放

此 TPR,并沿连接请求包中的路径向量再后退,直到连接请求的源点,且不存在其他可行并路情况下连接失败.当连接请求包到达连接请求的目的节点并成为预留需求的资源,然后沿请求包中的路径向量反向确认连接.节点的一个 TPR 项,在收到所有 N_j^i 中节点的确认信息后,把 A_j^i 上预留的资源合并到 TEC 中,并把此 TPR 的前序节点加入到 TEC 的输入节点集,释放此 TPR 项,然后沿路径向量向下一节点确认连接.在节点收到连接释放请求包时,先确认是否存在此连接请求的 TPR,若存在,释放 TPR 中在 A_j^i 上的预留的资源,并沿它的 A_j^i 发送释放请求包,最后释放 TPR 项;在不存在连接释放请求的相应 TPR 项,只存在该连接的 TEC 时(连接结束),释放 TEC 中所有预留的资源,并沿除此连接释放包来自的邻居节点外的所有输入节点和输出节点发送连接释放包.

算法 2. 处理路由连接请求.

Procedure main_setup

Begin

Do Sleep until i receives a routing packet

Switch (packet type)

Case request:

If it is the first routing packet for R then initiate connection data structure for R ;

End if

If $i == t^R$ then acknowledge(i_{prev}, R);

Else

calcu_AOS(R);

If $V_{AMP_j^i} < D^i$ then forward (N_j^i, R);

Else traceback(i_{prev}, R);

End if

End if

Break;

Case traceback:

Release($A_j^i \setminus R.sender, R$);

If exists other fp^r then $V_{AMP_j^i}(i) = \min_{fp^r[w] \in FP^r} (V_{fp^r[w]}(i));$

$N_j^i = fp^r[w];$ forward(N_j^i, R);

Else

If $i == s^R$ then reject route setup request R ;

Else

Remove this TPR;

traceback(i_{prev}, R);

End if

End if

Break;

Case acknowledge:

If i has received acknowledge packets from all node of A_j^i then

Merge this TPR into TEC;

Remove this TPR;

Acknowledge the previous node of i ;

End if

Break;

Case release:

If exists correspond TPR to this release packet then

release(A_j^i, R); remove TPR;

If TEC is empty and no other TPR then

remove connection data structure correspond to R

End if

Else if there is only TEC of R

For $u \in R.PREV \setminus R.NEXT$ \the neighbor of release packet comes from

Do release(u, R);

```

        End for
      Else discard this release packet;
    End if
  End Switch
End until
End Procedure

```

3 仿真实验

3.1 性能评价标准

以前的大多数路由算法都倾向于单一的连接优化延迟和路由距离.但是,除了显式路由方式以外,应用连接只要网络提供保证其服务质量所需的连接,并不关心具体的连接路径.因此,网络在保证连接的服务质量的同时,还应该提高网络资源的利用率,优化整个网络系统性能.从整个网络性能角度评价路由算法的参数主要有:

概念 8. 平均呼叫接受率(average call acceptance rate,简称 ACAR):接受实时通道建立请求的概率.

$$ACAR = \frac{\sum_{i=1}^N accepted(R)}{N} .$$

概念 9. 平均通信开销(average communication overhead,简称 ACOH):转发的所有寻路信息包与总连接请求数的比例.

$$ACOH = \frac{\sum_{i=1}^N tps(R)}{N} ,$$

其中 N 为总请求数,若连接 R 被准入 $accepted(R)=1$,否则 $accepted(R)=0$, $tps(R)$ 为连接请求过程中转发寻路信息包的次数.

3.2 仿真环境

我们通过对 DMPRA,PP(parallel probing)^[1]以及 DRA^[2]三者的仿真比较来评估算法的平均性能.在仿真实验中使用全双工,有向同构容量 155Mbps(OC3)链路简单连接网络,线路传播速度近似光速的 2/3.随机网络由改进的 ns-2 软件包生成.节点数为 50,节点间距离服从 10~10³km 间均匀分布,节点最小连接度为 2,平均连接度为 4,接近于当前网络的平均连接度.端到端的延迟由发送延迟和线路传播延迟决定*.我们主要比较了 3 种算法在不同网络负载情况下的平均接通率和相应的通信开销.

曲线上每点是 5 次仿真结果的平均值,每次仿真产生 1 000 连接请求.请求的产生服从以下规律:

- 呼叫的源和目的对以均匀的概率随机地从节点集中选取.
- 呼叫请求的带宽、持续时间、端到端的最大允许延迟均服从均匀分布.区间分别为[0.064~2]Mbps, [0.8~1]s, [100~200]ms.
- 每个节点连接请求的到达服从参数为 $\lambda(=0.5)$ 指数分布.

3.3 仿真结果

图 3 中的结果表明,在网络轻载时,每条线路上的可用带宽大多能满足连接请求,DMPR 与 DRA 通信开销相当,选择单路径即能满足连接要求.由于 PP 中间目的节点^[1]之间按轻负载优先(lightly loaded link first,简称 LLF)和最短路优先(shortest path first,简称 SPF)两种策略并行探测,PP 的通信开销接近 DMPR 的两倍.由于算法 PP 使用 LLF 能起到平衡负载的效果,PP 的平均接通率要好于 DRA.但是,PP 和 DRA 都只能使用单路径连接,随着网络负载的加重,每条线路上的可用资源减少,一些链路无法满足连接要求,致使更多的连接请求被拒绝.当 DMPRA 在每条链路上资源匮乏时,使用多条并路支持连接,从图 3(a)可以看出,在链路负载率 80%~85%时,接通

* 像其他很多算法仿真一样,这里忽略排队延迟.

率还维持在 70% 左右.在负载进一步增加的情况下,由于资源的极度匮乏,多条并路也不能满足连接要求,连接率急剧下降.因多路径要同时在多条虚拟不相交的通路预留资源,所以连接请求包数量随并路数的增加而增加.

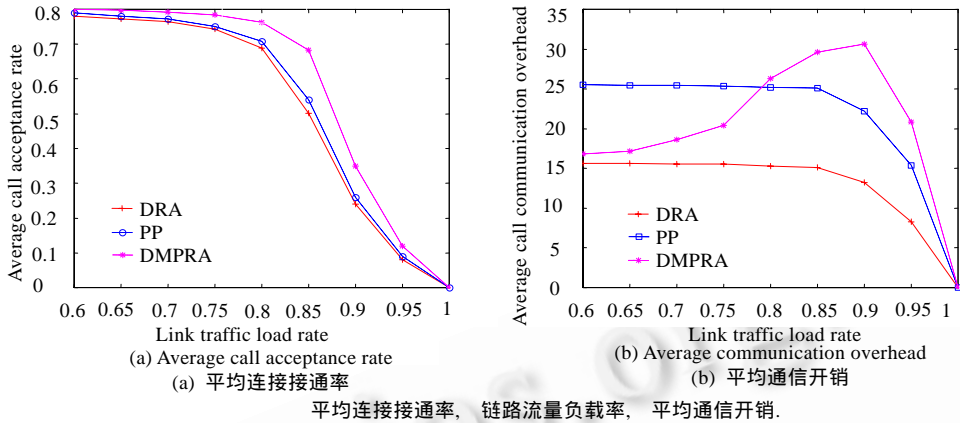


Fig.3 Simulation results

图 3 仿真结果

4 结束语

本文提出了虚拟不相交积极并路的概念,并且在无环多路路由算法^[6]的基础上给出了一种并行多路选择算法.算法在连接请求的源节点和目的节点之间选择建立满足带宽要求的最短并路.通过仿真实验与 PP,DRA 进行性能比较.结果表明,本文提出的算法能很好地适应网络运行状况,在轻负载的情况下,通信开销小;在网络重负载的情况下,使用并行多路增加连接请求的接通率.

多路路由近年来由于多媒体业务的大量出现,为了满足用户服务质量的要求而渐渐成为一个热点,但是还存在通信开销随通路数的增长而增长以及多通路传输数据同步的问题.

将此算法应用于支持服务质量保证的多媒体数据网,能提高网络传输多媒体信息的能力.

References:

- [1] Mainmaran, G., Rahul, H.S., *et al.* A new distributed route selection approach for channel establishment in real-time networks. *IEEE/ACM Transactions on networking*, 1999,7(5):698~709.
- [2] Chen, Shi-gang, Nahrstedt, K. Distributed quality-of-service routing in high-speed networks based on selective probing. In: Strayer, T., eds. *Proceedings of the 23rd Annual Conference on Local Computer Networks LCN'98*. 1998. 80~89.
- [3] Shin, K.G., Chou, C. Distributed route-selection scheme for establishing real-time channels. *IEEE Transactions on Parallel and Distributed Systems*, 2000,11(3):318~335.
- [4] Vogel, R., Herrtwich, R.G., Kalfa, W., *et al.* QoS-Based routing of multimedia streams in computer networks. *IEEE Journal on Selected Areas in Communications*, 1996,14(7):1228~1234.
- [5] Wang, Z. On the complexity of quality of service routing. *Information Proceeding Letters*, 1999,69(369):111~114.
- [6] Vutukury, S., Garcia-Luna-Aceves, J.J. MPATH: a loop-free multipath routing algorithm. *Microprocessors and Microsystems*, 2000, 24(6):319~327.
- [7] Vutukury, S., Garcia-Luna-Aceves, J.J. A simple approximation to minimum delay routing. *Computer Communication Review*, 1999,29(4):227~238.
- [8] Zaumen, W.T., Garcia-Luna-Aceves, J.J. Loop-Free multipath routing using generalized diffusing computations. In: Choudhury, A., Akyildiz, I., eds. *Proceedings of the IEEE INFOCOM'98*. New York: IEEE Computing Social Press, 1998. 1408~1417.
- [9] Cidon, I., Rom, R. Multi-Path routing combined with resource reservation. In: Miyahara, H., Suda, T., eds. *Proceedings of the IEEE INFOCOM'97*. Los Alamitos, CA: IEEE Computing Social Press, 1997. 92~100.

A Multi-Path Routing Algorithm with Optimal Bandwidth*

WANG Hong-bo¹, ZHANG Yao-xue¹, GUO Guo-qiang², GU Jun³

¹(Laboratory of Network Interconnection, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China);

²(Department of Computer Science, Changde Normal University, Changde 415003, China);

³(Department of Computer Science, Hong Kong Science and Technology University, Hong Kong, China)

E-mail: wuanghb@yahoo.com; zyx@moe.edu.cn

<http://sun475.cs.tsinghua.edu.cn>

Abstract: For realizing quality of service guarantee, all multi-user networks adopt the approach of reserving resource per connection requirement, but this will induce many 'resource pieces' in some parts of network, and traditional single routing approach cannot utilize these resources for new connection. In order to improve resource utilization ratio, a multi-path routing algorithm is proposed, which probes and reserves resources in one of several parallel paths according to the status of network and the connection Qos requirements. For verifying the validity, the proposed algorithm is implemented in the simulator of ns2 and its performances is compared with some algorithms. The simulator result shows that the proposed multi-path routing algorithm can well adapt to the status of network. Under the circumstance of the network light load, its communication overhead is smaller, closing to that of single path routing algorithms. Under the circumstance of the network overload, it can combine with several paths to admit new connection, so it improves the network resource utilization ratio and throughput. In conclusion, applying the proposed algorithm to connection-oriented multimedia data network will improve the capability of transmitting multimedia information.

Key words: multi-path routing; active out arc; Qos; virtual disjoint path; feasible parallel paths

* Received November 23, 2000; accepted May 9, 2001

Supported by the National Natural Science Foundation of China under Grant No.69873024; the National Grand Fundamental Research 973 Program of China under Grant No.G1998030406