

区分服务网络中确保业务端到端时延的接纳控制算法研究

史 琰 刘增基 盛 敏

(西安电子科技大学 ISN 国家重点实验室 西安 710071)

摘 要 该文提出了一种对区分服务网络中时延敏感业务进行接纳控制的 SACiD(Service Admission Control in DiffServ)算法,它不仅保证用户业务的端到端时延不超过确定的上界,而且能比较充分利用网络资源。仿真结果表明, SACiD 算法可以保证时延敏感业务的服务质量,而且其业务接纳率和带宽利用率接近综合服务网络中接纳控制算法的性能,但它的存储器需求却远小于它们。此外, SACiD 算法运算简单,具有与区分服务一致的可扩展性,可以为区分服务网络的资源管理提供支持。

关键词 区分服务, 接纳控制, 带宽分配, SACiD 算法

中图分类号: TN915.03

文献标识码: A

文章编号: 1009-5896(2006)05-0774-06

Study of Admission Control Algorithms with Deterministic Delay Guarantees in DiffServ Networks

Shi Yan Liu Zeng-ji Sheng Min

(ISN State Key Laboratory, Xidian Univ., Xi'an 710071, China)

Abstract An admission control algorithm with deterministic delay guarantees in DiffServ networks, Service Admission Control in DiffServ (SACiD), is developed in this paper. It can not only prevent the end-to-end delay of admitted services from violating the allowed upper bound, but also utilize network resources efficiently. Simulation results indicate that the performance of SACiD is close to that of the similar algorithm in IntServ networks with little memory requirement. Due to its simplicity and scalability, SACiD can support resource management in DiffServ networks.

Key words DiffServ, Admission control, Bandwidth allocation, SACiD (Service Admission Control in DiffServ) algorithm

1 引言

相对于综合服务(IntServ), 区分服务(DiffServ)具有简单和扩展性好的优点,被认为是下一代因特网中提供服务质量保证的重要机制。在区分服务网络中,边缘路由器根据用户业务的 QoS 要求,将用户业务流映射到预先定义的业务类,并设置对应的区分服务编码点(DSCP),核心路由器根据分组头中的 DSCP 对其进行区分服务。由于核心路由器不区分单个用户业务、不维护其状态信息,路由器功能被大大简化,可以将有限的处理能力集中于分组的快速转发。在区分服务模型中,EF(Expedited Forwarding)或 AF(Assured Forwarding)业务分组比 BF(Best effort Forwarding)业务分组得到相对好的服务质量,但对于具有确定性服务质量要求的业务,这种相对的服务质量优势可能会存在问题。

服务质量要求的满足依赖于网络资源的保证,而且QoS要求越严,其需要的网络资源越多^[1]。相对于用户业务的需

求来讲,网络资源总是稀缺的,必须对进入网络的用户业务进行接纳控制。因此,一些应用于区分服务网络的接纳控制算法被相继提出,但它们都有各自的缺点。文献[2,3]中的算法只能依据带宽要求来接纳业务请求,并不考虑其它 QoS 参数,如端到端时延。当网络处于重负荷时,这种接纳控制方法并不能确保已接纳业务的端到端时延不超出规定的上界。在文献[4,5]中,提出了具有统计性服务质量保证的接纳控制算法,它们是依据网络探测分组所接受的服务质量进行接纳控制。由于探测分组的服务质量可能受到突发数据的影响,导致其测量结果不准确,因此这种方法并不适用于具有严格服务质量要求的业务。

本文提出了一种区分服务网络接纳控制算法-SACiD (Service Admission Control in DiffServ),它不仅可以保证用户业务的端到端的时延不超过规定的上限,而且可以使网络资源得到较充分的利用。虽然 SACiD 算法是根据用户业务的 QoS 要求进行接纳控制,但是在网络结点中并不需要保存每个被接纳用户业务的状态信息和资源分配信息,所以 SACiD 算法具有与区分服务一致的良好扩展性。

2004-10-14 收到, 2005-03-17 改回

国家自然科学基金重大研究计划项目(90104012)资助课题

2 区分服务的确定性时延保证问题

2.1 问题描述

区分服务的优点是以模糊不同用户业务间的服务质量要求差异为代价的。如图1所示, S_1 和 S_2 分别为结点A和C到达结点E的分组话音业务, 它们具有相同端到端时延要求, 被标记为相同的业务类。因此, 结点D对这两个业务的分组作相同的处理。但实际上它们对结点D的处理要求是不同的。由于结点A到E具有较长的路径, S_1 的分组到达D时可能已经过较长的时间, 因此要求D进行较快的转发。而C到E的路径较短, S_2 的分组即使在D经历较大的时延, 也容易满足端到端时延要求。

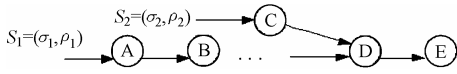


图1 区分服务的确定性时延保证问题示意图

当网络负荷较轻时, 尽管上述问题存在, S_1 和 S_2 的分组端到端时延都不会超过规定的上界, 但当网络处于中等或较重负荷时, 例如 S_2 的分组只能恰好满足端到端时延要求, 那么 S_1 的分组就很可能超过时延上界, 尽管它仍比同一路径上的BF业务有更短的时延。这就是“区分服务的确定性时延保证”问题。本文的SACiD算法就是针对这个问题提出的。

2.2 假设条件

假设网络运行 OSPF 路由协议, SACiD 算法所需要的网络链路状态信息在 OSPF 协议分组中携带, 并且具有最高的优先级, 不会在网络中丢失。因此, 根据 SACiD 算法的接纳控制可以在网络边缘结点完成。为了简化分析, 本文只考虑相同业务类中不同用户业务流间的相互影响, 因此后文中提到的链路容量只是对应于该类别业务的服务速率。

3 区分服务的确定性时延保证分析

3.1 结点时延模型

如图2(a)所示, R为网络结点, C为某输出链路上对应于时延敏感业务类的服务速率。n个相同类别的用户业务流在结点R被服务, 这n个流分别受漏桶参数 (σ_i, ρ_i) , $i=1, 2, \dots, n$ 的约束, 并满足 $\sum \rho_i \leq C$ 。

因此, 该类业务在结点R的最大时延D如式(1)所示^[6]。由于服务速率C被这n个业务流共享, 可以假设C被划分为n个虚服务速率 V_i , $i=1, 2, \dots, n$ (满足 $\sum V_i=C$), 其中 V_i 单独为 (σ_i, ρ_i) 服务, 如图2(b)所示。那么业务流i在 $[0, D)$ 内的最小平均服务速率如式(2)所示(因为D为最大时延, 所以平均服务速率为最小)。

$$D = \sum_{i=1}^n \sigma_i / \left(C - \sum_{i=1}^n \rho_i \right) \quad (1)$$

$$V_i = \frac{\rho_i D + \sigma_i}{D} = \rho_i + \sigma_i \left(C - \sum_{j=1}^n \rho_j \right) / \sum_{j=1}^n \sigma_j, \quad i=1, 2, \dots, n \quad (2)$$

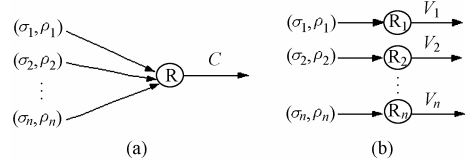


图2 区分服务网络的单结点时延分析

式(1)和式(2)表明: 如果每个业务流的虚服务速率按照式(2)进行分配, 则从最大结点时延的角度来讲, 图2(a)所示的网络结点模型可以等效为图2(b)所示的模型。由于区分服务网络的边缘路由器会对进入网络的业务流进行整形, 可以允许那些较高平均速率的业务具有较大的突发长度, 而那些较低平均速率只允许具有较低的突发长度。为了简化后面的分析, 假设 $\sigma_i/\rho_i = \sigma_j/\rho_j, \forall i \neq j$ 成立, 则式(2)可以被简化为式(3):

$$V_i = C \sigma_i / \sum_{j=1}^n \sigma_j = C \rho_i / \sum_{j=1}^n \rho_j, \quad i=1, 2, \dots, n \quad (3)$$

3.2 端到端时延模型

根据3.1节的分析, 图2(b)中的虚服务速率就是网络结点提供给每个用户业务的最小服务速率。因此, 可以看作网络结点采用PGPS(Packetized Generalized Processor Sharing)^[1]调度算法在进行服务。在这种条件下, 区分服务网络中的分组端到端时延可以由以下的分析得出。

图3表示区分服务网络中的一条路径, 它由m条链路组成, 其中链路l上分配给时延敏感业务类的, 的带宽为 C_l , $l=1, 2, \dots, m$ 。在图3(a)中, S为用户业务流 (σ, ρ) 的源结点, D为其目的结点, 而 (σ_l, ρ_l) , $l=1, 2, \dots, m$ 是经过第l条链路、与 (σ, ρ) 属于相同类别的聚合业务流。根据3.1节的分析, 在图3(b)中各条链路上对应于 (σ, ρ) 的虚服务速率分别为 $V_l=C\rho/(\rho+\rho_l)$, $l=1, 2, \dots, m$ 。在图3(a)和图3(b)中, 业务流 (σ, ρ) 在各个结点的最大时延是等效的, 由此可知其端到端的时延上界也是等效的。根据PGPS调度算法下的分组端到端时延公式^[1], 用户业务流 (σ, ρ) 的时延上界由式(4)中的等式表示, 其中 H_l 为链路l的容量。对于宽带高速网络而言, 可认为 $V_l \ll H_l$, $l=1, 2, \dots, m$ 成立, 则端到端时延近似由式(4)中的约等式表示。

$$D_{ete} = \frac{\sigma}{\rho} + \sum_{l=1}^m \left(\frac{L_{max}}{V_l} + \frac{L_{max}}{H_l} \right) \approx \frac{\sigma}{\rho} + \sum_{l=1}^m \frac{L_{max}}{V_l} \quad (4)$$

式(4)表明, 在区分服务网络中, 只要满足业务流在各条链路上的虚服务速率, 就可以保证该业务流的端到端时延不超过

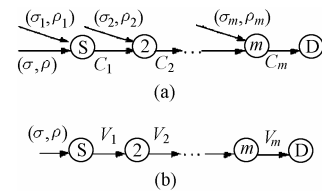


图3 区分服务网络中端到端时延分析

确定的上界。

3.3 新业务流对已接纳流 QoS 的影响

以图 2 为例, 假设一个新用户业务流 $(\sigma_{n+1}, \rho_{n+1})$ 被接纳, 根据式(3), 已接纳流 $(\sigma_i, \rho_i), i=1, 2, \dots, n$ 的虚服务速率会减小, 导致它们的端到端时延增加。但是, 只要它们的虚服务速率仍然大于其要求的服务速率, 就不会影响其服务质量。假设已接纳流 $(\sigma_i, \rho_i), i=1, 2, \dots, n$ 要求的服务速率分别为 $s_i, i=1, 2, \dots, n$, 而新业务流 $(\sigma_{n+1}, \rho_{n+1})$ 要求的服务速率为 s_{n+1} 。为了保证新接纳流和已接纳流的服务质量, 必须使式(5)成立:

$$V_i = \rho_i C / \sum_{j=1}^{n+1} \rho_j \geq s_i, \quad i=1, 2, \dots, n+1 \quad (5)$$

定义业务流加速参数 $A_i = s_i / \rho_i \geq 1$, 这个参数描述了业务流对网络带宽的要求, 这个值越大, 表明其所需的带宽越多, 时延要求越严。定义链路参数 $A_{\max} = \max\{A_i, i=1, 2, \dots, n\}$, 则从式(5)可以得到式(6):

$$\rho_{n+1} \leq \min_{1 \leq i \leq n} \left\{ \frac{\rho_i C - \sum_{j=1}^n \rho_j}{s_i} \right\} = \frac{C}{\max_{1 \leq i \leq n} \{s_i / \rho_i\}} - \sum_{j=1}^n \rho_j = \frac{C}{A_{\max}} - \sum_{j=1}^n \rho_j \quad (6)$$

可见, 如果 ρ_{n+1} 满足式(6), $(\sigma_{n+1}, \rho_{n+1})$ 就不会影响已接纳流的服务质量。如果新业务被接纳, 为了保证后续接纳控制的正常进行, 应该更新 $A_{\max} = \max\{A_{\max}, s_{n+1} / \rho_{n+1}\}$ 。从式(6)可知, 只要知道链路的 A_{\max} 和 $\sum \rho_i$, 就可以保证新业务流的接纳不会影响与该链路有关的已接纳业务流的QoS。更进一步, 如果知道新业务流路径上所有链路的 A_{\max} 和 $\sum \rho_i$, 就可以保证新业务流的接纳不会影响整个网络中已接纳流的QoS。

4 SACiD 算法分析

4.1 接纳控制

以图 3 为例, 假设新用户业务流 (σ, ρ) 要求网络提供服务, 其端到端时延要求为 Q_D 。SACiD算法的接纳控制过程分为以下的两个步骤:

(1)根据式(6)中的等式, 在新业务流路径的各个链路上计算允许的最大业务流平均速率 $\rho_l, l=1, 2, \dots, m$ 。如果 $\rho \leq \rho_l, l=1, 2, \dots, m$ 成立, 进入第(2)步判断; 否则, 拒绝该业务流。

(2)根据式(3)计算各段链路上对应于业务流 (σ, ρ) 的虚服务速率 $V_l, l=1, 2, \dots, m$, 然后带入式(4)计算端到端最大时延 D_{ete} 。如果 $Q_D < D_{ete}$, 拒绝该业务流; 否则, 接纳该业务流。

4.2 带宽分配

如果新业务流 (σ, ρ) 被接纳, 则表明图 3 所示的路径上可以提供满足其服务质量要求的资源。从另一方面讲, 一定量的资源应该以某种方式分配给新业务流, 从而确定性地保证其端到端的服务质量。对于网络的运营者来讲, 要求有限的网络资源可以为尽可能多的用户服务, 因此资源预留过程应

该有利于网络资源利用的最大化。根据式(6)可知, A_{\max} 描述一条链路的带宽使用状况, 它越大表明该链路的可分配带宽

就越小。因此, 可以使用 A_{\max} 的增加量来表示整个路径预留的资源。设业务流 (σ, ρ) 在各条链路上要求的服务速率为 (s_1, s_2, \dots, s_m) , 可以使用最优规划(P1)来求解它们。规划P1为

$$\begin{aligned} & \min \sum_{l=1}^m f(s_l) \\ \text{s.t. (a)} & \frac{\sigma}{\rho} + \sum_{l=1}^m \frac{L_{\max}}{s_l} \leq Q_D \\ & \text{(b)} \rho \leq s_l \leq V_l, \quad l=1, 2, \dots, m \end{aligned}$$

其中 $f(s_l) = \max\{0, s_l / \rho - A_{\max, l}\}$, $A_{\max, l}$ 为链路 l 的 A_{\max} 参数。

如果 $s = (\rho A_{\max, 1}, \dots, \rho A_{\max, m})$ 满足式P1-(a), 又根据式(6)可知 $\rho \leq \rho A_{\max, l} \leq V_l$, 因此 s 是规划P1的可行解。又因为 $f(\rho A_{\max, l}) = 0, l=1, 2, \dots, m$, 表明可行解 s 对应的目标函数值为0, 已经达到了规划P1目标函数值的下限。显然, $s = (\rho A_{\max, 1}, \dots, \rho A_{\max, m})$ 是规划P1的最优解。在解规划P1时, 首先观察 $(\rho A_{\max, 1}, \dots, \rho A_{\max, m})$ 是否满足式(P1-(a))。下面考虑 $(\rho A_{\max, 1}, \dots, \rho A_{\max, m})$ 不满足P1-(a)的情况, 并证明命题1。

命题 1 如果 $(\rho A_{\max, 1}, \dots, \rho A_{\max, m})$ 不能满足P1-(a), 则规划P1的最优解必满足P1-(a)中的等式。

证明 命题1等效为: 如果 $(\rho A_{\max, 1}, \dots, \rho A_{\max, m})$ 不能满足P1-(a), 则规划P1的不满足P1-(a)中等号的可行解一定不是其最优解。用反证法证明。假设 $s = (s_1, s_2, \dots, s_m)$ 是规划P1的最优解, 且不满足P1-(a)中的等式。根据已知条件和假设条件, 必存在 $i \in \{1, 2, \dots, m\}$, 满足 $s_i > \rho A_{\max, i}$ 。根据 s 可以构造 $s' = (s'_1, s'_2, \dots, s'_m)$, 其中 $s'_i = s_i, i \neq i$; $s'_i = \alpha \rho A_{\max, i} + (1 - \alpha) s_i$ 。由于 $\alpha (0 < \alpha < 1)$ 取值的连续性, 一定可以找到相应的 α 满足式(7)。因此, s' 也是规划P1的可行解。

$$\sum_{l=1}^m \frac{L_{\max}}{s'_l} < \sum_{l=1}^m \frac{L_{\max}}{s_l} \leq Q_D - \frac{\sigma}{\rho} \quad (7)$$

并且, 由于式(8)成立, 表明 s 不是规划P1的最优解, 这与假设矛盾。

$$\sum_{l=1}^m f(s_l) = f(s_i) + \sum_{l=1, l \neq i}^m f(s_l) > f(s'_i) + \sum_{l=1, l \neq i}^m f(s'_l) = \sum_{l=1}^m f(s'_l) \quad (8)$$

命题1有以下两个推论。由于它们的证明比较简单, 这里略去其证明过程。证毕

推论 1 如果 $(\rho A_{\max, 1}, \dots, \rho A_{\max, m})$ 不满足P1-(a), 唯一最优解 $s = (s_1, \dots, s_m)$ 满足 $s_l \geq \rho A_{\max, l}, l=1, 2, \dots, m$ 。

推论 2 如果 $(\rho A_{\max, 1}, \dots, \rho A_{\max, m})$ 不满足P1-(a), 唯一最优解使 $\sum_{l=1}^m s_l$ 最小。

根据推论2, 当 $(\rho A_{\max, 1}, \dots, \rho A_{\max, m})$ 不满足P1-(a)时,

可以用下述过程求得一个可行解:

```
loop from l=1 to l=m  s_l = \rho A_{\max, l};
set B = \emptyset;
```

```

loop from  $i=2$  to  $i=m$  {
   $F = \emptyset$ ;
  loop from  $j=1$  to  $j=i-1$  {
    if ( $V_j < \rho A_{\max,i-1}$ ) {
       $s_j = V_j$ ;   add  $j$  to set  $B$ ;
    } else if ( $V_j < \rho A_{\max,i}$ ) {
       $s_j = V_j$ ;   add  $j$  to set  $F$ ;
    } else
       $s_j = \rho A_{\max,i}$ ;
  }
  if ( $(s_1, s_2, \dots, s_m)$  satisfies (P1-(a))) break;
}
set  $k = i - 1$ ;
loop {
  flag = 0;
  solve  $x$  from equality:
  
$$\sum_{l=1, l \in B}^k \frac{L_{\max}}{x} + \sum_{l \in B} \frac{L_{\max}}{V_l} + \sum_{l=k+1}^m \frac{L_{\max}}{\rho A_{\max,l}} = Q_D - \frac{\sigma}{\rho}$$

  if (any  $i \in F$  satisfying  $V_i < x$ ) {
    add  $i$  to  $B$  and delete  $i$  from  $F$ ;   set flag = 1;
  }
  if (flag == 0) break;
}
loop from  $l=1$  to  $l=i$  and  $l \notin B$     $s_l = x$ ;
return solution  $(s_1, s_2, \dots, s_m)$ ;

```

我们将证明这个可行解就是规划P1 的唯一最优解。为了描述方便, 又不失一般性, 假设 $A_{\max,1} \leq A_{\max,2} \leq \dots \leq A_{\max,m}$ 。

命题 2 在 $(\rho A_{\max,1}, \dots, \rho A_{\max,m})$ 不能满足 P1-(a) 的条件下, 上述过程求得的可行解 $s=(s_1, \dots, s_m)$ 就是规划 P1 的最优解。

证明 使用反证法证明。设 s 不是最优解, 根据推论 1, 必有 $s'=(s'_1, s'_2, \dots, s'_m)$ 是规划 P1 的唯一最优解。根据已知条件, 一定存在 $\{1, 2, \dots, m\}$ 的非空子集 G 和 H , 满足以下 3 个条件: (a) $G \cap H = \emptyset$; (b) $s'_i < s_i, i \in G$ (c) $s'_j < s_j, j \in H$ 。又根据 s 的求解过程, 它具有以下的特点。其中, B, k 和 x 与上述过程中含义相同。

$$s_l = \begin{cases} V_l, & l \in B \\ x, & 1 < l \leq k, l \notin B \\ \rho A_{\max,l}, & k < l \leq m \end{cases}$$

而且, V_l, x 和 $\rho A_{\max,l}$ 满足 $\max_{l \in B} \{V_l\} < x < \min_{l \in \{k+1, m\}} \{\rho A_{\max,l}\}$ 。

显然, $H \cap B = \emptyset$ (V_l 已经是取值的上限) 和 $G \cap \{k+1, \dots, m\} = \emptyset$ 成立 (如果 $G \cap \{k+1, \dots, m\} \neq \emptyset$, 表明存在 $l \in \{k+1, \dots, m\}$ 满足 $s'_l < \rho A_{\max,l}$, 根据推论 1, s' 一定不是规划 P1 的最优解。)。因此, $\forall i \in G, \forall j \in H, s'_i < s_i < s_j < s'_j$ 成立。

由于 s 和 s' 都满足 P1-(a), 因此式(9)成立。

$$\sum_{i \in G} \frac{1}{s_i} + \sum_{i \in H} \frac{1}{s_j} = \sum_{i \in G} \frac{1}{s'_i} + \sum_{i \in H} \frac{1}{s'_j} \Leftrightarrow \sum_{i \in H} \frac{s'_j - s_j}{s_j s'_j} = \sum_{i \in G} \frac{s_i - s'_i}{s_i s'_i} \quad (9)$$

若 $W_1 = \min_{j \in H} \{s_j s'_j\}$ 和 $W_2 = \max_{i \in G} \{s_i s'_i\}$, 则 $W_1 > W_2$ 成立。那

公式(9)可以变换为式(10):

$$\begin{aligned} \frac{\sum_{j \in H} (s'_j - s_j)}{W_1} &\geq \sum_{j \in H} \frac{s'_j - s_j}{s_j s'_j} = \sum_{i \in G} \frac{s_i - s'_i}{s_i s'_i} \geq \frac{\sum_{i \in G} (s'_i - s_i)}{W_2} \\ &\Rightarrow \sum_{j \in H} (s'_j - s_j) > \sum_{i \in G} (s_i - s'_i) \Leftrightarrow \sum_{i \in G} s'_i + \sum_{j \in H} s'_j \\ &> \sum_{i \in G} s_i + \sum_{j \in H} s_j \Rightarrow \sum_{l=1}^m s'_l > \sum_{l=1}^m s_l \end{aligned} \quad (10)$$

根据推论 2, s' 不是规划 P1 的最优解。因此, 命题 2 得证。 证毕

通过上述分析和证明, 可以计算出为确保接纳业务流在各条链路上应该预留的带宽 (s_1, \dots, s_m) , 然后更新各段链路上的 A_{\max} 和 $\sum \rho_i$ 参数, 不再需要记录各个业务流的状态信息。

4.3 带宽释放

如果一个平均速率为 ρ 的业务流结束了, 网络必须释放预留给该用户流的带宽。SACiD 算法采用了一种不精确带宽释放的方法, 更新相应网络结点的 A_{\max} , $\bar{A} = \sum S_i / \sum \rho_i$, $\sum S_i$ 和 $\sum \rho_i$ 参数, 它不仅很简单, 而且不会对 SACiD 算法的接纳控制性能产生不利的影响。

更新 $\sum \rho_i = \sum \rho_i - \rho$, 保持 \bar{A} 不变, 并使 $\sum S_i = \sum S_i - \bar{A} \rho$ 。由于网络结点不保留每个用户流的状态信息, 因而无法确定对应于 A_{\max} 的业务流是否结束。但从另外一方面讲, 用户业务流的生存期总是有限的, 随着时间的增加, 与当前 A_{\max} 对应的业务流结束的可能性越大。因此, 可以用式(11)来估计它。其中, T_L 为用户业务生存期参数, T_L 越大表明用户业务的平均生存周期越长, t_u 和 t_n 分别为上一次更新 A_{\max} 时刻和当前时刻。

$$A_{\max} = \begin{cases} \bar{A}, & t_n - t_u \geq T_L \\ A_{\max} - \frac{(A_{\max} - \bar{A})}{T_L} (t_n - t_u), & t_n - t_u < T_L \end{cases} \quad (11)$$

虽然式(11)估计的 A_{\max} 不是准确的, 但是如果其值一直在下降, 就表明没有新的业务流接入, 或者只是接纳了时延要求较宽松的业务流, 这对所有业务的确定性端到端时延保证不会有不利的影响。如果 A_{\max} 在接纳控制中被更新, 则 A_{\max} 是准确的, 就更不会对后续的接纳控制产生不利影响。在上述过程中, 保持 \bar{A} 不变的原因是: (1) 如果 $\sum \rho_i \gg \rho$ 成立, 明显 $\sum S_i \gg S$ 成立 (在接纳控制时可以拒绝那些使 A_{\max} 过大的业务流), 则 \bar{A} 近似不变; (2) 如果 $\sum \rho_i \gg \rho$ 不成立, 虽然 \bar{A} 的更新不准确, 但是由于整个链路的空闲带宽比较多, 不会对整个接纳控制的性能造成影响。

4.4 SACiD 算法的计算复杂度

SACiD 算法的计算复杂度包含两个部分: (1) 接纳控制与带宽分配计算复杂度; (2) 带宽释放计算复杂度。假设在区分服务网络中为接纳业务选择的路径长度为 m , 则对该业务进行接纳控制的复杂度为 $O(2m)$, 而带宽分配的计算复杂度为 $O((m-1)(m-2)/2) + O(m)$ 。因此, 接纳控制和带宽分配的总计算复杂度为 $O((m-1)(m-2)/2) + O(3m)$ 。SACiD 算法的带

宽释放算法比较简单,其复杂度为 $O(m)$ 。

5 计算机仿真与性能分析

5.1 仿真目的与条件

从第4节的分析可知, SACiD算法具有与区分服务模型一致的可扩展性。但是作为一种针对时延敏感业务的接纳控制算法,我们希望它在保证用户业务端到端时延的前提下,具有与综合服务的接纳控制算法相近的性能。因此,在本次仿真中,将SACiD算法与综合服务模型下实现接纳控制和带宽分配的LSS(Load-based Slack Sharing)算法^[7]进行性能比较。LSS算法根据链路负荷在相应的链路上进行比例分配。评估参数包括,端到端时延、业务接纳率以及带宽利用率。该仿真基于Windows平台,使用Visual C++开发的仿真程序。

仿真网络如图4所示,链路 L_2 、 L_3 和 L_4 的带宽为80Mbps,其它链路均为40Mbps。结点(1,6)、(7,8)和(9,10)为3对源宿结点,用户业务从源结点产生,到达相应的目的结点。为了简化仿真,各个源产生相同类型的业务流,它们均受漏桶参数 (σ, ρ) 的约束,其中 $\sigma=4\text{ kbit}$, $\rho=100\text{ kbps}$,业务端到端的时延要求均为20ms。各个源结点的业务到达服从平均间隔为 $1/\lambda$ (λ 为业务流到达率)的指数分布,持续时间服从均值为25s的指数分布。此外,网络中分组长度服从[64byte, 250byte]间的均匀分布。在仿真中,比较了多种网络负荷情况下(λ 从0.5到4.5)SACiD算法与LSS算法的性能差别,仿真时间为10000s。此外,在网络结点处设置了足够的缓冲区,因而不存在分组丢失现象。

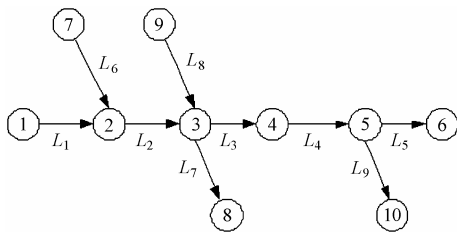


图4 仿真网络示意图

5.2 结果与分析

首先对SACiD和LSS算法接纳控制的效果进行比较。

表1 不同算法下接纳业务的平均端到端时延

到达率(λ)		0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5
平均时延(ms)	LSS	2.45	5.12	6.72	8.23	9.47	10.42	11.35	12.27	12.95
	SACiD	2.39	5.04	6.63	8.44	9.75	10.63	11.58	12.36	13.11

表2 不同算法下的网络总接纳率

到达率(λ)		0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5
接纳率	LSS	1.0	1.0	0.998	0.970	0.940	0.903	0.860	0.805	0.754
	SACiD	1.0	1.0	0.997	0.943	0.901	0.861	0.817	0.758	0.703

当这两种接纳控制算法被分别执行时,被接纳用户业务的端到端平均时延如表1所示,其中的值是对3个目的结点的统计结果进行平均得到的。从表1可见,在SACiD和LSS算法下,被接纳业务的端到端时延都没有超过规定的时延上界,也就是说,两种接纳控制算法都可以确保用户业务的端到端时延。

表2和图5显示了SACiD算法和LSS算法的性能。表2列出了在不同业务到达率下的网络总接纳率,而图5是链路 L_2 、 L_3 和 L_4 的平均带宽利用率随网络负荷增加的变化曲线。平均带宽利用率是通过以下方式计算得到的:在网络稳定后,以固定时间间隔(本次仿真中为100s)对带宽利用率进行采样,最后通过统计平均得到。

从表2可见,SACiD算法的接纳率略低于LSS算法,而且它们都随着网络负荷的增加而下降。这也是接纳控制算法的目的,即防止过多的用户业务进入网络而影响已经接纳的服务质量。图5的平均带宽利用率变化曲线可以解释表2中两个算法在接纳率上的性能差异。相对于SACiD算法,LSS算法比较充分地利用网络带宽,由于3条链路的利用率比较接近,不会因为某条链路缺乏带宽而拒绝源结点的业务流。图5中SACiD算法仿真结果验证了前面章节的分析,即为了保证同业务类中不同业务流的端到端时延要求,必然会在某种资源浪费。虽然SACiD算法的性能略低于综合服务模型下LSS算法,但SACiD算法可以大大节省核心路由器的存储器空间,避免业务流区分带来处理复杂度,我们认为SACiD算法的总体性能是优于综合服务体制下的接纳控制算法的。

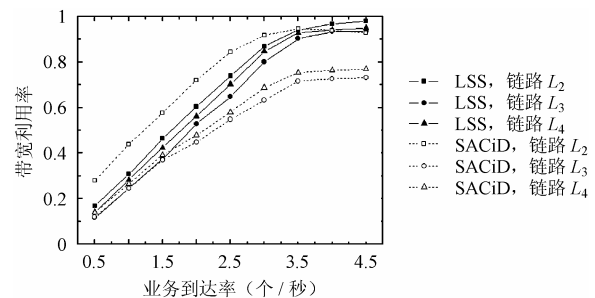


图5 平均带宽利用率

6 结束语

根据区分服务网络中的确定性端到端时延保证问题, 本文提出了一种接纳控制算法, SACiD 算法。它不仅可以保证被接纳的时延敏感业务不会超过确定的端到端时延上界, 而且不需要在网络的核心结点处保存被接纳业务流的状态信息。此外, 仿真结果也表明, SACiD 算法可以比较充分地利用网络资源, 获得接近于综合服务体制下接纳控制算法的接纳率和带宽利用率。由于该算法具有区分服务体制相一致的可扩展性, 它可以为区分服务网络中的资源管理提供有效的支持。

参 考 文 献

- [1] Zhang H. Service disciplines for guaranteed performance service in packet-switching networks. *Proce. IEEE*, 1995, 83(10): 1374 – 1396.
- [2] Bhatnagar S, Nath B. Distributed admission control to support guaranteed services in core-stateless networks. *IEEE INFOCOM 2003*, San Francisco, USA, 2003, 22(1): 1659 – 1669.
- [3] Blefari-Melazzi N, Femminella M. Stateful vs. stateless admission control: Which can be the gap in utilization efficiency? *Proc. of IEEE GLOBECOM 2002*, Taipei, China, Nov. 2002, 3: 2553 – 2557.
- [4] Tse D, Grossglauser M. Measurement-based call admission control: Analysis and simulation. *Proc. of IEEE INFOCOM 1997*, Kobe, Japan, April 1997: 981 – 989.
- [5] Breslau L, Jamin S, Shenker S. Comments on the performance of measurement-based admission control algorithms. *Proc. of IEEE INFOCOM 2000*, Tel Aviv, Israel, March 2000: 1233 – 1242.
- [6] Schwaetz M. *Broadband Integrated Network*, Prentice Hall, 1996:252 – 254.
- [7] Gopalan K. Efficient network resource allocation with QoS guarantees. Ph.D. Dissertation Proposal, Department of Computer Science State University of New York at Stony Brook Stony Brook, NY, 11794-4400, 2003.

史 琰: 男, 1975年生, 博士生, 研究方向为现代通信网络中的路由技术、QoS保障机制以及流量管理。

刘增基: 男, 1937年生, 教授, 博士生导师, 中国通信学会会士, 当前主要从事宽带网络技术的研究。

盛 敏: 女, 1975年生, 博士, 副教授, 当前主要从事移动 AD HOC 网络、Sensor 网络及无线个人通信的研究。