

基于收入值的 IP 网络带宽分配研究

吉 萌^{1,2}, 余少华^{1,2}

(1 华中科技大学计算机学院, 湖北武汉 430074; 2 武汉邮电科学研究院, 湖北武汉 430074)

摘 要: 现有 IP 网络采用尽力而为的方式传送数据, 存在可运营性和可管理性差的问题. 未来的 IP 网络必须能够对带宽进行有效的分配, 保证对用户的服务质量, 并实现运营商利益的最大化. 本文提出一种基于收入值的通用带宽分配模型, 将基于收入值的带宽分配问题分为完全满足带宽需求的严格带宽分配问题和部分满足带宽需求的灵活带宽分配问题, 并证明其等效于背包问题. 由于现有算法无法应用于实际的嵌入式系统, 本文提出了一种改进贪婪算法 (EGA) 对基于收入值的带宽分配问题进行求解. 在交换容量为 128Gbps 的分布式路由器上的实验结果表明, EGA 是一种兼顾公平性和有效性的算法, 能够在确定性时间内执行完毕. 采用 EGA 的网络设备已运行于实际的运营商网络中.

关键词: 基于收入值的带宽分配; 通用带宽分配模型; 改进贪婪算法

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2006) 08-1502-06

The Research on Revenue-based Bandwidth Allocation in IP Networks

JIM eng^{1,2}, YU Shao-hua^{1,2}

(1. School of Computer Science, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China;

2 Wuhan Research Institute of Posts and Telecommunications, Wuhan, Hubei 430074, China)

Abstract The current IP network utilizes best-effort mechanism to satisfy bandwidth requirements lacking of manageability and maintainability. The next generation IP networks should have the ability to allocate bandwidth efficiently to satisfy different QoS requests from different users and maximize the revenue for carriers. This paper proposes a generic revenue-based bandwidth allocation model and presents the RBA (Revenue-based Bandwidth Allocation) problem can be classified into FRBA (Flexible RBA) and SRBA (Strict RBA) problems. This paper proves that bandwidth allocation problem is equivalent to Knapsack problem and provides Enhanced Greedy Algorithm (EGA) to resolve it. The experiments on a 128Gbps distributed router show that EGA is fair and can be calculated in polynomial time. EGA has been implemented on the realworld network devices which are running on the carrier's network.

Keywords revenue-based bandwidth allocation; generic bandwidth allocation model; enhanced greedy algorithm

1 引言

Internet 应用的持续增长使得 IP 成为提供数据业务的首选协议. 而随着网络的融合, IP 网络已逐渐成为承载语音、视频和数据等多种业务的基础网络 (Infrastructure). 电信运营商 (Carrier) 也随之对现有 IP 网络进行扩容或建设新的 IP 网络, 但很快发现其运营收入并未随着带宽的增长而线性增长. 这是因为现有 IP 网络中的带宽分配 (Bandwidth Allocation or BA) 不是基于收入值 (Revenue) 进行分配, 无法满足运营商增加业务收入的需求.

目前, IP 网络中的带宽分配是基于三种服务模型: 尽力而为^[1] (Besteffort)、集成服务^[2] (IntServ) 和区分服

务^[3] (DiffServ). 在尽力而为模型中, 应用程序不需要事先建立连接, 就能够任意发送数据, 而路由器对所有的报文都均无区别地等同对待, 采用先入先出 (FIFO) 的方式进行处理. 尽力而为适用于绝大多数的现有 IP 网络应用如 HTTP 和 Email 等, 但无法满足实时业务对服务质量的需求. 相比之下, 基于 RSVP 协议的集成服务模型, 借鉴了电信网中 PSTN 的思想, 为每个业务流进行端到端的带宽预留, 可以保障不同流的服务质量. 但由于 IP 网络中的业务流具备突发性和不确定性的特点, 支持 RSVP 的核心路由器可能瞬间要为成千上万的业务流提供服务, 不具备可扩展性和实用性. 为了解决可扩展性问题, IETF 于 1998 年提出了区分服务模型. 区分服务采用了两级粗粒度的 QoS 方

式,网络边缘的节点实现复杂的分类和调节功能,而网络核心节点完成简单的逐跳行为(PerHop Behavior or PHB),避免了在核心节点对每一个流进行带宽预留。由于区分服务具有较好的可扩展性,当前学术界的很多研究^[4-8]集中在如何在区分服务域中进行有效的带宽分配。

随着新的网络技术的引入,带宽分配需要在链路层甚至物理层进行,如基于弹性分组环(Resilient Packet Rings or RPR)和无源光网络(Passive Optical Network or PON)的带宽分配^[9,10]。因此,现有的带宽分配机制存在以下局限性:(1)现有的带宽分配并非优先考虑收入值,无法满足运营商对增加业务收入的需求;(2)尽管区分服务是一个有效的 QoS 模型,由于它需要解析 IP 包头的区分服务编码点(DS codepoint or DSCP)字段,它无法解决三层以下网络的带宽分配问题。

本文提出了一种通用的基于收入值的带宽分配模型。它具备以下特点:首先,它是一个通用模型,逻辑上独立于具体物理实体,可以用于链路层甚至物理层的带宽分配中;其次,它基于收入值进行带宽分配,可以满足运营商的收入最大化(Maximal Revenue)需求。根据这一模型,我们将基于收入值的带宽分配问题(Revenue-based BA or RBA)分为严格的 RBA(Strict RBA or SRBA)问题和灵活的 RBA(Flexible RBA or FRBA)问题。我们证明了 RBA 问题理论上等效于背包问题,并提供了解决该问题的有效算法。本文其余部分具体组织如下:第二部分提出了通用的 RBA 模型;第三部分论述了如何用改进贪婪算法(Enhanced Greedy Algorithm or EGA)来解决 RBA 问题;第四部分介绍了我们在一台 128Gbps 交换容量的分布式路由器上的实验和结果分析;最后一部分总结了全文并对未来的工作进行了展望。

2 基于收入值的通用带宽分配模型

在 IP 网络中,进行带宽分配的实体是多种多样的。它们可以是路由器、交换机、宽带接入服务器(Broadband Remote Access Server or BRAS)或者是弹性分组环设备。而用户的接入方式也具有多种形式,如以太网、xDSL 或光纤接入。因此,一个通用的 RBA 模型应该满足以下要求:其一,它应该在逻辑上独立于特定的设备或底层技术;其二,它能够应用于不同的应用环境。

图 1 是本文抽象的通用

RBA 模型。在该模型中,Client 表示带宽请求的最小单元,它包含以下服务请求参数:带宽请求值(Requested Bandwidth)、带来的收入值

(Generating Revenue)和业务等级(Class of Services)。其中,业务等级决定了带宽的收入密度(Revenue Density)。在本文中,带宽的收入密度被定义为收入值和带宽请求值的

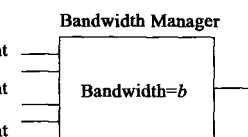


图 1 通用 RBA 模型

比值,即单位带宽的收入值。需要注意的是,IP 网络中的一个具体用户(如宽带接入服务器中的一个接入用户)可能包含多个不同业务等级的 Client 因为一个用户可能同时发起语音、视频和数据的多种不同业务等级的请求。

该模型的具体描述如下:给定一个正整数 k , 一个带宽容量为 b 的链路和 Client 集合 $\{e_1, e_2, \dots, e_k\}$; 该 Client 集合的业务等级为 $\{c_1, c_2, \dots, c_k\}$, 请求的带宽为 $\{b_1, b_2, \dots, b_k\}$, 能够带来的收入值为 $\{r_1, r_2, \dots, r_k\}$, 由带宽管理器(Bandwidth Manager or BM)进行管理;当这个 Client 动态地向带宽管理器进行请求时,带宽管理器能否进行最优化的带宽分配,并获取最大的收入值。

当 BM 必须完全满足各个 Client 的带宽需求时,我们称之为严格的 RBA(SRBA)问题;而当 BM 可以部分满足 Client 带宽需求时,我们称之为灵活的 RBA(FRBA)问题。

其中,带宽管理器可以通过用户和网络提供商的用户网络接口(User Network Interface or UNI)或认证信息来获取每个用户中的服务协定(Service Level Agreement or SLA)信息,从而识别不同的 Client 的服务请求参数。由于本文的研究对象为与平台无关的带宽分配问题本身,而服务请求参数的获取和具体的系统相关,因此,其不在本文讨论的范围之内。

显然,上述模型符合上文中提到的对通用带宽分配模型的要求:首先,它是一个单节点的模型,不需要和其他节点进行信令交互,也和具体的物理设备和底层技术无关;其次,该模型不需要解析具体的分组内容(如 DiffServ 中的 DSCP 域),因此可以应用于三层以下的的应用环境,并可以和 DiffServ 模型结合使用,用于 DiffServ 域的边缘节点中。这是因为 DiffServ 域中的边缘节点可以通过该模型进行基于收入值的带宽分配,并根据业务等级信息修改分组的 DSCP 域,并供 DiffServ 域中的核心节点进行高速转发。

3 求解 RBA 问题的算法

3.1 对求解 RBA 问题算法的要求

带宽管理器中一个有效的 RBA 算法必须具备以下特点:

(1)确定性(Deterministic)。由于请求服务的 Client 的带宽需求在实时变化,因此,为了实现动态带宽分配,RBA 的算法必须能够在确定性的时间内实现。该算法的时间复杂度不能是请求服务的 Client 个数的 k 的指数形式,必须是在多项式(Polynomial)时间内能够计算完毕,便于嵌入式系统的实现。

(2)公平性(Fair)。带宽管理器对各个 Client 带宽的分配应该是公平的。在带宽资源充裕的条件下,按需分配;而在用户带宽的请求量超过总带宽时,按一定的算法进行分配,但需要考虑对不同业务等级带宽分配的公平性。在进行 SRBA 时,应该优先满足高业务等级请求的带宽要求,即保证对高付费用户的公平性和运营商利益的最大化;而

在 FRBA 时, 应该考虑对低业务等级的 Client 的公平性, 防止出现饥饿 (Starvation) 现象, 并保证相同业务等级的 Client 之间的公平性.

3.2 RBA 问题和背包问题的等效

首先, 我们更准确地对 RBA 问题进行描述:

条件: 存在一个正整数 k 和编号为 $\{e_1, e_2, \dots, e_k\}$ 的 Client 集合; Client 的业务等级为 $\{c_1, c_2, \dots, c_k\}$, 请求的带宽值为 $\{b_1, b_2, \dots, b_k\}$, 能够带来的收入值为 $\{r_1, r_2, \dots, r_k\}$; 链路的带宽容量为 b , 收入值目标为 r .

问题: 带宽管理器能否进行最优化的选择, 使得产生的收入值总和超过 r , 且需求的带宽总和小于 b .

我们用设置变量 x_j 为 1 (或 0) 来表示加入 (或删除) 一个 Client e_j , 这样 SRBA 问题的每一个解可以用向量 $x = (x_1, x_2, \dots, x_k)$ 来表示. 我们用 Z^* 来表示 SRBA 问题的最优解, 则 SRBA 问题可由下式表达:

$$Z^* = \max \left\{ r(x) = \sum_{j=1}^k r_j x_j \mid b(x) = \sum_{j=1}^k b_j x_j \leq b, x_j \in (0, 1) \right\} \quad (1)$$

我们将给出证明 SRBA 问题是 NP 完全的, 不存在多项式时间算法.

定理 1 SRBA 问题是一个 NP 完全问题.

我们知道证明问题 L 是一个 NP 完全问题, 必须具备两个条件:

- (1) L 属于 NP;
- (2) 其他所有的属于 NP 的问题 L, 可以在多项式时间内规约至 L;

由于任何一个属于 NP 的问题 L 都可以在多项式时间内规约至一个已知的 (Well-known) NP 问题, 因此, 为了证明 SRBA 是一个 NP 完全问题, 我们只需要首先证明它属于 NP, 然后找到一个已知的 NP 问题 (如 3SAT 问题或 0-1 背包问题^[11]), 可以在多项式时间内规约至 SRBA.

从对 SRBA 问题的描述, 我们发现它和 0-1 背包问题非常相似. 0-1 背包问题的描述如下:

条件: 存在一个有限集 U , 对应每个物品 $u \in U$, 存在正整数重量 $w(u)$ 和收益 $p(u)$; 此外, 还存在正整数 W 和 P 分别为背包的容量和预期收益.

问题: 能否找到一个子集 $U' \subseteq U$, 使得其中物品的收益超过预期收益 P , 且总重量不超过背包的容量 W ?

以下, 我们给出对定理 1 的证明.

证明: (1) SRBA 问题属于 NP. 考虑一个图灵机 M , 它对于给定的 SRBA 问题中的 $\{b_1, b_2, \dots, b_k\}$, $\{r_1, r_2, \dots, r_k\}$, b 和 r , 非确定性地从 $\{0, 1\}$ 中赋值给 $\{x_1, x_2, \dots, x_k\}$, 检查是否 $\sum_{j=1}^k r_j x_j \geq r$, 并接收这组赋值当且仅当 $\sum_{j=1}^k b_j x_j \leq b$. 由于图灵机 M 是能够在多项式时间内完成, 因此, SRBA 属于 NP.

- (2) 从 0-1 背包问题到 SRBA 问题的多项式时间归

约. 我们可以证明 0-1 背包问题是 SRBA 问题的一个实例. 有限集 U 可以等效于 SRBA 中的 Client 集合; $w(u)$ 等效于 b_j , $p(u)$ 等效于 r_j , 背包容量 W 等效于 b , 收益目标 P 等效于 r . 以上归约可以在 $O(1)$ 时间内完成.

由式 (1)、(2) 可知定理 1 得证.

由于 SRBA 问题是 NP 完全问题, 因此, 不存在多项式时间内的确定算法可以求解出其最优解.

FRBA 问题的定义和 SRBA 问题的定义十分相似, 唯一的区别是将 SRBA 问题中的严格满足改为部分满足. 因此, FRBA 问题可由下式表达:

$$Z^* = \max \left\{ r(x) = \sum_{j=1}^k r_j x_j \mid b(x) = \sum_{j=1}^k b_j x_j \leq b, 0 \leq x_j \leq 1 \right\} \quad (2)$$

同样, 我们可以证明 FRBA 问题可以等效为连续背包问题. 限于篇幅, 此处证明略.

定理 2 FRBA 问题是一个 P 问题, 存在确定性算法在多项式时间内计算出最优解.

3.3 基于收入值的严格带宽分配问题求解

学术界已有很多的研究^[12-16] 集中于 0-1 背包问题的求解, 其中包括分支限界、遗传算法和启发式算法等; 实现的方式也包括基于单处理器的串行算法和基于多处理器的并行算法等. 由于并行算法主要应用于主机系统, 不适用于嵌入式的网络设备, 因此, 本文主要研究求解 0-1 背包问题的串行算法. 到目前为止, 考虑到运行时间, 最佳的串行算法仍然是由 Horowitz 和 Sahni 提出的二表算法 (Two-list Algorithm)^[14], 其时间复杂度为 $O(\sqrt{2})$. 然而, 当 n 比较大时, 其执行时间过长, 无法满足 3.1 节中提到的确定性要求, 不适用于实际的嵌入式系统. 因此, 考虑到实用性和可扩展性, 我们将尝试采用近似算法 (Approximate Algorithm) 求解 SRBA 问题.

贪婪算法是针对 0-1 背包问题的一种明显的近似算法. 贪婪算法的基本思想是将 Client 按照收入值或收入密度的非升序 (Non-increase) 进行排列, 并依次将 Client 放入输出集合中, 如果其请求的带宽不会超出剩余的带宽容量. 显然, 贪婪算法是确定性算法, 但其在最坏情况下性能较差. 例如, 考虑到仅有两个 Client 请求的情况. 其中, Client e_1 请求的带宽是 1, 业务等级是 3 带来的收入值也是 3 而 Client e_2 请求的带宽等于链路带宽, 业务等级是 1, 带来的收入值是 b . 采用基于收入密度的贪婪算法的执行结果是 e_1 , 而最优解应该是 e_2 .

因此, 我们提出了一种改进贪婪算法 EGA-SRBA 来求解 SRBA 问题. 和简单的贪婪算法不同, EGA-SRBA 进行两次贪婪算法选择: 第一次是进行基于收入密度的贪婪算法, 第二次是首先选择出收入值最大的 Client, 然后对其他 Client 进行基于收入密度的贪婪算法; 最后, 比较两次结果, 并将较优解输出, 具体算法见图 2.

```

Algorithm EGA-SRBA:
INPUT: link bandwidth  $b$ , the item bandwidth request  $\{b_1, b_2, \dots, b_k\}$ ,
CoS  $\{c_1, c_2, \dots, c_k\}$  and revenue  $\{r_1, r_2, \dots, r_k\}$ 
OUTPUT: Subset of the items
1. Sort the items in non-increasing order of their revenue densities of CoS  $(r_i/b_i \text{ or } c_i)$ .
2.  $U' \leftarrow \Omega$ 
3. for  $i = 1$  to  $k$  do begin /* first pass GA selection */
   if  $\sum_{j \in U'} b_j + b_i \leq b$ , then  $U' \leftarrow U' + i$ 
4.  $U'' \leftarrow \Omega$ 
5. Find the client  $e_m$  with largest revenue
6. if  $m \in U'$  then output  $U'$ ; return
   else  $U'' \leftarrow U'' + m$ 
7. for  $i = 1$  to  $k$   $i \neq m$  do begin /* second pass GA selection */
   if  $\sum_{j \in U'} b_j + b_i \leq b$ , then  $U'' \leftarrow U'' + i$ 
8. Compare the generated revenue from  $U'$  and  $U''$ , output the one with larger revenue
END

```

图 2 基于 SRBA 问题的改进贪婪算法

EGA-SRBA 包括一个排序阶段和两个基于收入密度的贪婪算法选择阶段。排序阶段可以由一个快速排序算法实现,其时间复杂度为 $O(N \lg N)$ 。而随后的两次基于贪婪算法的选择过程的时间复杂度均为 $O(N)$ 。因此,整个算法的时间复杂度为 $O(N \lg N + N + N) = O(N \lg N)$ 。而算法需要的存储空间为三个长度为 k 的数组,因此空间复杂度为 $O(N)$ 。

3.4 基于收入值灵活带宽分配问题求解

从定理 2 中,我们已知 FRBA 问题等价于连续背包问题,可以由基于收入密度(或业务等级)的贪婪算法在确定性时间内计算出最优解。但是,考虑到对算法公平性的要求,对于 FRBA 场景(Scenario),我们将兼顾对低业务等级和相同业务等级的 Client 的公平性。因此,我们为 FRBA 场景也提出了一种改进贪婪算法(EGA-FRBA)。EGA-FRBA 的具体步骤如下:

(1) 带宽管理器为所有的 Client 分配基本带宽进行数据传输。分配的带宽占有链路带宽的比例为 R , 分配方式采用加权轮转方式(Weighted Round-Robin 或 WRR)。

(2) 带宽管理器采用基于收入密度的贪婪算法将剩余的带宽进行分配。考虑到对相同收入密度的 Client 的公平性,EGA-FRBA 首先将具有相同收入密度的 Client 进行合并,然后对合并后的 Client 进行贪婪算法,具体算法描述见图 3。

EGA-FRBA 包括三个步骤:基本带宽的加权轮转分配、Client 重组和基于收入密度的贪婪算法选择。其中,加权轮转可以在线性的 $O(N)$ 时间内完成;重组过程可以由快速排序算法完成,时间复杂度为 $O(N \lg N)$,而最后的贪婪算法步骤的时间复杂度为 $O(N)$ 。因此,EGA-FRBA 的

```

Algorithm EGA-FRBA:
INPUT: link bandwidth  $b$ , the item bandwidth request  $\{b_1, b_2, \dots, b_k\}$ ,
CoS  $\{c_1, c_2, \dots, c_k\}$  and revenue  $\{r_1, r_2, \dots, r_k\}$ 
OUTPUT: Subset of the items
1. Allocate initial bandwidth of ratio  $R$  to each client in WRR manner
 $b_{init}(i) = B \times R \times (c_i \setminus \sum_{j \in U} c_j)$ 
2. Merge clients with the same revenue densities or CoS  $(r_i/b_i \text{ or } c_i)$ 
   into new set  $U'$  with bandwidth request  $\{b'_1, b'_2, \dots, b'_m\}$  and revenue
    $\{r'_1, r'_2, \dots, r'_m\}$  and CoS  $\{c'_1, c'_2, \dots, c'_m\}$ 
3. Sort the items in non-increasing order of their revenue densities or
   CoS  $(r'_i/b'_i \text{ or } c'_i)$  in  $U'$ 
4.  $U'' \leftarrow \Omega$ 
5. for  $i = 1$  to  $m$  do begin /* Greedy Algorithm */
   if  $\sum_{j \in U'} b'_j + b'_i \leq b \times (1 - R)$ , then  $U'' \leftarrow U'' + i$ 
6. Calculate the assigned bandwidth for each client
   for  $i = 1$  to  $k$  do begin
 $b_{allocate}(i) = b_{init}(i) + b_{GA}(i)$ 
END

```

图 3 基于 FRBA 问题的改进贪婪算法

时间复杂度为 $O(N + N \lg N + N) = O(N \lg N)$ 。此外,和 EGA-SRBA 相同,EGA-FRBA 的空间复杂度也是 $O(N)$ 。

4 实验和结果分析

4.1 实验平台

本文的实验平台采用我们为 863 重大课题“新型城域网关键技术和实验系统”开发的分布式多业务平台 DEEP (Distributed Extensible Services Platform)。

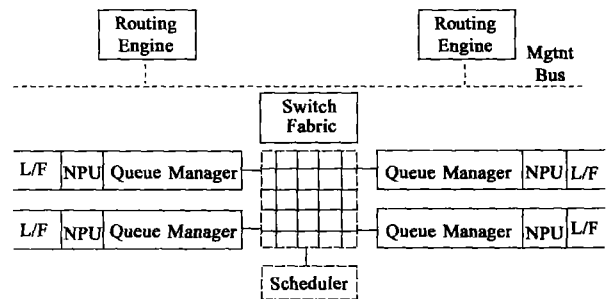


图 4 DEEP 体系结构简图

DEEP 采用分布式网络处理器结合高速 Crossbar 交换矩阵的体系结构,具有 128Gbps 的交换容量,可以作为高性能的路由交换设备和宽带接入服务器,并可以通过动态的带宽分配实现对 TDM 和视频等多业务的支持。图 4 是该设备的体系结构简图,其中路由引擎上运行系统软件,包括路由和信令协议;而线卡采用双网络处理器架构,实现高速的分布式转发。系统路由引擎上的主控 CPU 采用 Motorola 的 PowerPC 7410 处理板,该 CPU 采用 PowerPC 604 内核,带有 1Mbps 二级缓存,内存为 512Mbps。我们采用的操作系统为 Windriver 公司的 VxWorks 5.5,而采用的

编译系统是 Tornado 2.2 所带的 GNU C 编译器, 代码编译时采用三级优化的方式。

我们用 DEEP 模拟了一台典型的宽带接入服务器系统, 并测试了在接入的 Client 数为 10, 20, 40, 60, 100, 1, 000 和 10, 000 的情况下, 不同算法的执行时间和产生的收入值。在 SRBA 应用场景下, 我们比较了 EGA-SRBA、基于收入密度的贪婪算法和二表算法的执行结果; 在 FRBA 应用场景下, 我们比较了 EGA-FRBA 和基于收入密度的贪婪算法的执行结果。

其中, 每个 Client 的请求参数由软件随机产生。考虑到实际的应用情况, 我们设置的 Client 业务等级 C_j 的取值范围为 1 到 32, Client 请求的带宽 b_j 取值范围为 64 kbps 到 8 M bps 而每个 Client 产生的收入值 r_j 由 $(b_j \times c_j) / 64k$ 进行计算。此外, 带宽管理器管理的带宽为 100 M bps 在 EGA-FRBA 中的比例 R 设置为 20%。算法由 ANSIC 语言实现, 采用的排序算法为快速排序。

4.2 结果分析

表 1 和表 2 是各种算法在不同场景下的执行结果, 所有结果均为算法 100 次执行的平均值。从这些结果中, 我们分析了执行结果有影响的两个因素: 选择的 RBA 算法和 Client 数。

表 1 SRBA 场景下算法执行结果比较

Client Number	Execution Time		Generated Revenue	
	EGA-SRBA	Greedy Algorithm	EGA-SRBA	Greedy Algorithm
10	14.8us	13.4us	1946	1946
20	30.7us	29.6us	3872	4234
40	64.7us	58.5us	4296	4308
60	105us	96us	4670	4894
100	175us	160us	5607	5972
1,000	2.10ms	1.94ms	6847	7120
10,000	29.83ms	30.3ms	11644	13075

Note1: T W O-L S T represents two-list Horowitz and Sahni algorithm.

Note2: The result is meaningless because the execution time is unacceptable for critical missions.

表 2 FRBA 场景下算法执行结果比较

Client Number	Execution Time			Generated Revenue		
	EGA-FRBA	GA	T W O-L S T ⁽¹⁾	EGA-FRBA	GA	T W O-L S T
10	16.0us	14.6us	11.8us	1421	1421	1421
20	35.7us	30.8us	51.2us	3125	2693	3675
40	76.0us	67.3us	40.8ms	3345	2800	4210
60	119us	105us	41.0s	3527	2906	4429
100	177us	159us	— ⁽²⁾	3800	3152	—
1,000	2.37ms	2.14ms	—	6209	5467	—
10,000	35.67ms	32.23ms	—	10430	6846	—

从表 1 中, 我们可以看出, 在 SRBA 场景下, 二表算法能够带来最大的运营收入, 但由于算法的执行时间过长而不具备实用性 (Practicability)。因为当 Client 数目为 60 时, 二表算法执行时间已经达到 41.0 秒, 无法适用于实际应用中数千甚至上万的并发在线的 Client。通过表 1, 我们还可

以看出相对未改进的基于收入密度的贪婪算法, EGA-SRBA 能够提高 16~27% 的收入值。这说明我们提出的 EGA-SRBA 是有效的。此外, 我们发现 EGA-SRBA 的执行时间比改进前的贪婪算法运行时间长 8~11%, 这是因为尽管两者的主要执行时间是排序过程, 但是 EGA-SRBA 需要两次贪婪算法选择, 而基于收入密度的贪婪算法只需要一次。尽管如此, EGA-SRBA 的执行时间是确定性的, 在 Client 数为 10,000 算法的执行时间仍在毫秒级。由于执行时间可以通过采用更高频率的处理器或者用硬件固化算法来提高, 收入值的提高具有更大的实际意义。因为, 收入值提高 16~27% 意味着运营商在网络带宽不变的前提下, 净利润至少可以提高 15% 以上, 有效地提高了投资收益比。

从表 2 中, 我们可以看出, 在 FRBA 应用场景下, 基于收入密度的贪婪算法可以在确定性时间内获得最优解。这是因为 FRBA 是一个 P 问题, 存在多项式时间内的最优解。由于 EGA-FRBA 将比例为 R 的带宽进行加权轮转分配, 产生的收入值比基于收入密度的贪婪算法低 11~14%。但是, 我们可以通过降低 R 来提高 EGA-FRBA 的收入值。当 R 为 0 时, EGA-FRBA 将不需要进行加权轮转分配, 产生的最优值为最优解。此外, 由于 EGA-FRBA 相对基于收入密度的贪婪算法多进行了加权轮转分配和相同业务等级的 Client 重组过程, 执行时间也相比而言长 7~13%。同样, EGA-FRBA 的执行时间也是确定性的, 在 Client 数为 10,000 算法的执行时间仍在毫秒级。

和基于收入密度的贪婪算法相比, EGA-FRBA 主要增加了对公平性的考虑, 而由 3.1 节的分析可知公平性是 RBA 算法的基本需求之一。在 FRBA 应用场景中, 基于收入密度的贪婪算法虽然可以提供最优解, 但其无法避免低业务等级 Client 出现饥饿, 也没有考虑同一优先级的 Client 之间的公平性, 无法应用于实际环境中。而 EGA-FRBA 由于预先进行了基本带宽的加权轮转分配, 且将同等优先级的 Client 进行了合并, 可以使低业务等级 Client 具备基本带宽进行数据传送, 不至出现饥饿, 并能保障同等优先级的 Client 之间的公平性。

5 结论和未来工作展望

下一代的 IP 网络应该是可运营和可管理的网络, 因此, 它必须能够实现对带宽进行有效的动态分配, 并同时考虑到运营商的收入最大化。本文提出一种通用的基于收入值带宽分配模型, 并根据它将 RBA 问题划分为 SRBA 问题和 FRBA 问题。前者可以等效于 0-1 背包问题, 是一个 NP 完全问题, 不存在多项式时间的确定性算法求出最优解; 而后者可以等效于连续背包问题, 可以用针对收入密度贪婪算法求出最优解。本文提供了针对严格带宽分配问题的一种近似求解算法, 它可以在 $O(N \lg N)$ 的时间内求出近似解, 适合于嵌入式系统的应用; 而针对灵活带宽分配问题, 本文提出了一种改进贪婪算法, 考虑了对低业务

等级的 Client 和对相同优先级 Client 之间带宽分配的公平性。

实验表明, 我们提出的 EGA-SRBA 相对未改进的基于收入密度的贪婪算法可以提高 16~27% 的收入值; 而 EGA-PRBA 在 R 不为 0 时, 产生的收入值比最优化结果略低, 但它有效地兼顾了执行效率和公平性。此外, 以上算法均可以在确定时间内计算完毕, 采用 EGA 的网络设备已应用于实际的运营商网络之中。

我们注意到当 N 的规模较大时, 算法在实验平台上的执行时间较长 (数十毫秒)。因此, 我们下一步的工作是计划采用更高性能的嵌入式 CPU 或者用网络处理器和 ASIC 执行这一算法。此外, 考虑到算法可扩展性和快速性, 我们下一阶段将把 EGA 算法分布到各个线路接口卡上实现, 而不是集中在路由引擎上, 这样就可以避免主控 CPU 的性能瓶颈。

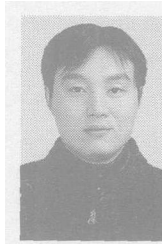
参考文献:

- [1] J Postel. Internet protocol [S]. IETF, RFC791, 1981.
- [2] R Braden, et al Resource reservation protocol [S]. IETF, RFC 2205, 1997.
- [3] S Blake, et al An architecture for differentiated service [S]. IETF, RFC 2475, 1998.
- [4] Seddigh N, Nandy B, Piedra P. Bandwidth assurance issues for top flows in a differentiated services network [A]. Proc of IEEE GLOBECOM '99 [C]. Rio De Janeiro, Brazil, 1999, 1792-1798.
- [5] Eun-Chan Park, Chong-Ho Choi Adaptive token bucket algorithm for fair bandwidth allocation in diffserv networks [A]. Proc of IEEE GLOBECOM '2003 [C]. San Francisco, 2003, 3176-3180.
- [6] Eun-Chan Park, Chong-Ho Choi Proportional bandwidth allocation in diffServ networks [A]. Proc of IEEE INFOCOM '2004 [C]. Hong Kong, 2004, 2038-2049.
- [7] 隆克平, 等. 区分服务结构及其 TCP 性能分析 [J]. 电子学报, 2001, 29(11): 1540-1545, 1548.
Long K P, et al Implementing mechanisms and tcp performance in diffserv capable ip Networks [J]. Acta Electronica Sinica, 2001, 29(11): 1540-1545, 1548 (in Chinese).
- [8] 刘正蓝, 朱森良, 汤旭红. 区分服务网络中带宽利用的公平性 [J]. 计算机学报, 2004, 27(1): 107-114.
Liu Z L, Zhu M L, Tang X H, Fairly realizing bandwidth

guarantees in diffserv networks [J]. Chinese Journal of Computers, 2004, 27(1): 107-114 (In Chinese).

- [9] Fu-Tai An, et al A new dynamic bandwidth allocation protocol with quality of service in Ethernet-based passive optical networks [A]. WOC 2003 [C]. Banff, Canada, 2003, 383-385.
- [10] Ahabie F, Ansari N. Low complexity distributed bandwidth allocation for resilient packet ring networks [A]. Proc of IEEE HPSR '2004 [C]. Phoenix, AZ, 2004, 277-281.
- [11] Garey M J, Johnson D S. Computers and intractability: a guide to the theory of np-completeness [M]. San Francisco: Freeman, 1979.
- [12] Martello S, Pisinger D, Toth P. Dynamic programming and strong bounds for the 0-1 knapsack problem [J]. Management Science, 1999, 414-424.
- [13] Martello S, Pisinger D, Toth P. New trends in exact algorithms for the 0-1 knapsack problem [J]. European Journal of Operational Research, 2000, 325-332.
- [14] E Horowitz, S Sahni. Computing partitions with applications to the knapsack problem [J]. Journal of ACM, 1974, 21(2): 277-292.
- [15] A G Ferreira. A parallel time/hardware tradeoff $T_H = O(2^{n/2})$ for the knapsack problem [J]. IEEE Transactions on Computers, 1994, 40(2): 221-225.
- [16] E D Kam in. A parallel algorithm for the knapsack problem [J]. IEEE Transactions on Computers, 1984, 33(5): 404-408.

作者简介:



吉 萌 男, 1976 年出生于河南信阳, 现为华中科技大学博士研究生, IEEE 会员, 主要研究方向为高性能 IP 网络系统和光因特网技术。

E-mail: jupite@hhu.com.cn



余少华 男, 1962 年出生于湖北武汉, 华中科技大学教授、博导, 武汉邮电科学研究院总工程师, 国际电信联盟第 15 研究组 (光和其他传送网络) 副主席, 曾获全国信息产业重大技术发明奖 3 项, 国家科技进步二等奖 2 项, 中国青年科技奖 1 项。目前主要从事城域网和 IP 网络相关技术研究。E-mail: shyu@hhu.com.cn