

基于边际效用函数的网络资源调度

宋亚楠^{1,2}, 仲 茜², 刘 斌¹

(1. 清华大学计算机科学与技术系, 北京 100084; 2. 72241 部队, 山东济南 250029)

摘 要: 为解决当前基于效用的网络资源调度中模型通用性不强、求解算法效果差、速度慢等问题, 提出了基于边际效用函数的效用最优资源调度方法. 它根据边际效用函数的特点, 将网络应用分为弹性和非弹性应用, 并通过每个应用的边际效用函数求出其效用函数. 将上述效用函数应用于网络资源调度问题中, 给出了问题求解的高效算法. 仿真实验表明, 与目前最新的同类算法及经典优化求解工具 Lingo9.0 中的算法相比, 本文算法求得的总效用值平均提高 5% 和 4%, 而所用时间仅为上述算法的 0.2% 和 0.003%.

关键词: 效用; 边际效用; 边际效用函数; 资源调度

中图分类号: TP393.05

文献标识码: A

文章编号: 0372-2112 (2013) 04-0632-07

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.3969/j.issn.0372-2112.2013.04.002

Marginal Utility Function Based Networking Resource Scheduling

SONG Ya-nan^{1,2}, ZHONG Qian², LIU Bin¹

(1. Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China;

2. 72241 Unit, Jinan, Shandong 250029, China)

Abstract: For the poor scalability, effectiveness and efficiency of the up-to-date utility based scheduling solutions, in this paper, we proposed a marginal utility function based scheduling method. In accordance with the feature of the marginal utility functions, the networking applications were divided into two classes, e. g. elastic and inelastic class. And then we built a utility based network resource scheduling model with the utility functions obtained from the corresponding marginal utility functions. The simulation experiments show that compare to the latest similar method and the corresponding approach in Lingo9.0, our algorithm increased the total utility by average of 5 and 4 percent with only 0.2% and 0.003% in the elapsed time.

Key words: utility; marginal utility; marginal utility function; resource scheduling

1 引言

为不同应用提供有区分的服务质量(Quality of Service, QoS)一直是网络资源调度的核心目标. 随着网络应用多样性和复杂性的增加, 网络资源调度面临新的挑战. 传统的基于带宽共享的资源分配策略的目标是使网络达到较高的平均吞吐量和较好的资源利用率, 但在考虑满足用户对网络所提供服务质量的满意度方面存在不足. 近年来, 人们发展了基于效用(Utility)的调度方法来改进网络的资源调度, 以更好地区分不同类别应用对网络资源的需求.

效用原本是经济学中的概念, 由 Shenker^[1]引入到网络研究中, 表示人们对所获得的网络服务质量的满意程度. 目前, 在不同的网络环境下, 研究人员提出了一些基于效用的资源调度方法^[2~10]. 在上述方法中, 通常首先通过协议分析提取网络应用特征, 将具有相似特征的应用聚合成一类, 并根据效用值与网络服务质量参数

(如带宽)之间的关系确定其效用函数; 然后根据效用函数得到基于效用的调度模型; 最后求解模型. 典型的研究如文献[8, 9]. 由于上述方法根据网络协议对网络应用进行分类, 可扩展性较差, 当引入新类型的网络应用时, 往往需重建模型. 针对这些方法的局限性, 文献[10]直接按已有网络应用的效用函数特征对其进行分类, 给出通用的效用函数, 简化了效用模型, 提高了模型的可扩展性. 然而, 求解模型时, 该方法采用迭代求解, 效率较低, 在实际应用中存在一定问题.

为了解决上述问题, 考虑到衡量效用的增减比直接确定效用函数值本身更准确, 本文引入边际效用函数(Marginal Utility Function)去研究效用函数. 边际效用是指在一定时间内消费者增加一个单位商品或服务所带来的新增效用, 也就是总效用的增量, 边际效用函数为其度量, 它是效用函数的导函数. 在经济学中, 对效用的度量, 通常采用基于边际效用分析的方法^[11]. 参照文献[1, 10], 本文将网络应用分为弹性和非弹性两类, 通过

边际效用函数求出效用函数,并应用于调度模型;在求解模型时,根据模型最优解的必要条件设计了求解算法 MU(Marginal Utility),并对算法误差和复杂性作了详细地分析.为了验证文中方法的求解效果(模型的总效用值)和速度,设计了仿真实验,将本文中的模型求解算法与文献[10]中的算法和经典最优化求解工具 Lingo9.0 中的对应的通用求解算法(通常求解精度高,但求解速度有时较慢)作了对比.结果显示 MU 算法平均总效用值比文献[10]中的算法高 5%,比 Lingo9.0 中方法高 4%;而平均求解时间只有 0.4s,为文献[10]中方法的 0.2%,是 Lingo9.0 中方法的 0.003%,表明本文方法在保证良好求解效果的同时具有很高的求解速度.本文的主要贡献有:(1)利用取最优值时边际效用函数值相等的特点,给出了求解算法,在有效提高求解效果的同时大大提高了求解速度;(2)引入边际效用函数,并利用其研究效用函数;(3)根据边际效用函数的特点,将各类网络应用分为弹性应用和非弹性应用,并给出各类应用的边际效用函数.

2 边际效用函数

边际效用函数是相关网络资源参数(如带宽)与边际效用间的映射关系,是度量边际效用的主要手段,记作 u ,是效用函数 U 的导函数.

2.1 基于边际效用的网络应用分类

根据边际效用函数曲线增减、凸凹的特点,参考文献[1,10],将网络应用分为弹性应用和非弹性应用.

弹性应用对于带宽没有严格要求,如 Web、email 等应用,当开始获得资源分配时边际效用达到最高值,随后边际效用开始下降,直至降为 0.而非弹性应用,如 VoIP、IPTV 等对带宽分配有较高的要求,若小于所需最低带宽,用户效用将变得很低甚至为 0.这类应用的边际效用值在达到最大之前,其边际效用值从 0 开始增长,到达最大之后边际效用随之下降.上述两类应用典型的边际效用曲线如图 1 所示.

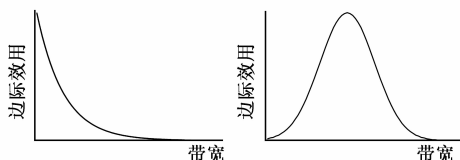


图1 弹性(左),非弹性(右)应用的边际效用函数曲线

2.2 弹性、非弹性应用的边际效用函数

本文选择连续型分布的概率密度函数作为边际效用函数,其原因在于:(1)效用函数与边际效用函数之间的关系与连续型分布的分布函数和密度函数之间的关系具有较大的可类比性,如效用函数和分布函数分别是边际效用函数和密度函数的原函数,且二者均为

值为 $[0,1]$ 的非递减连续函数;(2)已存在的概率分布函数具有良好的数学性质和明显的物理意义;(3)连续型分布的概率分布函数的数字特征,如:均值、方差等,能够很好地反映出函数曲线的某些特征,通过设置这些参数可以比较精确地拟合具体应用的效用函数.

2.2.1 弹性应用的边际效用与效用函数

根据对弹性应用边际效用的分析,采用指数分布的概率密度函数作为弹性应用的边际效用函数,如式(1)所示.

$$u_e(b) = \begin{cases} \lambda e^{-\lambda(b-\mu)}, & b \geq \mu \\ 0, & b < \mu \end{cases}, \quad (1)$$

$$U_e(b) = \begin{cases} 1 - e^{-\lambda(b-\mu)}, & b \geq \mu \\ 0, & b < \mu \end{cases}$$

其函数曲线如图 2 所示(带宽值为示意值),如不特别说明,本文图中带宽的单位均为 bps.由图 2 可见函数曲线符合弹性应用效用函数的特点.通过调整参数 λ , μ 的值可以得到一个函数簇,其中通过 λ 可调用增长快慢,通过 μ 可调整基本可用状态*的带宽.弹性应用效用函数曲线可涵盖文献[8]中 TCP 和 HTTP 两类效用函数和文献[1]中的弹性应用的效用函数曲线.

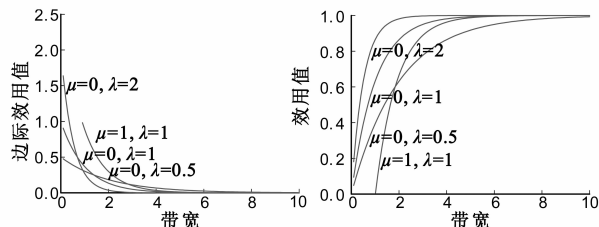


图2 弹性应用的边际效用函数与效用函数

2.2.2 非弹性应用的边际效用与效用函数

非弹性应用的边际效用函数曲线与高斯分布的概率密度函数曲线类似,但由于后者无法显式求出原函数(即:效用函数),本文中选择不具有相似曲线的函数来代替它.

$$u_{ne}(b) = \frac{\sigma \cdot e^{-\sigma(b-\mu)}}{(1 + e^{-\sigma(b-\mu)})^2}, \quad (2)$$

$$U_{ne}(b) = \frac{1}{1 + e^{-\sigma(b-\mu)}}$$

函数曲线如图 3 所示.其中, σ 表示梯度, μ 表示边际效用最大时的带宽.上述曲线可用于表示文献[8]中 VoIP 和 IPTV 两类效用函数和文献[1]中的三种实时应用的效用函数.

2.2.3 效用函数参数的确定

由式(1)、(2)和图 2、图 3 可知,对于一个具体的网络

* 基本可用状态是随着带宽的增长效用增长最快的点所对应的状态,反映在边际效用函数中是曲线的最大值点.

应用,需通过确定相关参数的值,获得实际的边际效用和效用函数.求解参数之前需首先确定两个量,即: b_{\max} 和 b_{\min} . b_{\max} 表示服务的最大带宽需求,它是使效用函数值达到最大(通常为1)的带宽需求,当实际带宽分配超过最大带宽需求后效用值也不再增加. b_{\min} 表示最小带宽需求,是网络服务达到可用时所需要分配到的最小带宽,实际分配的带宽若小于最小带宽需求效用值为0.

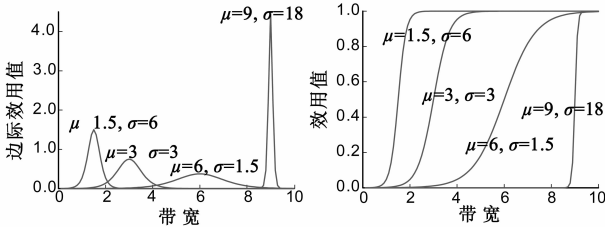


图3 非弹性应用的边际效用函数与效用函数

(1) 弹性应用的效用函数参数推导

由于弹性应用遵循 Best-Effort 原则,它追求网络吞吐量的最大化,因此通常可设置理论可用最大带宽作为 b_{\max} ,0作为 b_{\min} .另外,由于弹性应用的效用曲线随着带宽分配的增加而趋于1但不会达到1,于是引入参数 ϵ , ϵ 为一个很小的值,当效用值大于等于 $1 - \epsilon$ 时令其为1.参数 μ 和 λ 的求解过程如下:设当带宽分配为 b_{\max} 的时候,效用值为 $1 - \epsilon$;带宽为 b_{\min} 时,效用值为0.于是由式(1)可得:

$$1 - e^{-\lambda(b_{\min} - \mu)} = 0 \quad (3)$$

$$1 - e^{-\lambda(b_{\max} - \mu)} = 1 - \epsilon \quad (4)$$

式(3)、(4)联立得:

$$\mu = b_{\min}, \lambda = \frac{-\ln \epsilon}{b_{\max} - b_{\min}} \quad (5)$$

(2) 非弹性应用的效用函数参数推导

非弹性应用对带宽有较为严格的要求,如:VOIP要求带宽为64kbps与音频编码对应带宽系数的乘积,因此需根据协议中有关带宽的要求来设置 b_{\max} 和 b_{\min} .与弹性应用类似也引入 ϵ , ϵ 为带宽分配为 b_{\min} 时的效用值,当带宽分配为 b_{\max} 时,效用值为 $1 - \epsilon$.

由式(2)可得:

$$\frac{1}{1 + e^{-\sigma(b_{\min} - \mu)}} = \epsilon \quad (6)$$

$$\frac{1}{1 + e^{-\sigma(b_{\max} - \mu)}} = 1 - \epsilon \quad (7)$$

联立式(6)、(7)得:

$$\sigma = \frac{-2 \ln(\frac{\epsilon}{1 - \epsilon})}{(b_{\max} - b_{\min})}, \quad \mu = \frac{(b_{\max} + b_{\min})}{2} \quad (8)$$

3 模型与求解

本节将弹性、非弹性两类效用函数应用于网络资源调度模型中,并给出求解方法.

3.1 基于效用的资源调度模型

参考文献[8,9]给出本文的调度模型

$$\begin{aligned} \max U &= \sum_{i=1}^{N_1} U_i(b_i) + \sum_{i=N_1+1}^N U_i(b_i) \\ \text{s.t. } &\sum_{i=1}^N b_i \leq C \end{aligned} \quad (9)$$

其中, $U_i(b_i) = 1 - e^{-\lambda(b_i - \mu)}$ ($1 \leq i \leq N_1$), $U_i(b_i) = \frac{1}{1 + e^{-\sigma(b_i - \mu)}}$ ($N_1 + 1 \leq i \leq N$), U_i 为第 i 个流的效用函数, C 为总带宽, $N_1, N_2, N = N_1 + N_2$ 分别为弹性、非弹性和网络流的个数, b_i 代表流 i ($i \in [1, N]$)所分配的带宽.

3.2 模型求解

显然,式(9)模型属于非线性规划问题,通常难以求得全局最优解.因此在本节中首先利用拉格朗日(Lagrangian)方法和KKT(Karush-Kuhn-Tucker)条件对求解过程进行分析,然后给出一个高效的启发式近似求解算法.

3.2.1 最优解的必要条件

根据式(9)中的模型和Lagrangian方法,可以得到下列Lagrangian函数.

$$L(b_1, b_2, \dots, b_N, \lambda) = U - \lambda \left[\sum_{i=1}^N b_i - C \right]$$

基于KKT条件,有

$$\frac{\partial L}{\partial b_i} = U'_i(b_i) - \lambda = u_i(b_i) - \lambda = 0 \quad (i \in [1, N]) \quad (10)$$

$$\frac{\partial L}{\partial \lambda} = \sum_{i=1}^N b_i - C = 0 \quad (11)$$

其中, u_i 为 U_i 的边际效用函数.

由式(10)和(11)可以推出式(9)中模型最优解的必要条件.

必要条件 1 对于任意数对 i 和 j ($i \neq j, i, j \in [1, N]$),式(9)中模型全局最优解必需满足以下约束:

$$(1) \quad u_i(b_i) = u_j(b_j) \quad (i \in [1, N])$$

$$(2) \quad \sum_{i=1}^N b_i = C$$

3.2.2 模型求解

由于本文中两类边际效用函数均包含有指数函数,很难由式(10)和(11)求出模型的解析解.因此基于必要条件1,提出一个启发式近似算法.

首先通过图4中的例子来说明求解算法的基本思

路.图中包含 4 个网络流,其中, e_1, e_2 为弹性流, ne_3, ne_4 为非弹性流. 直线 $y = m_0$ 与 e_1, e_2 的边际效用函数曲线各有一个交点, 记为 (b_1, m_0) 和 (b_2, m_0) , 与 ne_3 和 ne_4 的曲线分别有两个交点其横坐标分别为 $b_{3,1}, b_{3,2}$ 和 $b_{4,1}, b_{4,2}$. 如果分配 $b_1, b_2, b_{3,1}$ 或 $b_{3,2}, b_{4,1}$ 或 $b_{4,2}$ 给 e_1, e_2, ne_3, ne_4 , 它们将获得相同的边际效用. 若此时 $b_1, b_2, b_{3,i}, b_{4,j}$ ($i, j = 1$ 或 2) 的和等于 C , 则上述带宽分配满足必要条件 1, 此时的解本文中称为模型的候选最优解. 由 3.2.1 知模型的全局最优解必满足必要条件 1, 于是考虑从 0 到 m (边际效用函数的最小的最大值, 见图 4) 按增量 Δ_m 遍历所有的候选最优解, 然后取其中的最大值作为模型的最优解.

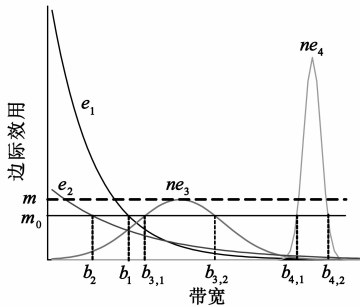


图4 示例

下面首先给出模型求解算法, 然后对求解算法的有效性作必要的证明.

(1) 式(9)模型求解算法

算法 1 式(9)模型求解算法

输入: $E = \{e_i \mid 1 \leq i \leq N_1, e_i \text{ 为弹性流}\}$, $NE = \{ne_j \mid N_1 + 1 \leq j \leq N, ne_j \text{ 为非弹性流}\}$; 边际效用增量 Δ_m

输出: $S = \{b_i \mid 1 \leq i \leq N, b_i \text{ 为分配给流 } i \text{ 的带宽}\}$

第 1 步: 令 m 等于 $E \cup NE$ 中所有流对应边际效用函数的最小的最大值, 设 $m_1 = \Delta_m, U = 0$.

第 2 步: 当边际效用为 m_k ($k > 1$ 且 $m_k < m$), 对于弹性流 i 直接求得其带宽 b_i ; 对于非弹性流 j , 在两个解 $b_{j,1}, b_{j,2}$ 中选择其一作为带宽 b_j , 使得流的带宽和等于或接近于 C . 如果此时的总效用 $U_k > U$, 则令 $U = U_k$, 并将各流的带宽存储到 S 中.

第 3 步: 若 $m_k = m$, 输出 S , 算法结束. 否则, $m_{k+1} = m_k + \Delta_m$, 若此时 $m_{k+1} > m$, 令 $m_{k+1} = m$, 转第 2 步.

(2) 算法 1 求解有效性证明.

定理 1 通过算法 1 可以获得本文模型的近似最优解.

证明 设 $S^* = \{b_i^* \mid i \in [1, N]\}$ 为通过算法 1 获得的解, b_i^* 为分配给流 i 的带宽, U^* 为对应的总效用值; 设 m_0 为模型最优解 $S^0 = \{b_i^0 \mid i \in [1, N]\}$ 时的边际效用函数值, U_0 为总效用值, 显然 $U_0 \geq U^*$; $Q = \langle m_1, m_2, \dots, m_n, m \rangle$ 为算法 1 的第 2 步中边际效用值序列.

若 $m_0 \in Q$, 则 $U^* \geq U^0$, 另一方面, $U^0 \geq U^*$, 于是有 $U^* = U^0$, 即 S^* 为全局最优解. 若 $m_0 \notin Q$, Q 中必存在 m_k ($m_k < m_0 < m_{k+1}$), 见图 5. 由图 5 可知边际效用取 m_k 与 m_{k+1} 时总带宽与边际效用取 m_0 时总带宽的差分别为

$$\begin{aligned} d_{0,k} &= \sum_{i=1, \dots, N} (b_i^0 - b_i^k) \\ &= \Delta_l \times \sum_{i=1, \dots, N} (\text{sgn}(b_i^0 - b_i^k) \times \tan \theta_i) \end{aligned} \quad (12)$$

$$\begin{aligned} d_{0,k+1} &= \sum_{i=1, \dots, N} (b_i^0 - b_i^{k+1}) \\ &= \Delta_u \times \sum_{i=1, \dots, N} (\text{sgn}(b_i^0 - b_i^{k+1}) \times \tan \theta_i) \end{aligned} \quad (13)$$

由于 b_i^0 介于 b_i^k 和 b_i^{k+1} 之间, 于是

$$\text{sgn}(b_i^0 - b_i^k) = -\text{sgn}(b_i^0 - b_i^{k+1})$$

这样式(13)可被转化为

$$\begin{aligned} d_{0,j+1} &= \sum_{i=1, \dots, N} (b_i^0 - b_i^{k+1}) \\ &= -\Delta_u \times \sum_{i=1, \dots, N} (\text{sgn}(b_i^0 - b_i^k) \times \tan \theta_i) \end{aligned}$$

因此, $d_{0,k}$ 和 $d_{0,k+1}$ 异号. 不失一般性, 这里不妨假设 $d_{0,k} \geq 0$ 且令 $B^k = \sum_{i=1, \dots, N} b_i^k$, 则 $B^0 - B^k = C - B^k = d_{0,k} \geq 0$, 于是 $B^k \leq C$, 又由于 $|b_i^0 - b_i^k| \leq \Delta_m \tan \theta_i$ 且在具体问题中 $\tan \theta_i$ 有界, 因此

$$\Delta_m \rightarrow 0 \Rightarrow b_i^0 \rightarrow b_i^k \Rightarrow d_{0,k} \rightarrow 0 \Rightarrow B^k \rightarrow C$$

解 $S^k = \{b_i^k \mid i \in [1, N]\}$ 为算法第 2 步中的候选最优解, 则 $U^k < U^*$, 于是

$$U^0 - U^* < U^0 - U^k = \sum_{i=1, \dots, N} (U_i(b_i^0) - U_i(b_i^k))$$

由于 U_i 单调增、有界且连续可微, 则

$$\Delta_m \rightarrow 0 \Rightarrow b_i^k \rightarrow b_i^0 \Rightarrow U_i^k \rightarrow U_i^0 \Rightarrow U_i^* \rightarrow U_i^0$$

因此可设置较小的 Δ_m , 使 U^* 为 U^0 的近似最优解, 证毕.

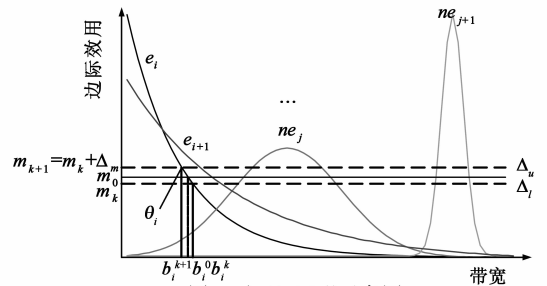


图5 定理1证明示意图

(3) 非弹性流解的选择算法

由算法 1 的描述知第 2 步为整个算法的关键, 其主要问题是如何选择非弹性边际效用函数的带宽, 使总效用最大. 该问题等效于式(14)中的模型描述:

$$\max U = \sum_{i=N_1+1}^N U_i(b_i^*)$$

$$\begin{aligned} \text{s. t. } b_i^j &= b_{i,1}^j \text{ or } b_{i,2}^j, (N_i + 1 \leq i \leq N) \\ \sum_{i=1}^N b_i^j &\leq C \end{aligned} \quad (14)$$

其中, $\epsilon \geq 0$ 且通常值较小. 下面通过证明上述模型与经典的 0-1 背包问题模型等价来说该模型求解问题是 NP 完全问题.

0-1 背包问题 给定一个背包, 其最大允许装载重量为 W , S 为拥有 n 个物品的集合, 其中第 i 个物品的重量为 w_i 价值为 b_i . 求在满足背包最大装载量的约束下, 选择装载哪些物品, 使所装载物品的总价值最大.

$$\begin{aligned} \max B &= \sum_{i=1}^n B_i(x_i) \\ \text{s. t. } x_i &= 0 \text{ or } 1 (1 \leq i \leq n) \\ \sum_{i=1}^n (x_i \cdot w_i) &\leq W \end{aligned} \quad (15)$$

其中, $B_i(0) = 0, B_i(1) = b_i$

定理 2 式(14)和式(15)中的模型等价.

证明 令 $b_i^j = a_i^j \times (b_{i,2}^j - b_{i,1}^j) + b_{i,1}^j (N_i + 1 \leq i \leq N)$, 式(14)模型与以下模型等价

$$\begin{aligned} \max U &= \sum_{i=N_i+1}^N A_i(a_i^j) \\ &= \sum_{i=N_i+1}^N (U_i(b_{i,2}^j) - U_i(b_{i,1}^j)) \\ \text{s. t. } a_i^j &= 0 \text{ or } 1, (N_i + 1 \leq i \leq N) \\ \sum_{i=N_i+1}^N a_i^j \times (b_{i,2}^j - b_{i,1}^j) &\leq C - \sum_{i=1}^{N_i} b_i^j - \sum_{i=N_i+1}^N b_{i,1}^j = D \end{aligned} \quad (16)$$

其中, $A_i(0) = 0, A_i(1) = U_i(b_{i,2}^j) - U_i(b_{i,1}^j)$. 显然, 令 $w_i = b_{i,2}^j - b_{i,1}^j, b_i = U_i(b_{i,2}^j) - U_i(b_{i,1}^j)$, 式(16)模型转化为式(15)模型. 另一方面, 式(15)中令 $w_i = w_i / (\max(w_i) + 1), W = W / (\max(w_i) + 1)$, 转化后的模型与原模型等价. 同样令 $w_i = b_{i,2}^j - b_{i,1}^j, b_i = U_i(b_{i,2}^j) - U_i(b_{i,1}^j)$, 带入式(2)可求得 U_i , 然后根据 W 设置 b_i^j 可将其转化为式(16)模型. 于是可知式(14)模型与式(15)模型等价, 证毕.

由于式(14)模型求解是 NP 完全问题, 文中通过以下贪婪算法对问题进行求解.

算法 2 非弹性流解的选择算法

第 1 步: 对于算法 1 第 2 步中的边际效用 m_k , 根据每个非弹性边际效用函数 $u_i (N_i + 1 \leq i \leq N)$ 计算相应的带宽, 此时得到的两个解分别记为 $b_{i,1}^k$ 和 $b_{i,2}^k (b_{i,2}^k \geq b_{i,1}^k)$. 令 $b_i^k = b_{i,2}^k$, 对于序列 $\langle u_{N_i+1}, u_{N_i+2}, \dots, u_N \rangle$ 按照 $b_{i,2}^k - b_{i,1}^k$ 的降序排序, 得到序列 $Q = \langle u_1', u_2', \dots, u_{N_2}' \rangle$. 令 $d = \sum_i u_i^k - C$, 其中 u_i^k 为由 u_i' 求得的带宽.

第 2 步: 若 $d \leq 0$, 则转 3. 否则, 在 Q 中找到第一个满足下列条件的边际效用函数 u_i' : (1) $d_i = b_{i,2}^k - b_{i,1}^k \leq d$; (2) $d_{i-1} > d (2 \leq i \leq N_2)$. 如果 u_i' 存在, 将其自 Q 中取出, 同时令 $b_i^k = b_{i,1}^k, d = d - d_i$, 转 2, 否则转 3.

第 3 步: 存储结果, 结束.

第 2 步中之所以选取符合条件的最大 u_i' 主要基于以下两方面考虑, 一是收敛速度快, 二是使带宽和尽量接近最大可用带宽.

(4) 时间复杂度分析

算法 2 第 1 步中相关计算与排序的时间复杂度为 $O(N_2 \times \lg N_2)$, 第 2 步迭代时间复杂度为 $O(N_2)$, 总时间复杂度为 $O(N_2 \times \lg N_2)$.

算法 1 第 1 步时间复杂度为 $O(N)$, 第 2 步每次迭代的时间复杂度为 $O(N + N_2 \times \lg N_2)$, 迭代次数为 $\lceil m / \Delta_m \rceil$ 记为 s , 则总时间复杂度为 $O(s \times (N + N_2 \times \lg N_2))$.

4 实验

为了验证本文算法的求解效果和速度, 通过 Java 语言实现了本文中的算法, 设计了模拟仿真实验, 同最新的同类算法^[10]及著名优化问题求解工具 Lingo9.0 中的相关算法在不同拥塞度下*进行了对比. 实验的软硬件环境为 Windows Vista 操作系统、Intel Core 2 Duo T8100 2.10GHz 处理器、2GB 内存.

4.1 实验设置

在本文的仿真实验中, 共选择 4 类不同的网络应用, 分别是弹性应用 VoIP、IPTV 和非弹性应用 TCP、HTTP 应用, 其流的数量比为 1:1:3:5. 参考文献[8]设置上述 4 类应用的最大带宽为 64002、 10^7 、 10^8 和 10^7 , 最小带宽为 64000、0、0 和 0, ϵ 的取值为 10^{-7} 、 1.67×10^{-5} 、 3.2×10^{-4} 和 3.2×10^{-4} , 链路总带宽 C 为 10^9 bps.

4.2 参与对比的算法

MU: 本文提出的求解算法.

MTRA: 文献[10]提出的多网络资源分配 (Mixed Traffic Resource Allocation) 算法.

LINGO: Lingo9.0 根据输入问题自动选择的非线性规划算法.

4.3 实验结果及分析

本小节给出具体的实验结果并分析各方法的求解效果(模型的总效用值)和求解速度(模型求解时间).

4.3.1 求解效果及分析

本实验通过对比 MU、MTRA 和 LINGO 等方法在不同网络拥塞度下求得的式(9)模型的总效用值, 验证

* 拥塞度 cd 为最大带宽需求与链路带宽的比值.

MU 算法的求解效果. 通过对图 6 中的结果进行分析, 不难发现随着拥塞度的增加, MU 算法结果中流的平均效用会随之下落, 从 $cd = 1$ 时的 1 下降到 $cd = 10$ 时的 0.75. 在所有拥塞度下, MU 算法求得的最大总效用均是最高的(平均比 MTRA 算法高 5%, 比 LINGO 方法高 4%). 另外图 6 中还给出了各拥塞度下流的总数, 由于文中效用函数的值最大为 1, 因此流的总数就是总效用值的理论最大值. 图 7 为各方法总效用值与流总数的比值. 可见, 随着拥塞度的提高, 实验中各方法的总效用值相对流总数的比值不断下降, 符合拥塞度提高效用值降低的预期. 但下降的幅度并不是很大, 如在拥塞度为 10 时, MU 算法的比值为 0.75, MTRA 算法的比值为 0.73, LINGO 方法为 0.68. 主要原因是非弹性应用中 VOIP 应用的带宽需求比较低, 较易满足; 而弹性应用中的 TCP 应用和 HTTP 应用早期效用随带宽的增长快, 后期随带宽的增长缓慢, 因此最初分配带宽会带来较高的效用值, 从而使总效用较大.

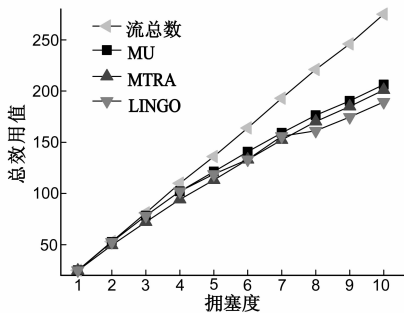


图6 求解效果

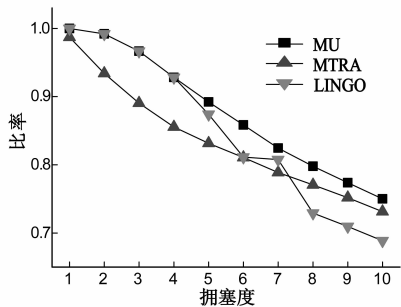


图7 总效用与流总数比值

图 8 为不同协议流的效用对比实验结果, 从中可见, MU 算法和 MTRA 算法表现出较高的一致性, 当拥塞度超过 4 后二者与 LINGO 方法在 IPTV 和 TCP 应用带宽分配差异较大, 从而导致效用值具有较大的不同. MU 和 MTRA 算法将更多的资源分配给 IPTV 流而 LINGO 中的方法将资源分配给 TCP 流. 参照图 2 和图 3, 考虑 TCP 流属于弹性应用, 效用增长速度随带宽的增加不断降低, 最初带宽的增加会获得较高的效益; 而 IPTV 属于非弹性应用, 该类流的效用函数曲线具有明显的拐点, 最

初效用值随带宽的增加增长缓慢, 但在接近拐点处增长十分迅速. 当拥塞度较大时, 由于实验中的流较多, LINGO 方法将较多的带宽分配给 TCP 流, 没有预见到当带宽分配达到一定值后 IPTV 流的效用增速会更快, 因此只获得了局部最优解. 而 MU 算法从整体范围去获取最优解, 其结果更接近全局最优解. 而 MTRA 采用迭代收敛算法较 MU 更易收敛到局部最优解.

4.3.2 求解时间及分析

求解速度对于资源调度算法而言是至关重要的, 它决定算法的实用性. 图 9 在求解时间上对各方法作了对比. 可以发现 MU 算法的求解速度远远优于其它两个方法, 平均耗时只有 0.4s, 而 MTRA 算法为 231.1s, LINGO 方法更高达 12662.1s. MTRA 算法和 LINGO 方法耗时平均是 MU 算法的 578 倍和 31655 倍. 另外, 从图 9 中各个方法的曲线看, LINGO 方法的曲线波动幅度远远高于 MU 和 MTRA, MTRA 算法求解用时随流总数递增而增加的规律与 MU 算法类似, 均较为平缓.

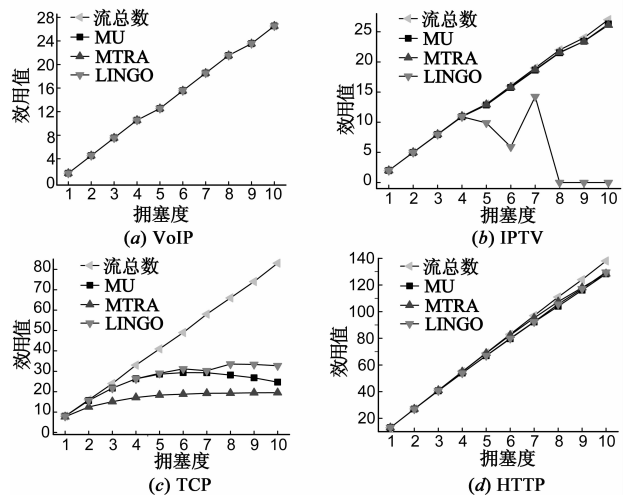


图8 不同协议下效用值

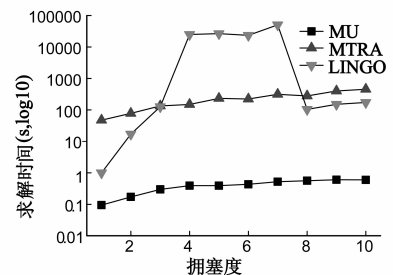


图9 求解时间

4.3.3 实验结论

(1) MU 算法对于式 (9) 给出的问题具有较好的求解效果;

(2) MU 算法具有很高的求解速度, 对于实际的资源调度具有较高的实用性;

(3) MU 算法具有较低的时间复杂度, 求解用时随流总数递增而增加的速度较慢。

5 结论及未来的工作

本文利用边际效用函数来构建网络应用的效用函数, 并通过它们将网络应用分为弹性应用和非弹性应用, 最终将其应用于网络资源调度的建模和求解过程中。仿真实验表明, 本文的方法具有良好求解效果和速度。今后, 将进一步研究把本文的方法应用到新的网络(如内容标记网络 NDN: Named Data Networking)中。

参考文献

- [1] Shenker S. Fundamental design issues for the future internet [J]. *IEEE Journal on Selected Areas in Communications*, 1995, 13(9): 176 – 1188.
- [2] Buyya R, Ranjan R, Calheiros R N. Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services[A]. *Proceedings of International Conference on Algorithms and Architectures for Parallel Processing [C]*. Busan, KR: Springer, 2010. 21 – 23.
- [3] 李世勇, 秦雅娟, 张宏科. 基于网络效用最大化的一体化网络服务层映射模型[J]. *电子学报*, 2010, 38(4): 282 – 289.
Li Shiyong, Qin Yajuan, Zhang Hongke. Mapping model for the service layer of universal network based on network utility maximization[J]. *Acta Electronica Sinica*, 2010, 38(4): 282 – 289. (in Chinese)
- [4] 陈明欣, 刘干, 朱光喜. WLAN 中基于效用的呼叫接纳控制策略[J]. *电子学报*, 2008, 36(7): 1429 – 1434.
Chen Mingxin, Liu Gan, Zhu Guangxi. Utility based call admission control in WLAN[J]. *Acta Electronica Sinica*, 2008, 36(7): 1429 – 1434. (in Chinese)
- [5] 牛志升, 王兰, 段翔. 多媒体 DS-CDMA 系统中基于效用函数的无线资源优化策略[J]. *电子学报*, 2004, 32(10): 1594 – 1599.
Niu Zhisheng, Wang Lan, Duan Xiang. Utility-based radio resource optimization for multimedia DS-CDMA systems[J]. *Acta Electronica Sinica*, 2004, 32(10): 1594 – 1599. (in Chinese)
- [6] Cao Z, Zegura E. Utility max-min: An application-oriented bandwidth allocation scheme[A]. *Proceedings of IEEE INFOCOM [C]*. New York, USA: IEEE Press, 1999. 793 – 801.
- [7] Harks T. Utility proportional fair bandwidth allocation: An optimization[A]. *Proceedings of International Workshop on QoS in Multiservice IP Networks [C]*. Catania, IT: Springer, 2005. 61 – 74.
- [8] Liu Changbin, Shi Lei, Liu Bin. Utility-based bandwidth allocation for triple-play services[A]. *Proceedings of the European Conference on Universal Multiservice Networks [C]*. Toulouse, FR: IEEE Press, 2007. 327 – 336.
- [9] Satya R, Liu Changbin, Liu Bin. Max-min utility fairness in link aggregated system[A]. *Proceedings of the International Workshop on High Performance Switching and Routing [C]*. New York, USA: IEEE Press, 2007. 954 – 962.
- [10] Chen Li, Wang Bin, Chen Xiaohang, Zhang Xin, Yang Dacheng. Utility-based resource allocation for mixed traffic in wireless networks[A]. *Proceedings of IEEE INFOCOM 2011 International Workshop on Future Media Networks and IP-based TV [C]*. Shanghai, China: IEEE Press, 2011. 91 – 96.
- [11] William J Baumol, Alan S Blinder. *Macroeconomics: Principles and Policy [M]*. Cincinnati, OH, USA: South-Western College Pub, 2011. 105 – 125.

作者简介



宋亚楠 女, 1977 年生于河南虞城, 清华大学计算机科学与技术系博士生。研究方向为网络资源调度和服务质量。

E-mail: songyn77@gmail.com;

song-yn07@mails.tsinghua.edu.cn



仲 茜 男, 1975 年生于山东济南, 2010 年在清华大学计算机科学与技术系取得工学博士学位。研究方向为语义数据集成、数据搜索和语义 Web。