

一种基于 Agent 的网络资源调整策略

黄敏, 张小真

(西南师范大学 计算机与信息科学学院, 重庆 400715)

(hmin@swnu.edu.cn)

摘要:当前服务质量是多媒体信息传输的关键,但是传统的 Internet 缺乏动态的服务质量控制机制。针对网络过载和闲置情况,以带宽为例提出了一种基于 Agent 的网络资源调整策略,通过用户 Agent 之间的协商来实现网络资源优化分配。并将这种调整策略进行了实验,实验结果表明了该方法的有效性。

关键词:服务质量;网络资源分配;Agent 协商

中图分类号: TP393.07 **文献标识码:** A

Agent-based strategy for network resource adjustment

HUANG Min, ZHANG Xiao-zhen

(School of Computer & Information Science, Southwest China Normal University, Chongqing 400715, China)

Abstract: The Quality of Service (QoS) is one of the key problems of multimedia data transportation. However, the traditional Internet lacks dynamic QoS control mechanism. The overloading and underloading problem of the network were handled with proposing an Agent-based strategy to adjust the network resources, which was represented by bandwidth. Experiments proved the negotiation among user Agents could realize the optimized allocation of network resources.

Key words: quality of Service; network resources allocation; Agent-based negotiation

0 引言

Internet 的高速发展使得越来越多的应用相继产生,而应用领域的不扩展和深入又对 Internet 的发展提出了更高的要求。例如:在我国,远程教育第二代互联网的典型应用之一。但是随着远程教育的发展,网络及其技术开始暴露出越来越多的不足。远程教育中巨大的视频音频及其控制信息的传输就为网络带来了难以承受的负荷;网络提供的尽力而为型服务已经不能满足应用要求,网络服务质量 (Network Quality of Service) 已经成为大家关注的问题。而一个有效的 Internet 的资源分配方案又是实现服务质量保证的关键技术,所以,本文针对网络资源过载和闲置提出了一种基于 Agent 协商的网络资源调整分配方法。

1 QoS 定义及资源预留

QoS 是指网络中发送和接收信息的用户之间以及用户和传输信息的集成服务网络之间关于信息传输的质量约定,它在某种意义上指的就是用户对网络传输服务的满意程度,具体可以量化为建立连接延迟、吞吐量、传输延迟、固有出错率等参数。而多媒体实时应用的主要参数是端到端延迟、延迟抖动、丢包率、带宽等。具体参数含义见文献[1,2]。IETF 提出了几种服务模型和机制,主要有:集成服务和资源预留协议、区分服务、多协议标记交换、子网带宽管理和流量工程等。

传统的集成服务(资源预留)模型如图 1。其中,RSVP 是由用户端发起的,包括 TSpec、RSPEC 和 Filter Spec,这里的主要参数就是带宽。但是存在的问题是:网络是个动态变化

的环境,但是网络 QoS 管理控制却难以适应网络的动态性,一旦预留的资源得不到保证时,则只能让用户选择降级或进行重协商,而重协商是一个新的端到端的资源预留过程,且只能在资源得不到保证时使用。

传统的资源预留方法约定:一旦用户的资源请求获得了批准,则不能动态调整预约的服务质量,即使网络资源有所闲置。这意味着资源的浪费。如何在这种情况下把浪费掉的资源用于向用户提供更高质量的服务,这就是本文致力于解决的问题(图 1:资源预留过程)。由于网络资源本身的复杂性,在该文中,我们以带宽作为网络资源代表,对我们的方法进行阐述。

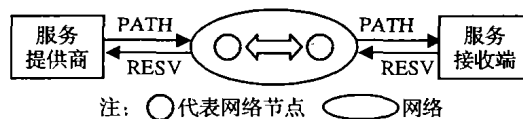


图 1 资源预留过程

2 动态 QoS 管理资源调整策略

首先我们给出本文中有关节点的定义:节点就是网络元素,即任何一个可以在 Internet 中处理数据报文的构件,它具有在数据通过时进行 QoS 控制的能力,包括路由器、子网、端主机系统等^[3]。为了避免上述的节点过载和闲置问题,本文提出一种基于 Agent 协商的网络资源优化分配方法,其核心是由各相关用户 Agent 进行协商,将闲置的节点资源进行重新调整分配,以实现资源的优化。而传统的方法只是在用户预留资源得不到保证时激发重协商。

2.1 改进的资源预留方法

传统的资源预留只提供了一种服务参数集,包括:带宽、

延迟、延迟抖动等参数^[4]。系统在预留资源时往往根据最大资源要求,使得网络利用率很低。本文中,将传统的资源预留作了改进:用户需要在提出 QoS 服务质量的时候提出两种参数:一个是服务质量的最低限;一个是服务质量的期望限。网络提供的服务质量应以用户的期望限为准,我们允许在期望限附近小范围波动,该范围的限定以用户能接受的抖动参数为准,毕竟,要网络完全以一种不变的速度来为用户提供服务是很困难的。只要网络提供的服务质量在用户的最低限以上,我们都认为网络履行了自己的给用户的质量保证承诺。

2.2 资源分配策略

由于 QoS 提供的是一种端到端的服务保证,所以在进行资源分配时应在每一个网络节点处都进行协商,只有每一个节点协商成功才认为整个协商是成功的。在每一个节点处的协商策略是以降低某一范围内用户的服务质量来进行的,在协商过程中,被降低了服务质量的我们称为受损用户。本文中的分配策略就采用逐步扩大受损用户范围的方法,在选取受损用户时参照用户的优先级来进行。

资源的优化分配由用户 Agent 在各个网络节点协商实现。

3 Agent 构建

每一个网络用户有一个 QoS agency(如图 2 所示)。用户 QoS agency 由两类 Agent 组成:一类是静态 Agent,一类是移动 Agent。它们共享用户的 QoS 文件库,并由 Agent 工厂动态生成。当用户第一次注册登录系统时,系统在客户端实例化生成一个静态 Agent,它的职责见 3.1 部分。客户每次登录系统,系统都实例化生成一个移动 Agent 并发送至客户端^[4]。

3.1 静态 Agent

考虑到用户通常都只能通过应用层对 QoS 进行控制操作,网络的瞬息变化使得用户无力去应付。因此我们给该类 Agent 赋予的职责有四个:

- 1) 为用户提供直观的 QoS 选择。对于大多数网络用户来说,众多的 QoS 性能参数不管是理解上还是从记忆上来说都是烦琐的,在这里,我们屏蔽掉传统的参数填写方法,直接由 Agent 为用户提供不同等级的视音频资料,用户可以根据该资料很直观的选择出自己所需服务质量的最低限和期望限;
- 2) 对 QoS 参数进行映射;
- 3) 在本地进行资源预留;
- 4) 在服务过程中执行 QoS 监控。当 Agent 发现有不符合 QoS 的流到来时,该 Agent 将信息流进行整形到 QoS 可接受的级别^[3]。

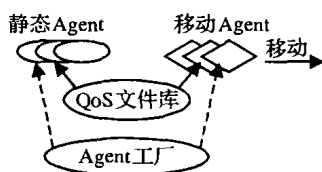


图2 QoS Agency 结构图

3.2 移动 Agent

在这里,移动 Agent 的任务是登录其他网络节点,与其他共享网络节点资源的 Agent 协商优化分配网络资源,为用户提供更高质量的服务^[3]。

用户对服务质量要求的最低限和期望限由静态 Agent 获得以后,将期望限映射为相应的性能参数,并提交给网络,再

由网络各层进行参数的映射。而用户的最低限由静态 Agent 保存,在以后的网络传输服务过程中,该最低限将作为衡量是否接受该次协商结果的标准。每个网络节点有一个资源管理器,该管理器的功能是管理节点的资源,并周期性地向用户 Agent 汇报节点资源的使用情况,由静态 Agent 根据得到的信息决定是否激发资源优化协商。

移动 Agent 的体系结构图如图 3 所示。推理机决定是否产生及如何产生反应。知识库包含当前世界状态的知识和 Agent 自身内部状态数据,通过和推理机进行交互,知识库的内容不断进行更新。控制器包括了规划、调度、执行的功能,它使得 Agent 具有自主性和智能性。

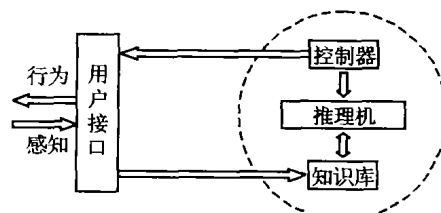


图3 移动 Agent 的体系结构

4 动态 QoS 管理的协商过程

4.1 资源优化分配的激发

本文中的静态 Agent 并不参与资源的协商,协商是由动态 Agent 来完成的。资源调整并不意味着一定发生 Agent 的协商,协商有一定的激发条件。当节点 a_i 的资源管理器发现节点资源出现闲置,进行的只是资源优化分配。资源优化分配按以下步骤开始具体实施:

Step1: a_i 将闲置信息通知所有通过该节点的用户 Agent;

Step2: 用户 Agent 根据当前用户的情况决定是否参与资源的调整,愿意参加资源调整的用户向节点 a_i 发送调整请求;

Step3: 节点 a_i 采用资源占有百分比不变来进行计算,所有参与资源调整的 Agent 都为自己的用户分到了一定的资源;

Step4: 分到资源的 Agent 进行端到端的资源优化分配协商判断。只有当网络节点资源不足以满足用户时才激发节点协商,所以移动 Agent 需要在每一个节点处进行是否进行协商的判断。协商发生与否由 $Agent_i$ 在节点处进行判断,判断过程见算法 1。

算法 1: 协商判断

R : 原有带宽, R' : a_i 运算后的带宽, x : 节点 a 的剩余带宽;

if $x > R' - R$

then 下一节点判断

else 进行端到端协商(算法 2)

4.2 协商引发的系统抖动问题的处理

$Agent_i$ 在每个节点处,首先访问节点的资源管理器,如果此时管理器标签为 1,则表明此管理器正处于另一个协商阶段,一个资源管理器不能同时进行多处协商。这样做的目的是为了避免频繁资源协商而引起的系统抖动。在管理器标签为 0 时, $Agent_i$ 获得该节点的资源使用情况,再由该节点的资源闲置情况决定 $Agent_i$ 是否与其他共享该节点的 Agent 进行协商。 $Agent_i$ 经过的每一节点被首次访问后都将标签置为 1,直到所有的协商结束,标签置 0。这样依次要遍历该路径中的所有其他节点。

4.3 节点协商

协商算法 2:

Flag: 资源管理器标签

$d_{i,j}$: 用户 Agent 优先级

Suser = {user₁, ..., user_n | user_i ∈ Auser}: 受损用户集合

Auser: 所有用户集合

① if Flag = 1

then wait

else go②

② Flag = 1;

③ Collect(m);

/* 收集共享该节点 a_m 的所有 Agent 的优先级等级 $d_{i,j}$ */

④ Sort($d_{i,j}$); /* 对 $d_{i,j}$ 进行排序 */

⑤ ArisenProject(d);

/* d 为 Sort 函数返回的排序结果, 本函数根据排序结果产生受损用户及受损方案 */

⑥ Send(p);

/* p 为 ArisenProject 返回结果, 本函数将受损方案发送给各受损 Agent */

⑦ if Responds = yes

/* Responds = Respond₁ and Respond₂ and and Respond_i */

then /受损 $d_{i,j} = d_{i,j-1}$

其他 $d_{i,j} = d_{i,m}$

go ⑩ }

else go ⑧

⑧ Enlarge(d) /* 扩大受损范围 */

⑨ if SuserAuser

then go ⑦

else go ⑩

⑩ 结束

协商时, 我们首先给用户 i , 即代表用户的移动 Agent_i 初始化一个优先级数, 我们用一个二维变量 $d_{i,j}$ 来表示, 其中 i 表示优先级等级, j 表示在该等级内的优先级。该优先级等级值 i 根据用户对网络服务的付费多少有所不同。

当 Agent_i 在某一节点发起协商后, 它首先询问其他 Agent 的优先级数, 得到答案以后, Agent_i 对答案进行排序, $d_{1,1} < d_{1,2} < \dots < d_{1,m} < d_{2,1} < \dots < d_{2,m} < \dots < d_{n,m}$, 这里充分考虑了网络付费的情况, 在保证付费最多的用户受损最小的前提下, Agent_i 根据排序结果向其他用户提取部分资源, 其他用户的静态 Agent 在保证用户最低限的前提下决定是否同意该方案, 如果让用户的所享受的服务质量降低到了最低限以下, 则向 Agent_i 传递否定的信息, Agent_i 根据该方案的反馈情况来扩大受损用户的范围, 同时也降低了每个用户受损的程度。当解决方案确定下来以后, 每个相关的 Agent 的优先级数都要发生变化, 没有为 Agent_i 贡献资源的 Agent 的 $d_{i,j} = d_{i,j} - 1$, 为 Agent_i 贡献了资源的 Agent 的 $d_{i,j} = d_{i,m}$ 。如果所有的相关 Agent 受损都无法满足 Agent_i 的资源要求, 则 Agent_i 放弃整个网络节点的协商。

受损用户再分别进行端到端资源的调整, 此时的调整只是一个简单降低资源预留的过程, 不需要任何协商, 且不再影响其他用户。

5 设计分析及部分实验

本文提出的基于 Agent 协商的动态 QoS 调整方法满足 QoS 控制设计的五大原则: 集成原则、分离原则、透明原则、异步资源管理原则和性能原则, 主要表现为 Agent 的自主性和

异步执行, Agent 在应用、系统、网络三层上对 QoS 的处理以及整个方法以提高整个网络资源利用率为目的, 并不影响其他的网络协议。

移动 Agent 的优点是明显的, 它减轻了网络的负载, 减少客户端资源的使用, 进行异步处理, 提供可配置服务, 但它的响应时间受到移动和服务时间的限制^[5]。

由移动 Agent 的协商过程, 响应时间 $t = \sum_{i=1}^n t_i + \sum_{i=1}^N t_{ser,i}$

其中,

N : 一条链路上的网络节点数目 (包括用户节点和服务提供者节点);

n : 移动次数, $n = 2 * (N - 1)$;

t_i : 每次移动的传输时间;

$t_{ser,i}$: Agent 在某节点的服务时间 (等待时间 + 协商时间)。

等待时间与移动 Agent 本身的性能结构是无关的, 它与网络节点的负载有关。而协商时间则由算法来决定, 协商的次数 $\leq N - 1$ 。

通过分析, 整个网络资源的调整分配成功与否与网络状况密切相关。传统的重协商策略发生在网络状况过载的情况下, 本文提出的资源调整方法可以作用于负载较轻和过载的情况下。且都能起到资源优化分配的作用。

对 Agent 协商, 我们采用单机模拟实现, Agent 的移动时间受具体地理位置距离的限制 (这里不考虑 Agent 丢失等因素)。Agent 由 Java 语言编写模拟, 模拟的用户个数为 5 个 (5 用户经过同一路由器)。很显然, 此时该路由器没有资源闲置, 导致用户 2 预留资源下降至最低限以下, 为 24 Mbps, 则此时要进行协商。下面给出一种协商前后的带宽情况如表 1 和表 2 所示。

表 1

用户	优先级	资源预留期望限	资源预留最低线
用户 1	$d_{3,2}$	27.5 Mbps	25 Mbps
用户 2	$d_{2,2}$	27 Mbps	25 Mbps
用户 3	$d_{1,3}$	70 kbps	64 kbps
用户 4	$d_{1,2}$	1.76 Mbps	1.6 Mbps
用户 5	$d_{2,1}$	22 kbps	20 kbps

表 2

用户	优先级	资源预留期望限	资源预留最低线
用户 1	$d_{3,2}$	26.562 Mbps	25 Mbps
用户 2	$d_{2,2}$	25 Mbps	25 Mbps
用户 3	$d_{1,3}$	67 kbps	64 kbps
用户 4	$d_{1,2}$	1.622 Mbps	1.6 Mbps
用户 5	$d_{2,1}$	21 kbps	20 kbps

协商过程的分步结果如表 3 所示。

表 3 协商过程的分布结果

协商循环次数	受损用户	贡献带宽 d 的最大量	协商成功否
1	4	0.16 Mbps	否
2	3, 4	0.166 Mbps	否
3	3, 4, 5	0.168 Mbps	否
4	1, 3, 4, 5	2.668 Mbps	成功

可以看出, 到第四次循环结束时, 已经为用户 2 协商到了 1 Mbps 带宽, 则用户可以继续接受网络服务。在这种网络过

载的情况下,用户2需要中断服务等待协商结果(同重协商策略),而在网络资源闲置的情况下则不需要中断服务。

从对算法的分析中,我们可以看到,当网络中用户巨大时,这种算法可以很容易确定受损用户,随着用户数目增加,能降低每个受损用户的受损程度,使受损用户几乎感觉不到服务质量的降低。

6 结语

本文在现有的 QoS 参数定义模式下提出了一种新的资源调整算法,该算法不仅可以在网络不能保证资源要求时对资源进行动态调整,也可以在网络资源闲置的情况下对资源进行优化分配,提高了网络的资源利用率。在调整过程中提出的受损是指以带宽为代表的所有参数整体降低,这是以现有的 QoS 参数定义模式为基础的。考虑到特定传输媒体的不同参数对媒体的影响效果不同,所以将在以后的研究中,将 QoS 参数定义模式进行扩展,并改进本文中的调整算法。

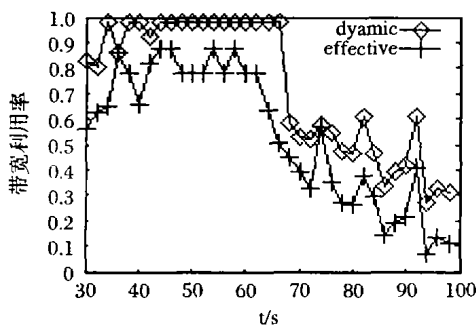
(上接第181页)

如果缓冲的利用率超过了上门限值则减少补偿量。这主要是为了实现对数据流的整形,以减少丢包。

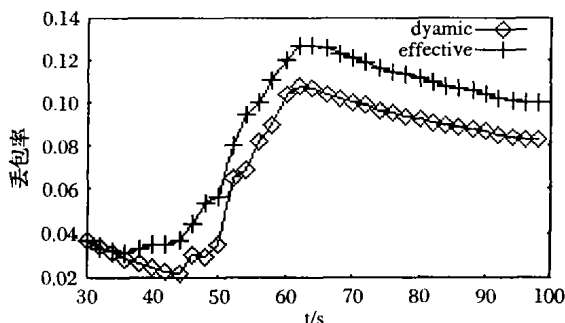
如上所述,算法每隔固定时刻 T ,通过预测来计算应该给每个业务流分配的带宽资源(输入队列的服务速率)来动态的分配资源以提高链路的利用率和减少丢包。

2 仿真试验的结果及其分析

作者在 ns-2 的仿真环境下(版本号 ns-2.1b9a)对本算法实现了仿真,并且和基于有效带宽理论的资源分配方案进行了对比。ns 系列仿真器由美国劳伦斯伯克利国家实验室网络研究组(Network Research Group at the Lawrence Berkeley National Laboratory, LBNL)开发,已经成为网络研究学术界进行网络和协议仿真研究的社区标准。



(a) 两种算法的带宽利用率对比图



(b) 两种算法的丢包率对比图

图3 仿真试验结果对比图

图3所示是试验仿真后得到的对比结果。从图3可以得

参考文献:

- [1] 顾冠群,等.网络服务质量(QoS)参数研究[J].计算机研究与发展,1998,35(6):543-547.
- [2] 白成林. QoS 研究的理论框架及研究现状[J]. 计算机工程与应用,2002,38(24):154-164.
- [3] 赵彤宇,等. 基于移动 Agent 技术的 QoS 保障机制[J]. 通讯技术,2002,128(8):45-47.
- [4] MAILLER R, LESSER V, HORLING B. Cooperative Negotiation for Soft Real-time Distributed Resource Allocation[A]. Proceedings of Second International Joint Conference on Autonomous Agents and MultiAgent Systems[C], 2003. 576-583.
- [5] GUEDES LA, OLIVEIRA PC. QoS Agency: An Agent-based Architecture for Supporting Quality of Service in Distributed Multimedia Systems[A]. IEEE Conference on Protocols for Multimedia Systems - Multimedia Networking[C], 1997. 24-26.
- [6] MANVI SS, VENKATARAM P. QoS Management By Mobile Agents in Multimedia Communication[A]. 11th International Workshop on Database and Expert Systems Applications(DEXA'00)[C], 2000.

出以下结论:

1) 因为本算法根据每个业务流的流量状况来动态的分配当前可以利用的资源,这样就能容忍单个流的短暂突发,并且最大限度的利用了输出缓冲,实现了对突发业务流的整形,从而降低了整个系统的丢包率。

2) 另外本算法不再为每个业务流预先分配可用资源,而是不停的根据每个业务流的状况在各个业务流之间动态的分配资源,这样就不存在只能为某个业务流使用而不能被其他业务流使用的带宽资源,必然能够提高带宽的利用率。

3 结语

本文首先分析了传统资源分配算法的理论基础:有效资源带宽理论。从理论上解释了这个理论的不足之处。在此基础上提出了改进算法,通过给每个业务流动态分配资源,较好的解决了传统方案的不足之处。最后,经过仿真对比试验说明,本算法无论在丢包率还是在带宽利用率上均取得了预期的目的,改进了传统算法。

参考文献:

- [1] CHANG CS. Stability, queue length and delay of deterministic and stochastic queueing networks[J]. IEEE Trans. Automat. Contr., 1994, 39(5):913-931.
- [2] CHANG CS, THOMAS JA. Effective bandwidth in high-speed digital networks[J]. IEEE J. Select. Areas Commun., 1995, 13(6):913-931.
- [3] GLYNN PW, WHITT W. Logarithmic asymptotics for steady-state tail probabilities in a single-server queue[J]. J. Appl. Prob., 1994, 31A:131-156.
- [4] BLAKE S, BLACK D, CARLSON M, DAVIES E, et al. RFC 2475 An Architecture for Differentiated Services[A]. IETF[J], December 1998.
- [5] NICHOLS K, BLAKE S, BAKER F, et al. RFC2474 Definition of the Differentiated Services Field (DS Field) in the Ipv4 and Ipv6 Headers[A]. IETF[C], December 1998.
- [6] ROCHE C, FDIDA S. A Dynamic Resource Management Mechanism for LAN Interconnection across High-Speed Networks[A]. INFOCOM[C], 1994. 884-891.
- [7] GALLARDO JR, MAKRAKIS D, ANGULO M. Dynamic Resource Management Considering the Real Behavior of Aggregate Traffic[J]. IEEE Transactions on Multimedia, 2001, 3(2):177-185.