具有QoS特征宽带IP网络队列调度算法的研究*)

刘宴兵 李乘智
(重庆邮电学院计算机系 重庆400065)

The Study of Queue Scheduling Algorithm with QoS Character in Broadband IP Network

LIU Yan-Bing LI Bing-Zhi
(Department of Computer Science, Chongqing University of Post and Telecommunications, Chongqing 400065, China)

Abstract IP Quality of Service (QoS) guarantee is realized by managing and avoiding congestion in network. In this paper, the primary queue algorithms are compared in managing congestion. On the basis of these theories, a new LLQ+CBWFQ algorithm is achieved, and the minimum bandwidth is assigned according to priority or request.

Keywords IP network, Queue scheduling, Algorithm

1 引言

随着信息技术的发展,人们对信息的需求已不满足于传统的电报电话业务,甚至传统的文件传输、电子邮件等数据业务,而是追求更高品质的集视频、图像、声音、文字、甚至动画等为一体的多媒体宽带应用服务。这些应用不仅要有带宽保证,而且还需要有时延、时延抖动、分组丢失率的限制。而目前的Internet带宽管理不够合理,不同设备使得路由瓶颈仍然存在,接入速率低,延迟大而不确定,这些都使得实时性强的音视频流质量不能得到保证,限制了网络多媒体技术的发展。但是,IP网要成为未来信息网络的基础网,就必须能支持上述多媒体传输。因此,IP网的设计一方面要求网络有很宽的带宽,另一方面对时延及时延抖动也有严格的要求。

2 队列调度算法的比较

在网络中常见有4种队列调度机制:先进先出队列(FIFO)、优先权队列(PQ)、自定义队列(CQ)、公平队列(FQ)和加权公平队列(WFQ)。其中,FIFO队列是最基本的队列机制,实现简单。但它不考虑数据包的缓急,一律按照时间先后发送数据包,使得一些时间敏感的数据包不能正常工作。并且FIFO队列的尾丢弃,容易产生全局同步现象,不利于带宽资源的有效利用。PQ队列将数据包分为四个优先级,分别为高、中、普通和低优先级,每个优先级队列为一个FIFO队列。PQ队列严格按照优先级来服务各队列,只有在较高优先级队列均为空时,才可以服务低优先级的队列。这会使得低优先级队列长时间得不到服务,甚至根本得不到服务,产生队列饥荒。CQ队列循环地服务各个队列,克服了PQ队列的队列饥荒现象。同时,CQ队列允许网络管理员设置每个队列的字节计数,以达到按优先级或按需求分配带宽的目的。但CQ仍然会造成敏感数据包的长时延,类似队列饥荒,只是程度不同。公平队列和加权公平队列给所有网络流量提供平等的带宽分配。WFQ将已分类的流放入动态创建的队列中,以循环方式服务队列,高优先权被指定给低带宽队列。

网络中重要的通信经常被低优先级的通信所延迟,因为在使用FIFO排队的路由器队列上,来自所有活动流的分组都被同等对待。为有助于根据服务要求来区分分组,站点的网络管理员在入口根据通信的重要程度标记分组的IP优先级,并实施了基于流的WFQ,这样:

·IP优先级相同的活动流获得的接口带宽相同

·优先级高的活动流获得的接口带宽比低优先级的多

需要指出的是:"基于流"是指通信中鉴别流的一个方法。它表示WFQ怎样分配流量给不同的队列。一般情况下,WFQ根据协议类型、源地址、目的地址、源端口、目的端口、ToS字段等参数的结合来对流量进行分类。举例来说,从主机A到主机B的Telnet流量将被认为是一个流,从主机A到主机C的Telnet流量被认为是另一个流。

基于流的加权公平队列WFQ是一个简单、动态的排队机制,它确保在网络里的每个会话公平享用带宽。与需要手工配置的静态排队机制PQ和CQ不同,WFQ动态的适应网络里的变化,包括新的协议和应用。如果不需要设置为比其他的高优先权的重要任务的流量,WFQ是用来为每个网络用户提供最好级别服务的一种简单和高效的方法。

在基于流的WFQ实现中,权重严格地基于优先级,不能改变。加权公平队列根据ToS字段或IP优先权的不同而给流不同的权重。当存在不同的ToS值时,流的权被计算,每个数据的优先级值加1。每个流的权值除以所有流的总权值就是每个流所占总带宽的比例。例如,如果3个流都使用默认IP优先权0,那么它们被给定权1(0+1)。总带宽的权是3(1+1+1),每个流得到1/3的总带宽。另一方面,如果两个流的IP优先权是0,第三个的为5,则总权是8,开始的2个流分别得到1/8的总带宽,第三个得到6/8的总带宽。

由于流是根据协议类型,源地址,目的地址,源端口,目的端口,ToS字段来分类的,所以同一个流中优先级值不同的分组将被分到同一个队列中,同一个流队列中的分组将按照FIFO的顺序获得服务。

3 改进的LLQ+CBWFQ算法

CBWFQ算法为每个通信类分配不同的队列,而不像基

*)基金项目:重庆市科委应用基础项目资助。刘宴兵 硕士,主要从事宽带网络性能分析及设计研究。李乘智 教授。

于流的 WFQ 那样为每个流分配一个子队列，因此可以使用已有的基于流的 WFQ 实现来提供分布式和非分布式两种运行模式的 CBWFQ，方法是：添加一个通信分类模块，让每个 WFQ 队列传送一个通信类，而不是一个流。CBWFQ 机制使用模块化 QoS CLI(命令行界面)框架，因此它支持该框架的所有类。可以根据各种通信参数(如 IP 优先级、DSCP(区分服务码点)、输入接口以及 QoS 组)对通信进行分类。CBWFQ 使得用户可以直接指定每个通信类所需的最小带宽，该功能与基于流的 WFQ 相同，在基于流的 WFQ 中，流的最小带宽间接地决定于 WFQ 系统为所有活动流指定的权重。CB-WFQ 用来运行基于类的 WFQ。在 CBWFQ 中，默认的通信和常规 WFQ 流相同。

在 CBWFQ 中，为通信类指定的带宽为在拥塞时保证的最小带宽，如果某类通信没有用完它指定的带宽，那么排队系统中的其他通信类就可以根据给它们指定的带宽，按比例使用这些剩余带宽，即它的主要优点有：权重保证最小带宽；缓冲控制时延；剩余带宽可被其它类分享；每个队列根据 QOS 要求可分别配置。

但 CBWFQ 不能有效地处理实时业务。这里共622个队列，我们把类编号为0的队列定为系统队列，它是一个单一的队列。系统把优先级别最高的数据包，如保持活动数据包和信令数据包，安排到这个队列，它总是得到绝对优先的服务，其它通信不能使用这个队列。第1类队列为实时业务队列，例如用于 IP 电话业务，也是一个单一的队列。由于要求实时业务的排队等待时延小于10微秒，因此在分配给这类业务的带宽范围内，这个队列总是得到绝对优先的服务。但 CBWFQ 算法不提供时延保证，所以在我们的调度算法中结合了 LLQ (Low-Latency Queuing)思想，即在 CBWFQ 队列前面加上 PQ(Priority Queue)，为语音和实时业务等提供时延保证，它具有柔软性和容易配置等优点，其示意图如图1。

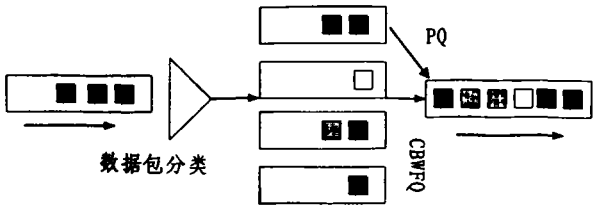

图1 LLQ+CBWFQ 调度示意图

我们称以上调度算法叫 LLQ+CBWFQ 算法。下面我们给出其实现过程。

4 LLQ+CBWFQ 算法实现

对前两个类的队列，我们采用 LLQ，即在满足其带宽的同时，也提供时延保证，具体做法是，每次扫描端口调度时，只要有数据包就调度它们，类0队列为空后，调度类1队列数据包。只有类0、1队列均为空时才调度后面类队列，即 CBWFQ 算法开始工作。

由于示意图中前端对业务进行了分类，CBWFQ 算法思想与 WFQ 算法雷同，因此在这里 CBWFQ 调度算法实现包括了三个部分：计算类的最小虚拟结束时间，带宽分配，调度。

4.1 计算各类每个数据包的最小虚拟结束时间

入队时计算类的最小虚拟结束时间和前面 WFQ(在 WFQ 中针对的是流，而 CBWFQ 算法针对的是类)算法一致，即指当有一个数据包到达的时候，判断该数据包是否属于当前已有的类，或者是一个新的类。如果该数据包属于一个已有的类，则将该数据包入对。如果该数据包是一个新的类，则为该数据包创建一个新的队列。

当一个数据包 k 进入该数据包所属的类 i 对应的队列的时候，要计算该数据包的虚拟的到达时间 $S(k,i)$和结束时间 $F(k,i)$。计算的时候要考虑到各个类的公平性，尤其要照顾到一个新创建的类的公平。

当一个数据包 k 属于一个已有的类 i 的时候，计算如下：

$$F(k,i)=S(k,i)+L(k,i)*W(k,i)$$

$$S(k,i)=F(k-1,i)$$

当一个数据包 k 属于一个新的类 i 的时候，计算如下：

$$F(k,i)=C(t)+L(k,i)*W(k,i)$$

以上两个式中，$F(0,i)=0$，$L(k,i)$是第 i 个类中第 k 个包的长度，$W(k,i)$是第 i 个类中第 k 个包的加权，C(t)是当前的服务时间。

同时对缺省类(63类)中的每个流的数据也算出数据包的虚拟的到达时间 $S(k,i)$和结束时间 $F(k,i)$。

4.2 带宽分配

设系统初始配置类有 CN(CN≤63)个，每个类要求的最小带宽为 LBW_i(i=1,…,CN)，缺省类最小带宽 $LBW_{63}=622-\sum_{i=1}^{CN}LBW_i$。

设在调度的某一时刻非活动类(队列为空)CM 个(CM≤CN)，这时就有剩余带宽 $RBW=\sum_{i=1}^{CM}LBW_i$，排队系统中的其他通信类就可以根据给它们指定的最小带宽比例(最小带宽作权重)使用这些剩余带宽。因此可求出每类实际分配到的带宽 BW_i，计算如下：

$$BW_i=LBW_i+RBW*LBW_i/622$$

此处需要注意的是：分配给0、1类的最小带宽在非处于活动(队列为空)时其指定带宽不能拿去剩余分配，要进行带宽资源预留(RSV)。

4.3 调度数据包

调度的时候，扫描端口，只要有数据包就调度它们，类0队列为空后，调度类1队列数据包，即优先调度类0、1队列。只有类0、1队列均为空时才调度后面类队列，即 CBWFQ 算法开始工作，即从当前所有队列中挑选出具有最小虚拟结束时间 C(t)的包按实际带宽 BW 调度出队列到输出链路上，最小虚拟结束时间 C(t)计算如下：

$$C(t)=\min(F(k,i))。$$

4.4 形式描述

初始化：
CN＝配置活动类的个数；
LBW_i＝第 i 类要求的最小带宽；
C(t)＝0；F(0,i)＝0；CM＝0；
入队：
IF (检测到一个新的数据包 k)
　检测该包所属的类
IF (到达数据包属于0、1类)
THEN 不计算最小虚拟结束时间直接入队列；
ELSE{
　IF (该包属于一个已有的类 i)

(下转第175页)

偷窃，因为此时数目相对比较庞大，则不得不考虑安全问题。最简单的考虑就是此时使用公钥加密技术来保证硬币传送的安全性。但是考虑到效率的问题，就尽量避免使用公钥加密技术，并通过使用效率比较高的 Hash 函数来保证安全性。

第一种方法就是组相关，即中间人将消费者分成若干个组。消费者硬币的有效性与他所在的组建立关联。具体地讲就是，中间人分给每一个消费者一个数字 ID 和硬币，商家通过如下的附加条件很容易进行验证：

$h'(x_1, x_2, \cdots, x_k) = h'(ID)$

h′是产生短的散列和的加密散列函数。散列和说明客户从属的组。

还有一种方法就是使消费者的硬币只能用于特定的商家。这对硬币的盗窃者来说就更没有吸引力了。

3）**重复消费** 由于 MicroMint 并不是基于匿名的，所以中间人可以发觉是哪一个商家兑换的重复消费的货币，甚至可以找到是哪个消费者消费的这种货币。因此，中间人可以不兑换这种货币；而且当发生大规模的重复消费时，中间人可以剔除相应的商家或消费者。但是这样做，可能伤害那些诚实的但是被动收到重复消费货币的商家的感情。并且中间人也只是被动地剔除那些有欺骗嫌疑的商家或消费者，而不能主动地对这些欺骗行为进行惩罚。

3.3 小结

MicroMint 没有采用公钥和对称加密技术，整体的安全性不如 PayWord，但由于采用了四向 hash 函数冲突，大规模的欺骗在计算上是不可行的。同 PayWord 不同，MicroMint 货币不是针对某一特定 M 的，所以，可允许 C 高效地和多个 M 交易，这也是 MicroMint 区别于其它微支付的显著特点。另外，与 Millicent 不同，C 可以在本地验证硬币的真伪。

结束语 一般来说，所有的微支付都是建立在效率和安全平衡的基础上的。由于微支付的交易额比较少，所以我们可以寻找效率较高但是又可以保持适当安全的方法进行改进。如利用椭圆曲线来替代现有的 RSA 算法，可以在充分利用公钥技术特性基础上，有效提高系统效率。还有就是对现有的微支付进行局部的改进，以提高微支付的效率和安全性。

随着网络技术及网络应用的发展，微支付的应用将会越来越广泛。例如网络出版或者网络信息提供就可能是微支付应用的一个重要方面。还有就是结合移动通信和移动电子商务中支付的特点，微支付在移动计费中的应用也显得越来越重要，这也是微支付的一个重要发展方向和研究热点。

参考文献

1 Glassman S, et al. The Millicent Protocol for Inexpensive Electronic Commerce. http://www.w3org/Conferences/WWW4/Papers/246/

2 Lang P. Product review MilliCent micropayment system. http://sellitontheweb.com/ezine/millicent.shtml. 1998

3 Glassman S, Jones R, Manasse M. Microcommerce On The Horizon. http://research.compaq.com/SRC/articles/199705/Millicent.html

4 Rivest R, Shamir A. Security Protocols Workshop. http://citeseer.nj.nec.com/rivest-payword.html

5 Puhrerfellner M. An implementation of the Millicent micro-payment protocol and its application in a pay-per-view business model. http://citeseer.nj.nec.com/507471.html. 2000

6 Rivest R, Shamir A. Payword and MicroMint Two simple micropayment schemes. http://theory.lcs.mit.edu/~rivest/Rivest-Shamir-mpay.pdf. 2001

7 Technological Foundation of E-Commerce-chapter5: Digital Payment System. SIMENS AG, CTIC 3. Security/Electronic Commerce

8 李明柱，李志江，杨义先. 微支付机制及应用分析综述. 计算机工程与应用，2002，38(3)

9 林枫等编著. 电子商务安全技术及应用：第6章. 安全电子支付概论. 北京航空航天大学出版社，2001

10 Vesna Hassler 著，钟鸣等译. 电子商务安全基础. 人民邮电出版社，2001

（上接第133页）

```
    THEN { F(k,i)=S(k,i)+L(k,i) * W(k,i);
           S(k,i)=F(k-1,i); }
  IF (该包不属于任何一个已经有的类)
  THEN { F(k,j)=C(t)+L(k,j) * W(k,j)}
  }
队列扫描:SM=非活动类数目;
调度:
  IF (类0队列非空)
    THEN 按实际分配带宽 C0=LBW1调度类0队首数据包
    ELSE IF (类1队列非空)
      THEN 按实际分配带宽 C1=LBW2调度类1队首数据包
    ELSE
      { 扫描后面类队列和缺省类中的不同流子队列的数据包;
        根据 CN,LBW_i,CM 计算实际分配带宽 BW_i;
        选择 C(t)=min(F(k,i))按实际分配带宽 BW_i 调度出队列;}
```

结束语 调度算法是 IP 宽带网络交换机中的重要部分，它为传输实时业务提供 QoS 保证。本文对几种常见的调度算法进行了比较研究，在此基础上进行改进得到 LLQ+CB-WFQ 算法，并给出其实现过程。这些研究成果我们应用于“大唐光通信公司多业务交换平台开发实现项目”中，取得较好的 QoS 性能特征。

参考文献

1 Chen J S, Guerin R. Performance study of an input queueing packet switch witch with two priority classes. IEEE Trans. Commun [J], 1991, 39(1): 117～126

2 Hluchj M G, Karol M J. Queueing in high-performance packet-switching, IEEE J. Sel. Areas Communcation[J], 1998, 6(9): 1587～1597

3 Lee T. A modular architecture for very large packet switches. IEEE Transactions on Communcation [J], 1990, 38(7): 1097～1106

4 戴礼森，洪佩琳. 高速信元交换调度算法研究. 电子学报[J]，2000，28(5)：96～98

5 黄立群. FQLP：ATM 网中一种新的实时业务调度算法. 电子学报[J]，2000，28(4)：20～23