基于EDF的多优先级队列管理方案研究

周荣华，钱光明
（湖南师范大学 数学与计算机科学学院，湖南 长沙 410081）

摘 要：提供服务质量保证是目前Internet的重要研究课题之一，其核心问题是实现不同业务流的分类转发和缓冲队列管理。分析了基于EDF的优先级队列(PQBEDF)，由于PQBEDF方案中动态优先级随时间片变化过快从而降低了高优先级队列服务质量，针对这个不足引入一组概率序列 P_i 来控制计数器一个时间片以后是否加1，从而为每类业务的信元保证一个最小的服务速率。

关键词：优先级队列；堆；信元区；服务质量
中图分类号：TP393 **文献标识码**：A **文章编号**：1673－629X(2008)06－0083－03

Queue Management Scheduling Research of Multi－priority Based on EDF

ZHOU Rong-hua, QIAN Guang-ming
(College of Mathematics and Computer Science, Hunan Normal University, Changsha 410081, China)

Abstract: Providing quality of service(QoS) is one of the most important researches in the Internet at present. The key issue of QoS is to provide different services for different kinds of flows and buffer area management. Analyzes a priority queue based on EDF(Earliest Deadline First): PQBEDF, because change of the dynamical priority is too quik so as to reduce the high－priority queue quality of service. Aiming at the problem of the rapid modification, introduce a probability sequence P_i in order to control a couter which increase by one or not each slot, so that a minimum service rate can be guaranteed for each kind of business.

Key words: priority queue; heap; cell sector; quality of service

0 引 言

随着Internet上业务种类的日益丰富，用户对其要求也各不相同，因此有必要根据不同的QoS需求，按照业务量类型或级别对其中的数据包赋以不同的优先级，以便IP网络对各级别的业务量进行区分处理[1,2]。在文中，利用基于截止期优先算法(EDF)的思想结合DSCP中对应的多优先级来研究基于EDF的优先级队列(Priority Queue Based on EDF)及改进算法，从而为每类业务的信元保证一个最小的服务速率。

1 基于EDF的优先级队列(PQBEDF)分析

1.1 PQBEDF中的动态优先级构造

针对多媒体信息传输要求而提出了一种基于信元的、能够提供服务质量的网络技术，参照ATM使用53个字节的固定大小的信元[3~5]，PQBEDF方案正是采用了这一做法，将一个缓冲区等分成若干信元区(以下简称区)，一个区可以为一个信元大小，也可以为多个。文中以下部分的讨论是假定一个缓冲区恰好被等分成8个区，每个区为一个信元大小，并以8个区组成一个队列来讨论。考虑到在缓冲区中这些信元可能来自优先级不同的数据包，需要对各信元的优先级状态进行记录静态优先级(PQ)算法中优先级是静态配置的，不会随时间的流逝而改变，这就是低优先级可能被“饿死”的原因。笔者利用EDF算法的思想，使优先级低的信元，随着时间的流逝，它应得到服务的优先级也应随之提高，即优先级是动态变化的。

为了实现动态优先级算法，针对进入缓冲区的每一信元，设置一个计数器[6]，使其初始值等于所属数据包的优先级；每过一个时间片，每一信元的计数器加1；每次挑选优先级最高的信元转发。在文中为了便于讨论都使用IPv4 TOS字节的前3位，8种状态来表示优先级，0(零)表示最低优先级，数字越大优先级越高(最高为7)。如图1所示，假定一个缓冲队列中有6个

收稿日期：2007－09－05
基金项目：国家自然科学基金(10571052)
作者简介：周荣华(1978－)，硕士研究生，研究方向为网络服务质量(QoS)；钱光明，教授，硕士生导师，研究方向为网络服务质量(QoS)、嵌入式系统。

信元等待处理,优先级分别为 7、6、5、0、4、3,一个时间片过后,优先级为 7 的信元将被转发,其它信元优先级增大为 7、6、1、5、4。只有优先级小于 7 的信元一个时间片过后优先级加 1。

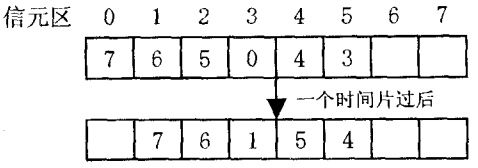

图 1 PQBEDF 的队列调度举例

1.2 动态优先级与堆实现

优先级队列与普通的先进先出队列都是队尾加入,队头删除,不同之处在于优先级队列的删除总是把具有最大值(或者最小值)的选项从队头删除,即从队列中转发出去,而从数据结构的观点来看,堆正是这样的队列的最好实现结构,"堆"要求是一棵完全二叉树,其时间复杂度均只有 $O(\log N)$。

堆中各结点的值是由优先级所构成的,那么在删除优先级最大的堆头的同时,也应该从相应的缓冲区中找到堆头所对应的信元区号,将此信元区清空(即表示该信元已转发)。因此堆中的每个结点不仅要记录优先级的值(即计数器值),而且还应记录该优先级所对应的信元区号(内存索引),因而提出一个新的概念:双单元堆[6]。在堆中每个结点由两个单元组成:第一个单元的内容为优先级的值,第二个单元的内容为该优先级所对应的信元区号,如图 2 所示,其中黑体显示的是优先级的值。

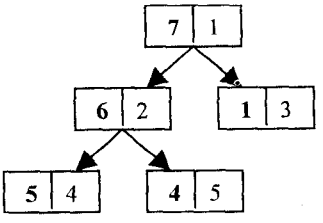

图 2 一个以完全二叉树表示的双单元堆

双单元堆实现的具体算法如下:

```
Private Sub heapinsert(ByVal n As Integer)
'进行双单元堆构造,采用自底向上构造法
n1 = 2 * (n \ 2) - 1
For i = n1 To 1 Step -2
  k = i: temp1 = h(k): kk = k + 1: temp2 = h(kk)
  heap = True
  While heap And((2 * k + 1) < = (2 * n - 1))
    j = 2 * k + 1
    If j < (2 * n - 1)Then
      If h(j) < h(j + 2)Then
        j = j + 2
      End If
    End If
    If temp1 > = h(j)Then
      heap = False
    Else
      h(k) = h(j): k = j: jj = j + 1: h(kk) = h(jj): kk = jj
    End If
  Wend
  h(k) = temp1: h(kk) = temp2
Next i
End Sub
```

从以上算法可以看出,在双单元堆的调整过程中,虽然每次地址的增量是普通堆的两倍,每次调整时要一个双单元与另一个双单元同时交换,但调整的依据仍然是优先级,也就是每一个双单元中第一个单元的内容。因此,无论是结点的删除还是插入,时间复杂度仍然只有 $O(\log n)$,不足之处是,双单元堆的空间比普通堆大。

2 基于 EDF 的优先级改进

2.1 PQBEDF 不足与概率序列的引入

如图 1,信元区 2 中的信元(初始优先级为 5)经过 2 个时间片后优先级变为 7,假设这时缓冲区进来一个优先级为 7 的信元,如果这时调度器选择原本优先级为 5 的信元转发,这样初始优先级为 7 的信元却得不到服务,在短时间内,优先级提升太快这对于原本为高优先级的信元是不公平的。为了解决计数器值对于时间片过于敏感,可以根据总的优先级个数 N 来为计数器每过一个时间片它的值加 1 来确定一组概率序列 $\{P_i\}(i=0,1,\cdots,N-1)$,而称序列 $\{T_i\}(i=0,1,\cdots,N-1)$ 为计数器每过一个时间片它的值保持不变的概率序列,具体表达见公式(1)。

$$\begin{cases} P_i = \sum_{k=0}^{i} \frac{1}{N} & i = 0,1,2,\cdots,N-1;\ N \geqslant 1 \\ T_i = 1 - P_i \end{cases} \tag{1}$$

从上式可以推出:i 的取值越大,P_i 的值也就越大。假设计数器中优先级的值与 i 值对应,说明了优先级越高,一个时间片后计数器加 1 的概率就越大,同时,低优先级信元每过一个时间片优先级增加的趋势变缓,这样就能够真正保护高优先级信元,同时也能避免低优先级信元"饿死"。

2.2 概率序列的模拟实现和结果分析

假定拿 8 个信元优先级进行讨论,通过随机函数每过一个时间片随机产生一个 0 到 7 的数字 D(也即是控制计数器值加 1 的动态门限值),当计数器中的值

大于等于 D 时,计数器的值加1,否则计数器值保持不变,但是优先级为7的信元计数器值不需要再加1。

为了验证改进的效果,采用8种优先级(0～7),将它们划分为高优先级(优先级为6～7),中优先级(优先级为3～5),低优先级(优先级为0～2)三种。动态门限值 D 的取值,通过随机函数Rnd模拟实现产生一系列的0～7之间的值,从中取8个连续产生的 D 值:5,4,4,2,2,6,0,6。通过对原PQBEDF方案与引入了 D 的动态门限值的改进方案对比结果,如图3和图4所示。

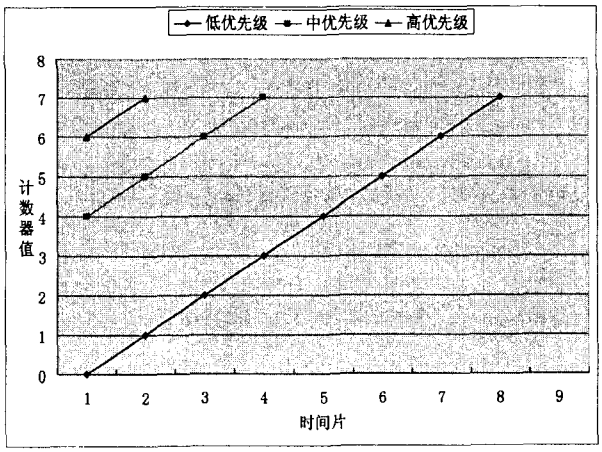

图3　原PQBEDF方案中计数器中的值随时间片变化示意图

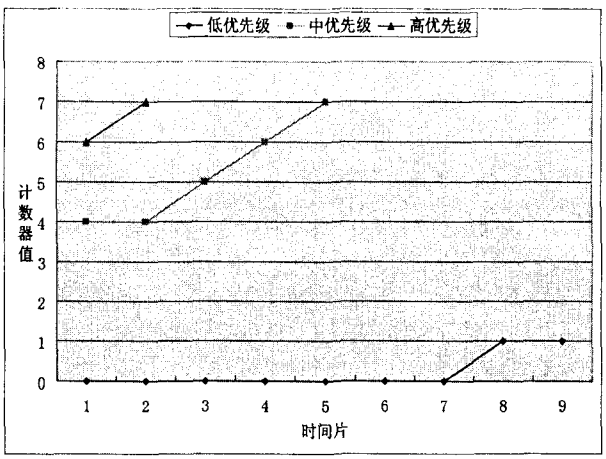

图4　PQBEDF方案改进后计数器值随时间片变化示意图

通过分析图3与图4,可以看出在原PQBEDF方案中低优先级的信元在3个时间片后就可以提升为中优先级的信元,在6个时间片后就可以提升为高优先级的信元。而通过改进后的方案,低优先级的信元经过7个时间片后虽然优先级由0提升到1但还是属低优先级范畴,这样低先级信元在几个时间片内提升为高先级信元的可能性大大减低,说明低优先级的动态变化是缓慢增加的。而对于高优先级信元而言,优先级的变化在两个方案中没有明显的差别。

2.3　PQBEDF的队列管理策略改进

PQBEDF采用的是缓冲区资源完全共享策略与简单的尾部或头部丢弃策略相结合的方式,虽然获得了相当高的系统资源利用率,但无法为用户应用提供服务质量支持,同时无法提供公平性保障。基于这些缺点,推出将完全共享策略和选择性丢弃的分组丢弃策略相结合的方案。为了便于讨论,使用IPv4 TOS字节的前3位,8种状态来表示优先级,0(零)表示最低优先级,数字越大优先级越高(最高为7)。只要队列中仍有足够的空闲缓冲资源,就允许到达的信元进入队列,并占用该缓冲资源;而当系统中的缓冲资源被完全占用时,只有最低优先级的信元被无条件丢弃,而具有较高优先级的信元则可以依据一定的策略"推出"(覆盖)队列中的已经缓冲的较低优先级的信元,并占用相应的缓冲资源,如图5所示。

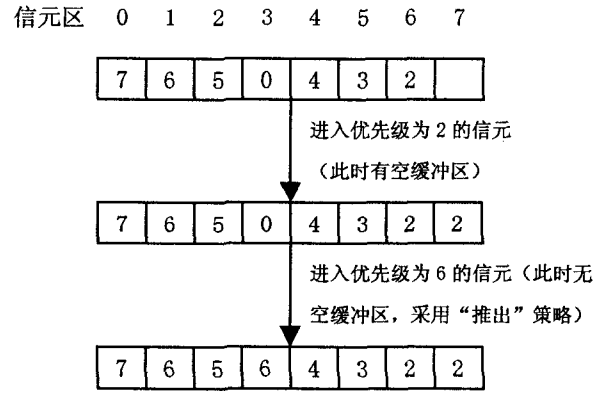

图5　PQBEDF的缓冲资源管理举例

3　结束语

利用静态优先级算法与EDF算法的优点,把优先级与时间片相结合,随时间的流逝优先级不断提高来实现PQBEDF方案中的动态优先级。通过对PQBEDF方案分析得知,优先级的动态改变对于计数器每过一个时间片加1过于敏感,因而引入一组概率序列 $\{P_i\}$,此序列的实现是通过动态门限值 D 来控制,使优先级的动态改变不致太快,这样,在保护高优先级信元得到相应服务质量的同时低优先级的信元在有限的时间内也能得到质量保证,不致长期得不到服务而"饿死"。

参考文献:

[1] 林　闯,单志广,任丰原.计算机网络的服务质量(QoS)[M].北京:清华大学出版社,2004.

[2] Zheng W. Internet QoS: Architectures and Mechanisms for

(下转第89页)

Web Services时,工作流系统应该有专门在客户端处理交互工作的API包(或者控件),业务系统直接使用该API包(或者控件),而不需要执行处理Web服务或者EJB调用等[6]。

XForms并非唯一的选择,还有其他技术可以应用到JBPM中来实现快速原型开发和与业务系统进行交互(数据交换和状态展示)。有些工作系统(如Runa)支持FreeMaker,HTML表单等多种状态展示方式。而JBPM-3.2控制台中则引入了JSF作为工作流程中的状态展示方式。总的来说,各种系统使用这些表现层技术,主要目的在于将流程状态与用户接口(HTML,Freemaker,JSF)对应起来,从而向终端用户展现流程状态,同时接收输入。

用XForms进行工作流引擎服务和业务系统之间的交换,主要有以下几个优势:

(1) 使用可视化设计工具可以帮助业务人员设计业务流程,或者可以帮助开发人员快速开发系统原型。

(2) 使用XForms展现与流程相关的数据并处理数据交互,从而将这些数据交互工作交给工作流系统内部处理,而不需要业务系统干预。当流程变化时,不会带来业务系统的变化。图3中,实线代表XForms与工作流之间的交互;而虚线是不使用工作流的情况。

(3) XForms直接访问业务系统数据,可以通过URL的形式,也可以通过Java接口的形式。其中,URL形式可以是XML文件,也可以是返回XML的服务器URL,例如Servlet等。

(4) Web服务和EJB访问:XForms插件提供了对于Web服务的访问,这对于在任何需要的时候在您的系统中支持EJB和Web服务等,是有益处的。

(5) 数据验证:XForms提供了便利的数据验证机制,可以简化业务系统客户端和服务器端的数据验证工作。

(6) XForms中集成了AJAX技术以提升用户体验,从而在提升软件可用性的同时,也节省了大量开发。

使用XForms的劣势在于:

1) 将加大工作流管理系统开发任务和以及本身复杂度。

2) 工作流客户端开发成本和难度很大。例如将工作流嵌入到JSP中存在较大的难度。

结论:可以采取分步实现的方式来完成这个目标。可以在先期使用简化的XForms代替完整功能的XForms。一个建议的方案:使用XML传递数据,同时,对于应用数据建立展现模型(简化的XForms),利用现有网络协议传递给业务端。在客户端,根据该展现模型展现数据。

5　结束语

传统意义上的信息系统没有把过程管理和应用软件区分开,业务过程被隐藏在系统中难以识别。随着我国信息化建设的加速发展,对于在各个时期构建的信息系统进行业务流程的整合与管理已经提上日程。工作流技术为适应复杂的应用环境必然要借鉴日趋成熟的J2EE、Web Services、XForms等技术与标准。文中在广泛的研究与技术实践的基础上提出了一种交互接口设计模型,具有较强的实用价值。在目前的基础上,还将对JBPM BPEL Extension做细致的研究与技术实践,降低实现Web Services的技术复杂度,并在Façade下面提供多个delegate来提高工作流的可扩展性。

参考文献:

[1] Workflow Management Coalition. The Workflow Reference Model[S].1995.

[2] Workflow Management Coalition. Workflow Managnement Application Programming Interface (Interface 2&3) Specification[S].1998.

[3] Jang Jinyoung, Choi Yongsun. uEngine: Web service based Workflow Management System[EB/OL].2004-02.http://uengine.sourceforge.net/files/uEngine-web-service-support.pdf.

[4] Graham S. Building Web Services with Java : Making Sense of XML , SOAP , WSDL , and UDDI[M].刘晓晖,等译.北京:机械工业出版社,2003.

[5] Workflow Management Coalition. Workflow Standard Interoperability Abstract Specification[S].1999.

[6] 夏　冬,白树仁,邓惠建. 基于J2EE的工作流管理系统模型[J].计算机工程与科学,2006,28(3):123-133.

(上接第85页)

Quality of Service[M].[s.l.]:Morgan Kaufmann,2001.

[3] Dysan D M. QoS & Trdfie Management in IP&ATM Networks[M].New York:McGraw—Hill Companies,Inc,2000.

[4] Charidar M, Naor J, Schieber B. Resource Optimization in QoS Multicast Routing of Real-Time Multimedia[J].IEEE/ACM Trans on Networking,2004,12(2):340-348.

[5] Marsan M A, Bianco A, Giaccone P, et al. Packet-Mode Scheduling in Input-Queued Cell-Based Switches[J].IEEE/ACM Trans on Networking,2002,10(6):666-678.

[6] 钱光明.基于业务的多优先级队列区别服务方案[J].计算机工程与应用,2006,42(10):118-120.