分组交换网络中队列调度算法的研究及其展望

王重钢,隆克平,龚向阳,程时端
(北京邮电大学程控交换技术与通信网国家实验室,北京 100876)

摘 要: 本文主要讨论分组交换网络中的队列调度算法,对现有的调度算法进行了分类和比较研究,分析了其性能指标和技术特点,最后结合我们的相关研究工作讨论了未来的发展趋势并给出了有待研究的一些课题.

关键词: 队列调度算法; 交换节点; 通用处理机共享; 分组公平排队; 服务曲线; 动态分组状态
中图分类号: TN393.0 **文献标识码**: A **文章编号**: 0372-2112 (2001) 04-0553-07

The Study and Perspective of Queue Scheduling Algorithms in Packet Switching Networks

WANG Chong-gang, LONG Ke-ping, GONG Xiang-yang, CHENG Shi-duan
(National Laboratory of Switching Technology & Telecommunication Networks Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: In this paper, we mainly discuss quesue scheduling algorithms in packet switching network. We firstly classify the recent scheduling algorithm, then analyze their performance objectives and technology attributes thoroughly. At last, we give out our related research works and discuss their future trends and present several topics remained to be studied.

Key words: queue scheduling algorithm; switching node; generalized processor sharing; packet fair queuing; service curve; dynamic packet state

1 引言

队列调度算法运行在网络节点中发生冲突需排队等待调度之处,它按照一定的服务规则对交换节点的不同输入业务流分别进行调度和服务,使所有的输入业务流能接预定的方式共享交换节点的输出链路带宽,如图 1 所示,输入业务到达交换节点后,分别暂存到相应的队列中,假设总共有 N 个队列,队列调度算法的任务是如何从这 N 个队列中选择下一个要传输的分组. 如何把输入业务流对应到不同的队列中,不同的调度算法在不同的网络环境里有不同的方法,先到先服务(FCFS: First Come First Service)只根据分组的到达时间对之进行服务,这时队列数为 1,这种调度算法的力度较大,因为把所有输入业务流无区别地放在一个队列里. 而较复杂的调度算法则会根据一定的规则把输入业务流对应到不同的队列里,从而对输入业务进行有区别的服务. 比如在因特网中,可以基于网络层源/目的地址和传送层源/目的端口对输入业务流进行分类,每一类可能对应一个队列(目前因特网业务流分类算法还是有待进一步研究的课题[1]). 而在基于异步转移模式(ATM. Asynchronous Transfer Mode)的网络中,则可基于虚通道标识(VCI: Virtaual Path Identifier)和虚通路标识(VPI: Virtual Path Identifier)对输入业务进行分类. 本文不讨论的队列调度算法不涉及到对业务流的分类,只讨论对不同业务流所属队列的调度.

根据不同的服务规则,队列调度算法可以分为以下几种: 先到先服务、循环调度、处理机共享、优先级服务、随机服务等. 根据调度算法的调度目标,也可分为基于时延的和基于速率的两类. 根据通信网络环境,又可分为无线环境下的队列调度和有线环境下的队列调度(注: 本文只讨论有线环境下的队列调度),对于无线环境下的队列调度算法,基本上是在有线队列调度算法基础上考虑到无线网络环境下的一些特定情况,进行了相应的改进,有兴趣的读者可以查阅文献[2~5]). 根据调度算法的工作状态,又可以分为工作保持和非工作保持[6]. 其中工作保持算法表示只要系统中有等待分组,调度算法就一定会工作; 而非工作保持算法则意味着即使系统中有等待分组,调度算法也可能暂时不对其进行调度,这类调度算法一般要求在输入业务流被调度之前需经过一个整形器进行整形处理. 工作保持算法具有更高的链路利用率,而非工作保持算法能对端到端时延及时延抖动进行控制. 文[6]对它们进行了较为详细的分析比较. 实际上,对于某一特定的调度算法,根据不同的分类标准,又可属于多个不同的类,所以并没有一种统一的分类标准. 根据调度算法的服务规则、调度目标及其发展趋势,同时为了能更清晰地说明各类算法之间的区别,本文把目前已出现的队列调度算法大致分为如下五类: 基

收稿日期: 1999-09-30; 修回日期: 2000-11-26
基金项目: 国家自然科学基金(No. 69972008)

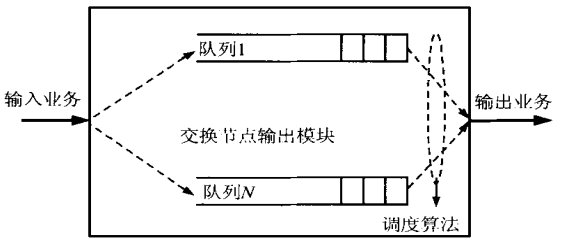

图 1 队列调度算法的原理

于轮循的调度算法、基于通用处理机共享(GPS: Generalized Processor Sharing)的算法、基于时延的调度算法、基于服务曲线(Service Curve)的算法及一类新的调度算法. 本文首先简要讨论队列调度算法应达到的性能指标,然后对这几类算法分别进行详细的分析和比较,最后讨论该领域有待研究的一些课题及我们的相关研究工作.

2 队列调度算法的性能指标

队列调度算法性能的好坏主要涉及到时延性能、公平性、复杂性这三个方面. 队列调度算法可能在不同环境下有不同的应用. 例如,队列调度算法可能被用于隔离恶意业务流来为正常业务流提供服务质量保证;队列调度算法还可能用来让用户平等地共享链路的可用带宽;或者用来实现分级的链路共享等. 实际上,有效的队列调度算法应该拥有诸多好的特性[7],即下面要讨论的队列调度算法应达到的主要性能指标.

时延性能:队列调度算法应为不同的业务流提供端到端的时延保证,而且只与此业务流的某些参数(如带宽需求等)有关,而与其他的业务流无关. Stiliadis 和 Varma[8]首先提出了一种分析网络中不同队列调度算法带来的端到端时延的模型;时延速率调度器(LRS:Latency-Rate Server). Francini[9]随后又提出了另一种分析端到端时延的模型:速率分隔时签调度器(RST: Rate-Spaced-Timestamp Scheduler),此模型的限制条件比 LRS 要少且在定长分组环境下应用时更加有效,详见文献[9,10].

公平性:可用的链路带宽必须以公平的方式分配给共享此链路的各业务流:此外队列调度算法必须能够隔离不同的业务流,让不同的流只享用自己可以享用的带宽,这样即使存在恶意或高突发性业务,它也不致影响到其他的正常业务流. 一个不公平的调度算法可能会在一较短的时间间隔里给预约了相同带宽的两个业务分配不同的服务速率. 关于算法公平的定义有:服务公平指数(SFI:Service Fairness Index)[11,12]和最坏公平指数(WFI:Worst-case Fairness Index)[11~13]两种. SFI 表示任意两个活动队列在任意时间间隔内受到的归一化服务量(等于服务量与其分配的服务速率的比值)的最大差值;WFI 用来表示一个队列在分组级系统和相应流系统上接受到的服务量的最大差值,较大的 WFI 意味着调度输出业务较大的突发性.

复杂性和可扩展性:调度算法实现起来应该比较简单. 在高速网络中,传输一个分组的时间很小,所以调度算法必须在短时间里完成对分组的调度,这就要求调度算法尽量简单,易于实现. 另外当业务流数量增加和链路速率变化范围较大时,调度算法仍应有效工作;这要求调度算法应该具有良好的可扩展性.

3 现有队列调度算法的性能比较

3.1 基于轮循的调度算法

传统的轮循(RR:Round Robin)算法对不同队列(业务流)进行无区别的循环调度服务. 这样,如果不同的队列具有不同的分组长度,则分组长度大的队列可能会比分组长度小的队列接受更多的服务,使队列之间产生不公平的现象;而且,这种算法不能对业务提供时延保证. 为了改进 RR 算法的时延特性和其在变长分组环境下的不公平性,出现了一些改进型的算法,如加权轮循(WRR[14]:Weighted Round Robin)、差额轮循(DRR[15]:Deficit Round Robin)、紧急轮循(URR[16]:Urgency-based Round Robin). 这些算法力图在尽量保持 RR 算法实现简单性的同时,从不同的方面改进 RR 算法的时延特性和其在可变长分组环境下的不公平性.

WRR[14]算法最初是用在 ATM 交换机上,它在 RR 的基础上为每个队列赋予了一个权值(可以理解为信元数),同时为每个队列维护一个计数器. 在每次轮循时,计数器为非零的队列可以允许发送仅一个信元. 计数器的计算方法为:初值为权值;每发送一个信元就减一:当所有队列的计数器为零时,则都重置为权值. WRR 算法能提供很好的公平性,且同时以较为平滑的方式调度输出业务.

DRR 算法的提出是为了解决传统 RR 算法的不公平性,与 WRR 有一些相似之处. DRR 也为每个队列赋予了一个计数器. 在每次轮循时,只有待发分组长度小于计数器值,才允许发送分组. 计数器的计算方法为:初值为定额值;每发送一个分组就减去此分组长度值;每经过一次轮循就加上定额值. DRR 解决了传统 RR 算法中由于变长分组带来的队列间的不公平性,从而可以应用于变长分组的环境,且实现较为简单. DRR 的缺陷在于不能很好地满足业务的时延特性,不能有效地支持实时业务,不能象 WRR 那样以较为平滑的方式调度输出业务.

URR 算法的主要目的是在不过度提高复杂性的情况下,改善传统 RR 算法的时延特性. 在 URR 算法中,系统给每一个队列赋予一个紧急参数 $U_i(t)$,在每一轮循环开始之前,算法都要计算每个队列的 $U_i(t)$ 参数(等于其队列长度与其速率的比值),并按降序排列,服务顺序从大到小. URR 算法改善了 RR 算法的时延特性,但仍然存在传统 RR 算法中的不公平性.

我们可以在 DRR 算法的基础上,再引入 URR 算法中"动态改变对队列循环调度次序"的思想,来实现在变长分组环境下的公平性和可以得到改善的时延特性,对时延特性的改进效果有待进一步的分析和探讨.

3.2 基于 GPS 模型的 PFQ 调度算法

GPS 是一个理想化的流模型[17],它根据各队列的共享比例对所有的活动队列同时服务,所以能使各业务流真正公平地共享链路带宽. GPS 对每个队列业务流保证有明确的端到

端的时延上限,而且与其他队列业务流无关. GPS 模型是流系统,但是实际的系统都是分组系统:在任何给定的时刻只能有一个分组可以得到服务,分组的传输是不能被抢占的. 因此出现了一类用来逼近基于流的 GPS 模型的分组算法:分组公平排队(PFQ:Packet Fair Queueing)算法.

PFQ 逼近 GPS 模型的方法是[13]:在假设无后续分组到达的前提下,选择在 GPS 中最先接受服务或最先完成服务的分组并对之进行调度,因为 GPS 模型具有"系统中当前分组的服务完成顺序与将来的到达无关"的特性. 其具体方法是:1、引入了虚时间(Virtual Time,在文[18]中被称为 Potential)的概念,系统虚时间函数表示系统在当前活动期间已提供给所有业务流的归一化的服务量,而每个队列的虚时间函数表示此队列已受到的归一化的服务量(等于服务量与分配速率的比值);2、当分组到达时就对其赋予一个虚开始或完成时间标签(而队列的队头分组的虚时间标签就是此队列的虚时间函数值);3、系统根据时间标签的大小对所有队列的队头分组逐一进行服务(这里便涉及到一个排序的问题,对于所有 PFQ 算法都是不可避免的). 所有的 PFQ 算法都是在虚时间函数的基础上按照一定的分组选择策略对队列进行调度,不同之处主要体现在虚时间函数的计算和分组选择策略这两个方面,从而导致了算法在时延特性、公平性和复杂性方面的差异.

分组时间标签的计算如下:首先 PFQ 算法需要维护一个系统虚时间 $V(t)$;对每一个会话(Session)也维护一个虚开始时间 $S_i(t)$ 和一个虚完成时间 $F_i(t)$;$S_i(t)$ 和 $F_i(t)$ 分别在每一次会话被激活和在对属于该会话的一个分组的服务完成的时候被更新.

$$S_i(t)=\begin{cases}\max(V(t),F_i(t-)) & \text{session } i \text{ becomes active}\\ F_i(t-) & P_i^{k-1}\text{finishes service}\end{cases}$$

$$F_i(t)=S_i(t)+L_i^k/r_i \qquad (1)$$

其中:r_i 表示会话 i 所预约的服务速率;P_i^k 表示会话 i 所属队列的第 k 个分组;L_i^k 表示 P_i^k 的分组长度;$F_i(t-)$ 表示会话 i 在紧跟 t 时刻之前的虚完成时间.

从直观上说,$V(t)$ 是到时间 t 为止,每一个会话 i 应该公平地收到的一个归一化的服务量:$S_i(t)$ 代表到时间 t 为止,会话 i 实际收到的归一化的服务量. 所有 PFQ 算法的目的就是尽量使得 $S_i(t)$ 和 $V(t)$ 之间的差别达到最小,因为在 GPS 模型里,二者之间的差值为零. 系统虚时间函数的作用就是:当一个会话变为活动时用来重置这个会话的虚开始时间,以保证算法的公平性. 不同的 PFQ 算法可能有不同的虚时间的函数,它们都是力图在精确性和复杂度之间取得不同的折衷.

分组选择策略主要有以下三种[19]:(1)最小虚完成时间优先(SFF:Smallest virtual Finished time First),即系统中具有最小虚完成时间的分组先得到服务. (2)最小虚开始时间优先(SSF:Smallest virtual Started time First)即系统中具有最小虚开始时间的分组先得到服务. (3)最小合法虚完成时间优先(SEFF:Smallest Eligible virtual Finished time First),即系统中具有最小虚完成时间且"合法"的分组先得到服务,其中"合法"的含义是:若一个分组的虚开始时间不大于当前的系统虚时间,则称这个分组是合法的分组. 前两种策略的共同点是对分组的选择只用一个时间标签,可能会与 GPS 产生一个很大的偏差,会产生较大的 WFI[13],影响系统的公平性.

PFQ 算法的性能指标包括:复杂度、公平性、时延特性. 复杂度主要包括计算虚时间函数的计算复杂度和排序的复杂度,由于排序是所有 PFQ 算法都必须有的,所以不同算法复杂度的差异主要体现在虚时间函数的计算上. 目前,PFQ 主要有:WFQ[13](Weighted Fair Queueing),WF^2Q[13](Worst-case Fair Weighted Fair Queueing),WF^2Q+[20],SCFQ[12](Self-Clocked Fair Queueing),MD-SCFQ[21](Minimum-Delay SCFQ),SFQ[22](Start-time Fair Queueing),FFQ[7,23](Frame-based Fair Queueing),SPFQ[7,11,23](Start-Potential Fair Queueing),LFVE[24](Leap-Forward Virtual Clock),TSFQ[25](Time-Shifting Fair Queueing)算法等,下面分别对它们进行简要论述(性能比较请见表 1).

表 1　FPQ 算法性能比较

Scheduler	Latency	SFI	WFI	Complexity
GPS[17]	0	0	0	无法实现
WFQ[13]	$L_i/\rho_i+L_{max}/r$	$O(\max(L_i/\rho_i))$	$O(N)$	$O(N)$
WF^2Q[13]	$L_i/\rho_i+L_{max}/r$	$O(\max(L_i/\rho_i)$	$O(L_i/\rho_i)$	$O(N)$
WF^2Q+[20]	$L_i/\rho_i+L_{max}/r$	$O(\max(L_i/\rho_i))$	$O(L_i/\rho_i)$	$O(\log N)$
VC[12,21]	$L_i/\rho_i+L_{max}/r$	∞	∞	$O(1)$
LFVC[24]	$2L_{max}/r+L_i/\rho_i$	$3L_{max}/\rho_{min}+6L_{max}/r$	$O(N)$	$O(N\log^{\log N})$
SCFQ[12]	$L_i/\rho_i+(N-1)L_{max}/r$	$\max(L_i/\rho_i+L_i/\rho_i)$	$O(N)$	$O(1)$
MD-SCFQ[21]	$L_i/\rho_i+L_{max}/r$	*Like WFQ*	$O(N)$	$O(1)$
FFQ[23]	$L_i/\rho_i+L_{max}/r$	$O(F)$	$O(F)$	$O(1)$
SPFQ[23]	$L_i/\rho_i+L_{max}/r$	$O(\max(L_i/\rho_i))$	$O(N$	$O(\log N)$
SFQ[22]	$L_i/\gamma+(N-1)L_{max}/r$	$\max(L_i/\rho_i+L_i/\rho_i)$	$O(N)$	$O(1)$

表 1 中:L_i 表示 Session i 的最大分组长度;ρ_i 表示分给 Session i 的速率;L_{max}表示系统中的最大分组长度;r 表示输出链路速率;F 表示 FFQ 算法中的帧长度;F_1 表示 DRR 算法中的帧长度.

WFQ 算法能到达很好的公平性和时延保证,但是其系统虚时间函数计算复杂度为 $O(N)$(N 为总的队列数),且具有较大的 WFI,使得输出业务的突发度增加[13]. 为了改进 WFQ 算法中 WFI 的缺点,Bennett[13]等人提出了 WF^2Q 算法,把 WFQ 的 SFF 分组选择策略改为 SEFF,但是其虚时间函数的计算复杂度仍然为 $O(N)$. 所以,随后又提出了 WF^2Q+[20]算法,对

虚时间函数的计算进行了改进,复杂度降为 $O(\log N)$,且实现了同 WF^2Q 接近的时延性能和公平性.

虚时钟(VC[12]:Virtual Clock)算法的虚时间函数的计算复杂度为 $O(1)$,时延特性同 WFQ 接近,但是公平性较差. 为了改进 VC 算法公平性差的缺点,出现了 LFVC[24]和 SCFQ[12]算法. LFVC 在 VC 的基础上引入了"隔离分组"和"提前系统虚时间函数"的机制,实现了同 WFQ 接近的时延特性和公平性,而且其虚时间函数的计算复杂度为 $O(1)$,且采用了复杂度为 $O(\log^{\log N})$ 的排序算法,但"隔离分组"机制在最坏情况下的复杂度为 $O(N\log^{\log N})$. SCFQ 则改变了 VC 中系统虚时间函数的计算方法,从而实现了较好的公平性,但是其时延特性却很差. Francini 在文[21]中提出了 SCFQ 的改进型算法 MD-SCFQ,它采用了 RPS[18]方法,且每发送完一个分组就对系统虚时间函数进行校正,以减小同每个队列时间标签的差距,从而实现了同 WFQ 很接近的公平性和时延特性,而且系统虚时间函数的计算复杂度为 $O(1)$.

文[18]中提出了设计公平排队算法的一种通用方法:速率成比例调度器(RPS:Rate-Proportional Server),引入了 Potential 函数的概念(类似 WFQ 中的虚时间函数),但它没有具体定义 Potential 函数的计算形式,设计者通过选择不同的 Potential 函数,可以在算法的公平性和复杂性之间取得一个折衷. 在文[7]中,Stiliadis 利用 RPS 方法提出了两个实际的算法:FFQ 和 SPFQ,在这两个算法中又引入了对系统 Potential 函数进行校正的机制,只是校正的频率有所不同(FFQ 每隔一个时间周期进行校正,SPFQ 则每发送完一个分组就进行校正);通过校正机制可以减小系统 Potential 值同分组时间标签的差值,使算法的公平性得到改善. TSFQ[25]算法采用了同 SPFQ 算法同样的机制,只是校正频率更快(每当发送完一个分组或每当有连接突然激活时进行校正),这样公平性得到了一定的提高. 实际上,SPFQ 和 WF^2Q+ 这两个算法是一样的(唯一的差别是分组选择策略的不同),尽管二者是由不同的作者从不同的角度提出的. 需要说明的是,RPS 模型是比 GPS 更广义的一种流模型,GPS 是其特例之一,RPS 的分组级模型几乎包括了所有的 PFQ 算法(SCFQ 和 SFQ 除外),它具有同 WFQ 一样的时延特性和一定的由 Potential 函数决定的公平性.

综上所述,在 PFQ 算法中,SPFQ 和 MD-SCFQ 算法的综合性能更好.

3.3 基于时延的调度算法

基于轮循和 GPS 模型的调度算法可以看成是基于速率的调度算法,这种算法通常为每个队列提供一定的速率保证来达到提供时延保证的目的. 而基于时延的调度算法则是以(为各队列)直接提供时延保证为目的,这类算法的代表是最早期限优先(EDF[26]:Earliest Deadline First). 在 EDF 算法中,给每个队列赋予了一个时延参数 D_i,表示系统对此队列提供的时延上界;EDF 为每个到达的分组计算一个时签(等于分组到达时间与其所属队列 D_i 值之和,时签最小的最先得到服务.) EDF 算法中由于涉及到对时签的排序,其复杂度为 $O(\log N)$. EDF 能很好地提供时延保证,但输入业务必须满足一定的属性要求. 在网络环境下,进入到下游交换节点的业务属性早已发生了变化,所以在实际中 EDF 无法提供端到端的时延保证. 为了改进这种缺点,H. Zhang 等人在文[27]中提出 RCS(Rate-Controlled Service)调度算法. RCS 在 EDF 的基础上,引入了整形机制. 到达的业务流首先进入整形器,经过整形后的业务必须满足了一定的属性要求,然后进入 RCS 中的 EDF 调度器,这样在每个节点处都能得到时延保证(但是等于整形时延与调时度延之和). 需要注意的是,由于 RCS 引入了整形机制,所以存在这种情况:某一时刻,系统中只有整形器中有分组,而调度器为空;所以 RCS 算法为非工作保持,其链路利用率有一定程度的下降.

3.4 基于服务曲线的算法

GPS 模型的局限性在于业务只用了一个参数(速率)来指定,使得时延与带宽的分配互相耦合在一起,也就是说低时延就需要高带宽. 这对于低时延、低带宽的业务就不能有一个很高的资源利用率. 为了解决这个问题,Cruz[28]首先提出了服务曲线的 QoS 模型,如图 2 所示. 每一种业务种类被赋予一条服务曲线,这条服务曲线指定了它不同时刻应该收到的最小的服务量;最上面一条为到达曲线,下面三条为服务曲线,可见对于相同的到达曲线,不同服务曲线所提供的时延是不同的. GPS 保证的是一种过原点的线性的服务曲线,低时延就需要预留高的带宽. 而用非线性的服务曲线就可以实现时延和带宽的解耦,但是一个根本的矛盾就是由于有了非线性的服务曲线和有优先级的业务,就有可能对所有的业务种类不能同时提供保证的服务曲线,也有可能不能同时保证实时性和公平性. 基于服务曲线的算法有:基于服务曲线的最早期限优先(SCED[29,30],Servcie Curve-based Earliest Deadline)和分级的公平服务曲线(HFSC[31],Hierarchical Fair Service Curve)

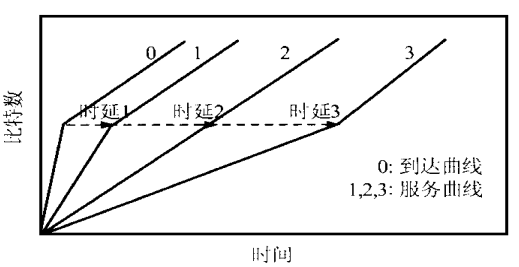

图 2 服务曲线模型

$$S(t)=\begin{cases}0, & \text{if}\quad 0\le t\le d^{\max}-1\\ b(t-d^{\max}), & \text{if}\quad t\ge d^{\max}\end{cases}\qquad(2)$$

SCED 算法中服务曲线 $S(t)$ 的定义如式(2)所示. 其中 $b(t)$ 为到达曲线,$d^{\max}$ 为最大时延:即服务曲线是直接由到达曲线平移 $d^{\max}$ 来确定,所以 SCED 算法能够保证业务的时延特性. 当有分组到达时,SCED 根据其服务曲线计算此分组应该被发送的一个期限,然后按照由小到大的顺序依次发送各分组. SCED 能保证所有服务曲线的前提是所有服务曲线之和不大于系统总的服务曲线(等于 $R\times t$,其中 R 为输出链路的总带宽). SCED 能保证业务实时性,但是却不能同时保证业务之间的公平性.

在 HFSC 算法里,服务曲线由三个参数决定:最大分组长度、最大时延、平均速率. 时延要求高的业务分配一个凸的服

务曲线,而时延要求低的业务分配一个下凹的服务曲线,其实质在于给需要小时延的业务提供一个短时的高速率(大于其平均速率),而让能忍受时延的业务在一个较短的时间里不接收服务或服务速率小于其平均速率. HFSC 中分级的概念在于让额外的带宽在相邻的兄弟种类中被公平地分配,而不是被所有的其它业务种类共享. HFSC 以"实时性"或"链路共享"标准来选择要发送的分组,且分组时签的计算复杂度为 $O(1)$. "实时性"标准在于对所有叶结点种类提供保证的业务曲线;"链路共享"标准在于给中间结点提供保证的业务曲线并公平地分配额外的带宽. 当这两条标准发生冲突时,优先满足实时性的标准. 由于 HFSC 引入了"链路共享"的标准,所以能从一定程度上改善类似 SCED 算法中的"不公平性".

3.5　一类新的调度算法

区分业务(DiffServ[32,22]:Differentiated Service)体系结构正成为解决因特网上服务质量的一种有效办法,能支持 DiffServ 技术的一个子网被称为 DiffServ 域,它由一些边缘路由器和域内路由器组成,边缘路由器执行较为复杂的业务流分类、业务量调节及队列管理和调度的功能,而域内路由器则执行较为简单的队列管理和调度的功能. 前面介绍的队列调度算法中皆没有边缘交换节点和域内交换节点的区分,而且都是基于每个业务流的调度算法,它们需要交换节点维护每个业务流的一些状态信息(比如分组的时间标签),尽管这样可以达到很好的调度性能,但同时却带来了不易扩展和不强壮的缺点.

基于这种考虑,Stoica 提出了两种新的调度方法:CSFQ[34](Core Stateless Fair Queueing)和 CJVC[35](Core-Jitter-Virtual-Clock),其核心在于对交换节点进行了(与 DiffServ 里类似的)"边界交换节点"和"域内交换节点"的区分,从而不需要每个交换节点都维护所有业务流的状态信息. 这两种调度算法的主要机制如下:1、边缘交换节点和核心交换节点需要配合运行,边缘交换节点需要保存每个业务流的状态信息,并给每个要转发的分组附上一些有关状态信息的标记,核心交换节点需要保存每个业务流的状态信息;2、算法的核心在于使用了"动态分组状态"(DPS,Dynamic Packet State)技术,即在分组传输的每一跳上,利用分组中附加的标记信息对分组进行调度,且在分组转发之前修改其标记信息以供下一站使用. 这种算法的主要优点在于免去了核心交换节点的基于每个业务流的调度,使算法的复杂性得到很大的降低. 但同时也带来了如下缺点:算法的有效运行需要边缘交换节点和核心交换节点的相互配合;需要每个分组附加一个标记,并在每次转发之前需要修改其标记;算法的时延特性较差. CSFQ 和 CJVC 算法与 DiffServ 的思想较为吻合,且降低了算法实现的复杂性,其缺点是时延性和公平性有所下降、以及需要网络中各交换节点之间的配合;如果能进一步提高这类算法的时延性和公平性,而且与 DiffServ 结合起来,则这种不基于每个业务流的调度算法是一个很好的选择.

在 CSFQ 的基础上,Zhi-Li Zhang 等人提出了一种统一的调度框架模型[36],虚时间参照系统(VTRS,Virtual Time Reference System). VTRS 具有两个优点:可以支持多种调度算法,包括 CSFQ 无状态算法和前面的 PFQ 有状态调度算法;基于 VTRS 可以方便地设计出可扩展性更好的调度算法. 另外,Zhi-Li Zhang 等人在 VTRS 的基础上,提出了一种区分业务体系结构下新的带宽代理(Bandwidth Broker)[32]结构[37],它能灵活有效地提供服务质量保证和访问控制,并且具有良好的可扩展性. VTRS 是一种较新的调度框架模型,它把较为复杂的处理功能推到网络的边缘,而在网络内部只执行相对简单的处理任务,这与 DiffServ 的思想较为一致. VTRS 值得跟踪研究.

3.6　小结

队列调度算法的目的都是以可实现的复杂性为代价来提供更好的服务质量:公平性和时延性能. 除了先入先出、优先级和传统轮循调度外,先进的队列调度算法都是把分组放到不同的队列里,然后再为其计算一个时签,根据时签的大小来对分组进行调度. 对于 PFQ 算法,其出发点在于为每个队列提供带宽保证(从而时延得到一定的保证),所以在其时签的计算中只用到了速率参数和分组长度参数;而基于时延的调度算法,则以提供时延保证为主要目的,所以在其时签的计算中,只引入了队列的时延参数. 对于 HFSC 算法,则引入了速率和时延参数,以同时提供带宽和时间保证. 在考察算法的公平性时,主要应该考察有分组堆积的队列之间的时签差值,如果此差值有限,则算法公平性就有保障. 而在分析算法的时延性能时,则可以借鉴 LRS[8]模型.

4　队列调度算法的发展趋势和我们的相关研究工作

队列调度算法虽然能提供确定性的服务质量保证,但它要求网络必须进行严格的访问接纳控制,随之而来的缺点是较低的资源利用率. 而未来的网络终端和应用可能具有更好的智能性和自适应性,它们不必一定需要确定性的服务质量;另外,网络运营商也希望网络资源能够得到充分的利用. 因此,概率意义上的"统计服务质量"(主要思想是进行较松的访问接纳控制算法,允许更多的连接进入网络,以牺牲一定的服务质量来提高整个网络资源利用率)具有一定的研究价值. 文[38～40]对"统计服务质量"方面进行了较为深入的研究,得出了"服务质量的降低"和"资源利用率的提高"二者之间的统计关系.

基于每个业务流的调度算法能对业务流进行更好的隔离和实现更好的公平性及时延特性,但是它们需要每个网络交换节点保存与业务队列数成比例的状态信息,这给骨干交换节点带来了一定的困难和负荷. 目前,这类调度算法还难于在高速网络中得到有效的应用,所以应首先考虑算法的简单性和易实现性,同时保证一定时延性能和公平性. 对于 PFQ 算法,出现了一些使算法简化的技术[19,41～45],它们的基本思想是输入业务流的速率进行离散化,把速率相同的队列归为一类,这样减少了调度算法本身处理的队列数,提高调度分组的速度. 文[45]对这几种技术进行了较为详细的描述和比较.

DiffServ 体系结构正成为研究的热点,在将来队列调度算法的研究中,可同时考虑 DiffServ 中已经定义的未来可能出现的区分业务种类,以期研究出现有针对性和更有效的队列调度算法.

在高速路由器队列调度算法的研究工作中,我们对 MD-SCFQ 算法进行了一定的研究,因为这个算法具有更好的整体性能.我们在 OPNET[46]仿真环境和微机环境下,对该算法的性能进行了大量的仿真研究,结果表明 MD-SCFQ 算法确实具有较好的时延特性和公平性,但是其虚时间函数的计算需要涉及到除法和乘法操作,所以给分组调度的速度和算法的硬件实现带来了一定的限制.在 RPS 模型的基础上,我们提出了两种更有效的队列调度算法[47,48],它们具有同 WFQ 一样的时延特性和接近的公平性,同时分组调度的速度比 MD-SCFQ 提高了 17 %左右(当队列数为 64000 时),且更容易硬件实现.另外,在文[49]中提出了 RPS 的增强模型-Enhanced RPS. ERPS 可以更加有效地指导我们设计具体的队列调度算法,并可以方便地得到其时延特性和公平性的定量分析.在下一步的工作中,我们将以 VTRS 为基础,力求研究出适合 DiffServ 的高效队列调度算法;同时对无线网络环境下的队列调度算法进行研究.

总之,未来的队列调度算法一定要适合网络带宽高速化和业务多样化的发展趋势.首先要保证高的分组调度速度,同时在时延特性和公平性方面有较好的保证.

5 结束语

随着网络带宽的提高,交换节点的分组转发速率必须加快,否则将会成为整个网络的瓶颈所在.队列调度算法是分组交换节点的一个重要组成部分,其性能的好坏将会影响到整个交换节点的性能.文中首先讨论了队列调度算法的复杂性、时延特性、公平性等性能指标,实际应用中应针对不同的情况设计不同的调度算法,以便在复杂性、公平性、时延特性方面取得一个最好的折衷:尤其是应考虑简单和易于实现.本文对目前已提出的几类基于业务流的分组调度算法进行了分析和比较,并提出了有待改进的地方.最后,给出该领域有待研究的一些课题、提出了对队列调度算法未来发展趋势的一些看法及在该领域内的一些研究工作.在实施队列调度算法的时候,除了算法本身以外,还可以在排序[10,19,31]和缓冲区管理[1,50]及分级链路共享[31,51]等方面进行考虑,以使算法易于实现且整体性能更好.

参考文献:

[1] V. P. Cumar, et al. Beyond Best Effort: Router Architecture for the Differentialed Services of Tomorrow's Internet[Z]. http://www.belllabs.com/user/stiliadi/router/router.html.

[2] T. S. E. Ng, I. Stoica, H. Zhang. Packet fair queueing algorithms for wireless networks with location dependent errors[A]. IEEE INFOCOM'98[C], San Francisco, CA, 1998.

[3] S. Lu, et al. Fair scheduling in wireless packet networks[J]. IEEE/ACM Trans. on Networking, Aug. 1999, 7(4): 473 - 489.

[4] V. Bharghavan. Fair queueing in wireless networks: issues and approaches[J]. IEEE Personal Communications, Feb. 1999: 44 - 53.

[5] A. Stamoulis, G. B. Giannakis. Packet fair queueing scheduling based on multirate multipath-transparent CDMA for wireless networks[J]. Proceeding of IEEE, Oct. 1995, 83: 1347 - 1399.

[6] H. Zhang. Service discipline for guaranteed performance service in packet-switching networks[J]. Proceeding of IEEE, Oct. 1995, 83: 1374 - 1399.

[7] A. Varma, D. Stiliadis. Hardware implementation of fair queueing algorithms for asynchronous transfer mode networks[J]. IEEE Communications Magazine, December 1997: 54 - 67.

[8] D. Stiliadis, A. Varma. Latency-rate servers: a general model for analysis of traffic scheduling algorithms[J]. IEEE/ACM Transactions on Networking, October 1998, 6(5).

[9] F. M. Chiussi, A. Francini. Implementing fair queueing in ATM switches-Part 1: A practical methodology for the analysis of delay bounds[A]. IEEE GLOBECOM'97[C], 1997: 509 - 518.

[10] F. M. Chiussi, et al. Implementing fair queueing in ATM switches-part 2: The logarithmic calendar queue[A]. IEEE GLOBECOM'97[C], 1997: 519 - 526.

[11] D. Stiliadis, A. Varma. A general methodology for designing efficient traffic scheduling and shaping algorithm[A]. IEEE INFOCOMM'97[C], 1997: 326 - 335.

[12] S. J. Golestani. A self-clocked fair queueing scheme for broadband applications[C]. IEEE INFOCOMM'94, 1994: 636 - 645.

[13] R. Bennett, H. Zhang. WF^2Q: Worst-case fair weighted fair queueing[C]. IEEE INFOCOM'96, Mar. 1996: 120 - 128.

[14] H. Shimonishi, M. Yoshida. An improvement of weighted round robin cell scheduling in ATM networks[A]. IEEE GLOBECOM'97[C], 1997, 2: 1119 - 1123.

[15] M. Shreedhar, G. Varghese. Efficient fair queueing using deficit round-robin[J]. IEEE/ACM Transactions on Networking, June 1996, 4(3): 375 - 385.

[16] O. Altintas, et al. Urgency-based round robin: A new scheduling discipline for packet switching networks[A]. IEEE INFOCOMM'98[C], 1998: 1197 - 1183.

[17] A. K. Parekb, r. g. Gallager. A generalized processor sharing approach to flow control in integrated servicesnetworks: the single-node case[J]. IEEE/ACM Trans. on Networking, June 1993: 344 - 357.

[18] D. Stiliadis, A. Varma. Rate-proportional servers: a design methodology for fair queueing algorithms[J]. IEEE/ACM Trans. on Networking, April 1998, 6(2): 164 - 173.

[19] D. C. Stephens, et al. Implimenting scheduling algorithms in high-speed networks[J]. IEEE Journal on Selected Areas in Communications, June 1999, 17(6): 1145 - 1159.

[20] J. C. R. Bennett, H. Zhang. Hierarchical packet fair queueing algorithms[J]. IEEE/ACM Trans. on Networking, Oct. 1997, 5: 675 - 689.

[21] F. M. Chiussi, A. Francini. Minimum-delay self-clocked fair queueing algorithm for packet-switched networks[A]. IEEE INFOCOMM'98[C], 1998: 1112 - 1121.

[22] P. Goyal, H. M. Vin. Start - time fair queueing: A scheduling algorithm for integrated services packet switching networks[J]. IEEE/ACM Trans. on Networking, October 1997, 5(5): 690 - 703.

[23] D. Stiliadis, A. Varma. Efficient fair queueing algorithms for packet-switched networks[J]. IEEE/ACM Trans. on Networking, 1998, 6(2): 175 - 185.

[24] S. Suri, et al. Leap forward virtual clock: a new fair queueing scheme

with guaranteed delays and throughput fairness [A]. IEEE INFOCOMM'97 [C],1997:557 - 565.

[25] J. A. Cobb et al., Time-shift scheduling: Fair scheduling of flow in high-speed networks [J]. IEEE/ACM Trans. on Networking, June 1999:274 - 285.

[26] R. Gurin, V. Peris. Quality-of-service in packet networks basic nechanisms and directions [J]. Computer Networks, February 1999, 31(3): 169 - 179.

[27] H. Zhang, D. Ferrari, Rate-contrled service discipines [J]. Journal of high speed networks, 1995, 3(4):389 - 412.

[28] L. Cruz. Quality of service guarantees in virtual circuit switched networks[J]. IEEE Journal Selected Areas in Communications, August 1995, 13(6):1048 - 1056.

[29] H. Sanriowan, L. Cruz. Scheduling for quality of services guarantees via service curves[A]. IEEE INFOCOMM'95 [C],1995:512-520.

[30] H. Sanriowan, et al. SCED: A generalized scheduling policy for guaranteeing quality-of-service [J]. IEEE/ACM Trans. on Networking, Oct. 1999, 7(5):669 - 684.

[31] I. Stoica, et al. Hierarchical fair service curve algorithm for link-sharing, real-time and priority services[J]. IEEE/ACm Trans. On Networking, April 2000, 8(2):185 - 199.

[32] K. nichols, et al. Definition of the Differentiated Service Field (DS Field) in the IPv4 and IPv6 Headers[Z]. IETF RFC2474.

[33] S. Blake, et al. An Architecture for Differentiated Services [Z]. IETF RFC2475.

[34] I. Stoica. et al. Core-stateless Fair Queueing: Achieving Approximately Fair Bandwidth Allocations in High Speed Networks [Z]: http://redriver.cmcl.cs.cmu.edu/~hzhangftp/SIGCOM98.pdf

[35] I. Stoica, H. Zhang. Providing Guaranteed Services Without Per Flow Management [Z]. http://redriver.cmcl.cs.cmu.edu/~hzhang-ftp/SIGCOM99.pdf.

[36] Z. Zhang, et al. Virtual time reference system: A unifying scheduling framework for scalable support of guaranteed services [J]. IEEE Journal on Selected Areas in Communication, Special Issue on Internet QoS, to Appear 2000.

[37] Z. Zhang, et al. Decoupling QoS control from core routers: A novel bandwidth broker architecture for scalable support of guaranteed servicel[A]. ACM SIGCOMM'2000[C].

[38] V. Siveraman, F. Chiussi. Providing end-to-end statistical delay guarantees with earliest deadine first schoduling and per-hop traffic shaping [A]. IEEE INFOCOM'2000[C].

[39] M. Andrews. Probabistic end-to-end delay bounds for earliest deadline first scheduling[A]. IEEE INFOCOM'2000[C].

[40] R. Boorstyn. Effective envelopes: statistical bounds on multiplexed traffic in packet networks[A]. IEEE INFOCOM'2000[C].

[41] F. M. Chiussi, A. Francini. Implementing fair queueing in ATM switches: The discrete-rate approach [A]. IEEE INFOCOMM'98 [C],1998: 272 - 281.

[42] F. M. Chiussi, A. Francini. A low-cost archiecture for thd implementation of worst-case-fair schedulers in ATM switches[A]. IEEE GLOBECOM'98[C], Nov. 1998.

[43] J. L. Rexford, et al. Hardware-efficient fair queueing architectures for high-speed networks [A]. IEEE INFOCOMM'96 [C], 1996: 638 - 646.

[44] R. Bennett, H. Zhang. High speed, scalable, and accurate implementation of packet fair queueing algorithms in ATM networks[A]. IEEE ICNP'99[C], Oct, 1997:7 - 14.

[45] F. M. Chiussi, et al. Adances in implementing Fair queueing schedulers in broadband networks[A]. IEEE IEEE IIC'99[C], 1999.

[46] OPNET, http://www.mi13.com.

[47] Wang Chonggang, Long Keping, Gong Xiangyang, Cheng shiduan. Fair virtual clock queueing scheduling algorithm [J]. Chinese Journal of Electronics: Jan. 2001, 10(1):42 - 47.

[48] Wang Chonggang, Long Keping, Gong Xiangyang, Cheng Shiduan. Effective fair queueing algorithms [A]. IEEE ICON2000[C], September 5 - 8, 2000, Singapore.

[49] 王重钢,隆克平,龚向阳,程时端. ERPS:一种增强的速率比例调度器[J]. 电子学报,2001,7.

[50] S. Floyd, V. Jacobson. Random early detection gateways for congestion avoidance[J]. IEEE/ACM Trans. on Networking, August 1993, 1(4): 397 - 413.

[51] S. Floyd, V. Jacobsion. Link-sharing and resource management models for packet networks[J]. IEEE/ACM Trans. on NetWorking, Aug. 1995, 3(4).

作者简介:

王重钢　北京邮电大学交换技术与通信网国家重点实验室博士研究生. 1974 年 5 月出生, 1996 年毕业于西北工业大学电子工程系, 1999 年 4 月在电子科技大学获得工学硕士学位, 1999 处 9 月考入北京邮电大学攻读博士研究生. 目前研究方向为网络流量控制和拥塞控制、队列管理和队列调度及无线网络服务质量. 电子邮件: pgzhang@bupt.edu.cn

隆克平　1968 年 5 月出生于四川省通江县, 1998 年获电子科技大学博士学位, 1998 年 9 至 2000 年 8 月,北京邮电大学通信网国家重点实验室博士后研究,现为该室副教授、硕士生导师. 主持和承担过国家级、省部级及国际合用项目 8 项. 发表学术论文近 50 篇,其中第一作者近 30 篇. 主要研究方向: SDH/ATM 网络生存性、TCP/IP 协议改进机制及性能分析、增强 Internet 实时多媒体业务 QoS 保障的策略及其实现机制、IP/ATM 综合技术、移动 IP 技术及应用、路由器的队列调度和缓存管理策略及算法等.