ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.18, No.10, October 2007, pp.2605−2612
DOI: 10.1360/jos182605

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于时槽预定的加权公平调度策略*

李 季[1+], 曾华燊[1], 郭子荣[1,2]

[1](西南交通大学 信息科学与技术学院,四川 成都 610031)

[2](包头铁路工程学校,内蒙古 包头 014040)

Timeslot-Reservation Based Weighted Fair Scheduling

LI Ji[1+], ZENG Hua-Xin[1], GUO Zi-Rong[1,2]

[1](School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, China)

[2](Bao Tou Railway Engineering School, Baotou 014040, China)

+ Corresponding author: Phn: +86-28-87601745 ext 602, E-mail: leejee@163.com, http://sist.swjtu.edu.cn/

Li J, Zeng HX, Guo ZR. Timeslot-Reservation based weighted fair scheduling. *Journal of Software*, 2007, 18(10):2605−2612. http://www.jos.org.cn/1000-9825/18/2605.htm

Abstract: EPFTS (Ethernet-oriented physical frame timeslot switching) is designed to carry the most popular data link frames, i.e., Ethernet MAC (media access control) frames, and the transmission time for an Ethernet-oriented physical frame (EPF) is defined as a timeslot for transmission and switching the physical layer frames. To resolve the data scheduling problem in EPFTS switches, this paper proposes a new type of scheduling mechanism called TRWFS (timeslot-reservation based weighted fair scheduling) based on the characteristics of EPFTS. The principle of TRWFS is to control the request time of inputs to outputs according to the number of reserved timeslots of each traffic flow and utilize a two-phase iteration mechanism to resolve request conflicts problem. This paper also puts forward three algorithms of TRWFS to show that the implementation complexity of TRWFS is about the same as round-robin based scheduling algorithms. Further simulation results show that even under heavy load conditions, TRWFS algorithms can better guarantee reserved timeslots of each I/O pairs and get better delay and throughput performance compared with other typical algorithms.

Key words: SUPANET; EPFTS (Ethernet-oriented physical frame timeslot switching); input queuing; timeslot reservation; fair scheduling

摘 要: 面向以太网的物理帧时槽交换(Ethernet-oriented physical frame timeslot switching,简称 EPFTS)技术以用户域内使用最为广泛的以太网 MAC(media access control)帧为运载对象、以定长物理层帧 EPF(Ethernet-oriented physical frame)的传输时间为时槽,作为数据传输与交换的基础.针对 EPFTS 交换技术的特点,提出了一类新的调度策略——时槽加权的公平调度原则(timeslot-reservation based weighted fair scheduling,简称 TRWFS),以解决 EPFTS 交换机中的业务数据调度问题.TRWFS 以连接建立阶段各业务流预定的时槽数为基础,控制交换矩阵仲裁过程中各输入端向输出端请求转发信元的时刻,借用一般轮询算法的二相迭代机制来解决端口冲突问题.还给出了 TRWFS 的 3 种实现算法,表明 TRWFS 的实现复杂度可与一般 Round-Robin 调度算法相当.仿真实验结果进一步表

* Supported by the National Natural Science Foundation of China under Grand No.60372065 (国家自然科学基金)
Received 2006-03-10; Accepted 2006-10-31

明,即使在重负载条件下,TRWFS 仍可有效保障 EPFTS 交换机各端口对上的预定时槽数,并在平均传输时延和吞吐率保障方面优于其他经典调度算法.

关键词: 单物理层用户数据传输平台网络;面向以太网的物理帧时槽交换;输入排队;时槽预定;公平调度

中图法分类号: TP393 **文献标识码:** A

光纤通信技术,特别是密集波分复用技术的迅速发展,将单根光纤上的数据传输率提高了几个数量级,达到了 40Gbps,并很快将达到 80Gbps.这为将来实现有线电视网、电话交换网和计算机网络的在物理上的合并、提供综合的数字网络业务奠定了通信基础.但视频、音频等新型业务的传输有确定的服务质量要求,传统因特网的"尽其所能"服务已难以满足这一要求.为了满足下一代因特网上新型应用业务的需要,西南交通大学四川省网络通信技术重点实验室提出了一种新型网络体系结构——单物理层用户平面的体系结构(single physical layer u-plane architecture,简称SUPA)[1].SUPA的核心技术是一种新型的具有服务质量保证的多粒度交换技术——面向以太网的物理帧时槽交换(Ethernet oriented physical frame timeslot switching,简称EPFTS)[2].我们称支持SUPA的网络为SUPANET,称支持EPFTS的交换节点为EPFTS交换机.

EPFTS 结合了光纤数据传输与电域内数据进行交换与处理的技术特点,以传输一个定长物理帧的时间——"物理帧时槽(physical frame timeslot,简称 PFT)"作为复用波长信道和传输交换的基本单元.SUPANET 以虚连接的方式向用户(业务)提供服务,一次完整的服务应包含连接建立、数据传输、连接撤除 3 个阶段.连接建立阶段主要用于在网络和用户之间对服务质量进行协商并预留资源以最终形成一条端到端的虚通路(virtual path——由沿途虚线路联接而成).EPFTS 则以业务流在沿途各波长内预订并保证单位时间内需要传输的时槽数的方式来保障业务流传输的服务质量.因此,在 EPFTS 交换机上严格控制每条虚线路的数据吞吐率,并尽量减少时延抖动,是保障实时业务流服务质量的关键.关于 SUPANET 和 EPFTS 的更多说明请参考文献[1,2].

交换机内部主要由输入和输出单元以及用来连接输入、输出单元的交换结构组成.高速交换环境下,交换机内交换结构多以 Crossbar 为主.交换机内部多业务流的交叉传输控制则主要由相应的排队调度机制实现.本文针对 EPFTS 交换技术的特点和 EPFTS 交换机采用单级 Crossbar 交换结构的前提下,提出了一类基于预定时槽数量进行公平调度的策略(timeslot-reservation based weighted fair scheduling,简称 TRWFS),目的是用以有效区分并保障不同输入输出端口对上聚集业务流的数据吞吐率.

本文第 1 节简要介绍基于 Crossbar 的两类调度策略以及经典实现算法.第 2 节具体描述 TRWFS 调度策略,并给出其实现的 3 种形式.第 3 节通过充分的仿真实验以证明该调度策略及其算法的有效性.第 4 节进行总结并结束全文.

1 相关研究工作

Crossbar 是一种无阻塞的交换结构,以时隙同步方式运作,在一个服务时隙内受限于每个入口(或出口)最多与一条出口(或入口)连通的条件,Crossbar 最多可以在 N 对输入输出、端口之间同时建立连接(N 为交换机规模),也即一个服务时隙内每个入口(或出口)最多发送(或接收)一个信元,同一个服务时隙内最多有 N 个信元被交换,在输入单元内部其他已接收信元则进入缓存等待调度.本文中我们始终作如下假定:

(1) 交换机端口数量为 N,且各端口线速率相同;

(2) 交换机内部以定长信元的方式交换数据;

(3) 交换结构加速比[3] S=1,即交换结构的服务时隙等于交换机端口发送或接收一个信元的时延;

(4) 各输入单元上的缓存按VOQ(virtual output queueing)方式存储等待信元[4].

为避免入口或出口上的信元发送发生冲突,在交换机内部设置仲裁机构用于协调各端口之间的信元交换过程.仲裁结构上运行的调度算法用于在每个服务时隙内从各输入缓存选择无冲突的信元进行交换.

目前,Crossbar上的输入排队调度策略可分为基于信元排队状态和基于预定速率两类[5].前者的调度问题等于解决一个二分图的匹配问题:在每个服务时隙内,入口向出口的发送请求形成一个请求二分图,其中入口和出

口分别构成二分图的左边顶点和右边顶点,入口向出口的发送请求构成二分图的边,根据需要,每条边还可以赋予不同的权值,每个服务时隙开始之前,调度算法应根据优化目标选择无冲突的边以允许其在本时隙内传输信元.选定的优化目标可以是稳定婚姻匹配[6,7]、极大匹配[8−12]、最大权重匹配[10,11]等.其中极大匹配算法一般采用轮询和迭代的思想,是基于信元的调度策略中较为简单的一类.如iSLIP算法只在所有非空的VOQ队列中寻找无冲突的传输请求,通过发送请求、请求仲裁和允许确认 3 个步骤进行迭代,以获得一个极大匹配.文献[13]证明了在均匀分布的业务到达模式下iSLIP(iteration round-robin match with slip)算法仅需一次迭代即可取得 100%的吞吐率.FIRM(Fcfs in round robin matching)算法通过改进iSLIP算法中输出端轮询指针的更新规则进一步降低了吞吐时延.DRRM(dual round-robin matching),EDRRM(exhaustive service dual round-robin matching)以及iSLOT[12]算法将iSLIP的三相迭代过程减少为二相迭代,进一步降低了复杂度,提高了算法对突发业务的自适应能力.

第 2 类基于预定速率的调度策略的前提是交换机各端口的业务负载已知,即速率需求矩阵R已知且是允许的,$R=(r_{ij})_{N\times N}$,其中,r_{ij}是输入端i到输出端j上聚集业务流的总需求速率[13−15].该类调度基本策略是,先将速率需求矩阵分解为若干匹配矩阵,每个服务时隙选择其中一个匹配矩阵,按匹配矩阵调度信元,使得平均T个服务时隙内各端口对上获得的服务时隙数k大于该端口对上的应保障时隙数量$k'=[T\times r_{ij}]$.idling-WRR(weighted round robin)算法[14]则是根据速率需求矩阵选取一个合适帧长(等于f个时隙),通过Slepian-Duguid算法计算一个规模为$N\times f$的时隙分配矩阵,该矩阵每个列向量对应一个时隙的匹配矩阵,以此作为每个服务时隙的信元调度依据,每f个时隙循环一次,使得每个服务周期内各输入、输出对获得的连接次数与其需求相当,从而保证其速率需求.类似地,在BvN- switch调度机制中,当速率需求矩阵R已知时,根据Birkhoff及von Neumann分解定理,最多存在$K(\le N^2-2N+2)$个实常数ϕ_i和K个匹配矩阵$P_i,i=1,2,\ldots,K$,可将R的双随机矩阵 $\tilde{R}$ 表示为如下线性代数和:

$$\tilde{R}=(\tilde{r}_{ij})_{N\times N}=\sum_{i=1}^{K}\phi_i P_i,\quad \sum_{i=1}^{N}\phi_i=1\ (0\le r_{ij}\le\tilde{r}_{ij}\le 1,\sum_{i=1}^{N}\tilde{r}_{ij}=1,\sum_{j=1}^{N}\tilde{r}_{ij}=1).$$

每个调度时间片内,调度器选择其中一个匹配矩阵,据此对信元进行调度.因此,BvN-switch通过分解速率需求矩阵将信元调度转变为K个置换矩阵的调度问题.调度时,BvN-switch利用WFQ(weighted fair queueing,即PGPS(packetized generalized processor sharing)[10])的调度机制,以系数ϕ_i为权值对置换矩阵P_i进行调度.BvN-switch在最坏情况下的服务延迟时间为$O(N^2)$,文献[5]还提出通过提高加速比$S(1<S<2)$,使最坏情况下的服务时间降到$O(N)$,但本文不倾向提高加速比,相关调度方案这里不再赘述.

以上两类调度策略的主要区别在于:前者无须提前明确各交换节点上的速率需求,仅根据各时隙队列状态调度,较后者更能适应业务流需求速率的动态变化;后者在确定的速率需求矩阵 R 基础上,按服务周期(如 f 个时隙长的帧)进行服务,服务周期内,每个服务时隙的调度是在服务开始之前即已确定的,当业务需求速率发生变化时,需要运行速率矩阵分解算法重新生成新的匹配矩阵,运算复杂度较高,因而更适合于速率需求矩阵长期不变的环境.

2 TRWFS 调度策略

本节将提出一种新的基于单级 Crossbar 的输入排队调度策略,该调度策略综合了以上两类调度策略的优点,通过对 EPFTS 交换机不同端口对上的聚集业务流进行区分,保障各聚集业务流上已预定的传输时槽数量.在 EPFTS 交换环境下,各端口最大吞吐速率以单位时间内可传输的定长物理帧数量(即物理帧时槽数量)来表示.由于允许进入 SUPANET 的业务流在虚通路建立过程中,其速率需求映射为适当的时槽数,并在沿途所有 EPFTS 交换节点上预定,虚通路建立成功后,其在各交换节点上输入、输出端口以及预定时槽数等信息均保存于本地信息库中.EPFTS 交换机各端口对上聚集业务流的速率需求可通过查询本地信息库得知.由于 EPF 帧本身为定长的数据块,为简便起见,我们假设 EPFTS 交换机即以 EPF 帧作为内部交换用的定长信元,各输入端上指向同一输出端 j 的 VOQ 子队列构成输出端 j 的虚拟输入队列(virtual input queueing,简称 VIQ).为叙述方便,我们定义以下参数或符号:

$(i\rightarrow j)$表示输入端 i 到输出端 j 的端口对,$1\le i,j\le N$(不特别指明时,i 和 j 均在此范围内发生变化);

VOQ_{ij}表示在输入端i上与$(i\rightarrow j)$对应的子队列,由入口i到达发往出口j的EPF帧均在VOQ_{ij}排队;

VIQ_{ji}表示在输出端j上与$(i\rightarrow j)$对应的虚拟子队列,VIQ_{ji}即VOQ_{ij};

$F=C/L$ 表示交换机端口每秒可传输的最大时槽数量,L 为 EPF 帧长,C 为端口线速率;

$S=(s_{ij})_{N\times N}$表示速率需求矩阵,其中,s_{ij}是指$(i\rightarrow j)$上的已预定时槽总数;

$f_i=\sum_{j=1}^{N}s_{ij}$ 表示输入端i上已预定时槽总数;$f_j=\sum_{i=1}^{N}s_{ij}$ 表示输出端j上已预定时槽总数;

$f_{\max}=\max\{f_i,f_j\}$表示所有f_i,f_j中的最大值,也即当前系统最大负载.

约定交换结构的服务时隙(service timeslot)等于一个物理帧时槽,并且速率需求矩阵S是允许的,即$f_{\max}\le F$.

2.1 基本原则

为了确保 EPFTS 交换机输入、输出端口对上预定的时槽数量能够得到满足,需要控制各端口对实际获得的系统服务速率,为此,我们在交换结构每个端口上维护一张端口(输入端口或输出端口,简称入口/出口)清单,根据各端口对上已预定时槽数量控制各端口发送或接受请求.下面分析在各端口上维护端口清单的机制和方法:

类似于WRR(weighted round robin)[16]调度算法,输入端i视所有的VOQ_{ij}为其服务对象,以$(i\rightarrow j)$上预定的时槽数s_{ij}作为队列VOQ_{ij}的权值;假设端口i上的服务周期长为M_i个服务时隙,则每个服务周期内VOQ_{ij}应获得的服务次数为$M_i\times s_{ij}/f_i$,即平均每隔f_i/s_{ij}个时隙,队列VOQ_{ij}即应得到一次服务.但由于端口发生冲突,VOQ_{ij}上的服务并不总是及时得到满足,为此,以s_{ij}为子队列VOQ_{ij}和VIQ_{ji}在每个服务时隙上的步进权值,并引入变量v_{ij}和变量c_{ij}用于记录VOQ_{ij}上应服务但尚未服务的时隙个数,分别称为累积信用和冗余信用;所有$v_{ij}=0$ 的子队列都将被拒绝服务.同样,输出端j视所有虚拟子队列VIQ为其服务对象,实施相同的控制机制.这样,各端口将为本地每个子队列新增两个变量v_{ij},c_{ij},用来记录该子队列当前的有效信用(有效信用定义为冗余信用+累积信用×端口预定时槽总数).各端口根据本地子队列的有效信用维护相应的端口清单.最终,TRWFS调度策略中每个服务时隙的仲裁过程由两部分组成:

(1) 每个端口更新本地子队列有效信用,建立相应的入口/出口清单:

a) $c_{ij}=c_{ij}+s_{ij}$;若$c_{ij}\ge M_i$,则$\{v_{ij}=v_{ij}+1,c_{ij}=c_{ij}\%M_i\}$(符号“%”表示取模运算);{本地清单维护更新;}

b) 若该服务时隙内VOQ_{ij}被服务 1 次,则$v_{ij}=v_{ij}-1$;

(2) 运行两相迭代过程[10,12]:

c) 未匹配入口基于本地出口清单选择首个未匹配出口向其发送请求;

d) 未匹配出口基于本地入口清单接收首个入口请求,并向所有请求入口发回确认信息.

在文献[17]中,作者提出了在 S=1 的基础上为输入排队交换系统提供 QoS 保证的一类算法,该算法基于将二分图中所有边赋予权重的思想,通过集中调度寻找一个稳定婚姻匹配,边的权重可以是信用加权、有效信元和有效等待时间,以保证带宽预定、信元延迟等 QoS 指标.TRWFS 调度策略与基于稳定婚姻的匹配算法的区别在于:用输入端和输出端各自的端口清单来解决端口冲突问题,以分布式而非集中式方式仲裁请求,迭代过程中已经匹配的边在后续迭代中不会拒绝.

2.2 实现算法

TRWFS 调度策略的实现主要体现在端口清单的维护方式上,我们从两个角度给出该策略的实现算法.其中,算法 1 和算法 2 是在各端口上按“有效信用越大,优先级别越高”的原则建立严格的端口优先级清单,算法 3 则将端口清单的维护隐含于两端的指针轮询过程中,算法 1 和算法 2 在服务周期的取值方式上有所不同.

算法 1. 分别选择f_i作为输入端i的服务周期,f_j作为输出端j的服务周期;各端口基于本地服务周期维护本地的端口优先级清单,v_{ij}(或v_{ji})越大,端口优先级别越高;当$v_{ij}=0$ 时,忽略端口对$(i\rightarrow j)$上的匹配请求.当时隙为 0 时,参数c_{ij},v_{ij}赋值 0;第k个服务时隙内,执行下面的伪代码:

(1) 各端口上的优先级清单维护

/*输入端i上*/$c_{ij}=c_{ij}+s_{ij}$;如果($c_{ij}>f_i$),则{v_{ij}++;$c_{ij}=c_{ij}\%f_i$};{本地优先级清单更新;}

/*输出端j上*/$c_{ji}=c_{ji}+s_{ij}$;如果($c_{ij}>f_j$),则{v_{ji}++;$c_{ji}=c_j\%f_j$};{本地优先级清单更新;}

(2) 初始化所有端口为未匹配状态,执行如下的迭代过程达到指定的次数

a) 未匹配入口选择优先级最高的未匹配出口向其发送请求;

b) 未匹配出口从多个服务请求中选择优先级最高的入口确认其请求.

(3) 如果($i\rightarrow j$)匹配成功,则输入端i和输出端j上的参数v_{ij}和v_{ji}分别减 1;

算法 2. 与算法 1 不同的是,各输入输出端口上的服务周期取相同值$f_{\max}$;其余代码同算法 1.

算法 3. 当v_{ij}(或v_{ji})≥1 时,称相应的端口对是有效的,否则称为是无效的.算法由两个阶段构成,第 1 阶段各输入、输出端基于相同的服务周期$f_{\max}$维护本地有效端口清单.第 2 阶段如同DRRM算法的迭代过程,在各输入端i和输出端j上分别维护请求轮询指针a_i和g_j,当($i\rightarrow j$)在第 1 次迭代中匹配成功时,两端的指针各向前移动到匹配端口的下一个位置;迭代过程中端口轮询时将跳过无效端口对上的匹配请求.算法 3 的伪代码如下:

(1) 端口有效性维护

/*输入端i上的参数更新*/$c_{ij}=c_{ij}+s_{ij}$;如果($c_{ij}>f_{\max}$),则{v_{ij}++;$c_{ij}=c_{ij}\%f_i$};

/*输出端j上的参数更新*/$c_{ji}=c_{ji}+s_{ij}$;如果($c_{ji}>f_{\max}$),则{v_{ji}++;$c_{ji}=c_{ji}\%f_j$};

(2) /*两相迭代过程*/

a) 输入端i从指针a_i指向的位置起,选择首个有效出口(对应的v_{ij}≥1),向其发送请求;如果请求匹配成功,则指针a_i移动到匹配出口的下一个位置,相应的v_{ij}减 1.

b) 当有匹配请求时,输出端j从指针g_j指向的位置起,批准首个有效的请求入口(v_{ji}≥1),并向所有入口发送确认信息,并将指针g_j移动到匹配入口的下一个位置,相应的v_{ji}减 1.

2.3 算法实现复杂度分析

TRWFS策略在传统的轮询迭代算法基础上,增加了基于预定时槽的速率控制机制,在空间上,每个I/O单元上需增加以下存储单元:① 本端口关联的所有端口对上预定时槽数s_{ij}共N个;② 端口负载f_i;③ 累计信用v_{ij}(或v_{ji});④ 冗余信用c_{ij}(或c_{ji}).时间上与DRRM,iSLOT等二相迭代过程相比,TRWFS增加的复杂度主要是第 1 阶段的端口清单更新过程,尤其是算法 1、算法 2 中按有效信用建立优先级清单的过程较为复杂.实现算法 2 只是改变算法 1 各端口的服务周期为相同值,用于服务周期取不同值时的对比实验;算法 3 去掉了算法 1、算法 2 中逐时隙的优先级排序过程,以较简单的轮询机制替代,而且由于各端口独立维护本地参数,因此,TRWFS实现算法 3 在第1阶段只涉及简单运算,时间复杂度非常小.而且TRWFS在第2阶段均采用了两相Round-Robin算法的框架,一次迭代中,每个端口只需作一次循环优先仲裁,且各端口的运算可并行执行,因此,总体上看,一次迭代的TRWFS实现算法 3 的时间复杂度是$O(1)$;已知循环优先仲裁器只需简单的逻辑即可实现[8],现已广泛应用于高速交换机中,因此,TRWFS调度策略具有很好的实用性.

3 仿真实验与仿真结果

为了验证 TRWFS 调度策略的有效性,我们用仿真软件实现了一个基于 8×8 Crossbar 交换结构的仿真模型(端口编号从 0~7),将 TRWFS 实现算法 1~算法 3 与 iSLIP,BvN-switch 以及 iSLOT 算法一起在多种流量模型下进行了对比实验.由于实际设计中要求交换结构的服务间隙尽可能小,特别是在高速交换环境下更是降至数百纳秒级别,因此在仿真实验中,需要时调度算法的迭代次数都设为 1.

3.1 均匀流量模型下的仿真实验

均匀流量模型是指各输入端上的信元到达过程是 Bernoulli i.i.d,并且到达信元的出口分布均匀.在均匀流量的仿真实验中统计了各调度算法在不同系统负载ρ(0<ρ<1)下的平均队列时延(单位:时槽),结果如图 1 所示(图中分别以 Algo.1~Algo.3 表示 TRWFS 实现算法 1~算法 3).

图 1 中所示的各次实验都取稳定状态下的统计值,可以看出,在平均时延上以ρ=0.75 为分界点,当系统负载

较低时,iSLIP 算法时延性能好于 iSLOT 算法,而后者时延性能又好于其他 4 类算法;当系统负载较高时,除 iSLIP 算法信元平均排队时延较高以外,其他几类算法性能接近,均好于 iSLIP 的时延性能.另外,对实验中系统吞吐率的统计结果表明:最大到 99%的系统负载下各调度算法都没有发生信元丢弃现象,表明在均匀流量业务到达模型下各调度算法的系统吞吐率都是渐近 100%,也表明图 1 中的时延统计结果是有效的.

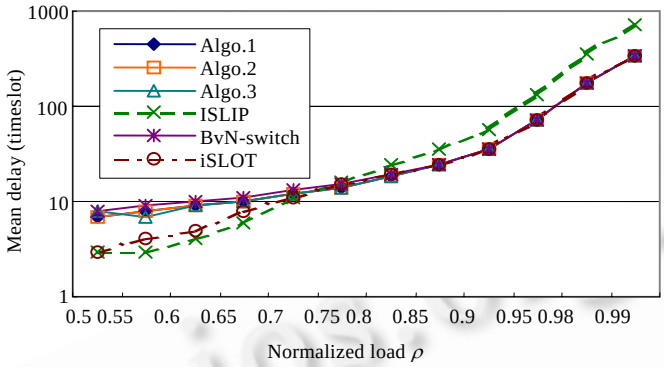

Fig.1 Comparison on mean delay under bernoulli i.i.d traffic

图 1 均匀流量下的平均排队时延比较

3.2 非均匀流量模型下的仿真实验

为了比较各调度算法尤其是 TRWFS 实现算法对不同端口对上的业务流的区分服务能力,我们参考文献[12]中提出的非均匀流量模型设计了如下两组仿真实验:

(1) 对角流量(diagonal traffic)仿真:信元到达过程是Bernoulli i.i.d的,对任一输入端口i,$r_{ii}=2\rho/3$,$r_{i,i+1}=\rho/3$,对其他输出端口,$r_{ij}=0$;仿真时设置各端口系统负载为 95%.在这种特殊的流量模型下,每个输入端除了两个VOQ上有信元到达外,其他队列皆空闲.

(2) 弱对角流量(weakly diagonal traffic)仿真:信元到达过程是Bernoulli i.i.d的,对任一输入端口i,$r_{ii}=2\rho/3$,对于其他输出端口j,$r_{ij}=\rho/(3\times(N-1))$;仿真时设置各端口对系统为 95%.

仿真实验中,我们首先统计了各调度算法下的系统平均排队时延以及吞吐率,分别如图 2、图 3 所示.

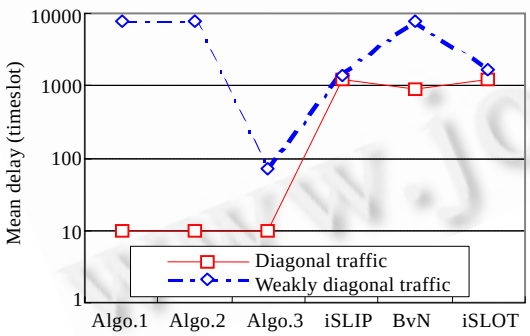

Fig.2 Mean delay under non-uniform traffic

图 2 非均匀流量模型下的时延性能对比

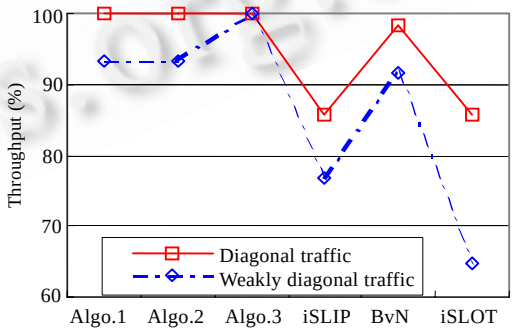

Fig.3 Throughput under non-uniform traffic

图 3 非均匀流量模型下的系统吞吐率对比

由图 2 和图 3 可知,在两种非均匀流量模型下,iSLIP,BvN 和 iSLOT 调度算法的信元平均排队时延都较大,实验中出现缓存溢出现象,导致系统吞吐率都小于 100%.产生这种现象的原因是,当各端口对上的负载不均衡时,这 3 类调度算法对端口对的系统服务速率无协调能力,无法保障各端口对上的异质业务需求,导致部分端口对上的系统吞吐率低于预定要求,引起系统性能下降.TRWFS 调度策略的实现算法在对角流量下取得一致的低时延和 100%吞吐率,但弱对角流量模型下也只有算法 3 仍然保持 100%的吞吐率,时延控制在 100 个时槽以内.

为了体现 TRWFS 策略对端口对上不同预定时槽的有效保障能力,实验中我们还统计了不同端口对上的信元平均排队时延以及实际吞吐率,其中,针对 0 号输出端口上的统计结果见表 1、表 2(表中同样以 Algo.1~Algo.3 分别表示 TRWFS 实现算法 1~算法 3),其他输出端口上的情况类似.

Table 1 The performance statistics on output No.0 under diagonal traffic

表 1 对角流量下 0 号输出端上的系统性能统计

Input No.	Algo.1	Algo.2	Algo.3	iSLIP	BvN-switch	iSLOT
0	14(100)	14(100)	14(100)	**–(77.85)**	9(100)	**–(77.68)**
7	30(100)	30(100)	30(100)	12(100)	**–(95.23)**	26(100)

注:由于其他入口上无信元到达,相应时延未记入表中.

Table 2 The performance statistics on output No.0 under weakly diagonal traffic

表 2 弱对角流量下 0 号输出端上的系统性能统计

Input No.	Algo.1	Algo.2	Algo.3	iSLIP	BvN-switch	iSLOT
0	19(100)	19(100)	16(100)	**–(63.33)**	10(100)	**–(44.73)**
1	**–(81.86)**	**–(81.86)**	197(100)	8(100)	**–(76.89)**	19(100)
2	**–(81.85)**	**–(81.85)**	191(100)	8(100)	**–(76.92)**	19(100)
3	**–(81.83)**	**–(81.83)**	207(100)	8(100)	**–(76.89)**	21(100)
4	**–(81.82)**	**–(81.82)**	166(100)	8(100)	**–(76.89)**	20(100)
5	**–(81.80)**	**–(81.80)**	161(100)	8(100)	**–(76.87)**	20(100)
6	**–(81.82)**	**–(81.82)**	199(100)	8(100)	**–(76.86)**	20(100)
7	**–(81.86)**	**–(81.86)**	205(100)	8(100)	**–(76.91)**	21(100)

表 1 和表 2 中,括号外数字是各端口对上的信元平均排队时延,括号内数字是实际信元速率与预约速率的百分比值,即吞吐率.由于没有比较意义,凡是缓存溢出或无信元到达的端口对上的时延在表 1 和表 2 中都省略并以符号“–”替代.综合图 2 和表 1 可知:6 种调度算法可分为 3 类,第 1 种是非均匀流量模型下性能一直表现较好的算法 3,各端口对上分布不均且大小不等的业务需求速率都得到了有效保障;第 2 类就是算法 1、算法 2 和 BvN-switch 算法明显倾向于尽量满足需求速率较大的端口对,从而造成其他端口对上的数据吞吐率不足,但算法 BvN-switch 的倾向程度较算法 1 和算法 2 更明显,所以吞吐率更低;第 3 类就是 iSLIP 和 iSLOT 算法倾向于公平调度的策略,虽然普遍保障了需求速率较小的端口对服务质量,但代价是降低了负载较大的端口对上的吞吐率.以上实验结果表明,算法 3 具备非均匀流量下的协调能力,可确保各端口对上的不同吞吐率需求.

4 结束语

TRWFS 调度策略是在文献[18]中提出的 TWFS(timeslot weighted fair scheduling)调度算法的基础上扩展形成的,TWFS 相同于本文中的算法 2.本文进一步提出算法 1 和算法 3,用以探讨 TRWFS 策略的多种实现形式.上面的各项实验结果表明,算法 1、算法 2 性能近似,但与算法 3 相比,不仅复杂度高,系统吞吐率低,而且对端口对上预定时槽的保障能力也较弱.另外也要注意到,本文研究的前提是 SUPANET 中时槽预定机制,即为了保障业务流的端到端的传输服务质量,要求终端应用在数据传输开始之前需先通过控制信令建立起端到端的虚通路,当虚通路建立时,对应于其业务传输速率需求的时槽数在沿途各 EPFTS 交换节点都得到了预定并在本地的业务速率需求矩阵中保存,因此,各端口对上当前活动业务流时槽需求总数是已知的.总之,TRWFS 的基本策略就在于通过各端口对上的预定时槽数量来控制二相迭代的匹配过程.本文还给出 TRWFS 的 3 种实现形式,用于探索该调度策略的复杂性和有效性.最终的实验结果表明,TRWFS 策略结合一般轮询迭代的实现方式,具有系统吞吐率高、业务流需求速率保障能力强的特点,并且在时间和空间上的实现复杂度与一般轮询算法接近,即使当业务负载发生变化时,相关参数的更新过程也比较简单,因此比较适合在高速 EPFTS 交换节点中实现.

References:

[1] Zeng HX, Dou J, Xu DY. Single physical layer U-plane architecture (SUPA) for next generation Internet. In: Comprehensive Report on VoIP and Enhanced IP Communications Services. IEC, 2004. 197–227. http://www.iec.org/pubs/pub.asp?pid=30&bsi=6

[2] Zeng HX, Xu DY, Dou J. Promotion of physical frame timeslot switching (PFTS) over DWDM. In: Annual Review of Communications, Vol. 57. IEC, 2004. 809–826. http://www.iec.org/pubs/pub.asp?pid=4&bsi=1

[3] Chuang ST, Goel A, McKeown N. Matching output queueing with a combined input/output-queued switch. IEEE Journal on Selected Areas in Communications, 1999,17(6):1030−1039.

[4] Anderson T, Owicki S, Saxes J, Thacker C. High-Speed switch scheduling for local-area networks. ACM Trans. on Computer Systems, 1993,11(4):319−352.

[5] Koksal CE, Gallager RG, Rohrs CE. Rate quantization and service quality over single crossbar switches. In: Proc. of the IEEE INFOCOM, Vol. 3. Hong Kong: IEEE Press, 2004. 1962−1973. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1354605

[6] Nong G, Hamdi M. On the provision of quality-of-service guarantees for input queued switches. IEEE Communications Magazine, 2000,38(12):62−69.

[7] Pang B, He SM, Gao W. A survey on input-queued scheduling algorithms in high-speed IP routers. Journal of Software, 2003,14(5): 1011−1022 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/14/1011.htm

[8] McKeown N. The iSLIP scheduling algorithm for input-queued switches. IEEE/ACM Trans. on Networking, 1999,7(2):188−201.

[9] Serpanos DN, Antoniadis PI. FIRM: A class of distributed scheduling algorithms for high-speed ATM switches with multiple input queues. In: Katzela I, ed. Proc. of the IEEE INFOCOM. Tel Aviv: IEEE Communications Society, 2000. 548−555.

[10] Chao HJ. Saturn: A terabit packet switch using dual round-robin. IEEE Communication Magazine, 2000,38(12):78−84.

[11] Li Y, Panwar S, Chao HJ. The dual round-robin matching switch with exhaustive service. In: Gunner C, ed. Proc. of the IEEE Workshop on High Performance Switching and Routing. Kobe: IEEE Communications Society, 2002. 58−63.

[12] Wu J, Chen Q, Luo JZ. A round-robin scheduling algorithm by iterating between slots for input-queued switches. Journal of Software, 2005,16(3):375−383 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/16/375.htm

[13] Li Y, Panwar S, Chao HJ. On the performance of a dual Round-Robin switch. In: Ammar M, ed. Proc. of the IEEE INFOCOM. Anchorage: IEEE Communications Society, 2001. 1688−1697.

[14] Hung A, Kesidis G, Mckeown N. ATM input-buffered switches with guaranteed-rate property. In: Kristine L, ed. Proc. of the IEEE ISCC'98. Athens: IEEE Press, 1998. 331−335.

[15] Chang CS, Chen WJ, Huang HY. Birkhoff-von Neumann input buffered crossbar switches. In: Sidi M, ed. Proc. of the IEEE INFOCOM. Tel Aviv: IEEE Communications Society, 2000. 1614−1623.

[16] Katevenis M, Sidiropoulos S, Courcoubetis C. Weighted round robin cell multiplexing in a general-purpose ATM switch chip. IEEE J-SAC, 1991,9(8):1265−1279.

[17] Kam AC, Siu KY. Linear-Complexity algorithms for QoS support in input-queued switches with no speedup. IEEE Journal on Selected Areas in Communications, 1999,17(6):1040−1056.

[18] Li J, Zeng HX. Timeslot weighted fair scheduling in EPFTS. Journal of Software, 2006,17(4):822−829 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/17/822.htm

附中文参考文献:

[7] 庞斌,贺思敏,高文.高速 IP 路由器中输入排队调度算法综述.软件学报,2003,14(5):1011−1022. http://www.jos.org.cn/1000-9825/14/1011.htm

[12] 吴俊,陈晴,罗军舟.时隙间迭代的输入队列交换机 Round_Robin 调度算法.软件学报,2005,16(3):375−383. http://www.jos.org.cn/1000-9825/16/375.htm

[18] 李季,曾华燊.EPFTS 中基于时槽加权的公平调度算法.软件学报,2006,17(4):822−829. http://www.jos.org.cn/1000-9825/17/822.htm

李季(1978－),男,安徽怀宁人,博士生,主要研究领域为计算机网络体系结构,高速路由与交换技术.

郭子荣(1961－),女,博士生,高级讲师,主要研究领域为计算机网络体系结构,高速路由与交换技术.

曾华燊(1945－),男,研究员,博士生导师,主要研究领域为下一代网络体系结构,高速交换技术,网络测试技术.