LHFS-支持公平服务的CICQ混合调度策略

扈红超,郭云飞,卜佑军,伊 鹏
(国家数字交换系统工程技术研究中心,河南郑州 450002)

摘 要: 针对现有联合输入交叉点排队交换结构(CICQ,Combined Input and Cross-point Queuing)调度策略无法提供基于"流"的服务质量保障,探讨了在CICQ交换结构实施基于流调度的可能性,提出一种能够为到达流提供公平服务的分层混合公平服务调度策略—LHFS(Layered and Hybrid Fair Scheduling).LHFS对每个输入、输出端口可独立地进行变长分组交换,其算法复杂度为 $O(1)$,具有良好可扩展特性.理论分析结果表明,LHFS能够为业务流提供时延上限和公平性保障.最后,基于SPES(Switching Performance Evaluation System)仿真系统对LHFS的性能进行了评估.

关键词: 带缓存交叉开关;调度策略;公平服务;分层混合
中图分类号: TN919.21 **文献标识码:** A **文章编号:** 0372-2112 (2012) 04-0717-07
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2012.04.015

A Layered and Hybrid Fair Scheduling (LHFS) Scheme for CICQ Switches

HU Hong-chao, GUO Yun-fei, BU You-jun, YI Peng
(*National Digital Switching System Engineering & Technological R&D Center, Zhengzhou, Henan* 450002, *China*)

Abstract: Providing fairness performance for traffic flows is an important requirement for today's routing and switching equipments. To meet this requirement, we first discuss the feasibility of implementing flow scheduling in this paper. Then, based on the discussion, it comes up with a layered and hybrid fair scheduling (LHFS) scheme. LHFS is a hierarchical and hybrid algorithm for CICQ (Combined Input and Cross-point Buffered) switches. With LHFS, each input and output port can schedule variable length packets independently with a complexity of $O(1)$. Theoreuical analyses show that LHFS can provide delay up-bound and fair service guarantees. Finally, we implement LHFS in SPES (Switch Performance Evaluation System) to verify the analytical results.

Key words: buffered crossbar; scheduling policy; fair service; layered and hybrid

1 引言

近年来,在交换单元内部集成一定容量的缓存成为了现实,基于带缓存交叉开关构建的联合输入交叉节点排队(CICQ:Combined Input-Crosspoint Queuing)交换结构备受关注[1].CICQ通过在每个交叉点集成了一定容量的缓存将 $N\times N$ 的交换结构分割成 N 个 $N\times 1$ 和 N 个 $1\times N$ 的子结构,并将输入、输出端口的带宽冲突隔离开来,使分布式调度成为了可能.当前,基于CICQ构建的调度策略的研究主要集中在提供高吞吐量[2~8]、模拟OQ[9~11]和性能保障方面.在提供高吞吐量的研究方面,当输入端口 i、输出端口 j 的业务流到达率 $\lambda_{ij}\leqslant 1/N$ 时,LQF-RR在无需加速的条件下即可获得100%的吞吐量[5].当交叉点缓存容量 $B>N$ 时,CICQ仅需 $\lfloor 2B/(2B-N)\rfloor$ 倍的加速便可为任意到达的"容许"业务提供100%的吞吐量.LIPS(Localized Independent Packet Scheduling)[6]基于Round Robin机制,在2倍加速条件下提供100%的吞吐量.Chang等基于CICQ结构提出了一种基于"帧"大小动态调整的交换机制,并证明了在每个交叉点维护两个信元的缓冲区便可为类似泊松到达的单播流量提供100%的吞吐量[7].最近,Y.Shen等人基于Hamiltonian行走提出了一种能够为到达业务提供100%吞吐量的调度算法—SQUID[8].

自适应的Max-Min公平调度策略AMFS(Adaptive Max-min Fair Scheduling)[12]在不加速下能够提供Max-Min公平性.基于速率的平滑交换结构sBUX(Smoothed Buffered Crossbar)[13]仅需在每个交叉点维护两个信元的缓存容量便可为到达业务提供100%的吞吐量,并能够

收稿日期:2010-08-31;修回日期:2011-12-10
基金项目:国家重点基础研究发展规划(973计划)项目(No.2007CB307102);国家自然科学基金(No.60803153);国家高技术研究发展计划(863计划)课题(No.2009AA01A346,No.2008AA01A323,No.2008AA01Z214,No.2007AA01Z218)

为每条流提供几近理想的平滑度.在变长分组交换方面,Deng Pan 提出一种提供性能保障的变长分组调度策略—FLAPS (Fair and Localized Asynchronous Packet Scheduling)[14].在 FLAPS 中每个输入和输出单元仅需本地队列信息进行调度.理论分析结果表明,无需加速 FLAPS 就可避免缓存溢出,然而 FLAPS 无法对同一端口流进行细分.

本文提出一种分层的混合调度机制—LHFS (Layered and Hybrid Fair Scheduling).LHFS 每个输入端口、输出端口可独立地进行变长分组交换.具体而言,LHFS 将到达交换系统的分组依据输入、输出端口进行分组,组间采用基于时间戳(timestamp)的公平调度机制,而组内采用基于轮询(round robin)的调度机制.LHFS 无需加速便能为到达业务提供时延上限、速率和公平性保障.本文剩余章节安排如下:第二部分对现有公平服务调度机制进行了概括和总结;第三部分阐述了 LHFS 交换机制;第四部分对 LHFS 的性能进行了理论分析;第五部分对 LHFS 算法的性能进行仿真评估;第六部分是结论.

2 分层混合调度策略

2.1 系统模型

图 1 给出了一 $N\times N$ 规模的 LHFS 交换系统总体结构,为避免队头阻塞(HOLB:Head Of Line Blocking),采用虚拟输出排队机制(VOQ:Virtual Output Queuing)缓存分组,其中 VOQ_{ijk}缓存到达输入端口 i、去往输出端口 j 的第 k 条业务流 f_{ijk};交叉点队列 XB_{ij}缓存到达输入 i、去往输出 j 的分组,不再为每条流 f_{ijk}维护独立的虚拟交叉点队列.到达交换系统的流分组首先进入虚拟输出队列 VOQ_{ijk},经输入调度后进入交叉点缓存队列 XB_{ij},最后由输出调度后发送到外部链路.由于交叉点队列的引入,每个输入、输出端口可独立地执行异步变长(Asynchronous and Variable)分组交换.

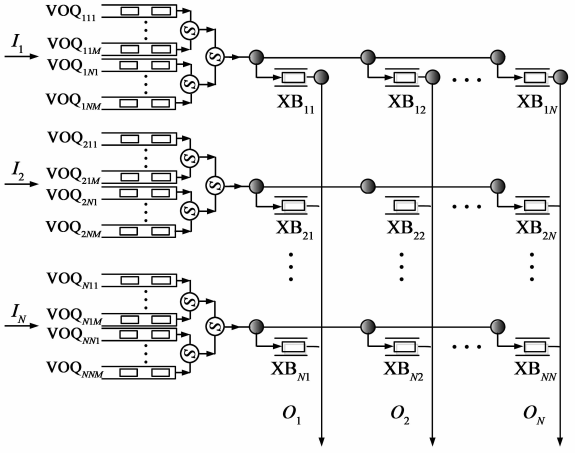


图1 LHFS交换系统结构

虽然针对共享链路的公平服务调度机制的研究成果相对颇丰,然而,由于 CICQ 结构受到输入、输出端口带宽的双重约束,f_{ijk}获得的带宽不仅受输入调度器 IS (Input Scheduler)带宽分配策略约束,同时受输出调度器 OS(Output Scheduler)的带宽分配策略约束.因而,无论是基于时间戳还是基于轮询的调度机制都无法直接应用到 CICQ 中.用 r_{ijk}标识f_{ijk}的带宽需求,r_{ijk}应满足:

$$\forall\, i,\sum_{j,k} r_{ijk}\leqslant R,\ \text{and}\ \forall\, j,\sum_{i,k} r_{ijk}\leqslant R$$

用 $w_{ijk}=\frac{r_{ijk}}{R}$表示流 f_{ijk}归一化带宽需求,则 r_{ijk}应满足:

$$\forall\, i,\sum_{j,k} w_{ijk}\leqslant 1,\ \text{and}\ \forall\, j,\sum_{i,k} w_{ijk}\leqslant 1$$

为便于描述,首先给出表 1 描述的相关符号定义.此外,为避免网络中传输的数据“分组”和 LHFS 对流进行“分组”概念的混淆,在下面的描述中,称数据“分组”为“包”.

表 1 相关符号定义

R,N,K	链路带宽、交换结构规模和流数
$r_{ijk}(t)$	t 时刻流f_{ijk}的带宽需求
$w_{ijk}(t)$	t 时刻流f_{ijk}的调度权重
$q_{ijk}(t)$	t 时刻 VOQ_{ijk}队列的长度
F_{ij}	到达输入 i、去往输出 j 的流集合(组)
B_{ij}	$B_{ij}=\{f_{ijk}:f_{ijk}\in F_{ij},q_{ijk}(t)>0\}$
F_i	到达输入 i 的流集合
p_{ij}^{HOL}	集合 F_{ij}的队首包
l_{ij}^{HOL}	p_{ij}^{HOL}的长度
$L_{\max}$	最大包长度
$SDRR_{ij}$	对应流集合 F_{ij}的集合(组)调度器
PGS_i	对应输入 i 的端口调度器
TOS_j	对应输出 j 的输出调度器
$A_{ijk}(t_1,t_2)$	$[t_1,t_2)$ 时间内流 f_{ijk}实际到达流量
IS_{ij}^{HOL}	集合 F_{ij}队首包的虚拟开始时间
IF_{ij}^{HOL}	集合 F_{ij}队首包的虚拟结束时间

2.2 LHFS 算法描述

LHFS 由输入调度器和输出调度器组成,其中输入调度由集合调度(ItraGS: Intra-Group Scheduling)和端口调度(PLS:Port Level Scheduling)组成,具体而言,将$\langle i,j\rangle$端口流集合 F_{ij}视为一调度集合;其后,PLS 采用基于时间戳的调度机制决定各流集合$\{F_{ij}:1\leqslant j\leqslant N\}$的调度顺序,而 ItraGS 基于轮询机制决定组 F_{ij}内各数据包$\{p_{ij}^1,p_{ij}^2,\cdots,p_{ij}^n,\cdots\}$的调度顺序.输出调度仍然采用基于时间戳的调度机制(TOS: Time-stamp based Output Scheduling)保证良好的公平性.

2.2.1 端口调度—PLS

端口调度策略决定集合 $F_i=\{F_{ij}:1\leqslant j\leqslant N\}$各子集合 F_{ij}的调度顺序,从 F_i 中选择一子集合 F_{ij}作为包输出集合,输入端口 i 的端口调度器用 PGS_i 表示.WFQ、

WF^{2}Q 通过引入虚拟时间 $V(t)$模拟对应的 GPS 过程,为每条流 f_k 的每个包 p_k^n 计算一个虚拟开始时间标签 S_k^n 和虚拟完成时间标签 F_k^n,每次调度时,选择队头包具有最小虚拟完成时间的流,即 $k=\arg\min\{F_l^n(t)\mid l=1,\cdots,K\}$.若带宽需求 r_k 在调度过程中恒定,则每个包 p_k^n 的虚拟开始 S_k^n 和结束时间 F_k^n 一旦计算便在系统调度过程中保持恒定,而和将来包到达过程无关.然而对于 PGS_i,包的到达/离去会导致流的短时"积压"或"空闲",引起流所在集合带宽需求的变化,进而引发流所在集合和其它集合包完成时间的变化,包虚拟开始和结束时间不再保持恒定,而和将来分组的到达相关,因而必须对包的到达/离去进行有效地跟踪.

定义 1　积压队列(backlogged queue)称 t 时刻队列 VOQ_{ijk}为积压队列,若 $q_{ijk}(t)>0$,并称流 f_{ijk}在 t 时刻为积压流,则 t 时刻 $\boldsymbol{F}_{ij}$积压流集合 $\boldsymbol{B}_{ij}=\{f_{ijk}\in\boldsymbol{F}_{ij}: q_{ijk}(t)>0\}$.

定义 2　事件(event)称包到达/离去调度器 s 为一次事件,用 e 表示,第 m 次事件 e_m 发生的时刻用 t_m 表示,可见在$[t_m,t_{m+1})$的时间间隔内集合 $\boldsymbol{F}_{ij}$的积压流集合 $\boldsymbol{B}_{ij}$无变化.

考察任意输入端口 i 的端口调度器 PGS_i,令 $r_{ij}(t)=\sum_{f_{ijk}\in\boldsymbol{B}_{ij}} r_{ijk}(t)$为 t 时刻流集合$\boldsymbol{F}_{ij}$的带宽需求,$r_i(t)=\sum_{j=1}^{N} r_{ij}(t)$为 i 端口带宽需求,则 $t\in[t_m,t_{m+1})$的 PGS_i 虚拟时间 $V(t)$定义为:

$$V(t)=\begin{cases}0, & \text{if}\quad t=0\\ V(t_m)+\tau, & \text{if}\quad w_i(t)=0\\ \max\left(V(t_m)+\dfrac{\tau}{w_i(t)},\min_j \mathrm{IS}_{ij}^{\mathrm{HOL}}\right), & \text{if}\quad w_i(t)>0\end{cases}\tag{1}$$

其中,t_m 为第 m 次流集合 $\boldsymbol{F}_i$ 的汇聚速率 $r_i(t)$的变化时刻,且 $\tau=t-t_m$.

假设 p_{ij}^{HOL}为时刻$t\in[t_m,t_{m+1})$流集合 F_{ij}的队头包,其长度为 l_{ij}^{HOL},则 p_{ij}^{HOL}的虚拟开始服务时间戳 $\mathrm{IS}_{ij}^{\mathrm{HOL}}$和虚拟完成时间戳 $\mathrm{IF}_{ij}^{\mathrm{HOL}}$的迭代过程计算如下式:

$$\mathrm{IS}_{ij}^{\mathrm{HOL}}(t)=\begin{cases}\max\{\mathrm{IF}_{ij}^{\mathrm{HOL}}(t),V(A_{ij}^{\mathrm{HOL}})\}, & \text{if } q_{ij}(A_{ij}^{\mathrm{HOL}-})=0\\ \mathrm{IF}_{ij}^{\mathrm{HOL}}(t), & \text{if } q_{ij}(A_{ij}^{\mathrm{HOL}-})\neq 0\end{cases}$$

$$\mathrm{IF}_{ij}^{\mathrm{HOL}}(t)=\mathrm{IS}_{ij}^{\mathrm{HOL}}(t)+\frac{l_{ij}^{\mathrm{HOL}}}{r_{ij}(t)},\tag{2}$$

其中,$q_{ij}(t)$为 t 时刻 F_{ij} 虚拟队列 VOQ_{ij} 的队长,$q_{ij}(A_{ij}^{\mathrm{HOL}-})$表示在 A_{ij}^{HOL}前一时刻队列 VOQ_{ij}的长度.事实上,每个 VOQ_{ij}是一逻辑虚拟队列,存放集合 F_{ij}队头包的相关信息(如 l_{ij}^{HOL}、$\mathrm{IS}_{ij}^{\mathrm{HOL}}$和 $\mathrm{IF}_{ij}^{\mathrm{HOL}}$),而实际的包仍存储在流队列 VOQ_{ijk}中.每次速率改变时 PGS_i 重新计算虚拟时间 $V(t)$,复杂度为 $O(e)$;重新计算各集合队头包的虚拟开始时间 $\mathrm{IS}_{ij}^{\mathrm{HOL}}(t)$和虚拟完成时间 $\mathrm{IF}_{ij}^{\mathrm{HOL}}(t)$的复杂度为 $O(N)$.记$\{F_{ij}:1\leqslant j\leqslant N\}$中各 F_{ij}队头包组成的集合 $P_i^{\mathrm{HOL}}=\{p_{ij}^{\mathrm{HOL}}:1\leqslant j\leqslant N\}$,$PGS_i$ 选择 P_i^{HOL} 中虚拟开始时间 $\mathrm{IS}_{ij}^{\mathrm{HOL}}(t)$不大于 $V(t)$且具有最小完成时间 $\mathrm{IF}_{ij}^{\mathrm{HOL}}(t)$的包所在集合,即:

$$j:\min_j\{\mathrm{IF}_{ij}^{\mathrm{HOL}}(t)\}\ \text{s.t.}\ \mathrm{IS}_{ij}^{\mathrm{HOL}}(t)\leqslant V(t)\tag{3}$$

从式(3)可以看出,PGS_i 最坏情况下需要从 N 个队头包中选择一个满足条件 $\mathrm{IS}_{ij}^{\mathrm{HOL}}(t)<V(t)$的包作为结果,当交换结构规模 N 给定时,其复杂度为 $O(\log N)$为较小常量值.

记 PGS_i 调度后选择的包为 p_{ij}^{HOL},其实际开始时间戳和结束时间戳分别为$\widetilde{\mathrm{IS}}_{ij}^{\mathrm{HOL}}$和$\widetilde{\mathrm{IF}}_{ij}^{\mathrm{HOL}}$,则$\widetilde{\mathrm{IF}}_{ij}^{\mathrm{HOL}}=\widetilde{\mathrm{IS}}_{ij}^{\mathrm{HOL}}+l_{ij}^{\mathrm{HOL}}/R$,随后 p_{ij}^{HOL}携带时间戳 F_{ij}^{HOL}进入交叉点队列 XB_{ij}.

2.2.3　集合调度—ItraGS

FRR 采用改进的 DRR 算法—LDRRWA(Lookahead Deficit Round Robin with Weight Adjustment),将 DRR 的一次遍历的所有包按轮询顺序组装成"帧",获得了短时 WFI 公平性.然而 FRR 需在帧边界重组下一个帧,直接带来两方面的问题:(1)帧边界重组"帧"的复杂度过高($O(K)$,其中 K 为流数目);(2)属于同一条流 f_k 的包背靠背(back-to-back)进行传输,同一条流的包最坏情况下传输间隔为 $O(L)$,其中 L 为帧长,具有较大突发度.RR/PFQ[16]对 LDRRWA 进行改进,将组"帧"复杂度降为 $O(1)$,然而仍然存在突发度问题.本文对 LDRRWA 进一步改进,采用平滑的 DRR 调度策略—SDRR (Smoothed DRR)解决 LDRRWA 的两个问题.

同 RR/PFQ,SDRR 以"帧"组织一次轮询调度过程,具体如算法(1).行(1)将集合共享份额计数器 set_deficit 清"0";行(2)~(20)对至少输出一个包的流组成的链表 active_list 进行轮询:首先为流 f_{ijk}分配份额 ζ_{ijk},更新计数器 $c_{ijk}=c_{ijk}+\zeta_{ijk}$,并置分配标识位 $s_{ijk}=1$(行(3)~(5)).若 f_{ijk}队头包 $p_{ijk}^{\mathrm{HOL}}=\mathrm{HOL}(f_{ijk})$长度 l_{ijk}^{HOL}不大于份额计数器 c_{ijk},则将 p_{ijk}^{HOL}加入到发送链表 fly_list,并更新计数器 $c_{ijk}=c_{ijk}-p_{ijk}^{\mathrm{HOL}}$;若 $l_{ijk}^{\mathrm{HOL}}>c_{ijk}$,则将 c_{ijk}累加到集合份额计数器 set_deficit = set_deficit + c_{ijk},并将流 f_{ijk}加入到"候选"流轮询链表 active_list′中.对 active_list 轮询后,非空的流 f_{ijk}转移到 active_list′中,以进行第二轮包的共享份额轮询调度过程.

行(21)至(38)为共享份额轮询调度过程,其中 active_list′中存储了队头包长度大于调度份额 c_{ijk}的流,即$\{f_{ijk}:c_{ijk}<l_{ijk}^{\mathrm{HOL}}\}$.在 set_deficit 大于"0"前,SDRR 依次轮询 active_list′:取出 p_{ijk}^{HOL},更新 set_deficit = set_deficit −

l_{ijk}^{HOL}和 $c_{ijk} = c_{ijk} - l_{ijk}^{\mathrm{HOL}}$. 由于 f_{ijk} 预用了一定的份额,若 p_{ijk}^{HOL}后一个包 $p_{ijk}^{\mathrm{next}} = \mathrm{next}(p_{ijk}^{\mathrm{HOL}})$的长度 l_{ijk}^{next}满足: $l_{ijk}^{\mathrm{next}} > c_{ijk} + \zeta_{ijk}$,即下一帧该流无法输出一个包,则将 f_{ijk}加入链表 pending_list′,否则加入 pending_list. 当 set_deficit≤0 时,调整帧发送速率,并将包加入到发送链表 fly_list,并产生新 active_list 和 pending_list(行(39) ~ (41)).

算法 1 SDRR 调度过程

SDRR 调度过程

```
1  set_deficit = 0
2  for   each flow f_ijk in active_list_do
3        if   s_ijk ≠ 1 then
4             c_ijk + = ζ_ijk, s_ijk = 1
5        end if
6        if   f_ijk ≠ NULL then
7             p_ijk^HOL = HOL(f_ijk)
8             if   l_ijk^HOL < = c_ijk then
9                  c_ijk - = l_ijk^HOL
10                 remove and add p_ijk^HOL to fly_list
11                 add f_ijk to the tail of active_list
12            else
13                 set_deficit + = c_ijk
14                 add f_ijk to the tail of active_list′
15                 break;
16            end if
17       else
18            remove f_ijk from group F_ij
19       end if
20 end for
21 for   each flow f_ijk in active_list′ do
22       p_ijk^HOL = HOL(f_ijk)
23       set_deficit - = l_ijk^HOL
24       c_ijk = c_ijk - l_ijk^HOL, p_ijk^next = next(p_ijk^HOL)
25       if   l_ijk^next < c_ijk + ζ_ijk then
26            add f_ijk to pending_list
27       else
28            c_ijk = c_ijk + ζ_ijk
29            add f_ijk to pending_list′
30       end if
31       if   set_deficit < 0 then
32            adjust F_ij's transmission rate
33       remove and add p_ijk^HOL to the fly_list
34       end if
35       if   set_deficit≤0 then
36            break
37       end if
38 end for
39 active_list = pending_list + active_list′
40 pending_list = pending_list′
41 pending_list′ = active_list′ = NULL
```

2.2.5 输出调度—TOS

输出调度 TOS 将包从交叉点缓存队列调度到输出端口,由于经输入调度对各条流的公平性进行了约束,因而输出调度仅需根据输入调度后分组携带的时间戳进行调度. 由于各输出端口具有相同的调度行为,这里仅以任意输出 j 为考察输出调度过程,第 j 个输出调度器记为 TOS_j.

TOS_j 根据交叉点队列集合 $X_B = \{\mathrm{XB}_{ij}: 1 \leqslant i \leqslant N\}$ 调度. 记 $P_j^{\mathrm{HOL}} = \{p_{ij}^{\mathrm{HOL}}: 1 \leqslant i \leqslant N\}$ 为 X_B 的队头包集合,定义 p_{ij}^{HOL}的虚拟完成时间戳 $\mathrm{OF}_{ij}^{\mathrm{HOL}}$和虚拟开始时间戳 $\mathrm{OS}_{ij}^{\mathrm{HOL}}$为:

$$\mathrm{OF}_{ij}^{\mathrm{HOL}} = \mathrm{IF}_{ij}^{\mathrm{HOL}} + \frac{(N-1)L_{\max}}{R}, \mathrm{OS}_{ij}^{\mathrm{HOL}} = \mathrm{OF}_{ij}^{\mathrm{HOL}} - \frac{l_{ij}^{\mathrm{HOL}}}{r_{ij}} \tag{4}$$

其中,$\mathrm{IF}_{ij}^{\mathrm{HOL}}$为包携带的输入虚拟完成时间戳,$r_{ij}$和 R 为包 p_{ij}^{HOL}所在集合 F_{ij}和输入端口速率,l_{ij}^{HOL}为 p_{ij}^{HOL}的长度. TOS_j 从 P_i^{HOL} 中选择最小时戳 $\mathrm{OF}_{ij}^{\mathrm{HOL}}$的包,也即

$$i: \min\{\mathrm{OF}_{ij}^{\mathrm{HOL}}\}$$

输出调度不再对进入交叉点队列的包细分为流,由于:(1) 在交叉点不可能为所有的流维护独立的缓存队列,无法实现基于流的调度;(2) 输入调度对流进行公平性约束,确保流进入交叉点的公平性. 用$\widetilde{\mathrm{OS}}_{ij}^{\mathrm{HOL}}$和$\widetilde{\mathrm{OF}}_{ij}^{\mathrm{HOL}}$表示输出调度包实际开始时间戳和结束时间戳,则

$$\widetilde{\mathrm{OF}}_{ij}^{\mathrm{HOL}} = \widetilde{\mathrm{OS}}_{ij}^{\mathrm{HOL}} + \frac{l_{ij}^{\mathrm{HOL}}}{R}$$

3 性能分析

本节从时延性能和公平性能对 LHFS 的性能进行理论分析,首先分析时延性能.

3.1 时延性能

记 LHFS 输入调度为 $\mho = \langle \mathrm{PS} - \mathrm{IG} \rangle$,其中 PS 和 IG 分别表示端口和集合调度策略,$\Omega = \langle \mathrm{IS} - \mathrm{OS} \rangle$为输入、输出调度策略对,若输入、输出都采用 GPS,则 $\Omega = \langle \mathrm{GPS} - \mathrm{GPS}$,简记为 $\Omega_{\mathrm{GPS,GPS}}$.

假设 f_{ijk}到达过程服从(σ_{ijk}, r_{ijk})漏桶约束,即$[t_1, t_2)$内 f_{ijk}的到达流量:

$$A_{ijk}(t_1, t_2) = \int_{t_1}^{t_2} r_{ijk}(t)\mathrm{d}t + \sigma_{ijk},$$

其中,σ_{ijk}为仅和流f_{ijk}相关的常量.

引理 1 PGS 为持续工作型(work conserving)调度策略.

证明 由于篇幅限制,这里不再给出证明,详细请参考文献[18].

引理 2 令$\widetilde{\mathrm{IF}}_{ij,\mathrm{PGS}}^{m}$和$\widetilde{\mathrm{IF}}_{ij,\mathrm{PGS}}^{n}$为独立 PGS 调度下 F_{ij}第

m 和 n 个包 p_{ij}^m 和 p_{ij}^n 离去时间,则 $\widetilde{\mathrm{IF}}_{ij,\mathrm{PGS}}^n \geqslant \widetilde{\mathrm{IF}}_{ij,\mathrm{PGS}}^m \Leftrightarrow V(\widetilde{\mathrm{IF}}_{ij,\mathrm{PGS}}^n) - V(\widetilde{\mathrm{IF}}_{ij,\mathrm{PGS}}^m) \geqslant \widetilde{\mathrm{IF}}_{ij,\mathrm{PGS}}^n - \widetilde{\mathrm{IF}}_{ij,\mathrm{PGS}}^m$.

证明　由于篇幅限制,这里不再给出证明,详细请参考文献[18].

事实上,文献[17]对模拟 GPS 的 PFQ 算法进行了研究,得出具有大于等于 1 的增长率的虚拟时间函数 $V(t)$是 PFQ 调度器获得低 GPS 相对时延保障的必要条件.

引理 3　对于流集合$\{f_k\}$,若采用 PGS 进行独立调度,则在 PGS 调度机制下 f_k 第 n 个包的离去时间与对应 GPS 系统中该包的离去时间满足如下关系:

$$\widetilde{F}_{k,\mathrm{PGS}}^n(t) - \widehat{F}_{k,\mathrm{GPS}}^n(t) \leqslant \frac{L_{\max}}{r},$$

且任一时刻 τ,在 PGS 和 GPS 下流 f_i 获得的服务量满足:

$$W_{k,\mathrm{PGS}}(0,\tau) - W_{k,\mathrm{GPS}}(0,\tau) \leqslant L_{\max},$$

其中,r 为流集合$\{f_k\}$的输出共享链路带宽.

证明　由于篇幅限制,这里不再给出证明,详细请参考文献[18].

定理 1　对于存在 N 个流集合 G_j 的集合 G,若集合间采用 PGS 调度机制,则集合 G_j 第 n 个包的离去时间与对应 GPS 系统中该集合第 n 个分组的离去时间满足:

$$\widetilde{F}_{j,\mathrm{GPS}}^n(t) - \widehat{F}_{j,\mathrm{GPS}}^n(t) \leqslant (N-1)\frac{L_{\max}}{r},$$

其中,r 为流集合 G 的输出共享链路带宽.

证明　该定理的证明基于引理(3),同文献[19]中的定理(4.3.1)和(4.3.2)的证明方法,采用了文[15]中定理(1)的证明原理.由于篇幅有限,这里不再赘述,具体参考文献[18].

定理 1 表明对于存在 N 流集合的集合 $\boldsymbol{G}$,集合 $\boldsymbol{G}_j$ 第 n 个包的 GPS 相对时延上限和集合数 N 成正比关系.虽然该时延上限同引理(3)相比较存在比例扩张关系,然而当 N 给定时,$(N-1)\frac{L_{\max}}{r}$为较小有限值,考察 $\mathfrak{V}$ =〈SDRR-PGS〉下包 GPS 相对时延上限.

推论 1　设 $\boldsymbol{F}_i$ =｛设 $\boldsymbol{F}_{ij}$:$1 \leqslant j \leqslant N$｝,若流 f_{ijk}服从(r_{ijk},σ_{ijk})漏桶约束,则 $\mathfrak{V}$ =〈SDRR-PGS〉调度下设 $\boldsymbol{F}_{ij}$第 n 个包的完成时间$\widetilde{F}_{ij,\mathfrak{v}}^n(t)$与 GPS 系统完成时间 $\widehat{F}_{ij,\mathrm{GPS}}^n(t)$满足:

$$\widetilde{F}_{ij,\mathfrak{v}}^n(t) - \widehat{F}_{ij,\mathrm{GPS}}^n(t) \leqslant L_{\max}\frac{N-1}{R}$$

其中,R 为端口 i 的汇聚输出速率.

定理 2　若交叉点队列$\{\mathrm{XB}_{ij}\}$在$[t_1,t_2)$内处以持续“积压”状态,那么对应于该队列的第 n 个包的实际服务结束时间$\widetilde{\mathrm{OF}}_{ij,\mathrm{TOS}}^n$和 GPS 下包的完成时间$\widehat{\mathrm{OF}}_{ij,\mathrm{GPS}}^n$满足:

$$\widetilde{\mathrm{OF}}_{ij,\mathrm{TOS}}^n - \widehat{\mathrm{OF}}_{ij,\mathrm{GPS}}^n \leqslant \frac{L_{\max}}{R}$$

证明　由于篇幅限制,这里不再给出证明,详细请参考文献[18].

推论 1 和定理 2 给出了输入调度和输出调度下包的 GPS 相对时延,下面考察调度策略 Ω_{LHFS} =〈$\mathfrak{V}$,TOS〉和 Ω_{GPS} =〈GPS,GPS〉下交换系统 GPS 相对时延.

引理 4　若 CICQ 交换系统采用 Ω_{LHFS}调度策略,则集合 $\boldsymbol{F}_{ij}$的第 n 个包 p_{ij}^n的实际离去时间$\widetilde{F}_{ij,\Omega_{\mathrm{LHFS}}}^n$和采用调度策略 Ω_{GPS}时该包的离去时间$\widehat{F}_{ij,\Omega_{\mathrm{GPS}}}^n$存在如下关系:

$$\widetilde{F}_{ij,\Omega_{\mathrm{LHFS}}}^n - \widehat{F}_{ij,\Omega_{\mathrm{GPS}}}^n \leqslant L_{\max}\frac{N}{R}$$

证明　由于篇幅限制,这里不再给出证明,详细请参考文献[18].

3.2　公平性能

引理 5　输入调度分别采用 $\mathfrak{V}$ =〈SDRR-PGS〉和 $\mathfrak{V}'$ =〈SDRR-GPS〉调度机制时集合 $\boldsymbol{F}_{ij}$第 n 个分组实际完成时间$\widetilde{\mathrm{IF}}_{ij,\mathfrak{v}}^n$及$\widehat{\mathrm{IF}}_{ij,\mathfrak{v}'}^n$满足:

$$\widetilde{\mathrm{IF}}_{ij,\mathfrak{v}}^n - \widehat{\mathrm{IF}}_{ij,\mathfrak{v}'}^n \leqslant \frac{L_{\max}}{R}.$$

推论 2　CICQ 系统分别采用 Ω_{LHFS} =〈$\mathfrak{V}$,TOS〉和 Ω' =〈$\mathfrak{V}'$,GPS〉调度机制时集合 F_{ij}第 n 个分组实际完成时间 $\widetilde{F}_{ij,\Omega_{\mathrm{LHFS}}}^n$和$\widehat{F}_{ij,\Omega'}^n$满足如下关系:

$$\widetilde{F}_{ij,\Omega_{\mathrm{LHFS}}}^n - \widehat{F}_{ij,\Omega'}^n \leqslant \frac{2L_{\max}}{R}.$$

引理 6　若输入调度采用 $\mathfrak{V}'$ =〈SDRR-GPS-GPS〉,则任意集合 $\boldsymbol{f}_{ij}$每一帧包的传输时间在至少需要$\frac{L_{\max}}{r_{\min}}$的时长,其中 $r_{\min} = \min_{f_{ijk} \in \boldsymbol{f}_{ij}}\{r_{ijk}\}$为集合 $\boldsymbol{f}_{ij}$中最小速率流.

引理 7　假设流 f_{ijk}在(t_1,t_2)时间内处于持续“积压”状态,令 U 为包含时间区间(t_1,t_2)的最小 SDRR 传输帧数,则在(t_1,t_2)内流 f_{ijk}接受的服务量 $W_{ijk,\mathrm{SDRR}}(t_1,t_2)$满足:

$$(U-4)\zeta_{ijk} \leqslant W_{ijk,\mathrm{SDRR}}(t_1,t_2) \leqslant (U+2)\zeta_{ijk}.$$

证明　该引理可由 SDRR 调度过程直接得出,详细证明见文献[18].

定理 3　当采用 LHFS 调度策略时 CICQ 系统能够为流 f_{ijk}提供 WFI 公平性.

证明　由于篇幅限制,这里不再给出证明,详细请参考文献[18].

定理 4　当采用 LHFS 调度策略时 CICQ 系统能够为流 f_{ijk}提供 PFI 公平性.

证明　由于篇幅限制,这里不再给出证明,详细请

参考文献[18].

4 仿真实验

本节从时延抖动、带宽分配的公平性和吞吐量三个方面对 LHFS 算法的性能进行仿真评估.仿真环境配置如下:交换结构的规模为 16 × 16;输入、输出端口的带宽归一化为 1;由于 LHFS 为变长分组交换,仿真中产生的分组长度在[40,1500]内服从均匀分布,单位为字节(bytes);流量到达模型采用马尔科夫调制的泊松过程;流量分布采用均匀和非均匀两种业务流分布模型,用 λ_{ij}表示到达输入 i、去往输出 j 的速率,则对于均匀分布:$\lambda_{ij}=\frac{R}{N}$;对于非均匀分布:

$$\lambda_{ij}=\begin{cases}R\left(w+\dfrac{1-w}{N}\right), & \text{if} \quad i=j\\ R\dfrac{1-w}{N}, & \text{if} \quad i\neq j\end{cases}$$

其中,w 为不均衡指数,显然 w 越大,流量分布越集中.由于 LHFS 为基于流的调度机制,对于任意输入端口 i 和输出端口 j 产生两条流 f_{ij1}和 f_{ij2},且 $\lambda_{ij1}=2\lambda_{ij2}$.

4.1 GPS 相对时延

采用均匀和非均匀两种流量分布模型评估 LHFS 的 GPS 相对时延性能.为了便于说明问题,这里仅以流 f_{112}为例分析分组的 GPS 相对时延性能.

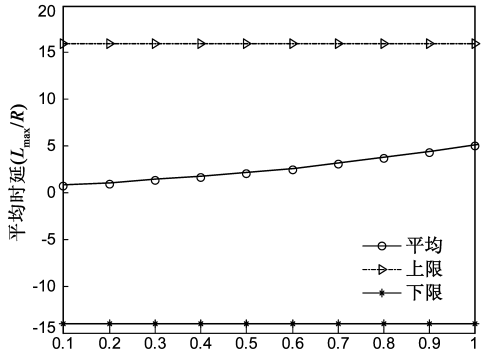
图 2(*a*)和(*b*)分别给出了均匀和非均匀业务到达下 LHFS 算法的 GPS 平均相对时延(ΔD_{av})性能随流量强度和分布的变化曲线.为了便于说明问题,同时给出了理论上 GPS 相对时延的上下限.可以看出,在均匀流量到达下,ΔD_{av}随着流量强度的增长而增长.在非均匀流量到达下,相对时延 ΔD_{av}随着不均衡指数 w 的不断增加而减小,并逐渐趋近于"0",并在 $w=0$ 时取得了极大值.

4.2 公平性能保障

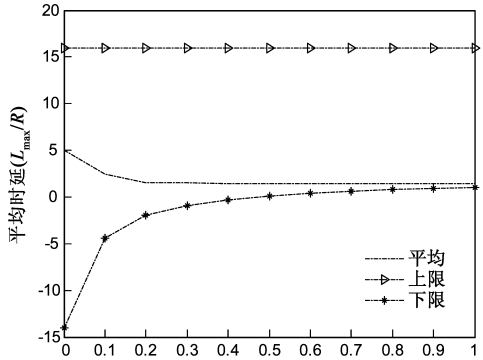
采用均匀和非均匀两种流量分布模型评估 LHFS 带宽分配的公平性.带宽公平性通过各条流 f_{ijk}的预约带宽和获得的实际带宽之间相对比率 ξ 衡量:

$$\xi_i=\frac{W_{ijk}/r_{ijk}}{W_{ijk'}/r_{ijk'}}$$

图 3(*a*)和 3(*b*)给出了均匀和非均匀业务分布下 LHFS 算法的公平性能仿真结果.在均匀分布下,流 f_{111}和 f_{112}获的带宽随到达强度的增大而增大,然而 ξ_1 始终约为 1;在非均匀分布下,流 f_{111}、f_{112}、f_{211}和 f_{212}获得带宽随 w 变化而变化,然而相对比率的值都在 1.0 附近波动.

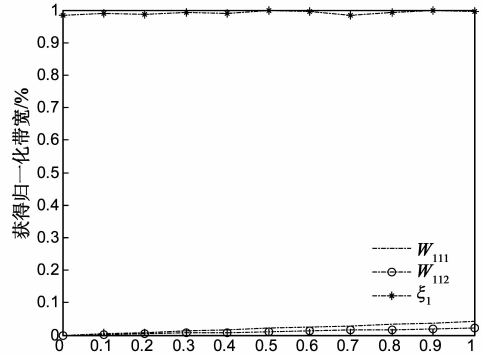


(*a*) 均匀

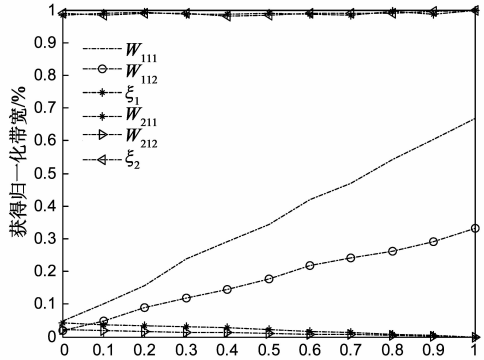


(*b*) 非均匀

图2 GPS相对性能仿真结果



(*a*) 均匀



(*b*) 非均匀

图3 公平性能仿真结果

4.3 吞吐量性能

吞吐量性能评估采用非均匀业务流分布模型,业务源到达分布具体产生过程详见本节开始.由于交换系统各输入端口具有类似的统计行为,这里在任意输入端口 j 为例统计交换系统的吞吐量性能.用 λ_{max} 表示交换系统稳定条件下任意输入端口 i 容许的最大业务流到达速率,则交换系统的吞吐量 $T=\lambda_{max}$.

图 4 给出了非均匀分布下 LHFS 算法的吞吐量性能仿真结果.可以看出,当不均衡指数 w 从 0.1 增长到 1.0 过程中,LHFS 算法的吞吐量均接近理论上限值 1.0.

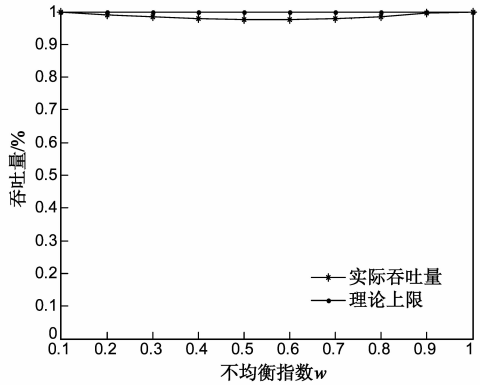


图4　吞吐量性能仿真结果

5 结束语

本文提出了一种能够为流的提供公平服务的混合调度策略—LHFS.同现有调度策略不同,LHFS 采用分层混合调度机制以获得 $O(1)$ 的时间复杂度,且每个输入、输出可独立地变长分组交换,无需加速便能为流提供时延上限和公平性保障.最后,基于 SPES 对 LHFS 的性能进行了仿真评估.依托于这一思想,后期将在支持区分服务和组播交换方面进行更深入研究.

参考文献

[1] Nabeshima M. Performance evaluation of a combined input- and crosspoint-queued switch [J]. IEICE Trans on Commun, 2000, E83-B(3): 737 – 741.

[2] Rojas-Cessa R, Oki E, Jing Z, Chao JH. On the combined input-crosspoint buffered switch with round-robin arbitration [J]. IEEE Trans on Commun, 2005, 53(11): 945 – 1951.

[3] Mhamdi L, Hamdi M. MCBF: A high-performance scheduling algorithm for buffered crossbar switches [J]. IEEE Commun Letters, 2003, 7(9): 451 – 453.

[4] Zhang X, Bhuyan LN. An efficient algorithm for combined input-crosspoint-queued (CICQ) switches [A]. Shah R. Proc EEE Globecom [C]. Dallas: IEEE Communications Society, 2004. 1168 – 1173.

[5] Javidi T, Magill R, Hrabik T. A high-throughput scheduling algorithm for a buffered crossbar switch fabric [A]. Navid Lashkarian. Proc IEEE ICC 2011 [C]. Helsinki: IEEE Communications Society, 2001. 1586 – 1591.

[6] Deng Pan, Y Y Yang. Localized independent packet scheduling for buffered crossbar switches [J]. IEEE Trans on Computers, 2009, 58(2): 260 – 274.

[7] Cheng-Shang Chang, Yu-Hao Hsu, Jay Cheng, Duan-shin Lee. A dynamic frame sizing algorithm for CICQ switches with 100% throughput [A]. Jim Kurose. Proc IEEE INFOCOM 2009 [C]. Rio de Janeiro: IEEE Communications Society, 2009. 738 – 746.

[8] Y Shen, S S Panwar, and H J Chao. SQUID: A practical 100% throughput scheduler for crosspoint buffered switches [J]. IEEE/ACM Trans on Networking, 2010, 18(4): 1119 – 1131.

[9] Magill B, Rohrs C, Stevenson R. Output-queued switch emulation by fabrics with limited memory [J]. IEEE JSAC, 2003, 21(4): 606 – 615.

[10] Chuang S-T, Iyer S, McKeown N. Practical algorithms for performance guarantees in buffered crossbars [A]. Taieb Znati. Proc IEEE INFOCOM 2005 [C]. Miami: IEEE Communications Society, 2005. 981 – 991.

[11] J Turner. Strong performance guarantees for asynchronous crossbar schedulers [J]. IEEE/ACM Trans on Networking, 2009, 17(4): 1017 – 1028.

[12] Xiao Zhang, Satya R Mohanty, Laxmi N Bhuyan. Adaptive max-min fair scheduling in buffered crossbar switches without speedup [A]. Robert L Baldwin. Proc IEEE INFOCOM 2007 [C]. Anchorage: IEEE Communications Society, 2007. 454 – 462.

(下转第 733 页)

作者简介

扈红超　男,1982 年 3 月出生于河南省商丘市.现为国家数字交换系统工程技术研究中心讲师.在国内外发表学术论文 30 余篇.
E-mail: huhongchao@gmail.com

郭云飞　男,1963 年 10 月出生于河南省郑州市.现为国家数字交换系统工程技术研究中心(NDSC)副主任,教授、博士生导师.
E-mail: gyf@ndsc.com.cn

band users[J]. Radio Science, 1997, 32(5): 2037 – 2045.
[10] 李雪,冯静,邓维波,等.返回散射电离图智能判读[J].电波科学学报,2010,25(2):160 – 162.
Li X, Feng J, Deng W B, et al. New order-select method of polynomial modeling for ionosphere phase perturbation correction[J]. Chinese Journal of Radio Science, 2010, 25(2): 160 – 162. (in Chinese)
[11] 孙广俊,齐东玉,李铁成.利用返回散射系统监测海洋回波[J].电子学报,2005,44(7):1334 – 1337.
Sun G J, Qi D Y, Li T C. Sea echo detection with the system of ionospheric backscatter sounding[J]. Acta Electronica Sinica, 2005, 44(7): 1334 – 1337. (in Chinese)
[12] 于洋.实时选频技术及其在短波自适应系统中的应用[J].电子工程师,1999,25(10):17 – 18.

作者简介

王俊江　男,1978 出生于河南,中国电波传播研究所高级工程师,研究方向为电离层波传播及其应用、数值计算等.
E-mail: wangjj61@163.com

柳　文　男,1973 年出生于湖南,中国电波传播研究所高级工程师,博士.研究方向为电离层物理及电离层波传播及其应用等.

焦培南　男,1939 年出生于广东,中国电波传播研究所研究员,博士生导师,国家有突出贡献专家.1962 年毕业于武汉大学物理系.获国家科技进步二等奖两项、三等奖一项、省部科技进步奖十一项,发表论文 110 多篇.研究方向为 HF 天波超视距雷达、电离层及其电波传播、特殊介质的波传播和散射等.

(上接第 723 页)

[13] Si-Min He, Shu-Tao Sun, Hong-Tao Guan, Qiang Zheng, You-Jian Zhao, Wen Gao. On guaranteed smooth switching for-buffered crossbar switches [J]. IEEE/ACM Trans on Networking, 2008, 16(3): 718 – 731.
[14] D Pan, Zhenyu Yang, Kia Makki, Niki Pissinou. Providing performance guarantees for buffered crossbar switches without speedup [A]. Ozgur Akan. Proc ICST QShine [C]. Berlin: Springer, 2009. 297 – 314.
[15] Parekh, A Gallager, R. A generalized processor sharing approach to flow control in integrated services networks: The single node case [J]. IEEE/ACM Trans on Networking, 1993, 1(3): 344 – 357.
[16] Douglas Comer, Maxim Martynov. Design and analysis of hybrid packet schedulers [A]. Douglas Merrill. Proc IEEE INFOCOM 2008 [C]. New York: IEEE Communications Society, 2008. 1570 – 1578.
[17] J Bennett, H Zhang. Worst-case Fair Packet Fair Queueing Algorithms [R]. Pittsburgh: Carnegie Mellon University, 1996.
[18] Hongchao Hu, Yunfei Guo, Peng Yi. Design and Implementation of Fair Scheduling Algorithms in Combined Input and Crosspoint Queued Switches [R]. Zhengzhou: National Digital Switching System Engineering & Technological R&D Center, 2010.
[19] Maxim Martynov. Design and Implementation of Hybrid Packet Scheduling Algorithms for High Speed Networks [D]. West Lafayette: Purdue University, 2007.