MUMF-支持单多播公平服务的调度策略

扈红超，郭云飞，陈庶樵，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：基于联合输入交叉点排队 (CICQ, combined input and cross-point queuing) 交换结构探讨了单多播混合调度的公平性问题，提出了能够为单多播业务提供混合公平性的 CICQ 理想调度模型。基于理想调度模型，提出了逼近理想调度模型的 MUMF(mixed uni- and multicast fair)调度算法，MUMF 调度算法采用了分级和层次化的公平调度机制，通过输入调度和交叉点调度确保单多播业务混合调度的公平性。MUMF 交换机制的每个输入、输出端口可独立地进行分组交换，具有良好可扩展特性。最后，基于 SPES(switching performance evaluation system)的性能仿真结果表明 MUMF 调度算法具有良好的时延、公平性和吞吐量性能。

关键词：带缓存交叉开关；多播；公平服务；混合调度
中图分类号：TP393 **文献标识码**：A **文章编号**：1000-436X(2012)01-0053-11

MUMF: a mixed uni-and multicast fair scheduling scheme for CICQ switches

HU Hong-chao, GUO Yun-fei, CHEN Shu-qiao, YI Peng
(National Digital Switching System Engineering & Technological R&D Center, Zhengzhou 450002, China)

Abstract: Based on the CICQ (combined input and cross-point buffered) switches, how to support fair scheduling for uni- and multicast traffic was discussed thoroughly. Then, it comed up with an ideal theoretical switching model for CICQ switches. Based on the theoretical model, a mixed uni- and multicast fair scheduling (MUMF) scheme was proposed. With MUMF, each input port and output port can schedule variable length packets independently. Simulation results based on SPES (switch performance evaluation system) show that MUMF can provide good delay, fair and throughput performance.

Key words: buffered crossbar; multicast; fair service; mixed scheduling

1 引言

以应用和服务为特征的网络承载网的发展使得网络业务呈现多元化的发展趋势，视频点播、VoIP 等为特征的多媒体及相关技术(如 P2P)成为了下一代互联网(NGI, next generation internet)和三网融合下网络业务的典型特征之一。多媒体业务的重要特征是具有多播性，相对于单播业务<源 *s*-目的 *d*>一对一的传输，多播业务存在多个目的 $D=\{d_i\}_{i\geqslant 1}$，这一特征要求网络核心交换节点能够支持多播交换，从而提高网络传输效率和吞吐量[1,2]。就现有交换机制而言，输出排队(OQ, output queuing)交换结构交换和存储单元都需工作于 N 倍线路速率，构建大容量交换系统时不具

收稿日期：2010-09-01；**修回日期**：2011-01-20
基金项目：国家重点基础研究发展计划(“973”计划)基金资助项目(2007CB307102)；国家高技术研究发展计划(“863”计划)基金资助项目(2009AA01A346, 2008AA01Z214, 2008AA01A323)
Foundation Items: The National Basic Research Program of China(973 Program) (2007CB307102);The National High Technology Research and Development Program of China(863 Program)(2009AA01A346, 2008AA01Z214, 2008AA01A323)

备良好的扩展性；输入排队(IQ, input queuing)交换结构需要集中式的调度[3]，构建大容量交换系统时存在瓶颈。近年来随着微电子技术的进步，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[4]。CICQ 通过在每个交叉点集成了一定的缓存将 $N\times N$ 的交换结构分割为 N 个 $N\times 1$ 和 N 个 $1\times N$ 的子结构，并将输入、输出的带宽冲突隔离开来，使分布式调度成为了现实。目前基于 CICQ 构建的调度机制集中在提供高吞吐量[5~11]、模拟 OQ[12~14]和性能保障[15~17]和多播交换[18~20]方面。

在 CICQ 交换结构的多播研究方面，文献[18]分析了 $(K,N_a)-$complex 多播流量模型下 CICQ 的吞吐量，指出当加速比 S 有限时吞吐量随交换结构规模 $N\to\infty$ 大幅下降；在均匀“可允许”到达下，当 $S>1$ 和交叉点缓存容量 $B\geqslant\lfloor k/(S+1)\rfloor+1$ 时交换系统是稳定的，为更好地理解 CICQ 多播交换机制奠定了基础。单多播轮询调度 (MURS, multicast and unicast round robin scheduling) 策略[20]根据轮询优先级的不同可细分为单播优先(MURS_uf)、多播优先(MURS_mf)和混合轮询(MURS_mix)算法。同 IQ 结构相比较，MURS 大幅提升了交换系统的时延和吞吐量性能。基于输出的共享交叉点缓存 O-SMCB (output-based shared-memory cross-point- buffered)调度机制[21]，在“可允许”均匀和“对角”多播流量到达下，O-SMCB 可在节省 50%的缓存条件下提供接近 100%的吞吐量性能。

当前基于 CICQ 构建的多播调度策略主要集中在如何提高系统的吞吐量和时延性能，尚无工作着眼于单多播混合调度的公平性问题。本文在这一方面进行了探索，提出支持单多播混合公平性的理想调度——(MUMF, mixed uni-and multicast fair scheduling) 模型，MUMF 调度算法采用基于时间戳(timestamp)的调度机制，通过输入调度和交叉点调度确保单多播流的公平性。针对其复杂度过高，提出了改进调度算法 MUMF。MUMF 调度算法每个输入、输出可独立进行变长分组交换，无需加速便可为流提供时延和公平性保障。本文剩余章节安排如下：第 2 部分是相关工作；第 3 部分详细阐述了 MUMF 和 MUMF 交换机制；第 4 部分对 MUMF 调度算法的性能进行仿真评估；第 5 部分是结束语。

2 相关工作

现有公平服务调度策略的研究主要集中在如何在共享输出链路的单播竞争流之间提供公平服务，具体而言，若 K 条流 $\boldsymbol{F}=\{f_1,f_2,\cdots,f_K\}$ 共享输出带宽为 R 的链路 L，流 f_k 的预约带宽为 r_k，且 $\sum_{k=1}^{K}r_k\leqslant R$,调度器 s 根据 r_k 为每条流 f_k 计算权重 w_k 表征其归一化需求带宽，即 $w_k=\dfrac{r_k}{R}$。设 $W_{k,s}(t_1,t_2)$ 为调度器 s 下 $[t_1,t_2)$ 间隔内流 f_k 获得的服务量，理想的公平服务调度器 s 满足 $\dfrac{W_{k,s}(t_1,t_2)}{r_k}=\dfrac{W_{g,s}(t_1,t_2)}{r_g}$。然而，受硬件处理器处理机制和分组处理系统特性的限制，实际调度系统无法达到这一目标。最坏服务公平指数(WFI, worst-case fairness index)[22]和比例服务公平指数(PFI, proportional fairness index)[23]是衡量实际交换系统调度器公平性的重要指标。用 f_{ik} 表示到达交换系统输入端口 i 的第 k 条多播流，目的输出端口集合为 o_{ik}，预约带宽为 r_{ik}，$W_{ikj,s}(t_1,t_2)$ 表示 $[t_1,t_2)$ 内调度策略 s 下输出端口 $j\in o_{ik}$ 为流 f_{ik} 提供的服务量，理想公平多播调度器满足 $\dfrac{W_{i_1k_1j,s}(t_1,t_2)}{r_{i_1k_2}}=\dfrac{W_{i_2k_2j,s}(t_1,t_2)}{r_{i_1k_2}}$。若 $o_{i_1k_1}\bigcap o_{i_2k_2}\neq\varnothing$，且

$$\left|\frac{W_{i_1k_1j,s}(t_1,t_2)}{r_{i_1k_2}}-\frac{W_{i_2k_2j,s}(t_1,t_2)}{r_{i_1k_2}}\right|\leqslant c_{i_1k_1,i_2k_2}\tag{1}$$

则称交换系统在调度策略 s 下能够为多播流提供 PFI 公平性。

由于受 Crossbar 输入和输出带宽竞争的双重约束，基于共享链路的调度策略无法直接应用到基于虚拟输出排队的 IQ 或者 CICQ 交换结构中。MFS(multicast fair scheduling)[24]为每个输入端口分配一份额计数器 c_i 记录该端口获得的调度份额，并为每个分组分配一到达时间戳，同时分两级对多播进行调度：首先输入端口选择具有最小虚拟时间戳的流 $k:\min V_{ik}(t)$ 作为该端口的候选流，其中，$V_{ik}(t)$ 为端口 i 第 k 条流的虚拟时间；其次，选择输入端口 $i:\min\dfrac{c_i}{\sum_k B_{ik}}$ 进行调度，其中，B_{ik} 为端口 i 第 k 条流的预约带宽。MFS 能够在高负载下提供良好的公平性。CMF(credit based multicast fair schedul-

ing)[25]依据预约带宽 $r_{ij}(t)$ 为每个输入、输出对计算一可用份额 $c_{ij}(t)$ 和累计调度差额 $A_{ij}(t)$。在 Request 阶段选择 $A_{ij}(t)>0$ 和具有最小到达时间的分组，而 Grant 阶段选择具有最大 $A_{ij}(t)$ 的分组。CMF 将多播分组复制为单播分组并采用逼近 GPS 的调度策略，获得了较低的相对时延和公平性。

由于 IQ 结构集中式的调度机制无法构建大规模交换系统，导致 MFS 和 CMF 在实际应用中受到限制。此外，CICQ 结构还受到交叉点队列的竞争约束，因而 MFS 和 CMF 也无法直接应用到 CICQ 结构中，然而它们为 CICQ 混合公平调度策略的设计提供了重要思路。另外，虽然针对单播调度公平性的研究成果相对较丰，然而多播分组具有多个可选目的输出端口，必须采用适用于多播的公平调度策略和公平性评价基准，且采用扇出分割和不分隔的调度策略对性能有较大影响，因而适用于单播的公平调度策略也无法直接应用到混合调度中。下面首先介绍 MUMF 调度策略。

3　MUMF 混合调度策略

3.1　交换系统模型

图 1 给出了 $N\times N$ 规模的支持单多播混合调度的 CICQ 交换系统总体结构。为解决分组的队头阻塞(HOLB, head of line blocking)问题，采用虚拟输出队列(VOQ, virtual output queuing)排队机制，为单/多播分组分别维护了独立的单/多播虚拟输出队列 $\{UVQ_{ij}\}$ 和 $\{MVQ_{im}\}$。对于单播分组，采用 N 个虚拟输出队列可完全避免队头阻塞；对于多播分组，理论上需要在每个输入端口 i 维护 2^N-N-1 个 MVQ_{im} 队列。为此，必须通过设计有效的分组入队机制(QM, queuing mechanism)降低多复制端口的队头阻塞，在每个输入端口维护 $M(M<2^N-N-1)$ 个 MVQ_{im} 队列；经输入复制后进入 Crossbar 的多播分组仅存在单一输出端口，因而仅需在每个交叉点维护一交叉点单/多播虚拟输出队列。为防止 $UXB_{ij}(MXB_{ij})$ 溢出，为每个 $UXB_{ij}(MXB_{ij})$ 维护一流控状态信号 $s_{ij}^{\mathrm{u}}(s_{ij}^{\mathrm{m}})$。假定到达各输入的分组定长，标记为 cell。以线路速率 R 传输一个 cell 的时间称为一个时隙(slot)。

3.2　理想调度模型

要实现单多播混合调度的短时公平性必须为输入端口 i 的所有多播流 $\left\{f_{ik}^{\mathrm{m}}\right\}$ 分别维护独立的虚拟多播输出队列并采用变长分组交换机制以逼近 GPS 系统。参考图 1，则此时多播缓存队列数目 $M=K$，其中，K 为端口 i 多播流数，因而队列缓存维护复杂度为 $O(K)$。最新研究成果表明，虽然 internet 骨干网络同时存在的流可达几十万甚至上百万，然而同时处于活动状态的流仅为上万条[26]。用 r_{ik}^{m} 表示到达输入端口 i 的第 k 条多播流的预约带宽，则去往输出 j 的多播流的汇聚预约带宽和输入

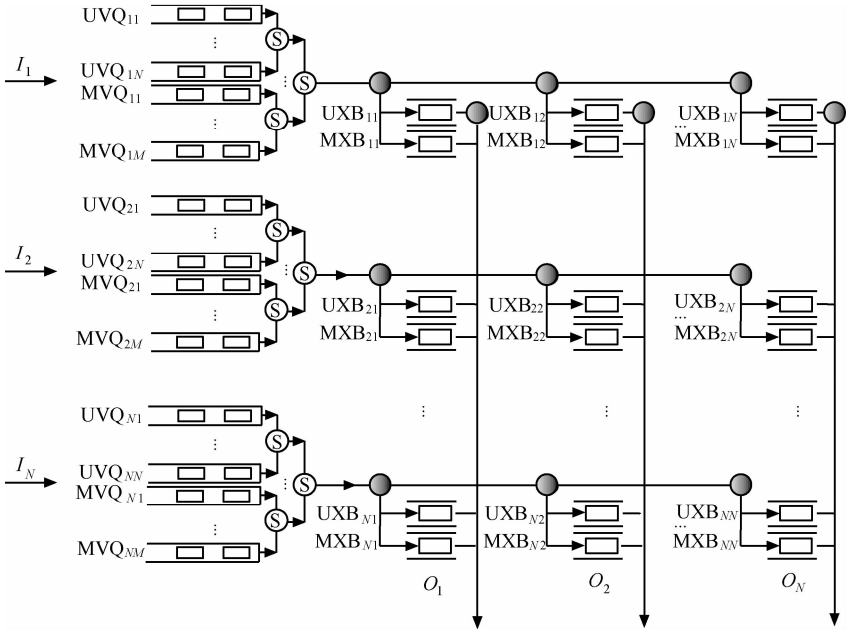


图 1　MUMF 交换系统总体结构

端口 i 的多播流汇聚预约带宽分别为 $r_{ij}^{\mathrm{m}}(t)=\sum_{k:j\in o_{ik}} r_{ik}^{\mathrm{m}}(t)$ 和 $r_i^{\mathrm{m}}(t)=\sum_j\sum_{k:j\in o_{ik}} r_{ik}^{\mathrm{m}}(t)$，去往输出端口 j 的单播和多播业务流的累积预约带宽为 $r_{ij}(t)=r_{ij}^{\mathrm{u}}(t)+r_{ij}^{\mathrm{m}}(t)$。对累积预约带宽 $r_{ij}(t)$ 进行归一化可得

$$\omega_{ij}(t)=\frac{r_{ij}(t)}{\sum_{j=1}^{N} r_{ij}(t)}=\frac{r_{ij}(t)}{\sum_{j=1}^{N}\left(\sum_{k:j\in o_{ik}} r_{ik}^{m}(t)+r_{ij}^{u}(t)\right)}$$

且 $\omega_{ij}(t)$ 满足

$$\sum_{i=1}^{N}\omega_{ij}(t)\leqslant 1,\sum_{j=1}^{N}\omega_{ij}(t)\leqslant 1 \tag{2}$$

为了便于描述，首先给出本文用到的以下相关术语和缩略语，如表 1 所示。

表 1　　相关符号定义

符号	意义
R, N, M	链路带宽、交换结构规模和多播队列数
$\mathrm{IS}_i, \mathrm{CS}_j$	对应输入 i 和输出 j 的端口调度器
UVQ_{ij}	缓存到达输入 i、去往输出 j 的单播队列
MVQ_{im}	缓存到达输入端口 i 多播流的第 m 个队列
UXB_{ij}	缓存到达输入 i、去往输出 j 的交叉点单播队列
MXB_{ij}	缓存到达输入 i、去往输出 j 的交叉点多播队列
$r_{ij}^{\mathrm{u}}(t)$	时刻 t 到达输入 i、去往输出 j 的单播预约带宽
$w_{ij}^{\mathrm{u}}(t)$	时刻 t 到达输入 i、去往输出 j 的单播归一化带宽
$r_{ik}^{\mathrm{m}}(t)$	时刻 t 到达输入 i 的第 k 条多播流的预约带宽
$r_{ij}^{\mathrm{m}}(t)$	时刻 t 到达输入 i、去往输出 j 的多播预约带宽
$w_{ij}^{\mathrm{m}}(t)$	时刻 t 到达输入 i、去往输出 j 的多播归一化带宽
$q_{ij}^{\mathrm{u}}(t)$	时刻 t 虚拟输出队列 UVQ_{ij} 的长度
$q_{im}^{\mathrm{m}}(t)$	时刻 t 虚拟输出队列 MVQ_{im} 的长度
f_{ij}, VQ_{ij}	汇聚流 $f_{ij}=f_{ij}^{u}\cup\{f_{ik}^{m}: j\in o_{ik}\}$ 及其虚拟队列
$q_{ij}(t)$	时刻 t 虚拟队列 VQ_{ij} 的长度
IS_{ij}^{HOL}	汇聚流 f_{ij} 队首分组的虚拟开始时间
IF_{ij}^{HOL}	汇聚流 f_{ij} 队首分组的虚拟结束时间
IS_{ijh}^{HOL}	汇聚流 f_{ij} 中第 h 条流队首分组的虚拟开始时间
IF_{ijh}^{HOL}	汇聚流 f_{ij} 中第 h 条流队首分组的虚拟结束时间

3.2.1 输入调度模型

输入调度从虚拟输出队列集合 $\boldsymbol{VQ}_i=\{UVQ_{ij}\}_{1\leqslant j\leqslant N}\cup\{MVQ_{im}\}_{1\leqslant m\leqslant M}$ 中选择一单播队列 UVQ_{ij} 或者多播队列 MVQ_{im}。由于各输入端口具有相同的调度行为，以任意输入 i 为例描述输入调度过程，记 i 端口调度器为 IS_i，IS_i 采用基于时间戳的调度策略确保去往不同输出端口流的公平性。定义去往输出 j 的汇聚流为 $f_{ij}=\{f_{ij}^{\mathrm{u}}\}\cup\{f_{ik}^{\mathrm{m}}\}_{j\in o_{ik}}=\{f_{ijh}: f_{ij1}=f_{ij}^{\mathrm{u}}, f_{ijh}=f_{ik:j\in o_{ik}}^{\mathrm{m}}\}$，$IS_i$ 为 f_{ij} 队头分组 p_{ij}^{HOL} 计算一虚拟开始时间戳 IS_{ij}^{HOL} 和虚拟完成时间戳 IF_{ij}^{HOL}。由于分组的到达/离去会导致流的短时“积压”或“空闲”，引起流所在汇聚流需求带宽的变化，进而引发其他汇聚流分组完成时间的变化，导致分组虚拟开始和结束时间和将来分组的到达相关，因而必须对分组的到达/离去进行有效地跟踪。

定义 1 积压队列(backlogged queue)称时刻 t 队列 $UVQ_{ij}(MVQ_{im})$ 为积压队列，若 $q_{ij}^{\mathrm{u}}(t)\ (q_{im}^{\mathrm{m}}(t))>0$，并称 t 时刻 $f_{ij}^{\mathrm{u}}\ (f_{ik=m}^{\mathrm{m}})$ 为积压流，记 t 时刻积压流集合为 $\boldsymbol{B}_{ij}(t)$。

定义 2 事件(event)称分组到达/离去调度器 s 为一次事件，用 e 表示，第 n 次事件 e_n 发生的时刻用 t_n 表示，可见在 $[t_n,t_{n+1})$ 的时间间隔内汇聚流 f_{ij} 的积压流 $\boldsymbol{B}_{ij}(t)$ 不变化。

令 $\omega_i(t)=\sum_j\omega_{ij}(t)$ 为 t 时刻为输入端口 i 归一化汇聚带宽，$r_{ij}(t)=r_{ij}^{\mathrm{u}}(t)+r_{ij}^{\mathrm{m}}(t)$ 为去往 j 端口带宽需求，$[t_e,t_{e+1})$ 为 IS_i 的一个“忙”期，IS_i 虚拟时间函数定义为

$$V(t)=\begin{cases}0, & t=0\\ V(t_n)+\tau, & w_i(t)=0\\ \max\left(V(t_n)+\dfrac{\tau}{w_i(t)}, \min_j IS_{ij}^{\mathrm{HOL}}\right), & w_i(t)>0\end{cases} \tag{3}$$

其中，$\tau=t-t_m$。记 p_{ij}^{HOL} 为时刻 t 汇聚流 f_{ij} 的队首分组，其长度为 l_{ij}^{HOL}，则 p_{ij}^{HOL} 的虚拟开始和完成时间戳 IS_{ij}^{HOL} 和 IF_{ij}^{HOL} 的计算如下：

$$IS_{ij}^{\mathrm{HOL}}(t)=\begin{cases}\max\{IF_{ij}^{\mathrm{HOL}}(t), V(A_{ij}^{\mathrm{HOL}})\}, & q_{ij}(A_{ij}^{\mathrm{HOL}-})=0\\ IF_{ij}^{\mathrm{HOL}}(t), & q_{ij}(A_{ij}^{\mathrm{HOL}-})\neq 0\end{cases}$$

$$IF_{ij}^{\mathrm{HOL}}(t)=IS_{ij}^{\mathrm{HOL}}(t)+\frac{l_{ij}^{\mathrm{HOL}}}{r_{ij}(t)} \tag{4}$$

其中，$q_{ij}(A_{ij}^{\mathrm{HOL}-})$ 表示 A_{ij}^{HOL} 前一时刻 VQ_{ij} 的长度。每个 VQ_{ij} 是一逻辑虚拟队列，仅存放 f_{ij} 队头分组的信息(如 l_{ij}^{HOL}、IS_{ij}^{HOL} 和 IF_{ij}^{HOL})，实际分组仍存储在 UVQ_{ij} 或 MVQ_{im} 中。每次速率改变时 IS_i 重新计算虚

拟时间$V(t)$，该复杂度为$O(e)$；计算队头分组的虚拟开始和完成时戳$IS_{ij}^{\mathrm{HOL}}(t)$和$IF_{ij}^{\mathrm{HOL}}(t)$的复杂度为$O(N)$。记$\{f_{ij}:1\leqslant j\leqslant N\}$中各队头分组集合为$\boldsymbol{P}_i^{\mathrm{HOL}}=\{p_{ij}^{\mathrm{HOL}}:1\leqslant j\leqslant N\}$，$IS_i$选择$\boldsymbol{P}_i^{\mathrm{HOL}}$中虚拟开始时间$IS_{ij}^{\mathrm{HOL}}(t)$不大于$V(t)$且具有最小$IF_{ij}^{\mathrm{HOL}}(t)$的分组所在的汇聚流，即

$$j:\min_j\left\{IF_{ij}^{\mathrm{HOL}}(t)\right\}\ s.t.\ IS_{ij}^{\mathrm{HOL}}(t)\leqslant V(t) \tag{5}$$

可见，最坏情况下IS_i需从N个队头分组中选择一满足$IS_{ij}^{\mathrm{HOL}}(t)<V(t)$的分组，当交换结构的规模$N$给定时，其复杂度$O(\log N)$为较小常量值。当输出$j$确定后，$IS_i$再从流集合$f_{ij}$中选择一单播或多播流。记$\omega_{ij}^{\mathrm{u}}(t)$和$w_{ik}^{\mathrm{m}}(t)$为$t$时刻单/多播流的归一化带宽，到达输入$i$、去往输出$j$的流归一化汇聚带宽为$\omega_{ij}(t)=\omega_{ij}^{\mathrm{u}}(t)+\sum_{k:j\in o_{ik}}\omega_{ik}^{\mathrm{m}}(t)$，则单多播的虚拟时间$V'(t)$计算为

$$V'(t)=\begin{cases}0, & t=0\\ V'(t_n)+\tau, & w_{ij}(t)=0\\ \max\left(V'(t_n)+\dfrac{\tau}{w_{ij}(t)},\min_h IS_{ijh}^{\mathrm{HOL}}\right), & w_{ij}(t)>0\end{cases} \tag{6}$$

其中，$\tau=t-t_m$，IS_{ijh}^{HOL}为流$f_{ijh}\in f_{ij}$（f_{ijh}可为单播或者多播流)队首分组的虚拟开始时间。令t时刻f_{ijh}队头分组p_{ijh}^{HOL}的长度为l_{ijh}^{HOL}，则p_{ijh}^{HOL}虚拟开始和完成时间戳$IS_{ijh}^{\mathrm{HOL}}(t)$和$IF_{ijh}^{\mathrm{HOL}}(t)$计算如下：

$$IS_{ijh}^{\mathrm{HOL}}(t)=\begin{cases}\max\{IF_{ijh}^{\mathrm{HOL}}(t),V(A_{ijh}^{\mathrm{HOL}})\}, & q_{ijh}(A_{ijh}^{\mathrm{HOL}^-})=0\\ IF_{ijh}^{\mathrm{HOL}}(t), & q_{ijh}(A_{ijh}^{\mathrm{HOL}^-})\neq 0\end{cases}$$

$$IF_{ijh}^{\mathrm{HOL}}(t)=IS_{ijh}^{\mathrm{HOL}}(t)+\frac{l_{ijh}^{\mathrm{HOL}}}{r_{ijh}(t)} \tag{7}$$

其中，$q_{ijh}(t)$为时刻t流f_{ijh}对应虚拟输出队列VOQ_{ijh}的长度，$q_{ijh}(A_{ijh}^{\mathrm{HOL}^-})$表示在$A_{ijh}^{\mathrm{HOL}^-}$前一时刻$VOQ_{ijh}$的长度，$A_{ijh}^{\mathrm{HOL}}$为$p_{ijh}^{\mathrm{HOL}}$的到达时刻。$IS_i$从汇聚流$f_{ij}=\{f_{ijh}\}$中选择队头分组$IS_{ijh}^{\mathrm{HOL}}(t)$小于等于系统虚拟时间$V'(t)$且具有最小虚拟完成时间$IF_{ijh}^{\mathrm{HOL}}(t)$的分组所在那条流，即

$$h:\min_h\left\{IF_{ijh}^{\mathrm{HOL}}(t)\right\}\ \text{s.t.}\ IS_{ijh}^{\mathrm{HOL}}(t)\leqslant V'(t) \tag{8}$$

经IS_i选择后分组携带时间戳IF_{ijh}^{HOL}和IF_{ij}^{HOL}交叉点队列UXB_{ij}（MXB_{ij}）。可见，IS_i采用了层次化调度策略确保单多播混合调度的公平性和去往不同输出端口分组的公平性。

3.2.2　交叉点调度模型

交叉点调度策略从队列集合$\boldsymbol{XB}_j=\{UXB_{ij}\}_{1\leqslant j\leqslant N}\cup\{MXB_{ij}\}_{1\leqslant j\leqslant N}$中选择某一队列$UXB_{ij}$或者$MXB_{ij}$。同输入调度类似，以任意输出$j$为例描述交叉点调度器的调度过程。记输出$j$对应的交叉点调度器为$CS_j$。由于输入调度已经对单多播业务混合调度的公平性进行了约束，因而CS_j只需进一步强化公平性。记$\boldsymbol{XB}_j$各队列队头分组集合$\boldsymbol{P}_j^{\mathrm{HOL}}=\boldsymbol{P}_u^{\mathrm{HOL}}\cup\boldsymbol{P}_m^{\mathrm{HOL}}$，其中，$\boldsymbol{P}_u^{\mathrm{HOL}}=\{u_{ij}^{\mathrm{HOL}}\}_{1\leqslant j\leqslant N}$为单播交叉点队列队头分组集合，$\boldsymbol{P}_m^{\mathrm{HOL}}=\{m_{ij}^{\mathrm{HOL}}\}_{1\leqslant j\leqslant N}$为多播交叉点队列队头分组集合。

CS_j基于分组携带的时间戳IF_{ijh}^{HOL}和IF_{ij}^{HOL}选择队列，同输入调度策略类似，CS_j采用层次化的调度机制：首先从$\boldsymbol{P}_j^{\mathrm{HOL}}$中选择具有最小时间戳$IF_{ij}^{\mathrm{HOL}}$的那个分组，即

$$p_{ij}^{\mathrm{HOL}}:\min\left\{\left\{IF_{ij}^{\mathrm{HOL}}:u_{ij}^{\mathrm{HOL}}\in\boldsymbol{P}_u^{\mathrm{HOL}}\right\}\cup\left\{IF_{ij}^{\mathrm{HOL}}:m_{ij}^{\mathrm{HOL}}\in\boldsymbol{P}_m^{\mathrm{HOL}}\right\}\right\} \tag{9}$$

若选出的$p_{ij}^{\mathrm{HOL}}\in\boldsymbol{P}_u^{\mathrm{HOL}}$，则$CS_j$从$\boldsymbol{P}_u^{\mathrm{HOL}}$中选择具有最小时间戳$IF_{ijh}^{\mathrm{HOL}}$的单播分组作为调度结果；否则$CS_j$从$\boldsymbol{P}_m^{\mathrm{HOL}}$中选择具有最小时间戳$IF_{ijh}^{\mathrm{HOL}}$的多播分组，即

$$u_{ij}^{\mathrm{HOL}}:\min\left\{IF_{ijh}^{\mathrm{HOL}}:u_{ij}^{\mathrm{HOL}}\in\boldsymbol{P}_u^{\mathrm{HOL}}\right\} \tag{10}$$

或

$$m_{ij}^{\mathrm{HOL}}:\min\left\{IF_{ijh}^{\mathrm{HOL}}:m_{ij}^{\mathrm{HOL}}\in\boldsymbol{P}_m^{\mathrm{HOL}}\right\} \tag{11}$$

可以看出，CS_j基于时间戳IF_{ij}^{HOL}确保不同输出端口和混合调度的公平性，基于IF_{ijh}^{HOL}确保来自不同输入单多播业务的公平性。然而，IS_i和CS_j都需要为每个到达分组维护和计算开始和完成时间戳，理论模型实现上过于复杂，下面给出逼近理论模型的实际算法 MUMF。

3.3　MUMF 交换机制

MUMF 交换机制由分组入队机制(QM)、输入

端口调度(IS)器和交叉点调度(CS)器构成，其中 QM 调度机制将到达的多播分组根据目的端口进行映射和入队。下面详细描述各模块。

3.3.1 入队策略(QM)

用 $Q(\cdot)$ 表示分组入队策略，$Q(\cdot)$ 可描述为：给定分组 c 及其目的端口集合 $O_p=\{o_h^p\}_{1\leqslant h\leqslant U}$，确定 c 的入队队列 $\{UVQ_{ij}\}$ 或 $\{MVQ_{im}\}$，其中 $U=\left|O_p\right|$ 为目的端口数。单播分组 $(U=1)$ 由于采用虚拟输出队列可完全避免分组的队头阻塞，因而 $j=Q(O_p)=o_{h=1}^p$，即分组入队到 $UVQ_{io_1^p}$。对于多播分组，若分组序列 $\{p_0,p_1,\cdots,p_n,\cdots\}$ 的输出端口数为 $U(U\geqslant 2)$，则理论上需要在每个输入端口维护 C_N^U 个 MVQ_{im} 队列；而对于给定最大输出端口数 $U_{\max}$ 的分组序列 $\{p_0,p_1,\cdots,p_n,\cdots\}$，要完全避免队头阻塞问题，理论上需要在每个输入端口维护 $\sum_{u=2}^{U_{\max}}C_N^u$ 个 MVQ_{im} 队列，实现起来过于复杂。为此必须设计有效入队机制降低队头阻塞概率，本文采用最小覆盖调度(MCD, minimum cover dispatching)算法。

定义 3 覆盖(cover) 用 $\vec{I}=[1,1,\cdots,1]^{\mathrm{T}}$ 表示空间 R^N 的求和向量，称向量序列 $\{\vec{v}_1,\vec{v}_2,\cdots,\vec{v}_k\}$ 为 $\vec{I}=[1,1,\cdots,1]^{\mathrm{T}}$ 的一个覆盖，$\vec{v}_i=[x_0x_1\cdots x_j\cdots x_{N-1}]$，$x_j=0$ 或者 1，若满足：

$$\vec{I}=\vec{v}_1\oplus\vec{v}_2\oplus\cdots\oplus\vec{v}_k,$$

其中，“$\oplus$”为异或算子。可以看出，N 维求和向量是自身的一个覆盖。

定义 4 最小覆盖 $\{\vec{v}_1,\vec{v}_2,\cdots,\vec{v}_k\}$ 是 $\vec{I}$ 的覆盖，且 $\forall\vec{v}_i\neq\vec{v}_j,\vec{v}_i\odot\vec{v}_j=\vec{0}$，称 $\{\vec{v}_1,\vec{v}_2,\cdots,\vec{v}_k\}$ 为 $\vec{I}=[1,1,\cdots,1]^{\mathrm{T}}$ 的 k 阶最小覆盖，其中“$\odot$”为点积算子。向量 $\vec{v}_i$ 中“1”的个数称为 $\vec{v}_i$ 的度，用 $\left\|\vec{v}_i\right\|_1$ 表示；若 $\forall\vec{v}_i\neq\vec{v}_j,\left|\left\|\vec{v}_i\right\|_1-\left\|\vec{v}_j\right\|_1\right|\leqslant 1$，又称 $\{\vec{v}_1,\vec{v}_2,\cdots,\vec{v}_k\}$ 为 $\vec{I}$ 的最小均衡覆盖。

首先假设输入端口 i 维护的 MVQ_{im} 队列数目为 M, MCD 算法工作过程如下。

1) 首先根据定义 3 为队列集合 $\{MVQ_{im}\}$ 构造一个 M 阶 N 维最小均衡覆盖 $\{\vec{v}_1,\vec{v}_2,\cdots,\vec{v}_M\}$，并将每个 $\vec{v}_m$ 分配给 MVQ_{im} 作为该队列的特征覆盖。

2) 根据新到达分组 p 的目的端口集合 O_p 将其表达成分组标识向量 $\vec{v}=[x_0x_1\cdots x_j\cdots x_{N-1}]$，其中 $x_j=1$，若 $j\in O_p$，否则 $x_j=0$。

3) 将分组 p 按照如下规则入队到第 m_o 个多播虚拟输出队列 MVQ_{im_o}：

$$m_o:\max_m\left\{\left\|\vec{v}\odot\vec{v}_m\right\|_1\right\}$$

3.3.2 输入调度(IS)

用 $W_{ij}^{\mathrm{u}}(t)$ 和 $W_{ij}^{\mathrm{m}}(t)$ 标识时刻 t 去往输出 j 的单/多播流已获得的累积服务量，用 $W_{ij}(t)$ 标识去往输出端口 j 的业务流获得的累积服务量。IS_i 为单多播流计算更新份额 $\sigma_{ij}^{\mathrm{u}}(t)$ 和 $\sigma_{ij}^{\mathrm{m}}(t)$：

$$\sigma_{ij}^{u}(t)=\frac{\omega_{ij}^{\mathrm{u}}(t)}{\omega_i(t)},\quad \sigma_{ij}^{u}(t)=\frac{\omega_{ij}^{\mathrm{m}}(t)}{\omega_i(t)} \tag{12}$$

其中，$\omega_i(t)=\sum_j\omega_{ij}(t)$，$\omega_{ij}^{\mathrm{m}}(t)=\sum_{k:j\in o_{ik}}w_{ik}^{\mathrm{m}}$，$\omega_{ij}(t)=\dfrac{r_{ij}(t)}{\sum_{j=1}^{N}r_{ij}(t)}=\dfrac{r_{ij}(t)}{\sum_{j=1}^{N}\left(\sum_{k:j\in o_{ik}}r_{ik}^{\mathrm{m}}(t)+r_{ij}^{\mathrm{u}}(t)\right)}$。为确保不同输出端口间流的公平性，$IS_i$ 为汇聚流 f_{ij}，按照式(13)计算调度更新份额 $\sigma_{ij}(t)$：

$$\sigma_{ij}(t)=\frac{\omega_{ij}(t)}{\omega_i(t)} \tag{13}$$

在每次调度时，IS_i 首先更新累积调度份额：$W_{ij}(t)=W_{ij}(t)+\sigma_{ij}(t)$，$W_{ij}^{\mathrm{u}}(t)=W_{ij}^{\mathrm{u}}(t)+\sigma_{ij}^{\mathrm{u}}(t)$，$W_{ij}^{\mathrm{m}}(t)=W_{ij}^{\mathrm{m}}(t)+\sigma_{ij}^{\mathrm{m}}(t)$。其后，$IS_i$ 根据 $W_{ij}^{\mathrm{u}}(t)$、$W_{ij}^{\mathrm{m}}(t)$ 和 $W_{ij}(t)$ 的大小选择输出调度的单播或者多播流。记 $\phi=\{j_x\}_{1\leqslant x\leqslant N}$ 为对 $\{W_{ij}(t)\}_{1\leqslant j\leqslant N}$ 降序排列后对应的输出索引；$\boldsymbol{P}_{ij}^{\mathrm{HOL}}=\{p_{im}^{\mathrm{HOL}}:j\in o_{im}\}$ 为对应输出端口 j 的多播队头分组集合，记依据队头分组集合 $\boldsymbol{P}_{ij}^{\mathrm{HOL}}$ 的等待时间对各多播队列进行降序排列后的索引结果为 $\varphi=\{m_y\}_{1\leqslant y\leqslant M}$。$IS_i$ 输入调度过程如算法 1 所示。

算法 1 MUMF 输入调度过程

IS_i 调度过程

1) for $j=j_1$ to j_N do
2) if $W_{ij}^{\mathrm{u}}(t)\leqslant W_{ij}^{\mathrm{m}}(t)$ then
3) for $m=m_1$ to m_M do
4) if $j\in o_{im}$ and $s_{ij}^{\mathrm{m}}=1$ then
5) select p_{im}^{HOL}, return;
6) end if
7) end for
8) if $q_{ij}^{u}(t)>0$ and $s_{ij}^{\mathrm{u}}=1$ then

9)　　select p_{ij}^{HOL}, return;
10)　　end if
11)　　else
12)　　if　$q_{ij}^{u}(t)>0$ and　$s_{ij}^{\mathrm{u}}=1$then
13)　　select p_{ij}^{HOL}, return;
14　　end if
15)　　for　$m=m_1$ to m_M do
16)　　if　$j\in o_{im}$ and　$s_{ij}^{\mathrm{m}}=1$ then
17)　　select p_{im}^{HOL}, return;
18)　　end if
19)　　end for
20)　　end if
21)　end for

记 IS_i 选择分组为 p，$\Theta=\{j: j\in o_p, s_{ij}^{\mathrm{m}}=1\}$。若 p 为单播则根据目的端口传输到交叉点队列 UXB_{ij}，更新 $W_{ij}(t)=W_{ij}(t)-1$ 和 $W_{ij}^{\mathrm{u}}(t)=W_{ij}^{\mathrm{u}}(t)-1$；否则，根据扇出集合 o_p 复制到 $\{MXB_{ij}: j\in o_p, s_{ij}^{\mathrm{m}}=1\}$，并更新 $W_{ij\in\Theta}(t)=W_{ij\in\Theta}(t)-1$ 和 $W_{ij\in\Theta}^{\mathrm{m}}(t)=W_{ij\in\Theta}^{\mathrm{m}}(t)-1$。

3.3.3　交叉点调度(CS)

MUMF 交叉点调度根据单播累积份额 $W_{ij}^{\mathrm{u}}(t)$、$W_{ij}(t)$ 和多播累积份额 $W_{ij}^{\mathrm{m}}(t)$、$W_{ij}(t)$ 从集合 $\boldsymbol{XB}_j$ 选择交叉点队列。同理记输出 j 对应的交叉点调度器为 CS_j。记输出端口 j 对应的单播累积份额 $W_j^{\mathrm{u}}(t)=\sum_i W_{ij}^{\mathrm{u}}(t)$，多播累积份额 $W_j^{\mathrm{m}}(t)=\sum_i W_{ij}^{\mathrm{m}}(t)$，$\phi'=\{i_x\}_{1\leqslant x\leqslant N}$ 为对 $\{W_{ij}^{\mathrm{u}}(t)\}_{1\leqslant i\leqslant N}$ 降序排列后对应的索引，$\varphi'=\{\iota_y\}_{1\leqslant y\leqslant N}$ 为对 $\{W_{ij}^{\mathrm{m}}(t)\}_{1\leqslant i\leqslant N}$ 降序排列后对应的索引，$l_{ij}^{\mathrm{u}}(t)$ 为 t 时刻单播交叉点队列 UXB_{ij} 的队长，$l_{ij}^{\mathrm{m}}(t)$ 为 t 时刻多播交叉队列的队长。调度器 IS_i 首先根据 $W_j^{\mathrm{u}}(t)$ 和 $W_j^{\mathrm{m}}(t)$ 确定单多播流调度的公平性，再根据 ϕ' 和 φ' 确定各输入端口间的公平性，如算法 2 所示。

算法 2　MUMF 交叉点调度过程

CS_j 调度过程

1)　if　$W_j^{\mathrm{u}}(t)\leqslant W_j^{\mathrm{m}}(t)$ then
2)　　for　$\iota=\iota_1$ to ι_N in　φ'　do
3)　　if　$l_{\iota j}^{\mathrm{m}}(t)>0$ then
4)　　select MXB_{ij}, return;
5)　　end if
6)　　end for
7)　　for　$i=i_1$ to i_N in　ϕ'　do
8)　　if　$l_{ij}^{\mathrm{u}}(t)>0$ then
9)　　select UXB_{ij}, return;
10)　　end if
11)　　end for
12)　else
13)　　repeat line 7) to line 11)
14)　　repeat line 2) to line 6)
15)　end if

可见，MUMF 交叉点调度首先选择至当前时刻 t 单多播业务已获得的服务和预约服务之间的差额（$W_j^{\mathrm{u}}(t)$ 和 $W_j^{\mathrm{m}}(t)$）中的较大者对应的业务类型(说明该类业务类型与预约带宽之间的差距更大)，然后再从对应业务类型中选取具有最大累积份额（$W_{ij}^{\mathrm{u}}(t)$ 或 $W_{ij}^{\mathrm{m}}(t)$）的单/多播分组。

4　仿真实验

本节从时延、带宽分配的公平性和吞吐量 3 个方面对 MUMF 算法的性能进行仿真评估。实验采用 C++开发的交换系统性能仿真评估系统(SPES, switching performance evaluation system)[27]。SPES 仿真实验环境配置如下：交换结构的规模为 16×16；输入、输出端口的带宽归一化为 1；流量到达过程采用贝努利(Bernoulli)和突发(burst) 2 种业务源[27]；流量分布模型采用均匀业务流分布模型。用 λ_i 标识输入端口 i 的流量到达强度，且 $\forall i,j,\lambda_i=\lambda_j$，单多播业务比例分别为 α 和 $\beta(\alpha+\beta=1)$。假设每个单播分组具有相同的扇出分割数 $\boldsymbol{\Phi}$，本文仅研究“可允许”到达过程，即

$$\lambda_i(\alpha+\boldsymbol{\Phi}\beta)\leqslant 1$$

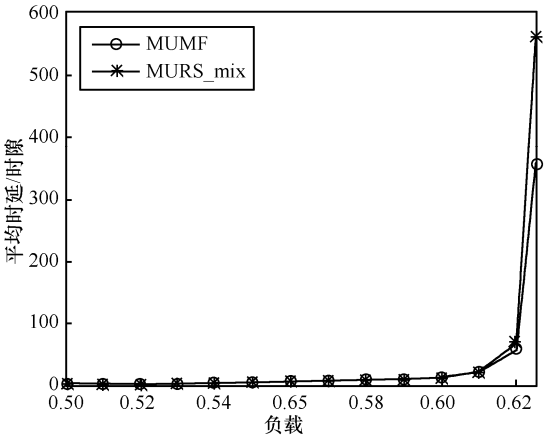
记 $\lambda=\lambda_i(\alpha+\boldsymbol{\Phi}\beta)$，用 λ_{ij} 表示到达输入端口 i、去往输出端口 j 的流量速率，则对于均匀流量到达分布：$\lambda_{ij}=\dfrac{\lambda}{N}$，且单播和多播混合下的到达速率 $\lambda_{ij}^{\mathrm{u}}$ 和 $\lambda_{ij}^{\mathrm{m}}$ 为

$$\forall\, j=i,\begin{cases}\lambda_{ij}^{\mathrm{u}}=\dfrac{\lambda_i\alpha}{N}\\ \lambda_{ij}^{\mathrm{m}}=\dfrac{\lambda_i\boldsymbol{\Phi}\beta}{N}\end{cases}$$

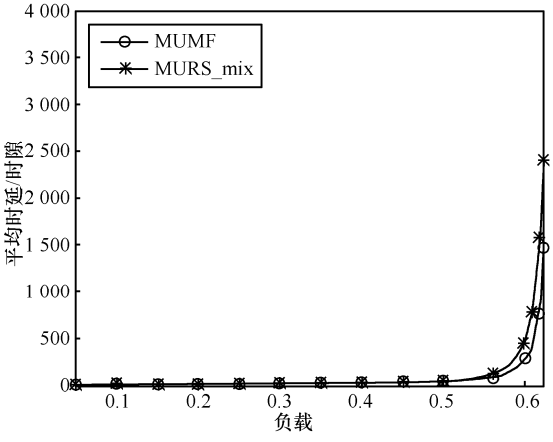
4.1 时延性能

本节分 2 种情形分析 MUMF 调度算法的时延性能：情形 1)，单多播流量混合下 MUMF 调度算法的时延性能，设置扇出端口为 4，单多播流量比例 $\alpha/\beta=4$，改变 λ 使其从 0.1~1.0 之间变化考察单多播分组平均时延大小；情形 2)，单多播业务混合条件下 MUMF 调度算法的时延性能，设置扇出端口为 4，考察在不同单多播流量比例 α/β 下，改变 λ 使其从 0.1~1.0 之间变化考察单多播分组平均时延大小。

图 2 给出了情形 1)流量到达下 MUMF 调度算法的时延性能仿真结果。可以看出，与 MURS_mix 相比较 MUMF 调度算法的性能更优。图 3 给出了情形 2)流量到达下，多播比例分别为 10%、20%和 30%时 MUMF 调度算法的平均时延性能。可见，随着多播流量比例的增加，单多播分组的平均时延不断增大，这是由于多播流量增加相应增大了队头分组阻塞的概率。

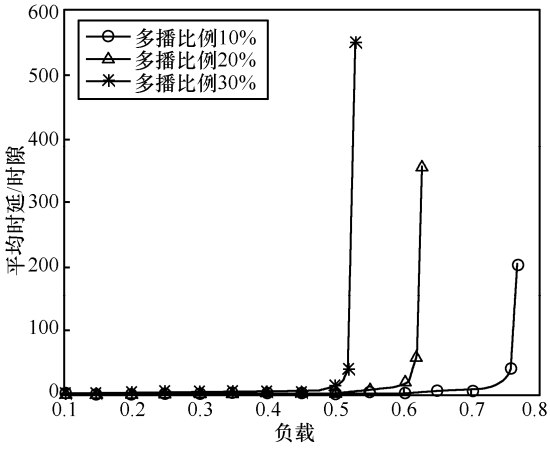


(a) Bernoulli

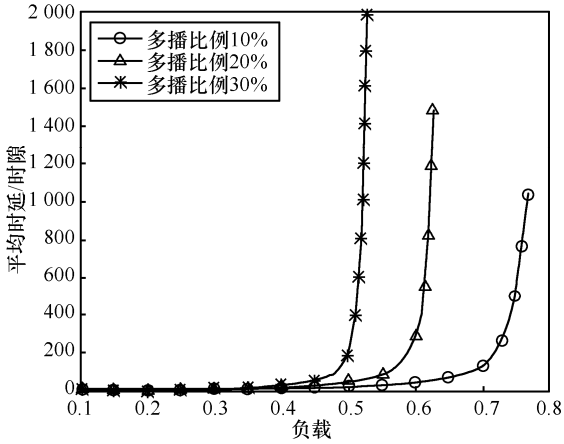


(b) 突发

图 2 情形 1)混合流量到达下分组的平均时延性能



(a) Bernoulli



(b) 突发

图 3 情形 2)混合流量到达下分组的平均时延性能

4.2 公平性能

公平性能评估采用 Bernoulli 和突发 2 种流量模型，为了便于分析，采用 4×4 规模的交换结构，设置每个多播分组的扇出端口数为 4，即每个多播分组产生 4 个输出端口。分 2 种情形进行评估：

情形 1)，仅存在多播流量，输入和输出端口的预约带宽 $r=\left[r_{ij}^{\mathrm{m}}\right]_{4\times4}$ 设置为

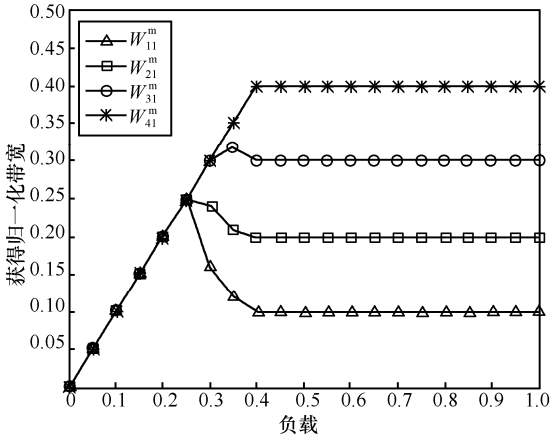
$$\boldsymbol{r}=\left[r_{ij}^{\mathrm{m}}\right]_{4\times4}=\begin{bmatrix}0.10 & 0.20 & 0.30 & 0.40\\ 0.20 & 0.30 & 0.40 & 0.10\\ 0.30 & 0.40 & 0.10 & 0.20\\ 0.40 & 0.10 & 0.20 & 0.30\end{bmatrix} \tag{14}$$

情形 2)，单多播比例 $\alpha/\beta=1$，预约带宽 $\boldsymbol{r}=\left[\left(r_{ij}^{\mathrm{u}},r_{ij}^{\mathrm{m}}\right)\right]_{4\times4}$ 为

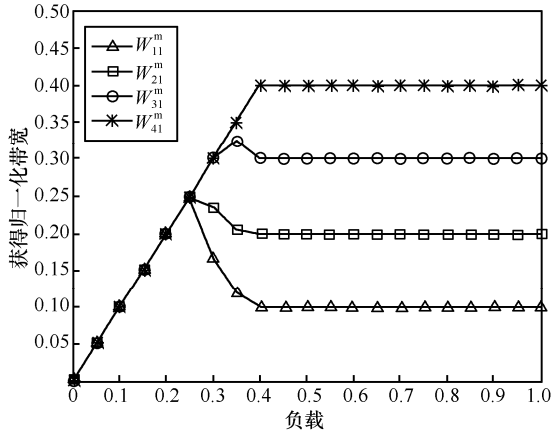
$$\boldsymbol{r}=\left[\left(r_{ij}^{\mathrm{u}},r_{ij}^{\mathrm{m}}\right)\right]_{4\times4}$$

$$=\begin{bmatrix}(0.20,0.05) & (0.15,0.10) & (0.10,0.15) & (0.05,0.20)\\ (0.15,0.10) & (0.10,0.15) & (0.05,0.20) & (0.20,0.05)\\ (0.10,0.15) & (0.05,0.20) & (0.20,0.05) & (0.15,0.10)\\ (0.05,0.20) & (0.02,0.05) & (0.15,0.10) & (0.10,0.15)\end{bmatrix} \tag{15}$$

首先考察情形 1)流量到达下 MUMF 调度算法的公平性能仿真结果，如图 4 所示。图 4 分别给出了 Bernoulli 和突发业务源下各输入端口获得输出端口 1 的调度带宽，其中，W_{ij}^{u} (W_{ij}^{m})分别表示输入端口 i 接受输出端口 j 提供的实际单播(多播)服务量。可以看出，无论是在 Bernoulli 还是突发业务流量到达下，各输入端口多播均获得正比于预约带宽的实际带宽。



(a) Bernoulli

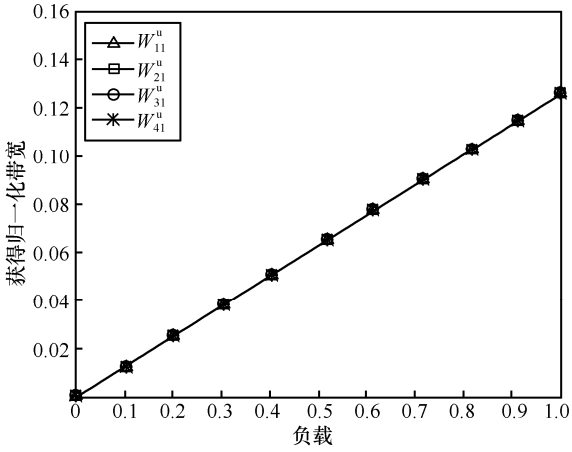


(b) 突发

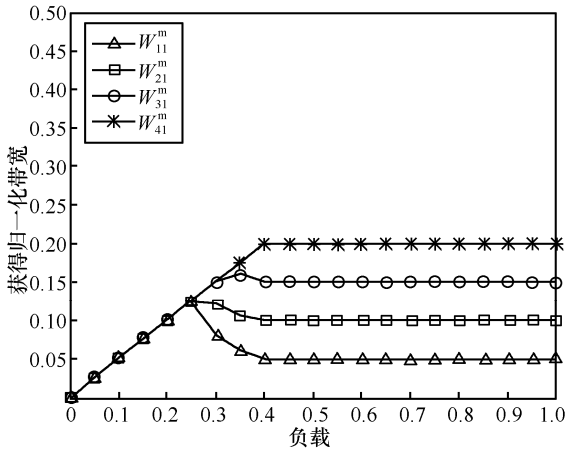
图 4　情形 1)Bernoulli 业务源下公平性能仿真结果

其次，考察情形 2)下 MUMF 调度算法的公平性能仿真结果，由于篇幅限制这里仅给出 Bernoulli 业务流到达下仿真结果，如图 5 所示。图 5 分别为 Bernoulli 单多播公平性能仿真结果，同样考察各输入端口获得输出端口 1 的实际带宽。对于单播业务，由于其预约带宽总和为 0.5，而到达输出端口 1 的单播流量总和为 0.5，因而所有单播分组都被调度输出。

由于输出端口 1 为所有的多播分组预留了 0.5 的预约带宽，且在负载为 0.125 时达到饱和负载，因而从 0.125 开始各多播流受到带宽公平性限制，最终稳定到预约带宽大小。



(a) 单播



(b) 多播

图 5　情形 2)Bernoulli 业务源下公平性能仿真结果

4.3　吞吐量性能

采用 Bernoulli 业务源进行吞吐量评估，多播扇出端口数为 4，单多播流量比例 $\alpha/\beta=4$，改变 λ 使其从 0.1 至 1.0 之间变化。图 6 给出了 Bernoulli 到达下 MUMF 调度算法的吞吐量性能仿真结果。可以看出，MUMF 调度算法的吞吐量随到达强度的增加而增加，最后达到接近 100%的吞吐量。

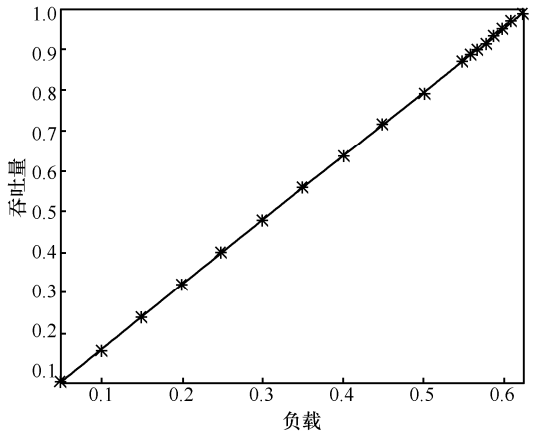


图 6 Bernoulli 业务源下 MUMF 调度算法的吞吐量性能仿真结果

4.4 QM 入队机制

本节通过设定不同的输出端口队列数评估入队机制 QM 的性能，QM 入队性能采用交换系统的平均时延进行衡量，并设定 3 种不同情形对 QM 的性能进行评估：①每个输入端口维护 4 个多播虚拟输出队列；②每个输入端口维护 8 个多播虚拟输出队列；③每个输入端口维护 16 个多播虚拟输出队列。采用 Bernoulli 流量到达过程，多播扇出端口数为 4，单多播流量比例 $\alpha/\beta=4$，改变 λ 使其从 0.1~1.0 间变化。图 7 给出了 Bernoulli 均匀流量到达下 MUMF 调度算法的时延性能。可以看出，随着多播队列数目的不断增加 MUMF 的时延性能不断改善，然而改善幅度变小。

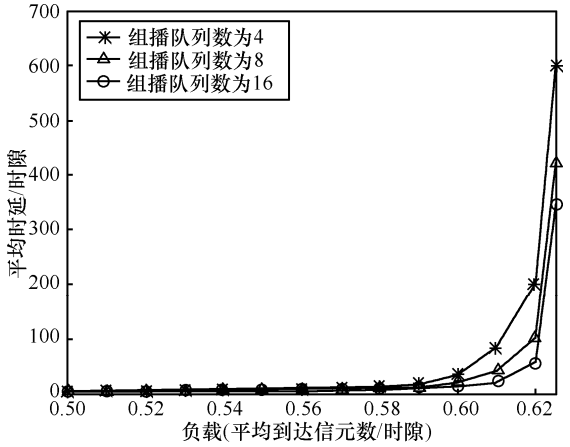


图 7 Bernoulli 均匀流量到达下，MUMF 调度算法的时延性能

5 结束语

以视频点播、VoIP 等为特征的多媒体业务及相关技术成为下一代互联网和三网融合下业务发展的典型特征。多播技术为多媒体业务的运营提供了重要支撑。基于联合输入交叉点排队交换结构，本文首先讨论了 CICQ 下单多播混合调度的理论模型，基于理论模型给出了一种逼近调度算法 MUMF。仿真结果表明 MUMF 调度算法具有良好的性能。依托于这一思想，后期将在支持区分服务方面进行深入研究。

参考文献：

[1] SHEN Y, PANWAR S S, CHAO H J. SQUID: A practical 100% throughput scheduler for crosspoint buffered switches[J]. IEEE/ACM Transactions on Networking, 2010, 18(4):1119-1131.

[2] PAN D, YANG Y Y. Localized independent packet scheduling for buffered crossbar switches[J]. IEEE Transaction on Computers, 2009, 58(2):260-274.

[3] NABESHIMA M. Performance evaluation of a combined inputand crosspoint-queued switch[J]. IEICE Trans on Commun, 2000, 83(3):737-741.

[4] ROJAS-CESSA R, OKI E, JING Z, *et al*. On the combined input crosspoint buffered switch with round-robin arbitration[J]. IEEE Trans on Commun, 2005, 53(11):1945-1951.

[5] MHAMDI L, HAMDI M. MCBF: A high-performance scheduling algorithm for buffered crossbar switches[J]. IEEE Communications Letters, 2003, 7(9):451-453.

[6] ZHANG X, BHUYAN L N. An efficient algorithm for combined input-crosspoint-queued (CICQ) switches[A]. IEEE GlobeCOM[C]. 2004.1168-1173.

[7] JAVIDI T, MAGILL R, HRABIK T. A high-throughput scheduling algorithm for a buffered crossbar switch fabric[A]. IEEE ICC[C]. 2001.1586-1591.

[8] PAN D, YANG Y Y. Localized independent packet scheduling for buffered crossbar switches[J]. IEEE Transaction on Computers, 2009, 58(2): 260-274.

[9] CHANG C S, HSU Y H, CHENG J, *et al*. A dynamic frame sizing algorithm for CICQ switches with 100% throughput[A]. IEEE INFOCOM 2009[C]. Rio de Janeiro, Brazil, 2009. 738-746.

[10] SHEN Y, PANWAR S S, CHAO H J. SQUID: A practical 100% throughput scheduler for crosspoint buffered switches[J]. IEEE/ACM Transactions on Networking, 2008, (99):1119-1131.

[11] MAGILL B, ROHRS C, STEVENSON R. Output-queued switch emulation by fabrics with limited memory[J]. IEEE Journal on Selected Areas in Communications, 2003, 21 (4):606-615.

[12] CHUANG S T, IYER S, MCKEOWN N. Practical algorithms for performance guarantees in buffered crossbars[A]. IEEE INFOCOM[C]. Miami, 2005. 981- 991.

[13] TURNER J. Strong performance guarantees for asynchronous crossbar schedulers[J]. IEEE/ACM Transactions on Networking, 2009, 17(4):1017-1028.

[14] ZHANG X, *et al*. Adaptive max-min fair scheduling in buffered crossbar switches without speedup[A]. IEEE INFOCOM[C]. Anchorage, Alaska, 2007. 454-462.

[15] HE S M, SUN S T, GUAN H T, *et al*. On guaranteed smooth switching for buffered crossbar switches[J]. IEEE/ACM Transactions on Networking, 2008, 16(3):718-731.

[16] PAN D, YANG Z Y, MAKKI K, *et al*. Providing performance guarantees for buffered crossbar switches without speedup[A]. ICST QShine[C]. Berlin, 2009.297-314.

[17] GIACCONE P, LEONARDI E. Asymptotic performance limits of switches with buffered crossbars supporting multicast traffic[J]. IEEE Trans on Information Theory, 2008, 54(2):595-607.

[18] SUN S T, HE S M, ZHENG Y F, *et al*. Multicast scheduling in buffered crossbar switches with multiple input queue[J]. IEEE Trans on Parallel and Distributed Systems, 2009, 20(6):818-830.

[19] MHAMDI L. On the integration of unicast and multicast cell scheduling in buffered crossbar switches[J]. IEEE Trans on Parallel and Distributed Systems, 2009, 20(6):818-830.

[20] DONG Z Q, ROJAS-CESSA R. Output-based shared-memory crosspoint-buffered packet switch for multicast services[J]. IEEE Communication letters, 2007, 11(12): 1001-1003.

[21] BENNETT J C R, ZHANG H. WF2Q: Worst-case fair weighted fair queuing[A]. IEEE INFOCOM[C]. 1996.120-128.

[22] GOLESTANI S. A self-clocked fair queuing scheme for broadband applications[A]. IEEE INFOCOM[C]. 1994.636-646.

[23] NI N, BHUYAN L N. Fair scheduling for input buffered switches[J]. Journal of Cluster Computing, 2003, 6(2):105-114.

[24] PAN D, YANG Y Y. Bandwidth guaranteed multicast scheduling for virtual output queued packet switches[J]. Journal of Parallel and Distributed Computing, 2009, 69(12): 939-949.

[25] HU C, TANG Y, CHEN X, LIU B. Per-flow queuing by dynamic queue sharing[A]. IEEE INFOCOM'07[C]. Anchorage, Alaska, 2007.

[26] HU H C, YI P, GUO Y F. Design and implementation of high performance simulation platform for switching and scheduling[J]. Journal of Software, 2008, 19(4):1036-1050.

作者简介：

扈红超（1982-），男，河南商丘人，博士，国家数字交换系统工程技术研究中心讲师，主要研究方向为高速路由器关键技术、网络流量均衡技术和业务管控技术。

郭云飞（1963-），男，河南郑州人，国家数字交换系统工程技术研究中心教授、博士生导师，主要研究方向为网络安全、宽带信息网络和高速路由器关键技术。

陈庶樵（1973-），男，河南郑州人，博士，国家数字交换系统工程技术研究中心副教授、硕士生导师，主要研究方向为互联网流量均衡技术和业务管控技术。

伊鹏（1977-），男，湖北黄冈人，博士，国家数字交换系统工程技术研究中心副教授，主要研究方向为高速路由器关键技术、网络流量均衡技术和高速流识别技术。