一种共享存储交换结构中的双门限队列控制策略

汪 洋[1],余少华[2]
(1. 华中科技大学电子与信息工程系 湖北省智能互联网技术重点实验室,湖北武汉 430074;
2. 武汉邮电科学研究院 光纤通信技术和网络国家重点实验室,湖北武汉 430074)

摘 要: 传统的共享存储交换结构的门限控制算法通常以当前各个队列长度为依据,缺乏对网络节点设备全局流量场景的考虑,且对组播的支持不足. 本文提出使用有效业务量作为控制各个端口队列门限的主要依据,让各个端口承担相同的流量压力,从而使系统保持均衡状态. 在经典的有效带宽理论的基础上,结合输入流量速率和分配的缓冲区大小一起来定义输出端口的有效业务量,给出了"缓存换带宽"的计算公式,对流量压力进行准确度量. 双门限的使用使得共享存储空间既能够保持在平均意义下的平衡,又能实现突发时段的调剂. 进一步,对组播信元的转发也可以纳入这个算法框架. 模拟结果显示,在不同的流量模式下,这个算法比传统动态门限算法在取典型值 $\alpha = 1$ 和组播浓度为 30 %时,对芯片的使用效率和端口的公平性分别提高 15 %和 25 %以上.

关键词: 交换结构; 共享存储; 队列门限; 有效流量; 组播
中图分类号: TP393 **文献标识码**: A **文章编号**: 0372-2112 (2009) 07-1400-07

A Dual Threshold Buffer Management Scheme in Shared Memory Switch Fabric

WANG Yang[1], YU Shao-hua[2]
(1. *Department of Electronic and Information Engineering, Internet Technology and Engineering R&D Center, Huazhong University of Science and Technology, Wuhan, Hubei* 430074, *China*;
2. *Wuhan Research Institute of Post and Telecommunications, State Key Laboratory of Optical Communication Technologies and Networks, Wuhan, Hubei* 430074, *China*)

Abstract: This paper concerns with the threshold control problem in a shared memory switch fabric. Traditional threshold control schemes are insensitive to the runtime traffic scenarios, and lack an efficient support for multicast forwarding. A novel scheme is presented in this paper, which aims to balance traffic pressure to the output ports and keep the system in an equilibrium state. Derived from a further exploitation of the developed effective bandwidth theory, the proposed scheme delineates an expressive formula that can calculate the shift of bandwidth burden to memory allocation. The association between bandwidth and memory utilization can be used to balance the traffic pressure by adjusting the per-queue thresholds in the shared memory. Furthermore, the presented framework can also encompass the support for multicast cells. The result of simulation shows that the scheme outperforms the traditional dynamic threshold in terms of efficiency and fairness under variant traffic modes.

Key words: switch fabric; shared memory; queue threshold; effective traffic; multicast

1 引言

交换结构是交换/路由设备完成转发功能的核心,按照进入信元的缓存策略分类,可以分为输入排队,输出排队和共享存储排队. 其中前两种实现方式通常还包括一个空分(space-division)单元,典型的如 Crossbar[1]. 经典的研究认为[2],输出排队能够实现最优的性能,同时避免诸如线头阻塞(HOL)之类降低系统效率的因素,但是这种理论分析的结论建立在空分结构能提供较大加速比(Speedup)的假设之上,当前的交换结构具有较高端口密度,使得提供较大的加速比非常困难. 与输入或输出排队类型不同,共享存储(SM)的交换结构将交换的功能全部在一个公共的存储器(也称共享缓存)中实现,其排队原理非常类似于输出排队,且由于各个队列共享存储空间,提高了对存储资源的利用率[3],加上没有使用其他空分结构,使得共享存储交换结构有较高的效率和较低的实现成本,从而在单板交换或大型交换系统的前端交换中占据主导地位[4].

SM 的一个难点在于存储器的管理和分配. 存储于共享存储中的信元按照到不同的输出端口存储形成不

收稿日期:2007-10-12;修回日期:2009-05-04
基金项目:电子基金 IPV6 网络设备研发与产业化(No. 04c262242109719);国家 863 高技术研究发展计划(No. 2003AA1Z1180, No. 2005AA121411)

同的输出队列,系统负载较重时,各队列会争用共享缓存,特别是在不均匀流量条件下,重负载端口的队列会大量挤占共享缓存而影响其他端口对缓存的正常使用.

对队列进行门限控制是通常采取的措施,此外还有不使用门限而对信元进行推出(Push Out, PO)操作的算法[5]. PO 算法理论上是最优的,但其实现困难,因此一般作为一个基准的参考算法和其他算法进行对比. 传统的门限算法包括静态门限(ST),动态门限(DT)[6,7]等. 其中 ST 缺乏对流量场景的适应性,DT 受到广泛关注,但文献[8]说明其表现不稳定,原因在于它试图用一个简单的线性模型来计算非线性的排队系统. 最近提出的研究包括使用衰减函数来计算每队列的门限[9],使用模糊逻辑和遗传算法来进行计算[8].

为了让共享存储交换结构也同时能够高效地转发组播流量,通常采用两种机制来进行组播和单播的混合分组转发[1]. 第一种是使用专用组播信元队列(Dedicated Multicast Queue,DMQ)的共享存储,组播信元总是有最高的优先级,只有当组播信元队列为空,或者组播队列的线头(Head of Line,HOL)不是发往同一个端口时,才对发往这个端口的单播信元进行输出操作;另一种是使用地址拷贝队列(Address Copy Queue,ACQ),每个输出队列以 FIFO 方式保存信元在存储空间中的地址. 一个组播信元的地址会出现在多个端口的输出地址队列中,而这个组播信元存放于共享存储空间.

DMQ 方案的缺陷在于,当突发组播流量的带宽达到一定程度时,将严重妨碍单播信元的正常转发,因为组播信元具有相对较高的优先级. ACQ 较好地解决了拥塞问题,因此虽然它也有别的弱点,如还需要额外的空间来存放地址队列,但仍然成为共享存储交换结构支持组播的主要机制. 缺乏对组播考虑是已有算法的另一个重要不足,为此,需要寻求能够在组播-单播的混合流量环境下都能够提供高效的信元准入(门限)机制,这种机制应该满足:(1)易于在线计算;(2)适应流量场景;(3)维持组播和单播信元占用缓存资源的合理平衡;(4)实现系统效率和公平的合理折衷.

2　基于有效带宽理论的流量均衡

关于信元在共享存储空间的准入问题,也就是端口队列的门限问题的研究一般都确定了几点可以衡量其算法性能的准则,譬如信元丢失率,共享缓存利用率等,从而对各自的提出的算法在性能上进行对比. 遗憾的是,虽然这些研究的出发点都试图解决各个队列之间的公平性的问题,但是都缺乏一种有效的准则来衡量算法的公平性,最后都只能按照系统总体性能或效率来度量其算法.

对有效带宽(effective bandwidth)或有效容量的研究始于对 ATM 交换系统对多种服务流的容纳问题,Chang,Kesidis,Courcoubetis 等人借助于平稳过程队列的渐进性质和概率中的大偏差理论,逐渐形成了比较成熟的有效带宽理论[10~12]. 本文试图在一定意义下借助有效流量带宽的观点来建立一种公平性的衡量机制. 这可以直观地描述如下:设一段时间到某个端口的流量为一个平稳过程,现在需要在共享存储区中划分一定大小的空间来容纳这个到来的信元流. 假如这个划分的空间为无限大,那么这个信元流的有效带宽可以按照上述经典的有效带宽理论,流量有效带宽[13]

$$B_\infty = \lim_{t\to\infty}\frac{1}{t\theta}\log E e^{\theta X(0,t)} \tag{1}$$

其中 $X(0,t)$ 表示时间区间 $(0,t)$ 之间到来信元的累加量,而 θ 是一个与 QoS 相关的变量,即队列长度溢出概率,也就是

$$\Pr(q(\infty) \geq x_0) \approx e^{-\theta x_0} = \varepsilon \tag{2}$$

其中 $q(t)$ 表示 t 时刻队列的长度. 这里队列溢出与假设的"无限大缓存"并无矛盾,因为无限大缓存是个假设的条件,在这种假设下,可以得到实际队列长度超过给定的长度 x_0 的概率. 然而实际分配的存储空间不可能是无限大,那么从直觉可看出,实际分配得到的存储空间越小,相对于这个分配到的越小的空间,实际的流量有效带宽就会越大. 这是因为,空间越小,可以腾挪或周转的空间就越小,所以流量的相对带宽就会越大.

基于以上分析,可以考虑将每个输出队列相对于分配得到的存储空间的有效流量作为衡量各个队列公平性的一种度量,也就是说,对于一个共同的信元丢失概率 ε,如果每个队列分配到的存储空间使得各自进入的流量觉得它们拥有相同"宽裕"程度的空间,从而溢出这个空间的概率都维持在 ε 水平,那么可以认为这种分配方案是公平的.

3　有效业务量的分析

设共享存储交换结构输入端口和输出端口均为 N 个,共享存储器大小为 M,将时间离散化为时隙,每个输入和输出端口的线速均为 1. 记时隙 t 到输出端口 i 的流速为 $x_i(t)$,而从时隙 s 到时隙 t 之间到端口 i 的累积流量为 $x_i[s,t]$.

3.1　"缓存换带宽"的理论解释及经验公式

直觉和启发式的描述说明了使用有效流量作为公平性基准的合理性. 但是需要分不同的场合讨论式(1)、(2)的有效性. 在用式(1)计算流量有效带宽的前提下,式(2)成立是有条件的,即它需要式(1)所求得的 B_∞ 不超过端口的输出速率. 这就是说,式(1)是根据统计观察的结果对输入信元的"等效带宽"的估计. 若这个等效带宽超过了端口的输出速率,即"服务能力",就

意味着该输出端口过载,这时显然的结果就是该端口的输出队列会无限增长,如果不加限制就会挤占其他端口的缓存额度从而导致系统性能的急剧下降;若式(1)的计算值小于端口的输出速率,式(2)的估计一般是保守的,也就是实际的信元丢失率往往低于式(2)中的ε;若式(1)的计算值恰好等于端口输出速率,则式(2)可以很好地成立,但这种"恰好"在实际场合几乎不会发生.尽管式(1)、(2)成立的条件受到上述限制,但是仍然提供了对信元输入速率和队列长度的一种有效的定界(bounding)方式,因此,在没有过载端口的情形下,把它作为诸队列占用缓存公平性的度量仍然是合理和可行的.此外,假定下列两个条件成立:(1)输入流量在足够长时间内是平稳增量的;(2)分配的存储区相对于信元长度非常大(即要讨论的 x_0 足够大).这两个条件的满足在实际网络环境中都不牵强,因为骨干网中的数据流汇聚了大量的业务,这些业务量的聚合流量表现出对时间增量的平稳性,即时齐的;对于第二个条件,各个端口分配的缓存比单个信元的大小要大很多.

式(1)的计算无疑十分复杂,作为一种近似,以一个较大的 T 来替换式(1)中的极限计算,因此,重新定义 $B_T=\log \mathrm{E}\mathrm{e}^{\theta X(0,T]}/(T\theta)$ 作为有效带宽的表达式.但即便如此,这个表达式仍然不适合在实际应用中采用.为了给出有效流量的可行计算方法,这里先采用两个具体的模型来分析.

先考虑泊松流.设 $x_i(t)$ 为一个输入强度为 λ 的泊松过程.显然 $x_i(0,T]$ 服从强度为 λT 的泊松分布,故 $B_T=\log \mathrm{E}\mathrm{e}^{\theta X(0,T]}/(T\theta)=\lambda(e^{\theta}-1)/\theta$.给定丢包率 ε 时,考虑到前面的条件 x_0 很大,从式(2)得知 θ 很小,因此忽略 θ 的高阶小量,得到

$$B_T\approx\lambda(1+\theta/2)=\lambda(1+(-\log\varepsilon)/(2x_0)) \tag{3}$$

可以看出在泊松流的假设下,B_T 具有 $\lambda(1+K/buf_i)$ 的形式.其中 K 为一个常数,依赖于QoS的选取,buf_i 为分配给指定端口的缓冲区大小.

再考虑被广泛采用的ON-OFF模型.这个模型的一个单数据源处于两种状态:处于ON状态时以恒定速率 R_p 发送数据,处于状态OFF时不发送数据.记 α 为从OFF状态到ON状态的变迁率,β 为从ON状态到OFF状态的变迁率,若 $x_i(t)$ 为 N 个独立的单数据的叠加,利用状态转移矩阵的最大特征值可以得到[14]:

$$B_T=\frac{1}{T\theta}\log \mathrm{E}\mathrm{e}^{\theta X(0,T]}=N\left(\frac{R_p}{2}-\frac{\alpha+\beta}{2\theta}\right)+N\sqrt{\left(\frac{R_p}{2}-\frac{\alpha+\beta}{2\theta}\right)^2+\frac{\alpha R_p}{\theta}}$$

对之进行一些代数处理:由式(2),$\theta=(-\log\varepsilon)/x_0$,此外记 $p=\frac{\alpha}{\alpha+\beta}$,$k=\frac{\beta x_0}{R_p(1-p)(-\log\varepsilon)}$ 则

$$\frac{B_T}{NR_p}=\frac{1-k}{2}+\sqrt{\left(\frac{1-k}{2}\right)^2+kp} \tag{4}$$

由假设 x_0 很大,因此 k 很大,对式(4)作泰勒展开并忽略 $1/k$ 的高阶小量(详细推导略),得到:

$$B_T=NR_p p\left(1+\frac{1}{k-1}\right) \tag{5}$$

同样的道理,如果使用输入流的Bernoulli模型或者Gaussian模型,计算等效带宽都得到形式上同式(3)和式(5)相似的表达式,即形如

$$B_T=\mu(1+K/Buf) \tag{6}$$

的等效带宽表达式.其中 Buf 为分配得到的缓存大小,而 K 在流量特征与QoS参数给定的条件下是一个常数,μ 是输入流量的平均速率.在等效带宽的意义下,式(6)正是直觉上"缓存换带宽"的理论解释:对于一个事先给定的缓存大小 Buf_1,稳定状态时的丢包率 δ_1 可以由式(2)近似得到;如果增加缓冲区,稳定状态时的丢包率 δ_2 必然降低.这意味着如果仍然满足于以前的丢包率 δ_1,这个增加了的缓冲区可以承受更大的平均流量,即式(6)中 Buf 的增大可以"换取"更大的 μ 而保持 B_T 不变.此外,B_T 通常比平均速率要大,这个性质不仅在式(3)、(5)中得到反映,还可以直接通过 B_T 的定义 $B_T=\log \mathrm{E}\mathrm{e}^{\theta X(0,T]}/(T\theta)$ 以及log函数的凹性直接得到.因此,在"实时"环境中,选用式(6)作为诸输出队列所承载的流量强度的经验表达式是合理的,其中 K 是一个常数.

3.2 端口缓存大小的静态计算

考虑式(6),为了让各个端口都具有相同的等效带宽,则下列两个等式成立:

$$\mu_i(1+K/Buf_i)=B_T \tag{7}$$

$$\sum Buf_i=M \tag{8}$$

其中 M 为端口输出队列所占用的缓存总合.在组播ACQ情形下,总的缓存大体可以划分为两部分,一部分用于存储组播信元,余下的部分才用于划分给诸端口.因此 M 实际上是全部存储空间 Buf_{Total} 除掉一部分用于存储组播信元空间 Buf_m 后的剩余部分.一般而言,这两个等式可以联合得到每个 Buf_i 的值,但是在特殊情形,式(7)、(8)可能无解,譬如 $\mu_i=0$ 时.因此根据具体的流量场景,式(7)、(8)式尚需做进一步的修改.

4 支持组播的队列门限算法

离线计算主要基于一段时间内对进入各个端口输出队列的流量采集,根据 μ_i 的大小,将输出端口按照流量强度分为三类:Ⅰ类为非活动端口,记为 S_{I};Ⅱ类为活动端口,记为 S_{II};Ⅲ类为过载端口,记为 S_{III}.对三者的分类按照经验的阈值进行:即指定两个阈值 $0<\Gamma_{act}$

<1 和 $1<\Gamma_{over}$,当 $0\le\mu_i<\Gamma_{act}$ 时, $Port_i\in S_{\mathrm{I}}$;当 $\Gamma_{act}<\mu_i<\Gamma_{over}$ 时, $Port_i\in S_{\mathrm{II}}$;当 $\Gamma_{over}\le\mu_i$ 时, $Port_i\in S_{\mathrm{III}}$.

4.1　端口输出队列的双门限机制

为了对输入的信元有高效而不失公平准入机制,采用每端口的双门限操作.对每个端口 $Port_i$ 确定两个门限值:低门限 Th_i^{lower} 和高门限 Th_i^{upper}, Th_i^{lower} 用于触发各个端口门限的再计算, Th_i^{upper} 用于判断是否丢弃信元,各个输入端口并行处理的流程如下.

Step0:取信元,判断目的端口,设目的端口为 $Port_i$;

Step1:取输出 $Port_i$ 队列长度 q_i 和 Th_i^{upper},若 $q_i=Th_i^{upper}$,丢弃该信元,转 Setp0;否则将信元加入端口 i 输出队列尾端等待发送, $q_i=q_i+1$;

Step2:取输出 $Port_i$ 队列长度 q_i 和 Th_i^{lower},若 $q_i\ge Th_i^{lower}$,通知控制器重新计算各个 Th_i^{lower} 和 Th_i^{upper};

Step3:转 Step0.

如何确定各个 Th_i^{lower} 和 Th_i^{upper} 是整个操作的关键.为了得到这两个值,需要结合式(7)、(8)和各个特定端口的类别来一起计算:

若 $Port_i\in S_{\mathrm{I}}$,此时因为 μ_i 太小,故该端口的流量几乎可以忽略,因此设置 Th_i^{lower} 在数值上等于活动端口的阈值 $\Gamma_{act}\cdot M/\Lambda$,此时设置 Th_i^{upper} 为一个块,大小为 M/Λ;

由于 S_{I}中的端口不参与以有效业务量为依据的输出端口门限计算,因此 S_{II}和 S_{III}中的端口门限计算需要对式(8)进行修改,这些端口的缓存在 S_{I}使用的剩余存储空间中划分:

$$\sum_{i\in S_{\mathrm{II}}\cup S_{\mathrm{III}}}Buf_i=M-|S_{\mathrm{I}}|\cdot M/\Lambda \tag{9}$$

对于 $Port_i\in S_{\mathrm{II}}\cup S_{\mathrm{III}}$,根据式(8)和(9)计算各个活动或过载端口的预分配缓存 Buf_i,取 $Th_i^{lower}=Buf_i$,而取 Th_i^{upper} 为能容纳 Th_i^{lower} 的以 M/Λ 为粒度的最小存储块.

4.2　组播信元的门限

ACQ 的组播实现方式中具有以下两个性质:(1)每个单播信元与其地址拷贝具有一对一的映射关系,而每个组播信元与其地址拷贝(实例)具有一对多的映射关系;(2)单播信元与组播信元共享公共的存储空间,且每个端口的门限实际上是为单播信元设置了准入机制.对于组播信元,由于存在不同的实例需要进入不同的端口地址拷贝队列,因此不能使用每个端口的门限来对组播信元进行准入控制.显然,应该为组播信元分配多大的空间,即组播信元的门限设置是一个需要和前面分析的端口门限并列考虑的问题.假设可以求得组播信元的缓存空间 Buf_m,则式(8)和(9)中的 M 满足

$$M=Buf_{Total}-Buf_m \tag{10}$$

为了计算 Buf_m 的大小,考虑熟知的 Little 公式:$L=\lambda W$,并将其限制于组播信元的缓存与释放,其中 L 是组播信元队列的长度,λ 是组播信元的平均到达率,而 W 是组播信元的平均等待时间.显然,在平衡状态下,L 可以理解为组播信元所要占用的存储空间.因此有:

$$Buf_m=L=\lambda W \tag{11}$$

上式中系统组播信元的到达率 λ 是一个较易测得的量,而对于组播信元的平均等待时间 W,若不对组播流量作充分简化的假设,难以给出显示的解.但考虑到式(11)的目的不外计算组播信元的进入共享存储空间的门限,如果能给出一个 W 的合理上界而不对组播流量作过多的假设,对于实际应用仍然是可行的.为此考虑任意一个组播信元 C_M,这个信元有 r 个实例,分别位于端口 $Port_{i_1},\cdots,Port_{i_r}$,这些端口如果存在地址拷贝的门限 $T_{i_1},\cdots,T_{i_r}$,则该组播信元的等待时间 $W_M\le\max\{T_{i_1},\cdots,T_{i_r}\}\le\max\{Buf_k|$ 端口 k 具有组播信元流$\}$,可以使用这样的记法是因为已经将每个端口的转发速度定义为单位 1.从而,组播信元的门限可以使用 L 这个宽松的上界来表示:$Buf_m=\lambda\cdot$具有组播信元端口的门限的最大者.

4.3　单播-组播的联合分析

由前面的分析,组播信元的门限和诸端口的门限共同划分了共享的存储空间(这里忽略信元地址拷贝所占的少量存储区域).因此,联合式(8),(9),(10),(11)可以得到计算端口门限和组播门限所共同需要满足的约束,集中表述如下:

$$\mu_i(1+K/Buf_i)=B,\ i\in S_{\mathrm{II}}\cup S_{\mathrm{III}} \tag{7'}$$

$$\sum_{i\in S_{\mathrm{II}}\cup S_{\mathrm{III}}}Buf_i+Buf_m=Buf_{Total}-|S_I|\cdot(Buf_{Total}-Buf_m)/\Lambda \tag{12}$$

$$Buf_m=\lambda\cdot\max_{i\in S_{\mathrm{II}}\cup S_{\mathrm{III}}}\{Buf_i|\text{端口 }i\text{ 具有组播信元}\} \tag{13}$$

上述方程中 μ_i、λ 为针对流量场景的测量值.在每个端口的双门限算法中,若 q_i 超过了低门限 Th_i^{lower},则触发整个系统各个输出端口门限的重新计算,这种情况容易在流量场景发生快速变化下发生.但是在骨干网中,由于聚合效应,各个端口的流量通常比较稳定,因而场景的变化通常是非常平滑的.为了能更有效地动态捕捉场景特征,而不是到了某个端口的队列要到达"警戒线"时才进行重计算,交换系统除了使用触发重计算外,还需使用定时重计算的机制.

5　实验分析

为了验证"缓存换带宽"经验公式(6)的有效性,对一款自己研制的接入交换设备 A1180 进行实验,该设备采用共享存储交换结构.搭建的拓扑环境如图 1 所示:A1180 的一侧端口连接了若干用户,而另一侧上联端口连接服务器,主要提供视频服务.服务器和 A1180 之间

做 8Mbps 的速率限制,并且限制上联端口的缓存门限为 24KB. 此条件下,各个用户在使用诸如 FTP 传输大文件时整个系统工作都正常,速率限制也保持得较好,但是如果使用视频点播,在各个用户点播的平均流量速率之和小于 8Mbps 的条件下,用户收看视频的质量较差,图像停顿明显.

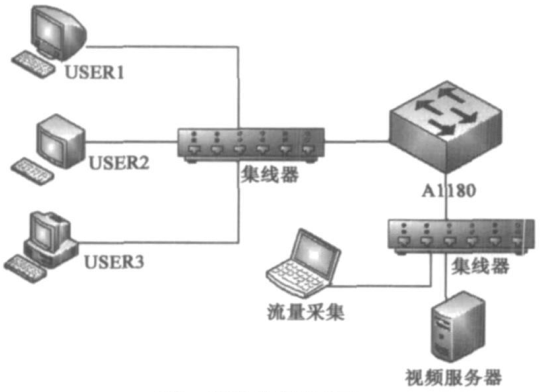

图1 实验分析的拓扑

在两侧使用集线器和流量监测工具 Capsa 对流量进行分析,表 1 显示了服务器侧的流量监测结果,流量的平均速率仅为 1.01Mbps. 在 24KB 缓存的配置下,平均速率仅为 1Mbps 的码流却消耗了 8Mbps 的带宽,可见此场景中视频流的实时波动较大. 考虑到用户的实际平均流速应该在 3M 左右,将现有的结果代入式(6),$B_T=8$, $Buf=24$, $\mu=1.01$. 如此可得到 $K=166.1$. 按照实际的用户需求,平均的码率应为 3Mbps,即 $\mu'=3$,重新计算可得 $Buf'=99.66$KB. 即将服务器侧的缓冲区大小设为 100KB 左右,理论上应该能有较好的效果. 由于寄存器设置的粒度限制,将其端口缓存设置为 96KB,且不对 8Mbps 的限速做任何调整. 再次试验,此时实际平均流量为 3.24Mbps,通过在用户主机上查看视频播放效果,发现图像非常流畅,状态恢复正常.

表 1 实验中两种缓存门限下的播放速率对比

Traffic	Buffer Size	Bytes	Packets	Avg Utilization	Avg bps	Avg pps
Total Traffic	96KB	41.752 MB	197,914	3.243 %	3.243 Mbps	1,832.537
	24KB	13.377 MB	34,877	1.011 %	1.011 Mbps	314.207
Broadcast Traffic	96KB	2.201 KB	29	0.000 %	166.963 bps	0.269
	24KB	1.188 KB	19	0.000 %	87.640 bps	0.171

6 数值计算与仿真验证

6.1 效率-公平性模型

对于门限控制算法的效率问题,有下列几个直觉上的准则:(1) 效率一般以整个共享存储交换结构运行期的性能来衡量而不针对单个特定端口或特定的数据流,在这个意义下,通常希望"过载"的输入条件下共享存储空间的利用率尽可能高;(2) 在存储区利用率一定时,系统应该有尽可能低的丢包率. 考虑下式:

$$input_i = output_i + dropped_i + \Delta q_i \qquad (14)$$

其中 $input_i$, $output_i$, $dropped_i$, Δq_i 分别表示端口队列 i 在一个时隙中的进入信元量,输出信元量,丢弃信元量,以及该时隙末的队列长度与时隙初时队列长度之差. 显然:

(1) $|\Delta q_i| \le Buf_i$;

(2) $dropped_i > 0 \Rightarrow input_i > output_i = 1$

定义单端口的相对丢包系数为: $RL_i = \frac{dropped_i}{input_i} \cdot \frac{1}{\mu_i}$ 相对丢包系数 RL_i 的含义是丢包率与端口输入流量均值的比,这样的定义可以避免单纯地从丢包的绝对数量来衡量端口的服务能力而不顾流量的大小. 显然,端口的相对丢包可以平行地应用于组播队列,不再赘述. 进一步定义系统的效率函数如下:

$$E = \exp(\text{共享存储利用率} - 1) \cdot \exp(-\sum RL_i)$$

这个定义符合直觉上对"效率"的理解:如果共享存储区已用满,则效率仅与各个端口的相对丢包系数相关;否则空闲的缓冲区越大,或者与此同时丢包的数量越小,系统效率越低. 例如在丢包了 100 字节时,空闲存储区为 200 字节时的系统应该比空闲存储区为 20 字节的效率要低.

与效率紧密相联系的另一个性能参数是公平性. 仍然以各个端口的丢包或缓冲区的空闲程度为主来考虑公平性. 前面已经对发生丢包的端口定义了相对丢包系数,对于队列未超过门限的端口,需要一种能够衡量在给定门限条件下端口空闲程度的参数. 为此定义非过载端口(S_I和 S_{II}类端口)在未发生丢包时的空闲系数为:

$$I_i = (Buf_i - Q_i)/Buf_{Total}$$

同相对丢包系数一样,空闲系数也可应用于组播队列,只需将相关的 Buf_i 和 Q_i 取为组播队列的门限和实际队长即可. 将组播队列看成是位于一个特殊的端口 P_M,记

$$B_i = \begin{cases} RL_i, & \text{若端口 } i \text{ 在 } \Delta \text{ 时间内发生了丢包} \\ -I_i, & \text{若端口 } i \text{ 在 } \Delta \text{ 时间内未发生丢包} \end{cases}$$

从而,定义系统的公平度

$$Fairness_\Delta = \max_{i,j \in \{1,\cdots,N\} \cup P_M} |B_i - B_j|$$

6.2 仿真研究与分析

模拟一个 8×8 的共享存储交换结构. 全部的存储区大小为 16000 信元. 数据按照 $100\mu_i$ 个 ON-OFF 单数据源的叠加生成,其中 μ_i 用来调节流量的强度,即载荷. 对于单个的 ON-OFF 数据源,以 0.2 的概率进入 ON 状态,平均逗留时间为 10 个时隙,平均每时隙产生 2.5

个信元;进入 OFF 的概率为 0.8,其平均逗留时间为 3.75 时隙.输入和输出的线速为 $\mu_i = 1$ 时叠加后数据的平均流量.全局存储空间被划分为 256 个相等大小的块,活动端口和过载端口的阈值分别为 $\Gamma_{act} = 0.25$ 和 $\Gamma_{over} = 1.15$.

用流量场景来描述单播的流量,流量场景用一个 1×8 的向量来表示,如[0,0.1,0.2,0.4,0.7,0.8,1,1.2],其中第 i 个分量表示发往端口 i 的流量载荷.对于组播信元而言,其信元的到达仍然按照前述 ON-OFF 源生成,组播的扇出按照典型的分布确定,取最小扇出 $f_{min} = 2$,最大扇出 $f_{max} = 8$,每次模拟生成的组播信元扇出 f_i 按照分别几何分布来生成.为了模拟网络流量的变迁,对单播数据流的模拟采用场景的变换来实现.模拟中采用了连续的三个场景:[0,0.1,0.2,0.4,0.5,0.7,1,1.2] →[0.6,0.7,0.2,1,1.1,0.6,0.5,0.4] →[1.2,1,0.7,0.5,0.4,0.2,0.1,0],每个场景持续一秒.

在模拟实验中,端口队列门限的求解主要通过方程联立(7')、(12)、(13)求得,求解的方法仍然使用便于实现的二分法,时段中的时隙数 $A = 32$,连续的时段数 $B = 16$.实验中用于计算各个门限值的过程耗时都保持在 10ms 以内.若选用一个时隙的时间长度为 1ms,则取样时间 $A * B = 512$ms.也就是从采样到计算出门限值的过程在 600ms 以内,因此最后按照每秒 1 次的频率进行门限定时重计算,这样对系统的开销是适度的.

表 2　本文算法和 DT 算法的组播丢包率对比

	Multi Portion = 0.1		*Multi Portion* = 0.3		*Multi Portion* = 0.5	
	$Fout_p$ = 0.5	$Fout_p$ = 0.2	$Fout_p$ = 0.2	$Fout_p$ = 0.5	$Fout_p$ = 0.2	$Fout_p$ = 0.5
DT(α=2)	< 0.001	< 0.001	0.29	0.29	0.33	0.31
DT(α=1)	< 0.001	< 0.001	0.26	0.25	0.22	0.19
DT(α=0.5)	< 0.001	< 0.001	0.13	0.11	0.17	0.17
本文算法	0	0	0	0	< 0.001	< 0.001

为了观察 DT 算法在不同的 α 取值下对组播信元的影响,实验调整输入组播信元的两个主要参数:组播信元在整个输入信元中的比例 *Multi Portion* 和组播扇出的参数 $Fout_p$.组播扇出数 f 采用以 $Fout_p$ 为参数的几何分布.得到的结果如表 2 所示.可以看出,对于固定的 *Multi Portion* 和 $Fout_p$,DT 算法随着 α 的增大而使组播信元丢包率增大.在组播信元的比例不高时,DT 算法不管使用何种 α,组播信元的丢失率都保持在较低的水平,而随着组播信元的比例升高,DT 算法逐渐显示出不足,即组播信元的丢失率显著增高.*Multi Portion* = 0.3 时 DT 算法在不同的 α 下的丢包率如图 2 所示.实验的结果和文献[8]所提到的结论是一致的,即 DT 算法选取较大的 α 对组播非常不利.事实上,DT 算法中关于 α 的选取一直都是个难点.DT 算法的提出者 Choudhury 在文献[6]中也这个问题做了充分的研究,最后他们认为,在还没有更好的算法能动态地实现 α 按场景调整时,选取 $\alpha = 1$ 应该是最合理而有效的.Choudhury 本人后来加盟了 Broadcom 公司,并将 DT 算法应用于 BCM 的一系列高端商用芯片(如 BCM56700 系列),这些芯片中 α 默认即取值为 1,虽然芯片提供了寄存器可以静态修改这个取值,但是几乎没有人愿意冒险去修改.因此,后续的实验均以 $\alpha = 1$ 作为 DT 算法的典型配置.

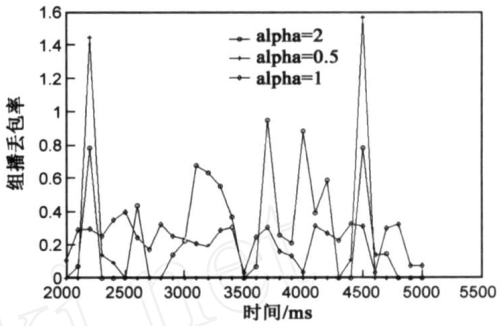

图2　DT算法在α=2, 1, 0.5时的组播丢包情况

图 2 还显示了一个特殊的现象,就是在 $\alpha = 0.5$ 时,DT 算法在局部的组播信元"丢包率"甚至超过了 1.这种不正常的现象是因为,在某个时隙,由于门限骤然下降,此时按照 DT 的丢包策略,不仅丢弃了本时隙进入的组播信元,而且还丢弃了前面若干个时隙堆积起来尚未转发出去的组播信元.故此时出现丢弃的信元比进入的信元还多的情形,也就在形式上造成了"丢包率"大于 1 的现象.

在 $\alpha = 1$,*Multi Portion* = 0.3,$Fout_p$ = 0.2 的情形,实验得到的空闲存储器的大小如图3所示.图中显示了本

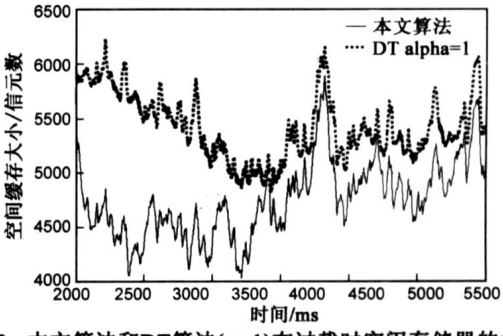

图3　本文算法和DT算法(α=1)在过载时空闲存储器的大小

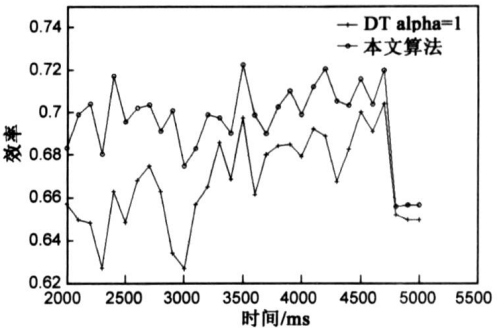

图4　本文算法和DT算法(α=1)在过载时的效率对比

文的算法能够在发生过载的情形了更充分地利用存储空间,这对于降低系统的丢包率是有正面作用的. 图 4 和图 5 别显示了运行期间按照前面建立的效率和公平性框架得到的归一化的效率和公平性曲线,对式(12)所定义的运行期效率指数,本文的算法较 DT 平均提高 15 %以上,而就式(13)所定义的运行期公平指数,本文的算法较 DT 平均提高 25 %以上.

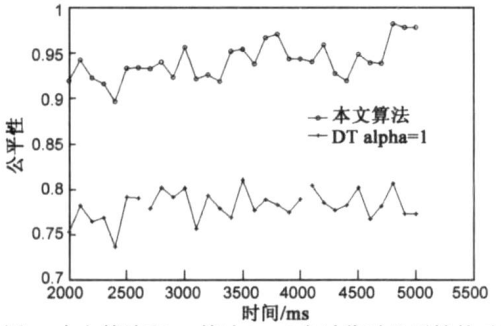

图5 本文算法和DT算法(α=1)在过载时公平性的对比

7 结论

双门限队列控制算法的主要思想是在平均的意义下让每个端口承担相同的流量压力,而在有突发流的片刻能彼此互相借用一定的存储空间. 这样的平衡能够使系统达到效率和公平性的最优. 本文利用经典的有效带宽理论,用有效流量的概念来作为"流量压力"的度量. 同时,本文提出了一个比较准确的"缓存换带宽"的估计式,能够根据 QoS 的需求合理估计当前缓存对流量的缓冲能力. 对门限的计算通过定时和触发两种机制一起实现,从而能实时地获取流量场景的当前值. 虽然本文的研究没有涉及 QoS[15,16],但是同传统的 DT 算法相比,利用有效流量的观点在 QoS 控制方面会有更大的优势. 我们将在未来的工作中把这个方法扩展到对 QoS 的支持上.

参考文献:

[1] H Jonathan Chao, Bin Liu, High Performance Switches and Routers[M]. New Jersey, US: John Wiley & Sons Inc, 2007.

[2] M Karol, M Hluchyj, S Morgan. Input versus output queueing on a space division packet switch[J]. IEEE Transactions on Communications, 1987, 35(12): 1347 - 1356.

[3] Sanjeev Kumar. The sliding-window packet switch: a new class of packet switch architecture with plural memory modules and decentralized control[J]. IEEE Journal On Selected Areas In Communications, 2003, 21(4): 656 - 673.

[4] Michael V Lau, Sam Shieh, Pei-Feng Wang, et al. Gigabit ethernet switches using a shared buffer architecture[J]. IEEE Communications Magazine, 2003, 41(12): 76 - 84.

[5] Ruey-Bin Yang, Yuan-Sun Chu, Cheng-Shong Wu, et al. Push out with virtual thresholds buffer management scheme in a shared buffer ATM switch international[J]. Journal of Network Management, 2003, 13(2): 147 - 154.

[6] Abhijit K Choudhury, Ellen L Hahne. Dynamic queue length thresholds for shared-memory packet switches[J]. IEEE/ACM Transactions on Networking, 1998, 6(2): 130 - 140.

[7] Tetsuo Zouta, Hiroshi Inai, Jiro Yamakita. Dynamic threshold control in an ATM switch with whared memory buffer[J]. Electronics and Communications in Japan, Part 1, 2000, 83(4): 1 - 11.

[8] Giuseppe Ascia, Vincenzo Catania, Daniela Panno. An Evolutionary management scheme in high performance packet switches[J]. IEEE/ACM Transactions on Networking, 2005, 13(2): 262 - 275.

[9] B Gaziz, Z Ghassemlooy. Dynamic buffer management using per-queue thresholds[J]. International Journal of Communication Systems, 2007, 20(5): 571 - 587.

[10] George Kesidis, Jean Walrand, Cheng-Shang Chang. Effective bandwidths for multiclass Markov fluids and other ATM sources[J]. IEEE/ACM Transactions on Networking, 1993, 1(4): 424 - 428.

[11] Costas Courcoubetis, Richard Weber. Buffer overflow asymptotics for a buffer handling many traffic sources[J]. Journal of Applied Probability, 1996, 33(3): 886 - 903.

[12] Costas Courcoubetis, Vasilios A Siris, George D Stamoulis. Application and evaluation of large deviation techniques for traffic engineering in broadband networks[J]. ACM SIGMETRICS Performance Evaluation Review, 1998, 26(1): 212 - 221.

[13] Cheng-Shang Chang. Performance Guarantees in Communication Networks[M]. London, UK: Springer Verlag, 2000.

[14] Mischa Schwartz. Broadband Integrated Networks[M]. New Jersey, US: Prentice Hall PTR, 1996.

[15] Alexander Kesselman, Zvi Lotker, Yishay Mansour, et al. Buffer overflow management in QoS switches[J]. SIAM J. COMPUT, 2004, 33(3): 563 - 583.

[16] Ellen L Hahne, Abhijit K Choudhury. Dynamic queue length thresholds for multiple loss priorities[J]. IEEE/ACM Transactions on Networking, 2002, 10(3): 368 - 380.

作者简介:

汪 洋 男,1976 年 1 月生于湖北武汉,目前为华中科技大学在站博士后,主要研究方向为下一代网络体系结构与高速交换系统.
E-mail: oxear@163.com

余少华 男,1962 年 9 月生于湖北武汉,武汉邮电科学研究院副院长,兼任华中科技大学教授、博士生导师. IEEE 会员,ITU-T 第 15 研究组副主席,主要研究方向为 IP 数据网和城域网.
E-mail: shyu@fhn.com.cn