

一种对数自适应队列调度算法

陈晓峰, 平玲娣, 陈健

(浙江大学 计算机科学与技术学院, 浙江 杭州 310027)

摘要: 针对 Internet2 研究计划提出的 Scavenger 服务(SS)的特点, 设计了一种对数自适应带权轮转队列调度算法. 该算法用滑动时间窗口算法统计活跃 SS 流数量, 以对数的规律自适应地调整带宽在 SS 流和尽力而为(BE)流之间的分配. 算法使用了分别对应 BE 流和 SS 流的两个虚拟队列, 出队时用 SS 流数量和轮转时间片数量对照表的方式来确定总轮转片数, 通过带权轮转的方式在队列间进行调度, 并加入了对 BE 队列的缓冲区保护. 仿真结果表明, 该算法在很好地保护 BE 流的同时, 为 SS 流提供更可靠的最小带宽保证, 和 Internet2 研究计划推荐的现有队列调度算法相比, 具有更好的性能和鲁棒性.

关键词: 队列调度算法; 对数自适应性; Scavenger 服务; 服务质量

中图分类号: TP393

文献标识码: A

文章编号: 1008-973X(2006)03-0381-06

Logarithmically adaptive queue scheduling algorithm

CHEN Xiao feng, PING Ling di, CHEN Jian

(College of Computer Science, Zhejiang University, Hangzhou 310027, China)

Abstract: To deal with particularity of scavenger service (SS) from Internet2 project, a logarithmically adaptive queue scheduling algorithm was presented. The number of active scavenger service flows was calculated by a time sliding window algorithm, and the bandwidth allocated for best effort (BE) flows and SS flows was accordingly tuned logarithmically. Two virtual queues corresponding to BE and SS flows were used, and total of slices was determined by a lookup table composed of flow number and slice number in a round robin cycle when scheduling the two queues. Protection for BE queue was added in the algorithm. Simulation results show that compared with current algorithms recommended by Internet2, the algorithm provides more reliable bottom bandwidth guarantee for scavenger service flows, protects BE flows well, and has better performance and robustness.

Key words: queue scheduling algorithm; logarithmic adaptation; scavenger service; quality of service (QoS)

互联网能迅速发展到目前的规模, 很大部分要归功于其 TCP/IP 协议簇尽力而为(best effort, BE)的设计思想, 这一做法大大降低了网络核心的复杂度和负担, 使其能够迅速扩展. 但是随着互联网规模的不断增大, 各种各样的网络服务争相涌现, 原有网络缺乏服务质量(quality of service, QoS)保证的缺点正日益突出.

传统的 IP 网络 QoS 研究关注如何提升重要流的服务质量, 试图保证这些流的数据包在整个路径上都得到优先于 BE 流的服务. 虽然业界和学术界在这方面已经做了大量研究工作并提出了多种实现 QoS 的体系结构(如 IETF 的 IntServ^[1]、Diff-Serv^[2])和协议(如 RSVP^[3]、MPLS^[4]等), 但是这些标准和协议始终未能在实际网络中得到广泛应用.

收稿日期: 2005-03-15.

浙江大学学报(工学版)网址: www.journals.zju.edu.cn/eng

基金项目: 浙江省科技计划资助项目(2005C21002); 华为高校科技基金资助项目(YJCB200402SSP).

作者简介: 陈晓峰(1975-), 男, 浙江温州人, 博士生, 从事网络 QoS、网络信息安全的研究. E-mail: cxf_wz@hotmail.com

究其原因,是因为现有的互联网极为庞大,要使 QoS 在互联网上实现,需要路径上的每个网络设备都能识别这些不同的标准和协议,才能使数据包得到必要的服务,只要有一处瓶颈设备不支持,整个路径的 QoS 就无法实现。

在 IETF 草案^[5,6]的基础上,美国大学先进网络研究联盟 UCAID 主导的下一代互联网 Internet2 研究计划^[7]正式提出了一种逆向思维的 QoS 技术: QBone Scavenger 服务^[8]。

本文提出了一种适用于 Scavenger 服务(SS)的对数自适应带权轮转队列调度算法(logarithmically adaptive weighted round robin queue, LAWRRQ)。它针对 SS 的特点,用滑动时间窗口算法统计活跃 SS 流数量,以对数的规律自适应地调整分配给 SS 流的带宽。算法使用了分别对应 BE 流和 SS 流的两个虚拟队列,通过带权轮转的方式在队列间进行调度,用 SS 流数量和轮转时间片数量对照表的方式来确定总轮转片数,并在缓冲区管理中加入了 BE 队列的保护。

1 Scavenger 服务及其队列调度

1.1 Scavenger 服务简介

SS 的基本思想是把那些不重要流的数据包加上标记(DSCP=001000B),网络边缘的接入路由器在处理包转发时,优先转发那些 BE 流,在网络带宽空闲时转发 SS 流,在降低 SS 流服务质量的同时,BE 流的服务质量就得到了提高;而且配置了 SS 的路由器在优先转发 BE 流的同时,给予 SS 流一个最小带宽保障,以避免标记为 SS 流的那些网络连接超时断开,并维持那些使用 SS 流的应用程序的响应时间不致变得太长而失去使用价值。如果网络带宽在保证了 BE 流后还有富余,SS 流照样可以使用剩下的所有带宽,避免了资源的浪费。

SS 的一个独特的优点在于它的实现不需要网络路径上所有的设备都支持,也不需要核心路由器维护每个流的状态信息。即使在路径上某些节点不能识别 2 种流而提供一样的服务,支持 SS 的那些节点对 SS 流已经起了节流阀的作用,带宽调节的目的照样能够实现。所以说 SS 是一种能够在现有 IP 网络中马上应用的 QoS 技术。

1.2 问题的提出

Internet2 研究计划在其构建的 QoS 试验床 QBone 和 Internet2 骨干网 Abilene 上实现了 SS 并推荐了 WRR、WFQ、MDRR 等几种队列调度机

制^[8],Stanislav 等人^[9,10]给出了网络设备配置范例,John 等人^[10,12]对 SS 进行了测试并给出了数据和结果。IETF 已经把这种 Lower Effort 的 QoS 技术作为一种新的 DiffServ 的每域行为(per domain behavior, PDB)写入 RFC 文档^[13]。

由于网络状态千变万化,SS 的路由器包转发调度算法的目标应当是在各种情况下满足 SS 流最低要求的同时最大限度减小对 BE 流的影响。通过对 WRR、WFQ、MDRR 等现有的队列调度算法^[14]实现 SS 的原理进行分析不难发现,它们对 SS 在网络繁忙的情况下所占的带宽比例都只能作静态的分配。由于网络繁忙时 SS 流可用带宽只占总带宽的一小部分,分配到每个 SS 流的带宽和 SS 流数量成反比,造成 SS 流容易陷入“饥饿”;如果分配给 SS 流较多的带宽则在 SS 流数量较小时浪费本应属于 BE 流的带宽。这些队列调度算法离上述目标还有很大差距,有必要针对 SS 的特点发展新的队列调度机制。

2 LAWRRQ 算法的设计思想和实现

LAWRRQ 队列调度算法最主要的目的是针对 SS 的特点,实现优先转发 BE 流数据包,同时把富余的带宽分配给 SS 流;在 BE 流繁忙时满足 SS 流的最低带宽要求,能够根据 SS 流数量的变化调整分配给 SS 流的最小保证带宽,同时最大限度减小对 BE 流的影响。LAWRRQ 队列机制拥有 2 个共享缓冲区的虚拟队列 vq_be 和 vq_ss ,数据包到达时根据 DSCP 标志将其区分为 BE 流和 SS 流数据包,分别进入 vq_be 和 vq_ss ;数据包转发时以带权轮转的方式在 2 个队列间选择一个,将队头数据包出队转发,因此算法包括队列间调度的机制和队列共享缓冲区的管理 2 部分的内容。

2.1 LAWRRQ 算法的队列调度机制

2.1.1 活跃 SS 流数量的计算 为了实现根据 SS 流数量的变化调整分配给 SS 流的最小保证带宽,队列调度算法首先要得到当前活跃的 SS 流数量。为此算法中设计了一个时间滑动窗口(time sliding window, TSW)算法,这个算法使用一个一维数组 *array* 来记录一个 SS 流的最近一个数据包到达时间,以流的源地址、端口和目的地址、端口组成 *flow_tag* 通过散列算法来寻址数组单元。这部分的伪代码如下:

```
Upon each packet of SS flows arrival:
array[hash(flow_tag)] = t_current;
```

```

if (t_current - t_last >= t_interval) {
    t_last = t_last + t_interval;
    flow_count = 0;
    for (i = 0; i < ARRAY_SIZE; i++) {
        if (array[i] - t_current < t_window)
            flow_count++;
    }
    total_slices = lookup_table(flow_count); /*
adjusts bandwidth allocation */
}

```

从伪代码中可以看到, 算法使用了 2 个时间阈值: $t_interval$ 和 t_window . 算法每隔 $t_interval$ 统计一次当前活跃的 SS 流个数而不是在每个包到达时都统计, 大大减小了运算量的开销. 阈值 $t_interval$ 的取值需要在算法对流数量变化的反应速度和运算开销之间折衷, 本文中取 0.1 s. 窗口时间 t_window 的取值应当考虑到一个流的数据包在短时间内没有出现并不意味着这个连接已经断开, 比如可能是在网络某处发生了丢包. 过小的 t_window 可能造成 $flow_count$ 数值抖动, 过大则使算法对流数量变化反应缓慢, 本文中取 $t_window = 5$ s. 这样这个算法对于新 SS 流的增加能够在 0.1 s 内作出反应, 对老 SS 流的消失在 5 s 后去除它的统计

通常要维护流状态的包调度算法在流数量很大时状态表变得过大, 往往由于无法承受维护开销而变得不实用, 但是 LAWRRQ 采取的以对数规律依据 SS 流数量调整带宽分配的方式避免了这一问题. LAWRRQ 选用相对较小的数组大小和相应的散列函数, 并且对散列冲突不加处理, 从而限制了状态表的大小. 多个流可能会共用一个单元记录最近包到达的时间, 统计的流数量比实际少. 在流数量小的情况下发生散列冲突而导致误差的可能性很小; 在流数量较大的情况下, 由于对数函数自变量增大到一定值时函数值上升缓慢的特性, 算法对误差的敏感性降低. 对于 LAWRRQ 统计活跃 SS 流数量来适当调节给 SS 聚合流的带宽分配的目的本身来讲, 也允许少量的误差.

2.1.2 分配给 SS 流的轮转片数比例的调整 得到 SS 流的数量 $flow_count$ 后算法还需要调整分配给 SS 流的轮转片数. 一个直观的方法是在 $flow_count$ 增加时, 增加分配给 SS 流的轮转片数, 但这并不是一个好的方法, 因为连续给 SS 流分配多个轮转片将会使网络 QoS 重要指标之一抖动大大恶化, 而要做到将这些轮转片平均分配在整个轮转周

期内是很困难的. 所以本文的 LAWRRQ 队列调度算法采用了保持分配给 SS 流的时间片为 1 而减小总轮转片数 $total_slices$ 的做法.

SS 流数量和 SS 流带宽增长比例的关系是队列调度算法的重点: 如果增长过快, 则会影响到 BE 流; 如果增长过小, SS 流还是易和现有的队列管理算法一样陷入饥饿. LAWRRQ 队列调度算法采用了对数函数 $f(x) = \log_a x$ 来确定这一比例, 它的增长曲线能够较好地适应 SS 的特点, 不同的对数底具有不同的上升速率, 适用于不同规模的应用场合. 本文采用了以欧拉常数 e 为底的自然对数来计算.

设 f 为 SS 流数量, s 为总轮转片数, 令 $\ln f \times s = C$ 可得出总轮转片数 $s = C / \ln f$, 其中 C 为一设定常数, C 实际上也可化入对数函数的底数内, 这样分子就为 1, 则 BE 流和 SS 流分配到的带宽比例表达式分别为

$$P_{vq,be} = \frac{s-1}{s} = \left[\frac{C}{\ln f} - 1 \right] \Bigg/ \frac{C}{\ln f} = 1 - \frac{\ln f}{C}, \quad (1)$$

$$P_{vq,s} = \frac{1}{s} = \frac{\ln f}{C}. \quad (2)$$

为了避免路由器运行时进行复杂的对数计算, LAWRRQ 通过事先计算各个自然对数值的倒数并在算法启动时乘以设定的 C 后取整, 形成一张表, 运行时根据第 2.1.1 节得到的 SS 流数量 f 查这张表, 即可得到对应的 s 值. 这个表的一部分如表 1 所示.

常数 C 决定了在一定 f 值下分配给 SS 流的保证带宽占链路总带宽的比例, 可以根据应用环境的要求来选择. 这里及后面的仿真中取 $C = 69.4$, 使 $f = 2$ 时 $s = 100$, 即所占带宽为总带宽的 1%. 在 $f = 1$ 时由于 $\ln 1 = 0$ 并无意义, 需对 s 直接取值, 本文中取对应的 $s = 150$. 由于自然对数的特性, 在 f 值增加到一定值时, s 的数值变化缓慢, 因此该表并不需要很多的表项. 上面程序中的 $lookup_table()$ 函数即是查表函数.

表 1 SS 流数量和总轮转片数对照表

Tab. 1 SS flow number and corresponding total slices in cycle

f	s	f	s
1	150	7	35
2	100	8	33
3	63	9	31
4	50	10	30
5	43	⋮	⋮
6	38		

2.1.3 出队队列的选择 在确定总轮转片数后, LA WRRQ 使用了一个轮转片计数变量 deq_turn 在出队的轮转算法中进行队列选择, 当 $deq_turn=0$ 时选中 vq_ss , 当 $deq_turn>0$ 时选中 vq_be , 算法如下:

Upon interface free and queue not empty:

```
if ( $deq\_turn > 0$ ) {
    if ( $vq\_be$  not empty){
        dequeue  $vq\_be$ ;
         $deq\_turn++$ ;
        if ( $deq\_turn \geq total\_slices$ )
             $deq\_turn=0$ ; /*if  $deu\_turn$  reaches
            to  $total\_slices$ , begin next cycle */
    }
    else {
        dequeue  $vq\_ss$ ; /*if  $vq\_be$  is empty, de
        queue  $vq\_ss$  */
         $deq\_turn=1$ ;
    }
}
else {
     $deq\_turn=1$ ;
    if ( $vq\_ss$  not empty)
        dequeue  $vq\_ss$ ;
    else
        dequeue  $vq\_be$ ;
}
```

可以看出, SS 流实际获得的带宽还和它的平均包大小有关, 对于避免标记为 SS 流的那些网络连接超时断开, 并维持使用 SS 流的应用程序有一个合理的响应时间的目的来说, 给予 SS 流一个最低包速率保障比最低带宽保障更合适。

2.2 LAWRRQ的缓冲区管理

缓冲区在分别对应 BE 流和 SS 流的 2 个队列间的分配可以有两种方法: 1) 2 个队列分别使用各自独立的存储空间, 但是在 2 种流繁忙程度不均时会浪费缓冲区资源; 2) 2 个虚拟队列共用一个物理缓冲区, 在入队时以它们的长度 $L_vq_be+L_vq_ss >= BUFFER_SIZE$ 作为判断队列是否已满的条件, 使资源得到充分利用. LAWRRQ 采用的是后者。

通过实验发现只做缓冲区满的判断还不够, 在 SS 流增加到一定程度时, BE 流有时会发生剧烈抖动, 这一现象在本文第 3 章的仿真中可以看到. 通过队列跟踪可以发现造成这种现象的原因是 SS 流和 BE 流的数据包在入队的几率上是一致的, 而在出队

时由于 SS 的特点, SS 流的包离开的几率远小于 BE 流的包, 因此 SS 流数据包有在缓冲区中堆积的倾向, 容易造成 BE 流连续大量丢包, 因此 LAWRRQ 引入了单向保护 BE 流的算法。

单向保护算法在 BE 流数据包到达且缓冲区满时, 有 2 种情况: 1) 如果 SS 流的队长大于一个阈值 M , 则丢弃 SS 流队列队尾的包, 腾出空间, 将到达数据包加入 BE 流队列; 2) 如果 SS 流队长小于等于阈值 M , 丢弃该 BE 流包. 入队算法中 BE 流和 SS 流对缓冲区的共享比例是动态竞争的: 1) 在只有一种流繁忙时, 繁忙的流将占用大部分缓冲区; 2) 在 2 种流都繁忙而缓冲区满时保护算法才起作用. 这里取 $M=BUFFER_SIZE/2$, 使得 2 种流都繁忙时, BE 流至少能获得 $BUFFER_SIZE$ 的一半, 超出部分再和 SS 流竞争其余的缓冲区空间, 因此这种保护是单向的, 这也符合 SS 的定位。

在虚拟队列内部的管理可以采用 Drop Tail、随机早期检测 (random early detection, RED) 等常用的缓冲区管理算法, 在本文中采用简单的 Drop Tail.

结合了 BE 流保护, 采用 Drop Tail 的入队算法的伪代码如下:

```
Upon each packet arrival:
if (packet is belong to SS flow){
    if ( $L\_vq\_be+L\_vq\_ss < BUFFER\_SIZE$ )
        enqueue packet to  $vq\_ss$ ;
    else
        drop packet;
}
else{
    if ( $L\_vq\_be+L\_vq\_ss < BUFFER\_SIZE$ )
        enqueue packet to  $vq\_be$ ;
    else if ( $L\_vq\_ss > BUFFER\_SIZE/2$ ){
        drop tail of  $vq\_ss$ ;
        enqueue packet to  $vq\_be$ ;
    }
    else
        drop packet;
}
```

3 仿真与分析

3.1 模拟环境

实验使用美国劳伦斯伯克利国家实验室 LBNL 开发的 NS Version 2.26^[15] 作为仿真工具, 用 C++

实现了 LAWRRQ 队列管理算法. 采用的网络拓扑结构如图 1 所示.

图中 R1 和 R2 是 2 个路由器, 它们由一条带宽 10 Mbps、延时 6 ms 的双向链路连接, LAWRRQ 算法就配置在 R1 和 R2 连接的接口上, 采用的缓冲区大小是 128 个包. 除了这条链路外, 其他所有的链路都是带宽 4 Mbps、延时 3 ms, 使用普通的先进先出的 Drop Tail 队列. S1 到 S10 是 10 个 SS 数据源, 它们分别和对面的 DS1 到 DS10 进行数据传送; B1 到 B4 是 4 个 BE 数据源, 它们分别和对面的 DB1 到 DB4 进行数据传送. 所有的数据连接使用 TCP/Reno 协议, 分组大小为 NS2 默认值 1000 字节, 以 FTP 作为数据源.

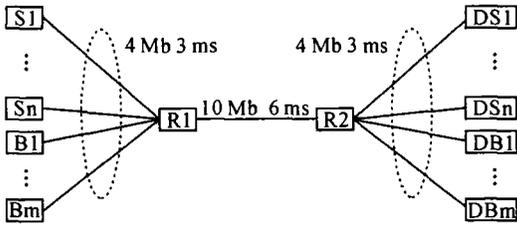


图 1 网络拓扑结构

Fig. 1 Network topology

3.2 Scavenger 服务的实现

为了验证 LAWRRQ 算法对 SS 的实现, 实验采取如下的模拟过程: 在模拟开始时启动一个 BE 数据源 B1 进行数据传送, 在 2.0 s 时 SS 数据源以间隔 2 s 依次启动, 直到所有 SS 数据源启动完毕, 然后在 22 s 时剩下的 3 个 BE 数据源以 2 s 的间隔启动, 在 30 s 时整个仿真结束. 在这个实验过程中重点观察网络中 2 种流各自的聚合带宽. 实验结果在图 2 中给出.

从实验结果中可以看到, 在 BE 流带宽较小时, SS 流能够充分利用网络空余的带宽, SS 流数量的增加没有影响 BE 流获得它应有的带宽; B2 数据源启动后, SS 流开始自动让出资源, 随着余下的 BE

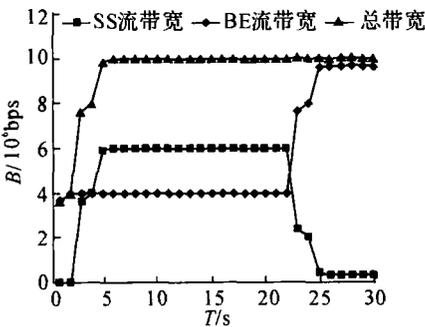


图 2 Scavenger 服务的实现

Fig. 2 Realization of scavenger service

数据源启动, SS 流最终只获得了一个很小的保证带宽, 而由 BE 流占用了绝大部分带宽. 可以看出 SS 得到了很好的实现.

3.3 对 BE 流的保护

为了更好地观察算法对 BE 流的保护, 在这部分的仿真中把总的缓冲区减小为 64 个包. 模拟执行以下过程: 在仿真开始时启动 4 个 BE 数据源, 在 5 s 后所有 SS 数据源以 0.1 s 的间隔依次启动, 在 25 s 时结束仿真. 在这个模拟中主要观察一个 BE 流(B1 数据源)的带宽和丢包情况. 首先在只以 $L_{vq_be} + L_{vq_ss} \geq BUFFER_SIZE$ 作为判断队列满条件的情况下执行了模拟, 结果如图 3 所示.

可以看到在 BE 流繁忙的情况下, SS 流个数增加到一定值时导致 BE 流出现连续的丢包, 伴随着 BE 流带宽的剧烈抖动. 如第 2.2 节中所述, 造成这种现象的原因是 SS 流数据包有在缓冲区中堆积的倾向, 在 SS 流数量较多且 TCP 窗口值逐渐放大时虚拟队列 vq_ss 占用了绝大多数缓冲区, 从而影响到 BE 流. 在加入了第 2.2 节中对 BE 流的保护算法后重复以上模拟, 得到的结果如图 4 所示.

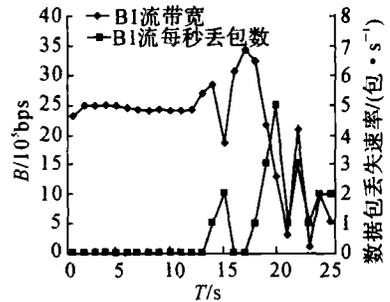


图 3 只作队列满判断的仿真结果

Fig. 3 Only test buffer full when enqueueing

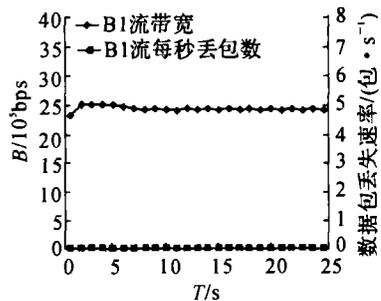


图 4 加入 BE 流保护算法后的仿真结果

Fig. 4 BE flows protecting algorithm added

3.4 BE 流满载时 LAWRRQ 算法对 Scavenger 服务流数量变化的反应

在这个模拟中把模拟过程设定如下: 在仿真开始时启动全部 4 个 BE 数据源, 在 2 s 时启动 S1, 余下的 SS 数据源以 2 s 的间隔启动, 在 20 s 时最后一

个 S10 数据源启动, 继续运行 5 s 后结束仿真. 在这个模拟中主要观察所有 SS 流所占的总带宽的变化. 为了方便比较, 首先选取一个静态配置 SS 流最小保证带宽的队列调度算法进行上述模拟.

图 5 为 Internet2 研究划推荐用于实现 SS 的队列调度机制之一 WRR 算法的仿真结果, SS 流的总的最小保证带宽配置为连接 R1、R2 路由器链路带宽的 1.5% 即 150 Kbps. 为了便于观察, 图中对 SS 流带宽曲线的纵坐标进行了放大. 可以看到 BE 流满载时, 在活动 SS 流数量从 1 逐渐增加到 10 的过程中, 它们获得的总带宽始终保持不变.

图 6 为 LA WRRQ 队列调度算法的仿真结果. 仿真结果表明, 随着活动 SS 流个数的增加, SS 流获得的总带宽大致以对数曲线上升, 符合算法设计的目标. 和图 5 中的结果相比, LA WRRQ 实现了 SS 流最小保证带宽随着活动 SS 流个数自适应调整: 在 SS 流数量很少时, 分配较少的保证带宽, 尽量将有限的资源分配给 BE 流, 提高了 BE 流的性能; 在 SS 流数量较多时, 适当增加分配的最小保证带宽, 使 SS 流不至于陷入饥饿, 从而具有更好的鲁棒性.

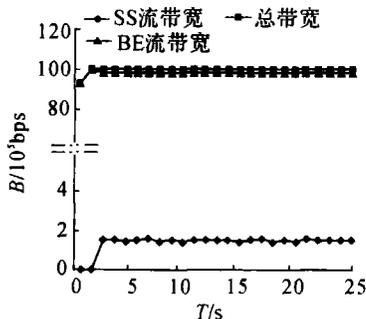


图 5 WRR 算法的仿真结果

Fig. 5 Result of WRR

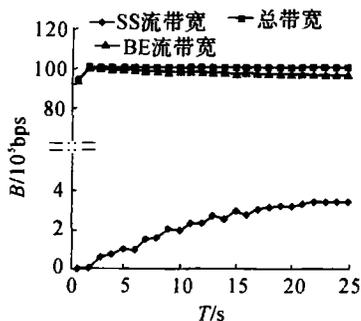


图 6 LA WRRQ 算法的仿真结果

Fig. 6 Result of LA WRRQ

4 结 语

本文提出了一种适用于 SS 的 LA WRRQ 对数自适应队列调度机制. 算法针对 SS 的特点, 使用了

SS 流数量和总轮转片数对照表的方式来克服计算复杂度, 根据活跃 SS 流数量的变化自动以对数的规律调整参数, 并引入保护 BE 流的算法来解决 SS 数据包在缓冲区中堆积的问题. 分析和仿真结果表明, 它能针对 SS 的特点, 根据网络流数量的变化自动调整参数, 在很好地保护 BE 流的同时为 SS 流提供更可靠的最小带宽保证, 有更好的性能和鲁棒性.

如何在 SS 队列管理算法中引入高性能低代价的机制来平衡带宽在 SS 流内部的分配是下一步要做的工作.

参考文献 (References):

- [1] BRADEN R, CLARK D, SHENKERS S. RFC1633, Integrated services in the internet architecture: an overview [EB/OL]. [1994 06 10]. www.rfc-editor.org/rfc/rfc1633.txt.
- [2] BLAKE S, BLACK D, CARLSON M, et al. RFC2475, An Architecture for Differentiated Service [EB/OL]. [1998 12 20]. www.rfc-editor.org/rfc/rfc2475.txt.
- [3] WROCLAWSKI J. RFC 2210, The use of RSVP with IETF integrated services [EB/OL]. [1997 09 10]. www.rfc-editor.org/rfc/rfc2210.txt.
- [4] LE FAUCHEUR F, WU L, DAVIE B, DAVARIS et al. RFC3270, Multi Protocol label switching (MPLS) support of Differentiated Services [EB/OL]. [2002 05 10]. www.rfc-editor.org/rfc/rfc3270.txt.
- [5] BLESS R, WEHRLE K. A lower than Best Effort per hop behavior [EB/OL]. [1999 12 20]. http://quimby.gnus.org/internet-drafts/draft-bless-diffserv-lbe-phb-00.txt. Work in Progress.
- [6] CAREPENTER B, NEICHOLS K. A Bulk Handling Per Domain Behavior for Differentiated Services [EB/OL]. [2001 01 15]. http://www.ietf.org/proceedings/01aug/1D/draft-ietf-diffserv-pdb-bh-02.txt. Work in Progress.
- [7] Internet2 Home [EB/OL]. [2001 01 15]. http://www.internet2.edu.
- [8] QBone Scavenger Service (QBSS) [EB/OL]. [2004 10 17]. http://QBone.internet2.edu/qbss/.
- [9] STANISLAV S. Configuring QBSS on a juniper [EB/OL]. [2001 05 10]. http://archives.internet2.edu/guest/archives/i2ss-dt/log200105/msg00024.html.
- [10] SLAC QBSS Testbed [EB/OL]. [2004 05 10]. http://www.iepm.slac.stanford.edu/monitoring/qbss/qbss.html 2004-05.
- [11] Juniper QBSS experiment [EB/OL]. [2003 12 15]. http://www.transpac.org/qbss.html/.

(下转第 413 页)

阀的电磁铁在额定电压下连续通、断电达万次后,检查被试阀主要零件,无损坏和异常磨损现象。

图 13(a)、(b)分别为入口压力为 12、14 MPa 时,万次寿命实验中 5 s 内连续换向的响应曲线。由图可知阀的动态响应特性基本一致,换向灵敏可靠。

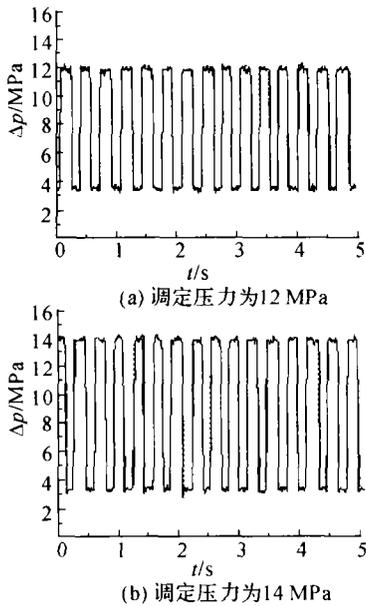


图 13 纯水换向阀寿命试验曲线

Fig. 13 Life testing curve of water hydraulic directional valve

4 结论

(1) 该纯水换向阀在额定工况下压力损失小,换向过程中,速度快而且冲击较小,动态性能较好。寿命试验中,换向阀工作正常,换向可靠,其性能与现有同类油压换向阀的性能相当。

(2) 针对水介质易泄漏的问题,用直接密封取代传统的间隙密封;针对换向阀本身存在的换向快速性与换向冲击之间的矛盾,对主阀芯在主阀套上的压紧力、阀口阻尼器与敏感腔匹配进行设计,这些措施为纯水换向阀的成功研制提供了保障。

(3) 仿真结果表明,主阀芯与主阀套形成的喷口

结构会造成阀的压力损失;根据流场形态,通过对喷口结构进行优化,尽量减小旋涡的形成是降低阀的压力损失,提高阀的性能的有效途径。

参考文献(References):

- [1] 杨华勇,周华. 水液压技术的研究现状与发展趋势[J]. 中国机械工程, 2000, 11(2): 1430-1433.
YANG Hua yong, ZHOU Hua. Research condition and development direction of water hydraulics [J]. **China Mechanical Engineering** 2000, 11(2): 1430-1433.
- [2] 宋鸿尧,丁鸿尧. 液压阀设计与计算[M]. 机械工业出版社, 1982.
- [3] BACKE W. Water or Oil hydraulics in the future[C] // **Proceedings of 6th Scandinavian International Conference on Fluid Power, Tampere, Finland**; Tampere University of Technology, 1999: 51-64.
- [4] 杨署东,李壮云. 水压传动的发展及其关键基础技术[J]. 机床与液压, 2000(5): 6-9.
YANG Shu dong, LI Zhuang yun. The development and the key basal technology of water transmission [J]. **Machine Tool & Hydraulics** 2000(5): 6-9.
- [5] 高殿荣,王益群. 液压锥阀流场的有限元解析[J]. 机床与液压, 2000(2): 12-14.
GAO Dian rong, WANG Yi qun. Finite element investigation of flow in hydraulic cone [J]. **Machine Tool & Hydraulics** 2000(2): 12-14.
- [6] 冀宏,傅新,杨华勇. 内流道形状对溢流阀气穴噪声影响的研究[J]. 机械工程学报, 2002, 38(8): 19-22.
JI Hong, FU Xin, YANG Hua yong. Effects of internal structure on cavitating noise of hydraulic relief valve [J]. **Chinese Journal of Mechanical Engineering** 2002, 38(8): 19-22.
- [7] GAO Hong, FU Xin, YANG Hua yong, et al. Numerical investigation of cavitating flow behind the cone of a poppet valve in water hydraulic system [J]. **Journal of Zhejiang University: SCIENCE** 2002, 3(4): 395-400.
- [8] PAOLUZZI R. Stationary and dynamic analysis of a water relief valve [C] // **Proceedings of the 4th JHPS International Symposium on Fluid Power, Tokyo**; Tokyo Institute of Technology, 1999: 561-566.

(上接第 386 页)

- [12] SHALUNOV S. Testing QBSS config at advanced.org; Good [EB/OL]. [2004-05-01]. http://archives.internet2.edu/guest/archives/i2ss_dt/log200105/msg00000.html.
- [13] BLESS R, NICHOLS K, WEHRLE K. RFC3662, A lower effort Per Domain behavior (PDB) for differentiated services [EB/OL]. [2003-12-01]. www.rfc-editor.org/rfc/rfc3662.txt.

- [14] VEGESNA S. **IP Quality of Service (Cisco Networking Fundamentals)** [M]. San Jose, California; Cisco Press, 23, 2001-01.
- [15] The Network Simulator ns 2 [EB/OL]. [2004-10-01]. <http://www.isi.edu/nsnam/ns/index.html>.