

# 一种用于分组交换机的缓存管理算法

胡冰, 李乐民

(电子科技大学 宽带光纤传输与通信网技术教育部重点实验室 成都 610054)

**【摘要】**提出了一种用于分组交换机中的缓存管理算法DT+SMA, 该算法把DT和SMA结合起来, 具有两者的优点。通过理论分析和仿真, 得知在单优先级情况下DT+SMA比SMA的公平性更好, 在多优先级情况下SMA比DT更易于实现, 而且同时也能获得和DT相近的性能。

**关键词** 缓存管理; 仿真; 共享存储器; 分组交换机

中图分类号 TN 915.05 文献标识码 A

## A New Buffer Management Scheme

Hu Bing, Li Lemin

(Key Laboratory of Broadband Optical Fiber Transmission and Communication Networks UEST of China, Ministry of Education Chengdu 610054)

**Abstract** In this paper a new buffer management scheme called DT+SMA is presented. This scheme combines sharing with minimum allocation (SMA) and dynamic threshold (DT) arithmetic, and we find it has the benefits of both DT and SMA. Through analysis the scheme in theory and simulation we can see that actually DT+SMA is better than SMA in fairness in single priority, and is easier to apply in multiple priorities model than DT when achieving the similar performance with DT.

**Key words** buffer management; simulation; shared memory; packet switch

近年来, 通信向分组化发展, 在通信网络中需要有分组交换机。为了避免冲突, 分组交换机采用了输入输出排队等结构。将输出排队的各队列中的缓冲存储器合在一起, 就成为共享缓存交换机, 如图1所示。由于在给定的丢失率的条件下共享缓存交换机所用的存储器容量最小, 所以共享缓存交换机得到了广泛应用。由于在实际中大部分交换机所交换的数据都为定长, 所以考虑定长数据的交换情况。一个高效率的缓存管理算法对于共享存储器交换机极为重要, 缓存管理策略将会直接影响到一个交换机的性能。有两种经典的缓存管理算法: 最小分配共享(Sharing with Minimum Allocation, SMA)和动态门限(Dynamic Threshold, DT)<sup>[1, 2]</sup>。

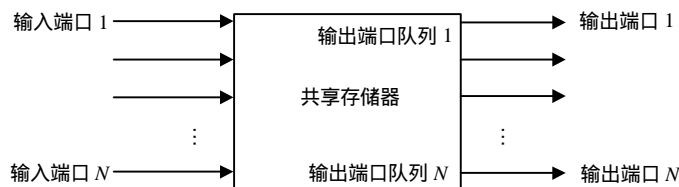


图1 共享缓存交换机模型

DT算法的核心是: 在任何一个瞬时, 输出队列的门限是交换机中当前未使用的缓存器大小的函数, 当输出端口的队列长度等于或者超过当前的门限, 该端口就会被锁定不会接收新的数据。在 $t$ 时刻,  $T(t)$ 是受控

收稿日期: 2004-07-06

作者简介: 胡冰(1980-), 男, 硕士生, 主要从事交换机中缓存管理算法方面的研究。

的门限,  $Q_i(t)$  是第  $i$  个队列的长度, 在  $t$  时刻如果  $Q_i(t) > T(t)$ , 端口  $i$  就会被锁定, 直到队列长度降低至低于门限或门限提高到高于队列长度 ( $Q_i(t) < T(t)$ ) 之前, 所有到达这个端口的信元都会被丢弃。

在 SMA 算法中, 存储器被分成了两个部分  $B_1$ 、 $B_2$ 。  $B_1$  是用于交换机中的每个端口队列的非共享空间, 每一个队列都有一个只能被它自己使用的最小空间, 当某个队列的长度超过分配给它的专用空间时, 这个队列才开始占用  $B_2$  的空间, 而  $B_2$  在所有的端口队列中完全共享。

DT 具有很好的灵活性和很高的效率, 但是当 DT 用于多优先级的时候<sup>[3]</sup>, 就变得比较复杂, 得为每一个优先级计算该优先级的总共有多少的队列长度。 SMA 在用于多优先级的时候很容易实现, 只要为不同的优先级队列提供不同的自己使用最小空间就可以实现, 但 SMA 的公平性(缓存管理中的公平性是指, 在相同优先级下, 带宽应该在各个端口队列公平的分配)却无法很好的保证。 本文提出一种新的缓存管理算法: DT+SMA。

## 1 DT+SMA 算法

在 DT+SMA 队列中, 缓存器也被分为两部分:  $B_1$ 、 $B_2$ 。

$$B = B_1 + B_2 \quad (1)$$

在数据占用  $B_1$  的时候 DT+SMA 算法和 SMA 完全相同, 可是在数据开始占用  $B_2$  的时候中, 要在  $B_2$  中使用 DT 算法, 设  $q_i(t)$  为每个队列超过专用的最小空间的队列长度, 既  $q_i(t) = Q_i(t) - V_i$ , 其中  $V_i$  是为每个队列分配的专用最小空间

$$T(t) = a [B_2 - \sum_{i=1}^N q_i(t)] \quad (2)$$

在这种方式下, DT 和 SMA 被结合起来, 得到了一种具有 DT 和 SMA 优点的缓存存储器管理算法。

## 2 性能比较和仿真结果

通过仿真来比较提出的算法和其他的算法的性能。 选择一个 8 端口共享存储器的输出排队交换机作为仿真的模型, 每一个端口连接着一个单一的突发 ON/OFF 模型, 每个 ON/OFF 模型的平均发送速率设为 8 M/s, 其中 ON 期持续时间服从均值为 0.001 的重尾(Pareto)分布

$$f(x) = \frac{ck^c}{x^{c+1}} \quad k \quad x < \infty \quad (3)$$

通过改变  $c$  可以改变函数的曲线形状也就是改变了源的突发性, 而通过改变  $k$  可以改变函数曲线的位置。 Pareto 分布的均值为

$$E(x) = \frac{ck}{c-1} \quad c < 1 \quad (4)$$

在不改变源平均速率的前提下(即保证 Pareto 分布的均值不变), 为了得到丢失率随  $c$  变化(即源突发性变化)的曲线, 就得必须相应的改变  $k$  值。 初始化时  $c = 1.9$ , 由式(4)可得到  $k = 0.00474$ , 不断地变化  $c$ , 同时由式(4)求出相应的变化  $k$  值, 就可得到丢失率随源突发性变化的曲线。 OFF 期的持续时间是服从均值为 0.001 的指数分布, 整个的缓存存储器大小为 160 个信元。 根据以上的参数设置, 缓冲存储器很快就能达到稳态, 所以把交换数据的虚拟时间设为 0.5 s 进行仿真即可。

### 2.1 单优先级

在单优先级的条件下, 通过仿真来证明 DT+SMA 的公平性要比 SMA 好。 把 DT+SMA 和 SMA 中的  $B_1$  都设为 80 个信元,  $B_2$  为 80 个信元。  $B_1$  在 8 个端口内均分, 每个端口的专用最小空间为 10 个信元。 为简单起见, 设 DT+SMA 中的参数  $a$  为 1, 则可以由式(2)得到 DT+SMA 的门限

$$T(t) = B_2 - \sum_{i=1}^N q_i(t) \quad (5)$$

为了检验算法的公平性, 令一个端口传输负载非常大, 只要根据其他正常端口的丢失率大小就可以得到算法的公平与否, 所以令端口 1 传输负载很大, 且端口 1 的 ON 期持续时间的 Pareto 分布的  $c$  参数不断变大, 使端口 1 的数据的突发性越来越小。 在这种条件下发现,  $r$  为丢失率如图 2 所示。 DT+SMA 算法中的正常端口的丢失率比 SMA 要低, 也就证明了 SMA+DT 比 SMA 的公平性要好。

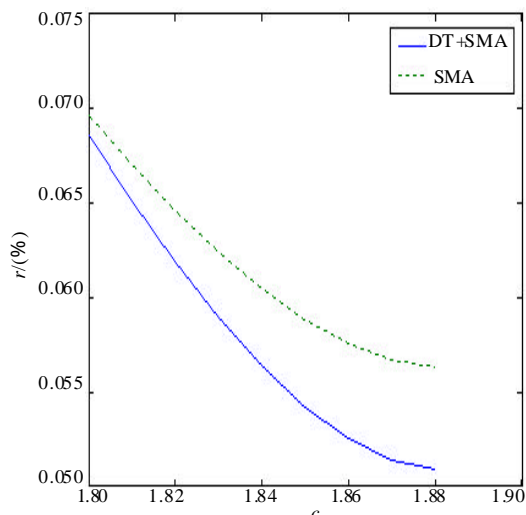


图2 端口7队列的丢失率变化曲线

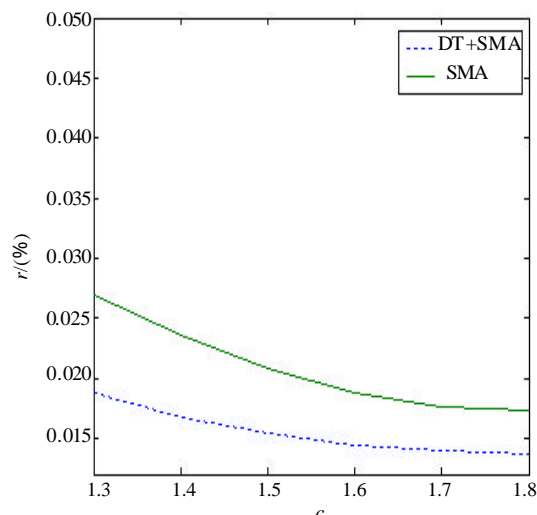


图3 高优先级端口队列的丢失率变化曲线

## 2.2 多优先级

在多优先级的情况下，DT根据文献[3]性能最好的是

$$Q_p^i(t) < a_p(B - Q(t)) \quad (6)$$

式中  $Q_p^i(t)$  是在  $t$  时刻端口  $i$  中的  $p$  优先级的信元的个数， $a_p$  是优先级  $p$  的参数。当  $i$  端口的  $p$  优先级不满足式(6)的时候， $i$  端口对  $p$  优先级的数据进行锁定，直到  $i$  端口的  $p$  优先级重新满足式(6)之前所有到达  $i$  端口的  $p$  优先级的数据都会被丢弃。在不同端口不同的优先级也是如此，各个端口的各个优先级只需满足自己参数下的式(6)即可。仿真的时候设有2个优先级(1个高优先级7个低优先级)，对于高优先级的  $a_p=2$ ，而对于低优先级  $a_p=1$ 。可以看出DT在用于多优先级时必须计算  $Q_p^i(t)$ ，即在一个端口的同一优先级信元的个数，在高速交换机中不易于实现的。

而对于多优先级下的DT+SMA，只需在  $B_1$  部分应用多优先级策略，而在  $B_2$  部分用DT单优先级算法即可对整体实现多优先级算法。在  $B_1$  部分只需要令不同的优先级分配到不同的专用最小空间，设有2个优先级(1个高优先级7个低优先级)，设高优先级的专用最小空间为24个信元，低优先级为8个信元。这样在多优先级的高速交换机中应用DT+SMA，仅需分配到不同的专用最小空间即可，可见DT+SMA比DT在多优先级条件下，更易实现。而DT和DT+SMA的性能比较如图3所示，DT和DT+SMA中高优先级端口队列的丢失率相差不多。

## 3 结 论

通过仿真发现，本文提出的DT+SMA算法在单优先级的情况下确实比SMA具有更好的公平性，而在多优先级的情况下，能获得和DT相近的性能，且比DT易于实现。所以DT+SMA具有DT和SMA两者的优点，在高速交换机中的实用性更强。

## 参 考 文 献

- [1] Kamoun F, Kleinrock L. Analysis of shared finite storage in a computer network node environment under general traffic condition[J]. IEEE Transactions on Communication, 1980, 28(7): 992-1 003
- [2] Choudhury A K, Hahne E L. Dynamic queue length threshold for shared memory packet switches[J]. IEEE/ACM Transaction on Networking, 1998, 6(2): 130-140
- [3] Hahne E L, Choudhury A K. Dynamic queue length threshold for multiple loss priorities[J]. IEEE/ACM Transaction on Networking, 2002, 10(3): 368-380