

分组公平排队算法的仿真研究*

涂晓东** 李乐民

(电子科技大学宽带光纤传输与通信系统技术国家重点实验室 成都 610054)

【摘要】 对 ATM 复接器和输出缓存 ATM 交换机采用各种分组公平排队算法情况下的信元时延进行了仿真。仿真结果表明, SCFQ 对连接之间的隔离性比其他算法差; 若一个 CBR 速率组内没有流量超过其预约带宽的连接, 则用 Per_Rate_Grouping 方式实现 WF²Q+和不用这种实现方式, 从最大时延和平均时延来看, 相差不大。提出了一个调度 CBR、VBR、ABR 和 UBR 的调度器结构, 既能满足 CBR 对时延的严格要求, 又能保证 VBR、ABR 和 UBR 的吞吐率, 同时还具有较低的实现复杂性。

关键词 分组公平排队算法; WF²Q+算法; Per_Rate_Grouping 实现方式; 固定比特率业务
中图分类号 TN919.4

ATM 是信息高速公路的关键技术^[1,2]。各种多媒体应用, 如 IP Phone、视频会议等要求 ATM 复接器、ATM 交换机能够提供 QoS(quality of service)服务。从80年代末以来, 国际上对分组公平排队(packet fair queuing—PFQ)进行了大量的研究。PFQ 能够保证连接的预约带宽、最大端到端时延以及时延抖动, 是实现 QoS 的关键技术。目前已有 WFQ(weighted fair queuing)、WF²Q(worst-case fair fair queuing)、WF²Q+、SPFQ(start potential fair queuing)、virtual clock、SCFQ(self clocked fair queuing)等多种 PFQ 方案^[3-9], 其中 WF²Q+性能较好。文献[10]提出了用 per_rate_grouping 方式实现 WF²Q+, 实现复杂性只与支持的速率个数有关, 而与连接的个数无关。但是连接的最大时延和时延抖动将增加 $\frac{1}{r}$, 这种性质对于 CBR 业务是不利的。本文用自编的 PFQ 仿真软件 fqsim 对 ATM 复接器(也可以是输出缓存 ATM 交换机)采用各种排队算法情况下的信元时延进行了仿真, 比较了各种情况下的仿真结果, 给出了一种调度器的结构, 既能保证 CBR 连接的最大时延, 又能满足 VBR、ABR 和 UBR 的要求, 同时还具有较低的实现复杂性。

1 仿真对象的设置

仿真对象是如图1所示的 ATM 复接器, 有10个连接经过这个复接器, 连接的信源特性及权重设置如表1所示。时间的单位为信元时隙(slot), 即输出链路发送一个信元的时间。速率的单位为信元/信元时隙(cell/slot), 链路速率为1 cell/slot。在仿真中, 每一个连接的信元的时延都被记录, 仿真结束后统计出每一个连接的最大信元时延和平均时延。在仿真中假设 ATM 复接器缓存无限大。

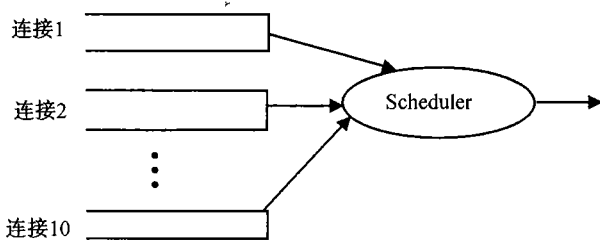


图1 仿真中 ATM 复接器的设置

在表1中, 连接8为 On_Off VBR, 主要模拟实时突发业务, 其峰值速率为平均速率的2倍, 其权重按平均速率设置。连接9为恶意用户, 速率为其权重的10倍。

2000年5月23日收稿

* 国家“九五”重点科研项目

** 男 29岁 在职博士生

2 仿真结果及分析

2.1 各种 PFQ 算法时延的比较

当调度器分别采用 GPS、WFQ、WF²Q、WF²Q+、SPFQ、SCFQ、VirtualClock、ShVC (Shaped Virtual Clock) 算法时, 对信元时延仿真的结果如表2和表3所示。

表1 连接的信源特性

连接编号	信源类别	峰值速率 (cell/slot)	平均速率 (cell/slot)	权重
0	CBR	0.05		0.05
3	CBR	0.05		0.05
6	CBR	0.05		0.05
1	CBR	0.10		0.10
4	CBR	0.10		0.10
7	CBR	0.10		0.10
2	CBR	0.15		0.15
5	CBR	0.15		0.15
8	On_Off VBR	0.30	0.15	0.15
9	Misbehaved Source	1.00	1.00	0.10

表2 最大时延的比较

连接编号	最大时延/slot							
	GPS	WFQ	WF ² Q	WF ² Q+	SPFQ	SCFQ	VC	ShVC
0	20.00	18.000	17.000	16.000	16.000	19.000	17.000	17.000
3	20.00	18.000	16.000	15.000	17.000	18.000	18.000	17.000
6	20.00	18.000	16.000	17.000	18.000	19.000	17.000	18.000
1	10.00	8.000	7.000	6.000	7.000	12.000	8.000	7.000
4	10.00	8.000	8.000	8.000	6.000	12.000	7.000	7.000
7	10.00	8.000	7.000	6.000	6.000	13.000	7.000	8.000
2	7.33	7.667	6.667	6.667	5.667	10.670	6.667	5.667
5	7.33	7.667	6.667	5.667	6.667	10.000	6.667	6.667
8	34.98	33.350	32.350	33.340	33.350	28.680	34.350	34.350
9	621.00	622.000	621.000	621.000	621.000	621.000	895.000	894.000

表3 平均时延的比较

连接编号	平均时延/slot							
	GPS	WFQ	WF ² Q	WF ² Q+	SPFQ	SCFQ	VC	ShVC
0	8.778	6.280	6.270	5.340	6.030	6.560	6.360	5.900
3	8.778	6.370	6.080	5.070	6.280	7.010	6.210	6.060
6	8.778	6.350	5.960	5.500	6.440	6.830	6.050	6.610
1	7.587	4.910	4.280	3.160	3.330	6.010	3.770	3.950
4	7.587	4.750	4.370	3.380	3.000	6.040	3.930	3.700
7	7.587	4.650	4.360	3.460	3.300	5.960	3.710	3.970
2	6.606	3.640	3.190	2.690	2.410	6.000	3.510	3.470
5	6.606	3.700	3.030	2.820	2.490	5.840	3.490	3.530
8	11.160	8.377	7.677	8.157	7.057	9.640	10.410	10.400
9	360.100	357.500	356.900	362.500	361.800	342.200	449.300	449.200

在表2、3中, WFQ、WF²Q、WF²Q+、SPFQ、VC、ShVC 对于 CBR 业务都能提供最大时延的保证, 即都小于 $\frac{1}{r}+1$, r 为 CBR 业务的速率。这些算法对于 CBR 业务提供的平均时延也相差不多。GPS 提供的时延较大, 这是由于 GPS 是流体系统, 各个连接的信元可同时接受服务或同时离开。SCFQ 与这些算法有明显的不同。在表2中, SCFQ 对连接1、2、4、5、7提供的最大时延都超过了 $\frac{1}{r}+1$ 。在平均时延方面, 对于连接0~8, SCFQ 提供的时延比 WFQ、WF²Q、WF²Q+、SPFQ 提供的大, 但对于连接9, 提供的时延比其小。这说明 SCFQ 的隔离性比较差, 恶意用户(如连接9)可以影响其他的连接, 而使自己得到某些利益。VC、ShVC 对于连接8、9提供的时延比其他算法大, 这是由于连接8、9产生的流量超过了其预约的带宽, 而 VC、ShVC 本身就有限定流量的作用。

2.2 WF²Q+的两种实现方式的比较

WF²Q+的实现方式有两种: 一种是按 WF²Q+的的定义的方式, 为每一个连接维护一个队列, 维护一个时标, 在不同连接的时标之间进行比较以确定下一个服务的分组。这种方式需比较的时标个数等于连接的个数。另一种方式是将同样预约速率的连接放在一个速率组内, 同一个速率组内的各个连接的头一个信元按 FCFS(first come first serve)方式排列, 比较时标只需在各个速率组的头一个连接的时标之间进行, 参与比较的时标个数为速率组的个数。显然, 后一种方式需要比较的时标个数比前一种方式少, 这种方式称为 Per_Rate_Grouping, 为了叙述方便, 本文称前一种方式为 No_Per_Rate_Grouping。表4是对这两种方式的信元时延的仿真结果。

表4 两种实现 WF²Q+方式的比较

连接编号	最大时延/slot				平均时延/slot			
	不分组	分组1	分组2	分组3	不分组	分组1	分组2	分组3
0	16.000	12.000	13.000	12.000	5.340	2.650	3.830	3.540
3	15.000	13.000	14.000	15.000	5.070	4.090	5.060	5.130
6	17.000	14.000	17.000	16.000	5.500	5.540	6.580	7.130
1	6.000	12.000	6.000	5.000	3.160	5.400	2.190	2.050
4	8.000	13.000	7.000	7.000	3.380	6.560	3.410	3.650
7	6.000	14.000	8.000	8.000	3.460	7.850	4.580	4.830
2	6.667	7.667	8.333	6.667	2.690	4.140	5.220	2.820
5	5.667	8.667	9.333	5.667	2.820	4.410	5.510	3.170
8	33.340	31.350	33.350	33.350	8.157	8.530	9.630	8.900
9	621.000	679.000	675.000	656.000	362.500	393.300	397.500	382.600

分组1表示第一种分组方法, 分组2、3类似。分组1的分组方法为: group1{0,3,6}、group2{1,4,7,9}、group3{2,5,8}。分组2的分组方法为: group1{0,3,6}、group2{1,4,7}、group3{2,5,8}、group4{9}。分组3的分组方法为: group1{0,3,6}、group2{1,4,7}、group3{2,5}、group4{8}、group5{9}。分组1和不分组相比, 分组1中的连接时延降低, 特别是最大时延比不分组情况有明显的减少; 分组2、分组3的连接时延增大, CBR 的时延超过了 $\frac{1}{r}+1$ 的限制, 但在 $\frac{1}{r} + \frac{1}{r} + 1$ 的限制之内^[10]。分组2将恶意用户连接9从分组2中分离出来, 单独组成分组4。分组2中的连接时延明显比分组1的情况减少, 基本恢复到不分组的情况, 特别是最大时延在 $\frac{1}{r}+1$ 的限制之内。分组3中的连接时延和分组1相差不多, 最大时延仍然超过了 $\frac{1}{r}+1$ 。分组3将连接8(On_Off VBR)从分组3中分离出来, 单独

组成分组4，而连接9单独组成分组5。由此可见，所有的 CBR 业务的最大时延都在 $\frac{1}{r} + 1$ 的限制之内，平均时延也与不分组情况相差不大。不分组情况需要比较的时标个数为10个，而分组3需要比较的时标个数仅为5个。

3 仿真结论及其应用

仿真结果表明：1) SCFQ 不是一个好的 PFQ 算法，对连接之间的隔离性比其他算法差；2) 如果一个速率组内没有流量超过其预约带宽的用户，则用 Per_Rate_Grouping 方式实现 WF²Q+和不用这种方式实现，从最大时延和平均时延来看，相差不大；3) 如果一个组内有恶意用户(流量超过其预约带宽)，则组内的其他连接会受到影响，其最大时延和平均时延会增大，最大时延可能超过 No_Per_Rate_Grouping 方式下的上限，即 $\frac{1}{r} + 1$ 。另外，一个组内的恶意用户对其他组内的用户影响不大，这是因为算法对属于不同速率组的连接进行了隔离。

根据上述仿真结果，本文给出一种调度器的结构，如图2所示。最上层采用 WF²Q+的 per_rate_grouping 实现方式，CBR 连接根据其预约速率分属于不同的 CBR 速率组。每一个 CBR 连接在进入速率组之前经过成形器进行流量控制。实时 VBR(rt_VBR)也根据其预约速率分属于不同的 rt_VBR 速率组，非实时 VBR(nrt_VBR)的所有连接作为一个组，nrt_VBR 的各个连接之间采用容易实现的 WRR(weighted round robin)调度算法来调度。ABR 和 UBR 的情况类似于 nrt_VBR。这种调度结构对于 CBR 业务可以保证其最大时延为 $\frac{1}{r} + 1$ ，对于 rt_VBR，当其流量满足参数为 (δ, r) 的漏桶限制时，保证其最大时延为 $\frac{\delta}{r} + \frac{\delta}{r} + 1$ ，对于 nrt_VBR、ABR 和 UBR 这些对于时延没有严格要求的业务，这种调度结构可以保证一定的吞吐率。这种调度结构的实现复杂性只取决于 CBR 速率组和 rt_VBR 速率组的个数，因此比较容易实现。在做 CAC(连接接纳控制)时，构对于 CBR 业务的最大时延是按 $\frac{1}{r} + 1$ 来计算的，比文献[10]给出的按 $\frac{1}{r} + \frac{1}{r} + 1$ 来计算可以接纳更多的 CBR 连接。

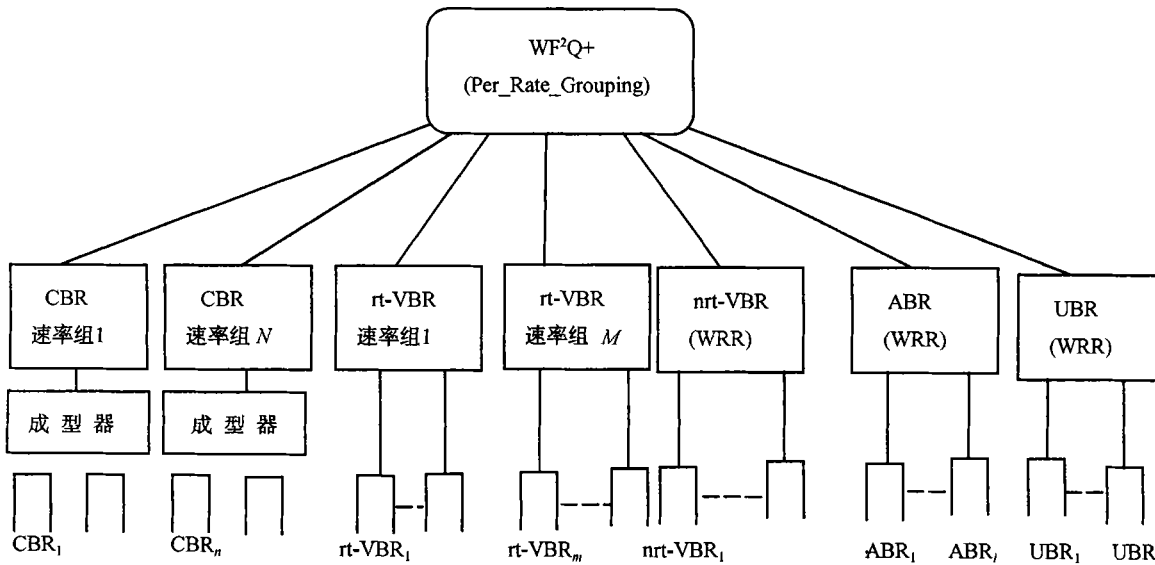


图2 一种调度器结构

参 考 文 献

- 1 李乐民, 谭真平. 信息高速公路与 ATM 网络技术. 电子科技大学学报, 1994,23(增刊): 197~201
- 2 来光明, 刘荣富. ATM 交换技术. 电子科技大学学报, 1994, 23(增刊): 214~219
- 3 Demers A, Keshav S, Shenker S. Analysis and simulation of a fair queuing algorithm. ACM Sigcomm, 1989: 1~13
- 4 Parekh A K, Gallager R G. A generalized processor sharing approach to flow control in integrated services networks: the single-node case. IEEE Trans Networking, 1993, 1(3): 344~357
- 5 Benett J, Zhang Hui. WF²Q:Worst-case fair weighted fair queuing. IEEE Infocom, 1996: 120~128
- 6 Benett J, Zhang Hui. Hierarchical packet fair queuing algorithms. IEEE Trans Networking, 1997, 5(5): 675~689
- 7 Varma A, Stiliadis D. Hardware implementation of fair queuing algorithms for asynchronous transfer mode networks. IEEE Comm Magazine, 1997, (12): 54~68
- 8 Zhang L. Virtual clock:a new traffic control algorithm for packet switching networks. ACM Trans Comp Sys , 1991,9(1): 101~124
- 9 Golestani S. A self-clocked fair queuing scheme for broadband applications. IEEE Infocom, 1994: 636~646
- 10 Bennett J C R, Stephens D C, Zhang Hui. High speed,scable,and accurate implementation of packet fair queuing algorithms in ATM networks. In Proceedings of IEEE ICNP', 1997: 7~14
- 11 Stephens D C, Bennett J C R, Zhang Hui. Implementing scheduling algorithms in high-speed networks. IEEE JSAC Special Issue on High Performance Switches/Routers, 1999, (9): 100~118

Research on Simulation of Packet Fair Queueing Algorithms

Tu Xiaodong Li Lemin

(National Key Lab of Optical Fiber Transmission and Communication Networks, UEST of China Chengdu 610054)

Abstract In this paper , the cell delay in ATM MUX and output-buffered ATM switch is simulated, which adopts various packet fair queuing algorithms. The simulation results are analyzed, which show that the isolation property of SCFQ(self clocked fair queuing) is weaker than others. If a CBR group has no connection in which traffic exceeds reservation, as to the maximum cell delay and cell average delay, the difference between per_rate_grouping implementation of WF²Q+ and no_per_rate_grouping implementation of WF²Q+ is negligible. According to the results, a scheduling structure dealing with CBR、VBR、ABR and UBR is proposed, which satisfies the strict delay requirement of CBR and guarantees the throughput of VBR、ABR and UBR. The implementation complexity of the scheduler is also discussed.

Key words packet fair queuing; WF²Q+; per_rate_grouping; constant bit rate