BUFFER MANAGEMENT AND SCHEDULING FOR TCP/IP OVER ATM-GFR *

Dapeng Wu and H. Jonathan Chao
Center for Advanced Technology in Telecommunications
Department of Electrical Engineering
Polytechnic University
6 Metrotech Center, Brooklyn, New York 11201
Email: chao@antioch.poly.edu

Abstract

Today ATM technology is facing challenges from Integrated Service IP, IP switching, Gigabit IP router and Gigabit Ethernet. Although ATM is approved by ITU-T as the standard technology in B-ISDN, its survivability is still in question. Since ATM-UBR (Unspecified Bit Rate) provides no service guarantee and ATM-ABR (Available Bit Rate) is still unattainable for most users, many existing users have little or no incentives to migrate to ATM technology. The Guaranteed Frame Rate (GFR) service is introduced to deal with this dilemma. The GFR can guarantee the Minimum Cell Rate (MCR) with fair access to excess bandwidth. This paper studies various schemes to support the GFR. We have studied different discarding and scheduling schemes, and compared their throughput and fairness when TCP/IP traffic is carried. Through simulations, it is shown that only per-VC queuing with Weighted Round Robin (WRR) can guarantee Minimum Cell Rate. Among all the schemes that have been explored, we recommend DT-EPD (Dynamic Threshold - Early Packet Discard) integrated with MCR+ (a WRR variant) to support the GFR service.

1 Introduction

One advantage of ATM technology is that significant performance and efficiency benefits can be achieved if and when applications/users are able to exploit the full range of ATM traffic parameters and service classes. However, this advantage is still unattainable for many users today. These users are either not able to specify the range of traffic parameters, i.e., PCR (Peak Cell Rate), SCR (Sustainable Cell Rate), MBS (Maximum Burst Size), which are needed to request most ATM services, or are not equipped with devices capable of effectively interacting with an ATM network, i.e., enforcing ABR source mechanism. The only access these users have to ATM networks is through UBR connections, which provide no service guarantees. Schemes such as EPD (Early Packet Discard) or PPD (Partial Packet Discard) [5], though improving the goodput of packet traffic over UBR, cannot guarantee a minimum packet rate. As a result, many existing users have little or no incentive to migrate to ATM technology.

To deal with this situation, Guerin and Heimanen proposed a new service which was originally called “UBR+” [2] but is now called GFR [7]. And ATM Forum is currently discussing the need for GFR service. The objective of the GFR service is to bring as yet unavailable benefits of ATM performance and service guarantees to users. Thus, the GFR service requires minimal interactions between users and ATM networks, but provides users with a certain level of service guarantees. The essence of GFR is to guarantee Minimum Cell Rate (MCR) with fair access to excess bandwidth.

To efficiently support the GFR service, some requirements need to be imposed on the network. This paper will answer the following questions:

What mechanisms in ATM switches are required to support the GFR service?

Are these mechanisms simple enough to be implemented?

In this paper we investigate TCP performance over several GFR implementations and present sim-
ulation results. The focus of our research is on whether the implementations discussed at least guarantee the MCR and reasonable fairness.

The remainder of the paper is organized as follows. Section 2 discusses an EPD (Early Packet Discard) variant using dynamic threshold [1]. Section 3 describes three pushout schemes and a quasi-pushout scheme [4] to implement the GFR service. Through simulations, we compare the performance of the schemes described above. Furthermore, our observation leads to the introduction of an improved scheme, called MCR+. Finally, we conclude our investigation on various discarding and scheduling schemes and give our recommendation on the implementation of the GFR service.

2 Dynamic Threshold EPD

EPD is proposed by Romenov and Floyd [1]. To implement ATM-GFR, Sla proposed an EPD variant using Virtual Queueing (VQ) technique [6]. Here we discuss Choudhury and Hahne’s scheme applied in EPD scenario by replacing the static threshold with Dynamic Threshold (DT) [1]. Per-VC queuing is utilized in Dynamic Threshold EPD (DT-EPD) (see Figure 1). The scheduling mechanisms are Weighted Round Robin and Round Robin. The dynamic threshold (DT) is determined as follows.

$$DT = \alpha \cdot (BufferSize - TotalQueueLength)$$  (1)

Note: BufferSize - TotalQueueLength = Free Space

If $\alpha = 2$, for example, DT-EPD tries to regulate each queue to be twice the free buffer space. So a single queue with no competition is allowed to take $2/3$ of the entire shared memory, and $1/3$ of the memory is held back. When two long queues are active, each queue gets $2B/5$, and $B/5$ is unallocated. When there are three long queues, each queue gets $2B/7$, with $B/7$ unallocated. If the number of very active VCs (which are greedy at this time) then increases from 3 to 10, the long queues (which are longer than DT) will drain and the newly active queues will grow until all ten stabilize at $2B/21$, with $B/21$ left unallocated.

DT-EPD deliberately wastes a small amount of buffer space. This “waste” actually serves two useful functions. The first advantage of maintaining some spare space at all times is that it provides a cushion during transient periods when a VC queue first becomes active. This reduces cell loss during such transient periods. Secondly, when a VC queue has such a load increase and begins taking over some of the spare buffer space, this action signals the allocation mechanism that the load conditions have changed and that a threshold adjustment is now required.

3 Pushout Discarding Schemes

3.1 Pushout Schemes

Pushout discarding schemes also treat TCP data stream in packet units. There are three Pushout schemes. Pushout-1 drops the HOL (Head Of Line) partial packet from the longest queue, which is delimited by the HOL cell and the first EOM (End Of Message) cell. Pushout-2 searches for an entire packet from the head of the longest queue and drops it. Pushout-3 drops the partial packet from the tail of the longest queue, which is delimited by the tail cell and the last EOM cell. Based on per-VC queueing, Pushout schemes utilize Weighted Round Robin and Round Robin as the scheduling policies. The three Pushout schemes are shown in Figure 2.

Through simulation [3], Lakshman, et. al., showed that, for the same buffer size, drop from front results in considerably higher TCP throughput than
tail drop, and for all but very small buffers even higher than tail drop combined with partial frame drop. Thus, we expect PO1 should be better than PO3, and PO2 should be better than PO1. Our simulation results substantiate our expectation.

3.2 Quasi Pushout Scheme

The Pushout (PO) discarding scheme has been shown to offer optimum cell loss performance [8]. However, the Pushout scheme is very difficult to implement because it requires O(N queue length comparisons to find out the longest queue, where N is the number of output queues. When N is large, these comparisons may become the speed bottleneck.

Y.S. Lin et al. [4] proposed the Quasi Pushout (QPO) cell discarding scheme, which features a much reduced hardware complexity than PO. A register Longest Queue (LQ) for the quasi-longest queue is maintained and updated during cell arrival or departure events. At the time when the buffer is full, one cell is discarded from the quasi-longest queue to make room for the incoming cell.

The following pseudocode describes the QPO discarding scheme based on PO2 (see Figure 3):

When a cell in VC1 reaches a switch:

if (buffer full) 

search for an entire packet from the head of quasi longest queue and drop it

QL.LQ = QL.LQ - x; /* x is the number of cells discarded */

accept the incoming cell into the VCi queue

QL[i] = QL[i]+1; /* buffering input cell*/

if (QL.LQ < QL[i])

LQ = i; /* input-comparison */

When a cell in VCj queue is transmitted

4 Simulation Experiment

4.1 Simulation Model

Our simulation tool is based on the NIST ATM Network Simulator. This simulator is an event-driven simulator composed of various components that send messages to one another. We have changed some ATM and TCP related components in this simulator to meet our own needs.

Figure 4 illustrates the simulation model of a network with ten peer-to-peer connections. On the sending side, the sources (source 1 to source 10) generate data for TCP/IP components, which form TCP/IP packets, and which in turn are passed on for AAL5 processing. The two ATM switches perform cell switching between their input and output ports. On the receiving side, cells are reassembled and passed to the TCP/IP components. By running ten concurrent connections, we create a congested link between the two ATM switches.

The following parameters are employed in our simulations:

TCP:

Mean Packet Processing Delay = 300 usec  
Packet Processing Delay Variation = 10 usec  
Packet Size = 2K Bytes  
Maximum Receiver Window Size = 64K Bytes  
Default Timeout = 500 ms  
Timer Granularity = 10 usec  
TCP Reno version
<table>
<thead>
<tr>
<th>MCR (Mbps)</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>Total</th>
<th>FairnessEff</th>
<th>diff%</th>
</tr>
</thead>
<tbody>
<tr>
<td>DropTail</td>
<td>13.26</td>
<td>11.97</td>
<td>11.42</td>
<td>11.7</td>
<td>10.27</td>
<td>11.89</td>
<td>11.16</td>
<td>11.27</td>
<td>11.64</td>
<td>13.04</td>
<td>117.54</td>
</tr>
<tr>
<td>VQ1</td>
<td>13.1</td>
<td>13.62</td>
<td>12.62</td>
<td>13.33</td>
<td>14.2</td>
<td>11.38</td>
<td>12.56</td>
<td>13.37</td>
<td>13.07</td>
<td>13.77</td>
<td>131.05</td>
</tr>
<tr>
<td>DT-FPD</td>
<td>13.48</td>
<td>13.23</td>
<td>13.15</td>
<td>13.27</td>
<td>13.29</td>
<td>13.01</td>
<td>12.98</td>
<td>12.96</td>
<td>13.28</td>
<td>13.15</td>
<td>121.82</td>
</tr>
<tr>
<td>PO1</td>
<td>13.75</td>
<td>13.51</td>
<td>13.65</td>
<td>12.66</td>
<td>13.16</td>
<td>13.17</td>
<td>13.06</td>
<td>13.26</td>
<td>13.51</td>
<td>13.26</td>
<td>122.65</td>
</tr>
<tr>
<td>PO2</td>
<td>13.69</td>
<td>13.5</td>
<td>13.33</td>
<td>13.33</td>
<td>13.21</td>
<td>13.55</td>
<td>13.35</td>
<td>13.36</td>
<td>13.39</td>
<td>13.34</td>
<td>134.28</td>
</tr>
<tr>
<td>PO3</td>
<td>12.19</td>
<td>12.34</td>
<td>11.78</td>
<td>11.45</td>
<td>12.83</td>
<td>11.63</td>
<td>11.96</td>
<td>12.78</td>
<td>13.1</td>
<td>121.32</td>
<td>0.9973</td>
</tr>
<tr>
<td>QPO</td>
<td>14.07</td>
<td>13.47</td>
<td>13.36</td>
<td>13.44</td>
<td>13.35</td>
<td>13.56</td>
<td>12.98</td>
<td>13.53</td>
<td>13.41</td>
<td>13.22</td>
<td>134.4</td>
</tr>
</tbody>
</table>

*The rates shown in the table are the average TCP rates of the entire simulation.

Table 1: Simulation results for equal MCRs.

Unidirectional traffic

Greedy sources

Link:

\[ \text{Speed} = 155.52 \text{ Mbps} \]
\[ \text{Delay} = 5 \text{ ms} (1 \text{Km}) \text{ in LAN or 5 ms (1000Km)} \text{ in WAN} \]

UBR-End System:

\[ \text{Packet Processing Delay} = 500 \text{ usec} \]
\[ \text{Buffer Size} = \text{infinity} \]
\[ \text{Cell Transmission Rate} = 155.52 \text{Mbps} \]

UBR-Switch:

\[ \text{Non-blocking Output-buffered Switch} \]
\[ \text{Packet Processing Delay} = 4 \text{ usec} \]
\[ \text{Buffer Size (Qmax)} = 3000 \text{ cells (for LAN)} \]
\[ \text{and 36000 cells (for WAN)} \]
\[ \text{EPD Threshold (Th)} = 1000 \text{ cells} \]
\[ T = 1 \text{ usec} \]

Simulation Time = 2 seconds (for LAN) and 5 seconds (for WAN)

4.2 Performance Metrics

The performance of TCP over GFR is measured by efficiency, fairness, and difference, which are defined as follows:

\[ \text{Efficiency} = \frac{\text{Sum of TCP throughput}}{\text{Maximum Possible TCP throughput}} \]

\[ \text{Fairness Index} = \frac{(\sum_n(\text{Throughput}_n - \text{MCR}_n))^2}{n \times (\sum_n(\text{Throughput}_n - \text{MCR}_n))^2} \]

\[ \text{Difference} = \frac{\text{MaxTh} - \text{MinTh}}{\text{AverageTh}} \]

where

\[ \text{MaxTh} = \text{Maximum Throughput of the TCP connection among all}, \]
\[ \text{MinTh = Minimum Throughput of the TCP connection among all}, \]

\[ \text{Average Th} = \frac{\text{Sum of TCP throughput}}{\text{Number of TCP connections}}, \]

\[ n = \text{the number of TCP connections}. \]

The TCP throughputs are measured at the destination TCP layers. Throughput is defined as the total number of bytes delivered to the destination application divided by the total simulation time. The results are reported in Mbps. The maximum possible TCP throughput is the throughput attainable by the TCP layer running over UBR on a 155.52 Mbps link. For 2048 bytes of data (TCP maximum segment size), the ATM layer receives 2048 bytes of data + 20 bytes of TCP header + 20 bytes of IP header + 8 bytes of LLC header + 8 bytes of AAL5 trailer. These are padded to produce 44 ATM cells. Thus, each TCP segment results in 2332 bytes at the ATM layer. From this, the maximum possible throughput = 2048/2332 = 87.8% = 136.6 Mbps approximately on a 155.52 Mbps link.

4.3 Simulation Results

4.3.1 Performance Comparison with Equal MCRs

In Table 1, the reserved MCR of each VC is shown in the first row. The rates in the table are TCP layer rates, which exclude TCP, IP, LLC, and AAL5 overheads. Since MCR is equal to each VC, each VC should therefore get the same throughput. As seen from Table 1, the fairness index cannot show the difference among the schemes. Thus we introduce the metric of difference in order to see how well each scheme performs.

Our simulation shows that PO2 has the best per-
formance in terms of fairness (difference is only 1.8%). QPO has the highest throughput because QPO favors some connections. As a result, QPO does not achieve as good a fairness as PO2. Since the performances of PO1 and PO3 are not as good as PO2, we will not discuss PO1 and PO3 in the following sections. DT-EPD is comparable to PO2. Compared with the no-control case (DropTail), VQ (Virtual Queueing) [6] can improve throughput and fairness but not as well as QPO.

### 4.3.2 Performance Comparison with Different MCRs

The reserved MCR of each VC is shown in the first row of Table 2. Since MCR is different for different VC, IdealRate parameters will be more straightforward to see how fair each scheme is. As shown from Table 2, PO2, DT-EPD and QPO are comparable, and none of these schemes can guarantee perfect fairness. We also observe that the larger the MCR is, the smaller share in excess bandwidth the VC gets. This is non-linearity. In addition, we can see that VQ cannot guarantee MCR.

$$IdealRate(i) = FairShare + MCR_i$$

$$FairShare = ExcessBandwidth/(Number of VC)$$

$$ExcessBandwidth = MaxTh-(sum of MCR_i)$$

### 4.3.3 Performance of MCR+

From Table 2, we observe that VCs with smaller MCR can get larger share in excess bandwidth. One possible solution is to increase MCR non-linearly by favoring VCs with larger MCR. We call this scheme MCR+. In MCR+ scheme $MCR'$ is used for Weighted Round Robin and given as follows.

$$MCR'_i = MCR_i + ReservedBandwidth \times MCR_i/(\sum MCR_i)$$

Through simulation (see Table 3 and 4), we find that MCR+ scheme can achieve much better fairness under different MCR combinations when Reserved-Bandwidth is 10% of the link rate. Reserving 10% of the link bandwidth is feasible since normally ATM switches will reserve at least 10% of the link bandwidth to deal with burstiness. In this simulation, MCR+ is based on PO2 and DT-EPD respectively. Table 3 shows the simulation results in LAN environment and Table 4 shows those in WAN environment.

Insight of MCR+

As we know, TCP is window-based flow control rather than rate-based flow control. When there is...
Table 4: Simulation results for MCR+ compared with other four schemes in WAN.

<table>
<thead>
<tr>
<th>MCR (Mbps)</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>10</th>
<th>15</th>
<th>18</th>
<th>18</th>
<th>20</th>
<th>Total</th>
<th>Fairness</th>
<th>Eff</th>
</tr>
</thead>
<tbody>
<tr>
<td>VQ</td>
<td>11.7</td>
<td>13.59</td>
<td>13.03</td>
<td>13.38</td>
<td>13.45</td>
<td>13.13</td>
<td>13.09</td>
<td>13.58</td>
<td>13.67</td>
<td>13.74</td>
<td>13.24</td>
</tr>
<tr>
<td>DT-EPD</td>
<td>4.31</td>
<td>5.953</td>
<td>9.67</td>
<td>11.61</td>
<td>11.16</td>
<td>15.81</td>
<td>16</td>
<td>18.53</td>
<td>18.55</td>
<td>20.53</td>
<td>13.1</td>
</tr>
<tr>
<td>PO2</td>
<td>4.58</td>
<td>6.256</td>
<td>9.992</td>
<td>11.5</td>
<td>11.58</td>
<td>16.05</td>
<td>16.07</td>
<td>18.62</td>
<td>18.72</td>
<td>20.55</td>
<td>13.39</td>
</tr>
<tr>
<td>QPO</td>
<td>4.53</td>
<td>6.364</td>
<td>9.952</td>
<td>11.66</td>
<td>11.73</td>
<td>16</td>
<td>16.15</td>
<td>18.59</td>
<td>18.61</td>
<td>20.39</td>
<td>13</td>
</tr>
<tr>
<td>MCR+(PO2)</td>
<td>3.48</td>
<td>5.456</td>
<td>9.382</td>
<td>11.54</td>
<td>11.42</td>
<td>16.38</td>
<td>16.28</td>
<td>19.35</td>
<td>19.23</td>
<td>21.34</td>
<td>13.8</td>
</tr>
</tbody>
</table>

5 Conclusion

To support the GFR service, we have studied and compared different buffer management schemes: Dynamic Threshold EPD (DT-EPD), Pushout 1, Pushout 2 (PO2), Pushout 3, Quasi Pushout (QPO), and Virtual Queuing. Through simulations, we reach the following conclusions.

- Scheduling policies:
  Only per-VC queuing with Weighted Round Robin for allocated MCR and Round Robin for excess bandwidth can guarantee MCR and achieve fairness.

- Discarding policies:
  The performances of PO2, DT-EPD and QPO are comparable. DT-EPD is the simplest to implement.

- Among all the schemes, we recommend DT-EPD combined with MCR+, which increases MCR non-linearly by favoring VCs with larger MCR, to support the GFR service. The simulation results show that DT-EPD combined with MCR+ scheme performs well in LAN and WAN environment.

References


