
ATM Forum Document Number: ATM_Forum/98-0405

TITLE: Buffer Management for the GFR Service

SOURCE: Rohit Goyal, Raj Jain, Sonia Fahmy, Bobby Vandalore
The Ohio State University,

**Raj Jain is now at
Washington University in Saint Louis
Jain@cse.wustl.edu
<http://www.cse.wustl.edu/~jain/>**

Computer Science,
63210-1277

This work is partially sponsored by the NASA Lewis Research Center Under Contract Number NAS3-97198

DISTRIBUTION: ATM Forum Technical Committee
Traffic Management Working Group

DATE: July, 1998 (Portland)

ABSTRACT: In this contribution, we present a buffer management scheme called Differential Fair Buffer Allocation (DFBA) that provides MCR guarantees to GFR VCs carrying TCP/IP traffic. DFBA can be used on a FIFO buffer shared by several VCs. Each VC can carry traffic from one or more TCP connections. We discuss the features of DFBA and present simulation results to analyze its performance.

NOTICE: This document has been prepared to assist the ATM Forum. It is offered as a basis for discussion and is not binding on the contributing organization, or on any other member organizations. The material in this document is subject to change in form and content after further study. The contributing organization reserves the right to add, amend or withdraw material contained herein.

1 Introduction

The Guaranteed Frame Rate service has been designed to support non-real-time applications that can send data in the form of frames. IP routers separated by ATM clouds can benefit from the MCR guarantees provided by the GFR service. As a result, GFR implementations must be able to efficiently support MCR guarantees for TCP/IP traffic. These guarantees should be provided on a per-VC basis, where each GFR VC may contain traffic from several TCP connections.

While per-VC rate guarantees can be provided with per-VC queuing and scheduling, for most best effort traffic, it may be cost-effective to be able to provide minimum rate guarantees using a single queue. Intelligent buffer management techniques can be used to provide minimum rate guarantees. Such buffer management schemes must work with TCP traffic, and take into account the conservative slow start mechanism used by TCP on packet loss. Modern TCP implementations are expected to use Selective Acknowledgements (SACK) to minimize the occurrence of timeouts that trigger slow start. However, even with SACK, large losses due to severe congestion or very aggressive switch drop policies, can trigger timeouts. In addition to MCR guarantees, the GFR VCs should also be able to fairly share any excess capacity. As a result, the design of a good buffer management scheme for providing minimum rate guarantees to TCP/IP traffic is an important step towards the successful deployment of GFR.

In this contribution, we present the Differential Fair Buffer Allocation buffer management scheme. This buffer management work is an extension of our previous work on buffer management presented in [GOYALa]. The DFBA scheme presented here is an improved version of the scheme presented in [GOYALb]. We first overview some of the previous results on TCP over GFR. We then discuss the DFBA scheme, and present simulation results for this scheme. We conclude this contribution with a discussion of some key buffer management policies and their limitations.

2 Previous Results on TCP/IP over GFR

Several proposals have been made ([BASAK],[BONAVEN97],[GOYALa])to provide rate guarantees to TCP sources with FIFO queuing in the network. The bursty nature of TCP traffic makes it difficult to provide per-VC rate guarantees to TCP sources using FIFO queuing. Per-VC scheduling was recommended to provide rate guarantees to TCP connections. However, all these studies did not consider the impact of TCP dynamics, and used aggressive drop policies. We show that rate guarantees are achievable with a FIFO buffer using DFBA.

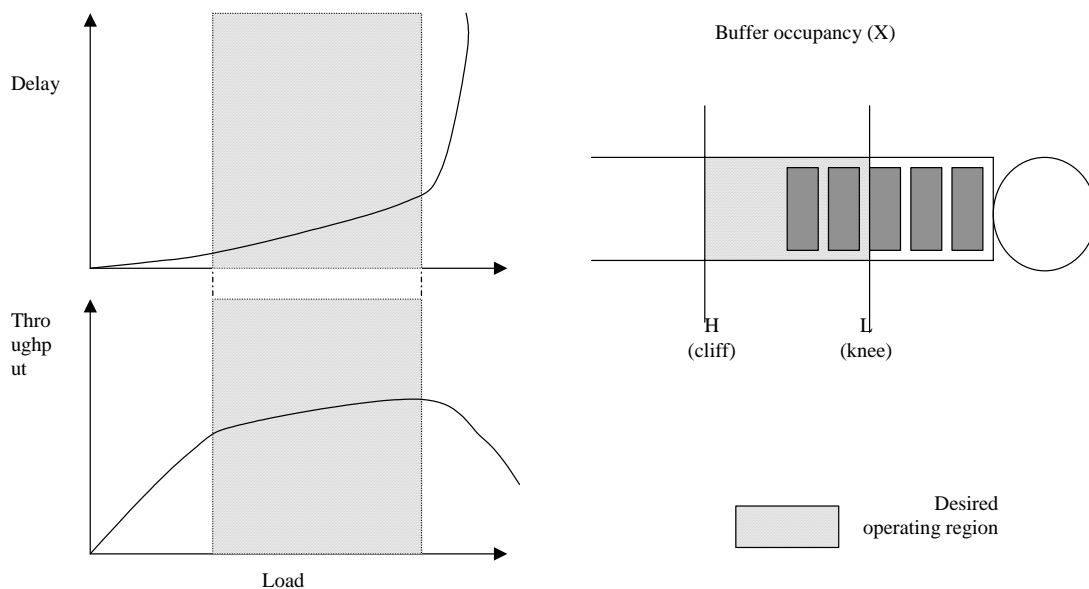
Many of the previous studies have examined TCP traffic with a single TCP connection over a VC. Per-VC buffer management for such cases, reduces to per-TCP buffer management. However, routers using GFR VCs would typically multiplex many TCP connections over a single VC. For VCs with several aggregated TCPs, per-VC control is unaware of each TCP in the VC. Moreover, aggregate TCP traffic characteristics and control requirements may be different from those of single TCP streams.

In [GOYALb], we have used FIFO buffers to control SACK TCP rates by buffer management using a preliminary version of DFBA. The scheme could allocate MCRs to TCP sources when the total MCR allocation was low (typically less than 50% of the GFR capacity). However, it was not clear how to allocate buffers based on the MCRs allocated to the respective VCs. Several other schemes have recently been presented for MCR

guarantees to GFR VCs carrying TCP traffic ([BONAVENb],[CHAO],[ELLOUMI]). In the following sections, we further describe the DFBA scheme, and discuss its design choices. We then present simulation results for both low and high MCR allocations using DFBA.

3 Differential Fair Buffer Allocation

DFBA uses the current queue length as an indicator of network load. The scheme tries to maintain an optimal load so that the network is efficiently utilized, yet not congested. The figure below illustrates the operating region for DFBA. The high threshold (H) and the low threshold (L) represent the cliff and the knee respectively of the load versus delay/throughput graph. The goal is to operate between the knee and the cliff. The scheme also assumes that the delay/throughput versus load curve behaves in a linear fashion between the knee and the cliff.

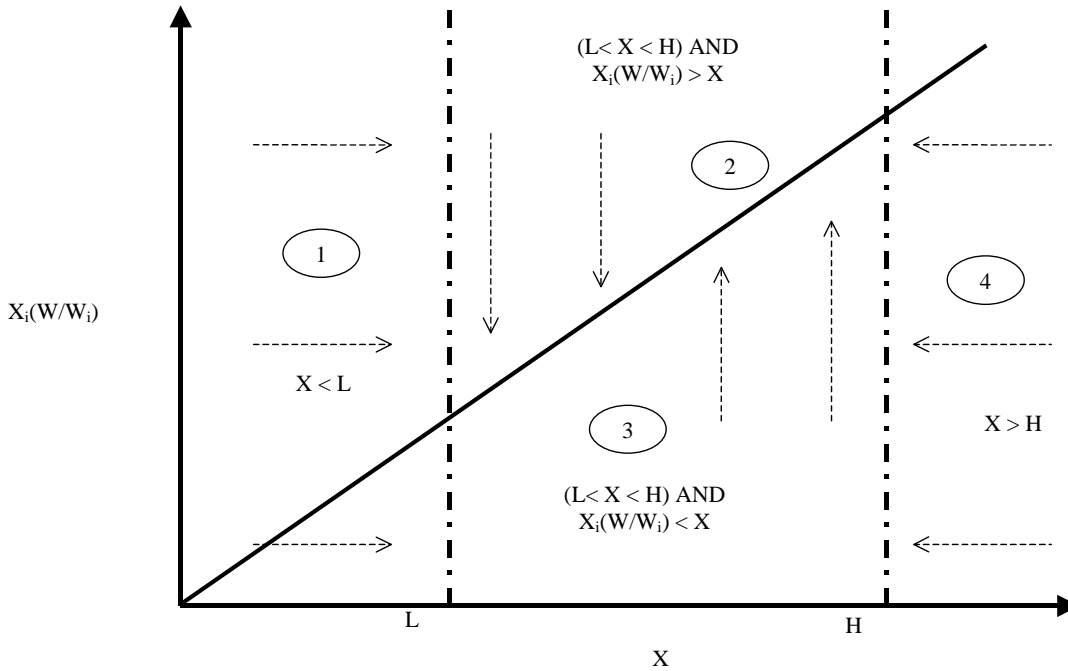


In addition to efficient network utilization, DFBA is designed to allocate buffer capacity fairly amongst competing VCs. This allocation is proportional to the MCRs of the respective VCs. The following variables are used by DFBA to fairly allocate buffer space:

- X = Total buffer occupancy at any time
- L = Low buffer threshold
- H = High buffer threshold
- MCR_i = MCR guaranteed to VC_i
- W_i = Weight of $VC_i = MCR_i / (\text{GFR capacity})$
- $W = \sum W_i$
- X_i = Per-VC buffer occupancy ($X = \sum X_i$)
- Z_i = Parameter ($0 \leq Z_i \leq 1$)

DFBA tries to keep the total buffer occupancy (X) between L and H . When X falls below L , the scheme attempts to bring the system to efficient utilization by accepting all incoming packets. When X rises above H , the scheme tries to control congestion by performing EPD. When X is between L and H , DFBA attempts to allocate buffer space in proportional to the MCRs, as determined by the W_i for each VC. When X is between L

and H, the scheme also drops low priority (CLP=1) packets so as to ensure that sufficient buffer occupancy is available for CLP=0 packets.



The figure above illustrates the four operating regions of DFBA. The graph shows a plot of the current buffer occupancy X versus the normalized fair buffer occupancy for VC_i . If VC_i has a weight W_i , then its target buffer occupancy (X_i) should be $X \cdot W_i / W$. Thus, the normalized buffer occupancy of VC_i is $X_i \cdot W / W_i$. The goal is to keep this normalized occupancy as close to X as possible, as indicated by the solid line in the graph. Region 1 is the underload region, in which the current buffer occupancy is less than the low threshold L . In this case, the scheme tries to improve efficiency. Region 2 is the region with mild congestion because X is above L . As a result, any incoming packets with $CLP=1$ are dropped. Region 2 also indicates that VC_i has a larger buffer occupancy than its fair share (since $X_i > X \cdot W_i / W$). As a result, in this region, the scheme drops some incoming $CLP=0$ packets of VC_i , as an indication to the VC that it is using more than its fair share. In region 3, there is mild congestion, but VC_i 's buffer occupancy is below its fair share. As a result, only $CLP=1$ packets of a VC are dropped when the VC is in region 3. Finally, region 4 indicates severe congestion, and EPD is performed here.

In region 2, the packets of VC_i are dropped in a probabilistic manner. This drop behavior is controlled by the parameter Z_i , whose value depends on the connection characteristics. This is further discussed below. The figure below illustrates the drop conditions for DFBA.

The probability for dropping packets from a VC when it is in region 2 depends on several factors. The drop probability has two main components – the fairness component, and the efficiency component. Thus, $P\{\text{drop}\} = \text{fn}(\text{Fairness component}, \text{Efficiency component})$. The contribution of the fairness component increases as the VC's buffer occupancy X_i increases above its fair share. The contribution of the efficiency component increases as the total buffer occupancy increases above L . Since we assume that the system is linear between regions L and H , we choose to increase the drop probability linearly as X_i increases from $X \cdot W_i / W$ to X , and as X increases from L to H . As a result, the drop probability is given by

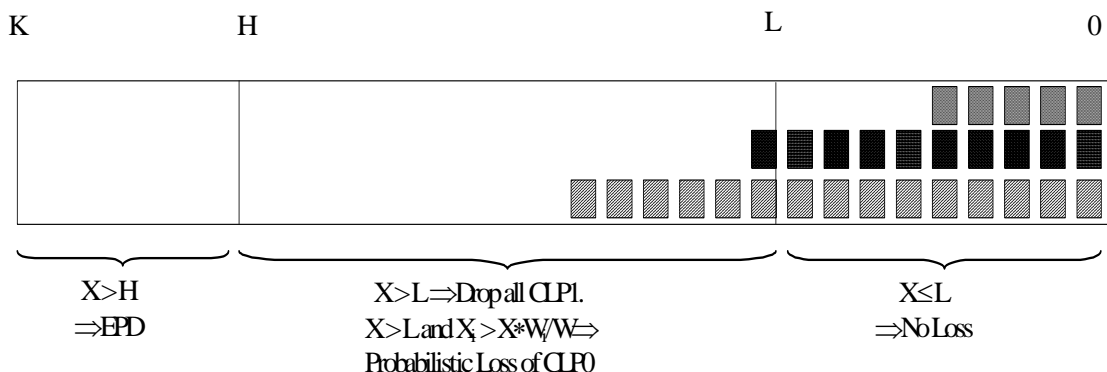
$$P\{\text{drop}\} = Z_i \left(\alpha \frac{X_i - X \times W_i / W}{X(1 - W_i / W)} + (1 - \alpha) \frac{X - L}{H - L} \right)$$

The parameter α is used to assign appropriate weights to the fairness and efficiency components of the drop probability. Z_i allows the scaling of the complete probability function based on per-VC characteristics.

It is well known that for a given TCP connection, a higher packet loss rate results in a lower average TCP window. As a result, a higher drop probability also results in a lower TCP window. In fact, it has been shown that for random packet loss, the average TCP window size is inversely proportional to the square root of the packet loss probability. As a result,

$$\text{TCP data rate} \propto \frac{\text{MSS}}{\text{RTT} \times \sqrt{P\{\text{drop}\}}}$$

This relationship can have a significant impact on TCP connections with either a high data rate or a large latency or both. To maintain high TCP data rate or when the RTT is large, one must choose a large TCP MSS, and/or must ensure that the average loss rate is low. As a result, DFBA can be tuned to choose a small Z_i for large latency VCs, as in the case of satellite VCs, or for VCs with high MCRs.



The following DFBA algorithm is executed when the first cell of a frame arrives at the buffer.

```

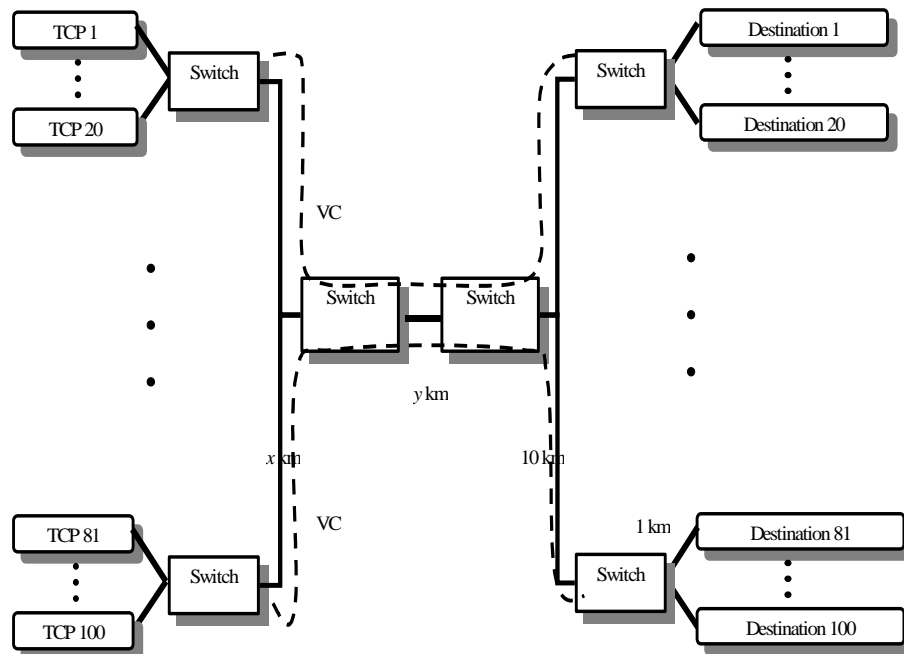
BEGIN
  IF (X < L) THEN
    Accept frame
  ELSE IF (X > H) THEN
    Drop frame
  ELSE IF ((L < X < H) AND (Xi < X*Wi/W)) THEN
    Drop CLP1 frame
  ELSE IF ((L < X < H) AND (Xi > X*Wi/W)) THEN
    Drop CLP1 frame
    Drop CLP0 frame with
      
$$P\{drop\} = Z_i \left( \alpha \frac{X_i - X \times W_i / W}{X(1 - W_i / W)} + (1 - \alpha) \frac{X - L}{H - L} \right)$$

  ENDIF
END

```

4 Simulation Configuration

We tested DFBA for ATM interconnected LANs with several scenarios. The following figure illustrates the basic test configuration. The figure shows 5 local switch pairs interconnected by two backbone switches that implement GFR. Each local switch carries traffic from multiple TCPs as shown in the figure. The backbone link carries 5 GFR VCs,



one from each local network. Each VC thus carries traffic from several TCP connections. The length of the local hop is denoted by x km, and the length of the backbone hop is denoted by y km. In this contribution, we present results with x=10 km and y=1000 km. The GFR capacity was fixed to the link rate of 155.52 Mbps (≈ 353,207 cells per sec). α was fixed to 0.5 in this study. All TCP sources were persistent TCPs with SACK. The SACK implementation is based on [FALL]. In our simulations, we varied four key parameters:

1. *Number of TCPs.* We used 10 TCPs per VC and 20 TCPs per VC for a total of 50 and 100 TCPs respectively.
2. *Per-VC MCR allocations.* Two sets of MCRs were chosen. In the first set, the MCR values were 12, 24, 36, 48 and 69 kcells/sec for VCs 1...5 respectively. This resulted in a total MCR allocation of about 50% of the GFR capacity. In the second set, the MCRs were 20, 40, 60, 80 and 100 kcells/sec for VCs 1...5 respectively, giving a total MCR allocation of 85% of the GFR capacity.
3. *Buffer size.* We first used a large buffer size of 25 kcells in the bottleneck backbone switch. We also analyzed DFBA performance with buffer sizes of 6 kcells, and 3 kcells.
4. Z_i . In most cases, the value of Z_i was chosen to be 1. We studied the effect of Z_i by decreasing it with increasing W_i .

5 Simulation Results

Table 1 shows achieved throughput for a 50 TCP configuration. The total MCR allocation is 50% of the GFR capacity. The W_i values for the VCs are 0.034, 0.068, 0.102, 0.136, and 0.170. The “achieved throughput” column shows the total end to end TCP throughput for all the TCP’s over the respective VC. The table shows that the VCs achieve the guaranteed MCR. Although the VCs with larger MCRs get a larger share of the unused capacity, the last column of the table indicates that the excess bandwidth is however not shared in proportional to MCR. This is mainly because the drop probabilities are not scaled with respect to the MCRs, i.e., because $Z_i = 1$ for all i . The total efficiency (achieved throughput over maximum possible throughput) is close to 100%.

Table 1 50 TCPs, 5 VCs, 50% MCR Allocation

MCR	Achieved Throughput	Excess	Excess / MCR
4.61	11.86	7.25	1.57
9.22	18.63	9.42	1.02
13.82	24.80	10.98	0.79
18.43	32.99	14.56	0.79
23.04	38.60	15.56	0.68
69.12	126.88	57.77	

Table 2 illustrates the performance of DFBA when 85% of the GFR capacity is allocated as the MCR values. In this case, the W_i 's are 0.057, 0.113, 0.17, 0.23, and 0.28 for VC's 1...5 respectively. The table again shows that DFBA meets the MCR guarantees for VCs carrying TCP/IP traffic.

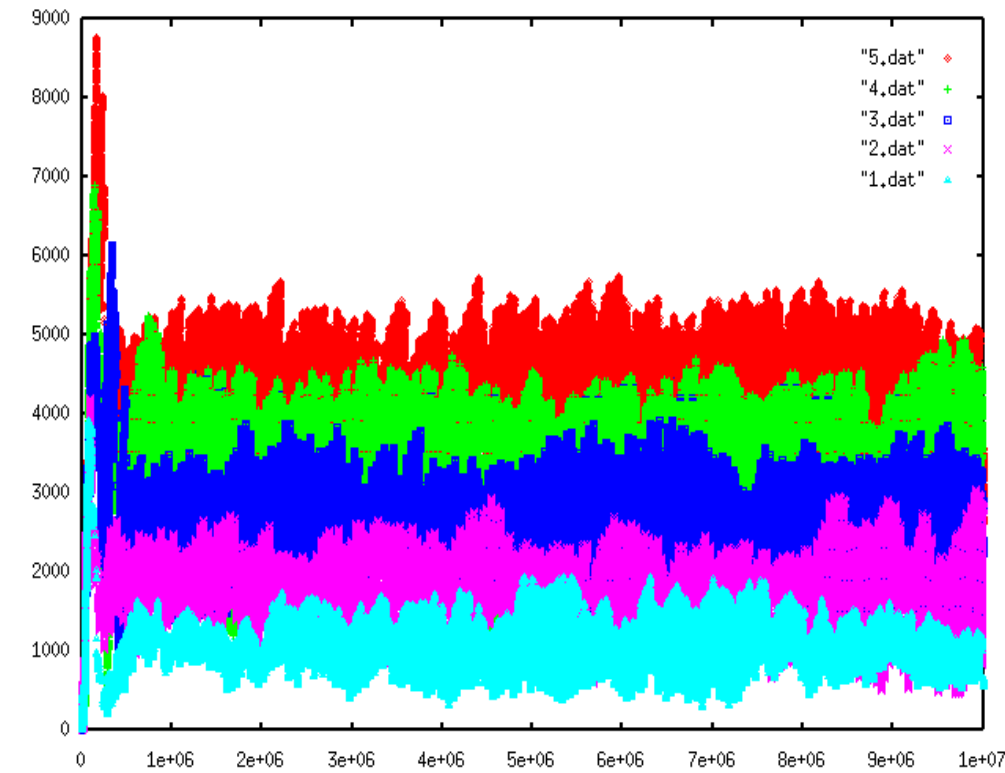
Table 2 50 TCPs, 5 VCs, 85% MCR Allocation

MCR	Achieved Throughput	Excess	Excess/MCR
7.68	12.52	4.84	0.63
15.36	18.29	2.93	0.19
23.04	25.57	2.53	0.11
30.72	31.78	1.06	0.03
38.40	38.72	0.32	0.01
115.2	126.88	11.68	

Table 3 validates the scheme for a larger number of TCPs. Each VC now carries traffic from 20 TCP connections, for a total of 100 TCPs. The total MCR allocation is 85% of the GFR capacity. All MCRs guarantees are met for a large number of TCPs and high MCR allocation.

Table 3 100 TCPs, 5 VCs, 85% MCR Allocation

MCR	Achieved Throughput	Excess	Excess/MCR
7.68	11.29	3.61	0.47
15.36	18.19	2.83	0.18
23.04	26.00	2.96	0.13
30.72	32.35	1.63	0.05
38.40	39.09	0.69	0.02
115.2	126.92	11.72	



The figure above illustrates the buffer occupancies of the 5 VCs in the bottleneck backbone switch. The figure shows that DFBA controls the switch buffer occupancy so that VCs with a lower MCR have a lower buffer occupancy than VCs with a higher MCR. In fact the average buffer occupancies are in proportion to the MCR values, so that FIFO scheduling can ensure a long-term MCR guarantee.

Table 4 and Table 5 show that DFBA provides MCR guarantees even when the bottleneck backbone switch has small buffers (6 kcells and 3 kcells respectively). The configuration uses 100 TCPs with 85% MCR allocation.

Table 4 Effect of Buffer Size (6 kcells)

MCR	Achieved Throughput	Excess	Excess/MCR
7.68	10.02	2.34	0.30
15.36	19.31	3.95	0.26
23.04	25.78	2.74	0.12
30.72	32.96	2.24	0.07
38.40	38.56	0.16	0.00
115.2	126.63	11.43	

Table 5 Effect of Buffer Size (3 kcells)

MCR	Achieved Throughput	Excess	Excess/MCR
7.68	11.79	4.11	0.54
15.36	18.55	3.19	0.21
23.04	25.13	2.09	0.09
30.72	32.23	1.51	0.05
38.40	38.97	0.57	0.01
115.2	126.67	11.47	

Table 6 shows the effect of Z_i on the fairness of the scheme in allocating excess bandwidth. We selected 2 values of Z_i based on the weights of the VCs. In the first experiment, Z_i was selected to be $(1-W_i/W)$ so that VCs with larger MCRs have a lower Z_i . In the second experiment, Z_i was selected to be $(1-W_i/W)^2$. The table shows that in the second case, sharing of the excess capacity is closely related to the MCRs of the VCs. An analytical assessment of the effect of Z_i on the excess capacity allocation by DFBA is a topic of further study.

Table 6 Effect of Z_i

$Z_i = 1 - W_i/W$		$Z_i = (1 - W_i/W)^2$	
Excess	Excess/ MCR	Excess	Excess/ MCR
3.84	0.50	0.53	0.07
2.90	0.19	2.97	0.19
2.27	0.10	2.77	0.12
2.56	0.08	2.39	0.08
0.02	0.02	3.14	0.08

6 A Framework for Buffer Management Schemes

Several recent papers have focused on fair buffer management schemes for TCP/IP traffic. All these proposals drop packets when the buffer occupancy exceeds a certain threshold. The proposals for buffer management can be classified into four groups based on whether they maintain multiple buffer occupancies (Multiple Accounting -- MA) or a single global buffer occupancy (Single Accounting -- SA), and whether they use multiple discard thresholds (Multiple Thresholds -- MT) or a single global discard Threshold (Single Threshold -- ST). The SA schemes maintain a single count of the number of cells currently in the buffer. The MA schemes classify the traffic into several classes and maintain a separate count for the number of cells in the buffer for each class. Typically, each class corresponds to a single connection, and these schemes maintain per-connection occupancies. In cases where the number of connections far exceeds the buffer size, the added over-head of per-connection accounting may be very expensive. In this case, a set of active connections is defined as those connections with at least one packet in the buffer, and only the buffer occupancies of active connections are maintained.

Schemes with a global threshold (ST) compare the buffer occupancy(s) with a single threshold and drop packets when the buffer occupancy exceeds the threshold. Multiple thresholds (MT) can be maintained corresponding to classes, connections, or to provide differentiated services. Several modifications to this drop behavior can be implemented. Some schemes like RED and FRED compare the average(s) of the buffer occupancy(s) to the threshold(s). Some like EPD maintain static threshold(s) while others like FBA maintain dynamic threshold(s). In some schemes, packet discard may be probabilistic (as in RED) while others drop packets deterministically (EPD/PPD). Finally, some schemes may differentiate packets based on packet tags. Examples of packet tags are the CLP bit in ATM cells or the TOS octet in the IP header of the IETF differentiated services architecture. Table 7 lists the four classes of buffer management schemes and examples of schemes for these classes. The example schemes are briefly discussed below.

The first SA-ST schemes included Early Packet Discard (EPD), Partial Packet Discard (PPD) [ROMANOV95] and Random Early Detection (RED) [FLOYD93]. EPD and PPD improve network efficiency because they minimize the transmission of partial packets by the network. Since they do not discriminate between connections in dropping packets, these schemes are unfair in allocating bandwidth to competing connections. For example when the buffer occupancy reaches the EPD threshold, the next incoming packet is dropped even if the packet belongs to a connection that is has received an unfair share of the bandwidth. Random Early Detection (RED) maintains a global threshold for the average queue. When the average queue exceeds this threshold, RED drops packets probabilistically using a uniform random variable as the drop probability. The basis for this is that uniform dropping will drop packets in proportion to the input rates of the connections. Connections with higher input rates will lose proportionally more packets than connections with lower input rates, thus maintaining equal rate allocation.

Table 7 Classification of Buffer Management Schemes

Group	Examples	Threshold Type (Static/Dynamic)	Drop Type (Deterministic/Prbabilistic)	Tag/TOS Sensitive (Yes/No)
SAST	EPD, PPD	Static	Deterministic	No
	RED	Static	Probabilistic	No
MAST	FRED	Dynamic	Probabilistic	No
	Selective Drop, FBA, VQ+Dynamic EPD	Dynamic	Deterministic	No
MAMT	PME+ERED	Static	Probabilistic	Yes
	DFBA	Dynamic	Probabilistic	Yes
	VQ+MCR scheduling	Dynamic	Deterministic	No
SAMT	Priority Drop	Static	Deterministic	Yes

However, it has been shown in [LIN97] that proportional dropping cannot guarantee equal bandwidth sharing. The paper also contains a proposal for Flow Random Early Drop (FRED). FRED maintains per-connection buffer occupancies and drops packets probabilistically if the per-connection occupancy exceeds the average queue length. In addition, FRED ensures that each connection has at least a minimum number of packets in the queue. In this way, FRED ensures that each flow has roughly the same number of packets *in the buffer*, and FCFS scheduling guarantees equal sharing of bandwidth. FRED can be classified as one that maintains per-connection queue lengths, but has a global threshold (MA-ST).

The Selective Drop (SD) ([GOYAL98b]) and Fair Buffer Allocation (FBA) [HEIN] schemes are MA-ST schemes proposed for the ATM UBR service category. These schemes use per-connection accounting to maintain the current buffer utilization of each

UBR Virtual Channel (VC). A fair allocation is calculated for each VC, and if the VC's buffer occupancy exceeds its fair allocation, its subsequent incoming packet is dropped. Both schemes maintain a threshold R , as a fraction of the buffer capacity K . When the total buffer occupancy exceeds $R \cdot K$, new packets are dropped depending on the VC's buffer occupancy (X_i). In these schemes, a VC's entire packet is dropped if

$$(X > R) \text{ AND } (X_i * N_a / X > Z) \text{ (Selective Drop)}$$

$$(X > R) \text{ AND } (X_i * N_a / X > Z * ((K - R)/(X - R))) \text{ (Fair Buffer Allocation)}$$

Where N_a is the number of active VCs (VCs with at least one cell the buffer), and Z is another threshold parameter ($0 < Z \leq 1$) used to scale the effective drop threshold.

The Virtual Queuing (VQ) [SIU97] scheme is unique because it achieves fair buffer allocation by emulating on a single FIFO queue, a per-VC queued round-robin server. At each cell transmit time, a per-VC accounting variable (X'_i) is decremented in a round-robin manner, and is incremented whenever a cell of that VC is admitted in the buffer. When X'_i exceeds a fixed threshold, incoming packets of the i th VC are dropped. An enhancement called Dynamic EPD changes the above drop threshold to include only those sessions that are sending less than their fair shares. Since the above MA-ST schemes compare the per-connection queue lengths (or virtual variables with equal weights) with a global threshold, they can only guarantee equal buffer occupancy (and thus throughput) to the competing connections. These schemes do not allow for specifying a guaranteed rate for connections or groups of connections. Moreover, in their present forms, they cannot support packet priority based on tagging.

Another enhancement to VQ, called MCR scheduling [WU97], proposes the emulation of a weighted scheduler to provide Minimum Cell Rate (MCR) guarantees to ATM connections. In this scheme, a per-VC weighted variable W_i is maintained, and compared with a global threshold. A time interval T is selected, at the end of which, W_i is incremented by $MCR_i * T$ for each VC i . The remaining algorithm is similar to VQ. As a result of this weighted update, MCRs can be guaranteed. However, the implementation of this scheme involves the update of W_i for each VC after every time T . To provide tight MCR bounds, a smaller value of T must be chosen, and this increases the complexity of the scheme. For best effort traffic (like UBR), thousands of VC could be sharing the buffer, and this dependence on the number of VCs may not be an efficient solution to the buffer management problem. Since the variable W_i is updated differently for each VC i , this is equivalent to having different thresholds for each VC at the start of the interval. These thresholds are then updated in the opposite direction of W_i . As a result, VQ+MCR scheduling can be classified as a MA-MT scheme. The Differential Fair Buffer Allocation Scheme discussed in this contribution is a MA-MT scheme as shown in Table 7.

[FENG] proposes a combination of a Packet Marking Engine (PME) and an Enhanced RED scheme based on per-connection accounting and multiple thresholds (MA-MT). PME+ERED is designed for the IETF differentiated services architecture, and can provide loose rate guarantees to connections. The PME measures per-connection bandwidths and probabilistically marks packets if the measured bandwidths are lower than the target bandwidths (multiple thresholds). High priority packets are marked, and low priority packets are unmarked. The ERED mechanism is similar to RED except that the probability of discarding marked packets is lower than that of discarding unmarked packets. The PME in a node calculates the observed bandwidth over an update interval by counting the number of accepted packets of each connection by the node. Calculating bandwidth can be complex since it may require averaging over several time intervals.

Although it has not been formally proven, Enhanced RED may suffer from the same problem as RED because it does not consider the number of packets actually in the queue.

A simple SA-MT scheme can be designed that implements multiple thresholds based on the packet priorities. When the global queue length (single accounting) exceeds the first threshold, packets tagged as lowest priority are dropped. When the queue length exceeds the next threshold, packets from the lowest and the next priority are dropped. This process continues until EPD/PPD is performed on all packets. The performance of such schemes needs to be analyzed. However, these schemes cannot provide per-connection throughput guarantees and suffer from the same problem as EPD, because they do not differentiate between overloading and underloading connections.

Table 8 illustrates the fairness properties of the four buffer management groups presented above.

Table 8 Properties of Buffer Management Schemes

Group	Equal bandwidth allocation	Weighted bandwidth allocation
SA-ST	6.1.1.1.1 No	No
MA-ST	Yes	No
MA-MT	Yes	Yes
SA-MT	-	-

7 References

[BASAK] Debashis Basak, Surya Pappu, "GFR Implementation Alternatives with Fair Buffer Allocation Schemes," ATM Forum 97-0528.

[BONAVENTURE97] Olivier Bonaventure, "A simulation study of TCP with the proposed GFR service category," DAGSTUHL Seminar 9725, High Performance Networks for Multimedia Applications, June 1997.

[BONAVENTUREb] Olivier Bonaventure, "Providing bandwidth guarantees to internetwork traffic in ATM networks," Proceedings of ATM'98, May 1998.

[CHAO] Dapeng Wu, H. J. Chao, "TCP/IP over ATM-GFR," Proceedings of ATM'98, May 1998.

[ELLOUMI] Omar Elloumi, Hossam Afifi, "Evaluation of FIFO based Buffer Management Algorithms for TCP over Guaranteed Frame Rate Service," Proceedings of ATM'98, May 1998.

[FALL] Kevin Fall, Sally Floyd, "Simulation-based Comparisons of Tahoe, Reno, and SACK TCP," Computer Communications Review, July 1996

[FENG] Wu-chang Feng, Dilip Kandlur, Debanjan Saha, Kang G. Shin, "Techniques for Eliminating Packet Loss in Congested TCP/IP Networks," _____.

[FLOYD93] Sally Floyd, Van Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transaction on Networking*, August 1993.

[GOYAL98b] Rohit Goyal, Raj Jain, et. al., "Improving the Performance of TCP over the ATM-UBR Service," To appear in *Computer Communications*, 1998.

[GOYALa] Rohit Goyal, Raj Jain et. al., "Design Issues for providing minimum rate guarantees to the ATM Unspecified Bit Rate Service," Proceedings of ATM'98, May 1998.

[GOYALb] Rohit Goyal, Raj Jain et. al, "Providing Rate Guarantees to TCP over the ATM GFR Service," To appear in the Proceedings of LCN'98, November 1998.

[HEIN] Juha Heinanen, Kalevi Kilkki, "A Fair Buffer Allocation Scheme," Unpublished Manuscript.

[LIN97] Dong Lin, Robert Morris, "Dynamics of Random Early Detection," Proceedings of SIGCOMM97, 1997.

[ROMANOV95] Allyn Romanov, Sally Floyd, "Dynamics of TCP Traffic over ATM Networks," *IEEE Journal of Selected Areas In Telecommunications*, May 1995.

[SIU97] Kai-Yeung Siu, Yuan Wu, Wenge Ren, "Virtual Queuing Techniques for UBR+ Service in ATM with Fair Access and Minimum Bandwidth Guarantee," Proceedings of Globecom'97, 1997.

[WU97] Yuan Wu, Kai-Yeung Siu, Wenge Ren, "Improved Virtual Queuing and Dynamic EPD Techniques for TCP over ATM," Proceedings of ICNP97, 1997.