ATM-GFR Buffer Management and Scheduling Issues: A Performance Study Using TCP

S. Puangpronpitag, M. Kara and K. Djemame
School of Computing
University of Leeds – Leeds LS2 9JT, UK
Email: {nuk, mourad, karim}@comp.leeds.ac.uk

Abstract

We overview in this paper the buffer management and scheduling issues and draw upon published work to illustrate the possible alternatives of GFR implementation. Performance evaluation has also been done on ATM-GFR's buffer management algorithms and scheduling algorithms. The simulation results have shown that two-static-threshold-buffer-management and per-VC-buffer management give the best performance comparing with plain UBR, UBR-EPD and per-VC scheduling. We have also found that the two-threshold buffer management scheme is significant to handle CLP0+1 traffic. This scheme can distinguish between the tagged (low priority) and untagged (high priority) packets of the traffic and protect the bandwidth from flushing by tagged packets. The per-VC scheduling without the sensitivity of tagged/untagged frames can lead to the poor performance for untagged (high priority) packets.

1. Introduction

Asynchronous Transfer Mode (ATM) Technology is a high-speed network technology. It aims at providing the quality of service to support voice, video and multimedia traffic as part of the work on broadband Integrated Service Digital Networks (B-ISDN). Yet, the majority of traffic is still in the form of normal data, especially as TCP packets. The main use of ATM at the present is therefore in non-ISDN environments to support TCP packets. Consequently, the co-operation between these technologies (TCP and ATM) has become a very significant issue.

Many studies have divulged the poor performance of running TCP over ATM. There are many proposals to solve this problem such as improving the buffer management scheme of ATM switch [RF95][FL95][GJF97a][GJF97b], improving congestion algorithms of TCP [MMF96][BK97], and providing a co-operation scheme between these two technologies [DK00a][DK00b]. One of the solutions is the minimum guaranteed rate, leading to the new category of ATM called Guarantee Frame Rate (GFR).

The implementation of GFR has been the focus of many studies recently. In this paper, our aim is to draw upon published work to illustrate the specification and the possible alternatives of its implementation. To do so, we analyse many recent published works on GFR and implement some buffer management algorithms. The experiments have been simulated to compare the performance among those algorithms.

We begin in section 2 discussing ATM services for TCP traffic and the motivation of GFR. In section 3, we analyse several published works and conclude on the alternatives of GFR implementation. The simulation model, tools, and performance metrics for the experiments have been presented in section 4. Section 5 discusses the performance results of the algorithms over LAN and WAN distances. In section 6, the conclusions of this work are given.

2. ATM Services for Data Traffic: ABR, UBR and GFR

Among the various categories of ATM service, ABR (Available Bit Rate) and UBR (Unspecified Bit Rate) have been introduced to support data traffic such as TCP packets. The ABR service relies on the rate-based scheme, which requires complicated rules for source behaviour and uses a special Resource Management (RM) cell to indicate the current state of congestion.

The UBR service is the lowest priority service as well as the cheapest service for transporting TCP packets. It does not guarantee any cell loss and cell delay performance but lets the higher layer protocol (i.e. TCP) encounter these effects. If there is capacity left over, UBR cells will be delivered. However, later when the switch buffers become full, the UBR cell(s) will be simply dropped out from the switch without any feedback to the sender and expectation to slow down.

Although both ABR and UBR can support TCP traffic, most TCP traffic over ATM still relies on the UBR category only due to the complicated rules required by the ABR source behaviour (for the rate based feedback control) [Bon97]. Many studies (such as [RF95] and [GJF97a]) have shown that UBR can provide very poor end-to-end performance. In the situation where a VBR (Variable Bit Rate) service is running on the same link, the problem becomes more serious. If the VBR traffic has strict priority higher than UBR, it can use up the entire link capacity and results in the bandwidth starvation of UBR traffic [GJF98a].

To fill the gap between UBR and ABR, Guerin and Heinanen have proposed a new service category called UBR+ (and later called Guaranteed Frame Rate) [ATM99]. The new category provides a minimum rate guarantee to VCs at the frame level to increase efficiency, and recommends the fair usage of any additional bandwidth left over from higher priority connection [ATM99]. It would be much better than UBR service in terms of end-to-end performance as providing the users with some level of service guarantees. At the same time, it is less complicated and has fewer overheads compared with ABR service.

3. Implementation Alternatives of GFR

GFR is a Frame-based service with minimum guaranteed rate while all other service categories of ATM are cell-based service. GFR design is more from the viewpoint of applications since most of applications rely on the Frame-based service of upper layer protocols more than the cell-based service of ATM categories.

In a GFR-capable switch, the complete frames are accepted or discarded. It uses AAL5 to make frame boundary visible at the ATM layer. The traffic contract of GFR consists of conformance definition and service eligibility, which are involved with the following parameters: (1) Maximum Frame Size (MFS), (2) Peak Cell Rate (PCR) and its Cell Delay Variation Tolerance ($CDVT_{PCR}$), (3) Minimum Cell Rate (MCR) and its Cell Delay Variation Tolerance ($CDVT_{MCR}$).

If the sources send packets smaller than the MFS, at a rate below or equal to the MCR, then all the packets would be delivered with minimum loss. If the sources send packets at a rate higher than the MCR, they would still be received (at least) with the minimum rate. The minimum rate is guaranteed to the untagged frames (CLP bit = 0) only. In addition, the source sending in excess of the MCR should also fairly share the unused network capacity [ATM99].

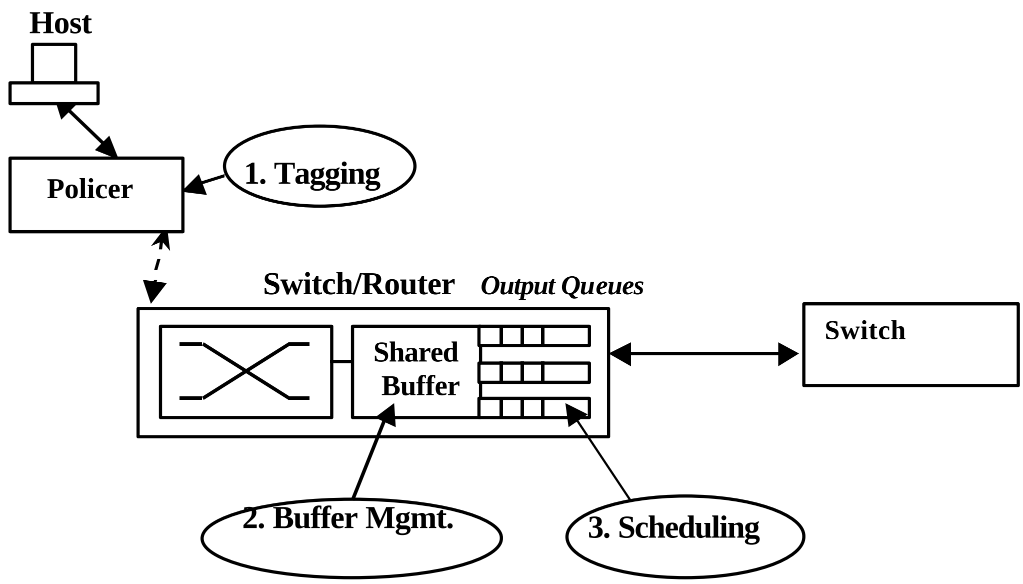


Figure 1: Main components for implementing GFR

There are three main components to provide frame guarantee [ATM99]: Per-VC Tagging, Frame-based Buffer Management and Scheduling. Per-VC network based Tagging is used to mark non-conforming packets before entering the network so that the conforming and non-conforming traffic of each VC can be isolated and later conforming packets will be scheduled in preference to non-conforming packets (figure 1) [GJF98a].

The buffer management is performed by network elements (switch or router) to control the number of packets entering its buffers. It is used as a mechanism for congestion avoidance and control including preferential dropping of tagged frames over untagged frames when mild congestion is experienced. Per-VC buffer management uses per-VC accounting to keep track of the buffer occupancies of each VC. This causes some overhead but may be necessary to provide fairness among the connections. There are a few per-VC buffer management schemes proposed such as Selective Drop (SD) [GJF97a], Fair Buffer Allocation (FBA) [FL95], Weighted Buffer Allocation (WBA) [GJF98a] and Differential Fair Buffer Allocation (DFBA) [GJF98b].

The scheduling determines how packets are scheduled onto the next hop. There are two schemes of scheduling: (1) Non-rate-guarantee service discipline such as FIFO (First In First Out) queue, Round robin queue; (2) Rate-guarantee service discipline such as WFQ (Weighted Fair Queue), WRR (Weighted Round Robin).

For FIFO queuing, the packets are scheduled in the order in which they enter the buffer and cannot be distinguished among various VCs. Per-VC queuing, on the other hand, maintains a separate queue for each VC in the buffer [ATM99]. Then the scheduling mechanism can select among the queues at each scheduling time. However, the cost of per-VC queuing is far more expensive than FIFO queuing. For the simple service like GFR, it is arguable whether this additional cost is desirable.

The possibility to provide GFR service from combining these three components has been studied by many researchers. We conclude from published work as shown in Table 1.

#	Per-VC Tagging	Per-VC buffer management	Per-VC queuing	Can provide GFR service?
1	No	No	No	**No**
2	Yes	No	No	**No**
3	No	Yes	No	**No conclusion**
4	No	No	Yes	**No**
5	Yes	Yes	No	**No conclusion**
6	No	Yes	Yes	**Yes**
7	Yes	No	Yes	**Yes**
8	Yes	Yes	Yes	**Yes**

Table 1: Alternatives of GFR implementation

From table 1, combination-1 obviously cannot provide GFR. Combination-2 was proposed by Guerin and Heinanen [ATM99] as a simple implementation of GFR without fairness of excess-bandwidth usage. However, many studies ([PB97a][Bon97][GJF98a]) have shown in their experiments that this combination is not enough to provide the bandwidth guarantee.

Combination-3 uses per-VC buffer management without per-VC tagging and without per-VC queuing. It is still questionable whether they can provide GFR service. The published works have provided some conflicting results. For example, [PB97b] and [GJF98a] run experiments using Fair Buffer Allocation (FBA) and Weighted Buffer Allocation (WBA) as the per-VC buffer management respectively. Their experiments show that per-VC buffer management alone (or with per-VC tagging) without per-VC queuing cannot support GFR. However, Goyal et al. propose the new per-VC buffer management algorithm called Differential Fair Buffer Allocation (DFBA) [GJF98b]. They show in their experiments [GJF98b] that DFBA without Networking Tagging and per-VC scheduling may be sufficient to provide rate guarantee under certain circumstances (SACK TCP, no other non-TCP or higher priority traffic, enough link capacity for the sum of Minimum Cell Rate guaranteed for all VCs).

For combination-5 (using per-VC queuing, with network tagging (GFR.2) without per-VC queuing), there are still no published results that show the possibility of providing GFR even for DFBA. According to [SB00], Spaey and Blondia assume that even using DFBA in this combination cannot provide guarantee for GFR since TCP may not be able to adapt its behaviour to the F-GCRA function. Together with per-VC tagging, a large percentage of the packets will be tagged and then discarded by DFBA during mild congestion. Hence, some sources cannot benefit from their minimum bandwidth guarantee.

For combination-4 (using per-VC queuing without per-VC tagging and per-VC buffer management), it has been shown by [PB97b] and [GJF98a] that all incoming packets from greedy sources sending untagged packets at full link bandwidth can quickly flood the buffer causing untagged packets from other sources to be dropped. Hence, some non-greedy sources can not get the minimum guarantee. Hence, per-VC queuing alone in the absence of per-VC buffer management and network tagging can not be sufficient to provide GFR service.

For combinations 6, 7, and 8 (using per-VC queuing with per-VC tagging and/or per-VC buffer management), several studies [PB97b][GJF98a][Bon97] have shown that they are clearly able to provide GFR service. Yet, the complication of per-VC queuing may not be desirable.

4. Experimental Setup

4.1 Simulation Model and Objectives

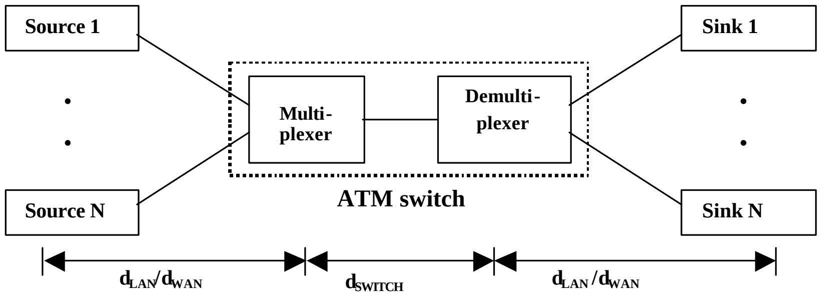


Figure 2: Simulation Model

The main purpose of the simulation is to observe and compare the performance improvement of the ideas proposed for GFR service relative to the plain UBR and the UBR with EPD. Considering three components of GFR implementation (Tagging, Buffer Management and Scheduling), our experiments are performed as follows:

- Tagging is enabled to generate both tagged and untagged traffic on the link. We focus on observing the performance improvement of the untagged (high priority) traffic during mild congestion.
- For the Buffer Management, the experiments mainly focus on two schemes that may be necessary for GFR: the two-threshold scheme and the per-VC accounting scheme (used for per-VC buffer management)
- For the Scheduling, the experiments are performed by using both FIFO and per-VC scheduling. WFQ algorithm (by J.W. Roberts) is picked up from various alternatives of per-VC scheduling [Bau97]. To observe the effect of WFQ to the CLP0+1 stream, we run experiment on WFQ alone without the support of any special buffer management.

Simulation is performed using the N source symmetrical configuration consisting of N identical TCP sources (figure 2). The link bandwidth is 155.52 Mbps (PCR =149.76 Mbps after SONET overhead). The traffic is unidirectional. Only the sources send data. The sinks send only ACK. The traffic in the network is simulated as CLP0+1 stream, in which half of VCs will tag CLP bit.

All TCP sources start sending at the same time to simulate the worst case. The TCP sources are greedy sources with an infinite supply of data and always have data ready to send as and when permitted by the TCP flow-control. TCP window size is set to 64 Kbytes. TCP packet size is set to 512 bytes/packet. TCP Fast Retransmit and Recovery, the timestamp option and Nagle's algorithm are enabled.

Two network scenarios (LAN and WAN) are considered. For the LAN scenario, the link delay ($d_{LAN} + d_{SWITCH} + d_{LAN}$) is set to 102 microseconds consisting of 11.3 microseconds of d_{SWITCH} (the internal switch delay) and 45.3 microseconds of d_{LAN} (the LAN delay). This is approximately equivalent to 20 km of transmission path. For WAN, the link delay is set to around one milliseconds ($d_{WAN} + d_{SWITCH} + d_{WAN}$) consisting of 11.3 microseconds of d_{SWITCH} and 498 microseconds of d_{WAN} (the WAN delay). This is approximately equivalent to

200 km of transmission path. We run the experiments using five different switches supporting as the following:

#	ATM switch support	To observe
1	The plain UBR service	as a base of comparison
2	The UBR with EPD	as a base of comparison
3	The plain UBR with two-static-threshold buffer management and FIFO queue	the effect of two-threshold-buffer-management-scheme
4	The plain UBR with per-VC buffer management (having two-threshold scheme and per-VC accounting) and FIFO queue	the effect of per-VC accounting scheme
5	WFQ	the effect of per-VC scheduling

Table 2: Various switches used in the experiments

For EPD, the threshold is set to 80% of switch buffer size. For two-static-threshold buffer management and per-VC buffer management, two static thresholds are set to 50% and 80% of switch buffer size in the order. The per-VC threshold of per-VC buffer management is set according to the following formula:

$$perVC\_Threshold = \max\left(2 * MaximumFrameSize),\frac{BufferSize}{NumberOfSources}\right.$$

For per-VC scheduling, the inverse cell rate of each VC (according to Weighted Fair Queuing algorithm) is set to 1/MCR [Bau97]. In order to induce variable congestion in the switches, the number of sources and the size of buffer are varied for the simulation cases.

4.2 Simulation Tool

The YATS simulation package is used for this experiment [Bau97]. It provides all the important TCP mechanisms and supports all the significant ATM features. However, the simulator does not support GFR. Several objects therefore need to be developed and modified to integrate buffer management algorithms used for GFR into YATS. We developed two new multiplexers by using two-static-threshold-buffer-management and per-VC-buffer-management algorithms. These algorithms are parts of implementations of GFR proposed by Guerin and Heinanen [ATM99]. Figure 3 shows the drop policy of this two-threshold-scheme.

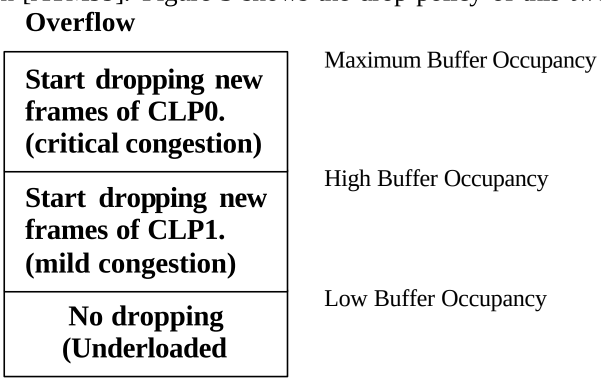


Figure 3: Drop Policy of two-thresholds-scheme

This scheme is also known as Double Early Packet Discard (D-EPD). Two thresholds (Low Buffer Occupancy (LBO) and High Buffer Occupancy (HBO)) are static and sensitive to the CLP bit. For the low priority stream, the CLP bit will be tagged (CLP1 stream). When the buffer occupancy reaches LBO (mild congestion), the EPD drop policy will be applied to low priority CLP1 stream (tagged packets). This gives the buffer space for CLP0 (high priority) stream and protect the problem of flushing buffer by CLP1 (low priority) stream. If the buffer occupancy reaches HBO (critical congestion), the EPD drop policy will be applied to CLP0 stream for two-threshold-buffer-management. However, for per-VC buffer management, the buffer occupancy of VC will be considered. If the buffer occupancy of VC is over per-VC threshold, the EPD drop policy is then applied to CLP0 stream of that VC.

From the original proposal, the per-VC buffer management would be implemented with WFQ. However, we implemented it alone without WFQ since the purpose of our experiment is to observe the performance improving from only the buffer management schemes without WFQ. There are also some changes in AAL5 Sender, Cell object and other parts of YATS source code to provide CLP (Cell Loss Priority) bit tagging option. The modified part has also been well validated. Full details of source code implementation and validation are available in [Pua99]. Traffic is generated by YATS specific objects using a TCP/IP/AAL5/ATM protocol stack.

4.3 Performance Metrics

The performance metrics (used in the evaluation of simulation results) are the throughput, the cell loss ratio (CLR) and the TCP packet retransmission ratio (PRR).

- CLR is defined as the ratio of number of cells dropped at the switches to the total number of cells sent by the source during the simulation (expressed as a percentage). It is measured at the AAL5 receiver of the ATM layer. A low CLR indicates efficiency of congestion control in the ATM layer.
- PRR is defined as the ratio of total number of TCP packets retransmitted to the total number of TCP packets sent by the TCP source (expressed as a percentage). It is measured at the TCP receiver.
- The average throughput per connection gives an indication of the fraction of the full bandwidth used to transfer TCP packets successfully from source to sink. Only the number of good (i.e., error-free data) bytes received at the sinks during the total transmission time are considered. No duplicate data bytes received at the sink are considered.

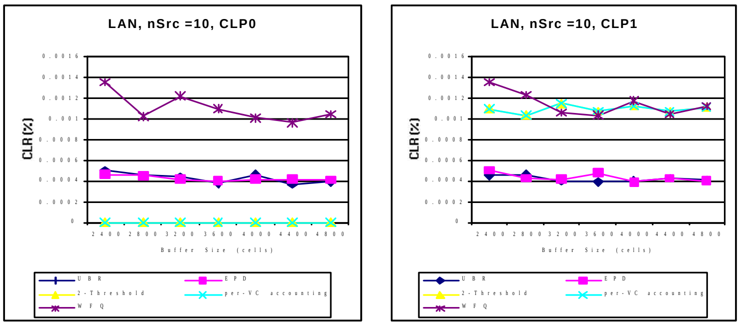
5. Results and Evaluation

In this section, the simulation results from different variables in section 4.1 and performance metrics in section 4.3 are presented. The focus is on the assessment of performance among different algorithms (UBR-EPD, two-threshold buffer management, per-VC buffer management and WFQ as well as plain UBR). In particular, we focus on the performance of the higher priority traffic (CLP0 stream) in the situation that both tagged and untagged packets appear together on the link (CLP0+1 stream).

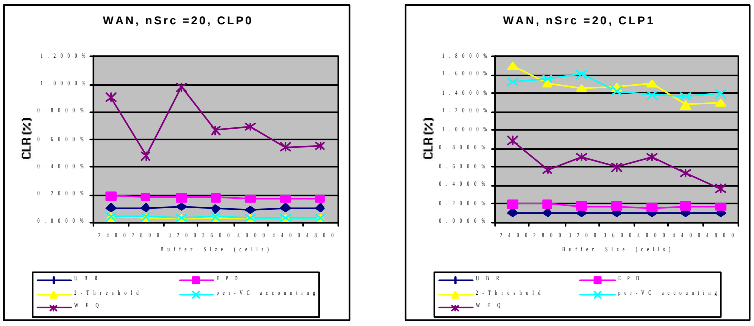
From the experiments, we find both LAN and WAN scenarios giving the same trend of results. Overall, the plain UBR and per-VC scheduling by WFQ give the worst results comparing with other switches. With EPD, the performance is better than UBR as expected. Two-threshold buffer management and per-VC-Buffer-Management algorithms give the best performance (in term of CLR, PRR and throughput) comparing with the other. It must be noted that these two algorithms seem to give similar results for our scenarios since they are actually based on the same two-threshold-scheme. The per-VC buffer management has just adapted by per-VC-accounting scheme to improve the fairness, which is not considered in our experiments. WFQ without sensitivity to tagged or untagged packets cannot help improve performance of UBR on the situation of having CLP0+1 stream on the link.

5.1 Varying the Buffer Size

From the experiment results, two-threshold buffer management and per-VC buffer management algorithms have consistently achieved the lowest CLR for high priority bandwidth (CLP0 stream) comparing with plain UBR, UBR-EPD and WFQ. This would indicate that the two-threshold scheme is necessary to improve performance of UBR in the situation of CLP0+1 stream (i.e., tagging is on).



(4a) LAN scenario



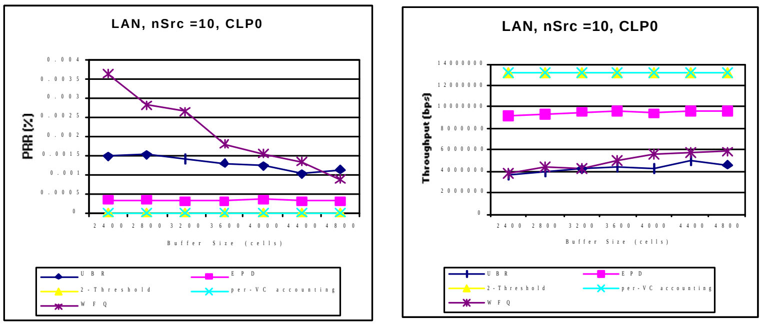
(4b) WAN scenario

Figure 4: Comparing CLR of CLP0 and CLP1 stream

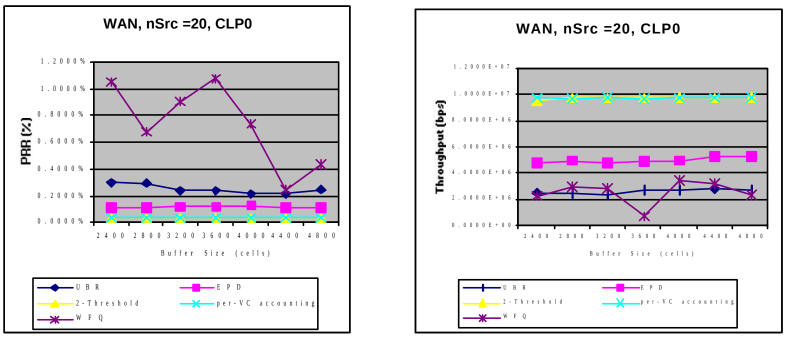
Figure 4a shows the result of LAN scenario using 10 TCP sources and varying buffer size from 2400 to 4800 cells. Figure 5b shows the result of WAN scenario using 20 TCP sources and varying buffer size from 2400 to 4800 cells. By comparing the results in both figures, we can notice the different treatments of each algorithm to CLP1 stream (the tagged packets) and CLP0 stream (the untagged packets). For two-threshold buffer management and per-VC buffer management, the CLR of CLP0 stream and CLP1 stream are different since they treat the CLP0 stream as higher priority traffic than CLP1 stream. In LAN scenario (figure 4a), the CLR of CLP0 stream is negligible while the CLR of CLP1 stream varies from 10^{-3} % to 1.2×10^{-3} %. In WAN scenario (figure 5b), the CLR of CLP0 stream is negligible while the CLR of CLP1 stream varies from 1.2 % to 1.7%

However, plain UBR, UBR-EPD and WFQ have similar CLR for both CLP0 and CLP1 streams. In LAN scenario (figure 4a), CLR of plain-UBR and UBR-EPD are around $5x10^{-4}$ % for both CLP0 and CLP1 stream. CLR of WFQ is also the same for both CLP0 and CLP1 stream (varying from $10^{-}10^{-3}$ % to $1.4x10^{-3}$ %). In WAN scenario (figure 4b), CLR of plain-UBR and UBR-EPD are around 0.1 - 0.2% for both CLP0 and CLP1 stream and CLR of WFQ is varying from 0.4% to 0.9% for both CLP0 and CLP1 stream.

This matches with our expectation. Since two-threshold buffer management and per-VC buffer management have two thresholds for different priorities of traffic (LBO and HBO for tagged packets and untagged packets respectively), they therefore can treat CLP0 stream (the high priority traffic) in preference to CLP1 (the low priority traffic). When there is a mild congestion and the occupancy of switch buffer reaches LBO, CLP1 packets will be dropped to reduce the congestion before becoming serious and causing CLP0 packets to be dropped. Yet, for plain UBR, UBR-EPD and WFQ, there is no clue for them to distinguish between tagged and untagged packets. Hence, they just treat both high (untagged) and low (tagged) priority packets equally.



(5a) LAN scenario



(5b) WAN scenario

Figure 5: Comparing PRR and Throughput of

Figure 5 compares the results of PRR and throughput in both LAN and WAN scenarios. The results for PRR are similar to the results of CLR since cells dropping in the ATM layer generally causes TCP packet to be retransmitted. Two-threshold buffer management and per-

VC-buffer management can achieve the lowest PRR for untagged stream in both LAN and WAN scenarios as shown in the figure.

In figure 5, throughput has been improved with the larger buffer sizes due to the enhanced ability of the ATM switch to tackle congestion. In LAN and WAN scenarios, two-threshold buffer management and per-VC buffer management also gain the highest throughput for untagged traffic by virtue of the lowest CLR and PRR.

In figures 4 and 5, CLR of UBR-EPD is a bit higher than plain UBR but the PRR is lower. The throughput of UBR-EPD is also higher than plain UBR as expected. By using EPD,

when the congestion reaches the threshold level, but before requiring to discard any cells due to the buffer overflow, an entire frame is dropped. This may cause CLR of EPD being a bit higher than CLR of plain UBR. Since EPD drops all cells of one packet rather than randomly drop cells from different packets, the PRR and throughput are therefore better than plain UBR.

From the experiment results, WFQ does not seem to help improving UBR and gives even worse performance in some simulations. This is because WFQ cannot distinguish between tagged and untagged traffic in CLP0+1 stream. Hence, it treats CLP0 and CLP1 streams with the same priority and can give the very poor performance when the CLP1 stream flushes the bandwidth. We tend to conclude that the per-VC queuing algorithm without sensitivity to tagging can experience low performance for CLP0 (high priority) stream since it can be disturbed by the CLP1 (low priority) stream.

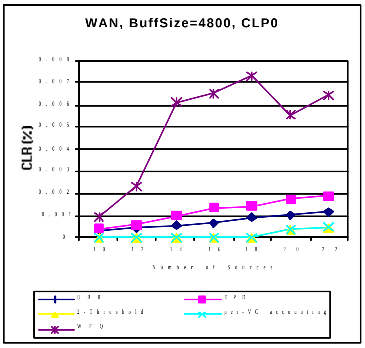
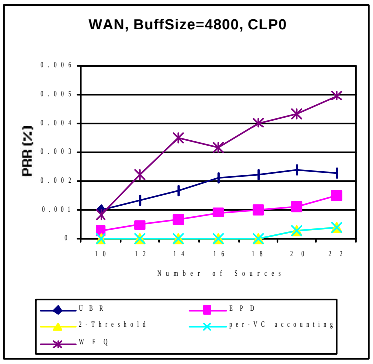
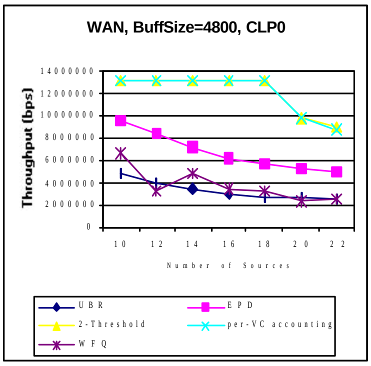


Figure 6:
Varying the number
of sources, WAN scenario

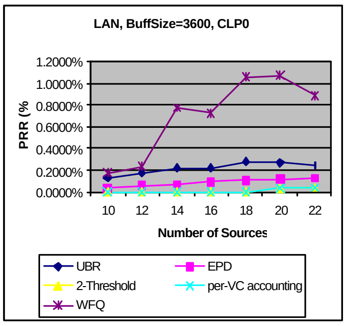




5.2 Varying the Number of Sources

Figures 6 and 7 are the experiment results of varying the number of sources in WAN and LAN environment respectively. They show the comparison of CLR, PRR and throughput among five switches (plain UBR, UBR-EPD, WFQ, per-VC buffer management and two-threshold buffer management). In figure 6 (WAN environment), the switch buffer size is set to 4800 cells and the number of TCP sources varies from 10 to 22. In figure 7 (LAN environment), the switch buffer size is set to 3600 cells and the number of TCP sources varies from 10 to 22.

As expected, the throughput decreases when the number of sources increases for all scenarios. This is obvious because more sources cause more congestion. Similar to the results in section 5.1, two-threshold-buffer-management and per-VC-buffer-management consistently give the best performance (the lowest CLR and PRR, the highest throughput). UBR-EPD gives better throughput comparing with plain UBR. WFQ alone without sensitivity of frame tagging gives almost no improvement of performance over plain UBR since the low priority (tagged) traffic can flush bandwidth and cause the high priority (untagged) traffic to suffer from bandwidth starvation.



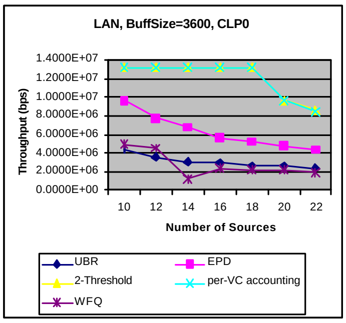


Figure 7: Varying the number of sources, LAN scenario

6. Conclusions

The GFR specification and its recent critical issues have been drawn out based on published materials. The alternatives of its implementation have also been discussed and analysed. Per-VC Tagging, Frame-based Buffer and Management and Scheduling are the main components to implement a GFR service. Using per-VC queuing together with per-VC tagging and/or per-VC buffer management has been shown by many studies to provide minimum guaranteed rate. However, the complexity and cost of per-VC queues are undesirable. So far, there are not any intelligent buffer schemes (without per-VC queuing) can fully provide rate guarantee for GFR.

The simulation study has confirmed that the two-threshold scheme is vital to handle CLP0+1 traffic to protect the bandwidth from flushing by the CLP1 (low priority) stream. It can help improve the performance for untagged packets (the higher priority traffic) if the link has both tagged and untagged packets that need to be treated in different priorities (i.e., tagging is on). On our experiments, the per-VC accounting does not help improve throughput, CLR and PRR. The per-VC queuing without the sensitivity of tagged/untagged packets can also give very poor performance for CLP0 (high priority) stream in case greedy CLP1 (low priority) sources exist.

7. References

[ATM99] ATM Forum Technical Committee. Traffic Management Specification version 4.1, March 1999.

[Bau97] M. Baumann. YATS User's and Programmer's Manual for Version 0.3. Dresden University of Technology, September 1997.

[Bon97] O. Bonaventure. A simulation study of TCP with the proposed GFR service category. DAGSTUHL Seminar 9725 Dagstuhl, Germany. June 1997.

[BK97] A. Bestavros and G. Kim. TCP Boston - A Fragmentation-tolerant TCP Protocol for ATM Networks. Proceedings of IEEE INFOCOM'97. 1997.

[DK00a] K. Djemame and M. Kara. TCP Explicit Congestion Notification over ABR and UBR: an Agent-Based Approach. Proceedings of the 3rd IEEE ATM & ICATM '2000, Heidelberg, Germany, pp. 215-223, June 2000.

[DK00b] K. Djemame and M. Kara. A New Rate Co-ordination Protocol for TCP and ABR Congestion Control. Proceedings of the IEEE International Conference on Networks (ICON'2000), Singapore, pp. 44-48, September 2000.

[DKB00] K. Djemame, M. Kara and R. Banwait. An Agent Based Congestion Control and Notification Scheme for TCP over ABR. Journal of Computer Communications. Vol. 23, Issue 16, pp. 1524-1536, September 2000.

[FL95] C. Fang and A. Lin. On TCP Performance of UBR with EPD and UBR-EPD with a Fair Buffer Allocation Scheme. ATM Forum contribution ATM95-1645, December 1995.

[GJF97a] R. Goyal, R. Jain and S. Fahmy. UBR+: Improving Performance of TCP over ATM-UBR service. ICC'97 Montreal. June 1997.

[GJF97b] R. Goyal, R. Jain, S. Fahmy and B. Vandalore. GFR: Providing Rate Guarantees with FIFO Buffers to TCP Traffic. ATM Forum contribution ATM97-0831, September 1997.

[GJF98a] R. Goyal, R. Jain, S. Fahmy, and et al. Design Issues for providing Minimum Rate Guarantees to the ATM Unspecified Bit Rate Service. May 1998.

[GJF98b] R. Goyal, R. Jain, S. Fahmy and B. Vandalore. Buffer Management for TCP over the ATM GFR Service. ATM Forum contribution ATM98-0405, July 1998.

[LH98] M. Laubach and J. Halpern. Classical IP and ARP over ATM. RFC-2225, April 1998.

[MMF96] M. Mathis, J. Mahdavi, S. Floyd and A. Ramanow, TCP Selective Acknowledgement Options. RFC-2018, April 1996.

[PB97a] S. K. Pappu and D. Basak. ATM over GFR with different service discipline: A simulation study. ATM Forum contribution ATM97-0310, May 1997.

[PB97b] S. K. Pappu and D. Basak. GFR implementation alternatives with Fair Buffer Allocation Schemes. ATM Forum contribution ATM97-0528, July 1997.

[Pua99] S. Puangpronpitag. Performance Analysis of TCP congestion control over ATM Networks. MSc. Report 1998/99, School of Computer studies, University of Leeds. September 1999.

[RF95] A. Romanow and S. Floyd, Dynamics of TCP Traffic over ATM Networks. IEEE Journal on selected area in communications. May 1995.

[SB00] K. Spaey and C. Blondia. Buffer acceptance schemes for the UBR and GFR ATM service categories. Proceedings of the 8th IFIP Workshop on Performance Modelling and Evaluation of ATM and IP Networks, pp.10/1-10/40, July 2000.