A Performance Evaluation of Buffer Management and Scheduling for ATM- GFR Using TCP

S. Puangpronpitag, M. Kara and K. Djemame

ATM-Multimedia Group
http://www.scs.leeds.ac.uk/atm-mm
School of Computer Studies
University of Leeds – Leeds LS2 9JT, UK
Email: {nuk, mourad, karim}@scs.leeds.ac.uk

Abstract

Among the various categories of ATM (Asynchronous Transfer Mode) service, ABR (Available Bit Rate) and UBR (Unspecified Bit Rate) have been introduced to support data traffic (such as TCP). GFR (Guarantee Frame Rate) is the current proposed ATM service to provide better performance for data traffic over ATM. The aim of this paper is two-folded. Firstly, we characterise the major research issues in GFR and draw upon the published works to illustrate some of the critical issues about the specification and the possible alternatives of its implementation. Secondly, we undertake a performance evaluation of ATM-GFR's buffer management algorithms (two-static-threshold scheme and per-VC accounting) and scheduling (FIFO (First In First Out) and WFQ (Weighted Fair Queue)). 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 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.

Keywords: GFR performance evaluation, Transmission Control Protocol over ATM Networks, Congestion Control

1. Introduction

Due to the tremendous growth of the Internet, the Transmission Control Protocol (TCP) has been the most ubiquitous transport layer protocol and is considered as the standard in data communication. It offers a connection-oriented, reliable end-to-end transport service by using a window based flow and error control. 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 Network (B-ISDN) [Sta98]. 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 both 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 buffer management scheme of ATM switch [RF95][FL95][GJF97a][GJF97b], improving congestion algorithms of TCP [MMF96][BK97], and providing the co-operation scheme between these two technologies [DK99][DK00a][DK00b]. One of the solutions is the minimum guaranteed rate [GH96], leading to the new category of ATM called Guarantee Frame Rate (GFR).

The implementation and specification of GFR have been the main focus of many studies recently. In this paper, our aim is to characterise the major research issues and draw upon the published work to illustrate some critical issues about the specification and the possible alternatives of its implementation. To do so, we analyse many recent published works of 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 about the general ideas of TCP over ATM – the degradation of performance and the ATM services for TCP traffic. In section 3, we analyse several published works and conclude on the recent critical issues of GFR implementation and specification. The simulation model, tools, and performance metrics for the experiments have been presented in section 4. Section 5 discusses the performance results among algorithms over LAN and WAN distances. In section 6, the conclusions of this work are given.

2. TCP over ATM

2.1 The Performance Degradation

Having the features of high-speed network, ATM would support the TCP packet with very low transmission error. It would also benefit from a proven transport layer protocol like TCP for data transfer application and would readily acquire the wealth of existing TCP based applications [KR97]. Unfortunately, several studies ([CL94], [KV95], [LI96] and [RF95]) have revealed the degradation of performance when integrating ATM technology with TCP. Some of the reasons for this degradation can be defined as the followings:

1. As shown in [CL94], [MG94], [RF95] and [GJF98a], the dynamics of TCP are deteriorated when implemented over ATM. This mainly results from the bursty nature of TCP and the inherent multiplexing nature of ATM.
   - The bursty nature of TCP causes the round trip time (RTT) (estimated for the TCP connection) to become highly variable. Hence, the RTT can either be underestimated or overestimated. An underestimation of the RTT results in a false timeout indicating congestion, although the TCP packet is actually queued behind other high priority traffic. An overestimation of the RTT also results in the idle period waiting for a timeout when a packet is dropped due to congestion. Both obviously deteriorate the performance [GJF98a].
   - The multiplexing nature of ATM causes cells from different sources appearing together in the switch. Since cell loss usually occurs in bursts, many TCP sources will reduce CWND (Congestion Window) at approximately the same time. This causes the link to become idle and in turn reduces throughput [RF95].
2. The fragmentation of TCP packets into small ATM cells (when a TCP packet flows into an ATM Virtual circuit (VC)) causes poor performance. The packet fragmentation at AAL5 (ATM Adaptation Layer 5) is compulsory since the typical size of a TCP packet is much larger than the size of a cell (48 bytes payload and 5 bytes header). Basically, TCP transmits packets within a window as fast as possible. When the traffic passes through an ATM switch with available buffer size smaller than the sum of TCP windows of all active TCP connections over the network, cells will be dropped at the switch. Due to the retransmission scheme in the ATM layer, the loss of a single cell will result in a corrupted packet at the sink. Hence, the TCP source has to retransmit the entire packet. This radically affects the degradation in performance and possibly contributes to worsen congestion.
3. From (1) and (2), we can also define the problem from the poor co-operation between ATM layer and TCP mechanism. Consequently, the coherent solutions are also important to make better collaboration between the two layers as they have been focused on by many studies ([RK98], [DK99], [DK00a], [DK00b], and [DKB00]).

2.2 ATM Services for TCP Traffic

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 the complicated rules for source behaviour and uses a special Resource Management (RM) cell to indicate the current state of congestion. To use ABR service is like standing by for the excess capacity of the network [Tan96]. If bandwidth is left over, the cells will be transported without delay. If there is insufficient capacity, they will have to wait. If there is congestion after being transported, this will be indicated by the rate-based scheme to slow down.

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 let 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. This can result in low throughput and high unfairness among TCP connections running over UBR with the limited buffer size of the ATM switch [GJF97a]. There are a few ideas to improve TCP over ATM-UBR proposed as the followings:

1. Increase the buffer size of ATM switch:
   In order to achieve maximum throughput at TCP level, ATM switch must have buffer capacity equal to the sum of the TCP receiving windows for all active TCP connections over the network [Sta98]. This will ensure zero cell loss in ATM layer. However, Goyal has shown in [GJF97a] that the ATM switch will need a huge buffer capacity to fulfil this requirement. Moreover, the exact sum of TCP receiving windows is dynamic. This is therefore impractical for the real life. Nonetheless, to increase ATM switch buffer size does help improving the performance of TCP over ATM-UBR. Consequently, a number of vendors have increased their ATM switch buffer size to be rather large [Sta98].
2. Improve TCP congestion control scheme:
   Fast retransmit and recovery and SACK TCP (TCP Selective Acknowledgement) have been taken into account by many studies. Fast retransmit and recovery seems to improve performance only for low latency configuration but degrade performance in long latency configuration. [MMF96] and [GJK97] have shown that SACK TCP may improve performance for large latency networks.
3. Improve buffer management policy:
   In [RF95], there are two buffer management schemes proposed for improving UBR – PPD (Partial Packet Discard) and EPD (Early Packet Discard). Both are based on the notion that when a switch is congested, it is better to drop all cells of one packet than to randomly drop cells from different packets. Presently, EPD is omnipresent in ATM switches from several vendors. Despite the improvement of throughput, EPD has shown serious unfairness among TCP connections. Hence, SD (Selective Drop), FBA (Fair Buffer Allocation) and WFB (Weighted Fair Buffer Allocation) have later been proposed to enhance the fairness among TCP connections ([FL95], [GJF97a], and [GJF97b]).
4. Provide a minimum rate guarantee:
   A minimum bandwidth guarantee may help improving TCP performance over ATM-UBR since a constant amount of bandwidth (provided by a guarantee rate) can ensure that TCP keep receiving ACK (Acknowledgement) from the destination. This reduces the variation of the RTT [GJF98a]. Hence, GFR has recently been proposed to the ATM Forum as another category to support non-real-time applications. It is designed for applications that may require a minimum rate guarantee and can benefit from accessing additional bandwidth dynamically available in the network [ATM99].

3. GFR (Guarantee Frame Rate) : Current Issues

3.1 Motivation of GFR

Although both ABR and UBR can support TCP traffic, there are several reasons why most of TCP traffic over ATM still relies on the UBR category only. Firstly, many existing users are not equipped to comply with the complicated rules required by the ABR source behaviour (for the rate based feedback control). A lot of existing installed ATM products do not support ABR and probably will never support it without a substantial upgrade [Bon97]. Secondly, ABR can significantly improve TCP throughput over UBR only when the buffer capacity of ATM network is small. If there is sufficient buffer capacity in the ATM network, the performance of TCP over ATM-ABR will be even worse than TCP over ATM-UBR. The extra delay incurred in the ABR end-systems and the overhead of RM cells result in the degradation of performance as shown in [MB97]. Due to the large size of buffer currently provided by many vendors, UBR would be the preferred choice.

However, many studies ([RF95], [GJF97a], and [GJK97]) have shown that UBR can provide very poor end-to-end performance as mentioned in section 2.2. In the situation where a VBR (Variable Bit Rate) service is running on the same link, the problem becomes more serious. Since 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) [GH96]. 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 term 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 comparing with ABR service. Its main goal is to bring the benefits of ATM performance and service guarantee to users that today are not able to benefit from such as the current internetworking applications, which are non-quality-service-based network, and non-ABR users [Bon97].

3.2 Specification of GFR

The specification of GFR has recently been proposed by the ATM Forum [ATM99]. Most of issues about GFR have not been well defined and still arguable. However, it is clear that GFR is a frame-based service with guaranteed rate. The complete frames are accepted or discarded in the switch. It uses AAL5 to make frame boundary visible at the ATM layer. This can be done by checking EOM (End Of Message) cell – the last cell of AAL5 frame. By using AAL5, this would imply that GFR is designed specifically for TCP packets. Currently, the GFR service only applies to Virtual Channel (VC) connections because the frame boundary can not be detected in Virtual Path (VP) connections.

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) 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 [GJF98c].

The conformance definition consists of three conditions:

(1) CLP bit conforming: The CLP bit of every cell in the frame must have the same value as the CLP bit of the first cell of the frame.
(2) Maximum Frame Size (MFS) conforming: The frame size is defined as the length in cells of the AAL5-PDU (Protocol Data Unit of ATM Adaptation Layer Type 5) accepted by both ends of the VCs. It has to be no bigger than the specified MFS.
(3) PCR and $CDVT_{PCR}$ conforming: Generic Cell Rate Algorithm – GCRA (1/PCR, $CDVT_{PCR}$) (defined in [ATM99] and [ITU93]) is used to ensure that sources have not sent at rate over PCR.

After passing the conformance definition, the frame will be tested for service eligibility. Simply put, it is a test to ensure that the user does not send over MCR. If a frame has a rate over MCR, the algorithm will tag CLP bits of all cells in the frame or discard the entire frame. This can be done by using the service eligibility-testing algorithm. However, the algorithm is still volatile. When Guerin and Heinanen firstly proposed the UBR+ (or GFR) service to ATM-Forum and ITU-T [GH96], they used the CF-GCRA (Conforming Frame based GCRA) algorithm for service eligibility testing. This algorithm was actually modified from continuous state leaky bucket version of GCRA. Later, the CF-GCRA was modified to F-GCRA [ATM99]. Many ideas have been recently proposed to adjust F-GCRA parameters and its definition such as [Wen97a], [Wen97b], [WKW97], and [Hei99]. One of the most recent algorithms is VRF-GCRA (Variable Rate F-GCRA) [ATM99].

If the frame does not conform to or is not eligible for the traffic contract of GFR, it would be tagged or discarded. Frame Discarding and Frame Tagging in GFR have some different constraints. Frame Discarding is recommended for the discarding scheme, but Frame Tagging is compulsory for the tagging scheme. From the definition in [ATM99], the network must tag all cells in that frame if the tagging scheme is used. However, the network may (should if possible) discard all cells in that frame if it uses the discarding scheme. In other words, the network has a choice to discard just the tail of the frame but will not be allowed to tag just the tail of the frame. Accordingly, Frame Tagging needs to start tagging cells from the first cell of the frame, then continuously tag all succeeding cells of the frame until the End of Message (EOM). If the first cell of the frame violates conformance or eligibility testing, tagging would be a possible choice. Yet, if the first cell does not violate the test and the succeeding cells of the frame later violate the test, the network can not tag all cells in the frame. Tagging therefore would not be done. This can happen in case of checking frame size in conformance testing. The network has not known the Maximum Frame Size violation when the first few cells just arrived. In such a case, the tail of that frame can be just discarded. Kenny has proposed even more flexible ideas that there may be some good reasons for passing cells of a frame that appear to be too long and that should not be prohibited without good reason [Ken97a]. Following this flexible idea, we may simply pass the non-conforming or ineligible cells if there are legitimate reasons. For example, the EOM cell (the last cell of AAL5-PDU) would not be dropped since it carries the information of frame boundary.

3.3 Analysis of GFR Implementation Alternatives

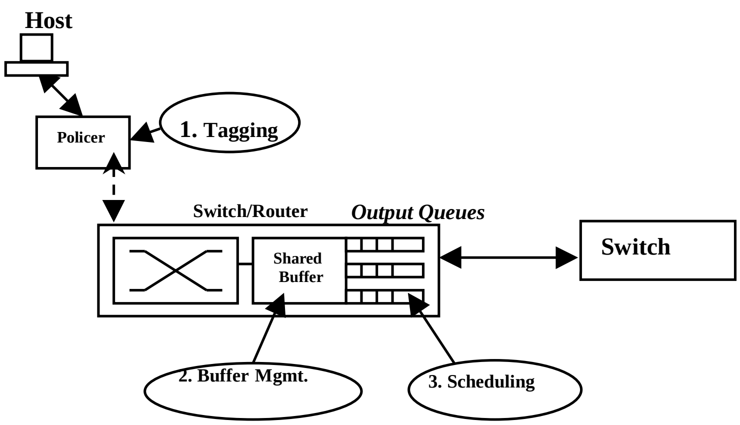

Figure 1: Main components of GFR implementation

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 the 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:

- Non-rate-guarantee service discipline such as FIFO (First In First Out) queue, Round robin queue.
- 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 from various VC. 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, this additional cost is still arguable whether it is desirable.

The possibility to provide GFR service from combining these three components has been studied by many researchers. The main questions are:

- Would the GFR implementation need per-VC Tagging?
- Is per-VC buffer management necessary for the frame-based Buffer Management?
- Does the scheduling need to be rate-guarantee service discipline (such as WFQ) or just FIFO queuing is enough?
- What are the possible combinations to provide GFR service?

We conclude the following:

#	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 [GH96] 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. Both Combinations 3 and 5 use per-VC buffer management 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 proposes the new per-VC buffer management algorithm called Differential Fair Buffer Allocation (DFBA) ([GJF98b] [GJF98c]) and shows that DFBA may be sufficient to provide rate guarantee under certain circumstances which are:

- SACK TCP
- no other non-TCP or higher priority traffic
- Sum of Minimum Cell Rate guaranteed for all VCs is much smaller than link capacity.

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.

3.4 Recent Proposed Implementations of GFR

In [ATM99], there are two implementation samples for GFR: using WFQ with per-VC buffer management and FIFO Queue with tagging. Both are proposed by Guerin and Heinanen [GH96][GH97]. Many studies have confirmed that the first one (WFQ together with per-VC buffer management) can provide GFR services. Yet, several studies have rejected the second one (FIFO Queue with tagging). In addition, Goyal has proposed his GFR implementation using DFBA together with FIFO Queue to the ATM Forum later [GJF98b]. This option is an interesting choice since it is far cheaper than using per-VC queuing like WFQ. Yet, there is no absolute conclusion whether it can provide GFR service. DFBA is a per-VC buffer management scheme with Multiple accounting, Multiple threshold (MAMT). The thresholds of DFBA are dynamic thresholds. The drop type of DFBA is probabilistic. Furthermore, it is a tag sensitive buffer management scheme (with two thresholds – Low Buffer Occupancy (LBO) and High Buffer Occupancy (HBO) for CLP1 and CLP0 stream respectively). We summarise the recent proposed implementations of GFR service in the following table:

Implementation	Buffer Thresholds	Comments
1. WFQ with per-VC buffer management	2 static buffer thresholds + per-VC threshold	Able to provide GFR service, but costly due to using WFQ.
2. FIFO queue with per-VC Tagging	2 static buffer thresholds	Unable to provide GFR and be rejected by many studies.
3. per-VC buffer management with DFBA	2 buffer thresholds + dynamic per-VC threshold + probabilistic drop policy	Just proposed to the ATM Forum and still under study.

Table 2: The proposed GFR implementation

4. Experimental Setup

4.1 Simulation Model and Objectives

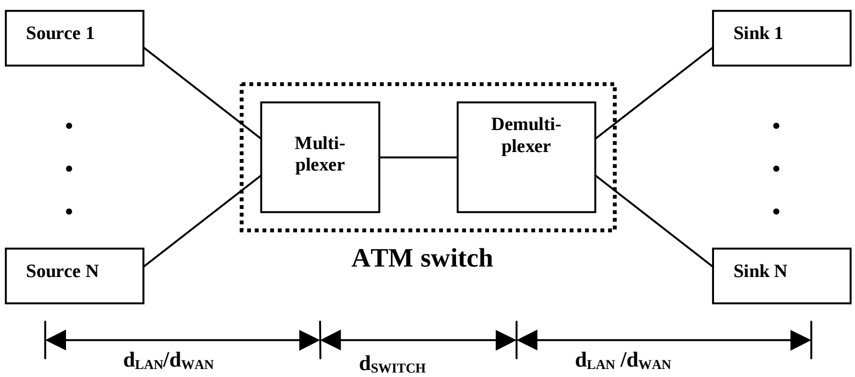

Figure 2: Simulation Model

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 followings:

- 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 the 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). N TCP sources and N sinks are connected together by an ATM switch. 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, which half of VCs will simply send cells without tagging but another half will tag CLP bit.

All TCP sources start sending at the same time. The TCP sources are greedy sources with 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 [JBB92] and Nagle's algorithm [Bra89] 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 1 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 the followings:

#	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 3: 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. The number of sources is in the range [10...22] while the buffer sizes are in the range [2400…4800] cells. The simulation is run long enough to capture congestion behaviour.

4.2 Simulation Tool

The YATS simulation package is used for this experiment [Bau97]. It provides all the important TCP mechanisms such as Slow-Start, Congestion Detection, Congestion Avoidance, Fast Retransmit and Fast Recovery, Nagle algorithm, Delay acknowledgements, Time stamp option. In addition, it also supports all the significant ATM features such as UBR, EPD, WFQ, and AAL5.

However, the simulator has no feature of GFR. Several objects therefore need to be developed and modified to integrate some 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. The algorithms are parts of implementations of GFR proposed by Guerin and Heinanen [ATM99], [GH96], [GH97]. 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 ration (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 ATM layer. A low CLR indicates the efficiency of congestion control in ATM layer. Larger buffer sizes are expected to provide better CLR performance.
- 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. PRR generally corresponds to CLR because the loss of a single cell in a packet will cause the retransmission of the whole packet. However, there may be cases, for example with bursty traffic, where the CLR pattern may not follow PRR.
- 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). Especially, 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 two-threshold scheme is necessary to improve performance of UBR in the situation of CLP0+1 stream (i.e., tagging is on). By comparing two graphs in figure 3, we can also 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 is different. As shown in the figure, the CLR of CLP0 stream is almost 0% while the CLR of CLP1 stream varies from 0.001% to 0.0012%.

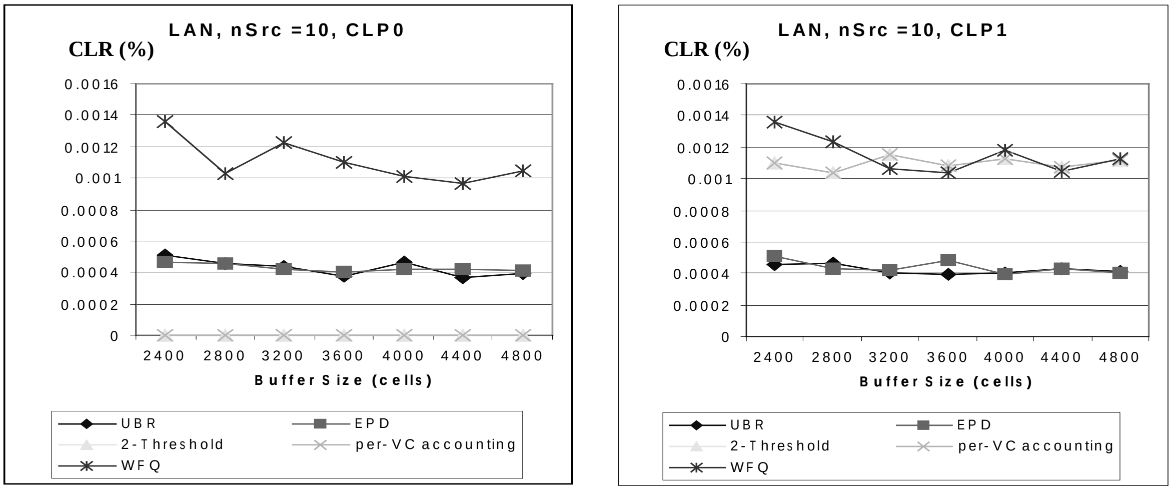

Figure 3: Comparing CLR of CLP0 and CLP1 stream

However, plain UBR, UBR-EPD and WFQ have similar CLRs for both CLP0 and CLP1 streams (around 0.0005%). 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.

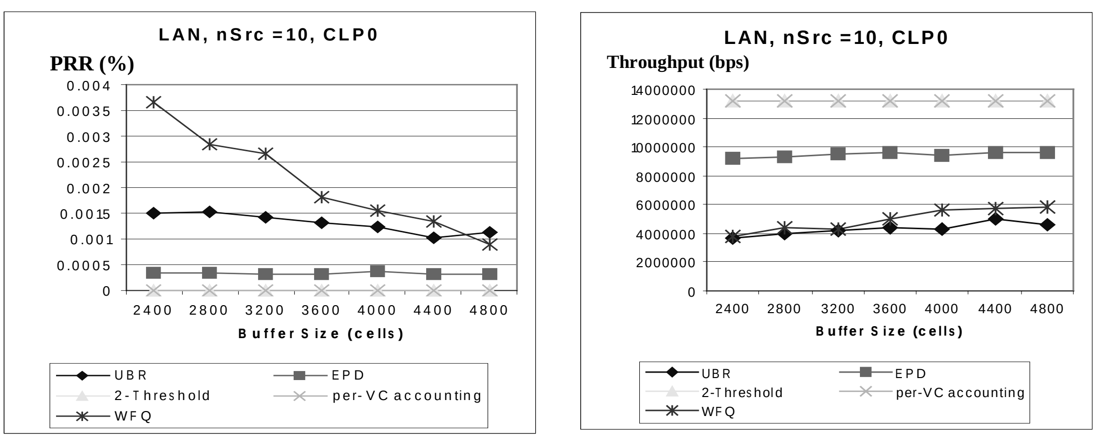

Figure 4: Comparing PRR and Throughput of CLP0 stream among algorithms

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 as show in figure 4.

In figure 4, throughput has been improved with the larger buffer sizes due to the enhanced ability of the ATM switch to tackle congestion. 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 3 and 4, 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. As 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 can not distinguish between tagged and untagged traffic in CLP0+1 stream. Hence, it treats CLP0 and CLP1 packets with the same priority and can give the very poor performance when CLP1 packets flush the bandwidth. We tend to conclude that per-VC queuing algorithm without sensitivity to tagging can experience low performance for CLP0 (high priority) stream since it can be disturbed by CLP1 (low priority) stream.

5.2 Varying the Number of Sources

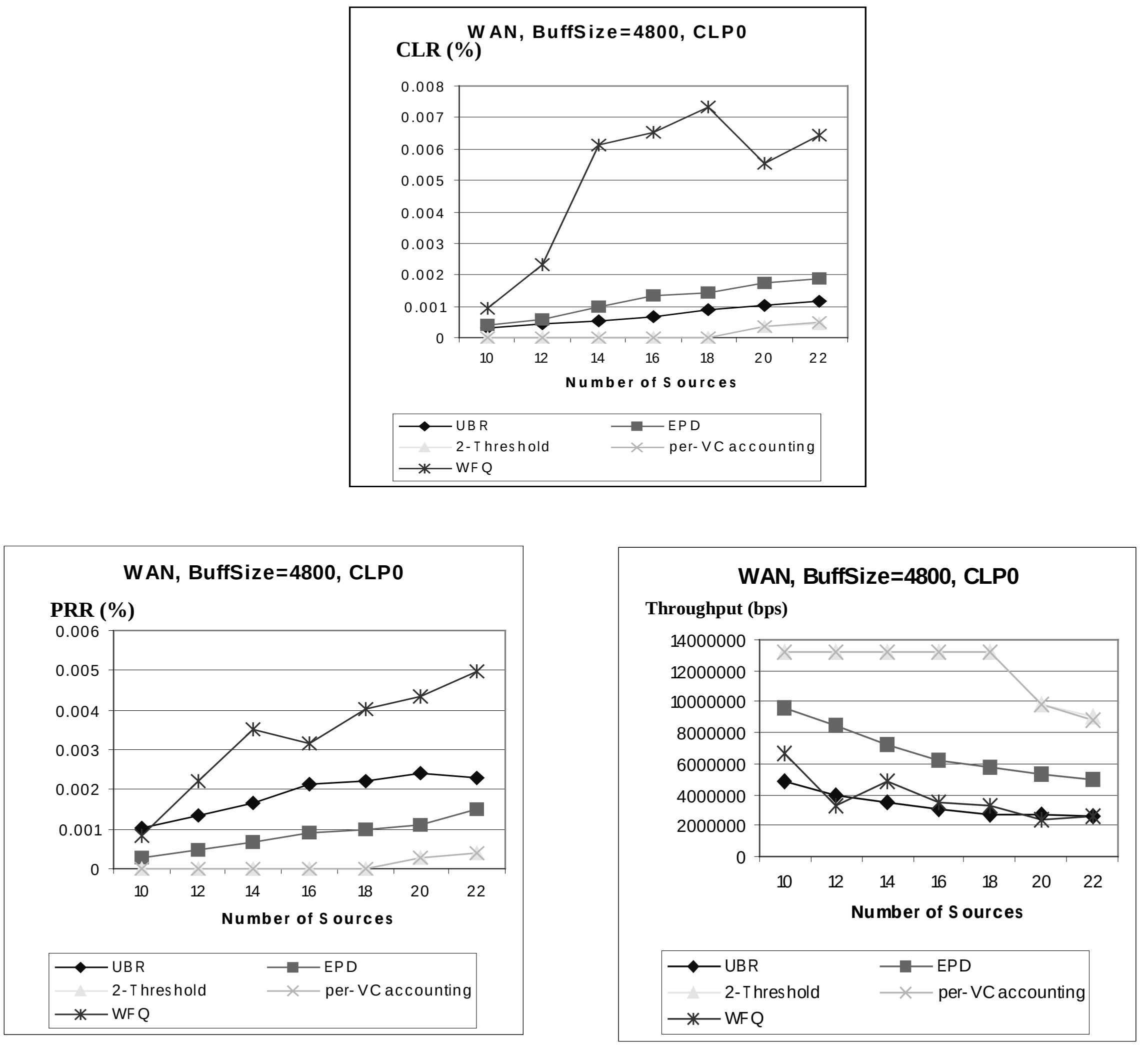

Figure 5: Comparing PRR and Throughput of CLP0 stream among different algorithms

As expected, the throughput decreases when the number of sources increases for all scenarios. This is obviously because more sources cause more congestion. Figure 5 shows the comparison of CLR, PRR and throughput among five switches (plain UBR, UBR-EPD, WFQ, per-VC buffer management and two-threshold buffer management).

Similar with the results in section 5.1, two-threshold-buffer-management and per-VC-buffer-management 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.

6. Conclusion

GFR specification and its recent critical issues have been drawn out based on published works. 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 GFR service. Using per-VC scheduling together with per-VC Tagging and/or per-VC buffer management has been ensured by many studies that can provide minimum guaranteed rate. However, the cost of per-VC scheduling is undesirable. So far, there are still some conflicting results (depending on the scenarios and algorithms used) whether or not GFR can rely on only an intelligent buffer management without per-VC scheduling.

The simulation study has confirmed that the two-threshold scheme is vital to handle CLP0+1 traffic to protect the bandwidth from flushing by CLP1 (low priority) stream. It can help improve 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). The per-VC accounting does not help improve throughput, CLR and PRR although it may help improve the fairness. The per-VC scheduling without the sensitivity of tagged/untagged frames 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. ftp://ftp.atmforum.com/pub/approved-specs/af-tm-0121-000.pdf

[Bau97] M. Baumann. YATS User's and Programmer's Manual for Version 0.3. Dresden University of Technology, September 1997. http://www.ifn.et.tu-dresden.de/TK/yats/yats.html

[Bon97] O. Bonaventure. A simulation study of TCP with the proposed GFR service category. DAGSTUHL Seminar 9725 Dagstuhl, Germany. June 1997. http://www-run.montefiore.ulg.ac.be/publications/papers/abstract-PP98-01.html

[BK97] A. Bestavros and G. Kim. TCP Boston - A Fragmentation-tolerant TCP Protocol for ATM Networks. Proceedings of IEEE INFOCOM'97. 1997. http://www.cs.bu.edu/techreports/1996-014-tcp-boston.ps.Z

[Bra89] R. Braden. Requirements for Internet Hosts: Communication Layers. RFC-1122, October 1989. http://sunsite.auc.dk/RFC/rfc/rfc1122.htm

[DK99] K. Djemame and M. Kara. Proposals for a Coherent Approach to Co-operation between TCP and ATM Congestion Control Algorithms. In Proceeding of the 15th Annual UK Performance Engineering Workshop (UKPEW'99), pp. 273-284, Bristol, UK, July 1999. http://www.scs.leeds.ac.uk/atm-mm/ papers/1999/ukpew99.ps.gz

[DK00a] K. Djemame and M. Kara. TCP Explicit Congestion Notification over ABR and UBR: an Agent-Based Approach. To appear in the Proceedings of the 3rd IEEE International Conference on ATM, Heidelberg, Germany, June 2000. http://www.scs.leeds.ac.uk/atm-mm/papers/2000/icatm2000.ps.gz

[DK00b] K. Djemame and M. Kara. A New Rate Co-ordination Protocol for TCP and ABR Congestion Control. To appear in the Proceedings of the IEEE International Conference on Networks (ICON'2000), Singapore, September 2000. http://www.scs.leeds.ac.uk/atm-mm/papers/2000/rate_coord.ps.gz

[DKB00] K. Djemame, M. Kara and R. Banwait. An Agent Based Congestion Control and Notification Scheme for TCP over ABR. To appear in Journal of Computer Communications. http://www.scs.leeds.ac.uk/atm-mm/tcpecnabr.ps.gz

[CL94] D. E. Comer and J.C. Lin. TCP Buffering and Performance over an ATM Network. Technical Report-CSD-TR-94-026, Computer Sciences Department, Perdue University, West Lafayette, IN 47907, USA. 1994.

[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.

[GH96] R. Guerin and J. Heinanen. UBR+ Service Category Definition. ATM Forum contribution ATM96-1598, December 1996.

[GH97] R. Guerin and J. Heinanen. UBR+ Enhancements. ATM Forum contribution ATM97-0015, February 1997.

[GJF97a] R. Goyal, R. Jain and S. Fahmy. UBR+: Improving Performance of TCP over ATM-UBR service. ICC'97 Montreal. June 1997. http://www.netlab.ohio-state.edu/~jain/papers/icc97.htm

[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. http://www.netlab.ohio-state.edu/~jain/atmf/a97-0831.htm

[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. http://www.cis.ohio-state.edu/~jain/papers/atm98.htm

[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. http://www.netlab.ohio-state.edu/~jain/atmf/a98-0405.htm

[GJF98c] R. Goyal, R. Jain, S. Fahmy and B. Vandalore. GFR Implementation Options. ATM Forum contribution ATM98-0406, July 1998. http://www.netlab.ohio-state.edu/~jain/atmf/a98-0406.htm

[GJK97] R. Goyal, R. Jain, S. Kalyanaraman and et al. Selective Acknowledgements and UBR+ Drop Policies to Improve TCP/UBR Performance over Terrestrial and Satellite Networks. ATM Forum Contribution ATM97-0423, April 1997. http://www.cis.ohio-state.edu/~jain/atmf/a97-0423.htm

[Hei99] H. Heiss. QoS results for the GFR ATM service. ITC16 Edinburgh UK, Page 227-236, June 1999.

[ITU93] Telecommunication Standardisation Sector of ITU. Traffic Control and Congestion Control in B-ISDN. ITU-T I.371, March 1993.

[JBB92] V. Jacobson, R. Braden and D. Borman. TCP Extensions for High Performance. RFC-1323, May 1992. http://sunsite.auc.dk/RFC/rfc/rfc1323.htm

[Ken97a] J. B. Kenney. Open Issues on GFR Frame Discard and Frame Tagging. ATM Forum contribution ATM97-0385, April 1997.

[Ken97b] J. B. Kenney. Issues on GFR Conformance/QoS Eligibility. ATM Forum contribution ATM97-1059, December 1997.

[KR97] M. Kara and M. Rahin. Towards a Framework for Performance Evaluation of TCP Behaviour over ATM Networks. ICCC 1997. http://www.scs.leeds.ac.uk/atm-mm/papers/1997/TCP-ATMPerfFrameWork.ps.gz

[KV95] L. Kalampoukas and A. Varma. Towards Performance of TCP over Multi-hop ATM Networks: A Comparative Study of ATM Layer Congestion Control Schemes. Technical Report UCSC-CRL-95-13, Computer Engineering & Information Sciences, University of California, Santa Cruz, CA 95064 USA, 1995.

[LH98] M. Laubach and J. Halpern. Classical IP and ARP over ATM. RFC-2225, April 1998. http://sunsite.auc.dk/RFC/rfc/rfc2225.htm

[LI96] H Li et al. TCP over ABR and UBR Services in ATM. Proceedings of IPCC, 1996.

[MB97] S. Manthorpe and J. L. Boudec, A comparison of ABR and UBR to support TCP traffic, Unpublished manuscript 1997. http://lrcwww.epfl.ch/PS_files/TR97_221.ps.gz

[MG94] K. Moldeklev, and P. Gunningberg, Deadlock situation in TCP over ATM, The 4th IFIP workshop for high speed networks, pp. 219-235, Vancouver, Canada, August 1994.

[MMF96] M. Mathis, J. Mahdavi, S. Floyd and A. Ramanow, TCP Selective Acknowledgement Options. RFC-2018, April 1996. http://sunsite.auc.dk/RFC/rfc/rfc2018.htm

[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. ftp://ftp.ee.lbl.gov/papers/tcp_atm.ps.Z

[RK98] M. A. Rahin and M. Kara. Indicators for the assessment of congestion in TCP over ATM-UBR. BC'98 conference. 1998. http://www.scs.leeds.ac.uk/atm-mm/papers/1998/BC98Paper.ps.gz

[Sta98] W. Stallings. High-Speed Networks: TCP /IP and ATM Design Principles. Prentice Hall, 1998.

[Tan96] Andrew Tanenbaum. Computer Networks. Prentice Hall, 3rd edition, 1996.

[Wen97a] R. Wentworth. Issues Related to the GFR Service Definition. ATM Forum contribution ATM97-0922, December 1997.

[Wen97b] R. Wentworth. Supplemental Text for the GFR Service Definition. ATM Forum contribution ATM97-0980, December 1997.

[WKW97] R. Wentworth, D. Kataria, T. Worster. Updated Text for the GFR Service Definition. ATM Forum contribution ATM97-0954R1, December 1997.