****

ATM Forum Document Number: ATM_Forum/97-0607

****

Title: Simulation Experiments with Guaranteed Frame Rate for TCP/IP Traffic.

****

Abstract: We study the effects of tagging, per-VC buffer allocation, and per-VC scheduling on providing GFR to TCP/IP traffic. We conclude that per-VC scheduling is necessary to provide GFR guarantees to TCP traffic. Per-VC scheduling alone is sufficient to provide minimum rate guarantees to the CLP0+1 stream. If the source end systems perform tagging (but not the network), then per-VC scheduling and per-VC buffer allocation are both required to provide GFR guarantees to the CLP0 stream.

****

Source:
Rohit Goyal, Raj Jain, Sonia Fahmy, Bobby Vandalore, Shiv Kalyanaraman
Department of CIS, The Ohio State University (and NASA)

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

Phone: 614-292-3989, Fax: 614-292-2911, Email: {goyal,jain}@cis.ohio-state.edu

Sastri Kota
Lockheed Martin Telecommunications/Astrolink
1272 Borregas Avenue,
Bldg B/551 O/GB - 70
Sunnyvale, CA 94089
Email: sastri.kota@lmco.com

Pradeep Samudra,
Broadband Network Lab
Samsung Electronics Co. Ltd.
Samsung Telecom America, Inc.
1130 E Arapaho, Richardson, TX 75081
email: psamudra@telecom.sna.samsung.com

****

Date: July 1997, Montreal

****

Distribution: ATM Forum Technical Working Group Members (AF-TM)

****

Notice: This contribution has been prepared to assist the ATM Forum. It is offered to the Forum as a basis for discussion and is not a binding proposal on the part of any of the contributing organizations. The statements are subject to change in form and content after further study. Specifically, the contributors reserve the right to add to, amend or modify the statements contained herein.

****

1 Introduction: Guaranteed Frame Rate

Guaranteed Frame Rate (GFR) is intended to provide a minimum rate guarantee to VCs at the frame level. GFR could be used by connections that can neither specify sustainable rate and burst size parameters, nor can be subject to the ABR source rules. These applications could benefit from a minimum rate guarantee by the network, along with an opportunity to fairly use any additional bandwidth left over from higher priority connections. The original GFR proposals [8, 9] describe two possible implementations for GFR:

- Using per-VC queuing (fair scheduling) and per-VC accounting (buffer management).
- Using FIFO queuing and per-VC policing (GCRA based tagging/dropping).

In the April'97 meeting, it was shown by [2] that it is difficult to provide end to end rate guarantees with per-VC policing and FIFO queuing for TCP traffic. It was also clear that per-VC queuing together with per-VC accounting can provide minimum rate guarantees. However, it was unclear at that point, if the addition of per-VC accounting to FIFO queuing with per-VC policing would be enough provide the necessary guarantees. Also, with per-VC queuing, the role of tagging needs to be studied carefully.

We explore three different mechanisms – policing, buffer management, and scheduling – for providing minimum guarantees to TCP traffic. We conclude that per-VC scheduling (fair queuing) is necessary to ensure minimum rate guarantees. We also discuss the dynamics of the interactions of per-VC scheduling with buffer management and policing. In this contribution, we present simulation results for infinite TCP traffic.

2 Design Options for Implementing GFR

There are three basic tools that can be used by the UBR service to provide the per-VC rate guarantees of GFR:

1. **Policing.**
2. **Buffer management.**
3. **Queuing.**

In the following subsections, we discuss the mechanisms for each of the three options and describe our implementations of each. We then present our simulation results with various combinations of policing, buffer management and queuing to provide GFR.

2.1 Policing

GFR conformance is specified by a generic cell rate algorithm (GCRA) based policing mechanism. The conformance definition maps the frame level GFR guarantees to cell level guarantees. The

GCRA parameters are 1/MCR (minimum cell rate) and BT(MBS) + CDVT. Here BT is the burst tolerance corresponding to the maximum burst size (MBS), and CDVT is the cell delay variation tolerance associated with the rate. MBS is defined as MBS = two times the CPCS-SDU (common part convergence sublayer service data unit) size in cells. This is twice the maximum frame size that can be sent by the application using the GFR connection.

Like GCRA, a token count is maintained by the leaky bucket, and a token is consumed by each conforming cell. A non-conforming cell does not consume a token. Cell conformance is determined as follows:

When the first cell of a frame arrives, if there are at least MBS/2 tokens in the bucket, then the cell (as well as the whole frame) is conforming, else the cell (frame) is non-conforming. If the frame is conforming, then every subsequent cell of the frame consumes a token, and if the frame is non-conforming then none of its cells consume a token. Tokens are replenished at the rate of MCR. Non-conforming frames may be tagged or dropped. This policing mechanism allows marking of complete frames.

We implemented the corresponding **continuous state leaky bucket algorithm** described as follows:

Let the first cell of a frame arrive at time t. Let X be the value of the leaky bucket counter, and LCT be the last compliance time of a cell. Let $I = 1/MCR$, and let $BT = (MBS-1)(1/MCR-1/PCR)$.

When the first cell of a frame arrives (at time t):

```
X1 := X - (t - LCT)

IF (X1 < 0) THEN

        X1 := 0
        CELL IS CONFORMING
        TAGGING := OFF

ELSE IF (X1 < BT/2 + CDVT) THEN

        CELL IS NON-CONFORMING
        TAG/DISCARD CELL
        TAGGING := ON

ELSE

        CELL IS CONFORMING
        TAGGING := OFF

ENDIF


IF (CELL IS CONFORMING)
```

```
        X := X1 + I
        LCT := t
ENDIF
```

When subsequent cells of a frame arrive (at time t) then:

```
IF (TAGGING == ON) THEN
        TAG/DISCARD CELL
ELSE
        CELL IS CONFORMING
        X1 := MAX(X - (t - LCT), 0)
        X  := X1 + I
        LCT = t
ENDIF
```

Notes:

- An exception can be made for the last cell of the frame (e.g., EOM cell for AAL5 frames). Since the last cell carries frame boundary information, it is recommended that the last cell should not be dropped unless the entire frame is dropped. For this reason, a network may choose to not tag the last cell of a frame. In our implementation we do not drop or tag the last cell of a frame.

- Policing is typically performed at the entrance to a network. The non-conforming frames can either be tagged or discarded. The source end system can also tag lower priority frames for preferential discard by the network. The stream of untagged frames in a VC is called the CLP0 stream, while the entire stream is called the CLP0+1 stream.

- When a network element sees a tagged frame, it cannot tell if the frame was tagged by the source end system or an intermediate network. This has significant influence in providing rate guarantees to the CLP0 stream, and is further discussed in section 2.2.

2.2 Buffer Management

Various buffer management schemes can be used as mechanisms for congestion avoidance and control. These include preferential dropping of tagged frames over untagged frames when mild congestion is experienced, and the use of per-VC accounting to fairly allocate buffers among the competing connections.

In the April'97 meeting, it was shown [2] that providing rate guarantees with policing and preferential dropping of tagged frames was very difficult for TCP traffic. A combination of large frame size and a low buffer threshold (at which tagged frames are dropped) is needed to provide minimum rate guarantees. This is undesirable because low thresholds result in poor network utilization.

However, it was unclear at that point, if adding per-VC accounting to FIFO queuing with per-VC policing would be enough provide the necessary guarantees. We have experimented with a buffer allocation policy called **Weighted Buffer Allocation (WBA)** based on the policies outlined in

[5] and [6]. This policy assigns weights to the individual VCs based on their MCRs. The weights represent the proportion of the buffer that the connection can use before it is subject to drop. The WBA policy is described below:

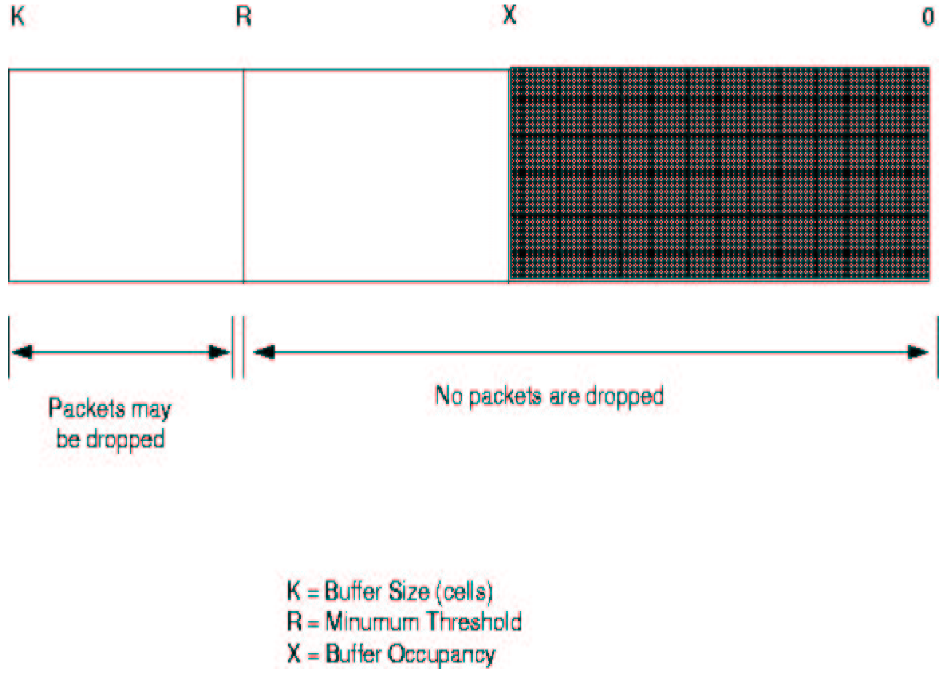

Figure 1: Drop behavior of buffer management policy

Figure 1 illustrates the conditions under which frames may be dropped due to buffer overflow. When a the first cell of a frame arrives on a VC, if the current buffer occupancy (X) is less than a threshold (R), then the cell and the remaining cells of the frame are accepted. If the buffer occupancy exceeds the congestion threshold, then the cell is subject to drop depending on two factors – is the cell tagged? and, is the buffer occupancy of that VC more than its fair share?

Under mild congestion conditions, if there are too few cells of a VC in the buffer, then an attempt is made to accept conforming cells of that VC. As a result, if the VC's buffer occupancy is below its fair share, then a conforming (untagged) cell and the remaining cells of the frame are accepted, but a tagged cell and the remaining cells of the frame (except the last) are dropped. In this way, at least a minimun number of untagged frames are accepted from the VC.

If the buffer occupancy exceeds the mild congestion threshold, and the VC has at least its share of cells in the buffer, then the VC must be allowed to fairly use the remaining unused buffer space. This is accomplished in a similar manner as the schemes in [6], so that the excess buffer space is divided up equally amongst the competing connections.

The switch output port consists of a FIFO buffer for the UBR+ class with the following attributes:

- **K:** Buffer size in cells.
- **R:** Congestion threshold in cells ($0 \leq \text{R} \leq \text{K}$).
- **X:** Number of cells in the buffer.

- **Yi:** Total number of cells of VCi in the buffer.
- **Li:** Total number of untagged cells of VCi in the buffer.
- **Z:** Parameter $(0 \leq Z \leq 1)$
- **Wi:** Weight of VCi (for example Wi = MCRi/(Total UBR capacity)). $\Sigma Wi < 1$
- **Na:** Number of active VCs, i.e, VCs with cells in the buffer.

When the first cell of a frame arrives:

```
IF (X < R) THEN

        ACCEPT CELL AND FRAME

ELSE IF (X > R) THEN

        IF ((Li < R*Wi) AND (Cell NOT Tagged)) THEN

                ACCEPT CELL AND FRAME

        ELSE IF ((Yi - R*Wi)*Na < Z*(X-R)) THEN

                ACCEPT CELL AND FRAME

        ELSE

                DROP CELL AND FRAME (except the EOM cell)

        ENDIF
ENDIF
```

Further drop policies like EPD and PPD can also be used with a more severe congestion threshold on top of the WBA threshold R. In our simulations we do not use EPD/PPD with WBA.

Per-VC buffer management can be effectively used to enforce tagging mechanisms. If the network does not perform tagging of VCs, but it uses the tagging information in the cells for traffic management, then per-VC accounting of untagged cells becomes necessary for providing minimum guarantees to the untagged (CLP0) stream. This is because, if a few VCs send excess traffic that is untagged, and no per-VC accounting is performed, then these VCs will have an excess number of untagged cells than their fair share in the buffer. This will result in conforming VCs (that tag all their excess traffic) receiving less througput than their fair share

The conforming VCs will experience reduced throughput because the non-conforming but untagged cells of the overloading VCs are not being isolated. This could be the case even if per-VC queuing in a *shared buffer* is performed. If all VCs share the buffer space, then without per-VC buffer management, the overloading VCs will occupy an unfair amount of the buffer space, and may cause some conforming frames of other VCs to be dropped. Thus, no matter how frames are scheduled (per-VC or not), if the switch allows an an unfair distribution of buffer occupancy, then the resulting output will also be unfair.

If a network does not tag the GFR flows, its switches should not use the information provided by the CLP bit unless they implement per-VC buffer management. We will show in section 4 that with per-VC queuing and EPD (without per-VC buffer management), a switch can provide rate guarantees for the CLP0+1 flow, but not for the CLP0 flow.

2.3 Fair Queuing (per-VC scheduling)

Fair queuing can control the outgoing frame rate of individual VCs. It was shown by [2] that with network tagging, fair queuing with preferential dropping of tagged frames (without per-VC management) can provide end to end rate guarantees for infinite TCP traffic. Our simulation results verify the intuition that that minimum rate guarantees for the CLP0+1 flow can be provided simply by fair queuing and EPD like buffer management. There is no need to look at the CLP bit to provide rate guarantees to the CLP0+1 flow. If the source end system does not tag its excess traffic, then it is arguable if the CLP0 flow has any special meaning for the end system. In this case, it might be just as appropriate in terms of end to end throughput to provide a GFR guarantee to the CLP0+1 flow.

To provide rate guarantees to the CLP0 flow, per-VC accounting of the CLP0 flow must be performed along with per-VC scheduling. This protects the network from overloading sources that do not tag their excess traffic.

We implement a Weighted Fair Queuing like service discipline to provide the appropriate scheduling. The details of the discipline will be discussed in a later contribution. The results of our simulation are discussed in section 4.

3 Simulation of SACK TCP over GFR

This section presents the simulation configuration of the various enhancements of TCP and UBR presented in the previous sections.

3.1 The Simulation Model

All simulations use the N source configuration shown in figure 2. All sources are identical and infinite TCP sources. The TCP layer always sends a segment as long as it is permitted by the TCP window. Moreover, traffic is unidirectional so that only the sources send data. The destinations only send ACKs. The delayed acknowledgement timer is deactivated, and the receiver sends an ACK as soon as it receives a segment. The version of TCP used is SACK-TCP. SACK TCP uses selective acknowledgements to selectively retransmit lost segments. SACK-TCP has been shown to improve the performance of TCP especially for large delay networks. Details about our implementation of SACK TCP can be found in [7].

Most of our simulations use 15 sources. We also perform some simulations with 50 sources to understand the behavior with a large number of sources.

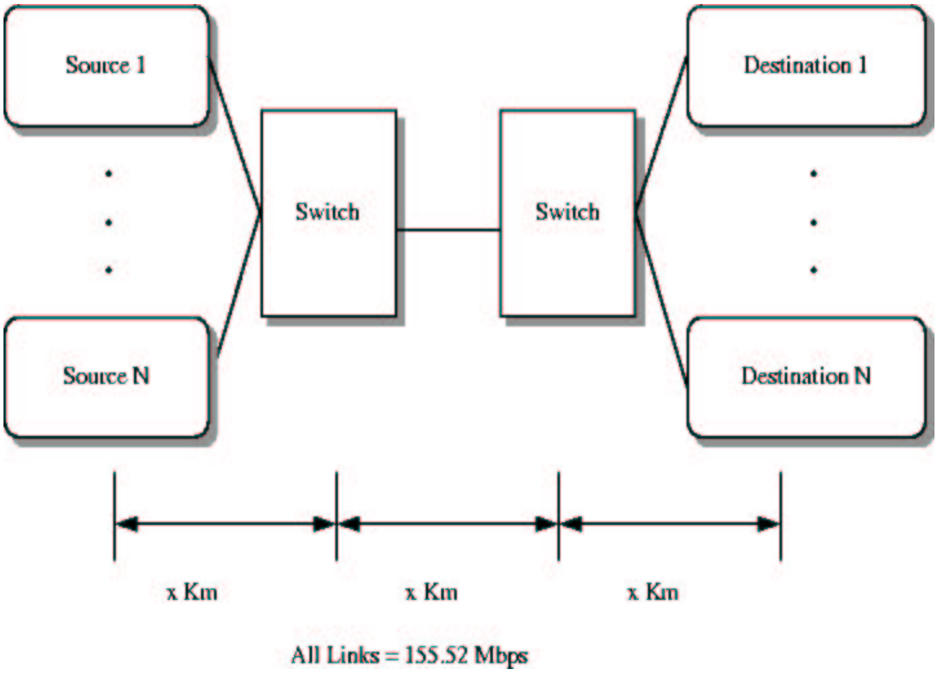

Figure 2: The N source TCP configuration

Link delays are 5 milleseconds. This results in a round trip propagation delay of 30 milliseconds. The TCP segment size is set to 1024 bytes. For this configuration, the TCP default window of 64K bytes is not sufficient to achieve 100% utilization. We thus use the window scaling option to specify a maximum window size of 600,000 Bytes.

All link bandwidths are 155.52 Mbps, and peak cell rate at the ATM layer is 149.7 Mbps after the SONET overhead. The duration of the simulation is 20 seconds. This allows for adequate round trips for the simulation to give stable results.

In our simulations, the end systems do not perform any tagging of the outgoing traffic. We measure performance by the end to end effective throughput obtained by the destination TCP layer. The goal is to obtain an end-to-end througput as close to the allocations as possible. Thus, in this contribution we look at the aggregrate CLP0+1 flow. Further work needs to be done to measure performance of the CLP0 flow.

4 Simulation Results

4.1 N TCP sources with equal rate allocation

We simulated N TCP sources on a 155 Mbps link. Each source was allocated 1/Nth of the link capacity. We used only a per-VC accounting based buffer management policy called Selective Drop [6]. Figure 3 shows the resulting end to end TCP performance with switch buffer sizes 12000 cells and 24000 cells for 15 and 50 sources.

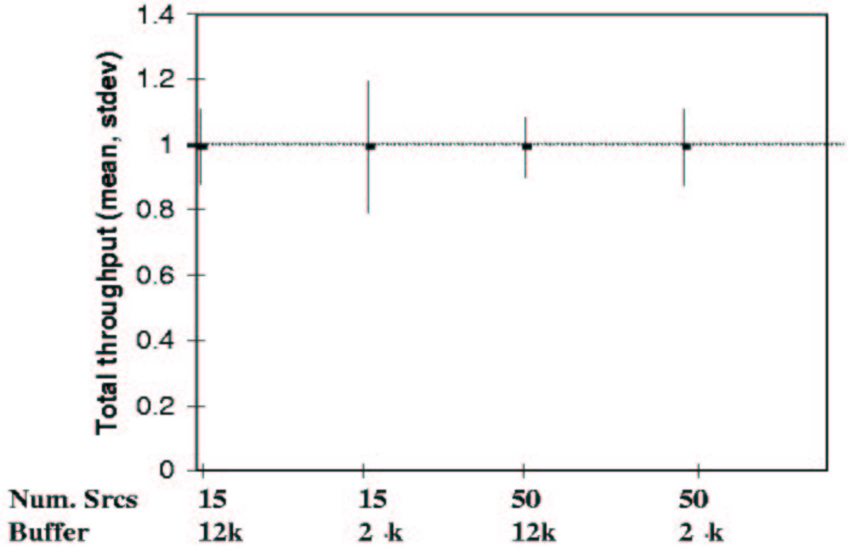

Figure 3: Per-VC Accounting: Equal Rate Allocations

Figure 3 plots the average efficiencies of the 4 simulation experiments. For the 15 (50) source configurations, each source has a maximum expected throughput of about 8.2 (2.5) Mbps. The actual achievied throughput is divided by 8.2 (2.5) , and then the mean for all 15 (50) sources is found. The point plotted represents this mean. The vertical lines show the standard deviation below and above the mean. Thus, shorter standard deviation lines mean that the actual throughputs were close to the mean. Standard deviation is an indicator of the fairness of the sources' throughputs. A smaller standard deviation indicates larger fairness.

In figure 3, all TCP throughputs are within 20% of the mean. The mean TCP throughputs in the 15 source configuration are about 8.2 Mpbs for both buffer sizes. For 50 sources, the mean TCP throughputs are about 2.5 Mbps. The standard deviation from the mean increases with increasing buffer sizes for the 15 source configuration. The increase in buffer sizes leads to more variability in the allocation, but as the number of sources increases, the allocation reduces, and so does the variability. All configurations exhibit almost 100% efficiency, and the fairness for large number of sources is high. *Equal allocation of the available bandwidth for TCP traffic to the CLP0+1 flow can be achieved by per-VC accounting only.*

4.2 N TCP sources with unequal rate allocations

The N TCP connections were divided up into 5 categories of bandwidth allocations. For example, 15 sources are divided into groups of 3, with MCRs of approximately 2.6 Mbps, 5.3 Mpbs, 8 Mpbs, 10.7 Mbps, 13.5 Mpbs.

To achieve unequal allocations of end to end throughputs, we tried two options:

1. Per-VC accounting (Weighted Buffer Allocation) with tagging.

2. Per-VC scheduling without tagging or accounting.

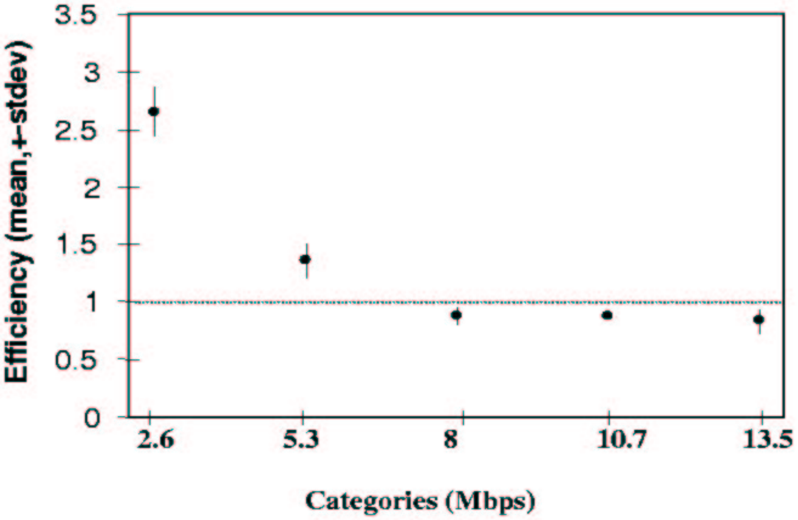

Figure 4: Per-VC Accounting+Tagging: Unequal Rate Allocations

Figure 4 shows the resulting allocations obtained from the simulations for tagging and per-VC accounting. The 5 points represent the efficiencies obtained by the 5 categories of bandwidth allocations. The horizontal dotted line represents the target efficiency (100Thus, the average allocation of the first category was over 2.5 times its target allocation (2.6 Mbps), while the last category received 0.8 of its allocation (13.5). The dotted line represents the target efficiency for each category. The total efficiency (not shown in the figure) was almost 100%. The figure shows that per-VC accounting with tagging cannot effectively isolate competing TCP traffic streams. We tried various drop thresholds R for WBA, and R = 0.5 achieved the best isolation. However, the achieved rates were not close to the target rates. *FIFO buffering with Per-VC buffer allocation together with tagging is not sufficient to provide GFR guarantees.*

Figure 5 shows the resulting allocations obtained for the 15 source unequal rate configuration with per-VC scheduling without tagging or accounting. The figure shows that the achieved throughputs are close (within 10%) to the target throughputs for the CLP0+1 stream. Also the total efficiency (not shown in the figure) is over 90% of the link capacity. This is is the target capacity set for the per-VC scheduler. From this, we conclude that *per-VC queuing is necessary to achieve minimum rate guarantees for infinite TCP traffic.*

To provide rate guarantees to the CLP0 flow, per-VC accounting is needed to separate the CLP0 frames of the different VCs. If tagging is performed by the network on all the VCs, then per-VC accounting may not be necessary, and preferential dropping of tagged frames with per-VC scheduling should be sufficient. Simulation results of this configuration are under current study.

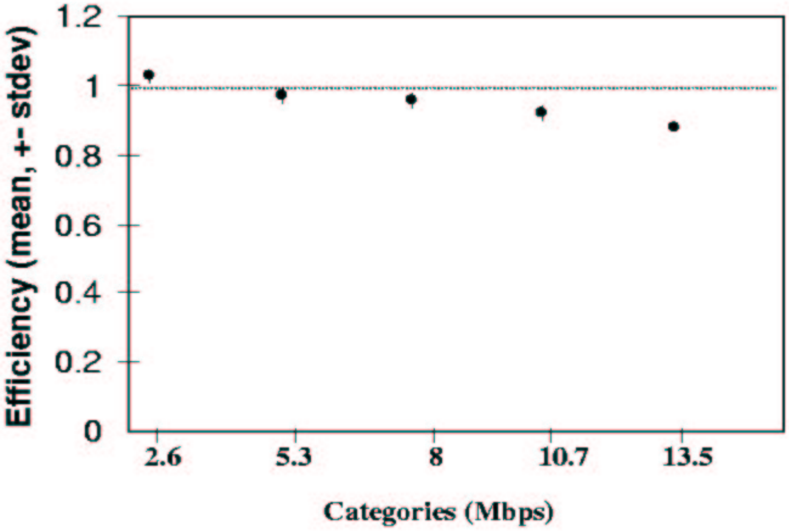

Figure 5: Per-VC Scheduling: Unequal Rate Allocations

5 Summary

To summarize, our goal has been to find ways to provide GFR guarantees to infinite TCP traffic flows. So far, we have considered the case where the TCP end systems do not perform tagging. We measure performance by the end to end effective TCP throughputs of the CLP0+1 stream. The following conclusions can be drawn for infinite TCP/IP traffic.

- FIFO queuing with per-VC accounting based buffer management is not sufficient to provide GFR guarantees.
- GFR guarantees to the CLP0+1 stream can be provided by per-VC queuing (fair scheduling) alone. This is irrespective of any tagging operations.
- To provide guarantees to the CLP0 flow, the network must perform per-VC buffer management in addition to per-VC queuing. This protects the CLP0 traffic of conforming flows from flows whose non-conforming frames are not tagged either by the end system or by the network.

The results from the April'97 meeting [2] and this contribution can be summarized by table 1. The table shows the eight possible options available with per-VC accounting, tagging (by the network) and queuing. An "X" indicates that the option is used, and a "-" indicates that it is not used. The GFR column indicates if GFR guarantees can be provided by the combination specified in the row.

Table 1: Design options for providing GFR guarantees

Per-VC Accounting	Per-VC Tagging	Per-VC queuing	GFR	Notes
-	-	-	No	Clearly
X	-	-	No	
-	X	-	No	Shown by [2]
X	X	-	No	Shown here
-	-	X	Yes	Only for the CLP0+1 stream
X	-	X	Yes	Also for CLP0 stream
-	X	X	Yes	Results under current study
X	X	X	Yes	Also for CLP0 stream

References

[1] ATM Forum, “ATM Traffic Management Specification Version 4.0,” April 1996, ftp://ftp.atmforum.com/pub/approved-specs/af-tm-0056.000.ps

[2] Debashis Basak, Surya Pappu, “TCP over GFR Implementation with Different Service Disciplines: A Simulation Study”

[3] John B. Kenney, “Satisfying UBR+ Requirements via a New VBR Conformance Definition,” ATM FORUM 97-0185.

[4] John B. Kenney, “Open Issues on GFR Frame Discard and Frame Tagging,” ATM FORUM 97-0385.

[5] Juha Heinanen, and Kalevi Kilkki, “A fair buffer allocation scheme,” Unpublished Manuscript.

[6] R. Goyal, R. Jain, S. Kalyanaraman, S. Fahmy and Seong-Cheol Kim, “UBR+: Improving Performance of TCP over ATM-UBR Service,” Proc. ICC’97, June 1997.

[7] R. Goyal, R. Jain et.al., “Selective Acknowledgements and UBR+ Drop Policies to Improve TCP/UBR Performance over Terrestrial and Satellite Networks,” To appear, Proc. IC3N’97, September 1997. [1]

[8] Roch Guerin, and Juha Heinanen, “UBR+ Service Category Definition,” ATM FORUM 96-1598, December 1996.

[9] Roch Guerin, and Juha Heinanen, “UBR+ Enhancements,” ATM FORUM 97-0015, February 1997.

[1] All our papers and ATM Forum contributions are available from http://www.cis.ohio-state.edu/~jain