

\*\*\*\*\*

ATM Forum Document Number: ATM\_Forum/98-0406

\*\*\*\*\*

TITLE: GFR Implementation Options

\*\*\*\*\*

SOURCE: Rohit Goyal, Raj Jain, Sonia Fahmy, Bobby Vandalore  
The Ohio State University,  
Department of Computer and Information Science,  
2015 Neil Ave, DL 395, Columbus, OH 43210  
Phone: 614-688-4482  
{goyal,jain}@cis.ohio-state.edu

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

\*\*\*\*\*

DISTRIBUTION: ATM Forum Technical Committee  
Traffic Management Working Group

\*\*\*\*\*

DATE: July, 1998 (Portland)

\*\*\*\*\*

ABSTRACT: Section VII.2 of the baseline text contains two example implementations of the GFR service. Several other GFR implementations have been proposed in the past meetings and in recent literature. This contribution provides text to enhance section VII.2 by providing the various design options for GFR.

\*\*\*\*\*

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

\*\*\*\*\*

## 1 Introduction

Section VII.2 of the baseline text describes two sample implementations of the GFR service. We propose the following text to be added to modify section VII.2.

## 2 Motion

The following text should be used as a replacement for section VII.2 in the baseline text document:

### VII.2 Example Designs and Implementations of the GFR Service

There are three basic design options that can be used by the network to provide the per-VC minimum rate guarantees for GFR -- tagging, buffer management, and queueing:

- **Tagging:** Network based tagging (or policing) can be used as a means of marking non-conforming packets before they enter the network. This form of tagging is usually performed when the connection enters the network. Network based tagging on a per-VC level requires some per-VC state information to be maintained by the network. Tagging can isolate conforming and non-conforming traffic of each VC so that other rate enforcing mechanisms can use this information to treat the conforming traffic in preferentially over non-conforming traffic. For example, policing can be used to discard non-conforming packets, thus allowing only conforming packets to enter the network.
- **Buffer management:** Buffer management is typically performed by a network element (like a switch or a router) to control the number of packets entering its buffers. In a shared buffer environment, where multiple VCs share common buffer space, per-VC buffer management can control the buffer occupancies of individual VCs. Per-VC buffer management uses per-VC accounting to keep track of the buffer occupancies of each VC. Examples of per-VC buffer management schemes are Selective Drop and Fair Buffer Allocation. Per-VC accounting introduces overhead, but without per-VC accounting it is difficult to control the buffer occupancies of individual VCs (unless non-conforming packets are dropped at the entrance to the network by the policer).
- **Scheduling:** While tagging and buffer management control the entry of packets into a network element, queuing strategies determine how packets are scheduled onto the next hop. In a FIFO queue, packets are scheduled in the order in which they enter the buffer. As a result, FIFO queuing cannot isolate packets from various VCs at the egress of the queue. Per-VC queuing, on the other hand, maintains a separate queue for each VC in the buffer. A scheduling mechanism can select between the queues at each scheduling time. However, scheduling adds the overhead of per-VC queuing and the service discipline.

Table 1 lists the various options available for queuing, buffer management and support for tagged cells. A switch could use any of the available options in each category for its GFR implementation.

Table 1 GFR Options

|  |   |  |
|--|---|--|
| <b>Queuing</b>                             | FIFO  | Per-VC                                     |
| <b>Buffer Management</b>                   | Global Threshold<br>(No per-VC<br>accounting) | Per-VC Threshold<br>(per-VC<br>accounting) |
| <b>Tag Sensitive Buffer<br/>Management</b> | Supported                                     | Not Supported                              |

The following subsections list some sample GFR implementations based on this framework. Section VII.2.1 presents an implementation that uses Per-VC queuing with per-VC thresholds for untagged cells, as well as support for treating tagged cells separately from untagged cells. Section VII.2.2 presents a sample implementation with FIFO queuing and two global thresholds, i.e., it is sensitive to tags, but does not employ per-VC buffer management. Section VII.2.3 describes the Differential Fair Buffer Allocation Policy that uses FIFO queuing, per-VC thresholds and supports tagging by the source or the network.

### **VII.2.1 GFR Implementation using Weighted Fair Queuing and per-VC accounting**

(Unchanged)

### **VII.2.2 GFR Implementation Using Tagging and FIFO Queue**

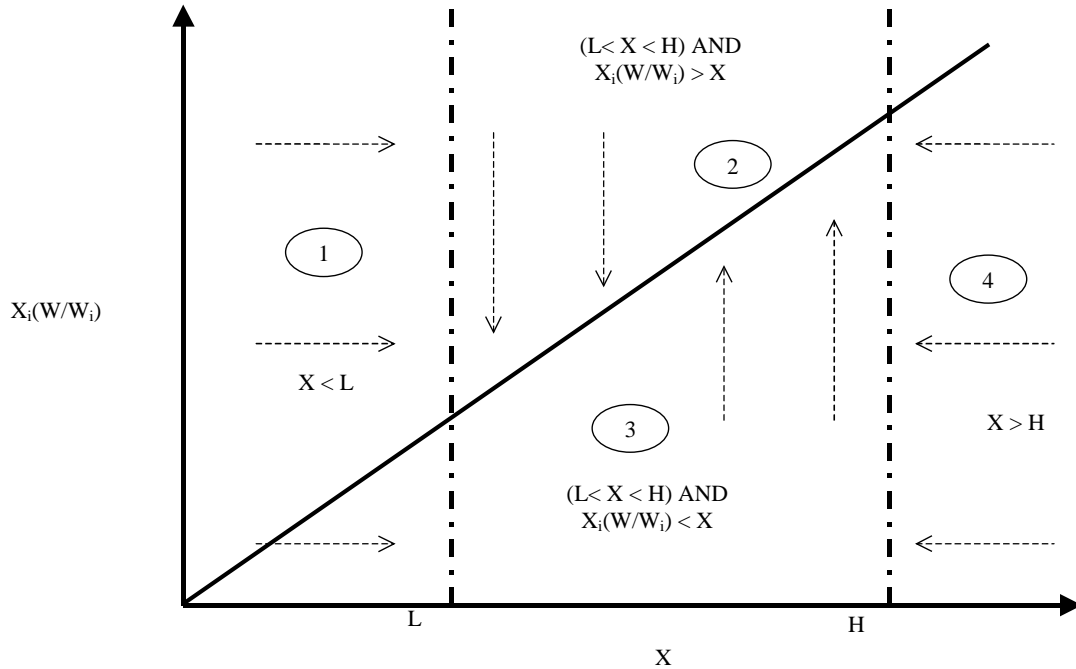
(Unchanged)

### **VII.2.3 GFR Implementation Using Differential Fair Buffer Allocation**

Differential Fair Buffer Allocation (DFBA) uses the current queue length as an indicator of network load. The scheme tries to maintain an optimal load so that the network is efficiently utilized, yet not congested. In addition to efficient network utilization, DFBA is designed to allocate buffer capacity fairly amongst competing VCs. This allocation is proportional to the MCRs of the respective VCs. The following variables are used by DFBA to fairly allocate buffer space:

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

DFBA tries to keep the total buffer occupancy ( $X$ ) between  $L$  and  $H$ . When  $X$  falls below  $L$ , the scheme attempts to bring the system to efficient utilization by accepting all incoming packets. When  $X$  rises above  $H$ , the scheme tries to control congestion by performing EPD. When  $X$  is between  $L$  and  $H$ , DFBA attempts to allocate buffer space in proportional to the MCRs, as determined by the  $W_i$  for each VC. When  $X$  is between  $L$  and  $H$ , the scheme also drops low priority (CLP=1) packets so as to ensure proportional buffer occupancy for CLP=0 packets.



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

In region 2, the packets of VC<sub>*i*</sub> are dropped in a probabilistic manner. This drop behavior is controlled by the parameter  $Z_i$ , whose value depends on the connection characteristics. This is further discussed below.

The probability for dropping CLP=0 packets from a VC when it is in region 2 depends on several factors. The drop probability has two main components – the fairness component,

and the efficiency component. Thus,  $P\{\text{drop}\} = \text{fn}(\text{Fairness component, Efficiency component})$ . The contribution of the fairness component increases as the VC's buffer occupancy  $X_i$  increases above its fair share. The drop probability is given by

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

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

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

BEGIN

IF ( $X < L$ ) THEN

Accept frame

ELSE IF ( $X > H$ ) THEN

Drop frame

ELSE IF ( $L < X < H$ ) AND ( $X_i < X * W_i / W$ ) THEN

Drop CLP1 frame

ELSE IF ( $L < X < H$ ) AND ( $X_i > X * W_i / W$ ) THEN

Drop CLP1 frame

Drop CLP0 frame with

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

ENDIF

END

## VII.2.4 Evaluation Criteria

(From VII.2.3 in the baseline text document.)