Comput Syst Sci & Eng (1999) 5: 275-282

Computer Systems
Science & Engineering

Impact of per-VC and single FIFO queuing on ABR congestion control

Ambalavanar Arulambalam*, Xiaoqiang Chen† and Nirwan Ansari‡

*Bell Laboratories, Lucent Technologies, Rm 2C-133, 600 Mountain Avenue, Murray Hill, NJ 07974, USA. Email: murali@emailbox.lucent.com
†Bell Laboratories, Lucent Technologies, Rm 4D-525, 101 Crawfords Corner Road, Holmdel, NJ 07733, USA. Email: xchen@research.att.com
‡Department of Electrical & Computer Engineering, New Jersey Institute of Technology, University Heights, Newark, NJ 07102, USA
Email: ang@njit.edu

Queuing disciplines and their impact on congestion of Available Bit Rate (ABR) service in Asynchronous Transfer Mode (ATM) networks are investigated. In particular, two queuing disciplines, namely, First-In-First-Out (FIFO) and per-VC (virtual connection) queuing, are examined. Performance in terms of fairness, throughput, cell loss rate, buffer size and network utilization are benchmarked via extensive simulations. Implementation complexity analysis and trade-offs associated with each queuing implementation are addressed. In contrary to the common belief, our investigation demonstrates that per-VC queuing which is costlier and more complex does not necessarily provide any significant improvement over the simple FIFO queuing. These significant results will be of great interest and industrial value to the ATM Forum and vendors. Congestion control algorithms should be designed in such a way that they function well even in the presence of simple queuing disciplines.

Keywords: Available Bit Rate, ATM, queuing disciplines, congestion control, rate-based control

1. INTRODUCTION

It has long been recognized that per-VC queuing in Asynchronous Transfer Mode (ATM) networks is prohibitive because its implementation does not scale with the number of connections to be supported, and it requires complex buffer management and scheduling algorithms and a high implementation cost. This very reason has led to the defeat of several congestion control proposals for ATM networks[1, 2]. The rapid growth in both use and size of ATM networks, however, has sparked a renewed interest in incorporating per-VC implementation in commercial ATM switches. Several ATM switch vendors have recently announced their per-VC implementation strategies. There is no clear technical evidence, however, to show what benefits per-VC queuing can provide to justify its implementation cost. The objective of this paper is to address this issue and provide insights and understanding of the impact of queuing disciplines on ATM networks.

As established principles, ATM provides the flexibility of integrating the transport of different classes of traffic with a wide range of service requirements. ATM also provides the potential to obtain resource efficiency through the statistical multiplexing of a diverse mix of traffic streams. Gains due to statistical multiplexing, however, come at the risk of potential congestion. Flow and congestion control in ATM networks is concerned with ensuring that users get their desired quality of service, and can broadly separate into two categories: open-loop control and closed-loop control. The objective of open-loop control is to ensure *a priori* that the network traffic intensity normally will never reach a level to cause unacceptable performance degradation. If a source is able to characterize in detail its traffic characteristics and performance requirements, the network can then take these values into account in the call acceptance decision (e.g. Call Admission Control). In addition, mechanisms such as policing units need to be provided at the entrance of the network to ensure that the source is consistent with the pre-specified traffic parameters. Constant Bit Rate (CBR) and Variable Bit Rate (VBR) services fall into this category. On the other hand, most data applications cannot predict their own traffic parameters, and usually require a service that dynamically shares the available bandwidth. Such a service is referred to as Available Bit Rate (ABR) which is based on closed-loop

feedback control that uses feedback information from the network to regulate the source rate.

At intermediate switches, queuing disciplines (buffer management and scheduling), which control the usage of buffer space and the order in which cells are sent, can affect the behavior of traffic flows, and if properly implemented, lessen congestion. However, queuing disciplines do not affect congestion and flow control directly in that they do not change the total traffic admitted into the network. Any flow controls, either open-loop or closed-loop, should be designed in such a way that they function well even in the presence of simple queuing disciplines in the network. We believe that implementing some class-based queuing strategies is necessary to handle complex dynamic traffic fluctuations in high-speed links, to isolate traffic flows and to ensure that quality of service for each class will be maintained. We shall argue, however, that per-VC queuing may not necessarily provide any significant performance improvement over simple queuing disciplines, and its implementation complexity may not be justified. To evaluate this, we examine two queuing disciplines and their impact on ABR traffic.

We had three goals in writing this paper. The first was to discuss the different queuing strategies possible to complement ABR traffic control. The second was to provide a good understanding of queuing disciplines versus different ABR implementations. This is done in section 4, where we present simulation results for a specific benchmark network. The third goal was to evaluate complexity and performance trade-offs between two queuing disciplines. This point is discussed in section 5. The main conclusion obtained in this study is that simple FIFO queuing is adequate for ABR control, where connection-based closed-loop control can handle congestion effectively and fairly. We observe that the explicit connection-based control can implicitly achieve the same effect as per-VC implementation. Following a similar line of reasoning, we believe that per-VC queuing may not even be required for open-loop controlled traffic such as CBR and VBR, and effective policing and traffic shaping on a per-connection basis may well protect well-behaved users, and ensure that quality of service can be achieved.

2. RATE-BASED CONTROL MECHANISM

The emergence of ATM technology into Local Area Networks (LANs) is the imminent avenue for various multimedia applications. Recently, the momentum for a rapid standardization of ATM has come from data networking applications. Most of these applications cannot predict their own bandwidth requirements. Thus, an explicit guarantee of service cannot be provided. ABR service, also known as best-effort service, was introduced to accommodate data applications which usually require a service that dynamically shares the available bandwidth among all active users. ABR service is specified in such a way that it can co-exist with prior traffic classes such as CBR and VBR. Users of ABR service will dynamically share the leftover bandwidth after all the guaranteed traffic classes have been served. The success for ABR services depend on how the traffic in the network is managed, and one of the challenges is how to react in the event of network congestion. Because of its bursty nature, congestion control for ABR service poses more challenging problems than other services, and it is the focus of recent standardization efforts at the ATM Forum (a consortium of companies that writes specifications to accelerate the development and standardization of ATM technology).

After a considerable debate, the ATM Forum has adopted a rate-based congestion control algorithm, which is based on the closed-loop feedback flow control principle, to control congestion for ABR traffic in ATM networks. The ABR congestion control scheme is a per-connection control that uses feedback information from the network to control the rate at which the source transmits cells for each virtual connection (VC). The basic element of the control is to provide a proportional rate adjustment, where the increase and the decrease in source rates are proportional to the current rate. Information about the state of the network, such as bandwidth availability, state of congestion, and impending congestion, is conveyed to the source through special probe cells called Resource Management Cells (RM-cells). The scheme is based on a closed-loop, 'positive feedback' rate control principle[3]. Here, the source only increases its sending rate for a connection when given an explicit positive indication to do so, and in the absence of such a positive indication, continually decreases its sending rate. The decision to 'remove an opportunity for a rate increase' is made independently by each intermediate network based on the state of the resources it is protecting. The decision may be based on a cell queue depth in a switch or a threshold on an aggregate rate of cells flowing on a link. This allows considerable freedom in network equipment design, and allows the network provider to trade off cell buffer memory and link bandwidth utilization.

2.1 Basic mechanisms

The typical operation of the rate-based control framework is illustrated in Figure 1. Once the source has received permission, it begins cell transmission. The rate at which an ABR source is allowed to schedule cells for transmission is denoted as Allowed Cell Rate (ACR). The ACR is initially set to the Initial Cell Rate (ICR) and is always bounded between the Minimum Cell Rate (MCR) and the Peak Cell Rate (PCR). Transmission of data cells is preceded by sending an RM cell. The source will continue to send RM cells, typically after every N_{RM} data cells. The source rate is controlled by the return of these RM cells, which are looped back by the destination or by a virtual destination.

The source places the rate at which it is allowed to transmit cells (its ACR) in the Current Cell Rate (CCR) field of

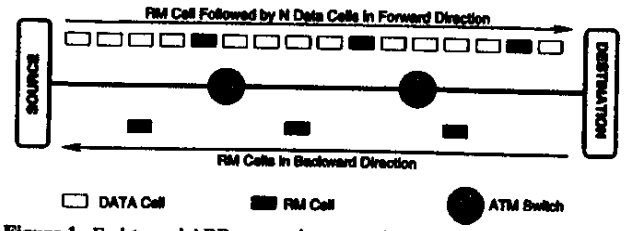

Figure 1 End-to-end ABR congestion control

the RM cell, and the rate at which it wishes to transmit cells (usually the PCR) in the Explicit Rate (ER) field. The RM cell travels forward through the network, thus providing the switches in its path with the information in its content for switches' use in determining the allocation of bandwidth among ABR connections. Switches may also decide at this time to reduce the value of the explicit rate field ER, or set the Congestion Indication (CI) bit to 1. Switches supporting only the Explicit Forward Congestion Indication (EFCI) mechanism (by which an indicator in the header of each data cell is set under congestion) will ignore the content of the RM cell. Switches optionally may generate a controlled number of ABR RM cells on the backward path, in addition to those originally supplied by the source. Switch-generated RM cells must have the Backward Notification (BN) bit set to 1 and either the CI bit or the No Increase (NI) bit set to 1.

When the cell arrives at the destination, the destination should change the direction bit in the RM cell and return the RM cell to the source. If the destination is congested and cannot support the rate in the ER field, the destination should then reduce ER to whatever rate it can support. If, when returning an RM cell, the destination observed a set EFCI since the last RM cell was returned, then it should set the RM cell's CI bit to indicate congestion.

As the RM cell travels through the network, each switch may examine the cell and determine if it can support the ER for this connection. If the ER is too high, the switch should reduce it to the rate that it can support. No switch should increase the ER, since information from switches previously encountered by the RM cell would then be lost. The switches should try to modify the ER for only those connections for which there is a bottleneck, since this promotes a fair allocation of bandwidth.

When the RM cell arrives back at the source, the source should reset its ACR, based on the information carried by the RM cell. If the congestion indication bit is not set (CI = 0), then the source may increase its ACR by a fixed increment determined at call setup, toward (or up to) the ER value returned, but never exceeding the PCR. If the congestion indication bit is set (CI = 1), then the source must decrease its ACR by an amount greater than or equal to a proportion of its current ACR, the size of which is also determined at call setup. If the ACR is still greater than the returned ER, the source must further decrease its ACR to the returned ER, although never below the MCR. A set NI bit tells the source to observe the CI and ER fields in the RM cell, but not to increase the ACR above its current value. This can be expressed as follows:

$$\text{ACR} = \begin{cases} \max(\min(PCR, ER, ACR + PCR\,RIF), MCR) \\ \qquad \text{if } CI = 0 \text{ and } NI = 0, \\ \max(\min(PCR, ER, ACR\,(1 - N_{RM}/RDF)), MCR) \\ \qquad \text{if } CI = 1, \end{cases} \quad (1)$$

where RIF is the Rate Increase Factor, and RDF is the Rate Decrease Factor. Note that the factors RIF and RDF control the rate at which the source increases and decreases its rate, respectively. Additional operational details of the ABR specifications can be found elsewhere[4, 5].

2.2 Fairness issues and switch mechanisms

The EFCI-based switches suffer from a phenomenon called the *beat-down problem*. In a network using only EFCI-based switches, where a congested switch marks the EFCI bit of the data cell, sources traveling more hops have a higher probability of getting their cells marked than those traveling fewer hops. Consequently, a source traveling more hops gets a lower share of bandwidth than the ones traveling fewer hops.

To overcome the beat-down problem, one may use the CCR value found in the RM cell and selectively indicate congestion on connections traversing a congested link to ensure fair rate allocation among the competing connections. That is, during congestion periods, some connections with a CCR value higher than their fair share will be signaled to reduce rate, while others whose CCR is lower than their fair share, may be allowed to increase rate. This is sometimes referred to as 'intelligent marking' or 'selective marking'.

The basic option offered by the approach described above consists of single-bit congestion feedback, supported via EFCI mechanisms in the switches and the CI bit in the RM cells. In addition to EFCI marking and selective marking, the switches may employ sophisticated switch mechanisms, which compute an explicit rate value that is fed back in the appropriate RM field[6, 7]. These more advanced approaches require switches that compute the fair share for each connection in a distributed fashion and explicitly set the source transmission rate (i.e. ER). This specific rate may be used by intermediate networks with small cell buffers to drive a connection rate lower to quickly respond to transient conditions where, say, a large number of idle connections sharing a link become active within a short time. Many example algorithms featuring sophisticated intelligent and explicit-rate mechanisms have been proposed which demonstrate the enormous potential of the rate-based approach[6–9].

In this study, the algorithm developed at the University of California, Irvine, is used as the ER calculating method. The main idea of this scheme is to use the CCR information available in the RM cells to adaptively adjust the cell rates so that the rates will converge to a fair share. Each queue in the switch maintains the mean of allowed cell rates (MACR) and selectively marks the ER value based on the CCR and the congestion level of the queue. Further details on this approach can be found in Siu and Tzeng[7].

In summary, the rate-based, closed-loop control mechanism for ABR services defines the source-end and the destination-end system behavior, and it defines the means of forward and backward notification of congestion information. The control is done on the ATM cell-level (the ATM Layer), and since an RM cell is created for each VC, the control is implicitly done on a per-VC basis. Through the effective congestion detection and congestion notification process (i.e. EFCI-based and ER-based), the congestion control scheme will provide fair and efficient service to ABR service users. The main goal of the rate-based approach is to provide a flexible framework of ABR congestion control, which includes a broad range of possible implementations characterized by varying degrees of cost and complexity. The present generation of ATM switches which will be deployed in the next couple of years have the capability of providing EFCI-based control. The new generation of switches may consist of various implementation styles with different levels of performance and complexity. It is vital that the implementation of new-generation switches not dictate the overall

functionality of the control mechanism, that is, they must co-exist with the simpler switches in the ATM network.

3. BUFFER MANAGEMENT AND CELL SCHEDULING

A network switch provides the necessary resources, namely, port bandwidth and buffers, for routing ATM cells. The port bandwidth and the buffer size are limited resources and are heavily contended. The heavy contention over a period of time may lead to congestion and, as a consequence, to possible loss and excessive delay of ATM cells. Therefore, in order to provide high quality-of-service, a switch must employ efficient buffer management techniques and cell scheduling policies. As we have already mentioned, the rate-based framework provides many degrees of freedom in the behavior and implementation of network switches. One such freedom is the capability of employing various buffer management strategies and cell scheduling policies.

Buffer management strategies control the usage of buffer space, monitor the level of congestion in its resources, and employ cell discarding strategies. Cell scheduling policies decide how to schedule each cell and transmit over the link (i.e. the order in which the cells are transmitted). There are several options for such a buffer management scheme. For example, single FIFO queuing or per-VC queuing can be used as two distinct techniques. In the single FIFO queue approach, the queue is served in a FIFO manner, and thus there is no need for a complex scheduling mechanism. The per-VC queuing approach may employ various cell scheduling mechanisms, such as weighted round-robin or weighted fair queuing[10]. In addition to queuing disciplines, effective congestion detection techniques that have broad implications on implementation complexity and performance must be implemented. The congestion detection may be based on the use of the following methods:

1. a single threshold,
2. multiple thresholds,
3. differential of queue length,
4. output port link utilization, or
5. variation of successive cell delay for same connection.

The single FIFO queuing and per-VC queuing techniques are discussed below. For both approaches a two threshold-based congestion detection mechanism is used.

3.1 Single FIFO queuing

In this approach, a centralized output-port memory is completely shared by a single queue, where all the cells from different sources form a single queue and the cells are scheduled in a FIFO manner. This type of discipline is the simplest, most economical and commonly implemented queuing discipline. If the queue length exceeds the available buffer space, the incoming cells are discarded. To minimize cell-loss, congestion must be detected effectively. In this work, congestion detection is done by using two queue thresholds. When the queue length is above the high-threshold (Q_{HT}) level, the congestion indication flag, $\sigma(t)$, is set to 1 (i.e. $\sigma(t) = 1$) and remains set until the queue length drops below the low-threshold (Q_{LT}) level. This can be mathematically expressed as follows:

$$\sigma(t) = \begin{cases} 1 & \text{if } Q(t) > Q_{HT} \\ 1 & \text{if } Q(t) > Q_{LT} \text{ and } \sigma(t) = 1 \\ 0 & \text{if } Q(t) < Q_{LT} \end{cases}$$

This two threshold detection method ensures that the oscillations are minimized. This detection mechanism is illustrated in Figure 2.

3.2 Per-VC queuing

Unlike the first approach where all the VCs are queued in one single queue, in the per-VC approach, cells from different VCs are queued in separate queues and the buffer space is allocated on a per-VC basis. The per-VC queuing approach is illustrated in Figure 3. Multiple classes of traffic with varying degrees of priority and delay requirements can be fairly served with the per-VC implementation in conjunction with a fair scheduling policy such as the weighted fair queuing or weighted round-robin scheduling policy. The output-port buffer space could be divided among all the VCs in a fixed manner or dynamically shared among VCs. In the fixed buffer size allocation method (static), each VC is only allowed to occupy its own VC buffer share, but in adaptive buffer management, VCs can take up more than their share. When the buffer becomes full, however, a cell from the queue with the largest queue length is dropped to allow room for the newly arriving cells. The adaptive buffer management scheme utilizes the buffer space more efficiently than a fixed buffer management scheme without wasting any buffers. In order to compare the performance of per-VC queuing with single FIFO queuing, the state of congestion is determined similarly using the two-threshold approach described earlier. When the aggregate sum of all individual

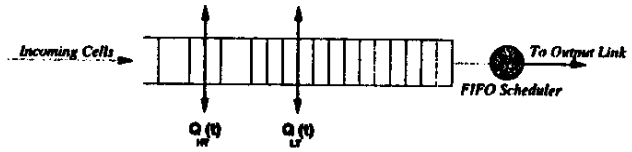

Figure 2 Single FIFO queuing and two threshold congestion detection approach

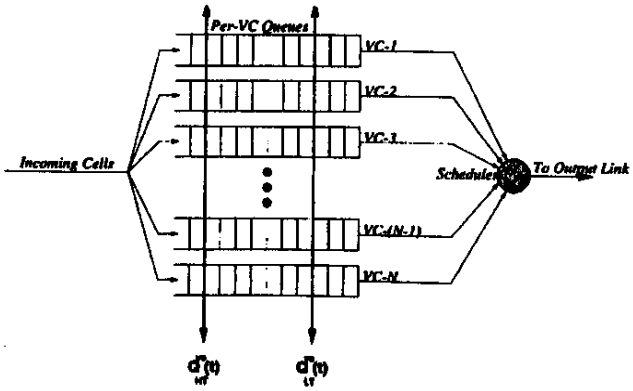

Figure 3 Per-VC queuing and two threshold detection approach

queue lengths is above the high-threshold (Q_{HT}) level, the individual queue lengths are compared with the per-VC-threshold, Q_{HT}^{vc}, and the congestion indication flag for the VC, $\sigma^{vc}(t)$, which exceeds the threshold, is set. This congestion flag remains set until the individual queue length drops below its low-threshold, Q_{HT}^{vc}

$$\sigma^{vc}(t) = \begin{cases} 1 & \text{if } Q^{vc}(t) > Q_{HT}^{vc} \\ 1 & \text{if } Q^{vc}(t) > Q_{LT}^{vc} \text{ and } \sigma^{vc}(t-) = 1, \\ 0 & \text{if } Q^{vc}(t) < Q_{LT}^{vc} \end{cases} \quad (3)$$

The per-VC queue lengths are calculated as follows.

$$Q_{HT}^{vc} = \frac{Q_{HT}(t)}{N_{vc}} \quad (4)$$

$$Q_{LT}^{vc} = \frac{Q_{LT}(t)}{N_{vc}} \quad (5)$$

It is essential to note that, in the simulations below, we are not concerned with any priority or delay requirements for ABR traffic, and thus the scheduling mechanism we incorporate (i.e. weighted round-robin) translates into a simple round-robin scheduling policy.

4. SIMULATION RESULTS AND DISCUSSIONS

4.1 Multi-hop network configuration

To study the impact of queuing policies on the ABR congestion control scheme in ATM networks, a simple network, shown in Figure 4, is simulated. This network configuration is referred to as the Generic Fairness Configuration 1 (GFC1), and it is one of the benchmark configurations recommended by the ATM Forum[5]. The GFC1 network consists of five switches and 23 connections grouped into six classes (A-F). In Figure 4, the number inside the parentheses next to the group label represents the number of VCs for that group. VCs in groups C, D, E and F are single-hop traffic, and VCs in groups A and B are three-hop cross-traffic. The links connecting hosts to switches have a capacity of 150Mbps. All the links are 400 meters in length, and have a propagation delay of 4μs per Km. For performance measurement purposes, the switches are assumed to be non-blocking and output buffered.

4.2 Source characterization

The sources are assumed to be well behaved, persistently greedy, and can transmit at the peak link rate when the bandwidth is available. The cell transmission rate, however, is flow controlled in accordance with the ABR control specifications. The use of persistent sources presents a tough challenge for multiple congested links.

In this study, the following scenarios, which represent possible combinations of implementations, are simulated and the results are compared in terms of fairness, throughput, link-utilization, cell-loss rate and switch output-port memory utilization. The parameters used throughout this study are tabulated in Table 1.

- **S1**: Simple EFCI marking with single-FIFO queuing
- **S2**: Simple EFCI marking with per-VC queuing
- **S3**: Selective marking with single-FIFO queuing
- **S4**: Explicit Rate marking with single-FIFO queuing
- **S5**: Explicit Rate marking with per-VC queuing

4.3 Throughput and fairness

One of the major goals of ABR service is to achieve fairness while maximizing the throughput. The ATM Forum has adopted the notion of the *max-min* fair allocation principle, which has been studied extensively in the literature as the basis for the evaluation of fairness[6, 11]. The max-min criterion provides the maximum possible bandwidth to the source receiving the least among all contending sources. This is done by first maximizing the link capacity allocated to the users with the minimum allocation, and then using the remaining link capacity for other users in a way that it maximizes the allocation of the most poorly treated users. The

Table 1 GFC1 simulation parameters

Parameter	*EFCI Mark*	*Intelligent Mark*	*ER Stamping*
N_{RM}	32	32	32
PCR (Mbps)	150	150	150
MCR (Mbps)	0.150	0.150	0.150
ICR (Mbps)	7.5	7.5	7.5
RIF	0.0001	0.0001	0.0001
RDF	256	256	256
Buffer Size	750	750	750
Q_{LT}	50	50	50
Q_{HT}	100	100	100
VCS	NA	0.875	0.875
DPF	NA	0.875	0.875
DQT	NA	500	500
AV	NA	0.0625	0.062
ERF	NA	NA	0.94
MRF	NA	NA	0.25

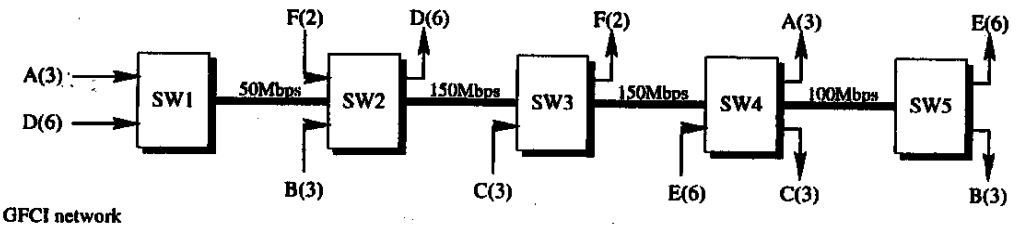

Figure 4 The GFC1 network

Table 2 Throughput comparison for the GFC1 network

Group	Expected	EFCI-single	EFCI-per-VC	Select-FIFO	ER-single	ER-per-VC
A	5.56	3.13	5.18	5.33	5.31	5.37
B	11.11	5.67	10.18	10.50	10.68	10.80
C	33.33	36.81	30.40	32.06	32.4	32.30
D	5.56	6.48	5.19	5.39	5.41	5.37
E	11.11	12.76	10.18	10.96	10.76	10.78
F	50.00	53.27	46.68	50.15	48.26	48.52

max-min principle is fair since all VCs sharing a link get an equal share of bandwidth provided they can all use the fair share. Moreover, the max-min principle is efficient in the sense that it maximizes the throughput. The second column in Table 2 shows the max-min fair allocation of each group. Table 2 also shows the average throughput (in Mbps) achieved with different simulation scenarios.

Figure 5 illustrates the impact of queuing policies on fairness using the EFCI and the ER marking schemes. The fairness percentage is calculated by computing average throughput over a fixed time and dividing it by the fair value of the throughput. From the figure it can be seen that marking only the EFCI bit and using the single FIFO queuing policy (case S1) leads to unfairness among classes. In particular, connections in Groups A and B, which compete for bandwidth in multiple links, receive a lower than the max-min fair share, while connections in the other classes take advantage of this situation and receive more than their fair share. Using a single FIFO queue in conjunction with selective marking results in a fair bandwidth allocation (case S3). Since only the VCs that exceed their fair share are marked, the beat-down effect disappears. Similarly, the EFCI marking with per-VC queuing (case S2) leads to global fairness among all the groups. A simple EFCI marking scheme with per-VC queuing solves the beat-down problem even though per-VC queuing is a local policy; namely, per-VC queuing isolates individual connections and has the selective effect under congestion, where only the cells from the VCs using the buffer extensively are marked. Implementing ER schemes using either queuing discipline (cases S4 and S5) produce almost identical fairness. Since the ER mechanism implemented in the switches computes the fair share and informs the ER via RM cells, queuing policies do not provide any additional advantages in terms of fairness using the ER method. The ER marking schemes allow the sources to reduce the sending rate very quickly under severe congestion situations and help the switches recover from congestion.

4.4 Link and buffer utilization

High link or network utilization is critical to ensure efficient and profitable operation of the network. The average link utilization at the steady state is shown in Figure 6. From the figure it can be seen that the ER mechanism using either queuing discipline (cases S4 and S5) and the selective marking scheme (case S3) results in high utilization of the links. However, the simple EFCI mechanism under-utilizes the network regardless of the queuing policies implemented. Network utilization is highly dependent on buffer occupancy levels. The utilization can be made very high by simply adjusting the parameters (HT, LT) to force the queue length to increase. With a larger queue, slight variations originating from on-off sources causing oscillations in buffer usage are absorbed by the queue without the queue ever becoming empty. However, the larger the queue length, the higher the delay. Therefore, there exists a clear trade-off among buffer size, delay and network utilization. Figures 7 and 8 illustrate peak queue length and the steady-state average queue length of switch ports. The queue peaks shown are for the worst case, including both the initial transient period and the steady-

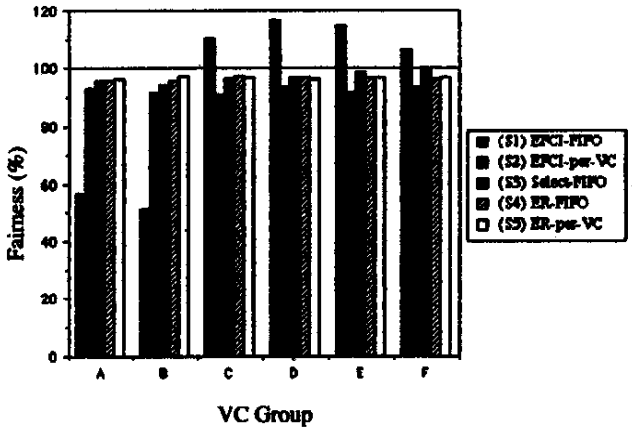

Figure 5 Impact of queuing policies on fairness

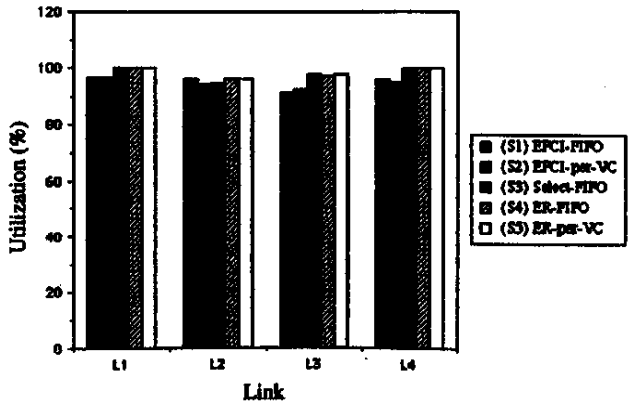

Figure 6 Impact of queuing policies on average queue length

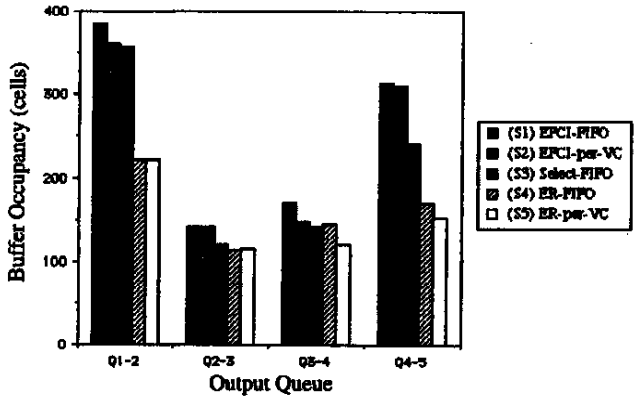

Figure 7 Impact of queuing policies on peak queue length

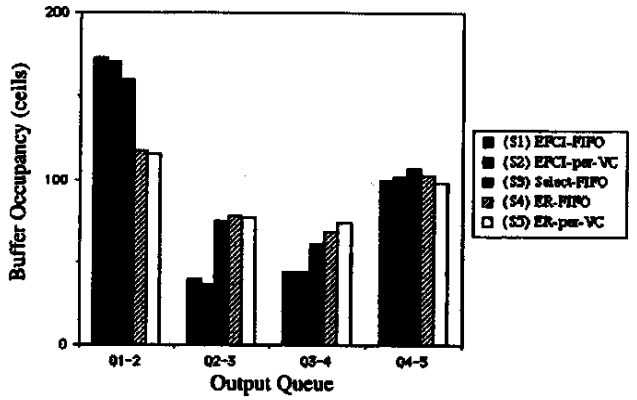

Figure 8 Impact of queuing policies on average queue length

state region. Using single-FIFO or per-VC queuing disciplines in conjunction with the ER mechanism does not offer any further advantages. In both cases, the resulting average queue length and peak queue length are almost identical. From these results it is clear that the per-VC approach does not save any buffer space or produce higher utilization than the single FIFO queue.

4.5 Cell-loss rate

For all the scenarios simulated, no cell loss was observed. This shows that for the same amount of memory, the queuing policies do not have an impact on cell-loss. It is evident from the peak queue length that the two queuing disciplines do not have significant differences. This indicates that for a given amount of buffer space, the queue behaves similarly for both the single FIFO and the per-VC queuing approaches. Thus, the queuing policies does not have an impact on ABR cells. Again, since the rate-based approach implicitly controls the rate at which each VC transmits cells, the overall system works very effectively without providing any complex local queuing and scheduling policy. A single queue with enough intelligence may be able to provide the minimum cell-loss requirement desired by the ABR traffic sources.

5. PERFORMANCE – COMPLEXITY TRADE-OFFS

The choice among different queuing disciplines influences the implementations in a significant way. The implementation of these two queuing disciplines comes widely varying complexity. In the past, with lower speed networks, the issue of implementations of buffer management and scheduling policies was somewhat less important, since much of it is implemented in software. ATM, however, is meant to be scalable to much higher link or port speeds (i.e. 2.4 Gbps), and thus the implementation of control schemes needs to be performed in hardware. In this section we will compare the advantages and disadvantages of the FIFO and per-VC queuing policies, and address the implementation issues.

A shared port buffer served in a FIFO fashion is the simplest, most economical and commonly implemented queuing discipline. Because of this simple nature it is easily implementable particularly at very high-speed ports. It does not, however, provide any local mechanisms to enforce a fair access to buffers and bandwidth, and it leaves such resources open to abuse by malicious users. Due to this unfair access to buffers and bandwidth, the simple EFCI-marking control scheme suffers considerably in terms of fairness. This fairness problem can be overcome by using ER-based switches, which calculate the fair share for each VC and inform the source of this fair rate. In addition to intelligent techniques, the single-FIFO queuing discipline will require an external mechanism that serves as a policing function to alleviate the problems caused by malicious sources.

The per-VC approach requires switches to keep a separate queue for each VC. The accounting of occupancy of the buffer is performed on an individual VC basis. The isolation provided by the separate queues ensures fair access to buffer space and bandwidth. This also allows the delay and loss behavior of individual VCs to be isolated from each other. Furthermore, per-VC information is readily available to help congestion control, such as early packet discard mechanisms. This per-VC information can also be used to help police misbehaving users effectively. It is important to note that in a static buffer management scheme, where the VCs are given a fixed buffer share, the policing can be completely eliminated. This is because in the static buffer scheme if a VC is misbehaving, only its queue will grow and overflow. On the other hand, the adaptive buffer scheme must utilize some intelligent mechanism in order to achieve efficient policing.

Although per-VC queuing offers some advantages over single FIFO queuing, per-VC implementation suffers considerably in terms of implementation and scheduling complexity. Since per-VC queuing causes switch complexity to be proportional to the number of VCs, the approach will not scale well given that some large switches will support millions of VCs causing considerable complexity. In addition, complex scheduling policies must be implemented on a per-VC basis, which adds extra complexity and cost. As an example, consider a switch-output port operating at 155 Mbps with corresponding cell time of 2.75μs. In order to achieve full efficiency of the line and cell-backlog, the switch must process and schedule each cell within 2.75μs. In the future, the link speeds may go up to 622 Mbps or even up to 2.4 Gbps with corresponding cell-times of 0.68μs and

0.18μs. In addition to link speeds, the size of the network is expected to grow, which makes it necessary for the switches to handle millions of VCs. Implementing per-VC queuing means that a switch must be able to handle complex cell scheduling techniques. The scheduling techniques may be implemented without any added cost for low-speed switch ports with a small number of VCs. On the other hand, at higher link speeds and with the increasing number of VCs, the complex scheduling mechanisms have to process each cell within the time specified above that will consequently result in very complex and expensive hardware.

6. CONCLUSIONS AND SUMMARY

We have examined two queuing disciplines in terms of their performance and complexity in the presence of ABR traffic and rate-based control. In studying the impact of queuing policies, it is clear that there is no significant increase in performance using per-VC queuing over single FIFO queuing. Per-VC queuing is good if selective marking has not been implemented for EFCI switches, but it is unnecessary for ER switches. In the context of ABR traffic, since the rate-based scheme implicitly controls congestion on a per-VC level, a single-FIFO queue with some intelligent marking or explicit rate setting scheme can achieve the same performance as the per-VC approach. Furthermore, selective EFCI marking is considerably simple in the implementation against per-VC queuing, and can sustain as good performance as per-VC queuing in terms of fairness, throughput, and link-utilization. Using a similar line of reasoning we can argue that the effective and properly implemented open-loop control for CBR and VBR traffic may eliminate the necessity to implement per-VC queuing discipline.

This conclusion, however, does not exclude the merits that per-VC queuing can provide; namely, isolation, fairness and elimination of policing. On the other hand, the hardware complexities and non-scalable nature of the per-VC queuing approach make it very costly to implement in real networks. It is no doubt that in some places in the network, such as virtual source (VS) and virtual destination (VD) terminating points, the per-VC queuing is required, At these points control loop needs to be segmented to apply proprietary control schemes or to add extra protection. Moreover, the per-VC-queuing approach may be cost-effectively used at the network entry points where traffic shaping and policing are necessary. This, however, in no way justifies per-VC queuing at every switching point.

This study has also shedded some insight into the performance of queuing disciplines in the presence of various classes of traffic, such as CBR and VBR traffic. It is certain that at the connection level, the single-FIFO approach will not be sufficient in guaranteeing the quality of service requirements for CBR and VBR traffic, and handling the complex dynamic traffic fluctuations. This, however, does not mean that we must resort to the per-VC approach. Rather, a per-class queuing approach may be implemented and justified in terms of performance, complexity and cost.

ACKNOWLEDGEMENTS

Received January 30 1996; accepted for publication September 16 1996.

REFERENCES

1 **Kung, HT and Morris, R** Adaptive Credit Allocation for Flow Controlled VCs. *IEEE Network Magazine* (March-April 1995) pp 40–48.

2 **Ramakrishnan, KK and Newman, P** Integration of Rate and Credit Schemes for ATM Flow Control. *IEEE Network Magazine* (March-April 1995) pp 49–56.

3 **Hluchyj, H and Yin, N** On Closed-Loop Rate Control for ATM Networks. *Proc. INFOCOM'94* (1994) pp 99–108.

4 **The ATM Forum** *The ATM Forum Traffic Management Specification, Version 4.0.* ATM Forum Contribution (1995)

5 **Bonomi, F and Fendick, KW** The Rate-Based Flow Control Framework for the Available Bit Rate ATM Service. *IEEE Networks Magazine* (March-April 1995) pp 25–39.

6 **Charny, A, Clark, DD and Jain, R** Congestion Control with Explicit Rate Indication. *Proc. ICC 95* (June 1995).

7 **Siu, K and Tzeng, H** Adaptive Proportional Rate Control for ABR Service in ATM Networks. *Technical Report 94-07-01*, Electrical and Computer Engineering, University of California, Irvine, CA (July 1994).

8 **Jain, R, Kalyanaraman, S and Viswanthan, R** The OSU Scheme for Congestion Avoidance using Explicit Rate Indication. *Technical Report OSU-CISRC-1/96-TR02*, Ohio State University, Columbus, OH (September 1996).

9 **Roberts, L** Enhanced PRCA (Proportional Rate-Control Algorithm). *The ATM Forum Contribution No. 94-0735R1* (August 1994).

10 **Demers, A, Keshav, S and Shenker, S** Analysis and Simulation of a Fair Queueing Algorithm. *Proc. of ACM SIGCOMM'89* (1989) pp 3–12.

11 **Bertsekas, D and Gallager, D** *Data Networks.* Prentice Hall, Englewood Cliffs, NJ, 2nd ed. (1987).