Quality of Service for Internet Traffic over ATM Service Categories[1]

Sonia Fahmy[†], Raj Jain[†], Sameh Rabie[‡], Rohit Goyal[†] and Bobby Vandalore[†]

[†]Department of Computer and Information Science

The Ohio State University, Columbus, OH, USA

New Address: Raj Jain, Washington University in Saint Louis, jain@cse.wustl.edu, http://www.cse.wustl.edu/~jain

[‡]Nortel Networks

Ottawa, Ontario, Canada

E-mail: rabie@nortelnetworks.com

ABSTRACT

Connecting enterprise sites requires innovative architectures. Virtual private networks (VPNs) linking different organizational sites over the *Internet* are a popular solution. Internet traffic, however, is rapidly growing and becoming increasingly diverse. There is a strong need for quality of service (QoS) support in the Internet. Asynchronous transfer mode (ATM) backbones supporting QoS are already widely deployed in carrier networks. ATM offers a number of service categories. Each of the ATM services has its merits and limitations, so a tradeoff is necessary in selecting the service category for carrying Internet traffic between enterprise sites.

In this paper, we compare the ATM service categories in terms of cost, buffer requirements, and performance with Internet traffic. We find that the ATM available bit rate (ABR) service provides a good synergy with the emerging Internet technologies for supporting end-to-end QoS. Connecting enterprise networks by ABR virtual path connections can guarantee quality of service and minimize queuing delay and loss in the backbone. In addition, it provides flexibility in supporting various implementations at the edge devices.

Keywords: virtual private networks (VPNs), enterprise networks, ATM networks, traffic management, ATM service categories, TCP/IP, UDP, Internet differentiated services

[1] To appear in "Computer Communications" special issue on enterprise networks. This paper and all our papers and contributions are available through http://www.cis.ohio-state.edu/˜jain/

1 Introduction

Virtual private networks (VPNs) are rapidly gaining popularity. A VPN uses the public Internet to transparently connect private networks or even users, as if they are on the same network. Enterprise sites connected through the Internet are becoming increasingly common, especially within companies with multiple locations separated by long distances (figure 1). VPNs provide an attractive solution because of their reduced costs (over leased lines), reduced administration overhead, and support for remote access and collaboration with partners.

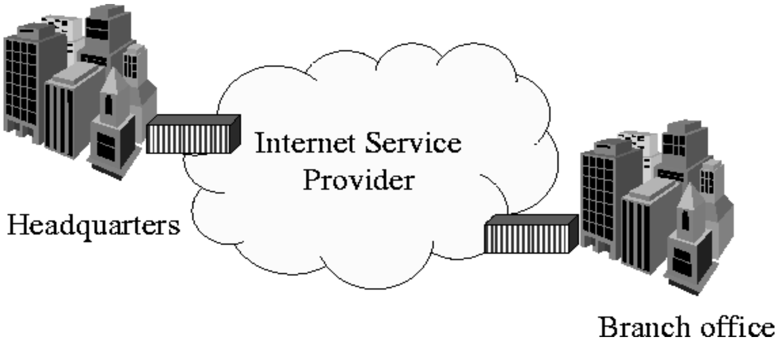


Figure 1: Virtual private networks connect enterprise sites over the Internet.

Internet traffic can be classified according to the application generating it, and the traffic characteristics. Applications may be real-time (voice or video) or non-real-time (data). Both the application type and the transport protocol affect the traffic characteristics. Unlike the user datagram protocol (UDP), the transmission control protocol (TCP) has built-in congestion avoidance mechanisms, which affect the traffic characteristics as seen at lower layers of the protocol stack.

End-to-end quality of service is critical to the success of current and future applications. QoS in the Internet is important for several reasons. Critical applications (such as real-time auctions and transactions) should be given priority over less critical ones (such as web surfing). Furthermore, many multimedia applications require delay or delay variation guarantees for acceptable performance. Weighted fairness is also important both among customers or aggregates (depending on the tariff or subscription), and also within an aggregate (for example, to prevent starvation among sessions or service categories).

Asynchronous transfer mode (ATM) is proposed to transport a wide variety of services, such as voice, video and data, in a seamless manner. ATM cells flow along predetermined paths called virtual channels (VCs). End systems must set up virtual channel connections (VCCs) of appropriate service categories prior to transmitting information. Service categories distinguish a small number of general ways to provide QoS, which are appropriate for different classes of applications. A representative list of current and future applications includes video, voice, image and data in conversational, messaging, distribution and retrieval modes. The required service categories can be derived from the properties of the application. ATM service categories distinguish real-time from non-real-time services, and provide simple and complex solutions for each case. The added mechanisms in the more complex categories are justified by providing a benefit or economy to a significant subset of the applications [8].

Since ATM is widely deployed in Internet backbones, traffic management for Internet traffic over ATM is becoming an increasingly important problem. Aggregation of Internet (IP) flows is necessary for scalability, overhead reduction, fast re-routing and simplified billing. Examples of aggregation in ATM include the use of virtual path connections (VPCs) that include several VCCs, and sub-multiplexing techniques within a VCC (for example, carrying multiple IP flows within an ATM VCC). In the Internet Engineering Task Force (IETF), Internet differentiated services are the best example of quality of service for aggregate flows. An example scenario is a customer buying a fixed width pipe, with multiple QoS streams occupying percentages of the pipe: 10% premium or guaranteed, 20% real-time, 30% excellent effort data and 40% best effort data.

This paper compares ATM service categories for Internet traffic transport, and shows that the ATM available bit rate (ABR) service can be used in backbones to connect various enterprise sites with QoS guarantees. The remainder of the paper is organized as follows. The next two sections give an overview of the ATM service categories and their applications. Then, we explain our proposed architecture and give sample simulation results. We also compare the costs of each category, and discuss the performance of TCP and UDP traffic over each. The paper concludes with a brief comparison of the service categories, and a discussion of their use for connecting enterprise networks.

2 Overview of ATM Service Categories and their Applications

ATM networks currently provide five service categories [7]: constant bit rate (CBR), real-time variable bit rate (rt-VBR), non-real-time variable bit rate (nrt-VBR), unspecified bit rate (UBR), and available bit rate (ABR). The CBR and rt-VBR services are intended to transport real-time traffic, while the nrt-VBR, UBR and ABR services are designed for non-real-time traffic. In addition to these categories, the guaranteed frame rate (GFR) service is currently being standardized at the ATM forum traffic management working group [18]. The ITU-T I.371 also defines similar (but not the same) categories called ATM transfer capabilities.

Service categories relate traffic characteristics and QoS requirements to network behavior. Table 1 shows the attributes supported for each service category (this table is adapted from the ATM forum traffic management specifications [7]). The traffic parameters of service categories are the peak cell rate (PCR), cell delay variation tolerance (CDVT), sustainable cell rate (SCR), maximum burst size (MBS), maximum frame size (MFS) and minimum cell rate (MCR). These parameters define the characteristics of the traffic being transported. Three quality of service parameters define the service level that can be expected for the connection. The quality of service parameters are the peak-to-peak cell delay variation (peak-to-peak CDV), maximum cell transfer delay (maxCTD), and cell loss ratio (CLR).

The CBR service is used by connections requesting that a constant amount of bandwidth (characterized by a peak cell rate) be available throughout the connection lifetime. The source can transmit at or below the PCR for any length of time, and the network assures the negotiated quality of service. Examples of applications that may use the CBR service are voice, video and circuit emulation applications requiring tight delay variation constraints.

The rt-VBR service is also intended for real-time applications requiring tight delay and delay variation constraints. Examples of such applications include voice with silence suppression, as well as emerging compressed video traffic. The difference between CBR and rt-VBR is that rt-VBR connections are characterized in terms of a sustainable cell rate and maximum burst size, in addition to the PCR. Thus, the source is expected to transmit at a variable rate. nrt-VBR connections are also characterized in terms of PCR, SCR and MBS. nrt-VBR, however, is intended for non-real-time

Table 1: ATM service categories

	ATM Service Category					
Attribute	CBR	rt-VBR	nrt-VBR	UBR	ABR	GFR
Traffic Parameters						
PCR and CDVT	specified					
SCR and CDVT	n/a	specified		n/a		only CDVT for MCR
MBS	n/a	specified		n/a		specified for MCR
MFS	unspecified					specified
MCR	n/a				specified	
QoS Parameters						
peak-to-peak CDV	specified		unspecified			
maxCTD	specified		unspecified			
CLR	specified			unspecified	low for conforming	
Other Attributes						
Feedback	unspecified				specified	unspecified

n/a = not applicable

bursty traffic with no delay or delay variation bounds, but with a low cell loss ratio requirement (for conforming cells).

The UBR service is the simplest service. It is intended for traditional data traffic, such as file transfer and electronic mail. No delay or loss guarantees are provided; the service is a best effort service. No fairness or isolation of connections can be assumed. Like UBR, the ABR service is intended for data applications with no delay guarantees. ABR, however, attempts to minimize the cell loss ratio, and give minimum cell rate guarantees through a flow control mechanism. The network provides feedback to the sources when network load changes, and the sources adjust their transmission rates accordingly. ABR sources share the available bandwidth *fairly*, and the source is never required to send below its specified MCR.

As with UBR and ABR, the GFR service is intended to support non-real-time applications. The service is particularly targeted at users who are not able to specify all the traffic parameters needed to request services such as VBR, and are not equipped to comply with the end system behavior required by ABR. Although such users can currently request UBR connections, UBR provides no service guarantees. GFR guarantees a minimum rate and low cell loss ratio for conforming frames, while requiring little interaction between users and the network. The key attractive feature of GFR is its frame level visibility and guarantees, resulting in useful data being delivered.

3 Mechanisms for Providing Guarantees

This section gives more details on the mechanisms the end systems and network elements use to provide the guarantees for each service category.

3.1 CBR, rt-VBR and nrt-VBR

The three categories: CBR, rt-VBR and nrt-VBR provide open loop traffic control and preventive congestion avoidance. During connection admission control (CAC), reservations are made in the network nodes to meet the traffic contract and QoS commitments. The traffic contract can be met by the source end system if appropriate traffic shaping is performed. Alternatively, the network or the destination end system may enforce the contract using usage parameter control (UPC) functions. Traffic shaping and UPC can be performed using the generic cell rate algorithm (GCRA) which essentially uses leaky bucket mechanisms. For each cell arrival, GCRA determines whether the cell conforms to the traffic contract of the connection or not. Non-conforming cells may be tagged/marked (their cell loss priority (CLP) bit is set to one) or dropped. All tagged cells are dropped before any untagged cell is dropped (i.e., untagged cells have a higher priority). Two GCRA leaky buckets are needed to shape or control traffic according to the VBR parameters. CBR, rt-VBR and nrt-VBR provide complete isolation between connections: connections exceeding their traffic contract should not affect the QoS experienced by the other connections [7].

3.2 UBR

The basic UBR service has no explicit congestion control mechanisms. UBR signaling and parameters are minimal: only PCR is specified, and even that may not be subject to CAC or UPC procedures. Switches respond to congestion by dropping UBR cells when their buffers become full. Intelligent switch drop policies and end system policies can improve the performance of UBR with limited buffers. Intelligent packet discard mechanisms are also applicable to all services which use ATM adaptation layer 5 (AAL5), such as VBR-nrt and ABR. Partial packet discard (PPD) and early packet discard (EPD) [19] have been shown to improve throughput. Per-VC accounting drop methods, and per-VC queuing and scheduling have been shown to improve both throughput and *fairness* [10]. A service using these mechanisms is usually referred to as UBR+. Throughout the remainder of this paper, we mean simple vanilla UBR implemented in first generation switches when referring to UBR.

3.3 ABR

ABR allows the network to divide the available bandwidth fairly and efficiently among active sources. The ABR traffic management model is: (1) "rate-based" because the sources transmit at a specified "rate," rater than using a window; (2) "closed-loop" because, unlike CBR and VBR, there is continuous feedback of control information to the source throughout the connection lifetime; and (3) "end-to-end" because control cells travel from the source to the destination and back to the source [14].

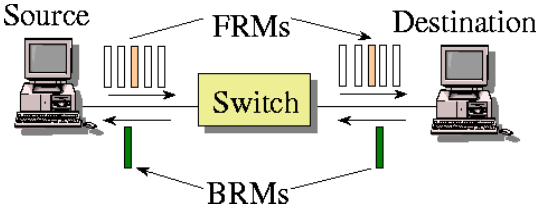


Figure 2: Forward and backward RM cells carry feedback information to the sources.

The components of the ABR traffic management framework are shown in figure 2. To obtain network feedback, the sources send resource management (RM) cells after every $Nrm - 1$ (Nrm is a parameter with default value 32) data cells. Destinations simply return these RM cells back to the sources. The RM cells contain the source current cell rate (CCR), in addition to several fields that can be used by the network to provide feedback to the sources. These fields are: the explicit rate (ER), the congestion indication (CI) flag and the no increase (NI) flag. The ER field indicates the rate that the network can support for this connection at that particular instant. The ER field is initialized at the source to a rate no greater than the PCR, and the CI and NI flags are usually reset. Each switch on the path reduces the ER field to the maximum rate it can support, and sets CI or NI if necessary. The RM cells flowing from the source to the destination are called forward RM cells (FRMs) while those returning from the destination to the source are called backward RM cells (BRMs) (refer to figure 2). When a source receives a BRM cell, it computes its allowed cell rate (ACR) using its current ACR value, the CI and NI flags, and the ER field of the RM cell [2, 3, 5, 7, 13, 14].

3.4 GFR

The GFR service requires user data to be divided into frames that can be delineated at the ATM layer. If the user sends frames not exceeding the maximum frame size (MFS) in a burst that does not exceed the maximum burst size (MBS), the user can expect its frames to be delivered with minimum losses. GFR also allows the user to send in excess of the MCR (and the associated MBS), and delivers excess traffic if resources are available. Such resources should be shared "fairly" among users [18].

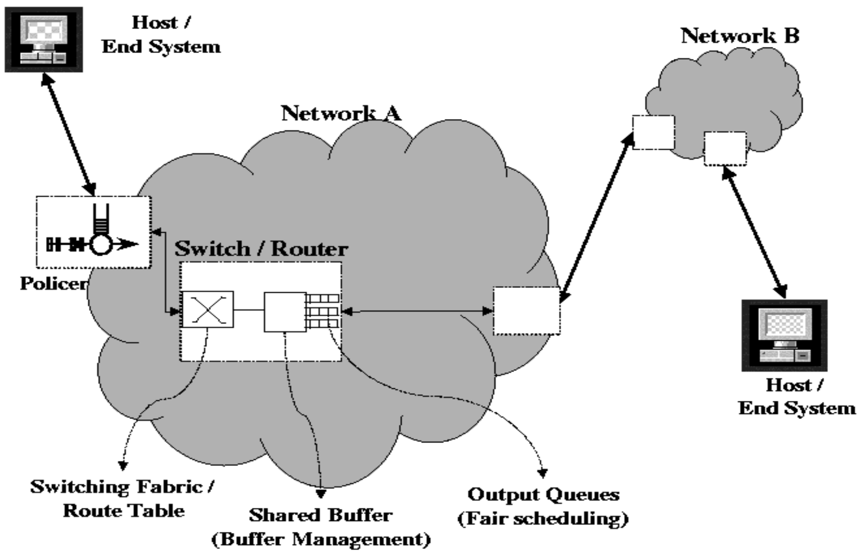


Figure 3: A network can use tagging, buffer management and scheduling to meet guarantees.

There are three design options that can be used by the *network* to provide the per-VC guarantees for GFR [9] (refer to figure 3): **(1) Tagging (Marking):** Network based tagging (or policing) can be used as a means of marking non-conforming frames. This requires some per-VC state information to be maintained by the network. Tagging can isolate the non-conforming traffic of each VC so that other rate enforcing mechanisms can schedule the conforming traffic in preference to non-conforming traffic. Policing can be used to discard non-conforming packets. **(2) Buffer management:** If multiple VCs share a 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 buffer occupancies [10]. **(3) Scheduling:** Scheduling and queuing strategies determine how packets are scheduled onto the next hop. First-in first-out (FIFO) queuing cannot isolate packets from various VCs. Per-VC queuing, on the other hand, maintains a separate queue for each VC in the buffer and can isolate VCs.

4 Proposed Architecture for Connecting Enterprise Networks

We propose an architecture that employs intelligent edge devices and an ATM backbone to connect enterprise networks, as shown in figure 4. Our architecture integrates real-time and data traffic of the enterprise on a single backbone virtual path connection (VPC) between sites. The architecture supports IP differentiated and integrated services traffic and policy control, in addition to ATM and frame relay (FR) traffic. The advantages of separating edge device functionality from backbone functionality include simplification and scalability of the network design and bandwidth management, as well as scalability of the number of connections [20]. Enterprise *voice, video and data* integration within a *single* carrier VPC decreases the costs the enterprise pays (one VPC is used instead of two or more between any two points), and also allows dynamic sharing of voice, video and data bandwidth.

The proposed network is thus a two-tiered network: the outer (access) tier and the inner (backbone) tier. The access tier performs flow identification and QoS management at the flow level. Each switching node manages a relatively small number of flows. It may use ATM, FR, integrated services, or differentiated services for quality of service, or classes of service (COS). Traffic is aggregated at the edge into an ATM backbone (forming the inner tier). The backbone works with aggregate flows, mapped to ATM VPCs or VCCs. The backbone traffic management is simple because of the large number of flows within each connection, and the high speed between the nodes. Backbone traffic management is at the granularity of aggregates, not for traffic within a flow.

This architecture can be used for VPNs, large local area network (LAN) or wide area network (WAN) enterprises, and carrier networks. We will focus on the VPN application. Typically, each enterprise site has a relative abundance of bandwidth (for example, using Fast or Gigabit Ethernet). The site implements the enterprise policy for managing the traffic. It performs flow identification and classification, QoS assignment, QoS management, and flow mapping within the local area network (the campus or the branch). QoS can be managed through: (1) tagging/marking, (2)

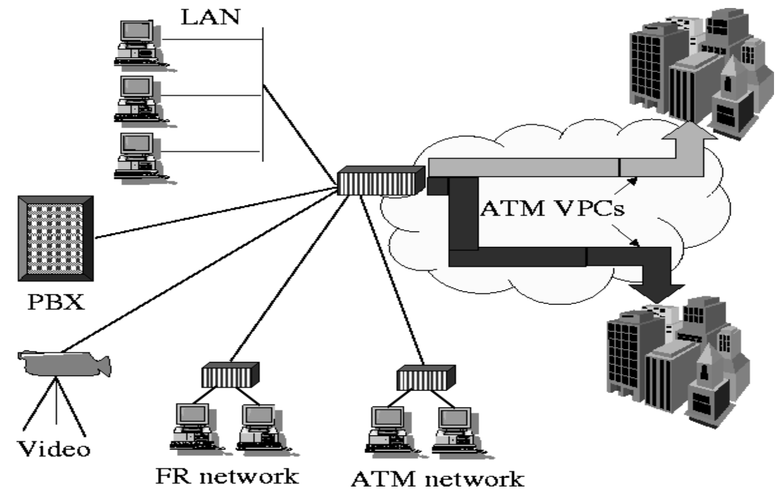


Figure 4: The proposed architecture uses a single VPC to connect enterprise sites. Voice, video and data traffic can be multiplexed on this VPC.

dropping, or (3) assigning scheduling priorities. At the edge of the campus enterprise network, traffic is aggregated into the ATM VPCs or VCCs for transport through the carrier network connecting the sites. The edge device uses a weighted fair queuing scheduler for scheduling traffic to the VPC(s), as shown in figure 5. The following subsections give more details on the design of the edge device and the choice of ATM service to use in the backbone.

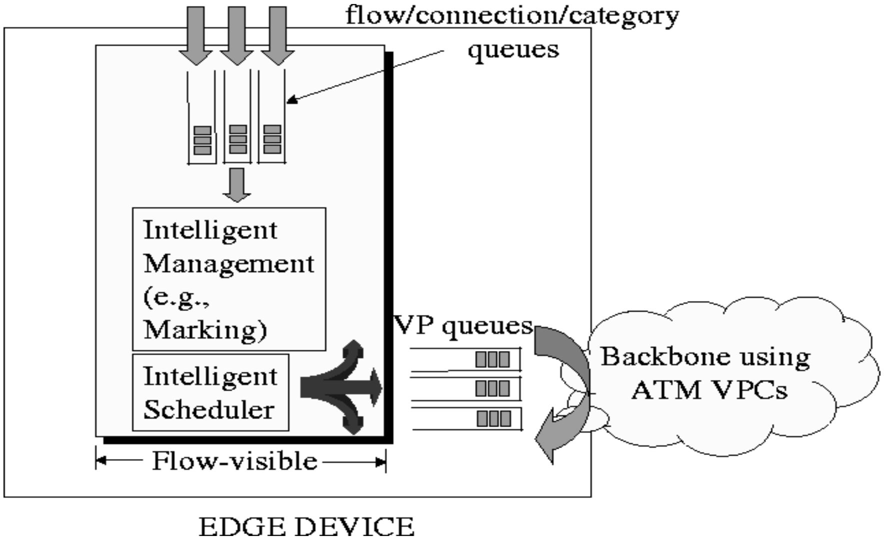


Figure 5: The edge device performs traffic management based on the flows, and then intelligently schedules traffic to the backbone VPCs.

4.1 The Scheduler

An intelligent scheduling mechanism is required in the edge device to feed traffic from multiple flows into the ATM VPC(s). An example of a scheduler is a weighted fair queuing (WFQ) scheduler for the individual connections or flows into the VPC pipe. Per-connection or per-flow queues may be maintained to control delay and loss, depending on the flow type.

The weights used by the WFQ scheduler for different traffic streams are assigned based upon:

- The enterprise policy rules for users or applications.
- The ATM (or FR) parameters negotiated during connection admission, and the ATM service category, in case of ATM or FR networks at an enterprise site.
- The integrated services requests signaled by the application (if integrated services and the reservation protocol (RSVP) are used at the enterprise site).
- The service requested by the hosts and set in the packet headers using the differentiated services framework (refer to section 4.4 for more details).

4.2 Choice of Service Category in the Backbone

The choice of service category to use in the ATM backbone is critical to the quality of service experienced by applications sending traffic to another site of the enterprise. As previously mentioned, each site is likely to have abundant bandwidth. Congestion most likely occurs on the relatively low-capacity WAN access link (for example, a Fast or Gigabit Ethernet feeding into a low capacity T1/E1 or T3/E3 link). Depending on the carrier ATM service category, congestion may occur in the carrier network leading to performance degradation. For example, if VBR is used and the traffic is aggressively shaped to the PCR and not the SCR, losses in the backbone can occur.

We will show that ABR performs well in the backbones connecting enterprise networks. The ABR service pushes congestion to the edge devices, where adequate buffering can be provided, and, more importantly, the flows are visible and the enterprise policy can be applied. The ABR VPCs perform flow control for the pipes between enterprise networks. With ABR, there is very little loss in the backbone, and hence higher priority traffic can be transported without loss. On the other hand, the application takes advantage of all the bandwidth given by the network and efficiently utilizes the buffer at the edge device. This is not the case with other services, such as VBR, where either (1) the traffic is shaped according to the SCR to avoid loss in the network, which is clearly inefficient and increases delay, or (2) the traffic is shaped according to the PCR, which risks random losses inside the backbone, unless intelligent cell marking according to SCR is used.

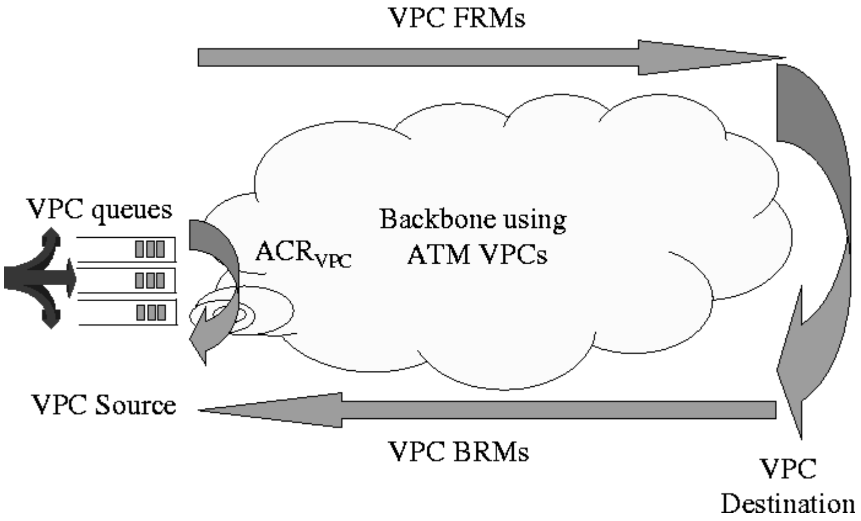


Figure 6: ABR VPCs can be used in the network backbones to minimize delay and loss.

Figure 6 shows the use of ABR VPCs. Per-VP queues are implemented at the VPC source to control the rate of the ABR VPC to the VP allowed cell rate (ACR), according to the feedback from the VPC BRM cells. (In the case of an ABR VCC multiplexed on the ABR VPC, per-VP accounting information and the VPC ACR are used to compute the rate indicated in the VCC BRM cells.) We will now discuss how the feedback indicated in the VPC RM cells is computed.

4.3 The ABR VPC Rate Allocation Scheme

As previously mentioned, enterprise real-time and non-real-time traffic can be mixed on an ABR VPC. ABR, however, provides no delay guarantees by the service provider. But the use of minimum cell rate (MCR) guarantees, a good explicit rate (ER) switch algorithm, small switch buffers (thus controlling queuing delays), and an intelligent edge scheduler (as explained in the previous subsection) can give delay and loss guarantees [22]. We emphasize the need for a good ER algorithm to reduce loss.

An example of a good ER algorithm is the ERICA+[2] algorithm [15]. ERICA+ gives MCR guarantees, which can provide a minimum acceptable quality of service, even for voice and video applications. The use of weights is allowed to give a generalized form of the fair allocations. Thus, the bandwidth in excess of the MCRs is divided proportional to a predetermined weight, associated with each ABR connection.

Switch queue size, and thus queuing delay, can be controlled using an appropriate "queue control function" to scale the available bandwidth estimate. ERICA+ uses a queue control function $f(Q)$ as follows:

$$\text{available bandwidth} = f(Q) \times \text{ABR bandwidth}$$

The value of $f(Q)$ depends on the current switch queuing delay. Figure 7 shows an example of a queue control function. A target queuing delay, Q_0, is specified, and the function is an inverse hyperbolic function for queuing delays larger than the specified value ($Q > Q_0$). The function, however, does not decrease beyond a minimum value, called the queuing delay limit factor (F_{min} in the figure). For more details on the performance of different queue control functions, refer to [23].

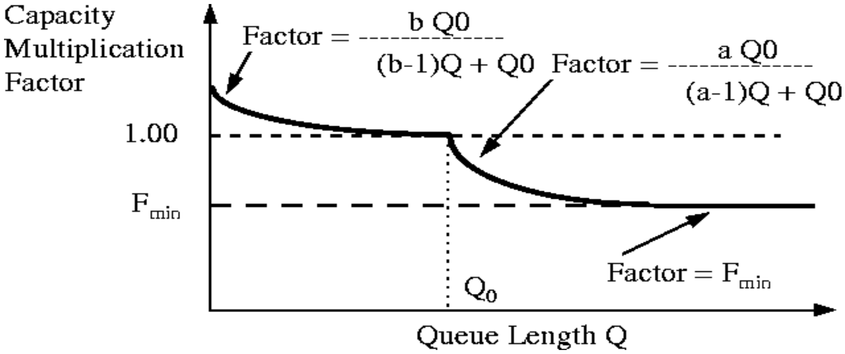


Figure 7: The inverse hyperbolic function can be used for queue control.

Figure 8 gives a simplified flow chart of the ERICA+ algorithm (for the complete algorithm, refer to [15]). Time is divided into successive intervals, and the algorithm performs a number of computations at the end of every interval. These computations include estimation of the load on the network, estimation of the ABR capacity, and averaging out the values across successive intervals to smooth out measurement variations.

When a BRM cell is received, the algorithm needs to compute the rate to allocate to this connection.

[2] Explicit rate indication for congestion avoidance

It first computes an overload factor as shown in the first step in figure 8. The overload factor is the ratio of (A) the average total "excess" load (i.e., after subtracting the MCR values), to (B) the average excess ABR capacity (also after subtracting MCR values), scaled by the queue control function as explained above.

In the next step, the overload is compared to 1+δ (usually δ is set to 0.1). If the overload is greater than 1.1, which means there is high overload, the algorithm scales down the current cell rate of the connection (in excess of MCR) by the overload factor, and then adds the MCR (refer to the rectangle on the right in figure 8). This brings down the load. Otherwise, if there is underload (overload is $\leq 1+\delta$), the algorithm also uses an additional quantity. This quantity is the weighted (according to user specified weights) excess (over MCR) maximum allocation allocated during the previous interval (also averaged). This quantity is the second parameter to the "max" operation in figure 8. Bringing up all allocations to this quantity ensures that all connections get fair rates according to the specified weights. Thus the algorithm guarantees MCRs, controlled queuing delays, and weighted allocations. The algorithm is described and analyzed in [21].

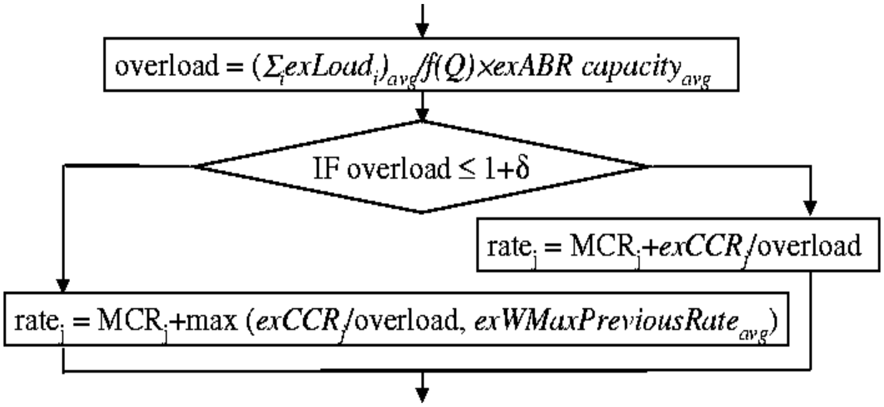

overload = $(\Sigma_i exLoad_i)_{avg} / f(Q) \times exABR\ capacity_{avg}$
IF overload ≤ 1+δ
rate$_j$ = MCR$_j$+$exCCR_j$/overload
rate$_j$ = MCR$_j$+max ($exCCR_j$/overload, $exWMaxPreviousRate_{avg}$)

Figure 8: The rate allocation algorithm provides weighted fairness with MCR support and controls network queues.

Some switches also implement the use-it-or-lose-it feature. The use-it-or-lose-it concept essentially reduces the rates allocated to any connection that is sending data at a much lower rate. Figure 9 demonstrates the problem that can arise when sources are allocated high rates without using them. In the figure, before time t_0 the source rate is much smaller than its ACR allocation. The ACR allocation remains constant. At time t_0, the source rate rises to ACR and the network queues correspondingly grow. Reducing the rates of such connections (usually referred to as "ACR retaining" or "ACR promoting" connections) reduces the potential cell loss if such connections all suddenly start transmitting at their full rate. More details on use-it-or-lose-it policies are given in [17].

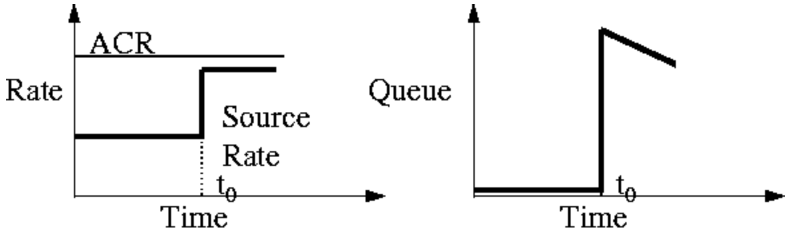


Figure 9: ACR retention and promotion can cause sudden network overload when sources use their full ACRs.

4.3.1 Sample Simulation Results

In this section, we give a sample simulation result using the ABR rate allocation algorithm discussed above. We use persistent sources (always sending at ACR) in the simulation. The data traffic is only one way, from source to destination. Using two-way traffic produces similar results, except that the convergence time is larger since the BRM cells travel with traffic from the destination to the source. The network configuration simulated has three sources sending data to three destinations over two switches and a bottleneck link, as shown in figure 10 ($n = 3$). All the link bandwidths are 149.76 Mbps (accounting for SONET overhead). The link distances were 1000 km each link.

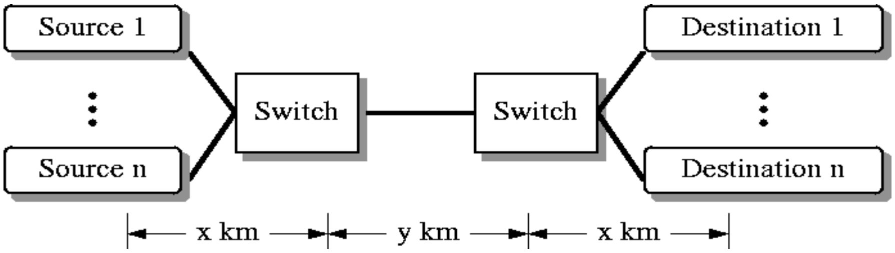
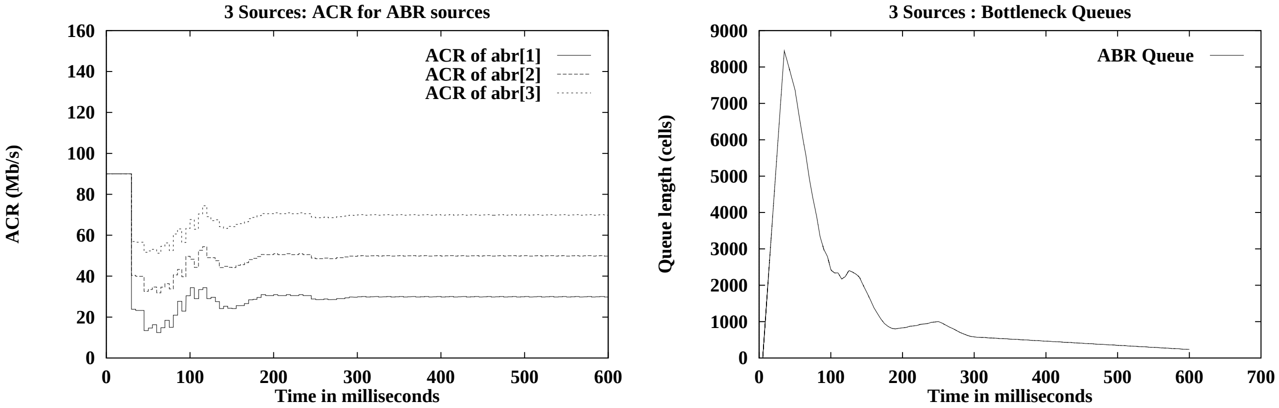


Figure 10: The n source configuration is a simple configuration with n sources sending to n destinations using the same bottleneck link.

We used the ERICA+ [15] algorithm (as explained above) implemented in an ATM network simulator. An ERICA+ interval length of 5 ms was used. As previously mentioned, a dynamic queue control function achieves a constant queuing delay in steady state. The "target delay" parameter (mapped to Q_0 used in figure 7) specifies the desired queuing delay. A value of 1.5 ms was used. The hyperbolic function curve parameters used were $a = 1.15$ and $b = 1$. The F_{min} value was set



(a) Rates of the 3 sources

(b) Queue length at the bottleneck

Figure 11: Results for the three source configuration with ABR show that weighted fairness with MCR is achieved and queues are bounded.

to 0.5. A δ value of 0.1 was used (refer to figure 8). Exponential averaging was used to decrease the variation in measured quantities such as the input rate and the available capacity. The exponential averaging parameter used was 0.8.

Weight values of one were used for all connections in this simulation. This corresponds to an allocation of MCR plus an equal share of excess bandwidth for each connection. If weights equal to MCRs are used, the remaining bandwidth will be shared in proportion to the MCR values. The MCR values used in the simulation were 10, 30 and 50 Mbps for sources 1, 2 and 3 respectively. The excess bandwidth ($149.76 - 90 =$) 59.76 Mbps is divided equally among the three sources. Therefore, the expected allocation vector is $\{10 + 59.76/3, 30 + 59.76/3, 50 + 59.76/3\} = \{29.92, 49.92, 69.92\}$.

Figure 11 shows the allowed cell rate values for the three sources. From figure 11(a), it can be seen that the expected allocation is achieved for the three sources. Each source is given its MCR plus an equal share of the remaining bandwidth. The rates converge after a short transient period. Figure 11(b) shows that the dynamic queue control function rapidly controls the queuing delay to the specified target. The initially large queues are the result of the large initial cell rate values used for the three sources. We used such large values to show that the algorithm rapidly adapts to the transient overload.

4.4 Internet Differentiated Services Support

Internet differentiated services enable the deployment of multiple services in large networks, providing an alternative to per-flow processing and per-flow state [1]. The use of the differentiated services model is envisioned in large core networks, and the use of integrated services with the resource reservation protocol (RSVP) [4] is foreseen to be in peripheral stub networks (for example, campus enterprise networks), as shown in figure 12.

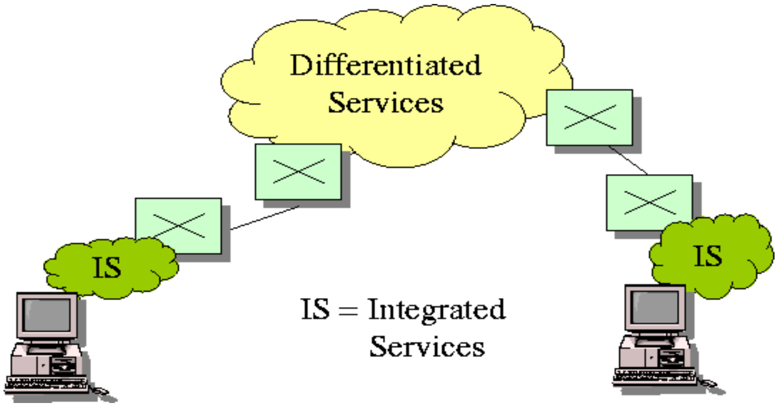


Figure 12: Differentiated services are used in Internet backbones, while integrated services are used in peripheral networks.

In large carrier networks, differentiated services IP traffic may be transported over ATM backbones. The mapping of differentiated services to ATM is not straightforward. For example, the Internet assured forwarding behavior provides multiple drop preferences and multiple classes, while ATM has only two drop preferences through the cell loss priority bit, and an unspecified number of queues for each service category.

Consider mapping the assured forwarding behavior onto an ATM backbone using ABR connections. In this case, edge routers can use different drop thresholds for different assured forwarding drop preferences, since most queues are at the edge router itself. Flows are visible at these edge routers, but not inside the ATM network (refer to figure 5). CBR may also be used in the backbones, but CBR is unsuitable for bursty traffic. The remaining service categories (rt-VBR, nrt-VBR, UBR and GFR) cannot easily handle more than two drop preferences since the queues may grow *inside* the network, and it is difficult to control discard priorities at that point (flows are not visible).

Having multiple queues with different priorities and weights (guarantees) may also complicate mapping differentiated services onto ATM. ABR and CBR can implement multiple priority queues by maintaining the queues at the edge routers and multiplexing all the queues onto connections. Again, since flows are visible at the edge routers, intelligent scheduling can be performed there, as discussed in section 4.1. This is adequate for giving different guarantees, because ABR (or CBR) queues inside the ATM network are very small, and hence the QoS inside the network is unaffected. The other service categories cannot guarantee bandwidth because priorities must be enforced inside the ATM network, since this is where longer queues exist. Enforcing priorities at the edge router is inadequate, and enforcing priorities inside the network cannot be performed through setting GFR MCR, or VBR SCR and/or PCR.

The remainder of the paper will give more details on the relative cost and performance of the ATM service categories, and will show that a well-designed ABR service provides good performance for TCP and UDP.

5 Cost/Performance Tradeoffs among Service Categories

Tables 2 and 3 compare the six ATM service categories in terms of the complexity of connection admission and connection aggregation, the cost of end systems (network interface cards or NICs) and network elements (switches), and the buffer requirements and guarantees provided. As seen in the tables, UBR is the simplest service in terms of signaling, as it has a single parameter, PCR. Furthermore, end systems and network elements are simple as they are not required to perform any functions, though they may police on PCR, and UBR+ may provide intelligent drop policies (thus needing to recognize frame boundaries). One simple first-in first-out (FIFO) queue is adequate for vanilla UBR. UBR, however, gives no guarantees and requires high switch buffering.

CBR and rt-VBR give the most strict guarantees, but they require the user to specify a number of parameters to exactly define the traffic contract and requirements. CBR requires the specification of the PCR and CDVT, in addition to the required QoS parameters, and it gives strict delay, delay variations and loss guarantees. The end systems and network elements are quite simple, since they only need to perform the functions required at connection admission control (provisioning), and perform policing functions. Very little buffering is needed at the switches. The problem with CBR, however, is that it is wastes bandwidth if traffic is bursty, and most data traffic is bursty. CBR is bandwidth-inefficient because during connection admission, the network elements assume that the connection will always be sending at the peak rate, and resources must be reserved to satisfy the QoS requirements under such conditions. There is minimal statistical multiplexing gain.

rt-VBR gives the same strict guarantees as CBR, and does not suffer from the bandwidth inefficiency problem. However, applications are required to specify their traffic precisely- in terms of a peak cell rate, a sustained cell rate, maximum burst size and tolerance values. But applications rarely know their traffic characteristics precisely. Connection admission decisions, billing, and aggregation of VBR connections are also difficult. This is because the accumulation of values, such as the cell delay variation, is not straightforward (not additive, for example). End systems and network elements must be slightly more complex than with CBR because policing is based on the sustained cell rate as well. nrt-VBR also gives loss and rate guarantees (though no delay guarantees) and provides isolation, but it also requires the user to define a specific traffic contract. Though QoS parameters and connection admission control decisions are simpler than with rt-VBR, they are non-trivial. Large buffers are required in network elements for nrt-VBR to reduce cell loss.

ABR minimizes cell loss for well-behaved connections and can give minimum rate guarantees, in addition to isolation and fairness, but the end system and switches need to perform complex functions. A large number of parameters are signaled, though the setting of these parameters is well understood. The connection admission is simply based on MCR, but it is slightly more complex than CBR because an overbooking factor must be determined. ABR does provide superior traffic management though, since it maximizes buffer and bandwidth utilization (because the end systems precisely know the network state). Switch buffer requirements are small, and extra buffering at the end systems or routers can be utilized. No other service category provides end systems with network state information. Little buffering is required at routers or end systems if TCP acknowledgment regulation schemes are used to convey the ABR rate information to the TCP flow control mechanism.

GFR gives similar guarantees to ABR, but reduces the signaling and end system complexity. A minimum cell rate value is negotiated and the user may request tagging. End systems only need to provide tagging and policing functions. Network elements, however, provide the required minimum rate guarantees and fairness among connections. In addition to tagging, intelligent buffer allocation

Table 2: Cost/Complexity of ATM service categories

	Complexity of				**Overall**
Category	Connection Admission	Connection Aggregation	End System	Network Element	**Complexity**
CBR	Low+	Low	Low+	Low	Low
rt-VBR	High	High	Medium	Medium	Medium
nrt-VBR	High	High	Medium	Medium	Medium
UBR	Low	Low+	Low	Low	Low
ABR	Medium	Medium	High	High	High
GFR	Medium	Medium (frame size)	Medium+	Medium+	Medium+

and scheduling may need to be performed. Buffer requirements at the switches may be high.

6 Performance of TCP over ATM

In this section, we compare the performance of TCP over different ATM service categories. The bursty nature of TCP traffic makes transporting it over CBR a poor design choice, as TCP traffic rarely requires delay guarantees, and the statistical multiplexing benefits of ATM are not fully utilized. Before we explore the transport of TCP over ATM, we will first discuss the TCP congestion avoidance mechanism and how it affects TCP traffic patterns as seen at the ATM layer.

6.1 TCP Congestion Avoidance

TCP is the most popular transport protocol. It provides reliable transfer of data using a window-based flow and error control algorithm. The key TCP congestion control mechanism is the TCP slow start [12]. TCP connections use a window to limit the number of packets that the source can send. The sender window is computed as the minimum of the receiver window (Wrcvr) and a congestion window variable (CWND).

Whenever a TCP connection loses a packet, the source does not receive an acknowledgment (ack) and it times out. The source remembers the congestion window (CWND) value at which it lost the packet by setting a threshold variable, SSTHRESH, at half the window size, and then CWND is set to one. The source retransmits the lost packet and increases its congestion window by one

Table 3: Performance of ATM service categories

	Buffer Requirements		**Guarantees**			
Category	Network Element	End System (or Router)	Fairness and Isolation	Delay	Loss	Rate
CBR	Very Low	Depends on traffic	Yes, but no full multiplexing	Yes	Yes	Yes
rt-VBR	Medium	Low	Yes	Yes	Yes	Yes
nrt-VBR	High	Low	Yes	No	Yes	Yes
UBR	High	Low	No	No	No	No
ABR	Low	High (except when acknowledgment regulation for TCP is used)	Yes	No	Yes	Yes
GFR	High	Low	Yes	No	Yes (frame-based)	Yes

every time a packet is acknowledged. This continues until the window reaches SSTHRESH. After that, the window w is increased by $1/w$ for every packet that is acked. The source window is always limited by the receiver window size. The typical changes in the source window plotted against time are illustrated by figure 13.

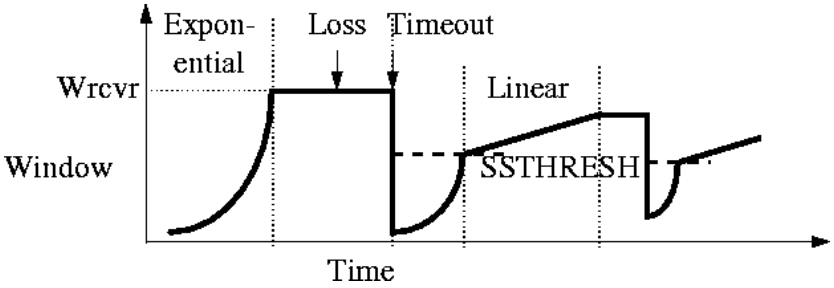


Figure 13: The TCP slow start and congestion avoidance mechanism manages traffic by controlling the growth of the TCP window.

6.2 TCP over UBR

The UBR service depends upon the transport layer to provide congestion and flow control functions. A single cell drop at the ATM layer results in an entire packet drop at the destination. TCP at the source times out and retransmits the lost packet. Low throughput and unfairness result from the time lost in the timeouts and retransmissions of packets. A TCP source stops increasing its transmission rate only when its congestion window reaches a maximum value. *TCP using basic vanilla UBR requires network buffers approaching the sum of the maximum window sizes of all TCP connections to completely avoid cell loss.* Thus vanilla UBR does not scale well in this sense. With limited buffering, however, TCP throughput and fairness can be improved using UBR+ [16] by: **(1) Drop policies** decide when to drop cells. PPD and EPD [19] drop full packets instead of random cells from multiple packets. **(2) Buffer allocation policies** decide how to divide the available buffer space among the cells from contending connections. Fair buffer allocation (FBA) schemes [10] improve fairness by selectively discarding frames from flows that are sending more than their fair share. **(3) Scheduling policies** divide the available bandwidth among contending queues. Scheduling may be implemented at a coarse granularity to divide bandwidth among service categories, or at a fine granularity to divide bandwidth among connections within a service category.

6.3 TCP over ABR

For TCP over ABR, the TCP window-based control is running on top of the ABR rate-based control. A steady flow of RM cells results in a steady flow of feedback from the network. In this state, the ABR control loop has been established, and source rates are primarily controlled by the network feedback (closed-loop control). When the source transmits data after an idle period, there is no reliable feedback from the network. For one round trip time (time taken by a cell to travel from the source to the destination and back), the source rates are primarily controlled by the ABR source end system rules (open-loop control). When the traffic is bursty, open-loop control may be exercised at the beginning of every active period.

ABR switch algorithms allocate high rates to ABR sources if insufficient load is experienced at the switches. This is likely to be the case when a new TCP connection starts data transmission. The connection is bottlenecked by the TCP congestion window size and not by the ABR source rate. The TCP active periods double every round trip time and eventually load the switches and appear as persistent traffic at the ATM layer. The switches ask sources to reduce their rates, and data is bottlenecked by the ABR source rate, not by the TCP congestion window size. Once the ABR rates converge to optimal values, the lengths of the ABR queues at the switches decrease.

Therefore, ABR flow control pushes the queues from the network to the end systems. In [16], we show that ABR is scalable for persistent applications running over TCP/IP (such as long file transfers) in the sense that, given the right implementation and parameters, its network buffer requirements for *zero* packet loss do *not* grow linearly with the number of TCP connections. If buffers overflow, smaller TCP timer granularity (which controls timeout durations) can help improve throughput.

6.3.1 Sample Simulation Results

As a sample result, we show the throughput and maximum queue length obtained for a simple 15 source configuration, as shown in figure 10 (with $n = 15$). The configuration has a single bottleneck link shared by 15 ABR sources. All links run at 155.52 Mbps and are of the same length. We experiment with various link lengths.

All traffic is unidirectional. A large (infinite) file transfer application runs on top of TCP. The link lengths are 1000 km×3 links, 500×3, 200×3, 50×3 (for round trip times (RTTs) of 30, 15, 6 and 1.5 ms respectively). We have verified that maximum queue bounds also apply to configurations with heterogeneous link lengths, multiple bottlenecks and with VBR traffic in the background causing variance in ABR capacity and errors in measurement. We use a TCP maximum segment size (MSS) of 512 bytes. The window scaling option is used so that the throughput is not limited by path length. The TCP window is set to 16×64 kB = 1024 kB. We define TCP throughput as the number of bytes delivered to the destination application in the total time. This is sometimes referred to as goodput by other authors.

Table 4 shows that the worst case maximum queue is less than $3 \times \mathrm{RTT} \times$ link bandwidth, even with transient bursts. Therefore, ABR is scalable because the maximum queue size is a small multiple of the RTT, and does not grow linearly with the number of connections. ABR buffer requirements are also much smaller (than the values shown) after the system reaches steady state.

6.4 TCP over GFR

Edge devices can use GFR to transport multiple TCP connections over a single GFR connection. The bursty nature of TCP traffic makes it difficult to provide per-connection GFR rate guarantees using FIFO queuing. *Per-VC queuing* and scheduling are recommended to provide rate guarantees

Table 4: Maximum queue size and throughput for TCP over ABR (transient bursts)

Number of Sources	RTT (ms)	Feedback Delay (ms)	Max Queue Size (cells)	Throughput
15	30	10	15073 = 1.36×RTT	107.13
15	15	5	12008 = 2.18×RTT	108.00
15	6	2	6223 = 2.82×RTT	109.99
15	1.5	0.5	1596 = 2.89×RTT	110.56

to TCP connections when GFR VCCs are fully using the buffers. Good TCP performance has been observed in such cases. Under conditions of low buffer allocation, however, it is possible to control TCP rates, even with FIFO queuing, by manipulating the TCP congestion window through setting buffer thresholds to drop packets. This assumes that in cases where the offered load is low, a queue is not built up and TCP is allowed to use as much capacity as it can. The average throughput achieved by a connection is proportional to the fraction of the buffer occupancy used by the cells of that connection. As long as the fraction of buffer occupancy of TCP can be controlled, its relative throughput depends primarily on the fraction of packets of that TCP in the buffer. At a very high buffer utilization, packets may be dropped due to buffer unavailability. This results in larger variations in TCP throughputs. At very high thresholds, the queuing delay also increases significantly, and may cause the TCP sources to time out [9].

6.5 TCP over VBR

TCP can be transported over the VBR service, given that the user selects an appropriate PCR, SCR, MBS and CDVT values. In [11], experiments are performed to evaluate the performance of TCP over VBR. TCP over VBR performed well except with 100% utilization and when MBS was set to extremely small values. The reason behind the poor performance of VBR for unreasonably small MBS is that a large number of the frames are partially tagged. This causes the corruption of frames, although they consumed tokens from the GCRA (leaky bucket). If the sum of reservations is only 50% of the link rate, VBR showed good fairness than in the full utilization case, even with small MBS. The reason for this is that if the bucket size is large, frame boundaries make no difference: in a large burst, only the one frame will be partially tagged [11]. As a rule of thumb, MBS must be greater than the maximum frame size (preferably greater than the maximum TCP window of 64 kbytes plus overhead). Most carriers set MBS in the 9 kbytes to 64 kbytes range.

6.6 Comparison of TCP Performance over UBR, ABR, GFR and VBR

Table 5 summarizes the previous discussion on the performance of TCP over the four service categories: VBR, UBR, ABR and GFR. Vanilla UBR performs poorly unless buffers are large or intelligent drop policies (UBR+) are employed. ABR pushes the queues to the ATM network edges and, in that sense, it is scalable because the network queue sizes are not a function of the number of connections, but of the round trip times, feedback delays and switch congestion avoidance scheme employed. GFR exhibits good efficiency and fairness with intelligent drop or tagging, and/or per connection queuing. VBR performs quite well except in cases where utilization is high and MBS is too small, because incomplete frames are tagged.

Since TCP losses result in long idle times waiting for a timeout and performing slow start, high utilization is directly linked to low packet loss. ABR provides control over queue length, and hence the low loss and high bandwidth utilization. In end-to-end ATM, ABR and CBR minimize losses. GFR and UBR can use fair buffer allocation or per-VC queuing to fairly distribute losses among the VCCs, since each TCP flow will most likely be carried on a separate VC. This is not the case for an ATM backbone situation, where each VC will carry multiple TCP flows and VCCs are only used between edge routers. In this case, most ABR and CBR losses are in the routers and not the ATM switches, so schemes such as Random Early Detection (RED) gateways [6] (or flow RED gateways) that perform selective drop at the routers can provide fairness for TCP over ABR. With VBR, UBR and GFR, most losses occur in the ATM switches and not at the edge routers. Fair buffer allocation in the ATM switches can ensure fairness among the VCCs, but not among the flows multiplexed on the same VCC. Hence, our results show that ABR is most suitable for bursty TCP traffic, followed by GFR, then VBR, then UBR and finally CBR.

7 Performance of UDP over ATM

The User Datagram Protocol (UDP) has no built-in flow control mechanism like the TCP slow start and congestion avoidance mechanisms. Therefore, losses may continue and have more effect than in the case of TCP. Several client-server transaction applications use UDP. An example of such servers is authentication servers used for security. Such applications handle retransmission

Table 5: Performance of TCP over ATM service categories

Category	Performance of TCP
UBR	Low efficiency and fairness, especially without intelligent drop and with FIFO queuing and scheduling, unless buffer approaches sum of receiver windows.
GFR	Very good performance with intelligent drop and tagging (better than VBR because of frame visibility).
VBR	Good (except with high utilization and very small MBS values).
CBR	Unsuitable for bursty TCP traffic.
ABR	Pushes queues to ATM network edges and provides high utilization and fairness. Network queues only depend on round trip time, feedback delay and switch scheme.

if necessary. In addition to loss-sensitive data traffic, UDP is also used to transport loss-tolerant traffic, such as voice over IP. Loss-tolerant applications are usually delay-sensitive applications, for example, voice applications have no use for the packets after a certain time delay, and thus can tolerate its a moderate amount of loss.

As with TCP, CBR is not ideally suited to UDP traffic which is generally bursty. VBR, UBR and GFR can make use of the drop priority to drop lower priority packets before dropping higher priority ones. However, since these categories have no information on the network state, cells may be dropped inside the ATM network. ABR, on the other hand, provides low cell and packet loss rates inside the ATM network, and most drops occur at the edge routers where the ABR queues may grow. If data is hierarchically coded and drop preference is indicated, these routers may be able to drop lower priority information before dropping the higher priority information.

8 Concluding Remarks

Table 6 gives a summary of the comparison of ATM service categories as discussed in this paper. As seen in the table, CBR, rt-VBR and nrt-VBR provide high quality of service, provided that the user can specify the traffic characteristics and quality of service requirements of the connection. Vanilla UBR is simple, but gives no guarantees. ABR provides fair and efficient utilization of bandwidth and exhibits good performance, though it requires the user to comply with the end system operations, and the network elements to indicate congestion state in the cells. The GFR service gives minimum rate and low loss without requiring end system cooperation, but network

Table 6: Comparison of ATM service categories

Category	**Nature**
CBR	Gives strict guarantees, but requires the user to define its traffic characteristics and QoS requirements, and is unsuitable for bursty traffic (little statistical multiplexing gains).
rt-VBR	Gives strict guarantees, but requires the user to define traffic characteristics and QoS requirements.
nrt-VBR	Gives loss guarantees, but requires the user to define traffic characteristics. Requires large network buffers.
UBR	Extremely simple, but gives no guarantees.
GFR	Gives loss and rate guarantees, but network elements must perform frame-level tagging/policing, scheduling or buffer allocation functions.
ABR	Gives loss and rate guarantees, but sources and network elements must perform a number of complex functions. Provides adaptive closed-loop feedback, and hence gives excellent control and utilization. Pushes queues to edge routers, with small queues inside the ATM network.

elements need to perform frame-level tagging, fair buffer allocation or scheduling operations to provide the guarantees.

ABR is unique because of its feedback control. It allows easier handling of drop preferences and priorities, and can best utilize added buffering. The edge devices can intelligently mark, drop and schedule flows based on the enterprise policy. Thus, our analysis indicates that a well-designed and engineered ABR implementation is capable of providing the most flexible QoS-based transport of enterprise traffic over ATM backbones. To support multimedia applications, the ABR service should evolve to provide end-to-end delay guarantees through the carrier network.

The basic concept and architecture developed in this paper are not only applicable to ATM networks, but also to any network implementing intelligence in the edge device, and flow control in the backbone. This includes frame relay networks using flow control.

References

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated services. RFC 2475, December 1998.

[2] Flavio Bonomi and Kerry W. Fendick. Rate-based flow control framework for the available bit rate ATM service. *IEEE Network Magazine*, 9(2):25–39, March/April 1995.

[3] Thomas M. Chen, Steve S. Liu, and Vijay K. Samalam. The available bit rate service for data in ATM networks. *IEEE Communications Magazine*, 34(5):12, May 1996.

[4] Sonia Fahmy and Raj Jain. *Handbook of Communications Technologies: The Next Decade*, chapter "Resource ReSerVation Protocol (RSVP)". CRC Press, 1999.

[5] Kerry W. Fendick. Evolution of controls for the available bit rate service. *IEEE Communications Magazine*, 34(11):35–39, November 1996.

[6] Sally Floyd and Van Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, August 1993.

[7] The ATM Forum. The ATM forum traffic management specification version 4.0. ftp://ftp.atmforum.com/pub/approved-specs/af-tm-0056.000.ps, April 1996.

[8] Mark W. Garrett. Service architecture for ATM: from applications to scheduling. *IEEE Network*, 10(3):6–14, May/June 1996.

[9] Rohit Goyal, Raj Jain, Sonia Fahmy, and Bobby Vandalore. Providing rate guarantees to TCP over the ATM GFR service. In *Proceedings of LCN'98*, October 1998. http://www.cis.ohio-state.edu/~jain/papers/lcn98.htm.

[10] Rohit Goyal, Raj Jain, Shivkumar Kalyanaraman, Sonia Fahmy, and Bobby Vandalore. Improving the performance of TCP over the ATM-UBR service. *Computer Communications*, 21(10):898–911, July 1998. http://www.cis.ohio-state.edu/~jain/papers/cc.htm.

[11] Fiffi Hellstrand and Andras Veres. Simulation of TCP/IP router traffic over ATM using GFR and VBR.3. ATM Forum/98-0087, February 1998.

[12] Van Jacobson. Congestion avoidance and control. In *Proceedings of the ACM SIGCOMM*, pages 314–332, August 1988.

[13] Raj Jain. Congestion control and traffic management in ATM networks: Recent advances and a survey. *Computer Networks and ISDN Systems*, 28(13):1723–1738, November 1996. http://www.cis.ohio-state.edu/˜jain/papers/cnis.htm.

[14] Raj Jain, Shivkumar Kalyanaraman, Sonia Fahmy, Rohit Goyal, and S. Kim. Source behavior for ATM ABR traffic management: An explanation. *IEEE Communications Magazine*, 34(11):50–57, November 1996. http://www.cis.ohio-state.edu/˜jain/papers/src_rule.htm.

[15] Raj Jain, Shivkumar Kalyanaraman, Rohit Goyal, Sonia Fahmy, and Ram Viswanathan. ERICA switch algorithm: A complete description. ATM Forum/96-1172, August 1996. http://www.cis.ohio-state.edu/˜jain/papers/atmf/a96-1172.htm.

[16] Shivkumar Kalyanaraman, Raj Jain, Sonia Fahmy, Rohit Goyal, and Seong-Cheol Kim. Performance and buffering requirements of internet protocols over ATM ABR and UBR services. *IEEE Communications Magazine*, 36(6):152–157, June 1998. http://www.cis.ohio-state.edu/˜jain/papers/ieee-mag.htm.

[17] Shivkumar Kalyanaraman, Raj Jain, Rohit Goyal, Sonia Fahmy, and Seong-Cheol Kim. Use-it or lose-it policies for the available bit rate (ABR) service in ATM networks. *Journal of Computer Networks and ISDN Systems*, 30(24):2293–2308, December 1998. http://www.cis.ohio-state.edu/˜jain/papers/uili.htm.

[18] John B. Kenney. Traffic management draft specifications. ATM Forum/TM 4.1, February 1999.

[19] A. Romanow and S. Floyd. Dynamics of TCP traffic over ATM networks. *IEEE Journal on selected areas in communications*, 13(4):633–641, May 1995.

[20] Steve Rosenberg, Mustapha Aissaoui, Keith Galway, and Natalie Giroux. Functionality at the edge: Designing scalable multiservice ATM networks. *IEEE Communications Magazine*, 36(5):88–99, May 1998.

[21] Bobby Vandalore, Sonia Fahmy, Raj Jain, Rohit Goyal, and Mukul Goyal. A definition of general weighted fairness and its support in explicit rate switch algorithms. In *Proceedings*

of the Sixth International Conference on Network Protocols 1998 (ICNP '98), pages 22–30, October 1998. http://www.cis.ohio-state.edu/˜jain/papers/icnp98_bv.htm.

[22] Bobby Vandalore, Sonia Fahmy, Raj Jain, Rohit Goyal, and Mukul Goyal. QoS and multipoint support for multimedia applications over the ATM ABR service. *IEEE Communications Magazine*, pages 53–57, January 1999. http://www.cis.ohio-state.edu/˜jain/papers/multabr.htm.

[23] Bobby Vandalore, Raj Jain, Rohit Goyal, and Sonia Fahmy. Dynamic queue control functions for ATM ABR switch schemes: Design and analysis. *Journal of Computer Networks*, 1999. http://www.cis.ohio-state.edu/˜jain/papers/cnis_qctrl.htm.

9 Vitae

Sonia Fahmy received her MS degree in Computer Science in 1996 from the Ohio State University, where she is currently a PhD candidate. Her main research interests are in the areas of multipoint communication, traffic management, and performance analysis. She is the author of several papers and ATM Forum contributions. She is a student member of the ACM, the IEEE, and the IEEE Communications and Computer societies.

Professor **Raj Jain** is an active member of ATM Forum Traffic Management group and has influenced its direction considerably. He is a Fellow of IEEE, a fellow of ACM, and serves on the editorial boards of Computer Networks and ISDN Systems, Computer Communications, and the Journal of High Speed Networks. He is the author of two popular books: "FDDI Handbook: High Speed Networking using Fiber and Other Media" published by Addison-Wesley and "The Art of Computer Systems Performance Analysis" published by Wiley. His publications and ATM Forum contributions and can be found at http://www.cis.ohio-state.edu/˜jain/

Sameh Rabie received his Master's and Ph.D. degrees in electrical engineering from McGill University in Montreal, Canada. He joined Bell-Northern Research in 1982, where he worked on a variety of advanced planning and development assignments, in the areas of modern software technology, multivendor network management, and multimedia and broadband networks planning and design. He is currently manager of the Carrier Traffic Management and Performance Group of Nortel Networks. Dr. Rabie is a member of IEEE and the Association of Professional Engineers of

Ontario. He has over forty published technical papers in major journals and international conferences.

Rohit Goyal is a PhD candidate with the Department of Computer and Information Science at the Ohio State University, Columbus. He received his BS in Computer Science from Denison University, Granville. His work has been published in several papers and ATM forum contributions. His other interests include distributed systems, artificial intelligence, and performance analysis.

Bobby Vandalore received his B.Tech degree in 1993 from Indian Institute of Technology, Madras and MS degree in 1995 from the Ohio State University, both in Computer Science. He is currently a PhD candidate at the Ohio State University. His main research interests are in the areas of multimedia communications, traffic management, and performance analysis. He is the author of several papers and ATM Forum contributions. He is a student member of the ACM and IEEE-CS.