NBBS traffic management overview

by H. Ahmadi
P. F. Chimento
R. A. Guérin
L. Gün
B. Lin
R. O. Onvural
T. E. Tedijanto

In this paper, we describe an integrated set of procedures used for bandwidth management and congestion control in high-speed packet-switched networks such as asynchronous transfer mode (ATM), which are part of IBM's Networking BroadBand Services (NBBS) architecture. These controls are designed to support a wide variety of services with different characteristics in the network and operate at different time scales: connection-level controls such as path selection, admission control and bandwidth allocation, and packet-level controls that discriminate between packets from different connections to support multiple levels of service guarantees. Connection-level controls are applied at connection setup time and are based on the connection characterization and the network state at that time. They perform efficient allocation of resources to ensure performance guarantees for connections while achieving high utilization of network resources. Various packet-level controls developed include access or rate control and intermediate node buffer management and scheduling. For connections that do not require explicit service guarantees, NBBS offers an available bit rate service. This service mostly relies on packet-level control in the form of an end-to-end rate-based flow control algorithm that regulates the flow of traffic into the network. This paper, in addition to providing an overview of the different mechanisms used for traffic management in NBBS, highlights how they interact to ensure efficient network operation.

Traffic management consists of the set of mechanisms that determine how to allocate and manage network resources as a function of connection requirements and characteristics. In traditional networks, this task was "relatively" simple, since connection requirements and characteristics were either deterministic (circuit-switched networks) or allowed for considerable flexibility in adjusting connection traffic in response to changes in the network state (packet-switched networks).

The task of traffic management in fast packet-switched networks such as asynchronous transfer mode (ATM) is, however, considerably more complex, because these networks support a much richer set of connection types, requiring a wide range of traffic characteristics and performance requirements to be supported in the network. For example, the *User-to-Network Interface Specification Version 3.1* of the ATM Forum (a communications industry consortium) allows connections to specify their peak rate, sustainable rate, and burst size and to select from different service classes that offer different types of service guarantees. The integration of different applications with different traffic characteristics and service requirements introduces a significant additional complexity to the management of network resources. Furthermore, although ATM standards specify dif-

0018-8670/95/$3.00 © 1995 IBM

ferent service classes and connection types and how to request them, they do not specify any mechanism to support them in the network. Corresponding mechanisms are viewed as *implementation* issues that are beyond the scope of the standardization efforts.

Networking BroadBand Services (NBBS) is a comprehensive network architecture; it goes beyond the specification of available connection types and services and defines the mechanisms and algorithms needed to actually support them. In this paper, we review the different components of NBBS that contribute to this support. We explain how they operate and relate to one another, and describe the functionality they provide. In addition, we also show how they are used to offer services that go beyond what is currently available in the standards. The description of the traffic management capabilities of NBBS is structured in three sections. The first two focus on the two levels at which NBBS traffic management operates: connection-level management and packet- or cell-level management. The third section illustrates how these mechanisms are put to use in NBBS to support a wide range of connection services.

Connection-level mechanisms are typically triggered whenever the network receives a new connection request. They involve issues such as specification and interpretation of traffic descriptors, selection of class of service, computation of bandwidth requirements, choice of a route across the network to the requested destination, and finally, the actual allocation of resources along the path of the connection. These mechanisms are closely related. For example, the bandwidth requirements of a connection clearly depend on its traffic characteristics and requested class of service. NBBS provides a coherent framework to deal with these issues, one that ensures efficient use of network resources.

Packet- or cell-level mechanisms operate at a smaller time scale since they specify the actions that the network is to take upon receipt of an individual packet from a connection. Such actions take place both at the entry points to the network and inside the network itself. At an entry point to the network, it is necessary to check that all connections comply with their traffic specifications (contract) so as to prevent misbehaving users from affecting the performance of well-behaving users inside the network. Checking also provides useful information to detect changes in users' traffic patterns, which can then be used by the network to automatically adjust the resources allocated to those users. Such access controls, however, do not eliminate the need for additional mechanisms to discriminate among packets from different connections at intermediate nodes. Such mechanisms typically consist of buffer management and packet scheduling policies, key items for the support of different delay and loss of service classes.

Connection-level controls

In NBBS, various mechanisms are invoked upon receipt of an incoming call request. The first step is to accurately determine the nature of the call request and its requirements for various network resources. NBBS supports several types of connections that correspond to different levels of services provided by the network. It is an important step to map the incoming request to the appropriate service class. The second step assumes that the connection has been properly characterized and proceeds with the actual computation of a path between the origin and destination(s) of the call. The last step is the call setup process during which resources are actually allocated to new connections by all the nodes and links along their paths.

Traffic descriptor. A first requirement for a network to be in a position to decide how to handle a new connection is that some form of descriptor be provided to it, identifying the traffic characteristics of the connection. There are a number of possible choices for traffic descriptors, reflecting different assumptions about the ability of the network to tolerate variations around the specified values. Broadly speaking, descriptors are either deterministic or statistical.

Deterministic descriptors are easier for the network to manage but impose hard limits on the traffic patterns generated by connections. For example, ATM standards allow connections to specify a maximum sustained rate and a maximum burst size. As these truly represent upper bounds on what can be transmitted to the network, users must typically make provision for these quantities to be significantly more than their "average" (typical) behaviors. In contrast, statistical descriptors can permit some amount of traffic fluctuation and, therefore, impose less stringent constraints on user traffic patterns. Such permission, however, requires that the network be capable of accounting for these fluctua-

tions when allocating resources and detecting variations that exceed the permitted range. The NBBS traffic management function provides this capability, and NBBS, therefore, supports statistical descriptors. Note that deterministic descriptors such as those specified in the ATM standard are also readily supported for those cases where they are the only ones available.

ATM traffic descriptor. In order to have an operational definition of the traffic parameters, the ATM Forum defined traffic parameters of an ATM connection with respect to a deterministic rule. The main advantage of a rule-based definition is that it is easy to determine, both by the user and the network, whether a cell is compliant with the definition or not.

In brief, traffic parameters of a cell stream are defined in terms of two "leaky-bucket-based" traffic descriptors: the peak cell rate and the sustainable cell rate. Each descriptor is defined in terms of a continuous-state version of the discrete-state leaky bucket mechanism. This algorithm is referred to as the generic cell rate algorithm, GCRA (T, τ). Both T and τ are in units of time. The parameter τ can be viewed as the amount of variation in time that the leaky bucket will allow a cell from its theoretical arrival time, which is at equally spaced intervals of length T.[1]

Three traffic parameters are identified to provide an envelope for the cell generation stream at the source. These are:

- Peak cell rate R_p
- Sustainable cell rate R_s
- Maximum compliant burst size B_c

They are given in terms of GCRA $(T_p, 0)$ and GCRA (T_s, τ_s), where

$$T_p = R_p^{-1},\ T_s = R_s^{-1}, \text{ and } \tau_s = (B_c - 1)(T_s - T_p)$$

The demarcation point between the user and the network is referred to as the user-to-network interface (UNI). A user is attached to an ATM network through a UNI. It is unavoidable that the user cell stream will incur a delay variation across a UNI caused either by traffic shaping at the customer premise network or by multiplexing cells of multiple connections onto the physical access channel due to the insertion of management cells. It is the responsibility of the user to account for this delay variation between the point of cell generation and the UNI. Therefore, the user "adds" (not necessarily linearly) the effects of these delay variations to its original traffic descriptors GCRA $(T_p, 0)$ and GCRA (T_s, τ_s) to characterize its traffic at the UNI through the descriptors GCRA (T_p, τ) and GCRA (T_s, τ_s'), where τ and $\tau_s', \tau_s' > \tau_s$ account for the cell delay variation. In fact, the parameter τ is called the cell delay variation (CDV) tolerance, whereas the parameter τ_s' is called the burst tolerance. Therefore, as far as the traffic contract is concerned, the effect of multiplexing on the original cell stream parameters is summarized by four parameters: the peak cell rate, the sustainable cell rate, the CDV tolerance, and the burst tolerance. These four parameters, defined through GCRA (T_p, τ) and GCRA (T_s, τ_s'), are part of a traffic contract.

The peak cell rate specifies an upper bound on the traffic that can be submitted on an ATM connection. The peak cell rate and the CDV tolerance are mandatory parameters and are supplied either explicitly (via UNI signaling) or implicitly (for permanent virtual circuits, or PVCs) during the setup. The other two traffic parameters, the sustainable rate and the burst tolerance, are optional parameters. They allow a somewhat more flexible definition of the traffic characteristics that enable the network to do more efficient resource allocation.

The rule-based deterministic traffic descriptors are attractive at the UNI where a "legal" contract may be required between the network and the user. However, this attraction comes at the expense of efficiency. In particular, as the traffic is defined using deterministic parameters, the statistical behavior of the cell arrival process cannot be characterized. This condition requires the network to allocate resources based on these deterministic requirements, which in turn would result in not-so-efficient use of network resources.

Frame-relay traffic descriptor. Similar to ATM traffic descriptors, frame-relay traffic descriptors are also rule-based. The peak rate is physically constrained by the access rate *AR*. Analogous to sustainable cell rate in ATM, frame relay defines the committed information rate *CIR*, but based on a sliding window mechanism as the rule. CIR is defined as the maximum number of bits (called committed burst size, B_c) that can be transmitted within any time interval T.

NBBS traffic descriptor. The NBBS traffic descriptor is statistical. It assumes that a simple two-state (on-off) source model can be used to capture the basic nature of the traffic generated by a connection. A source is either idle (off) generating no traffic, or active (on), transmitting traffic at its peak rate. The statistical nature of the descriptor means that various distributions can be selected for the active and idle periods. In this way, a flexible and general representation is provided. For example, the descriptor includes the previously described deterministic descriptors simply by selecting active and idle periods of constant duration. However, although the ability to specify any arbitrary distribution affords great flexibility, providing the necessary information may not always be feasible.

For that purpose, NBBS makes an initial assumption about the distribution of the active and idle periods, i.e., they are assumed to be exponentially distributed. As discussed next, the constraints imposed by requiring that the active and idle periods be exponentially distributed are minimal. In particular, it is possible to "map" sources with more general distributions onto "equivalent" exponential ones. The method underlying this generalization is described in the subsection on bandwidth adaptation function. The exponential distributional assumption provides a compact traffic descriptor and a simple procedure to determine the associated bandwidth requirements for the call.

The NBBS traffic descriptor **c** is of the form:

$$\mathbf{c} = (R_{\text{peak}}, \rho, b) \tag{1}$$

where the three components correspond to the peak rate at which the connection can transmit, its utilization, and the average duration of its active periods.

The peak rate R_{peak} specifies how fast a source is capable of generating data when it is active. Typically, the higher the peak rate of a source, the more resources are required from the network (even in the case of a fixed average rate). The utilization ρ gives the fraction of time the source is active. The peak rate and the utilization together determine the average and the variance of the bit rate of the source, two key factors in determining the amount of resources required. Finally, b is the average duration of the active period, indicating the average amount of data generated during an active period. The greater this quantity, the more bandwidth or buffering needs to be allocated to the connection. In the next subsection we describe precisely how the bandwidth required by a new connection is computed from these three parameters.

Bandwidth computation and accounting. Based on the traffic descriptor **c** and service requirements of a connection, the network must now determine the amount of bandwidth required to support this connection. This amount is between the mean rate and the peak rate of the connection. Allocating only the mean rate would be insufficient to meet a desired level of service. Peak allocation provides adequate performance guarantees. However, it usually causes inefficient use of network resources, particularly for connections that generate traffic at time-varying bit rates. The goal is, therefore, to provide a method for computing the "right" level of allocation. This goal must, however, be qualified by the additional constraint that the associated computational cost be compatible with the real-time processing requirements of connection management. The NBBS bandwidth computation method provides both accurate estimates of the bandwidth requirements of connections and low computational cost.

The approach is based on the combination of two approximations. The first one considers a connection in isolation and determines its bandwidth requirements as a function of its traffic parameters. The second approximation focuses on the interaction of connections within the network and captures the effect of statistical multiplexing on bandwidth requirements. As shown in Reference 2, these two approximations yield reasonably accurate bandwidth estimates over different ranges of connection characteristics. Together they give adequate and computationally efficient estimates for the bandwidth requirements of connections and the associated link loads.

The so-called "equivalent bandwidth" $\hat{c}$ required by an individual connection with traffic descriptor $\mathbf{c} = (R_{\text{peak}}, \rho, b)$ is estimated using a simple fluid-flow model as:

$$\hat{c} \simeq \Big[\alpha b(1-\rho)R_{\text{peak}} - x + \sqrt{[\alpha b(1-\rho)R_{\text{peak}} - x]^2 + 4x\alpha b\rho(1-\rho)R_{\text{peak}}} \Big] / 2\alpha b(1-\rho) \tag{2}$$

where x represents the available buffer space and $\alpha = \ln(1/\epsilon)$ where ϵ is the desired loss probability. In other words, Equation 2 gives the amount of bandwidth needed by a new connection with a traffic descriptor **c**, given that it requires a packet loss probability ϵ or lower when the buffer of size is x. Equation 2 is used for multiple connections[2] so that the total amount of bandwidth $\hat{C}_{(F)}$ needed by N connections with individual equivalent bandwidths $\hat{c}_i$, $1 \leq i \leq N$, can be approximated by:

$$\hat{C}_{(F)} = \sum_{i=1}^{N} \hat{c}_i \tag{3}$$

The equivalent bandwidth $\hat{c}_i$ for the i-th connection can be viewed as the circuit rate it requires from the network if viewed in isolation. The packet-switched nature of the network, however, allows sharing of resources, and there is no real dedication of bandwidth to individual connections. Although accurately reflecting their individual characteristics, this approximation may be conservative when the statistical characteristics of connections offer the potential for significant sharing of network resources. Another approximation is then needed to better capture the (statistical multiplexing) gain available from this sharing.

To obtain such an approximation, we focus on the stationary distribution of the aggregate bit rate of multiple connections. Based on this estimate, we then only allocate enough bandwidth to ensure that the aggregate bit rate remains below this allocated value with a sufficiently large probability. It essentially amounts to requiring that the probability of "overload" be kept below a desired level. There are many possible approaches to approximate the distribution of the aggregate bit rate of multiplexed connections, but a simple and effective one is to rely on a Gaussian distribution.[2] The availability of standard expressions for the tail probabilities of Gaussian distributions provides us with the necessary tools to estimate the amount of bandwidth that needs to be allocated.

In particular, the bandwidth $\hat{C}_{(S)}$ required by N connections multiplexed on the same link can be approximated by:

$$\hat{C}_{(S)} \simeq m + \alpha'\sigma, \text{ with } \alpha' = \sqrt{-2\ln(\epsilon) - \ln(2\pi)} \tag{4}$$

where m is the mean aggregate bit rate ($m = \sum_{i=1}^{N} m_i$), and σ is the standard deviation of the aggregate bit rate ($\sigma^2 = \sum_{i=1}^{N} \sigma_i^2$) of N connections. Equation 4 states that the aggregate bit rate exceeds the value $\hat{C}_{(S)}$ only with probability ϵ, under the assumption that its distribution is well approximated by a Gaussian distribution. Allocating this amount of bandwidth would in this case ensure a packet loss probability below ϵ. This approach provides a reasonably accurate bandwidth allocation rule when there is significant statistical sharing of network resources. However, as with Equation 3, it can also overestimate the required amount of bandwidth for certain types of connections.

These two approximations are inaccurate over different ranges of connection characteristics.[2] It is, therefore, possible to combine them to obtain a simple and yet reasonably accurate expression for the amount $\hat{C}$ of link bandwidth that the network should allocate to ensure the desired level of service to connections. In the case of N connections sharing the same network link, this expression is of the form:

$$\hat{C} = \min\left[m + \alpha'\sigma, \sum_{i=1}^{N} \hat{c}_i\right] \tag{5}$$

where the quantities $\hat{c}_i$ are computed from Equation 2 and m and σ stand again for the mean and standard deviation of the aggregate bit rate.

On the basis of this bandwidth allocation procedure, we are now in a position to compare the loading level of links in the network. Specifically, for each link the network maintains a set of link metric vectors (one for each quality-of-service class) from which the loading of that link can be readily obtained. For the k-th link, these vectors have the following form:

$$\mathbf{L}_k = \left[m = \sum_{i=1}^{N} m_i,\ \sigma^2 = \sum_{i=1}^{N} \sigma_i^2,\ \hat{C}_{(F)} = \sum_{i=1}^{N} \hat{c}_i\right] \tag{6}$$

where N is the number of connections currently multiplexed on link k, m and σ^2 are the mean and variance of the aggregate bit rate, and $\hat{C}_{(F)}$ is the sum of the N individual equivalent bandwidths.

An important property of the above link metric vector is that it allows incremental updates as connections are added or removed. Specifically, a request

vector is associated with each connection (or disconnection) based solely on the information provided in the traffic descriptor of the connection and the associated equivalent capacity computation described above. The request vector for the i-th connection is of the form

$$\mathbf{r}_i = [m, \sigma_i^2, \hat{c}_i] \tag{7}$$

where m_i, σ_i^2, and $\hat{c}_i$ are as previously defined. The new link metric vector $\mathbf{L}'_k$, after adding (or removing) the i-th connection, is then simply obtained by adding (or subtracting) the vector $\mathbf{r}_i$ to the current link metric vector, i.e.,

$$\mathbf{L}'_k = \mathbf{L}_k + \mathbf{r}_i \tag{8}$$

Based on the scheduling policy employed in the link, a given connection can be accounted for in multiple link vectors. For example, if real-time traffic is always transmitted before nonreal-time packets waiting in the transmission queue, a real-time connection is accounted for both in the real-time link metric and in the nonreal-time link metric. Therefore, in this case, the number of connections N in Equation 6 represents both real-time and nonreal-time connections. For the same real-time connection, the request vector added to (or subtracted from) the real-time link vector is different from the one that is added to (or subtracted from) the nonreal-time link vector. The reason is the buffer size x and loss target ϵ are generally different for each quality-of-service (QOS) class. For example, nonreal-time connections are accounted for only in the nonreal-time link metric, since they do not impact the transmission of the real-time packets.

The link metric vectors, or more specifically the information they provide on link bandwidth allocation levels, are the key to the ability of the architecture to compute routes capable of carrying new connection requests. A detailed description of the different NBBS routing algorithms can be found in Reference 3, but we provide next a brief outline of the different steps involved.

Path selection algorithm. In this subsection we outline the procedures NBBS uses to select and establish a route through the network. The main objective is to generate a route with enough available resources to accommodate the new connection, while attempting to optimize some long-term network revenue function, e.g., overall network utilization.

The computation of a route for an incoming connection request is performed at the node of origin of the request, i.e., NBBS relies on source routing. This is possible as each node in the network dynamically maintains a local replica of a network "topology" database. As its name indicates, this database contains information about overall network topology. It also includes the previously mentioned link metric vectors as well as additional link and nodal characteristics. The process used by NBBS to distribute and update this database involves minimal overhead.[4]

The objectives of the NBBS routing algorithm are somewhat different from those of traditional data networks. In legacy networks, a commonly used objective is to minimize quantities such as the average delay. (Reference 5 presents a review of various criteria and related algorithms.) Instead, the perspective in NBBS is closer to that of circuit-switched networks, where the goal is usually to maximize some measure of network performance such as the number of calls carried. (Reference 6 presents an introduction to these techniques.) This similarity to circuit-switched networks is a reflection of the guaranteed QOS requirements of connections.

There are, however, significant differences between the environment NBBS faces and that of a circuit-switched network. The heterogeneity of connection requests introduces different constraints and, in particular, as seen from Equation 5, can result in a connection being assigned different amounts of bandwidth on different links, depending on the respective traffic mix on the links. Similarly, although delay may not be a primary consideration given the high link speeds, the packet-switched nature of the network makes it a significant factor. Specifically, connection requests may often specify, in addition to their call metric, a maximum acceptable delay through the network. It is then necessary for the routing algorithm to also take this additional constraint into account.

Hence, the routing algorithm in high-speed networks is a rather complex optimization problem. Even in the simpler environment of a circuit-switched network with homogeneous and fixed bandwidth calls, the determination of "optimal" paths that maximize network throughput is a dif-

ficult task. Most proposals[6–11] rely on some form of approximations or heuristics. Because of this inherent complexity, which is further complicated in high-speed networks by the combination of both loss and delay constraints, we rely on a simple heuristic that reflects these different requirements.

The approach is to both balance the load in the network and favor short paths (fewer links), while controlling when and how calls can be routed over costlier longer paths. The routing algorithm is a modified shortest path algorithm where path length is a function of both hop count and individual link lengths. The length of a link is defined to be an increasing function of its load in order to promote load balancing. In addition, the algorithm accommodates the specification of a maximum path length constraint, which is useful in providing delay bounds. In order to avoid both instability and unfairness as the network load increases, the algorithm prevents the use of long *alternate* paths except when the associated alternate links are lightly loaded, using again technology similar to that of circuit-switched networks.[12–15] The motivation is to ensure that excess traffic is carried only when there are enough idle resources, so that it does not impact regular traffic. As mentioned earlier, details on the algorithm can be found in Reference 3.

Connection setup. Once a route has been selected, the last step before the traffic starts flowing is to secure the resources needed along the route. This is the purpose of the setup phase, which is responsible for verifying that the requested resources are indeed available, thereby preventing over-allocation of resources. This verification phase is necessary because of potential discrepancies between the load information used to compute the route and the actual state of the network. Such discrepancies are unavoidable because of the nonzero time needed to propagate changes in the network.

The reservation phase is carried out through a *setup* message, which is sent using a source-routed message addressed to the destination of the connection but with a copy being automatically dropped at each intermediate node. This method ensures rapid delivery of the message and minimizes setup delay. Upon receipt of such a message, each node checks whether enough bandwidth is available on the associated link. This check is performed using the procedure outlined in the subsection on bandwidth computation, after adding the request vector(s) of the new connection to the existing link metric vector(s) and verifying that the new load is acceptable. Note that if the connection (e.g., a real-time connection) is impacting multiple link metrics on a link, the admissibility criterion must be satisfied for all the affected link metrics. If this indeed is the case, a positive acknowledgment is sent back to the node of origin. Transmissions can start only after positive acknowledgments have been received from all intermediate nodes on the path.

Connection preemption and priority management. To reduce the impact of intermediate nodes possibly rejecting connection requests because of the unavailability of the amount of bandwidth requested, NBBS offers several options. For example, the source may recompute a new path when a setup request fails. Another option is to have the intermediate node allocate a smaller amount of bandwidth, one which can be accommodated on the link. This option then requires the source to adjust its traffic descriptor accordingly. Additional resources that may have been reserved on other links must also be released. Such adjustments can be performed either explicitly or by using embedded connection liveliness flows (similar to ATM Object Access Method, or OAM, flows[1]) that NBBS provides for connection management purposes. Finally, NBBS offers the possibility of connection preemption, which allows a high-priority connection to cause the disconnection of one or more low-priority connections already established along the path, in order to satisfy the requirements of the new connection. NBBS supports both setup and holding priorities, which let a connection specify how important it is for it to be successfully set up and avoid being preempted, respectively. Holding priorities are always greater than or equal to setup priority to avoid thrashing problems. In order to avoid major disruptions to preempted calls as well as to handle link failure, NBBS also supports a route-switching function that "automatically" reroutes calls. Details on this procedure can be found in the paper "NBBS Network Management" by Owen in this issue.[16]

After the connection has been established and data start flowing, NBBS continuously monitors its traffic. Monitoring serves several purposes. It is an effective means for capturing the impact of traffic statistics (nonexponential sources). It can also be used to identify significant changes in the characteristics of a connection, which may warrant an adjustment in the amount of allocated bandwidth.

Finally, it is also a powerful tool to identify misbehaving users, who can then either be warned or penalized accordingly. In the next section we describe these "packet-level" controls in greater detail.

Packet- and cell-level controls

End node controls. End node controls include the leaky bucket module, the bandwidth adaptation function, and the extended adaptive rate-based congestion control.

Leaky bucket module. Packet- and cell-level control is applied to connections that require performance guarantees to ensure that misbehaving connections do not degrade the quality of service of well-behaving connections. In NBBS, this control is provided by the leaky bucket module and becomes effective once a connection is established. The leaky bucket module performs traffic monitoring and shaping. It also performs traffic smoothing.

Traffic monitoring determines whether the connection is conforming, i.e., whether the statistical characteristics of the connection stay within the parameters used in allocating bandwidth through the network. If the connection is in a conforming state, packets or cells are sent to the network untouched. If the connection is determined to be nonconforming, traffic shaping is activated, i.e., selective packets are either marked, queued, or discarded so that the characteristics of the untouched traffic are kept within the negotiated parameters. Marked (red) packets have lower-loss priority in the network, i.e., intermediate links discard marked packets first when congestion starts to build up. We shall refer to packets that are not marked red as green packets. In the terminology of standards, red packets correspond to low-cell-loss-priority (CLP = 1) ATM cells or discard-eligible (DE = 1) frame-relay packets, whereas green packets are unmarked higher-loss priority packets or cells.

Traffic smoothing reduces and regulates the peak rate of the connection by putting appropriate spacing between packets using a spacer. Care is taken to make sure that the delay introduced by smoothing does not cause the end-to-end delay requirement to be violated by budgeting an amount of delay for traffic smoothing and computing the spacer rate based on this delay.

The NBBS leaky bucket module consists of two leaky bucket constructs—*green* and *red*—operating in tandem. Figure 1 shows the main components of NBBS traffic policing, monitoring, and shaping function (hereafter referred to as the *leaky bucket module*).

The parameters used to describe the operation of the leaky bucket module are:

γ_g = green token generation rate
M_g = green token pool size
γ_r = red token generation rate
M_r = red token pool size
β = spacer rate

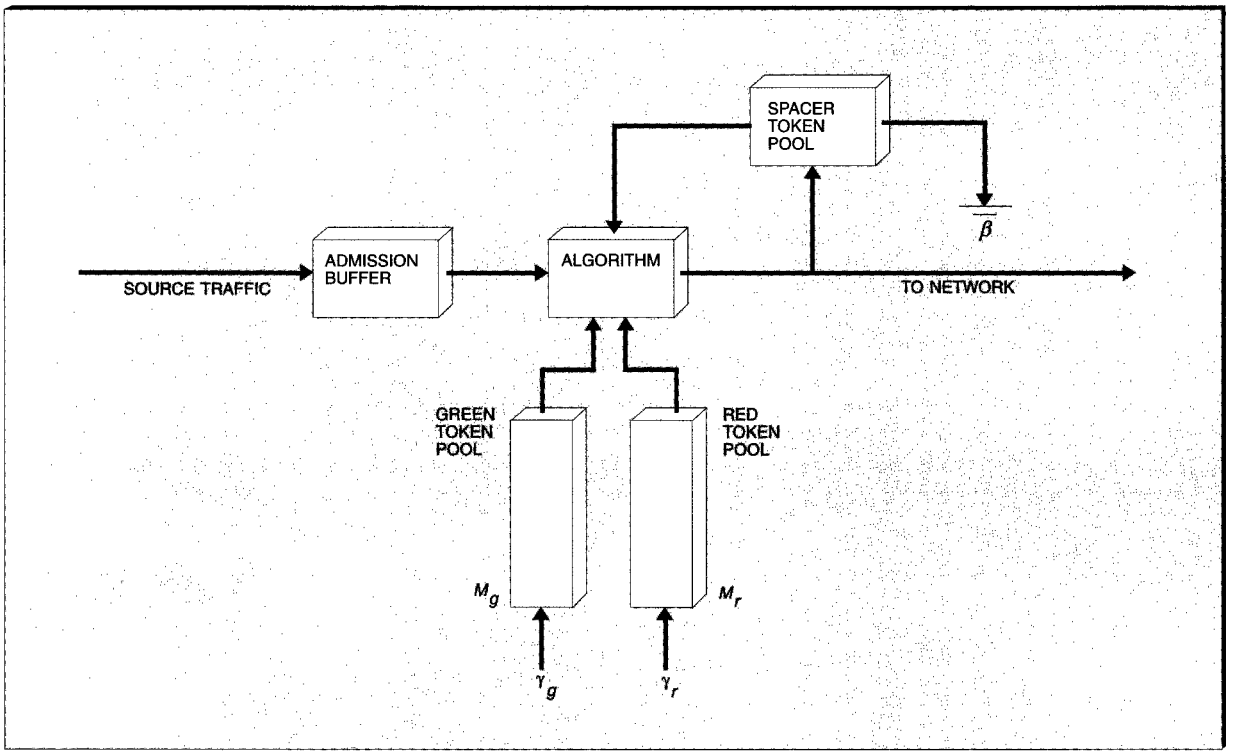
The leaky bucket module can be used in two different modes: standard and NBBS value-added. In the standard mode, the leaky bucket module operates identically to the GCRA mechanism used in ATM or frame relay. Proper mapping of parameters is done to ensure that the leaky bucket module performs the necessary level of policing given the rule-based traffic descriptors of the connection.

When the leaky bucket module is used in the NBBS value-added mode, it is used in conjunction with the bandwidth adaptation procedure described below. The functions of the green and red token pools in this case depend on whether the connection is currently in conforming state or not. When the connection is in conforming state, the green token pool primarily monitors the traffic characteristics by using the exponential substitution method described next.

The basic idea behind the exponential substitution method[17] is to use an exponential and independent on and off process as a substitute for a general on and off process. The substitute on and off process is chosen so that it approximates the behavior of the original process. Specifically, the substitute (or equivalent) on and off process is chosen such that it would experience a loss probability roughly equivalent to the original process if each were fed into an identical finite-buffer single-server queue (a link). This equivalence holds for a fairly large range of link parameters that covers most values of practical interest.

The exponential substitution method is a powerful tool for traffic monitoring, which provides a practical alternative to measurement-based approaches. Specifically, whereas peak rate can eas-

Figure 1 NBBS leaky bucket module



ily be enforced by a spacer and mean rate can readily be measured, mean burst length is in general much harder to estimate. Furthermore, if the traffic process is "known" to be nonexponential, it may then be necessary to also measure higher-order moments. This task can be very complex. The method of exponential substitution basically bypasses these difficulties by directly accounting for the impact of the different parameters, rather than trying to measure their values. This method enables us to readily capture generally complex traffic behaviors as a single parameter b_{eq}.

The exponential substitution method is based on monitoring the probability that arriving packets find no available tokens, and then determining the parameters of an equivalent exponential source that would experience the same probability. This method amounts to considering the green leaky bucket as a link with capacity γ_g and estimating the probability that the queue size exceeds the value M_g. Note that the value of M_g is computed so that the probability of running out of green tokens is normally below some nominal value ξ_T. This computation ensures that the exponential substitution method yields an equivalent exponential source that correctly estimates the actual traffic process over a wide range of parameters.

While the exponential substitution method allows the statistics of a complex traffic pattern to be accurately captured, it does not resolve all problems. In particular, an inherent drawback of using statistical traffic descriptors is that it takes some time to determine whether the traffic characteristics have indeed gone beyond the negotiated parameters. A misbehaving connection can potentially harm the network while the network is trying to determine whether the change in the traffic behavior represents a true shift in the statistical parameters of the traffic. The red token pool is used to guard against unacceptably large increases while the network is learning about the traffic. If the offered traffic load changes significantly within a measurement interval, such a change is detected, and traffic shaping is activated immediately before concluding that the connection is in nonconforming state.

For a conforming connection, the green and red token pools collectively limit the traffic mean rate entering the network to $\gamma_g + \gamma_r$. When the connection is in nonconforming state, the green token pool alone is used to shape the traffic entering the network, i.e., to limit the traffic mean rate and burstiness to γ_g and M_g, respectively. Traffic beyond these limits is marked red. The purpose of the red token pool in this case is to limit the amount of red traffic sent into the network.

The operation of the green and red token pools is as follows (assume one token represents one bit): Green (or respectively, red) tokens are generated at the rate γ_g (or respectively, γ_r) tokens per second. Tokens generated after the token pool of their respective color is full are discarded. A packet at the head of the admission buffer, after checking that the spacer token pool is empty (or waiting for the pool to become empty), checks the number of tokens in the green token pool. If there are sufficient green tokens (i.e., there are at least as many green tokens as the packet length), the packet is sent into the network as a green packet. The number of green tokens is reduced by the packet length. If there are insufficient green tokens, the packet checks the number of red tokens. If there are sufficient red tokens, the packet is sent into the network either as green or red, depending on whether the connection is in conforming state—green if conforming, red otherwise. The number of red tokens is decreased by the packet length. If there are insufficient red tokens, two cases are again considered, depending on whether the connection is in conforming state or not. If the connection is in conforming state, the packet is sent as red, and the number of red tokens is reduced to zero. If the connection is nonconforming, the packet is queued (or discarded if there is not enough space in the admission buffer). The packet is sent as green if green tokens become available first; it is sent as red otherwise.

The green token pool parameters γ_g and M_g are computed based on the following considerations:

- The mean rate of traffic entering the network is limited to $\gamma_g + \gamma_r$ when the connection is in conforming state and to γ_g when the connection is in nonconforming state.
- When the connection is in conforming state, we would like to choose M_g in such a way that the probability of a packet seeing insufficient green tokens (used in traffic monitoring for the exponential substitution discussed above) is at some target value, ξ_T.

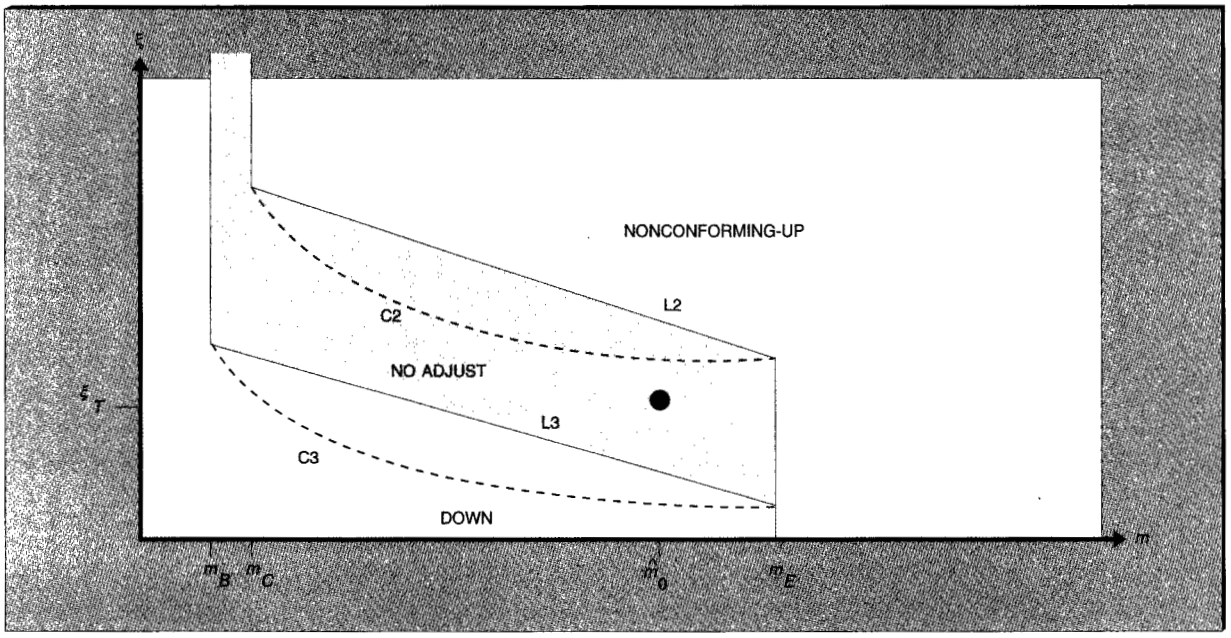
Bandwidth adaptation function. NBBS bandwidth adaptation provides for the packet- and cell-level access control to continuously monitor the user traffic parameters and dynamically initiate connection-level corrective actions (i.e., adjustments to bandwidth reservation) as the parameters change. This feature is particularly attractive to applications for which accurately specifying the traffic parameters during connection setup is not feasible. Bandwidth adaptation function consists of traffic monitoring, bandwidth estimation, and bandwidth reservation adjustment.

Traffic monitoring involves the measurements of the mean rate and mean burst length (peak rate is controlled by the spacer). The mean rate m is easily measured as the rate of traffic arriving into the system. As mentioned earlier, the effective mean burst length b_{eq} is measured indirectly by measuring the ratio ξ of arriving packets seeing insufficient green tokens.

Traffic estimation involves filtering the measurements of m and ξ measurements and determining that traffic parameters have changed sufficiently to warrant bandwidth adjustments. Traffic parameters are determined to have changed sufficiently if they stray out of a region around the old parameters in the parameter space. The bandwidth adaptation function uses an exponential filter to smooth out m and ξ measurements. The exponential filter and the adaptation regions are designed in such a way as to achieve the following tradeoff. On one hand, the system needs to be responsive to changes in traffic patterns when it is desirable to characterize the traffic as quickly as possible. On the other hand, the system should not respond to statistical variations in the measurements that do not represent real changes to the traffic patterns.

Figure 2 shows typical adaptation regions. The terms $\hat{m}_0$ and ξ_T are the current traffic mean rate and the target probability of running out of green tokens, respectively. The point $(\hat{m}_0, \xi_T)$ represents the current traffic parameters, i.e., the traffic parameters currently used in bandwidth reservation. The unshaded region around this point is the "no adjust" region. If the filtered m and ξ values fall within this region, no bandwidth adjustment is necessary. The shaded region below and to the left of

Figure 2 Typical adaptation regions



the "no adjust" region is the "down" region. If the filtered m and ξ values fall within this region, bandwidth needs to be adjusted downward. The shaded region above and to the right of the "no adjust" region is the "up" region. If the filtered m and ξ values fall within this region, bandwidth needs to be adjusted upward. Note that the regions have been simplified to be a polygon to allow real-time checking. Lines L2 and L3 approximate curves C2 and C3, which together with other lines defining the regions have been computed based on the knowledge of the traffic characteristics and of the confidence levels of the parameter estimation.

When bandwidth needs to be adjusted upward, the following steps are taken:

1. Bandwidth increase request: Once the connection is determined to be nonconforming, the origin of the connection sends a request message to each and every link along the path of the connection. This message contains the new traffic parameters, which are used by the links to decide if the increase can be accommodated.
2. Connection preemption and reroute: If a link cannot accommodate the bandwidth increase request, the link may preempt other connections of lower holding priorities to make room (see Reference 18). If this is not possible, the link rejects the request, and the origin in this case reroutes the connection.
3. Backpressure to the user: If the origin fails to increase the bandwidth of a nonconforming connection, possibly even after preemption and rerouting, it informs the user that its traffic needs to be reduced. The user may actually have detected the effects of traffic policing earlier and taken appropriate actions. Alternatively, the network may apply backpressure once the connection is determined to be nonconforming. Applying backpressure allows the user to reduce its traffic (by buffering data on the user side or slowing down the applications) before the traffic policing of the network adversely affects its performance objectives. Once bandwidth is successfully increased, the network can then signal the user to resume its normal behavior.

Reducing bandwidth is a much simpler process (i.e., the request to reduce the bandwidth is always granted by the links along the path) than increasing it.

Extended adaptive rate-based congestion control. High-speed, multimedia networks are expected to support a variety of services with different quality-

of-service requirements. With respect to their end-to-end delay and loss requirements, the set of such services can be classified into four categories:

1. Both delay- and loss-sensitive traffic, e.g., interactive video
2. Delay-sensitive but tolerant to moderate loss, e.g., voice
3. Loss-sensitive but tolerant to delays, e.g., file transfer
4. Tolerant to both moderate delay and loss, e.g., datagram services

The first two categories of services, hereafter referred to as reserved services, require end-to-end connections to be established before user traffic can start flowing. The last category of services is, in general, provided in the connectionless mode which may also be used for loss-sensitive but delay-tolerant traffic if appropriate controls are applied.

The latter types of services are referred to as best effort services in the literature. Their characteristics are summarized as follows:

- Bursty traffic that has an unpredictable behavior
- Delay tolerant
- No bandwidth allocation or explicit service guarantee

Best effort service increases utilization of network resources beyond what can be achieved by reserved traffic alone. However, when both types of traffic are integrated in the network, it is natural to grant the reserved traffic higher-service priority so that best effort service does not cause degradation to the service provided to reserved services beyond their acceptable values.

If no control is applied to best effort service, congestion at corresponding buffers in the network would rise as the rate of submitted traffic approaches the available capacity. This congestion would in turn cause packet and cell losses and end-to-end retransmission of packets, thereby reducing effective utilization of available bandwidth used for best effort service.

Extended adaptive rate-based (EARB) congestion control is an end-to-end congestion avoidance algorithm that employs a sequence of control packets called *sampling packets* to collect congestion information in the network. Its functions are split into two parts that are implemented at the access agents (AA) where the connection starts and terminates. The congestion information is interpreted and sent back to the originating AA by the destination AA. Figure 3 provides an overview of the EARB algorithm and the location of its functional components.

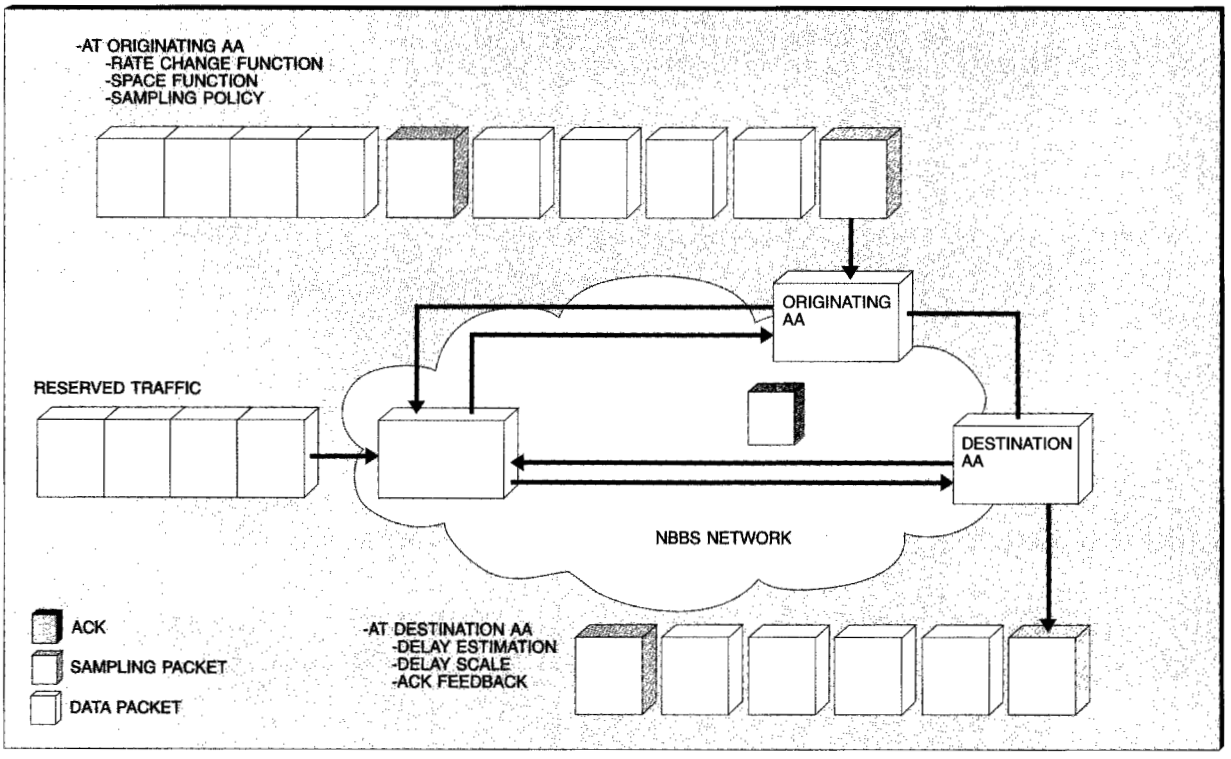
During the connection setup, the path selection function selects a set of links from the topology database that is considered to be the best route for the connection. EARB then estimates the congestion status, i.e., how many cells are queued in the switches, on the slowest link of the path. After the connection setup is complete, the originating AA is allowed to send packets that are spaced at an initial sending rate. The sending rate is modified to react to the feedback from the destination AA. The frequency of the sampling packets that are mixed with the data stream to the destination AA is determined by a combination of the round trip propagation delay and the desired sampling overhead. Once the sampling packets arrive at the destination, their delay will be estimated and weighed on a delay scale that results in sending acknowledgment packets to the originating AA. In summary, EARB consists of five functional components that will be further described later:

- At the originating access agent
  1. Spacing function
  2. Sampling policy
  3. Rate change policy
- At the destination access agent
  1. Delay estimation
  2. Delay scale

The unique feature that distinguishes EARB from the well-known FECN (Forward Explicit Congestion Notification)[19] and BECN (Backward Explicit Congestion Notification)[20] is employing the sampling packets to detect the network congestion status. It provides several advantages:

- It removes the congestion estimation algorithm from the intermediate switches to the end nodes so that a more sophisticated estimation algorithm can be implemented without increasing the complexity of the intermediate switches.
- It enables protection timers such as time-to-suspend and time-to-reset to prevent the network from getting into a severe cell loss situation and to recover from it if it does happen.

Figure 3 Overview of the ARB algorithm



Spacing function. The spacing function determines the minimum time between back-to-back packets entering the network. Given the current allowed rate $X(t)$ *Mbps* and the size of the first packet in the transmission buffer $b(n)$ *bits*, the next packet is allowed to enter the network at $b(n)/X(t)$ seconds later.

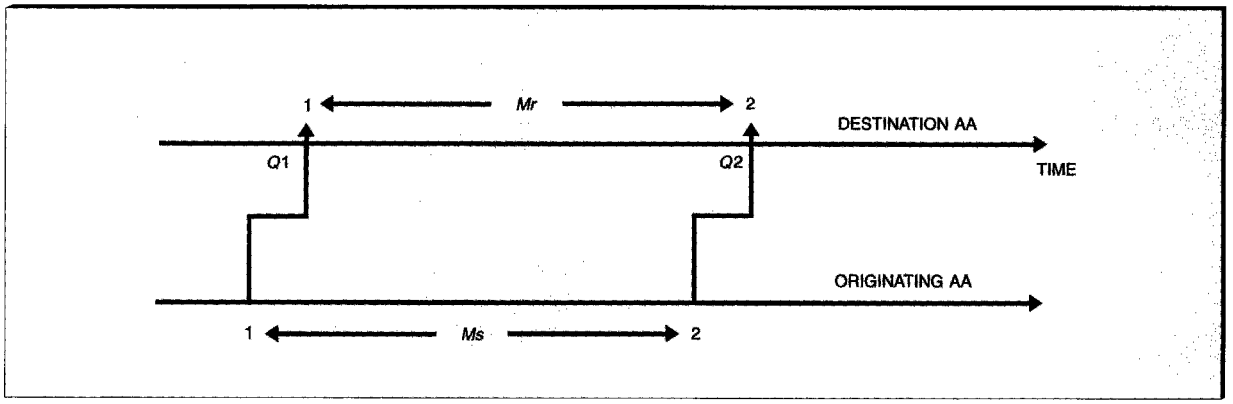
Sampling policy. A timer is activated and guarded by two thresholds, *T_SUSPEND* and *T_OUT*, right after a sampling packet is sent. The *T_SUSPEND* copes with the situation when the sampling packet or its associated acknowledgment is queued at a congested switch. The transmission is suspended until further information is available—either an acknowledgment arrives or *T_OUT* is reached. The magnitude of *T_SUSPEND* is on the order of the *round trip propagation delay*. If *T_OUT* is expired, EARB assumes that either the sampling packet or its acknowledgment is discarded by the severely congested network. EARB will reset and is ready to send the first sampling packet. *T_OUT* can be an integer multiple of *T_SUSPEND*.

A sampling packet is sent if the following conditions are true or the *T_OUT* threshold is reached:

1. The acknowledgment for the previous sampling packet has arrived.
2. A specified amount of data defined as the *sampling burst* has been entered into the network since the last sampling packet. This condition ensures that a desired sampling overhead can be achieved.

The extra bandwidth consumed by the sampling packets according to the sampling policy is bounded by the ratio of *sampling packet size* to *sampling burst size*. The overhead reaches its upper bound when the sampling frequency is determined by the first condition; acknowledgments come back before a sampling burst of data is sent. The overhead can be smaller if more than one sampling burst of data are transmitted before acknowledgments return.

Figure 4 Relationships of *Ms, Mr, Q1,* and *Q2*



Delay estimation. The queuing delays of the sampling packets, denoted by $Q2$, can be extracted from the following information:

- Sampling interval, Ms, carried in the sampling packets
- Interarrival time, Mr, measured at the destination AA
- Previous estimated network delay, $Q1$, initially set to zero

The relationship of these parameters is demonstrated by Figure 4, where the propagation delays and transmission times are not shown to simplify the diagram. From that, we can easily deduce the following equation:

$$Mr + Q1 = Ms + Q2 \tag{9}$$

By bounding $Q2$ to be nonnegative, Equation 9 can be rewritten as

$$Q2 = \max(Mr + Q1 - Ms, 0) \tag{10}$$

The first sampling packet is assumed to be delay free, and its arrival time is used as a reference point to estimate the queuing delays of the following sampling packets. By doing that, the estimation process is contaminated by the actual delay that the first sampling packet suffers. Fortunately, such error is bounded and converges according to the following observations:

Observation 1: If $D(n + 1) \geq D(n)$ then $ER(n + 1) = ER(n)$, where $ER(n) = |D(n) - Q(n)|$.

Observation 2: If $D(n + 1) < D(n)$ then $ER(n + 1) = ER(n)$, where $D(n)$ is the delay seen by the nth sampling packet and $ER(n)$ is the error associated with its delay estimate.

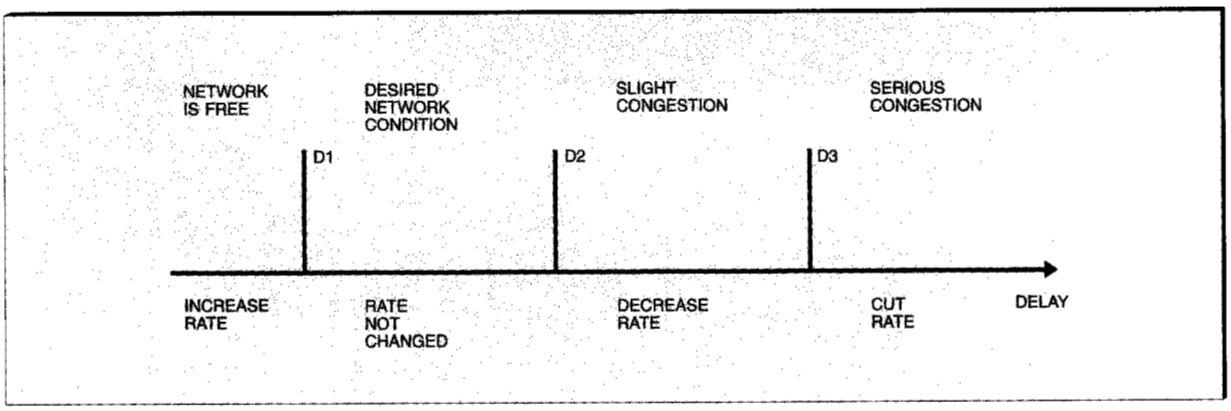
As the sampling packets are continuously sent during the connection, the absolute delays can be discovered when one of the sampling packets arrives without delay. Clearly, the initial phase of the process is critical. Therefore, in order to minimize possible damage, the slow-start policy is adopted: i.e., new connections will start with a relatively low transmission rate.

Delay scale. The estimated delays are weighted on the delay scale as shown in Figure 5 where four levels of congestion are classified. The classification is based on the consideration that network congestion is caused by the contention among the EARB users and the blocking by the high-priority reserved traffic.

These four levels are:

1. Network is free ($Q2 \leq D1$)—Virtually no delay is detected, connections are allowed to increase their rates.
2. Desired network condition ($D1 < Q2 \leq D2$)—A desired network utilization, connections maintain the current rates.

Figure 5 Network congestion scale



3. Slight congestion ($D2 < Q2 \leq D3$)—This region prevents the EARB connections from leaving the desired operation region. Connections take small cuts to the current rates.
4. Serious congestion ($Q2 > D3$)—It reflects the situation in which the increasing high-priority traffic blocks the EARB connections from using the transmission facility. All EARB connections take a deep cut to avoid congestion.

Rate change policy. Once the congestion level is determined, the allowed rate is adjusted accordingly. The rate change policy has to consider three conflicting criteria: efficiency, stability, and fairness. The compromise approach is large increases when the allowed rates are low for efficiency and smaller increases when the allowed rates are already high to gain stability. When congestion is detected, apply a larger reduction if the rate is high and a smaller reduction if the rate is low. The logarithmic function is chosen to meet these objectives:

$$F(1) = \textit{initial rate},$$

$$F(i) = C \times \ln\left(\exp\left(\frac{F(i-1)}{C}\right) + 1\right); \text{ for } i \geq 2 \quad (11)$$

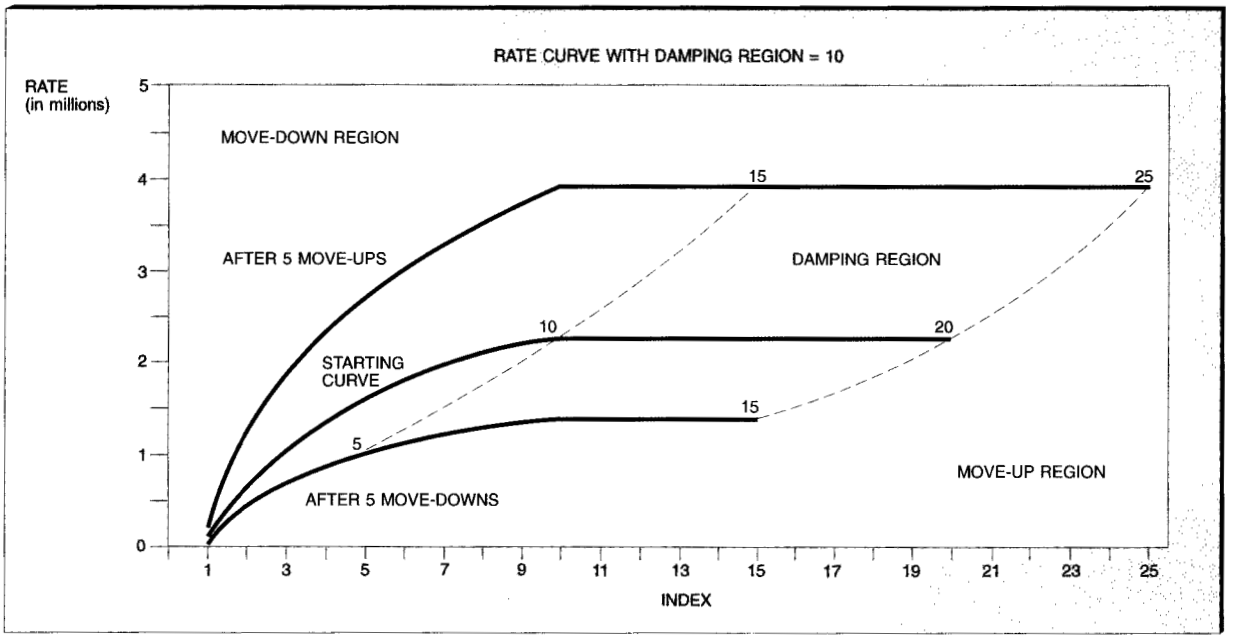
Considering its pseudo-asymptotic nature,[21] it is critical to choose adequate C so that one connection can reach its even share, which is determined by the total capacity, the reserved bandwidth, and the number of the nonreserved connections, in a desired number of steps in order to achieve aggressiveness and efficiency. It is expensive, however, to obtain this information in real time effectively. To cope with this problem, the rate change policy adopts a table-driven technique as follows:

Step 1. We construct a standard S-entry table R, $R(i) = F(i)$, for $i = 1 \cdots S$, where S is chosen so that $F(S)$ is the projected even share on the most likely congested link and S satisfies the desired aggressiveness. For clarity in presentation, we introduce a variable called *STATE* to trace the status of the connection with respect to its transmission rate. *STATE* is initially set to 1, pointing to the first value in the rate table $R(1)$.

Step 2. The first phase of the policy is to adjust the transmission rate using the values in the rate table. An *increase* acknowledgment will increase *STATE* by one to locate a new rate in the table. *STATE* can be reduced by a factor to move backward when a decrease acknowledgment is received. For stability and fairness considerations, the movement on the table is an additive increase and a multiplicative decrease based on the results published in Reference 22.

Step 3. The next phase of the policy will be triggered to adjust the table to match the current network condition. We define a right table as having been located if *STATE* oscillates within a so-called *damping region*, i.e., $F(S)$ is the even share at the moment. The damping region is initialized as $(L, L + DR)$, where L is the lower limit and DR is the size of the region. If *STATE* stays in the region, the table R is considered to be appropriate for the current situation. Once *STATE* grows over the region, the values in the table are increased

Figure 6 The movement of rate curve



multiplicatively, and the damping region is moved up by one step, $L = L+1$. After that, *STATE* is reset to S. Consequently, if *STATE* never reaches L, the table R is considered to be too high. By tracking the number of sampling packets sent while *STATE* is below L, a decision can be made as to when to reduce the table multiplicatively.

The multiplicative modification of the table preserves the logarithmic characteristic but may lose fairness among connections using different tables, i.e., when the starting times are different. This type of fairness is secured by linearly moving the damping region and resetting STATE to S when a modification takes place. That certainly favors the connections using the lower tables to compare the connections using the higher tables.

With the policy described above, a rate evolution with $S = 10$ is shown in Figure 6. The connection using the upper curve has received five "move-ups" from its initial table which is the middle curve. The connection has to move its *STATE* from 10 into the current damping region (15,25) in order to maintain the current table.

The EARB algorithm not only enables the BBNS network to provide the best effort service in an easy and effective way but also decouples the service capability from the intermediate switch architecture because of its end-to-end nature. The best effort connections controlled by EARB share the bandwidth efficiently when it becomes available and reduce or stop their traffic fully in time to avoid cell loss. A bounded memory size to achieve loss-free service is available using the EARB algorithm. EARB also distributes the available bandwidth evenly among the active connections regardless of the ages of the connections.

The end-to-end characteristic and rate-based approach of EARB provides seamless synergy with the flow control mechanism defined by the ATM Forum for its available bit rate (ABR) service. The resource management (RM) cells sent by the ATM end station across UNI samples the queue length at the originating AA. This AA, which implements the ABR destination functions (virtual destination), returns RM cells to the source end station across the UNI to adjust the allowed rate and forms an ABR control loop. Note that the queue length appearing at the originating AA is the result of the EARB flow control between the originating and destination AAs. At the destination UNI, the destination AA forms another ABR flow control loop with the destination end station by implementing the ABR

Figure 7 Interoperability of ABR flow control and EARB algorithm

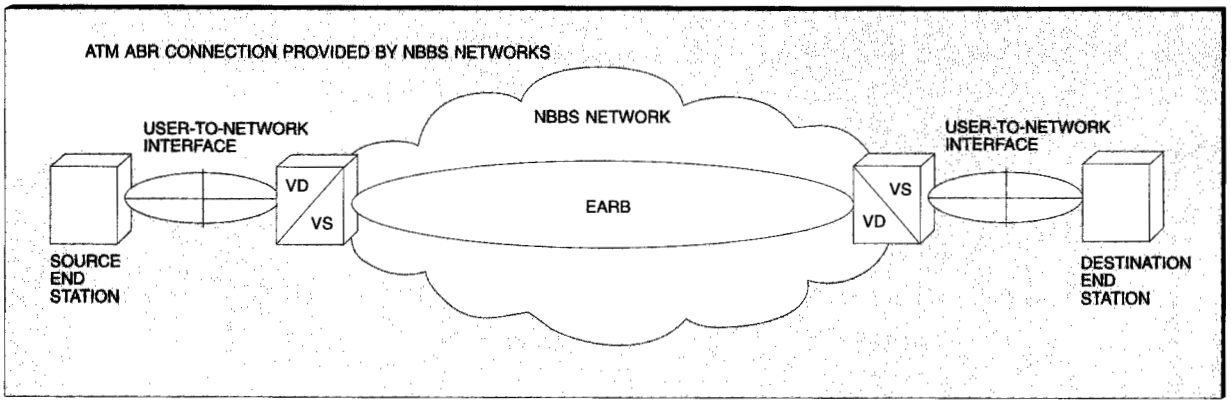
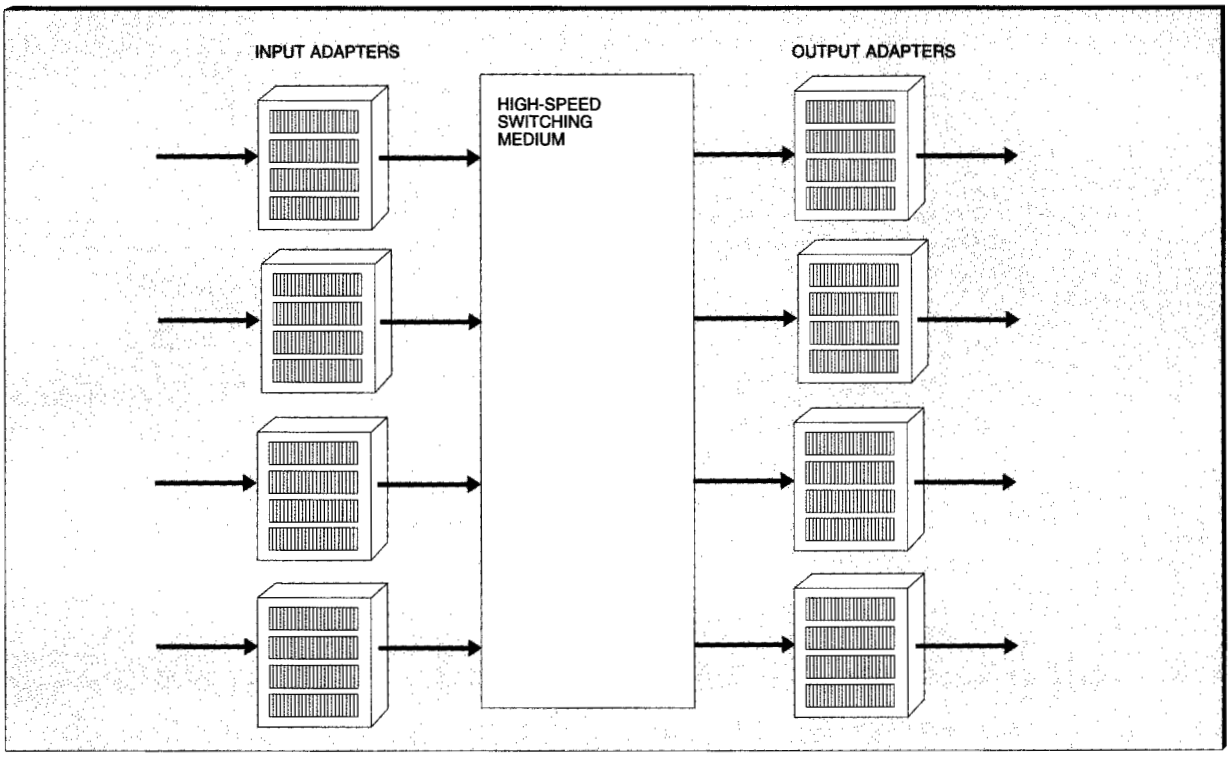


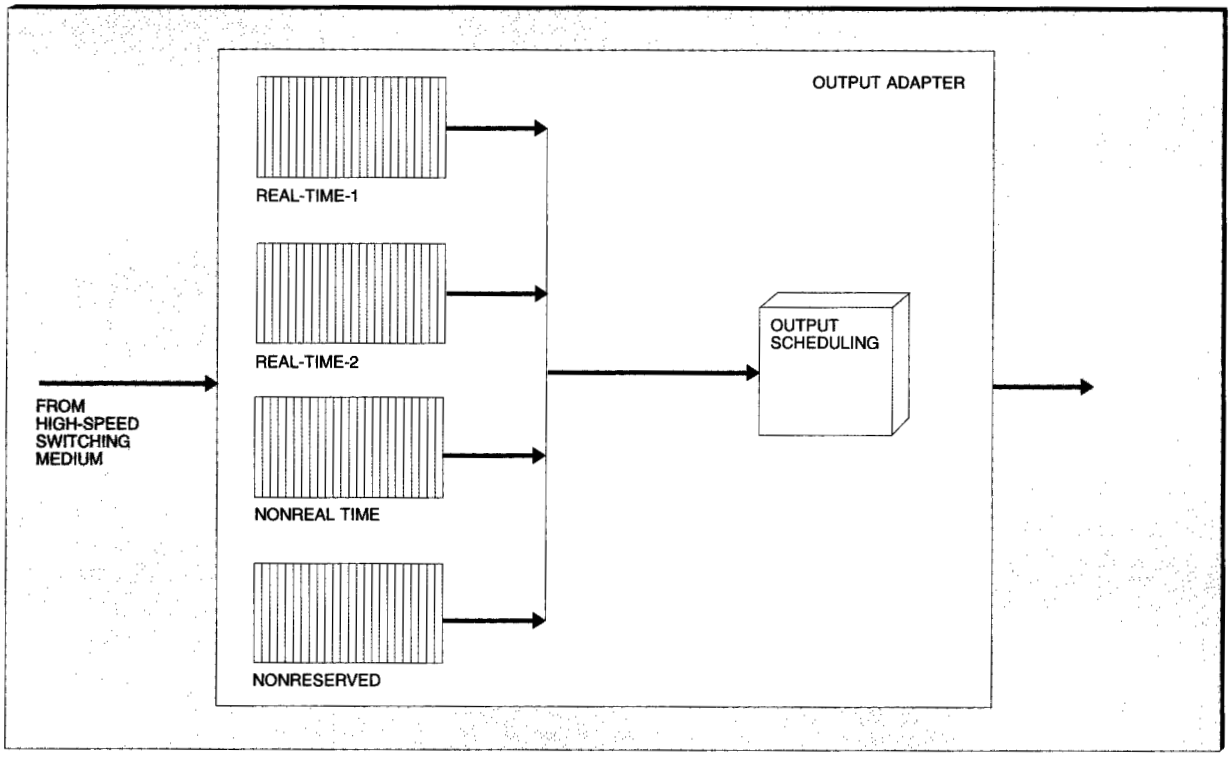
Figure 8 Transit node switching model



source functions (virtual source). The complete end-to-end flow control mechanism for ATM best effort service in BBNS networks is illustrated in Figure 7.

Intermediate node controls. The conceptual model for an NBBS transit node is composed of a high-speed switching medium, the input adapters attached to the incoming links, and output adapters

Figure 9 Output adapter structure



attached to the outgoing links (Figure 8). The actual placement of function among adapters, switches, etc., may vary depending on the implementation.

On the input side, an input adapter receives packets from its incoming link and may need to queue the packets while waiting for access to the switching medium. When the input adapter gains access to the switching medium, it sends one or more complete packets in the same order in which they were received.

The switching medium routes the packets sent by the input adapters to the appropriate output adapters. The switching medium must ensure that the sequence is maintained for packets sent between the same input and output adapters.

The output adapter takes the packet off the switching medium, buffers it, and transmits it on the outgoing link based on a scheduling policy that depends on the QOS priority of the packet. Figure 9 shows the structure of a typical output adapter.

In general, if the buffer space on either the input or the output adapter is full, arriving packets are discarded. For the purposes of this discussion, it is assumed here that the switching medium is fast enough so that there is minimal queuing at the input adapters. Therefore, the different types of priority classes defined below are for the output adapters only. However, depending on the switch-speed-to-trunk-speed ratio, a system may consider enforcing the same priority structure at the input adapters as well.

Priorities. Two different types of priorities are defined in NBBS: delay priority and loss priority. The *delay priority* differentiates between different classes of traffic, and each class shares a different logical buffer. Four classes are defined: circuit emulation (e.g., PCM voice), real-time traffic (e.g., voice, interactive video), nonreal-time traffic (e.g.,

data, playback video), and nonreserved traffic, (e.g., datagram-type traffic for which no explicit bandwidth reservation is required).

Loss priority. The *loss priority* differentiates the packets of the same class that share a common buffer, so that based on the level of congestion, packets with lower-loss priority will be discarded before the packets of higher-loss priority. In general, a packet may be assigned both delay priority and loss priority. The loss priority assignment is provided to those classes of traffic that are subject to explicit bandwidth allocation. No loss priority is associated with nonreserved traffic. For bandwidth reserved traffic, in general, two levels of loss priorities are defined. Packets with the higher-loss priority level are referred to as *green packets*, whereas those with the lower-loss priority level are referred to as *red packets*.

The marking of the loss priority of a packet is generally performed as part of the input congestion control function. Packets are either marked red because they are deemed not conforming to the traffic contract as explained earlier, or they may be marked (or treated as) red because they represent excess traffic per agreement with the user during connection setup. Examples in this category are frame-relay traffic that is marked discard eligible (DE = 1) or ATM traffic with cell loss priority bit set (CLP = 1) by the user. In either case, the intermediate nodes treat DE = 1 or CLP = 1 traffic as red only if the traffic contract does not include cell loss guarantees for this traffic. Therefore, red packets entering the network are excess traffic; thus intermediate nodes do not have to provide any implicit or explicit QOS guarantees for red packets.

There are various alternatives for the treatment of packets with different loss priorities sharing a common buffer. We would like to devise a control strategy that will allow us to transmit the guaranteed green packets through the node with as little interference from the red traffic as possible. At the same time, it is desirable not to impede the flow of the red traffic unless it is necessary to do so. The strategy must be simple enough so that it can be implemented in VLSI (very large scale integration) and does not result in processing bottlenecks.

One simple and effective buffer management scheme is based on a *threshold policy*. Traffic of all loss priority levels (both green and red traffic, and for real time, packets containing fields with various levels of significance) are allowed into the buffer until the total number of bytes in the buffer reaches some threshold, after which subsequent arrivals with low-loss priority are dropped. Only when the total number of bytes in the buffer goes below the threshold are arrivals with low-loss priority permitted into the buffer. This simple policy provides service to low-loss priority traffic, while protecting the high-loss priority packets from excessive delay (that may be due to an excessive number of red-marked cells waiting in the queue) and loss in the event of congestion.

In the case of real-time traffic, four levels of loss priority are defined. They are referred to as levels 0 through 3, with level 0 being the highest level of loss priority. For convenience, four colors have been defined: *green* (0), *white* (1), *red* (2), and *blue* (3). It is important to note that this encoding is not used for ATM cells, because this flexibility is not currently defined in the ATM standards.

The loss priority (i.e., color) of a packet is determined by the application based on the relative significance of the bits in the packet. As an example, speech sample values generated by a coder on a voice connection are collected into packets for transmission across the NBBS network. In general, some bits in the coder output are less significant than others in the sense that the perceived degradation in the quality of the reconstructed signal is smaller if the less significant bits are lost or in error than if the more significant bits are lost or in error.[23]

By arranging the bits in the coder output into packets in a way that all of the most significant bits collected over a single packetization interval are in one packet, all of the least significant bits are in another packet, and so on, it is possible to associate lower-loss priority with the packets containing the bits having lesser significance. Based on the level of congestion, intermediate nodes may drop the packets having lesser significance before they drop those having greater significance. The degradation resulting from taking advantage of loss priorities in this way for traffic such as voice and video is generally smaller than that resulting from the loss of an entire packet (i.e., all the bits collected from the source over a single packetization interval).

To get optimum performance for the four-color coding in the NBBS packet header, an NBBS trunk

can optionally support four discard thresholds, in addition to supporting the excess threshold to discard the usual red traffic described in the preceding sections. The thresholds of a real-time buffer supporting this tower are ordered as green (0), white (1), red (2), blue (3) and excess (red), green being the highest (typically buffer size) and excess (red) being the lowest threshold. How the threshold values are selected and how the notion of equivalent capacity is extended for multiclass (multicolor) traffic are discussed in Reference 24.

Delay priority. Circuit emulation traffic has the highest transmission priority to emulate the performance of a circuit-switched network. This priority is required to meet the strict delay and delay variation requirements of this type of traffic. Real-time traffic is given delay priority over nonreal-time traffic in scheduling transmissions on the outgoing link in order to reduce its delay, and nonreal-time traffic is given delay priority over nonreserved traffic in order to minimize the impact of this class on other classes. However, within a given delay priority, the packets are scheduled for transmission in the same order in which they arrived, independent of their loss priorities. This scheduling ensures FIFO (first-in first-out) service of user packets and enables delivery in sequence of packets to the receiver.

In general, there will be some interaction between loss priority and delay priority. For example, it is desirable that red real-time traffic be discarded before green nonreal-time traffic. This objective is achieved by making discard decisions on low-loss priority real-time traffic based on thresholds in both the real-time and nonreal-time buffers, i.e., red real-time packets are discarded if either the byte count in the real-time buffer or the byte count in the nonreal-time buffer is beyond the respective thresholds.

There are various alternatives for delay-priority scheduling among the traffic classes. *Nonpreemptive* scheduling could be implemented where the buffer for the lower-priority class is served only if the buffer of the higher-priority class is empty. However, the service of a lower-priority packet is not interrupted when a higher-priority packet arrives. This scheduling policy may be suitable for high-speed links.

Alternatively, *preemptive resume* scheduling could be implemented where the service of a lower-delay priority packet is interrupted upon the arrival of a higher-priority packet. The preempted packet is served again starting at the point where it was preempted, once there are no more higher-priority packets. Thus the only time at which a lower-delay priority packet can be in service is when there are no higher-priority packets. This scheduling policy may be used for low-speed links.

Supporting different QOS for different traffic types

The NBBS architecture supports a wide range of services. In this section, we describe various kinds of service classes that have been discussed in the ATM Forum and in the international standards bodies. NBBS supports all standard service classes while providing various value-added services for integrating them in the network. Conceptually, many different types of services are possible in an integrated network.

It is important to note that these descriptions involve the specifications of a number of traffic and service parameters, grouped by the standards organizations into service classes. Within each service class, specific parameters are supported in NBBS. In particular, the QOS parameter values may be different for different connections that request the same service class. NBBS is capable of supporting any set of QOS parameters that are specified (as long as they are consistent).

Constant bit rate service. Constant bit rate (CBR) service describes sources where all packets or cells are equally spaced in time. Such sources have a constant bit rate, and the packetization of information generated is periodic. For example, a source that has a constant bit rate of 2.048 Mbps will produce 5445 cells per second (assuming AAL-1 encoding of the information and using all 47 bytes per cell payload), which gives a cell rate of one cell per 184 microseconds (μs). Typically, CBR service is used for circuit emulation—that is, emulating for the user the characteristics of lower-speed access circuits. There are circuits (for example, International Telecommunication Union—Transmission Subsystem G.702 signals) that are not locked to a network clock and others that are locked. In either case, the source and target stand in some definite timing relationship to one another. The provider of CBR service is required to maintain that timing relationship. NBBS provides the basic frame-

work to support connections with strict delay and delay variation requirements from the network.

Although NBBS can support CBR traffic at any reserved delay priority, it is perhaps interesting to describe the technique for supporting circuit emulation. For this type of traffic, small transmission buffers may be required to bound the delay and delay variation. For example, the queue size may be chosen so that a connection taking a reasonable number of hops across relatively short distances will not need echo cancellers.

When such a connection setup is requested, NBBS first attempts to find a path through the network that meets tight delay and delay variation constraints. If an NBBS trunk supports this circuit emulation mode, it will so indicate in its topology database entry. The path selection algorithm will use this information in order to choose appropriate links for the path. The bandwidth for any CBR connection is very simple to compute. Assume, for example, that there are five 64 Kbps circuits being carried on the same connection and assume that the packetization delay is 0.5 millisecond (ms). Each circuit will produce four bytes in the packetization interval (assuming that one byte of information is produced every 125 μs). So the packet will have 20 bytes, plus a header of, say, five bytes. The bandwidth required will be $(25 \times 8)/0.0005 = 400\,000$ bps. The connection agent computes this value and sends it in the bandwidth request vector in the connection setup message and indicates that the connection being requested is a CBR connection. The connection setup message also includes an indication that the connection requires circuit emulation service and indicates the maximum packet size for the connection.

When the intermediate node receives the message, it performs two functions. First, the transit connection manager adds the bandwidth to the link metric for that priority (i.e., for the real-time priority). Next, it checks to see whether the remaining space in the circuit emulation buffer can accommodate the maximum packet size of the new connection. The reason for this operation is that if all CBR connections using the circuit emulation buffer are "in phase," the buffer must hold all the packets without loss. This provides circuit emulation connections with a very high QOS at the cost of more buffers per connection and fewer connections supported per trunk.

CBR connections that do not have the stringent requirements of circuit emulation service can use the normal buffer mechanisms of the real-time or nonreal-time reserved traffic priorities. They are set up by computing the bandwidth required to accommodate the connection, but without the additional step of subtracting the packet size from the remaining buffer space. Because with CBR traffic the mean bit rate is equal to the peak bit rate, there is no special bandwidth saving, and the equivalent capacity is just the peak bit rate of the connection.

Available bit rate service. Available bit rate (ABR) service is a "best effort" service for use with certain types of data traffic. ABR service does not provide any explicit QOS guarantees to the terminal equipment. One application of ABR service is datagram service; another is for off-shift file transfer where delay guarantees are not necessary. However, in order to provide some reasonable level of service to ABR traffic, some sort of control of these connections is necessary. Currently, the ATM Forum is finalizing the specification of a rate-based proposal for the flow control of ABR traffic. This proposal can be readily supported through the NBBS extended adaptive rate-based (EARB) flow control mechanism described earlier.

NBBS has the control mechanisms to provide ABR service on the same links as reserved traffic with QOS guarantees.

NBBS has the control mechanisms to provide ABR service on the same links as reserved traffic with QOS guarantees. Several mechanisms are provided to forward this objective: EARB flow control and the nonreserved delay priority at intermediate links, in addition to path selection and the reporting of measured utilization by the topology database algorithm. EARB acts as an admission control scheme for information flowing from "best-effort" connections into the network. Each EARB connection tries to gain a share of the bandwidth that is available on each link that it crosses. There is no connection-level control for best-effort service.

Each connection is controlled packet by packet, depending on the state of the current network connections as measured by the EARB algorithm. ABR traffic is placed in the nonreserved delay priority queue at the intermediate links. It has the lowest scheduling priority, and so the QOS of reserved connections is not affected. This does, however, have a significant benefit to the network, because ABR traffic is able to use bandwidth that is not used by the reserved classes of traffic. NBBS also provides information on the actual link utilizations in the topology database. The path selection algorithm can make use of such information in choosing a path through the network for an ABR connection and for load-balancing of the ABR traffic.

The current definition of ABR service allows a minimum cell rate (greater than or equal to zero) to be reserved for this type of traffic. This service guarantees some minimal throughput but allows the user to send more traffic into the network with the understanding that the additional traffic (beyond the guaranteed level) is best effort. This service is ideal for supporting frame relay. Frame relay offers a committed information rate (CIR) service that guarantees a level of throughput to the user. In addition, the user can send "excess" traffic into the network that is above the CIR, but with the understanding that the excess is subject to discard if the network becomes congested. In frame relay, the excess traffic is controlled by a traffic parameter called *excess burst* (denoted B_e).The B_e traffic is sent, but is marked "discard eligible," and may be delivered if there is enough spare capacity in the network at the moment. Traffic that is inserted beyond B_e has no service defined for it. It may be discarded immediately upon being received by the network.

NBBS makes a reservation for the minimum bandwidth specified by the connection. The flow into the network is then controlled by the EARB algorithm so that the source can send data over and above the minimum that has been reserved. EARB will not decrease the flow to any less than the minimum bandwidth reserved. Although throughput can be guaranteed in this case, strict delay and loss guarantees cannot be given.

Variable bit rate service. Variable bit rate (VBR) service has received a great deal of attention in the standards bodies and in the literature. This type of service can be used for data and for multimedia traffic. The standards bodies assume an "on-off" behavior of the source that can be characterized deterministically by a set of traffic descriptors (i.e., by a generic cell rate algorithm, or GCRA). As long as the traffic adheres to its descriptors, it is regarded as conforming, and the traffic receives the QOS guaranteed by the network. If the traffic violates the GCRA characterization, it is subject to policing action by the usage parameter control (UPC) of the network. It is a fundamental assumption of VBR service that the traffic descriptors do not change for the duration of the connection, or at least between traffic contract renegotiations at the UNI.

The NBBS traffic control mechanisms support the VBR service as defined in the standards. Leaky bucket parameters can be derived from the GCRA descriptors, and bandwidth allocations will be computed on the basis of them. Traffic that meets the GCRA descriptor has its QOS guaranteed. Traffic that violates the contract is subject to UPC actions. In addition, various additional features are available in NBBS. One of the three actions (or a combination of the three) is possible: first, packets or cells that violate the traffic contract by arriving too early can be queued and wait until they are in conformance; second, packets or cells violating the contract can be marked as having lower-loss priority; third, nonconforming traffic may simply be dropped. This flexibility allows the network provider many options for determining UPC actions for different classes of connections.

Enhanced variable bit rate service. Enhanced variable bit rate (VBR+) service has been proposed by several industry groups to the ATM Forum. The fundamental idea of VBR+ service is to add a dynamic conformance model to basic VBR service. The need for a dynamic traffic model becomes clear when multimedia connections or long-running data connections with highly variable traffic profiles are considered. These kinds of connections require stringent QOS guarantees, but at the same time, we would like to be as efficient as possible in allocating bandwidth. The bandwidth requirements for these types of connections vary over time. When it is determined that such a connection requires less bandwidth than it is currently allocated, the difference between the reserved and used bandwidth can be released along its end-to-end path. This would in turn allow more bandwidth to be available to new and already established connections. Similarly, if it is determined that a connection requires more bandwidth than it is currently allo-

cated, the service provided to the connection might degrade if this increased bandwidth demand is not met. Hence, the bandwidth allocated to connections in this type of service would vary over time, depending on their bandwidth requirements.

For VBR+ service in ATM, the source may specify a suitable sustainable cell rate (SCR) and a peak cell rate (PCR) at connection setup. The theory is that traffic may exceed the SCR specifications, but as long as it remains within the parameters of certain control information sent from the network to the user, the traffic receives the QOS negotiated at connection setup time.

One way to accomplish this function is to put the burden on the traffic source. The source could renegotiate the traffic contract each time the traffic changes significantly. The network side can also provide this service in a couple of different ways. One mechanism is an end-to-end flow control feedback mechanism rather than bandwidth allocation. The second mechanism uses network observation of the source and fast resource management (FRM) cells in order to control the source traffic profile between SCR and PCR. If the network does resource allocation, it can give strict delay and cell loss guarantees. A network could choose a combination of these approaches to provide VBR+ service, with renegotiation being initiated either by the network or by the user. This service is not yet defined in the standards.

NBBS has a very sophisticated and flexible bandwidth management function that can provide the kind of service described by VBR+. Given a set of initial traffic descriptors, the NBBS bandwidth management function will make an initial allocation of resources in the network. It then monitors the incoming traffic by measuring the mean rate and the fraction of packets or cells that arrive to find that the leaky bucket token pool is empty, i.e., the fraction of packets or cells that have to wait at the leaky bucket for entry when they arrive. With these measurements, the estimation and adaptation function is able to determine automatically whether the original (or current) bandwidth allocation needs updating.

By tracking the bandwidth requirements of the connection dynamically, the NBBS network can make an efficient allocation over time, save network resources, and allow a higher degree of multiplexing while providing QOS guarantees to connections.

In tracking the dynamic requirements of a connection, NBBS may find that the resources required increase or decrease. When the change is significant enough, NBBS will send a message to the intermediate nodes along the path of the connection and to the endpoint, requesting a resource allocation change. In the case of decreased allocation, the request can always be granted. In the case of an increased allocation, the intermediate nodes may not always be able to grant the request. In this case, several actions are possible. First, the network can attempt to reroute the connection on another path that can support the increased allocation. Second, if no feasible paths can support the request for additional resources, or if the connection did not request to be rerouted, the connection can be slowed down by increasing admission delay, or by feedback to the source, requesting that the source slow down. Third, if connection preemption priorities are defined in the network, a request for additional resources may result in lower priority connections being forced to give up their resources so that the new request can be satisfied.

NBBS can support the dynamic renegotiation of traffic descriptors by the source. However, the ability of the network to perform this function automatically is of great value. From the network provider's point of view, the reduction of resource does not depend upon the user signaling the request. Thus, the network provider can reduce costs and increase multiplexing (revenue) without end users becoming involved or even knowing about it (since QOS is maintained). From the network user's perspective, terminal equipment can be simpler since it need not have the capability to detect changes in the traffic characteristics. Also, the network user does not have to worry about the complex traffic characterization for multimedia or for other highly variable traffic sources.

Concluding remarks

Integrating different services with different traffic characteristics and service parameters in high-speed networks requires a comprehensive traffic management and congestion control framework.

To meet this challenge, NBBS includes an integrated set of procedures that include bandwidth computation and accounting, path selection algorithm, connection preemption and priority handling, policing and shaping, bandwidth adaptation, adaptive

rate-based congestion control, packet or cell discarding at the intermediate nodes, and so forth.

In this paper, we presented an overview of these algorithms, explaining how they operate and relate to each other and describing the functions they provide. These algorithms complement emerging high-speed networking standards and provide high utilization of network resources.

Cited references and note

1. The ATM Forum, *ATM User-Network Interface Specification: Version 3.1,* Prentice Hall, Englewood Cliffs, NJ (1994).
2. R. Guérin, H. Ahmadi, and M. Naghshineh, "Equivalent Capacity and Its Application to Bandwidth Allocation in High-Speed Networks," *IEEE Journal on Selected Areas in Communications* **SAC-9**, No. 7, 968–981 (September 1991).
3. T. E. Tedijanto, R. O. Onvural, D. C. Verma, L. Gün, and R. A. Guérin, "NBBS Path Selection Framework," *IBM Systems Journal* **34**, No. 4, 629–639 (1995, this issue).
4. N. Budhiraja, M. Gopal, M. Gupta, E. A. Hervatic, S. J. Nadas, P. A. Stirpe, L. A. Tomek, and D. C. Verma, "The NBBS Access Node," *IBM Systems Journal* **34**, No. 4, 694–704 (1995, this issue).
5. D. Bertsekas and R. Gallager, *Data Networks*, 2nd Edition, Prentice-Hall, Inc., Englewood Cliffs, NJ (1992).
6. A. Girard, *Routing and Dimensioning in Circuit-Switched Networks*, Addison-Wesley Publishing Co., Inc., Reading, MA (1990).
7. G. R. Ash, R. H. Caldwell, and R. P. Murray, "Design and Optimization of Networks with Dynamic Routing," *Bell Systems Technical Journal (B.S.T.J.)* **60**, No. 8, 1787–1820 (October 1981).
8. T. J. Ott and K. R. Krishnan, "State Dependent Routing of Telephone Traffic and the Use of Separable Routing Schemes," *Proceedings of the 11th International Teletraffic Congress,* Kyoto, Japan, M. Akiyama, Editor, Elsevier Science Publishers B.V. (North-Holland) (September 1985), pp. 5.1A-5.1–5.1A-5.6.
9. R. J. Gibbens, F. P. Kelly, and P. B. Key, "Dynamic Alternative Routing—Modelling and Behaviour," *Proceedings of the 12th International Teletraffic Congress*, Torino, Italy, M. Bonatti, Editor, Elsevier Science Publishers B.V. (North-Holland) (June 1988), pp. 1019–1025.
10. F. P. Kelly, "Routing in Circuit-Switched Networks: Optimization, Shadow Prices and Decentralization," *Advances in Applied Probability* **20**, No. 1, 112–144 (1988).
11. D. Mitra, R. J. Gibbens, and B. D. Huang, "Analysis and Optimal Design of Aggregated Least-Busy-Alternative Routing on Symmetric Loss Networks with Trunk Reservation," *Proceedings of the 13th International Teletraffic Congress*, Copenhagen, Denmark, A. Jensen and V. B. Iversen, Editors, Elsevier Science Publishers B.V. (North-Holland) (June 1991), pp. 477–482.
12. J. M. Akinpelu, "The Overload Performance of Engineered Networks with Nonhierarchical and Hierarchical Routing," *Bell Systems Technical Journal (B.S.T.J.)* **63**, No. 7, 1261–1281 (1984).
13. G. R. Ash, "Use of a Trunk Status Map for Real-Time DNHR," *Proceedings of the 11th International Teletraffic Congress*, Kyoto, Japan, M. Akiyama, Editor, Elsevier Science Publishers B.V. (North-Holland) (September 1985), pp. 4.4A-4.1–4.4A-4.7.
14. R. S. Krupp, "Stabilization of Alternate Routing Networks," *Proceedings of ICC'82*, Philadelphia, PA (June 1982), pp. 31.2.1–31.2.5.
15. E. W. M. Wong and T.-S. Yum, "Maximum Free Circuit Routing in Circuit-Switched Networks," *Proceedings of Infocom'90*, San Francisco (June 1990), pp. 934–937.
16. S. A. Owen, "NBBS Network Management," *IBM Systems Journal* **34**, No. 4, 725–750 (1995, this issue).
17. L. Gün and R. Guérin, "Bandwidth Management and Congestion Control Framework of the Broadband Network Architecture," *Computer Networks and ISDN Systems* **26**, No. 1, 61–78 (1993).
18. L. Gün, "An Approximation Method for Capturing Complex Traffic Behavior in High Speed Networks," *Performance Evaluation, Special Issue on Bandwidth Management and Congestion Control in High Speed Networks* (1993).
19. N. Yin and M. Hluchyj, "On Closed-Loop Rate Control for ATM Cell Relay Networks," *Proceedings of IEEE INFOCOM*, Vol. 1, Toronto (June 1994), pp. 99–108.
20. P. Newman, "Backward Explicit Congestion Notification for ATM Local Area Networks," *Proceedings of IEEE GLOBECOM*, Vol. 2 (December 1993), pp. 719–723.
21. $F(i)$ does not have an asymptote, but the increasing slopes reduce dramatically for larger i.
22. D. Chiu and R. Jain, "Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks," *Computer Networks and ISDN Systems* **17**, 30–47 (1989).
23. The difference between the effects of dropping more significant bits and the effects of dropping less significant bits depends on the characteristics of the coder employed. It should also be noted that similar considerations apply to multilayer video coders and connections carrying video traffic.
24. V. G. Kulkarni, L. Gün, and P. F. Chimento, "Effective Bandwidth Vectors for Multiclass Traffic Multiplexed in a Partitioned Buffer," submitted to *IEEE Journal on Selected Areas in Communications* (1995).

Accepted for publication June 6, 1995.

Hamid Ahmadi *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 704, Yorktown Heights, New York 10598 (electronic mail: hamid@watson.ibm.com).* Dr. Ahmadi received his B.S., M.S, and Ph.D. degrees in electrical engineering from Columbia University in 1976, 1978, and 1983, respectively. He joined the IBM Thomas J. Watson Research Center in 1984 as a member of the Telecommunication Systems department. He is currently senior manager of the Communications Networks department, responsible for research and systems projects on wireless and mobile communication networks, multimedia desktop personal conferencing, network security, and open system networks architecture. Before that he was managing a research group working on wireless and mobile communications systems and architecture for wireless LANs. He spent a two-year assignment at IBM Research in Zurich, Switzerland, during 1986-1987, working on fast packet switching and ATM network architecture. Prior to joining IBM, from 1980 to 1984, he was with the Switching and Signaling Systems department at Bell Laboratories, Holmdel, New Jersey, where he was involved in the area of performance analysis and

protocol studies of signaling networks. Dr. Ahmadi is the Editor-in-Chief of the *IEEE Personal Communications Magazine*, an editorial member of the *International Journal of Wireless Networks*, and a technical editor of the *IEEE Transactions on Communications*. Since 1988, he has been an adjunct professor in the graduate center at Polytechnic University in New York, where he teaches graduate courses in communication networks and performance modeling.

Phillip F. Chimento *IBM Networking Hardware Division, P.O. Box 12195, Research Triangle Park, North Carolina 27709.* Dr. Chimento received the A.B. degree in philosophy from Kenyon College in 1972, the M.S. degree in computer science from Michigan State University in 1978, and the Ph.D. degree in computer science from Duke University in 1988. He worked for IBM from 1978 to 1994, holding various positions in design, development, test, and architecture. Most recently, he was a member of the core team that developed IBM's Networking BroadBand Services architecture for high-speed packet and cell switching. In 1994, Dr. Chimento took a leave of absence from IBM to accept a visiting faculty position at the University of Twente in the Netherlands. There, as a member of the Centre for Telematics and Information Technology (CTIT) and the Tele-Informatics and Open Systems (TIOS) group, he is working on B-ISDN signaling and resource allocation issues and participating in Dutch and European telecommunications projects. He has had papers published in *IEEE Transactions on Computers, Operations Research*, and various conferences. He is a senior member of the IEEE and a member of the ACM and ORSA (INFORMS).

Roch A. Guérin *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 704, Yorktown Heights, New York 10598 (electronic mail: guerin@watson.ibm.com).* Dr. Guérin received the Diplome d'Ingénieur from the École Nationale Supérieure des Télécommunications, Paris, France, in 1983, and his M.S. and Ph.D. from the California Institute of Technology, both in electrical engineering, in 1984 and 1986, respectively. Since August 1986 he has been with IBM at the Thomas J. Watson Research Center, where he now manages the Broadband Networking group in the Advanced Networking Laboratory. His current research interests are in the areas of modeling, architecture, and quality-of-service issues in high-speed networks. In particular, he is interested in developing techniques to map network-level performance measures to corresponding quantities at the application level, and in understanding issues related to the interactions between different networking technologies and protocols. He is also interested in understanding how the new capabilities and flexibility available from high-speed networks can translate into better services to applications. Dr. Guérin is a member of Sigma Xi and the IEEE Communications Society, and is an editor for the *IEEE/ACM Transactions on Networking.* He was an editor for the *IEEE Transactions on Communications* and the *IEEE Communications Magazine.*

Levent Gün *IBM Networking Hardware Division, P.O. Box 12195, Research Triangle Park, North Carolina 27709.* Dr. Gün received a B.A. in mathematics and a B.S. in electrical engineering from Bogazici University, Turkey, in 1983. He earned M.S. and Ph.D. degrees in electrical engineering from the University of Maryland in 1986 and 1989, respectively. While a member of the IBM family (1989–1994), Dr. Gün worked in the Networking Architecture group at Research Triangle Park. In addition, he held the position of Adjunct Assistant Professor in the Operations Research Department at the University of North Carolina, Chapel Hill. His interest continues in the development and analysis of high-speed networking architectures. He has published over a dozen papers in various journals and conference proceedings in the areas of computational probability, queuing theory, and stochastic control. He is an active member of IEEE and ORSA.

Bouchung Lin *IBM Networking Hardware Division, P.O. Box 12195, Research Triangle Park, North Carolina 27709.* Dr. Lin received M.S. and Ph.D. degrees in electrical and computer engineering from North Carolina State University in 1985 and 1989, respectively. While a member of the IBM family (1989–1995), he first worked on performance and system issues on the FDDI, Bridge, and Router at Research Triangle Park. In 1993, he became a member of the Networking Architecture group working on the high-speed packet-switching network architecture, specifically in the area of traffic management. His current research interests are in the design and development of the hybrid fiber/coax network to provide new services, data, and telephony with the ATM technology. Dr. Lin is a member of IEEE and is also a voting member of the IEEE 802.14 SWG.

Raif O. Onvural *IBM Networking Hardware Division, P.O. Box 12195, Research Triangle Park, North Carolina 27709 (electronic mail: onvural@vnet.ibm.com).* Dr. Onvural is a senior engineer at IBM's Research Triangle Park facility in the Networking Architecture organization and manages the Networking Technology Architecture department that develops network control services for ATM networks. He is also IBM's venue owner for the ATM Forum and has been attending the forum meetings since February 1993. Dr. Onvural organized several international conferences on high-speed networks, in general, and ATM networks, in particular. He has published in various journals and conferences, and has edited five books. He is also the author of the book *Asynchronous Transfer Mode Networks: Performance Issues*.

Theodore E. Tedijanto *IBM Networking Hardware Division, P.O. Box 12195, Research Triangle Park, North Carolina 27709.* Dr. Tedijanto received B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Maryland in 1984, 1986, and 1990, respectively. While a member of the IBM family (1990–1995), he worked in the Networking Architecture group at Research Triangle Park. His area of interest includes traffic management and route selection algorithms for high-speed networking architectures. Dr. Tedijanto is an active member of IEEE and continues as a member of the ATM Forum, working on ATM routing and traffic management. He has published a number of papers in various journals and conference proceedings in the areas of queuing theory and bandwidth management.

Reprint Order No. G321-5584.