Overview of the NBBS architecture

by G. A. Marin
C. P. Immanuel
P. F. Chimento
I. S. Gopal

Networking BroadBand Services (NBBS) architecture has been developed by IBM to meet the requirements of high-speed networking. These include addressing problems caused by emerging multimedia applications with extremely high bandwidth requirements and quality-of-service guarantees, as well as those of supporting a variety of existing protocols and interfaces. This paper offers an overview of NBBS and provides insight into the technology and structure of the architecture. Included is a synopsis of the driving forces behind the architecture and a summary of how the challenges of this new environment are addressed.

In 1987 the CCITT (International Telegraph and Telephone Consultative Committee) selected asynchronous transfer mode (ATM) to be the architecture of the future Broadband Integrated Services Digital Network (B-ISDN). This choice reflected a recognition among the world's experts on broadband communications that ATM has the characteristics needed to accommodate future (even unanticipated) needs for communication services. The principal reasons for the choice of ATM are now well known[1,2] and include:

- Flexible use of bandwidth. Instead of making bandwidth available only as a multiple of a fixed basic rate (such as 64 kilobits per second [Kbps] as in ISDN technology), ATM offers complete flexibility in assigning cells (53-byte ATM packets) to user connections.
- Support of multiple types of transmission services. An ATM network is able to accommodate voice, video, and data services with their corresponding (greatly differing) traffic characteristics.
- Advantages of common infrastructure. The vision of B-ISDN is that of a single, universal network infrastructure that reduces costs through economies of scale, reduces complexity through unified networking, and enables products from multiple vendors to interoperate effectively.

At the same time, researchers at IBM's Thomas J. Watson Research Laboratory were beginning to work on the ideas and algorithms that ultimately led to IBM's Networking BroadBand Services (NBBS) architecture. They were motivated to find an integrated approach for building a high-speed packet-switched network that realized the advantages being sought for ATM on the world stage. Whereas the work in the ATM standards world initially concentrated on defining the physical layer, ATM cell layer, and ATM adaptation layer, IBM research concentrated on developing the network control algorithms needed to provide the kind of service and flexibility that is the promise of ATM. Beginning in 1989, architects and developers from IBM's Networking Hardware Division (NHD) joined the effort, and together NHD and the Research Division have built a comprehensive architecture for high-speed, multimedia networking. The architecture not only enables an efficient ATM service in wide area networks, but also offers customers flex-

0018-8670/95/$3.00 © 1995 IBM

ibility in moving from the voice and data networks they use today to the integrated networking they may choose tomorrow.[3]

Many of the NBBS concepts were prototyped in the context of a series of field trials conducted by IBM. We implemented an NBBS control point on a RISC System/6000* (RS/6000*) processor attached to the high-speed plaNET switch.[4] The plaNET switch was deployed in various field trials including the AURORA trial sponsored by the National Science Foundation (NSF) and the Advanced Research Projects Agency (ARPA).[5] Many of the NBBS algorithms were experimentally tested during these trials, resulting in modifications and optimizations that have been fed back into the product-level code.

NBBS customer value. As a major new architecture from IBM, NBBS is intended to provide significant value to IBM's customers, including:

- Guaranteed quality of service (QOS). NBBS enables a network to carry new kinds of traffic, for example, packetized voice and video along with data traffic, while guaranteeing the appropriate QOS for each type of traffic.
- Integrated subnetwork technologies. A single NBBS-controlled network can offer multiple subnetwork interfaces. These include frame-relay bearer service, the ATM user-to-network interface, simple high-level data link control (HDLC), and circuit emulation services. This gives customers maximum flexibility in evolving today's networks toward ATM.
- Bandwidth efficiency. NBBS bandwidth management can save customers an estimated 20 to 50 percent of recurring bandwidth costs. Actual customer savings will depend on network topology, traffic patterns, traffic characteristics, and on the structure of network tariffs and offerings. Nevertheless, NBBS will give many customers the capability to lower their costs by provisioning their private networks with fewer expensive links. Alternatively, NBBS offers service providers the opportunity to operate their networks much more efficiently and thus to achieve a competitive price advantage.
- Set management. Whether for building virtual private local area networks (LANs), for supporting video conferencing, for providing video-on-demand, or for any other use of groups whose members have some affinity, NBBS provides a powerful capability to create and manage groups, distribute information efficiently to group members, and maintain connectivity even in the face of link failures. The key to this flexibility is NBBS set management.
- Nondisruptive path switching (NDPS). NBBS will automatically reroute a network connection between endpoints to a different path through the network in case of node or link failures. The architecture minimizes disruption to users and delivers packets in sequence.
- Call preemption. The call preemption protocol enables the network to provide different levels of network availability to different classes of users. When required, NBBS will reroute existing connections to accommodate higher-priority connections using NDPS. Coupled with NBBS path selection algorithms, this capability gives NBBS networks a state-of-the-art load-balancing capability.

This paper begins by describing the principal objectives behind the development of NBBS and the key features of the architecture that serve each particular objective. Next is a high-level description of the NBBS architecture structure, followed by a description of how NBBS features and components interoperate to deliver customer value. Through this approach the paper introduces the reader to the relationships among many of the functions described in the subsequent papers in this issue.

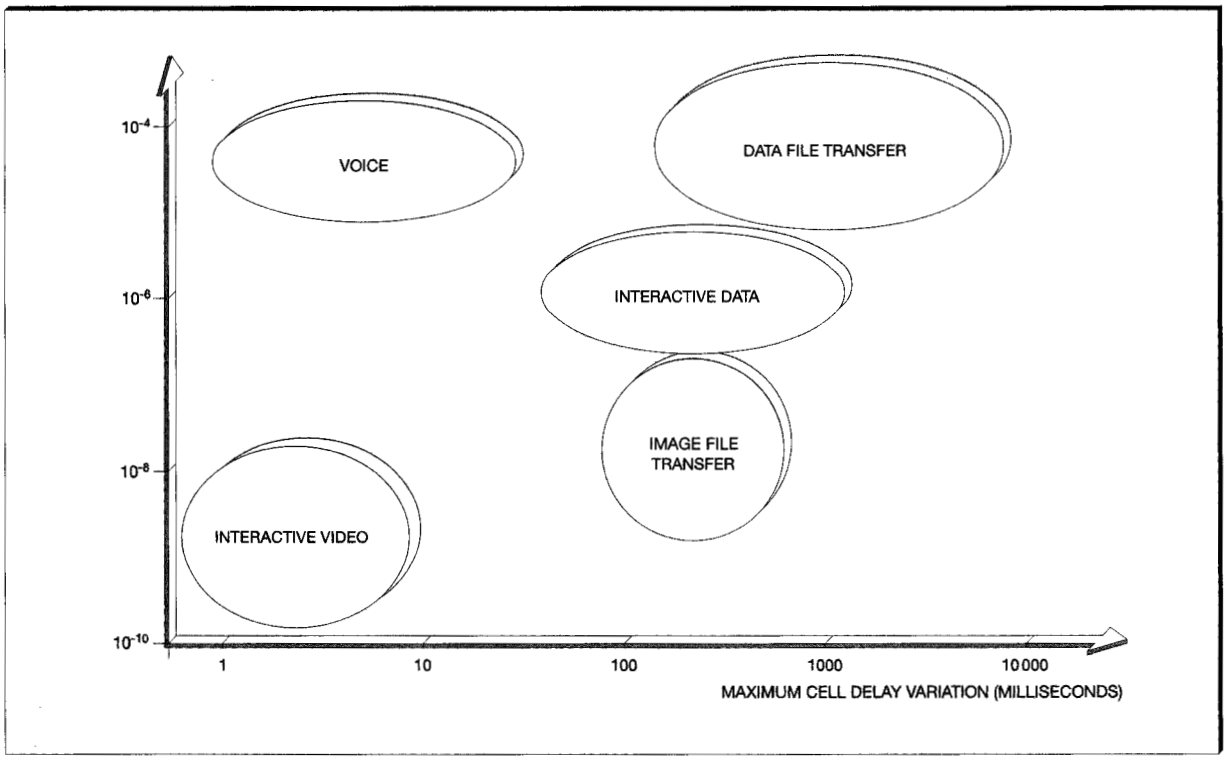
Providing quality-of-service guarantees

Within the framework of ATM standards, QOS implies that a user can request guarantees that the network will provide the service that is contracted at the user-to-network interface (UNI). Just as IBM communication architectures have been traditionally, ATM is *connection oriented*. However, it is extremely important to observe that an ATM network can offer guarantees for the mean data rate, the delay, the delay jitter (variance), and cell loss ratio of user connections.

Oversimplifying somewhat, there are two basic ways to provide QOS guarantees. The first is to provision the network in such a way that all users get the service they require. The second is to reserve certain network resources for each user, as needed, to guarantee the requested QOS for the user connection.

Today's data networks follow the first approach. In the case of IBM products implementing Systems Network Architecture (SNA) or Advanced Peer-to-

Figure 1 QOS requirements in packet-switched networks

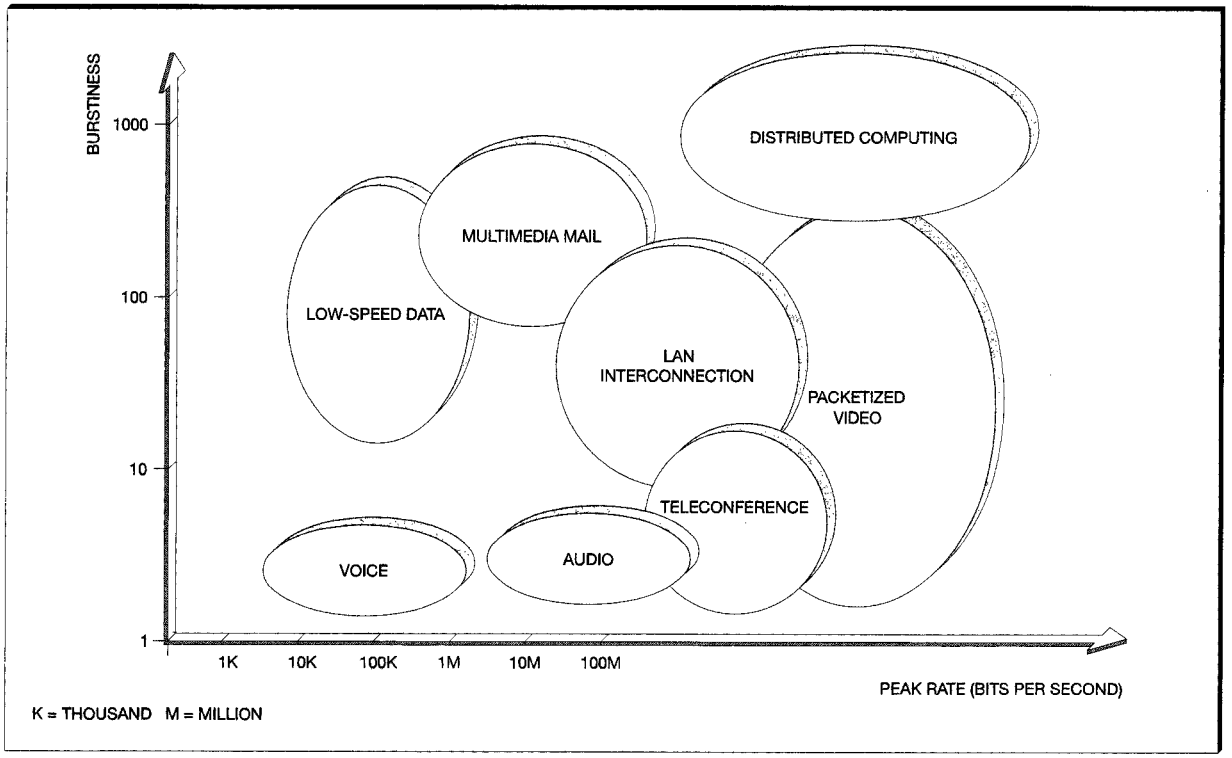


Peer Networking* (APPN*), for example, customers work with their account representatives and often with the SNAPSHOT performance group in Raleigh, North Carolina, to estimate their traffic requirements, model their network, and evaluate the effects of proposed changes on performance. Nodes and trunks in the network are typically sized to meet an anticipated average delay for interactive traffic (perhaps a half-second) and to ensure that delay is not unacceptable during busy periods. Any user with network access will be accommodated when he or she requests a connection. Data packets will queue at network nodes as they move through the network, and their end-to-end delay may vary greatly depending on the overall network load. (Of course, in SNA and APPN, network connections are assigned priorities based on their class of service—a somewhat different notion than the ATM QOS.[6]

In telephone networks, this data networking approach would not work and is not used. Telephone networks were designed for voice traffic. They too are connection oriented, but the connections generally persist for a short period (about three minutes on the average) and the network must be able to guarantee that delay in crossing the network is tightly controlled. Commonly, voice is supported in today's networks using a synchronous transfer mode in which clocking at both ends of a connection is used to insert or remove voice samples from a time slot that is reserved for the connection.

NBBS has been designed to support all types of traffic in a single network. This includes data, voice, LAN interconnection, image, full-motion video, and even traffic that we cannot anticipate today. Each type of traffic is more or less tolerant of cell loss, of delay, and of jitter (or cell delay variation). Each also offers its own characteristics in terms of required mean, burst and idle period characteristics, and peak rate requirements. Figure 1 illustrates the variation in delay and packet loss probability that a network must support, depending on the type of traffic. Figure 2 illustrates that the characteristics of the arriving traffic may vary significantly, thus

Figure 2 Characteristics of input traffic



complicating the challenge of guaranteeing the QOS requirements illustrated in Figure 1. "Burstiness" is a measure of variance in the rate that cells are transmitted. In Figure 2 we see that distributed computing is about 1000 times more bursty than voice.

NBBS actually supports both nonreserved-bandwidth connections and reserved-bandwidth connections in order to satisfy the full spectrum of traffic requirements. The architecture is connection oriented and includes admission control algorithms that will accept a new reserved connection only if resources are available in the network to establish the connection with the required QOS characteristics.

Accommodating traffic with varied characteristics

If the network must provide QOS guarantees to users, then it must use reserved-bandwidth connections. For such connections the question becomes: what does the network reserve and how much of it per connection? Essentially, the network must ensure that there are sufficient buffers available in each node in the network for whatever (tightly controlled) queuing must occur; and the network must also reserve part of the capacity of each link that is on the end-to-end path of the connection. Some applications and voice connections generate traffic at a constant bit rate. Others generate traffic at a rate that may vary both widely and abruptly. For example, a dormant file server may awaken in response to an incoming request, send out a large block of data, and then become dormant again. In the same way, other applications may generate traffic in bursts. That is, their peak rate may be much higher than their average rate of traffic generation. These applications are generally referred to as variable bit rate (VBR) sources.

Figure 3 illustrates input to the network that might be associated with a bursty VBR traffic source. We characterize the connection by the peak rate, the long-term mean rate, and the statistics of the burst

Figure 3 Variable bit rate traffic

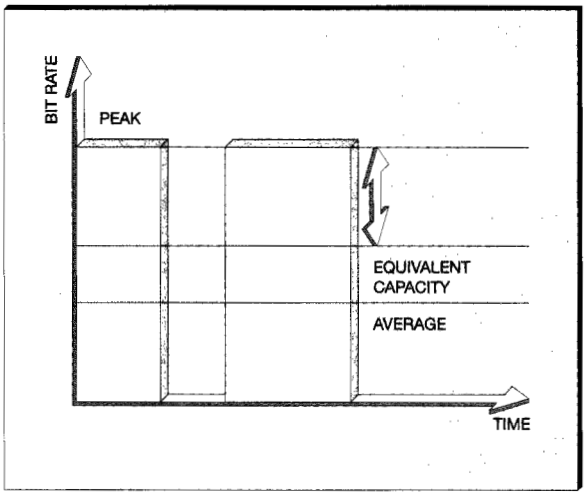
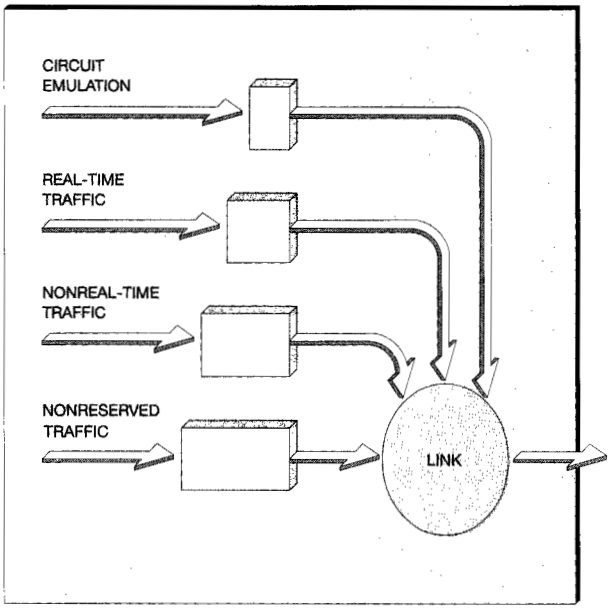


Figure 4 The four priority queues



and idle periods. The question now becomes, for a VBR source like the one illustrated: just how much bandwidth should the network reserve? NBBS answers this question with the concept of equivalent capacity, described in more detail in the accompanying paper on traffic management.[7] At a conceptual level, one can think of the bursts of traffic as being water that is being poured intermittently into a funnel. The rate at which the funnel admits the water into the system is determined by the radius of the opening at the bottom of the funnel. Of course, the QOS guarantees could easily be met by reserving the peak rate that is associated with the connection in Figure 3, but that could result in the network reserving many times the bandwidth the connection really needs to meet its requirements.

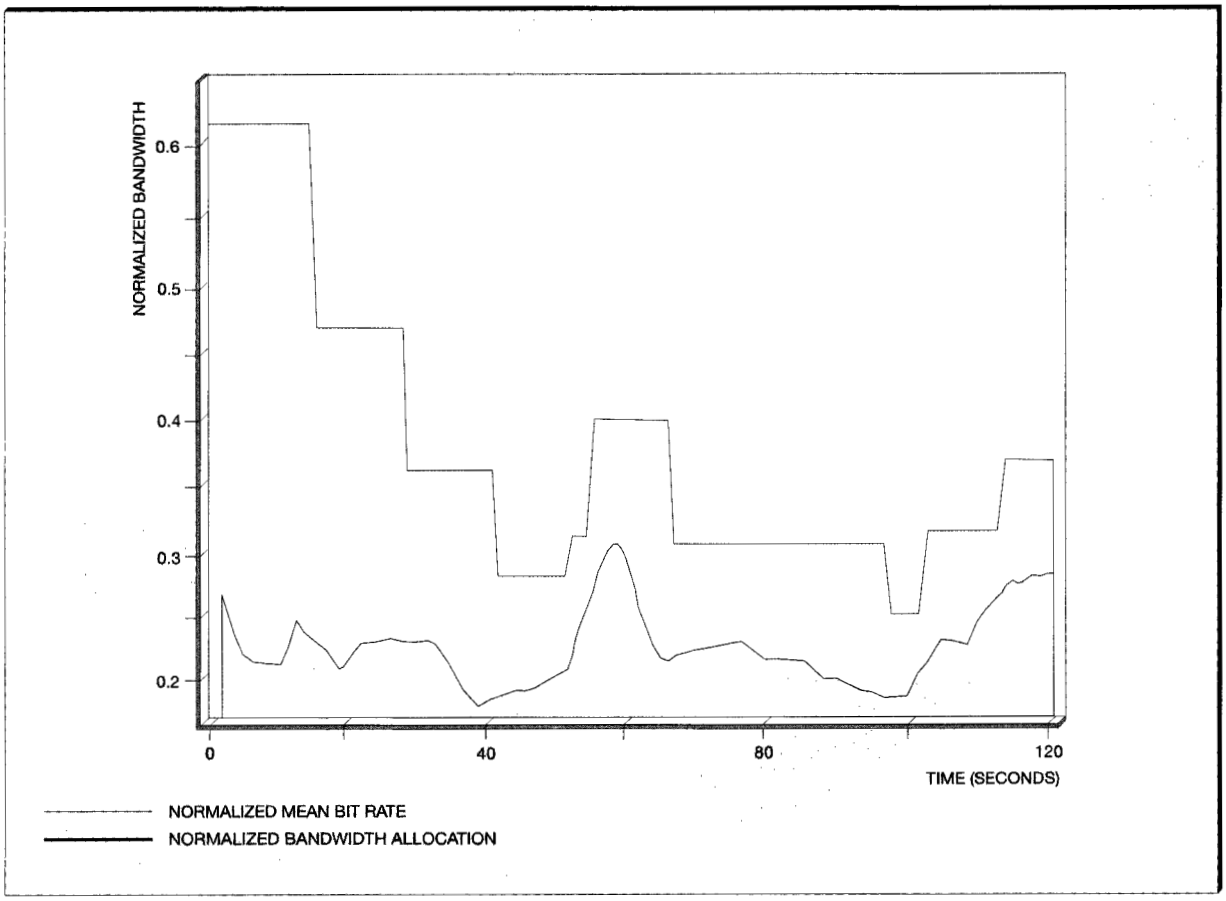
The calculation of equivalent capacity is based on the traffic metrics associated with the connection. The mean, burst, and average rates are used to calculate equivalent capacity.

To satisfy the different requirements of both reserved and nonreserved connections, each NBBS network node has multiple queues associated with each outbound trunk (a link from one network node to an adjacent network node that carries many connections). Figure 4 illustrates the four queues. The first two have very stringent delay requirements. The third queue is for nonreal-time but still reserved traffic that has more relaxed end-to-end delay requirements. (It could be used for interactive data session traffic.) The fourth queue is for nonreserved traffic. Each NBBS connection is associated with one of these buffers. The size of the buffer enters into the computation of equivalent capacity and is related to the maximum queuing delay that can be tolerated in a single node for the particular traffic class. To be more specific, equivalent capacity depends not only on the size of the buffers but also on the acceptable cell-loss ratio. Typically the resource reservation would guarantee that no more than one cell in 10 million will be discarded due to congestion in a node somewhere in the middle of the network.

Efficient use of network bandwidth

In addition to computing just the right amount of bandwidth for a given bursty connection, NBBS provides for statistical multiplexing of its connections on the network trunks and has the capability to adjust the amount of bandwidth reserved on a particular connection when connection requirements are seen to have changed. Statistical multiplexing can be used when the resources required for a given connection are small enough that many such connections can be carried on a given network trunk. Thus, described simply, NBBS computes the equivalent capacity when a new connection request is received; then it actually reserves the smaller of

Figure 5 A variable-bit-rate video trace of an auction

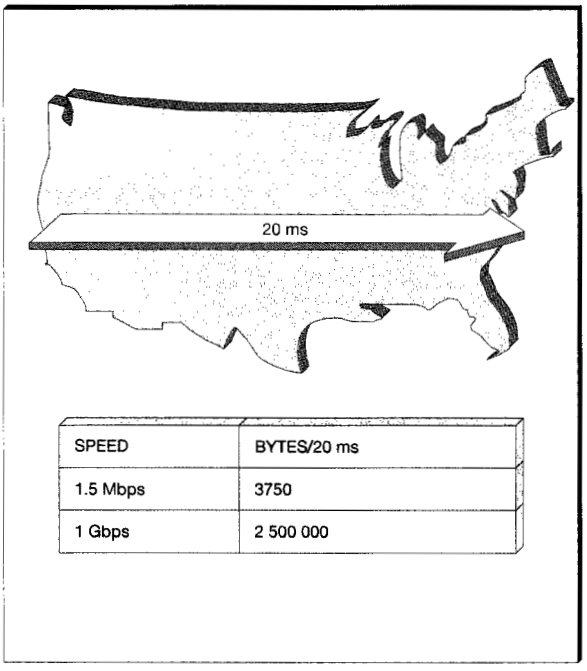


the equivalent capacity and the amount of bandwidth that is predicted to be needed by one of N connections sharing a trunk, governed by the law of large numbers.[8]

Bandwidth adaptation is an important capability that deserves an introduction here and is more fully described in the paper on traffic management.[7] In summary, this module provides the necessary functions to adjust the bandwidth reservation levels in response to the changes in the rate of incoming traffic. This function includes traffic monitoring, bandwidth estimation, and bandwidth reservation adjustment, all being done in real time. Bandwidth reservation is made on the basis of calculated equivalent capacity. If sufficient bandwidth is not available for a connection after connection establishment, several options are available, including the use of nondisruptive path switching and that of preempting lower-priority users.

The equivalent capacity reserved for a given connection is monitored through use of the "leaky bucket" mechanism, defined in the implementation agreements of the ATM Forum.[9] Somewhat like the funnel of our previous metaphor, the leaky bucket mechanism admits bursts of packets or cells into the network and enforces the bandwidth reservation contract that exists for the given connection. IBM's implementation can provide some added value over the standards and forum agreements. For example, the IBM leaky bucket has a provision to allow the user to exceed a reservation by a small percentage. Packets that are accepted in this way can be "flagged" and will be more likely to be discarded if a network node downstream becomes

Figure 6 Example of data in flight at high speeds



SPEED	BYTES/20 ms
1.5 Mbps	3750
1 Gbps	2 500 000

congested. However, a connection whose characteristics are changing more fundamentally could have difficulty in getting satisfactory service from a reserved-bandwidth connection based on fixed bandwidth.

Having recognized early in the development process that reserving a static amount of bandwidth (equivalent capacity) for an interactive data session was overly limiting, the NBBS team has included bandwidth adaptation as an option for users. The adaptation algorithm works on a tunable time scale (generally on the order of a second) by monitoring the behavior of a queue in the leaky bucket implementation. If the queue tends to be too full, the reserved bandwidth can be increased; if the queue tends to be empty, the reserved bandwidth can be decreased. This capability greatly mitigates the problem of trying to make an accurate initial estimate of the traffic characteristics associated with the connection.

Figure 5 shows the results of the NBBS bandwidth management approach when applied to actual data extracted from a video of an auction. The representation of the data stream has been smoothed to show one-second averages because the burstiness of the actual measurements makes graphical representation difficult. The data represent VBR data out of a compression/decompression (CODEC) adapter running the digital video interactive (DVI) compression algorithm. A reasonable approach to reserving peak bandwidth (if NBBS were not available) would be to set up the connection as if every frame contained the amount of data found in the largest frame from this video. That approach leads to a peak bandwidth reservation of 7.072 32 megabits per second. The figure shows the bandwidth that is actually reserved using the NBBS bandwidth estimation and adaptation algorithms. If bandwidth adaptation was not available, this would be the bandwidth reserved for the entire length of the connection and the curve showing bandwidth allocation would be a straight line. The curve showing normalized bandwidth allocation is the equivalent capacity normalized to peak-rate allocation. The parameters have been set so that bandwidth adaptation works quite conservatively (as can be seen by the distance between the actual data graph and the adaptation graph). Nevertheless, the NBBS approach results in saving 60 percent of the bandwidth that would be needed using peak reservation.

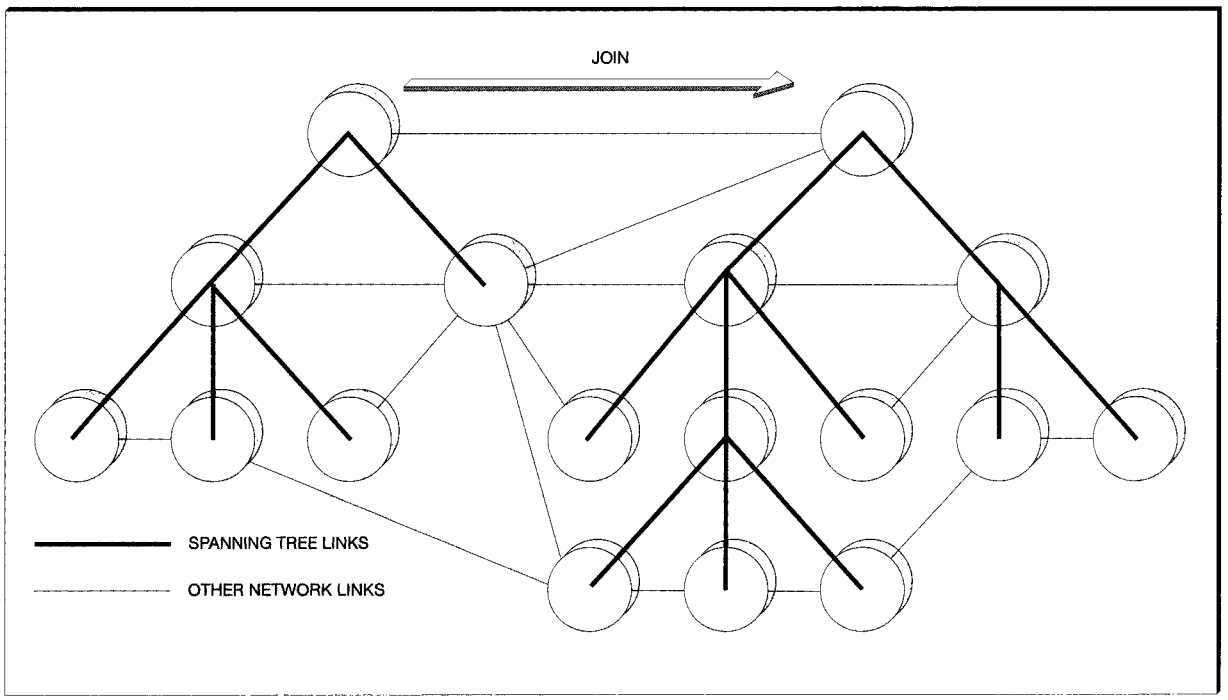
The actual saving a network will experience depends on the network topology, the traffic characteristics, and the tariff structure of the network. For wide-area links, IBM expects customers to be able to save 20 to 50 percent of the recurring cost of bandwidth in any private network based on NBBS technology. Alternatively, the NBBS bandwidth management services can increase the effective capacity of wide-area links or campus links from 20 to 50 percent, and thereby save the capital expense of upgrading to higher link speeds as network traffic grows.

The first implementation of NBBS technology by IBM's Networking Hardware Division is the IBM 2220 Nways* BroadBand Switch.[10]

Keeping control at gigabit speeds

The robust bandwidth management approach embodied in NBBS evolved from a recognition that a networking paradigm shift is really needed to accommodate voice, video, and data in a single network with appropriate QOS guarantees in each case. Early on, the NBBS team also recognized the challenges presented in controlling a network whose

Figure 7 CP spanning tree structure



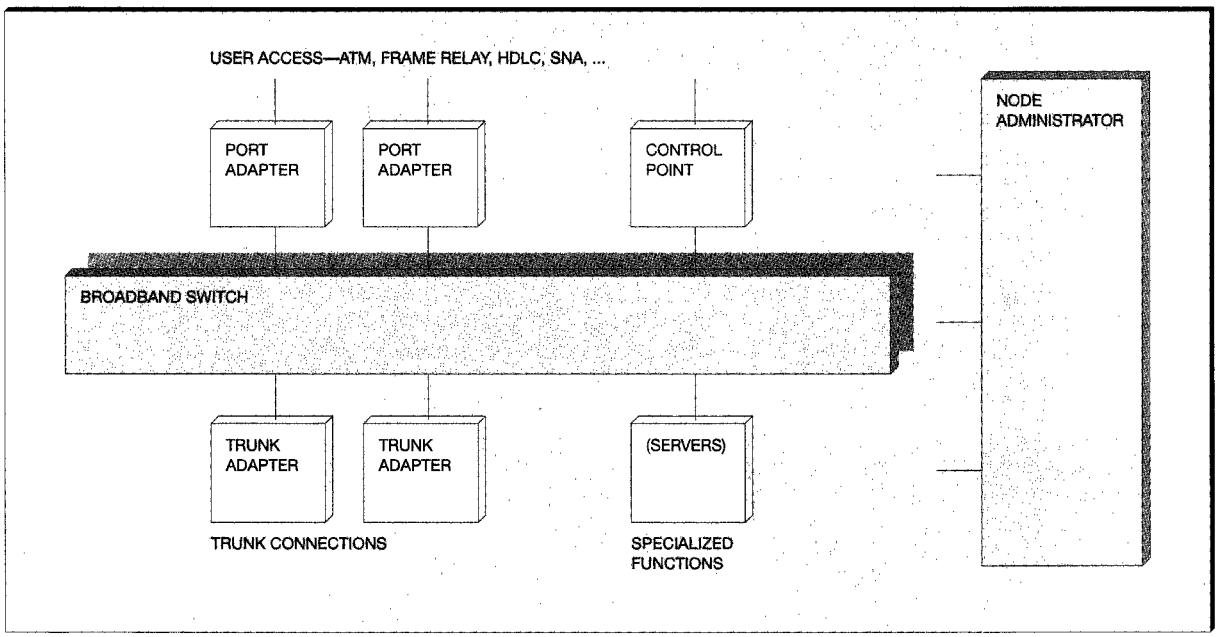
trunks could eventually reach gigabit speeds. As a result, NBBS is designed to address those challenges as well.

Figure 6 illustrates a network connection between the East Coast and the West Coast of the United States. The one-way propagation delay (time to move one bit over the distance shown) would be on the order of 20 milliseconds (ms). A network operating at 1.5 megabits per second (Mbps) could have as many as 3750 bytes "in-flight" during this propagation-delay interval. On the other hand, a network using trunks that accommodate 1 gigabit per second (Gbps) could have 2.5 million bytes in flight! Any problem that arises in the network—congestion or a link failure for example—is intensified by the requirement to keep up with such traffic rates. It is necessary, therefore, for high-bandwidth protocol to perform preventive congestion control and to have a mechanism for the rapid dissemination of control information.

NBBS was designed to scale to such high-speed links. This design point drove two fundamental decisions. First, the network relies on preventive congestion control. Second, the network has mechanisms that allow control information to be distributed to all network nodes at the fastest possible speed. That is, network control is distributed across the network at "propagation speeds," i.e., limited only by propagation delay.

NBBS uses reserved-bandwidth connections to provide QOS guarantees to connections where real-time or interactive performance is required. The network reserves bandwidth for such connections and uses the leaky-bucket mechanism as the fundamental mechanism that prevents congestion from occurring. The bandwidth management algorithms are used to determine the required equivalent capacity for connections and reservations are made based on the equivalent capacity. The bandwidth adaptation mechanisms will adjust according to the rate of incoming traffic on the access link. Thus, each node in the reserved-bandwidth connection—each intermediate node, and the exit point—has agreed on the parameters that govern the connection. If congestion does occur in the network, an intermediate node will discard excess packets. However, the NBBS bandwidth manage-

Figure 8 Functional decomposition of the 2220 Nways switch (NBBS node)



ment procedures guarantee this happens with extremely low probability. For nonreserved bandwidth connections, NBBS uses a congestion control mechanism called the *enhanced adaptive rate-based* (EARB) algorithm, described in the paper on bandwidth management.[7] Essentially this algorithm allows the entry and exit ports to exchange information on the rate at which nonreserved traffic is traversing the network. Based on information sent from the receiving node, the EARB algorithm adjusts the transmission rate at the entry port.

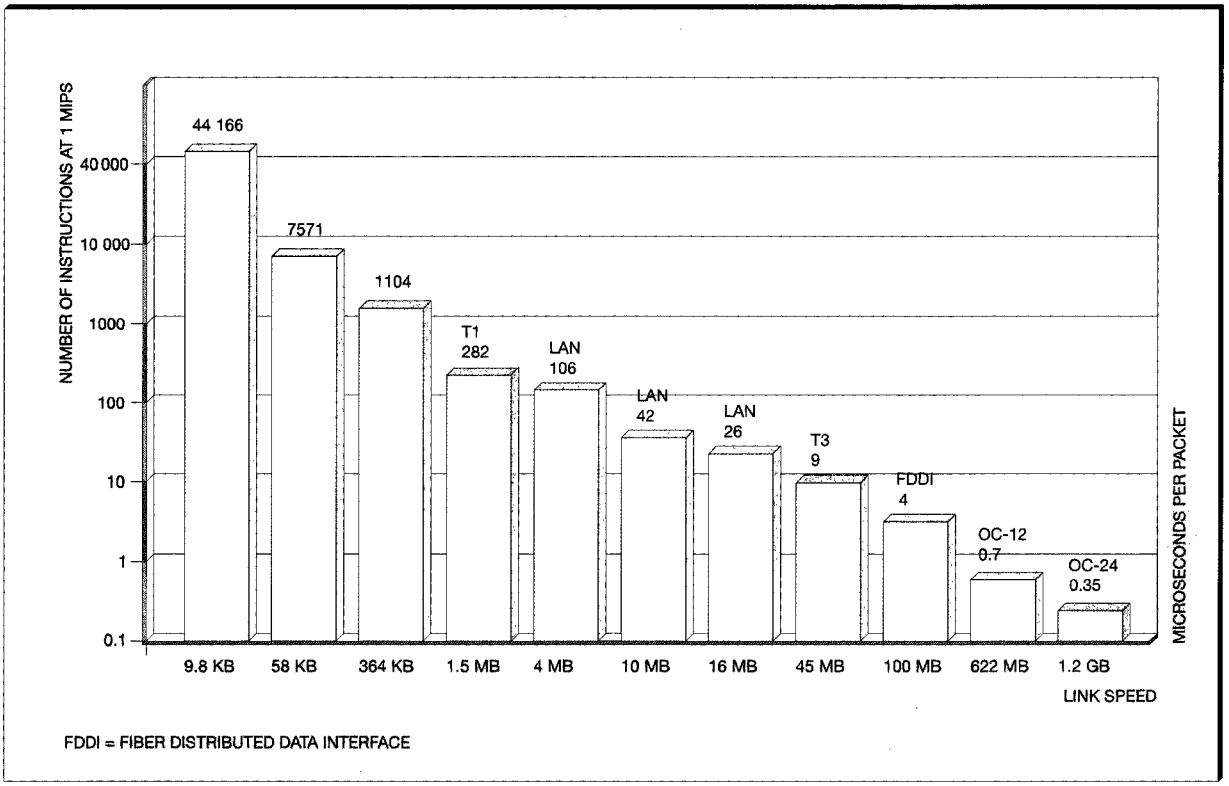
The control point (CP) spanning tree is the mechanism that NBBS uses to get control information distributed to every node in the network essentially at propagation speed. (NBBS keeps delays within the network node much lower than propagation delay, as we discuss later.)

The CP spanning tree algorithm is a distributed algorithm that builds and maintains the CP spanning tree. Figure 7 illustrates the CP spanning tree. This is described in more detail in the section on how a node joins the NBBS network.

In order for the CP spanning tree to work as intended, the routing of control traffic across the tree must be extremely efficient. From the beginning, the architects and 2220 Nways switch product developers worked closely together to assure this would happen. Essentially, all routing is done in hardware in IBM products that use NBBS, such as the Nways family of products. To understand this better, we refer to Figure 8, which provides a high-level functional layout of a 2220 Nways switch. Note that on both sides of the broadband switch, a trunk or port adapter is used. The port side is used for user access, with industry interfaces including HDLC, frame relay, ATM UNI, and others. The trunk side (lower left) is used to connect NBBS nodes together, through high-speed (T1–T3, or Synchronous Optical Network [SONET]) links. The control point (CP) functions, specialized servers (such as voice servers) and the node administrator function complete the functional decomposition of the node.

Note that in the 2220 Nways switch, the receiving *trunk–port adapter* (TPA) is able to direct an incoming cell across IBM's unique ATM switching chip to the outbound TPA. This chip has inherent multicast capabilities so that outbound cells can, in fact, be directed to all downstream nodes that are on the CP spanning tree. Each cell (or packet) traverses the switch without being handled by the control

Figure 9 Link-speed effects on end-to-end vs hop-by-hop control decisions



point processor. Information in the NBBS packet header directs a copy of each control packet to the control point processor in the node, which can then process the topology database (TDB) information without interfering with the flow of control point information downstream. Thus, the combination of the distributed CP algorithm implemented in software and the efficient routing implemented in hardware gets control information across the network effectively at propagation speeds.

End-to-end vs hop-by-hop

As discussed, the 2220 Nways switch implementation of NBBS causes the nodal processing time for a cell or packet to be small when compared with propagation delay. Figure 9 illustrates the challenge that confronts each node in the network for high-speed links. The figure attempts to illustrate the number of "instructions" that an intermediate node could afford to execute per cell while attempting to keep up with links of various speeds. The figure does not reflect realistic processing of a 2220 Nways node, which, as we have seen, includes hardware processing on its TPAs, plus switching (at hundreds of megabits per port). Rather, the figure illustrates a point by representing a fictitious node in the network that processes each cell with an effective rate of one million instructions per second. If the trunks in the network were as slow as 9.6 kilobits per second (Kbps), the processor could expend 40000+ instructions on each cell and still keep the link full. At Ethernet and token-ring speeds, the processor could expend 26 to 42 instructions per cell. But at OC-24 (1.2 Gbps), the processor would have to forward 3 cells with each instruction executed! The processor speed would have to be increased to 100 MIPS (millions of instructions per second) in order to allow 35 instructions to be executed per cell.

These observations make it extremely clear that cell switching must be done in hardware and, furthermore, that the processing of each cell must be

simple. This reasoning led the world's broadband experts to design ATM so that it is connection oriented with only a 5-byte header per cell. Little information is carried in the cell header other than the ATM-defined *Virtual Path Identifier/Virtual Channel Identifier* (VPI/VCI) fields used to switch the cell toward its destination. (Note: As in the case of public ATM networks, VPI/VCI translation may be performed at intermediate points whenever ATM cell-switching mode is used by NBBS.) Similar reasoning led to the decision not to implement hop-by-hop flow control and error recovery in NBBS. This decision is independent of the switching mode used in the trunks. The 2220 Nways product currently uses label swap and cell switching for data. For data connections, flow control and error recovery can be provided by transport-layer protocols running above the user interface. For voice and video connections, flow control and error recovery are typically not used.

Although the responsibility for error recovery falls primarily on the edge nodes (entry points and exit points) of an NBBS network, intermediate nodes examine the network header of each cell or packet and discard each packet whose header is corrupted by transmission error. Header transmission errors are detected using an 8-bit longitudinal redundancy check (LRC) that is present in each network header. The LRC is a special case of a cyclic redundancy check (CRC) and is simply implemented in hardware. The data payload of an NBBS packet may be protected by a 16-bit CRC used only by the edge nodes. Note that in the case where the NBBS network is providing ATM transport, the data may similarly be protected by a CRC at the ATM adaptation layer but not at the ATM cell layer.

Multicast support

The requirement to support multimedia applications essentially implies a requirement to support multipoint applications as well. This may arise in a video-on-demand application where educational or entertainment video is to be delivered from a source to multiple destinations simultaneously. It may also arise in desktop videoconferencing where voice, video, and data connections must be simultaneously maintained among all participants.

Supporting multicast requires a robust set of control procedures to establish multipoint connections, allow participants to join and leave a connection, and reroute multipoint connections in the case of link or node failures in the network. NBBS provides a sophisticated set of control features to provide this administrative control. They are described more fully in the multicast network connection architecture paper in this issue.[11]

Set management supports both *open* and *closed* sets in NBBS networks. A videoconference is an example of a closed set where a resource external to the network invokes the NBBS set manager to establish the set, to contact all participants, and to establish one or more point-to-multipoint connections to link all of the participants. An open set could include, for example, all nodes supporting attached Internet Protocol routers so that Address Resolution Protocol packets or other protocol flows could be restricted to specific multipoint connections established by the set manager for the purpose.

In addition to the set manager function for multipoint connections, the network must support efficient routing at the hardware level along the multipoint connections, which the 2220 Nways switch provides with its state-of-the-art trunk–port adapters (TPAs) and ATM switching.

Functional overview of NBBS

In the language of the IBM Open Blueprint*, as well as the language of the Open Systems Interconnection (OSI) protocol model, NBBS is a subnetwork architecture, where a subnetwork is a collection of equipment and physical media that form an autonomous whole and can be used to interconnect other systems for purposes of communication. In the interest of simplicity, the term "NBBS subnetwork" is shortened to "NBBS network" throughout this paper.

NBBS may also be viewed as a control point architecture for providing a variety of transport services, including transport of ATM. For this reason, NBBS should be viewed as an architecture that complements the B-ISDN Protocol Reference Model.[12] Current standards development defines interfaces at the UNI and the network-to-network interface (NNI), but does not address the problems of providing services such as routing and bandwidth management functions in an efficient way. NBBS technology offers the mechanisms to efficiently provide these functions.

An NBBS network connection comprises two paths, one path in each direction, joining a node at the edge of the NBBS network to another node at the edge of the NBBS network. The entry points and exit points mentioned earlier in this paper are in these edge nodes. Network connections may be thought of as the "pipes" in the network that carry information on behalf of user transport connections.

A transport connection is an end-to-end connection between a source and a target, both of which are resources (which are nonnative NBBS nodes) that lie outside the NBBS network. Each network connection can carry one or many transport connections.

Figure 10 is a high-level representation of a pair of NBBS nodes and the functional layer structure of the NBBS architecture. Access services provide the interface between the user nodes or resources and the internals of the NBBS network. Transport services carry traffic through the NBBS network. Network control services allocate and manage the assets of the NBBS network.

The ATM Forum has defined the interfaces (UNI and NNI) for an ATM network.[12] The value of NBBS is that it uses a control point architecture to efficiently provide transport of ATM data. However, NBBS is not limited to ATM data transport. Through the access agent concept, virtually any protocol can be transported efficiently. Furthermore, the value of NBBS is seen in a number of network enhancements including the ability to rapidly and efficiently disseminate network control information (through the CP spanning tree), efficient multicast (through set management), efficient address resolution, bandwidth management, and access services. One can therefore view the NBBS functions as those above the physical layer in the B-ISDN Protocol Reference Model, supplementing services performed by the upper layers. Interconnection between multivendor switches can take place through standardized interfaces like ATM UNI and NNI. In the future, these interfaces could include private-NNI (P-NNI) and B-ISDN User Part (B-ISUP).

Access services. External users are termed resources in NBBS architecture. These may be SNA nodes, APPN nodes, Internet Protocol (IP) routers, or other nodes. A resource uses the services of an NBBS network (through the mediation of an *access agent*) to communicate with other resources like itself. Access agents provide the points of attachment and the services required to join together an NBBS network with its users or resources. Consequently, access agents are needed only by nodes that serve external users or resources.

The purpose of an access agent is to isolate a user from the operation of the NBBS network while allowing the user to use the services of the network. Toward this end, the access agent emulates the communication protocol chosen by the user it serves. For example, an access agent may present the external appearance of a frame-relay bearer service, or an HDLC connection, or circuit emulation. IBM Nways switches implement all of these and more, including sophisticated voice services.

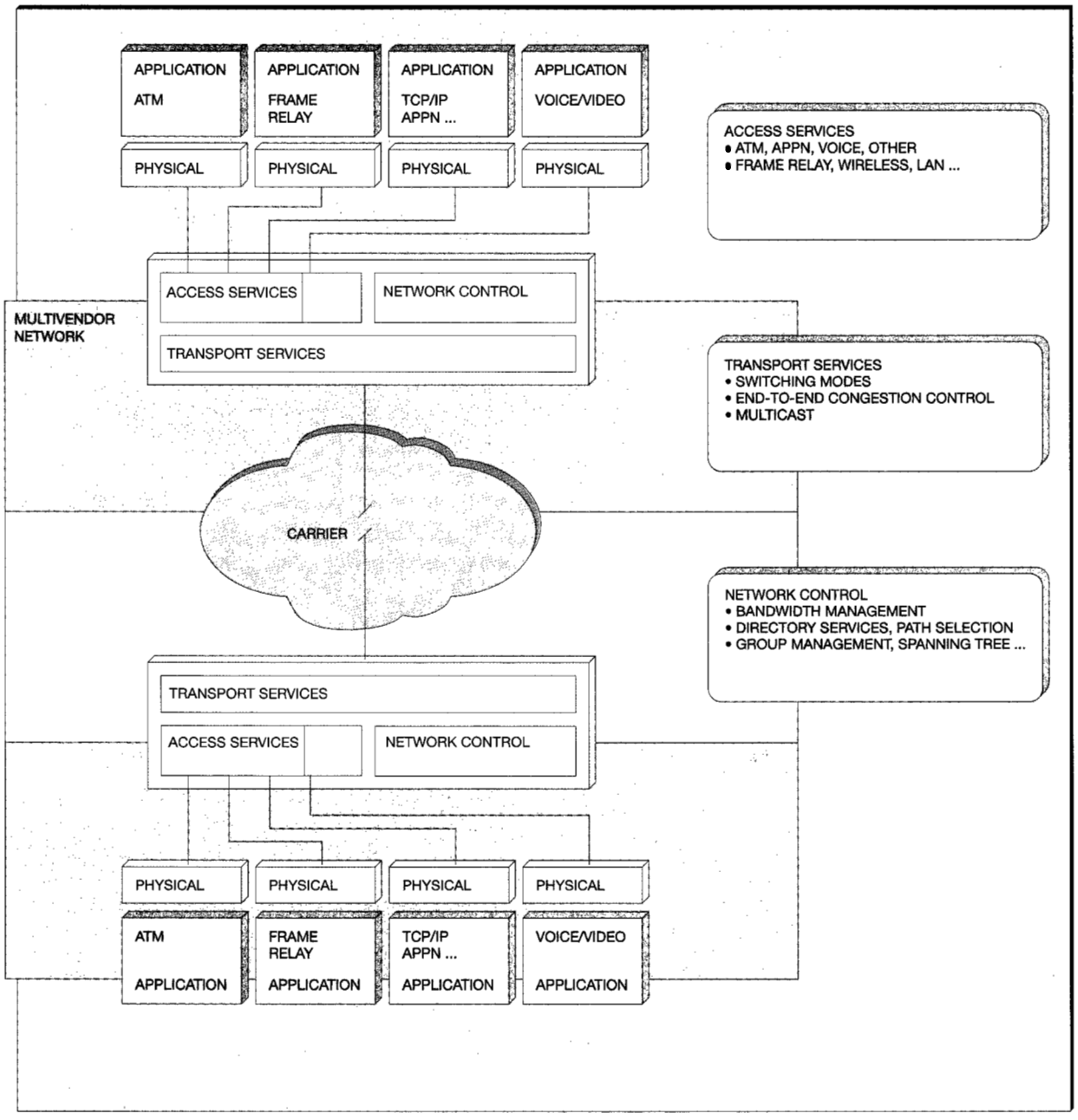
An access agent is comprised of three parts:

- A protocol agent (PA), which provides services that understand and interpret the communication protocol and network-connection requirements of the attaching user
- A connection agent (CA), which provides network connection services that establish and maintain NBBS network connections on behalf of the PA
- A directory agent (DA), which collaborates with other DAs throughout an NBBS network to provide directory services (DS) with two purposes, (1) to maintain a distributed directory database characterizing all of the users and resources of the network, and (2) to locate NBBS resources on behalf of the PAs

The three parts of the NBBS access agent work together to resolve addresses in order to locate the access agents that represent other external users or resources served by the NBBS network; to establish, maintain, and take down network connections to these other resources; to provide the connection setup QOS or compatibility information needed to ensure that these connections meet the appropriate QOS levels; and to interpret the communication protocol of the user. The paper on access services in this issue provides additional details on the services performed by access agents.[13]

NBBS transport services. NBBS transport services carry traffic through an NBBS network. Transport services can be divided into three layers, each of which corresponds roughly to a layer of the OSI seven-layer reference model:

Figure 10 NBBS structure



- Physical layer services
- Network layer services
- Transport layer services

Physical layer services. The purpose of the physical layer services is to allow the network layer to send and receive traffic by means of the transmission media. The physical layer services are the intermediary between the network services that lie above and the physical transmission media that lie below. In particular, the transmission media are not within the scope of the NBBS architecture. NBBS

uses industry-standard specifications of the physical-medium-dependent part of the architecture. Thus, an NBBS network relies on industry standard transmission services, such as those provided by the Synchronous Optical Network (SONET), for the physical transport of data.

NBBS physical layer services include manipulating cells or packets to support the preempt-resume option (a higher-priority packet can interrupt the transmission of a lower-priority packet), and framing packets for transmission. Three kinds of packet framing are offered:

- A low-speed bit-level interface
- A high-speed byte-level interface
- A medium-access-control- (MAC-) level local area network (LAN) interface

The low-speed bit-level interface is appropriate for point-to-point connections that use transmission media such as T1-rate and T3-rate private lines supplied by commercial telecommunication carriers. These connections use the standard HDLC data link control protocol. The high-speed byte-level interface supports SONET links. Mainly, it provides a way to map NBBS packets into frame formats that SONET can use. The MAC-level LAN interface joins an NBBS network layer to a LAN MAC protocol sublayer by encapsulating NBBS packets into the frame format used by the LAN.

Network layer services. Network layer services switch packets, detect and respond to transmission errors, manage buffers, and schedule transmissions in an NBBS network. In this section we concentrate on the transfer modes supported by NBBS. The next section will provide some additional information on transmission scheduling and buffer management.

Automatic network routing (ANR) transfer mode—ANR transfer mode is a form of source routing. The origin (the connection agent at the source) loads the ANR packet header with a complete specification of how the packet is to be switched or routed through the network. The specification is a sequence of ANR "labels," wherein each label identifies the outbound link of the corresponding node in the sequence of nodes the packet transmits through. These labels are static and are defined on a node basis and stored in the NBBS topology database. Because the complete routing specification is included in the ANR header, ANR is appropriate for connectionless transport. That is, ANR is useful when it is impossible or undesirable to set up the routing tables needed at the intermediate nodes to support connection-oriented communication. ANR is particularly important because it is the transfer mode used to transport NBBS control traffic.

Label-swap transfer mode—Label-swap transfer mode associates a label with each packet (or cell as discussed for ATM transfer mode just below). This label specifies which outbound link the packet is to take at the next node. When a packet leaves a node, its old label is replaced by a new label, which is used to index a table and determine routing instructions at the next node. These labels have local significance at the nodes where they are received. They serve to index the switching (routing) tables that are established along the path as the connection is set up, a process discussed later.

ATM transfer mode—Industry-standard ATM cells are sent through an NBBS network using ATM transfer mode. This is a particular version of label-swap transfer mode in which the packet (cell) format is that of a standard ATM cell, with its fixed-length 48-byte payload and 5-byte header. The switching is done using the VPI/VCI fields in the cell header.

Remote access to label-swap tree transfer mode—Remote access to label-swap tree transfer mode supports multicast trees (point-to-multipoint and multipoint-to-multipoint). This transfer mode enables a user to address a group that comprises an NBBS tree, whether or not the user is a member of the group, and whether or not the user is on the tree. This transfer mode is useful, for example, in allowing a local LAN adapter to send a request to a group of remote LAN adapters set up as an NBBS multicast tree.

Optional network-layer flexibility—Beyond the choice of transfer mode, network layer services offer other kinds of flexibility:

- Choice of packet-payload length
- Copy option
- Reverse-path-accumulation option

Except for ATM cells (packets associated with ATM transfer mode), the lengths of the payloads carried by NBBS packets are not bounded by the architecture. By allowing payload length to vary in response to incoming traffic characteristics (except for ATM standard-compliant transmission), and by

allowing relatively long payloads to travel with a single packet header, NBBS makes more efficient use of bandwidth than is possible with a single packet size.

The NBBS copy option allows network control services resident at any node to receive copies of any packets handled in transit, regardless of the ultimate destinations. The copy option is used mainly to disseminate control messages associated with connection setup and set management activities. The standard 5-byte headers in the ATM standard do not permit this, and therefore the copy option is not available for the ATM transfer mode.

The NBBS reverse-path-accumulation option provides routing information so that the receiver of a packet can reply to the sender in a connectionless manner, without having to compute or to find the path back to the sender. For each link taken by a packet along its forward path from the sender to the receiver, the ANR address of the corresponding link in the opposite direction is recorded node-by-node in the packet header. In this way, a packet traversing the forward path automatically accumulates the reverse path from the receiver to the sender. The reverse-path-accumulation option can be used with all of the switching modes except ATM transfer mode and is ideal, for example, for returning acknowledgments.

Transport layer services. Transport layer services organize and manage user traffic so that it can be carried by transport connections. Specifically, transport layer services segment or pack higher-layer messages into NBBS packets, assemble or unpack messages, ensure that messages are delivered, and work with the protocol agent to multiplex a number of transport connections onto fewer network connections when advantageous. All of these services are handled end-to-end rather than hop-by-hop.

Rapid transport protocol—Rapid transport protocol (RTP) transports all NBBS control messages across an NBBS network. RTP provides windowed flow control, message segmentation and reassembly, and connection maintenance that detects the loss of communication between partners. As an option, RTP offers reliable delivery, which means that the sender is informed when the message successfully reaches the receiver.[14]

RTP is a high-performance transport protocol designed to have the following characteristics:

- Optimism. RTP assumes that links and equipment are reliable and available.
- Fast setup. RTP dispenses with the traditional "handshaking" negotiation needed to set up a connection. Therefore, no connection setup flows are required for sending control traffic messages.
- Data-streaming. RTP assumes that the listener is ready and able to receive transmissions, and streams data to the listener until instructed to slow or stop.
- Piggybacking. RTP intelligently imbeds its own protocol-control information within packets that carry user traffic whenever it is advantageous to do so, thus achieving a higher level of efficiency.
- Streamlining. RTP relies on higher layers of the communication protocol to provide session services, such as data encryption, compression, and presentation services (like data conversion between applications), because these services are not universally needed.

NBBS control point services. The NBBS *control point* (CP) services lie at the heart of the NBBS architecture. The CP (the collection of all control point services) need not be implemented in full by any particular component of the network. It can be distributed across the NBBS network nodes. To communicate with each other, control points rely on transport layer services, specifically RTP.

NBBS control point services have five components:

- Link services activate links, deactivate links, and, optionally, authenticate links. A link is activated by link services either at the request of a network operator or upon receipt of a link-activation message from a connection agent. Link services establish physical connectivity, determine the characteristics of the link being activated, check link "liveness," notify the topology services of changes to the link status, and deactivate the link (when appropriate).
- Topology services collaborate to build and maintain a distributed repository of information known as the topology database and a control point spanning tree that connects the network. The primary purpose of the topology database is to provide each node with a repository of information that conveys a current and consistent

view of the network topology, mainly for the use of path selection services.

- Set management services group related NBBS resources into sets and provide multicast distribution trees that are used to send messages efficiently to set members. As noted previously, set management was developed specifically to address the requirement for multicast support in an NBBS network.
- Path selection services build the paths (the ordered sets of links) needed by the connection agents to establish network connections. Paths are built only from links that meet the QOS constraints for the connection; furthermore, paths are built in a way that attempts to maximize the aggregate throughput of the network.[15]
- Congestion control services monitor, estimate, and throttle the flow of cells (packets) at entry access points into an NBBS network to ensure that traffic sources do not unexpectedly flood the network and to meet the QOS requirements that have been established when connections are set up. The congestion control mechanisms of NBBS are introduced above and are more fully described in the companion paper on bandwidth management.[7]

NBBS components and their relationships

An NBBS network is built of components called *links, access links, subnodes*, and *nodes*, which come together to provide paths (where a path is a unidirectional, ordered set of links used for communication by a network connection), network connections, and transport connections for the use of resources. A *resource* is a source or target of traffic external to the NBBS network. A resource uses the services of an NBBS network to communicate with other resources like itself. The most straightforward example of a resource is an application that relies on NBBS to provide the transport connections that carry its traffic.

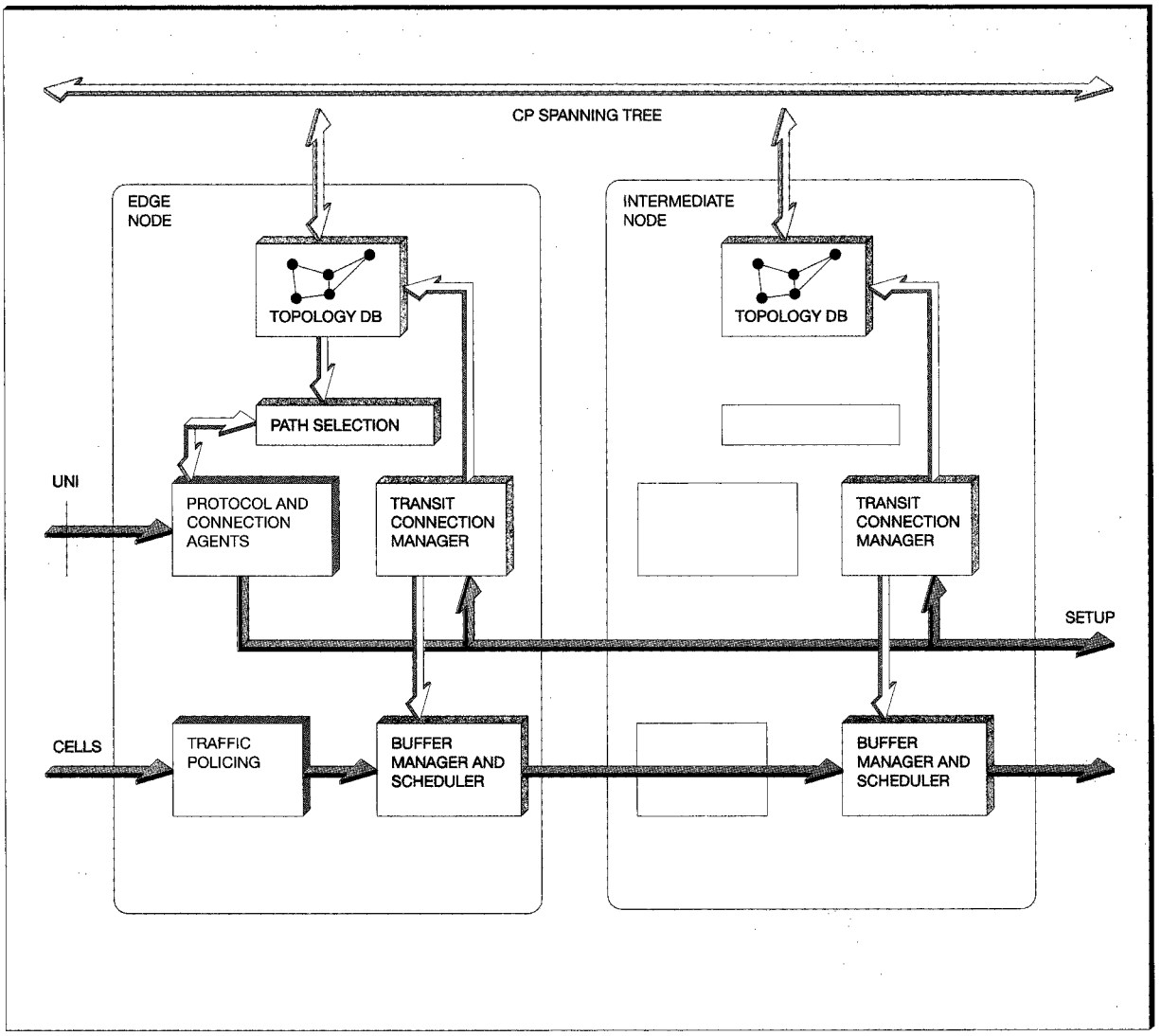
Access links give endpoint equipment an entry into the NBBS network. The user traffic that the NBBS network handles always enters and leaves the network through the access links. The traffic that enters an NBBS node through an access link may be separated into one or more streams and may be serviced by one or more virtual circuits, called network connections. Source and target resources are attached to access links.

A link represents a unidirectional use of a physical transmission medium connecting two nodes. Note that transmission media themselves may not be inherently unidirectional. A link is used to send packets of data from one node to another in an NBBS network. The links of the NBBS network form a backbone over which all user traffic is routed. These links are the network resources that are controlled by the NBBS algorithms. There are generally two competing objectives for the use of these resources: one is to utilize the links as highly as possible, the other is to guarantee QOS to connections using these links. The features of NBBS help network owners to achieve a combination of these two objectives that is right for their network.

A subnode is a switching fabric to which links are attached via trunk–port adapters (TPAs). A node is a collection of subnodes that share control services. In particular, subnodes belonging to a node share one copy of the NBBS topology database. There are two types of nodes in the NBBS network: access nodes and network nodes. An access node has only a subset of the function that is contained in a network node. For example, it does not maintain a full topology database in its node, and does not generate or receive topology updates. It serves as a traffic source or sink. An access node is a client of a network node and receives services from it. The access node therefore serves to reduce the amount of information that the network is required to keep so that a network may be scaled up in size. This reduces memory requirements, network control message overhead, and processing power required, and therefore can significantly reduce network cost.[16]

Network nodes are full-function NBBS nodes. They may also have access services. However, in addition to access services, they have all the network control algorithms necessary to participate fully in the NBBS backbone. The network nodes in an NBBS network can be thought of as multiplexers. Their job is to provide the network services necessary to transport user traffic from an origin to a destination point, while guaranteeing the quality of service (QOS) required by the originator of the traffic. At the same time, the nodes in the network need to use the links as efficiently as possible so that the network owner derives the economic benefit of having a broadband integrated services network. In order to perform these tasks, the NBBS architecture defines a set of functions that are briefly described in the next few sections. In what follows,

Figure 11 Overview of NBBS functions



we describe the functions necessary to maintain the NBBS network itself, and also to set up, maintain, and take down the user network connections that the NBBS network enables. Figure 11 illustrates some of these functions with a decomposition of two network nodes, one at the edge of the network and the other at an intermediate location.

The control point. The collection of control functions provided by the NBBS architecture comprise the *control point*. Each node in the NBBS network has a control point that implements the network control algorithms required to deliver the services expected by users. The NBBS control point was designed for flexibility in implementation. It can be implemented either as a centralized monolithic function within a node or distributed throughout the node to a greater or lesser degree. This flexibility allows different implementations to make design trade-offs that are optimized for their particular environment and market.

NBBS control points communicate with each other using the Rapid Transport Protocol (RTP). A num-

ber of services are required by the control point, either for transmission of control traffic or for user data. These include reliable and unreliable point-to-point communication, as well as services like linear multicast (where information is sent on a multicast tree and each node that is part of the group copies the packet or cell as the data go through the intermediate nodes), and unreliable multicast (which may be used for real-time audio or video transmission to a group of users).

RTP provides a number of service primitives, including reliable or unreliable point-to-point communication, linear multicast, and unreliable multicast, all of which are used by functions of the control point. It is important to note that in order to provide its service primitives, RTP uses the underlying transfer modes of the NBBS architecture described in the section on network layer services. Architecturally, the functions and algorithms of the control point that are visible are those that are externally addressable, that is, those that have network addresses such as E.164, X.121, SNA names, or IP addresses. RTP also provides integrity checking (through CRC) and flow control for the protection of hardware buffers at the entry and exit points, as well as retransmission for reliability.

How a node joins the NBBS network. In order to see how the different parts of the NBBS control point relate to one another, we consider two key scenarios. The first one, described in this section, shows how a node becomes part of the NBBS network. The second scenario, described in the next section, shows how a user connection is set up. The objective of these sections is to show how parts of the architecture that are concerned with maintaining the NBBS network work together to produce a system capable of delivering network services to users.

When an NBBS node comes up, it will, depending on the particular product implementation, execute a number of self-tests and self-initialization procedures. The NBBS architecture does not define these, since they are product-specific and even model-specific. Logically, however, the NBBS node is at this point a network of one node. There are no links represented in its topology database, and it is not yet in contact with any other node.

The first architecture function that will be executed is the link manager. The function of the link manager in NBBS is to initialize a link, maintain the link state, and monitor the health of the link. Links in NBBS are all unidirectional representations of the underlying physical media. The link manager owns the link that is in the outward direction from the subnode where the link manager resides. The NBBS link manager is responsible for ensuring that there is communication between the two ends of a point-to-point link. Although somewhat different procedures are used, the architecture will support non-switched lines, switched lines, and ATM permanent virtual path connections (VPCs) as transmission links. The link manager can perform line test procedures in order to ensure that communication is correct. An NBBS link becomes activated through the link initialization protocol executed by the link manager. Once activated, the link can then be used by the network and by user traffic. The link manager also periodically executes a link liveness protocol to ensure that the link remains a viable medium. When a link is activated, the link manager causes it to be represented in the topology database.

Once the links of an NBBS node are active, other algorithms can start to operate. The next step for a node joining the network is to execute the CP spanning tree (CPST) algorithm. The CP spanning tree is an important communications mechanism of the NBBS network for NBBS control traffic. The NBBS architecture uses the CP spanning tree in order to distribute topology and network load information. In some cases, the CP spanning tree is also used for directory searches. A single NBBS network has a single CP spanning tree. The CP spanning tree is, logically, a permanent multipoint-to-multipoint connection. A node not yet in the NBBS network constitutes a CP spanning tree of just one node. The CPST algorithm incorporates the idea of a leader, or *root*, of each CP spanning tree. When two trees join together, only one root remains. When a new node joins an NBBS network, the CP spanning tree is automatically enlarged to include it. When a failure causes a network to partition, a single CP spanning tree will automatically split into two separate ones. When a network partition is repaired, the CP spanning trees in the two (formerly disjoint) networks combine into a single tree again and update each other on the current topological states of their respective networks. The CPST algorithm is very resilient in the face of failures and tries to keep the largest possible part of the network operating as a single unit.

The most basic procedure in the CP spanning tree algorithm is the joining of two partitions. At any particular step in the algorithm there may be a number of partitions, each of which has several links that connect it to adjacent partitions. From its topology database, each node learns of all of the partitions' outgoing links (links to another partition). Each partition then selects the outgoing link with the largest weight (a unique weight assigned to each link). Because of the uniqueness of the link weight, each of the adjacent partitions will agree on the link that can then be used to join the two partitions.

The topology database contains information about the nodes, links, and other resources in the network.

This creates a single, larger partition, and the subtree that now spans the larger partition is used to update the topology database in each node of the new partition. In a hypothetical network in which all the nodes were powered on at once, pairs of nodes would simultaneously form spanning trees and these pairs would then join to form trees of four nodes, and so on, until all the nodes belonged to a single spanning tree.

Continuing our scenario, the single NBBS node will attempt to join one of the networks to which it is attached by its links. If the node is attached to multiple (disjoint) networks, it will attempt to join them one at a time. When the new NBBS node becomes part of a single CP spanning tree, it sends the topology it knows to the network that it just joined. At the same time, it learns the topology of the network of which it has just become a member. Once this initial topology distribution has taken place, our single NBBS node participates in the ongoing topology distribution algorithm.

All the network nodes in the NBBS network keep a copy of the topology database. The topology database is a set of records containing information about the nodes, links, and other resources in the network. It is a fully replicated, distributed database containing two kinds of information. *Configuration information* includes what links exist, their characteristics (speed, buffers, propagation delay, etc.), the functions that the link supports, whether or not the node is a registrar for set management services, and more. *Load information* includes current bandwidth reservation levels at each of the delay priorities and the current link-utilization levels on the links at each delay priority.

Correspondingly, there are two kinds of topology database updates that are distributed: configuration updates, which are multicast reliably on the CP spanning tree, and utilization updates, which are multicast unreliably on the spanning tree. Reliability is achieved for configuration updates by means of an acknowledgment scheme that takes advantage of the knowledge that each node has of its physical neighbors. These two types of updates are treated differently because of the expected frequency and extent to which they affect network operations. Configuration updates need to be sent reliably since they are expected to be relatively infrequent, but have a great effect on the network. For example, the addition of a new transmission link or the failure of a node in the network will affect both existing connections and connections that arrive shortly after the event happens. Therefore it is worth the extra flows (and extra time) to ensure that all configuration updates are received by all the nodes in the network. On the other hand, utilization updates are sent unreliably because they are relatively frequent (on the order of seconds between updates). If a node misses an update, or if an update is dropped from a transmission link queue, there will soon be another. The thresholds that trigger a topology update are tunable so that updates may be sent more or less frequently.

The topology database update distribution, since it is multicast on the CP spanning tree, is very efficient compared to other (broadcast) distribution algorithms. First, there are at most $N - 1$ links crossed (where N is the number of nodes in the network). This is an advantage for large networks, since there is less control traffic and it is limited to the CP spanning tree links. Second, the multicast is in hardware and consequently is very fast. That means that the information reaching each topology database copy is more recent and more accurate.

Supporting the network structure described here is NBBS network management services. Network management defines both architecture objects

Table 1 Connection classifications

Connection Type	Nonreserved	No QOS Guarantees, Labels Required	QOS Guarantees, No Labels Required	QOS Guarantees, Labels Required
Point-to-point	X	X	X	X
Point-to-multipoint		X		X
Multipoint-to-multipoint		X		

Note: "X" means that the type of connection is supported.

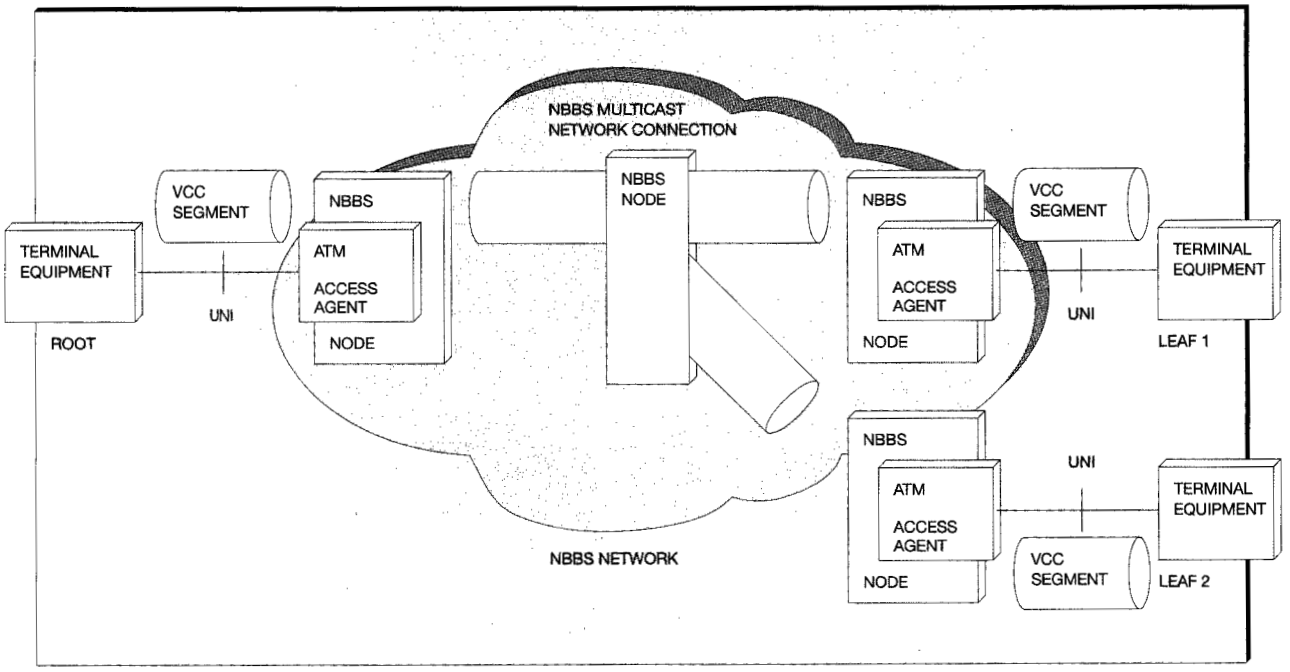
(such as connections, control points, etc.) and physical objects (such as nodes, adapters, power, etc.) and relates these objects to one another for purposes of configuration management. In addition, network management defines a number of "counter" objects that support the accounting and performance disciplines. The details of NBBS network management are described in a companion paper.[17]

Connections in NBBS. NBBS supports a rich set of connections across its high-speed backbone links. A connection in NBBS is a logical entity that is made up of at least one path and two or more endpoints. Data, and sometimes signaling information, flow over the path(s) between the endpoints. This is quite a general definition and encompasses ATM virtual path connections (VPCs) and virtual channel connections (VCCs) as well as other types of connections. NBBS supports both permanent virtual circuits (PVCs) and switched virtual circuits (SVCs). The types of connections supported can be divided into classes in several ways. Connections can be classified as point-to-point, point-to-multipoint, and multipoint-to-multipoint according to the number of endpoints and the number of sources of the information flow. Connections can also be classified according to whether or not they require reserved resources. There are two kinds of network resources that may be required for a connection: *labels* (including ATM virtual path identifiers [VPIs] and virtual channel identifiers [VCIs]) and bandwidth. A connection may require either labels or bandwidth, or both, to be reserved. For example, an ATM connection requiring only best-effort service (i.e., no QOS guarantees) would require only VPI/VCI labels to be reserved for the life of the connection. Connections that require QOS guarantees must have bandwidth reserved and may, depending on the transfer mode, require labels or VCI/VPI labels to be reserved. Table 1 shows how these two classifications are related.

Access services, introduced earlier, play a key role in establishing network connections in NBBS. The protocol agent's job is to interpret signaling from the user (terminal equipment, PBX, or network router) and translate it into requests for services from the NBBS network. For example, an off-hook signal followed by dial digits would be interpreted as a request for a 64 Kbps voice call. The protocol agent can use the information contained in the signaling to initiate the destination address resolution function through the directory agent and initiate the bandwidth reservation through the connection agent.

This ability of the protocol agent gives NBBS some important qualities. First, it provides protocol independence. The protocol agent does not convert protocol; it does participate in and interpret the end-system protocol, allowing NBBS to carry the traffic and provide the services required to the end user. In fact the protocol agent matches the features available in NBBS to those required, providing the best possible transport for the native protocol. For example, the label-swap transfer mode is selected by the frame-relay protocol agent when providing frame-relay bearer service over NBBS. The protocol agent concept allows its use for virtually any layer in the Open Systems Interconnection (OSI) Reference Model; protocol agents have been designed and implemented for private branch exchanges (PBXs), HDLC transport, frame relay, and ATM. Others may be added in the near future for SNA, IP, and switched multimegabit data service (SMDS), and all these may coexist in the same physical box. Next, the protocol agent gives NBBS the ability to supply completely standard services to end systems. For example, the frame-relay and ATM protocol agents completely support use by end systems of the published UNI protocols, providing standards compatibility for IBM's or other vendors' equipment. Finally, the concept of a protocol agent makes the NBBS architecture fully extendible. Any

Figure 12 ATM connections and NBBS network connections



end-system protocol can be accommodated by an NBBS network, once the access agent for that protocol is constructed. The choice of whether to support a particular protocol or not will depend on market considerations. Figure 12 provides a functional overlay of an ATM-defined connection over an NBBS network. The diagram shows how ATM UNI-defined parameters map to internals of an NBBS network. This diagram serves as a useful backdrop to the scenario that is described in the following pages.

The life of an NBBS connection. In this section, in order to make the concepts clearer, we discuss the life of an NBBS connection. To limit the discussion, we concentrate on a single point-to-point connection from one network node to another. However, in principle the steps to set up and maintain a point-to-multipoint or a multipoint-to-multipoint connection are the same. Rather than discuss the details of a specific access agent, we instead describe those functions that are common to all access agents. In order to simplify the explanation for the overview, we refer the readers to the companion paper on access agents[11] for details of the algorithms described.

To set the stage for our scenario, assume that the NBBS network is operating normally, control information is flowing, and network connections already exist in the network. Also assume that a new connection between two end systems is being requested that requires both QOS guarantees and labels to be reserved. Since labels are being reserved, the connection is using one of three possible transfer modes: label-swap, ATM, or ANR. This is determined by the access agent responsible for the connection.

Suppose, then, that a connection request arrives at the protocol agent of an NBBS node over one of the access links attached to the node. The access agent initiating the connection becomes the *originating access agent* and the other side becomes the *destination access agent*. The protocol agent determines either from the end-system signaling protocol (if this is an SVC) or from defined parameters (if this is a PVC) some important pieces of information: the *target resource* of the communication (represented by a string), a set of QOS parameters, and traffic descriptors for the *source resource* and the *target resource*. The string representing the target resource may take any one of

a number of forms: a name, an E.164 address, or an NSAP (network service access point) address (among other things). The protocol agent causes the directory agent to resolve the string to a network address. The directory agent takes a sequence of steps to determine whether it already has the network address that it needs and whether that address needs to be checked before a connection setup is attempted. If the directory agent has no knowledge of the address, it performs an *undirected explicit query*, which is a multicast directory search over the multicast tree associated with the set of directory agents belonging to that protocol. If the directory agent does have some knowledge of the address, then depending on the type of resource, the directory agent may simply return the address to the protocol agent that made the request. The network address is a qualified address with a network identifier as the top level of qualification. The network address gives the exact location of the target resource in the NBBS network. In addition, when the network address of the resource is returned, the network addresses of the connection agent, directory agent, and protocol agent serving the resource may also be returned. These addresses give the originating access agent flexibility in communicating with its counterparts about the connection. This is required to ensure compatibility across all types of access agents, including voice-PBX.

Once the address of the target resource is resolved, the connection setup procedure can begin. The first step is to find a feasible path through the NBBS network to the destination. As mentioned earlier, a path is a unidirectional, ordered set of links used for communication by a network connection. The path is found by path selection services. Path selection services depend upon the topology database for current information about the physical topology and the current load conditions in the network. Path selection services use this information along with the QOS requirements of the connection and the traffic descriptors in order to find the paths for the connection. Path selection provides the first level of admission control for the connection in the following way: if there are not sufficient network resources to support the required QOS, then the connection will be rejected. Note that this selection process is independent of adaptation; the connection request will result in a value of equivalent capacity being calculated. This value will be used as the initial bandwidth required for the connection setup. Connection admission control is a function essential to guaranteeing QOS, since it not only ensures that the QOS of a new connection can be met, but also guards against the degradation of the QOS of connections that are already established. In addition, the path selection services algorithm tries to optimize the use of network resources by placing connections on paths that use as few links as possible.

Before the connection is set up, the originator has the responsibility for the entire connection and so computes the path for the origin-to-destination direction as well as the destination-to-origin direction. The former is called the *forward path* and the latter the *return path*. Path selection computes both of these paths (they may be the same) so that the QOS requirements are met and the path is capable of supporting the bandwidth reservation computed from the traffic descriptors, called the *equivalent capacity*.

Path selection services get current information about the network from the topology database.

The reasons that the forward path can be different from the return path in NBBS are two: first, two paths give extra availability since a link failure may affect traffic only in one direction of flow. Second, two paths may allow more connections to be packed into a network because the asymmetric requirements of connections may not always allow the forward and return paths to be the same. If constrained to be the same, the network might reject some connections that would otherwise fit.

The connection agent uses the paths that have been computed to send out setup messages for each direction. The setup messages are sent simultaneously in order to reduce the probability that competing connections would take the resources needed by the path in one direction while the other was being set up. The connection setup flows are actually a multicast communication between the originating connection agent, the transit connec-

tion managers, which manage the links of the network, and the destination connection agent. The originating connection agent uses the reliable multicast transaction form of the RTP linear multicast service in order to send the connection setup flows reliably. As can be seen in Figure 11, earlier, the connection setup messages reach each transit connection manager along the path and the destination connection agent. As noted earlier, the setup message that is multicast is switched in hardware. Thus, the request will arrive at all the nodes along the path staggered only by the propagation delay between nodes. The setup request will be processed nearly in parallel by all the nodes involved in the path. The transit connection managers check their bandwidth resource tables and their label tables when they receive the message.[10]

The setup request will be processed nearly in parallel by all the nodes involved in the path.

The transit connection manager keeps track of the bandwidth that has been allocated for each delay priority on the link that it controls. Because this changes significantly over time, this information is relayed to the topology database algorithm, which distributes this new load information to all the nodes in the network. This information is then used by the path selection algorithm when computing paths for new or rerouted connections, thus closing the information loop.

The bandwidth resource allocation is checked to ensure that the equivalent capacity requested by the connection can be accommodated. Each link in the NBBS network has a fraction of its capacity that can be used for reserved bandwidth connections. This fraction, normally about 85 percent, is called the reservable capacity of the link. When a new connection requests resources, or when an existing connection requests increased or decreased resources, the transit connection manager checks the current allocation against the reservable capacity and determines whether or not the connection can be accepted.

When the transit connection manager determines the acceptability of a connection, it uses both the equivalent capacity and some statistical descriptors of the traffic—the mean bit rate and the standard deviation. The latter are used as part of a Gaussian approximation of the bandwidth necessary when there are a large number of sources multiplexed on a link. This approximation accounts for the statistical multiplexing effect of having many connections and is sometimes less than the equivalent capacity. The bandwidth reserved on the trunk is the smaller of the sum of equivalent capacities and a combination of the statistical descriptors for all the connections.

If labels are available, each transit connection manager communicates its label to its upstream neighbor (upstream to the direction of data flow). If labels and bandwidth can be allocated, the transit connection managers accept the connection and respond to the originating connection agent. If labels cannot be allocated, or if the current bandwidth allocation on the link plus the requirements of the new connection exceed the reservable capacity of the link, then the transit connection manager rejects the new connection. If the new connection has sufficiently high priority, and if, for example, the bandwidth required by the connection would cause the reservable capacity of the link to be exceeded, then the transit connection manager can preempt lower priority connections on behalf of the new connection. In this case, the new connection will be accepted and the lower priority connections may be rerouted. It is possible that the lower priority connections may be dropped if no bandwidth is available in the network, but our simulations show that this would be a very rare occurrence.

In addition, the transit connection manager causes the connection to be associated with the correct delay priority queue and may perform internal functions to associate the connection with internal switching fabric paths.

The destination connection agent also processes the connection setup messages and decides to accept or reject the new connection. In addition, the connection setup messages carry directory information and may carry protocol agent information. The directory information is the result of a *directed implicit query* and verifies that the target resource

is still at that network address. The protocol agent information in the flow provides a means for communication between protocol agents to be piggybacked on the connection setup, thereby reducing the time that the setup process takes.

After all the replies from the transit connection managers and the destination connection agents arrive at the origin connection agent, the network connection setup is almost complete, if all the replies are positive. At this point, the originating connection agent passes responsibility for the *return path* to the destination connection agent. From now on, the destination connection agent manages the services for the return path, including bandwidth adjustment and path switching.

Assuming that the labels and all the requested bandwidth were allocated, the connection agents (originator for the forward path and destination for the return path) set up the leaky buckets for the connection, one in each direction. The leaky buckets monitor the traffic coming in on the connection and ensure that the source or target resources do not exceed the allocation to which the transit connection managers of the paths have agreed.

Now the network connection enters the productive phase of its life. In this phase, there is little involvement of the NBBS control point. As far as the data path is concerned, the data from the source or target are placed in ATM cells or in NBBS packets and sent through the network using the transfer mode selected for the connection. The packetization and switching are done in hardware, so the delays introduced by these functions are minimal. However, there are two situations in which the network control functions become involved again: one is when the traffic characteristics of the connection change and the bandwidth needs to be adjusted; the other is when there is a failure in the network or the connection is preempted by a higher priority connection.

As pointed out earlier, the bandwidth of a connection may not stay the same during the time that the connection exists. Further, the traffic characteristics of a source may not be well known and may be difficult to estimate. In those cases, NBBS can supply an estimation and adaptation function. This function uses the leaky bucket not only as a policing device, but also as a measurement device to determine the traffic characteristics presented to the network over time. The leaky bucket measures, in real time, some of the characteristics of the traffic stream being carried by the NBBS connection. It defines a region in which the source can be said to be conforming to the agreement implicit in the initial bandwidth allocation. If the traffic measurements show that the source has left the region of conformance, then the bandwidth is automatically renegotiated by the connection agent through a process like the connection setup process. The bandwidth allocation can be negotiated either up or down, and the renegotiation can take place many times during the life of a connection. If the bandwidth cannot be increased for some reason and the connection requires it, then the path may be rerouted by the connection agent if bandwidth is available elsewhere in the network. The possibility exists that bandwidth may not be available anywhere in the network, causing the bandwidth increase request to fail; however, our simulations show that with our algorithms for determining equivalent capacity, this is extremely unlikely to happen.

A connection can protect itself against failures and increase the availability that it receives from the NBBS network by requesting the nondisruptive path switch function. If a connection has its network resources preempted by a higher priority connection, or if a failure occurs in the network that affects the connection, the connection agent is notified, by the transit connection manager in the first case and through the topology database in the second. When that occurs, the connection agent works with the path selection algorithm to find a new path for the connection, then sets it up in cooperation with the transit connection managers and the other connection agent.

Finally, as a result of signaling (for SVCs), or at a predefined time or upon a predefined event (for PVCs), the connection will be taken down. The disconnect process is very much the same as the setup process. When the disconnect message arrives at the transit connection managers and the opposite connection agent, the network resources that were allocated are released for the use of other connections.

Summary

In this paper, we have introduced the NBBS architecture. We discussed the need for a change in the networking paradigm because of the recent changes in communications technology. We dis-

cussed the motivation for building a new architecture and some of the factors that helped to shape it. We also discussed the value of the functions of the architecture to IBM's networking customers. We gave an overview of the structure of the architecture and of its major pieces. Finally, we used scenarios to show how some of the pieces fit together to provide the services that an integrated services high-speed network has to deliver.

The subsequent papers in this issue provide more information on the functions and characteristics of the architecture.

*Trademark or registered trademark of International Business Machines Corporation.

Cited references and note

1. M. de Prycker, *Asynchronous Transfer Mode: Solution for Broadband ISDN*, Ellis Horwood Ltd., Hertfordshire, UK (1993).
2. D. E. McDysan and D. L. Spohn, *ATM Theory and Application*, McGraw-Hill, Inc., New York (1995).
3. *Networking BroadBand Services (NBBS) Architecture Tutorial*, GG24-4486, IBM Corporation (September 1995); available through IBM branch offices.
4. I. Cidon, I. S. Gopal, P. M. Gopal, R. Guerin, J. Janniello, and M. Kaplan, "The plaNET/ORBIT High Speed Network," *Journal on High Speed Networking* **2,** No. 3, 171–208 (1993).
5. B. S. Davie, D. D. Clark, D. J. Farber, I. S. Gopal, B. K. Kadaba, W. D. Sincoskie, J. M. Smith, and D. L. Tennenhouse, "The AURORA Gigabit Testbed," *Computer Networks and ISDN* **25,** No. 2, 599–621 (January 1993).
6. R. J. Cypser, *Communications for Cooperating Systems: OSI, SNA, and TCP/IP*, Addison-Wesley Publishing Co., Reading, MA (1991).
7. H. Ahmadi, P. F. Chimento, R. A. Guérin, L. Gün, B. Lin, R. O. Onvural, and T. E. Tedijanto "NBBS Traffic Management Overview," *IBM Systems Journal* **34,** No. 4, 604–628 (1995, this issue).
8. W. Feller, *An Introduction to Probability Theory and Applications*, Volume 1, 3rd edition, John Wiley & Sons, Inc., New York (1968).
9. The ATM Forum is a worldwide industry consortium whose charter is to accelerate convergence on ATM interoperability specifications based on internal standards. The forum began in 1991 with four member companies, and it now has a membership of several hundred organizations.
10. G. Lebizay, C. Galand, D. Chevalier, and F. Barre, "A High-Performance Transport Network Platform," *IBM Systems Journal* **34,** No. 4, 705–724 (1995, this issue).
11. N. Budhiraja, M. Gopal, M. Gupta, E. A. Hervatic, S. J. Nadas, and P. A. Stirpe, "Multicast Network Connection Architecture," *IBM Systems Journal* **34,** No. 4, 590–603 (1995, this issue).
12. The ATM Forum, *ATM User-Network Interface Specification: Version 3.0*, Prentice Hall, Englewood Cliffs, NJ (1993).
13. C. P. Immanuel, G. M. Kump, H. J. Sandick, D. A. Sinicrope, and K. Vu, "Access Services for the Networking BroadBand Services Architecture," *IBM Systems Journal* **34,** No. 4, 659–671 (1995, this issue).
14. M. Peyravian, R. Bodner, C.-S. Chow, and M. Kaplan, "Efficient Transport and Distribution of Network Control Information in NBBS," *IBM Systems Journal* **34,** No. 4, 640–658 (1995, this issue).
15. T. E. Tedijanto, R. O. Onvural, D. C. Verma, L. Gün, B. Lin, and R. A. Guérin, "NBBS Path Selection Framework," *IBM Systems Journal* **34,** No. 4, 629–639 (1995, this issue).
16. 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).
17. S. A. Owen, "NBBS Network Management," *IBM Systems Journal* **34,** No. 4, 725–750 (1995, this issue).

Accepted for publication June 28, 1995.

Gerald A. Marin *IBM Networking Hardware Division, 800 Park Offices Drive, Research Triangle Park, North Carolina 27709 (electronic mail: marin@vnet.ibm.com).* Dr. Marin is the Director of Networking Architecture. As a technical team member, first-line manager, and second-line manager, he has contributed to the development of the Networking BroadBand Services (NBBS) architecture and to its design and development in the Nways switch. In addition to its responsibilities for NBBS, Networking Architecture is responsible for enhancements of the Advanced Peer-to-Peer Networking (APPN) architecture, multiprotocol transport networking (MPTN) architecture, and for Advanced Program-to-Program Communication (APPC) and Common Programming Interface for Communications (CPI-C). The area also has significant responsibility to help represent IBM in standards organizations (T1, ITU, IEEE 802) and in industry organizations (ATM Forum, Frame-Relay Forum, Internet Engineering Task Force, APPN Implementors Workshop). Dr. Marin is involved in many aspects of the Networking Hardware Division's ATM technology including LAN emulation, network management, and ATM interfaces. He was instrumental in making IBM a part of the ATM Forum and serves there as IBM's principal member. Prior to transferring to Networking Systems in 1989, he led the Network Systems Analysis organization in the Systems Integration Division. His responsibilities included working with customers from a number of different industries (banking, airlines, insurance, and others) to design and analyze potential communication network solutions. Dr. Marin has been with IBM since 1982. Before that he was a member of the professional staff at the Center for Naval Analyses. He received the Distinguished Public Service Award from the United States Navy for his work in antisubmarine warfare technology. He holds a Ph.D. degree in mathematics from North Carolina State University.

C. Paul Immanuel *IBM Networking Hardware Division, 800 Park Offices Drive, Research Triangle Park, North Carolina 27709 (electronic mail: immanuel@ralvm6.vnet.ibm.com).* Mr. Immanuel is a senior engineer at IBM in the Networking Architecture development organization. He is a team leader for access agent development for Networking BroadBand Services (NBBS) architecture and has designed access agent architectures for frame relay and voice. He also has interests and assignments in the areas of optical networking and intelligent networks. Prior to his present assignment, he was a lead architect for link services architecture and developed a variety of link-

level enhancements for both Systems Network Architecture and Advanced Peer-to-Peer Networking Architecture, including development of link-level security algorithms for connections through switched networks. Mr. Immanuel joined IBM in 1984 after 18 years of design and development experience in analog microwave, digital, and optical transmission systems. His experience includes hardware design as well as system design and planning of these systems for a number of companies, including the ITT Corporation. Since joining IBM, he has received a number of awards, including invention awards and a corporate Outstanding Technical Achievement Award for his work on NBBS architecture. He has a B.S. degree in electrical engineering from North Carolina State University and is a senior member of the Institute of Electrical and Electronics Engineers (IEEE). He has also taken several graduate-level courses in digital signal processing, communications, and computer science.

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).

Inder S. Gopal *IBM Research Division, Thomas J. Watson Research Center, 30 Saw Mill River Road, Hawthorne, New York 10532 (electronic mail: gopal@watson.ibm.com).* Dr. Gopal received the B.A. degree in engineering science from Oxford University, England, in 1977 and the M.S. and Ph.D. degrees in electrical engineering from Columbia University, New York, in 1978 and 1982, respectively. Since 1982 he has been at IBM, serving in various technical and management positions. From 1982 to 1989, he was a research staff member at the IBM Thomas J. Watson Research Center, working on topics related to distributed algorithms, communication protocols, network security, and high-speed packet switches. In 1990 he moved to the IBM Application Solutions Division and managed the development of the Transmission Control Protocol/Internet Protocol (TCP/IP) routers that form the Internet backbone. He also led IBM's participation in the NSF/ARPA-sponsored AURORA gigabit testbed and several other pilot projects in the area of high-speed networking. He then moved to IBM's Networking Hardware Division, where he was the product manager of IBM's low-end ATM switching product (Nways 200). He was director of architecture in IBM's newly formed Networked Application Services Division, developing IBM's technical strategy for networked services. He has recently moved back to the Research Division to take on responsibility for the networking research activities in IBM. He has published extensively in the field of networking, and he has received Outstanding Innovation Awards from IBM for his work on the PARIS high-speed network and on the Networking BroadBand Services architecture. Dr. Gopal is currently an editor for the *Journal of High-Speed Networking*. He has previously served as guest editor for *Algorithmica*, guest editor for *IEEE Journal on Selected Areas in Communications*, editor for *Network Protocols for the IEEE Transactions on Communications*, and technical editor for *IEEE Communications Magazine*. He has served on several program committees for conferences and workshops. He is a fellow of the IEEE.

Reprint Order No. G321-5582.