Data link switching: Present and future

by P. W. Gayek

The integration of computer networks has made it increasingly important for networking equipment to simultaneously handle a variety of data communications protocols. Networking products known as routers have proven themselves capable of handling many multiprotocol networking requirements, but have had difficulty addressing some important network configurations. Two of the most widely implemented protocols, IBM's Systems Network Architecture (SNA) and Network Basic Input/Output System (NetBIOS™), have characteristics that make it difficult for routers to support them in the same way as routers support other protocols. Networking vendors have devised a number of methods for transporting SNA and NetBIOS data traffic, but these methods have been largely nonstandard and have had other disadvantages. Data link switching (DLSw), initially developed by IBM, has attracted considerable interest among router vendors as a standard way to handle SNA and NetBIOS traffic and avoid some of the problems of earlier methods. A multivendor interest group within an IBM-sponsored forum on Advanced Peer-to-Peer Networking™ has developed and recently issued a standard DLSw specification. This paper briefly compares DLSw to the technologies that preceded it, provides a tutorial of the Version 1 DLSw standard, and discusses possible directions in which DLSw may evolve.

With the growth of multiprotocol router-based networks, network planners have faced the problem that two of their most widely deployed protocols, IBM's Systems Network Architecture (SNA)[1,2] and Network Basic Input/Output System (NetBIOS*)[3], were not easily handled by these routers. SNA subarea routing, as performed by front-end processors running IBM's Advanced Communications Function/Network Control Program (ACF/NCP), involves complex connection-oriented functions that are not natural for a router designed to perform datagram (an individual frame whose delivery is not guaranteed) forwarding at the network layer (layer 3 of the Open Systems Interconnection networking model).[4] Although frames normally contain a destination address, neither NetBIOS frames nor peripheral-node SNA frames have a location-dependent network layer address with which a router can construct a routing database and make simple frame forwarding decisions. For this reason, SNA and NetBIOS are sometimes referred to as "nonroutable" protocols.

To achieve the economies of having a single backbone wide area network (WAN) carry all of the networking protocols for an enterprise, networking vendors have developed various methods for accommodating SNA and NetBIOS traffic. These methods fall into two general categories: those implemented in end stations (e.g., host and personal computers), and those implemented in bridge and router products, which are separate from end stations and do not run user application programs. End-station solutions involve a layer of software that maps native requests of application programs for networking services to a different protocol in use on the attached network. Two important examples within this category are the concept of the Multiprotocol Transport Networking (MPTN) Access Node,[5] and the method defined by the Internet Engineering Task Force in Request for Com-

 0018-8670/95/$3.00 © 1995 IBM

ments (RFC) 1001 and RFC 1002 for mapping the NetBIOS application programming interface to services available in a network running the Internet Protocol (IP).[6] Approaches such as these allow a backbone WAN with a single protocol to support applications written to a variety of network programming interfaces. These methods are beyond the scope of this paper.

Data link switching (DLSw) is a method for handling SNA and NetBIOS data traffic that falls within the category of bridge and router-based solutions. Approaches in this category all leave the end station native SNA and NetBIOS software unchanged, and preserve normal external frame flows as much as possible. Any mapping from these native protocol flows to a single backbone network protocol takes place in a router, and not in the end station itself.

This paper introduces DLSw in the context of other bridge-based and router-based methods that preceded it, explains the history of DLSw, provides a tutorial of the current level of the DLSw standard, and discusses possible directions for the future of DLSw.

Predecessor technologies

The four most common approaches for handling SNA and NetBIOS traffic in a multiprotocol environment are illustrated in Figure 1 and are discussed below.[7] These methods are known by a variety of names other than those used here. Tunneling is also called *synchronous passthrough*, *IP encapsulation*, and *SDLC relay* (for the Synchronous Data Link Control protocol). Alternate terms for spoofing include *remote polling* (for SDLC), and *local acknowledgment*. These approaches all interact with end stations using Open Systems Interconnection (OSI) layer-2 functions—the *subnetworking* layer of the IBM Open Blueprint*.[8]

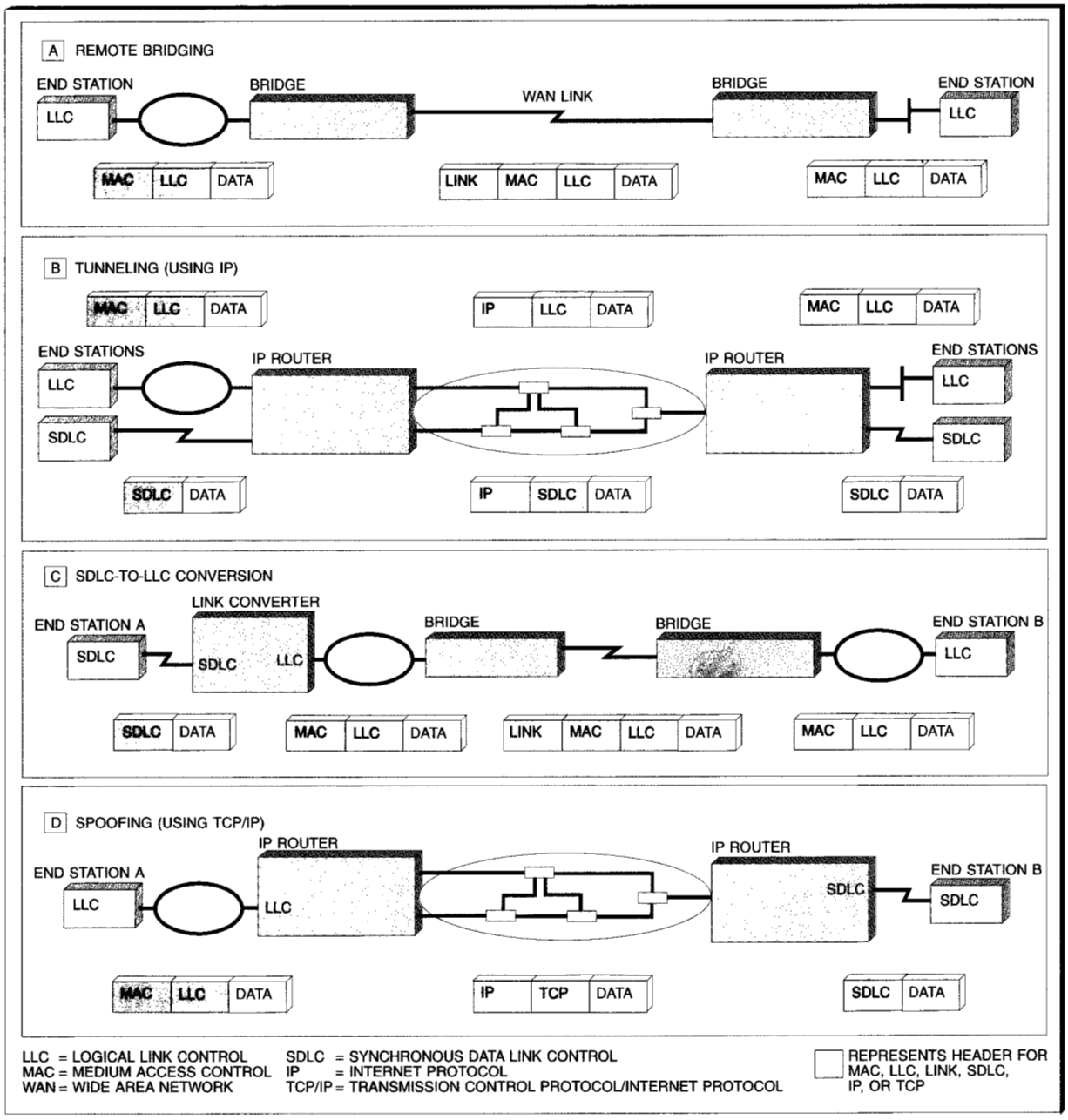
Remote bridging. With remote bridging, two local area networks (LANs) of the same or different types are connected by a long-distance physical link between two bridges.[9] Data link-layer procedures as defined by IEEE 802.2 logical link control (LLC)[10] operate between the LLC entities in the end stations—frames such as data acknowledgments flow end-to-end. SNA, NetBIOS, and all routed protocols on LANs use LLC frames, so bridging is protocol-independent. Depending on whether the bridge types are source-routing, transparent, or some combination of these, the bridges may or may not be visible to end stations, and may support differing medium access control (MAC) and physical layer types (e.g., token ring and Ethernet).[11]

SNA and NetBIOS both use LLC's connection-oriented service ("LLC type-2," or simply "LLC2"), which uses timed exchanges between LLC entities to detect connection failure. Since remote bridging extends these procedures across WAN links of restricted capacity with correspondingly increased delays, link congestion can result in logical link failure and SNA/NetBIOS session loss. These problems diminish as WAN link capacity increases. Another significant shortcoming is the nonproductive use of WAN bandwidth to carry LLC2 control frames.[12] Remote bridging also suffers from topology restrictions due to source-routing limits on the number of consecutive bridge "hops," and the possibility of network degradation from excessive bridge propagation of broadcast traffic.

Tunneling. In the tunneling approach, a router *encapsulates* the data link-layer frame within a packet of a network-layer routable protocol, typically IP. The resulting packet traverses the router network as a normal routed datagram, then a destination router removes the network-layer header and delivers the original frame to the target end station. As with remote bridging, data link control (DLC) procedures operate end-to-end, so two communicating end stations must use the same data link protocol (e.g., LLC or SDLC). Tunneling has the advantage of supporting non-LAN DLC protocols such as SDLC, which is widely used by older SNA devices. End-to-end operation of LLC2 and SDLC procedures means that tunneling also suffers the same problems of nonproductive WAN bandwidth utilization and congestion-induced data link timeouts as remote bridging.

SDLC to LLC conversion. Some network planners use SDLC-to-LLC conversion devices (generically called *link converters*) to ensure that LLC is the only DLC protocol to be carried across their multiprotocol network. LLC is preferable because its data transfer procedures are slightly more efficient than those of SDLC, it can run natively on frame relay links,[13,14] and because it can be bridged over WAN links directly onto destination LANs. In SDLC-to-LLC conversion, the link converter terminates SDLC procedures: it translates control frames between the two DLC protocols, acknowledges information

Figure 1 SNA and NetBIOS multiprotocol transport methods



frames, and transfers their data portion from one data link to the other. As shown in Figure 1C, end station B appears to end station A to be SDLC-attached, while A appears to B to be token-ring-attached running LLC. Because SDLC procedures operate between end station A and the link converter product, timeouts are a factor only on the LLC connection, which is either remotely bridged (as shown), tunneled, or carried natively on frame relay through the multiprotocol WAN. Another link

converter configuration, not shown, uses SDLC over the long-distance link and converts it to LLC on a central LAN.

Spoofing. Spoofing extends the practice of locally terminating DLC procedures to both sides of the WAN.[15] The two routers in Figure 1D each present to their local end station the appearance of the remote end station. The routers acknowledge connection-oriented frames locally, and use a reliable, sequenced protocol to transport information traffic across the WAN. Because DLC procedures are local, different DLC protocols may operate at each end, and the problems of link timeouts and WAN bandwidth erosion by DLC control frames are avoided. These advantages of spoofing over the other approaches described are significant, particularly for lower-speed WAN links, but they come at the cost of increased router complexity. The two routers participating in a spoofed connection must maintain and exchange connection state information, implement the data link layer functions of the end stations they represent, and support a transport protocol to deliver both datagram traffic and frames that they have acknowledged locally.

DLSw history and status

DLSw is a spoofing technology developed within IBM and first shipped in September 1992 in the 6611 Network Processor, an IBM router. It uses the Transmission Control Protocol (TCP), which runs on top of the Internet Protocol (IP) as the WAN transport protocol, and supports both SNA (on SDLC and LLC) and NetBIOS (on LLC) protocol flows. To avoid tedious manual definition of the associations between end-station resources and the routers that can reach them, DLSw provides a mechanism for routers to dynamically search the network for a target resource. A DLSw "switch-to-switch protocol" (SSP) defines search messages, as well as control messages that routers exchange to activate and deactivate a spoofed connection.

DLSw terminates the hop sequence recorded in a frame that reaches a DLSw router through a source route-bridged LAN. This means that the complete path between two end stations may include the maximum number of bridge hops[16] on each side of the WAN. For enterprises with large bridged campus LANs to interconnect, this can be an important characteristic.

In order to make its router technology openly available to the industry, IBM prepared a detailed specification of DLSw formats and protocols. In January 1993, the Internet Engineering Task Force (IETF), an industry/academic standards body, electronically published this document as an information-only RFC, RFC 1434.[17] Strong interest in this specification by various networking companies resulted in the mid-1993 formation of a multivendor DLSw interest group under the auspices of the Advanced Peer-to-Peer Networking* (APPN) Implementers' Workshop (AIW).[18] The goals of the interest group were to enhance RFC 1434 with a number of additional functions and publish it as an AIW standard. In December of 1994, the AIW accorded the revised specification final standard status under the title *Data Link Switching: Switch-to-Switch Protocol, Version 1*. The IETF subsequently published a reformatted version of the same document as RFC 1795.[19]

As its title suggests, the AIW DLSw standard limits its scope to defining the protocol flowing between DLSw routers. It describes the formats of all SSP messages, and defines finite-state machines to indicate what actions a DLSw product should take in response to both end-station DLC events and received SSP messages. It also defines rules for locally absorbing and remotely generating retries of certain datagrams—rules needed to ensure DLSw product interoperability.

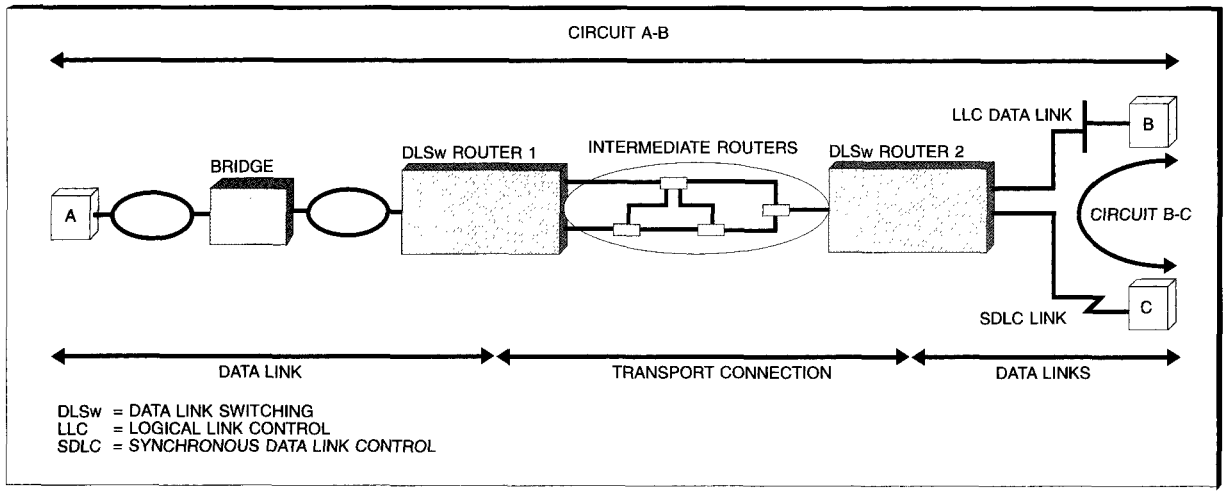
The DLSw standard briefly mentions *local* (single-router) switching between two data links. Since this function does not involve the SSP or affect product interoperability, the standard does not define its operation. Most DLSw products implement local switching of SDLC and LLC data links, in addition to remotely switching them via TCP/IP. This support allows the DLSw product to double as a link converter, providing the SDLC-to-LLC conversion functions previously discussed.

DLSw technical overview

In this section, we provide a technical overview of DLSw concepts and flows at the level of the Version 1 AIW standard. The term *router* is used for any product that implements DLSw, since most DLSw implementations are in routers.

General concepts. DLSw makes two communicating end stations each appear to the other to be directly adjacent on a shared data link. A *data link*

Figure 2 DLSw concepts



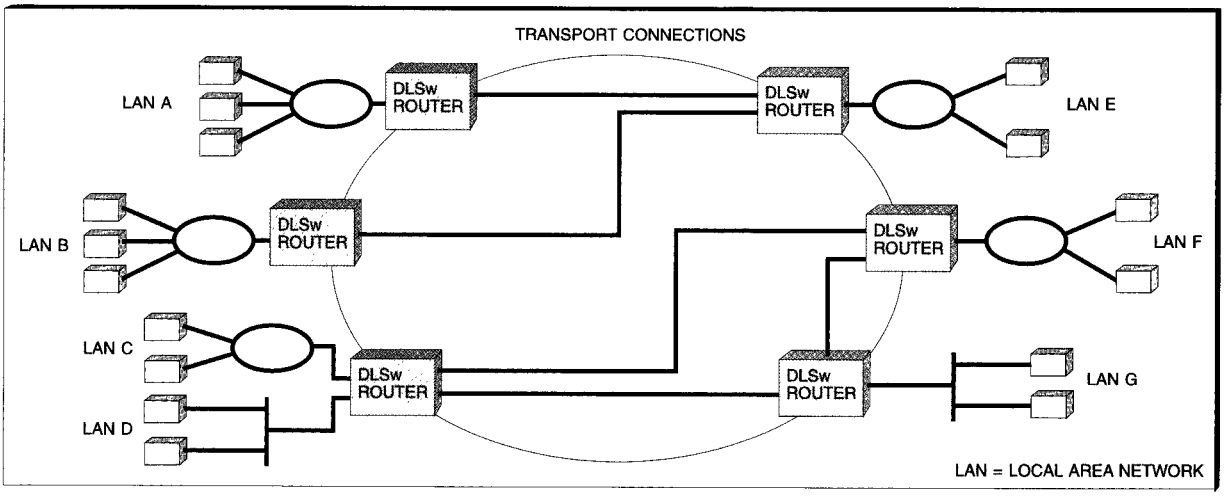
is an instance of procedures for exchanging information using OSI layer-2 functions, and corresponds to one of the following standard terms: LLC type-1 (connectionless service) logical data link, LLC type-2 data link connection, or SDLC data link.[20] DLSw concatenates two data links by terminating each and relaying the user data between them, either within a single DLSw router, or between two *partner* routers using a transport protocol such as TCP. A *circuit* is the end station-to-end station association of the two data links, as illustrated in Figure 2. *Transport connection* is a generic term for the reliable, full-duplex connection between partners. Multiple circuits can be multiplexed onto a single transport connection. The physical path for packets flowing on a transport connection typically includes intermediate routers that simply forward the packets and need not themselves support DLSw.

Two data links in a single circuit need not run the same data link layer protocol, or, when running LLC on LANs, use the same bridging discipline. In Figure 2, the circuit between end stations B and C comprises one LLC data link (from Router 2 to end station B) and one SDLC data link (from Router 2 to end station C). The circuit between end stations A and B has one data link from Router 1 to end station A) on a source route-bridged LAN, and another data link (from Router 2 to end station B) on a transparently bridged LAN. In each case, the router must be able to handle the DLC type and bridging discipline necessary to communicate with the supported end station.

When a DLSw node starts up, it establishes long-lasting transport connections with a user-defined set of partner DLSw routers. As shown in Figure 3, partner relationship topologies need not be fully meshed, and a single network may support disjoint sets of partnerships. These relationships determine which sets of end stations will be able to communicate. End stations on LANs A and B are able to find and communicate with end stations on LAN E, but cannot reach those on LANs C, D, F, or G. DLSw routers do not forward searches or map circuits between two transport connections, so end stations on LANs A and B cannot reach each other. Because the routers supporting LANs C, D, F, and G are fully meshed, however, all end stations on these LANs can reach all others. All six DLSw routers shown are on the same network and can transfer normal routed traffic to each other, but DLSw searches and traffic can flow over only defined DLSw transport connections.

After establishing its transport connections, the DLSw function is normally passive, acting to establish and take down circuits only in response to attempts by an end station to send frames or set up data links. When an end station (the *origin* station) first starts to send frames to a given *target* end station, the DLSw function in the router adjacent to the origin station initiates a search for the desti-

Figure 3 DLSw partner relationships



nation station, by sending to some or all of its partner routers certain SSP messages defined for locating remote resources. (The origin router may also search its local ports for the target station, in case both the the origin and target station are local.) The partners, in turn, search for the target station on their local LAN and SDLC ports, and return other SSP messages if the search is successful. All routers involved in a search may cache (retain in a database) information about which DLSw routers are serving various end stations, so the scope of future searches for the same stations can be more limited. DLSw implementations generally require very little manual definition of end-station resources, and instead rely on searching and caching to build and maintain a dynamic directory of resource locations.

Once a search has identified a pair of DLSw partner routers (or a single router performing local switching) that can provide a path between the origin and target end stations, subsequent frames from the origin station trigger the establishment of a circuit between the DLSw partner routers on that path. By exchanging certain SSP messages defined for circuit establishment, partner routers can each establish internal control blocks representing the circuit, and efficiently route all data frames flowing between the two end stations.

The DLSw standard defines a number of circuit states to describe the protocol of circuit management. If there is no circuit between a given pair of end stations, their association is in a *disconnected* state. If two end stations exchange only datagram traffic, their circuit reaches the *circuit_established* state, which provides for the equivalent of LLC type-1 services. If the end stations send DLC control frames to set their data link into a connection-oriented mode, the DLSw routers on the circuit exchange SSP messages to move the circuit into a *connected* state. The standard also defines a number of intermediate states for managing both network searches and circuit establishment and takedown. DLSw routers exchange SSP messages to disconnect a circuit when one of the end stations requests a disconnect of its data link, or when certain protocol errors occur.

As data flows on established circuits over a transport connection, it is multiplexed with data for other established circuits, and also with search and circuit-control messages. All messages flowing on a transport connection are SSP messages and include standard SSP header addressing fields to distinguish the circuits from one another. There are two SSP header formats: a longer control header containing native end-station addresses used for searching and setting up circuits, and a streamlined information header containing circuit identifiers exchanged during circuit setup. Generally speaking, the longer header is used in SSP search and circuit-control messages, and the shorter header is used in SSP messages that carry user data.

DLSw models all end stations as LAN-attached devices running LLC, so they are each known by a six-byte medium access control (MAC) address. Multiple circuits between two MAC addresses are distinguished by 1-byte link service access point (LSAP, more commonly just SAP) values assigned at each end station, so a circuit is uniquely identified by the 4-tuple (local MAC address, local SAP, remote MAC address, remote SAP). SDLC devices, which are natively known by a 1-byte SDLC station address, are assigned a MAC address and SAP within their attached DLSw router. This MAC/SAP pair represents the SDLC station to all other end stations and routers within the DLSw network.

On a LAN, a DLSw router receives frames sent to the MAC addresses of all of the end stations it is representing, or for which a search is taking place. To source-routing end stations, DLSw appears to be a bridge to a "virtual" LAN segment on which all DLSw-reachable end stations reside. This approach maximizes the number of bridge hops available on the LAN side of the router, as previously mentioned. On transparently bridged LANs, the DLSw router also functions like a bridge and is therefore not visible to end stations.

With these general concepts as background, we are now in a position to examine certain DLSw topics in greater detail.

Transport connections. The DLSw standard does not define how a DLSw product determines which IP routers in its network are DLSw-capable, nor how to know the subset of those with which it should establish a transport connection. The most common implementation choice is to have the user configure at each router a list of partner routers, identified by IP address. When a DLSw router starts up, it attempts to bring up a TCP connection with each partner router in its list. It is usually necessary for only one of a pair of routers, not both, to have the other in its partner-router list.

A more automated approach, implemented in some vendors' DLSw products, is to use multicast IP facilities to determine which other routers should be partners.[21] Because partner discovery procedures are outside the scope of the standard, this method of reducing static partner definition is vendor-specific and must be supplemented with the usual partner list approach in order to interoperate with all vendors' DLSw products.

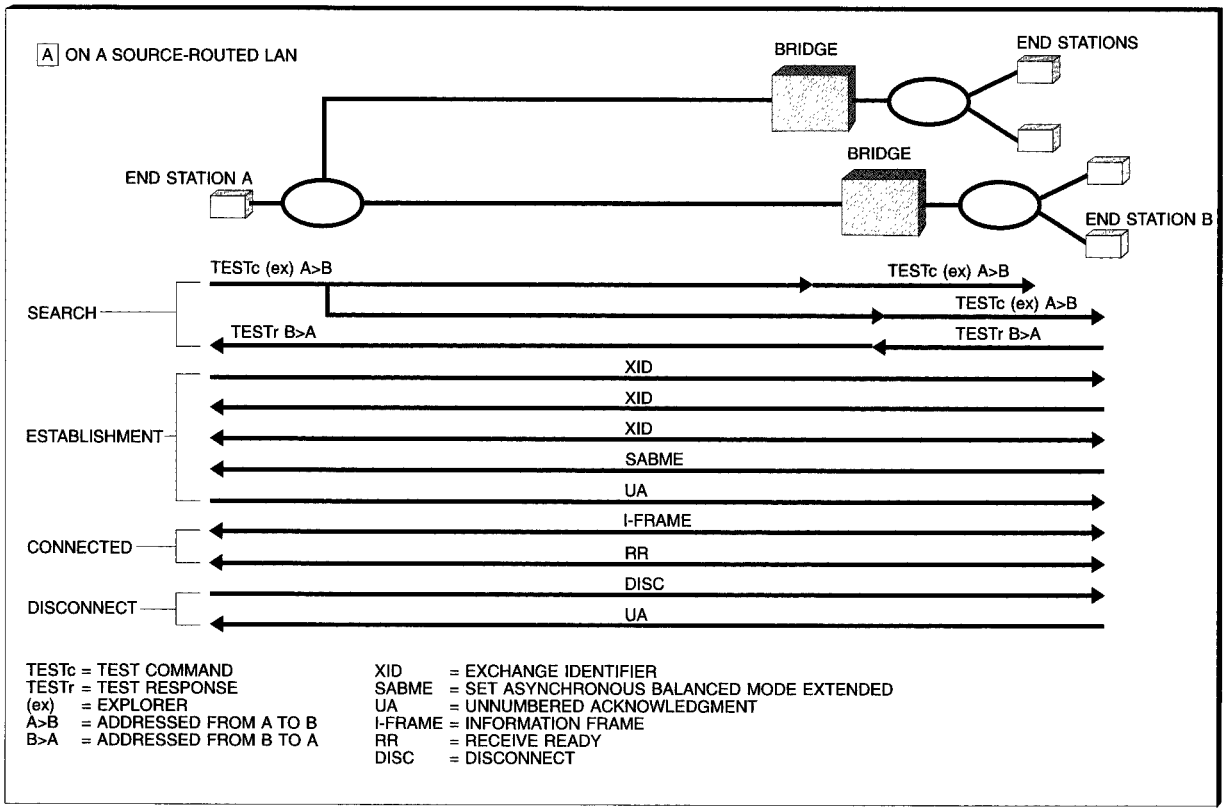
Once established, DLSw transport connections are normally long-lasting. As explained earlier, they are used not only for active-circuit data transfer but also for carrying network search messages, and therefore need to be kept active.[22] A DLSw router typically tries periodically to connect to all inactive partners, both those it has yet to reach and those with which it has lost a previously established transport connection. TCP transport connections may go down when one of the partners has a software or hardware fault, in response to an operator action, or when an intermediate IP router fails and there is no alternate path around the failed router. When an alternate path exists, the IP routing protocol operating in the network detects the loss and finds the new path without disruption to TCP connections.[23]

It is common to have the user configure at each router a list of partner routers identified by IP address.

A DLSw transport connection using TCP consists of two TCP connections, each used for data transfer in a single direction. A TCP connection is by definition full-duplex, but the choice of two simplex-mode connections was found to be preferable in the first DLSw implementation.[24] The AIW standard requires the initial establishment of a TCP connection pair between any two partners, but provides an optional mechanism for the partners to jointly switch to using a single full-duplex TCP connection. In all cases, DLSw TCP connections use well-known TCP port numbers on both sides of the connection.

Once the TCP connections are established between two partner routers, DLSw functions as the TCP "application" at each end, sending and receiving both control and information SSP messages for any number of circuits over the transport connection. The two DLSw routers exchange only SSP messages over the transport connection; each quietly discards any

Figure 4a Example SNA flows



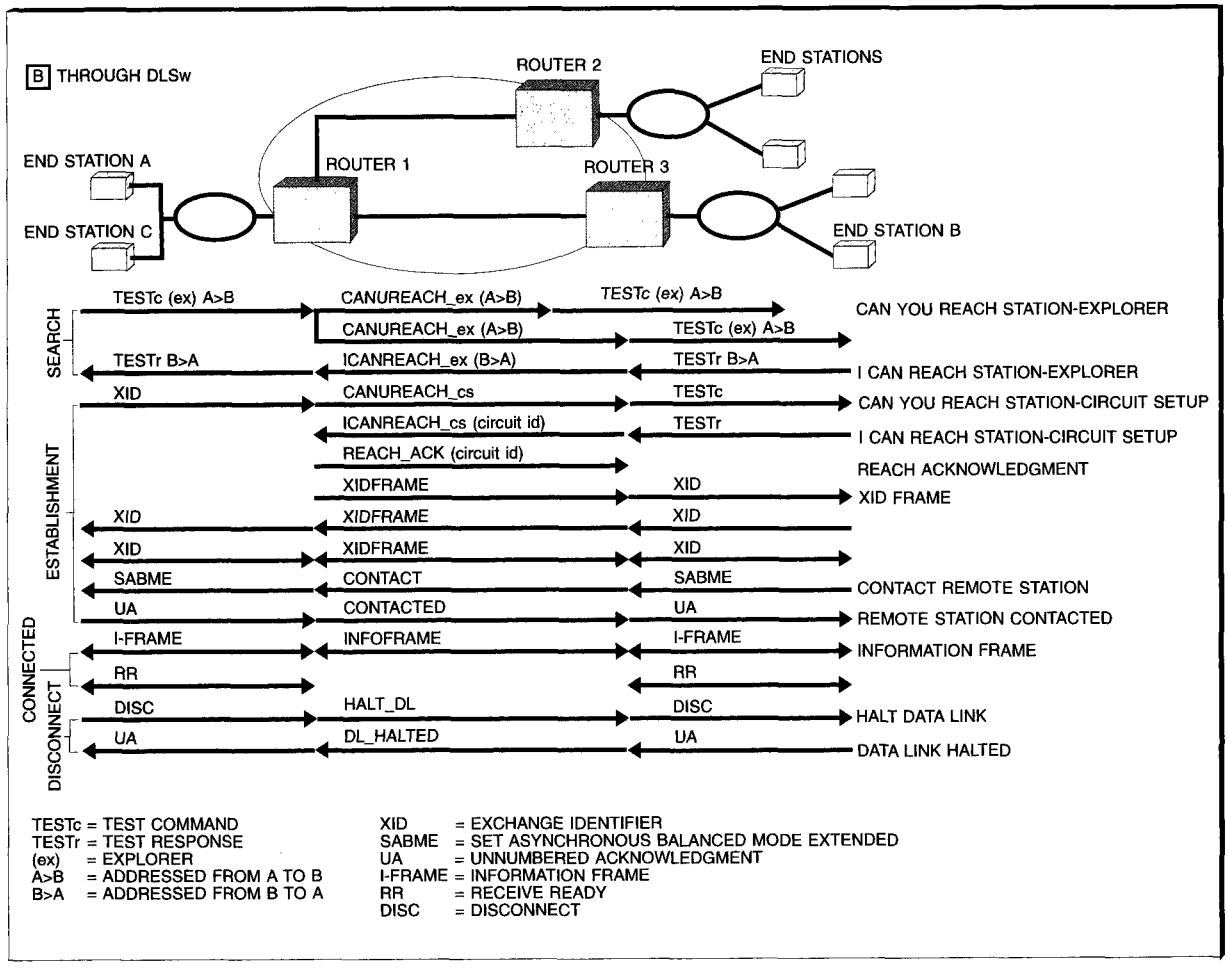
received data that contain an unrecognized SSP message type.

TCP provides sequenced, reliable, flow-controlled delivery of SSP messages in DLSw, but it models their transfer as a stream of bytes rather than as intact application messages. When a sending TCP chooses the number of bytes to send in its next *segment* (which will become one or more IP packets), it pays no attention to SSP message boundaries. SSP messages may be split across multiple TCP segments, or combined into a single segment, depending on timing conditions and the current congestion state of the connection. In a receiving router, DLSw simply reads bytes from TCP until an entire SSP message has been received (as known from a length field in the SSP message header), and then processes the message.

Capability exchange. After two DLSw partners have initialized the transport connection between them, they each pass information about their identity and capabilities to the other. This information exchange facilitates the interoperation of products with different levels of support for the base DLSw protocol, and with different sets of the options defined by the AIW. If a product does not send its capabilities, it is assumed to support a version of DLSw that preceded the AIW standard.[25]

Capabilities information is carried in an SSP message named cap_exchange, so this is the first message that each node sends on a new transport connection. An initial cap_exchange must contain the following information: a vendor ID, to indicate whose software is running; a number for the version of the AIW standard supported; an initialization value for the flow control algorithm; and a list of the LLC SAPs supported by the sender. Of these parameters, the cap_exchange receiver is required to operationally use only the initial pacing window size. The vendor ID and version number normally serve only as informa-

Figure 4b Example SNA flows



tion for problem determination. A DLSw implementation receiving cap_exchange may optionally use the SAP list to filter out unnecessary WAN searches in its role as an origin DLSw router. Because SAP values are protocol-specific, the SAP list also indicates which protocols the node intends to support using DLSw (i.e., whether it supports NetBIOS).

It is optional to send and be able to use the following information in an initial capability exchange: a free-format text string to identify the version of the sending software; the desired number of TCP connections; a list of MAC addresses for SNA end stations local to the sending partner; a list of NetBIOS names local to the sending partner; and any vendor-defined capabilities. If both partners indicate they prefer a single TCP connection, they drop one of the two they have established and begin to use the other in full-duplex mode. The MAC address and NetBIOS name lists may be used both to preload the receiving partner's directory cache, and to provide user control over which partners the receiver will use when searching for destination resources. These lists are intended to contain only user-defined resources, not the sending router's dynamically cached directory information. Vendor-defined capabilities allow DLSw products from the same vendor to signal support for a value-added feature that is not part of the DLSw standard. They also allow other vendors to support a vendor-defined feature for which the originating vendor has provided a specification.

The receiver of cap_exchange returns a positive or negative acknowledgment of the message, using the same SSP message type but different contents. The receiver may only accept or reject its partner's capabilities; it cannot negotiate an alternate set of parameters. The receiver of a negative capabilities-exchange response takes down the transport connection.

After both partners have sent their capabilities and received the other's positive response, the transport connection may be used to carry all the other SSP messages for searches, circuit control, and data transfer. Since it is possible that some of the reported capabilities of a product may change after the initial capability exchange, products may optionally support the sending and receiving of cap_exchange at any time following the initial exchange. The only information allowed in a "runtime" capability exchange is: the list of supported SAPs, the lists of local MAC addresses and NetBIOS names, and vendor-defined capabilities. Each of the lists replaces any previous copy sent to the partner router.

SNA circuit control. To understand how DLSw routers manage SNA circuits, it is instructive to compare the frame flows between two end stations on a single data link to the same flows passing across two DLSw partner routers. Figures 4A and 4B provide such a comparison, using as an example end stations on a source route-bridged token-ring LAN. In both the DLSw and pure LAN environments, connections pass through four phases: a *search phase*, where a path to the destination station is found; an *establishment phase*, where datagrams are sent along that path; a *connected phase*, where data are exchanged using reliable connection services; and a disconnect phase, where the connection is destroyed.

Search phase. A search on a source-routed LAN begins when the origin end station sends an explorer[26] TEST command frame addressed to the MAC address of the target end station.[27] The target station receives a copy of the TEST command for every path taken through the bridged network, and sends a TEST response for each copy it receives. The origin station selects one of the TEST responses, typically the first one received, as representing the best path to the target station, and saves from the TEST response the route the frame followed through the bridged network. Thereafter, both the origin and target stations exchange frames using this specific route.

The DLSw equivalent of route discovery is to identify which of its partner routers can reach the target end station through their LAN or SDLC ports, and select one of those as being on the best path to the target. When DLSw Router 1 in Figure 4B receives the explorer TEST command from the origin station, it sends an SSP message canureach_ex (explorer) to every one of its partners. This message contains the addresses of the origin and target end stations from the original TEST command. Each of the partners for Router 1 checks to see if it can reach the target end station through any of its local ports. For LAN ports, this may involve building and sending a new TEST command, as shown. For SDLC ports, this may mean polling an attached station in some other way. Partners that find they can reach the target end station (only Router 3 in Figure 4B) respond to the origin router with an icanreach_ex message. The origin router typically selects the first partner to respond with icanreach_ex as the best route to reach the target end station.

DLSw implementations make extensive use of caching to reduce the need for full broadcast searches such as the one just described. To extend the same example: if, shortly after the sequence shown, end station C were to send Router 1 a TEST command addressed to end station B, Router 1 may choose to send back a TEST response immediately (without sending canureach_ex) because it has already cached a destination router for end station B. Router 1 may also choose to send canureach_ex only to Router 3 rather than to all partner routers. In addition, an origin router can use MAC address lists received from its partners during capability exchange to direct canureach_ex messages to specific destinations instead of performing full broadcast searches. On the destination side of this example, if Router 3 were to receive another canureach_ex for end station B, it may choose to send icanreach_ex immediately, or send a TEST frame out on a particular cached LAN port rather than all local LAN ports. Caching and search algorithm choices such as these are purely implementation-specific; they are not defined by the DLSw standard.

Establishment phase. SNA end stations always use the connection-oriented services of their DLC, so the next phase in LAN-based link establishment is

to establish the LLC connection. As shown in Figure 4A, end stations A and B send each other exchange identifier (XID) frames to report and negotiate operational characteristics of both the data link and SNA protocol layers. When the XID exchange is complete, one of the end stations sends a Set Asynchronous Balanced Mode Extended (SABME) frame to set the data link into a connected state. The other end station responds with an Unnumbered Acknowledgment (UA), and the data link is then connected. XID, SABME, and UA frames themselves all flow as connectionless frames—end stations resend some of these frames several times to increase the likelihood of delivery.

DLSw interprets the first directed XID from an origin end station to mean that a series of connectionless frame flows is beginning, and that a circuit must be established to carry them. In our example, Router 1 sends a canureach_cs (circuit setup) only to Router 3, the destination router previously associated with end station B. Router 2 checks that it can still reach end station B, and sends back icanreach_cs. In this message and in the reach_ack acknowledgment that Router 1 then returns, Routers 1 and 2 exchange locally significant circuit identifiers. These identifiers provide the addresses for an established circuit, and are used by each router to efficiently associate the circuit with a local router port and data link. All subsequent SSP messages for a circuit carry the circuit identifier by which the destination router for the message knows the circuit.

Once reach_ack flows, the circuit has reached circuit_established state and is ready to carry connectionless traffic such as SNA XIDs. DLSw routers use the SSP message xidframe to transport the actual SNA XID frames between partners. Unlike TEST, SABME, and other DLC frames with purely semantic significance, the data in this frame are needed by the SNA protocol layer in each end station. None of the other SSP control messages discussed so far carries the actual data link frames that caused them.

To move a circuit into connected state, DLSw carries the DLC set mode command and response exchange (for LLC, this is the SABME/UA exchange) between partners using the SSP messages contact and contacted. The SDLC equivalents to SABME and UA are Set Normal Response Mode Extended (SNRME) and UA.

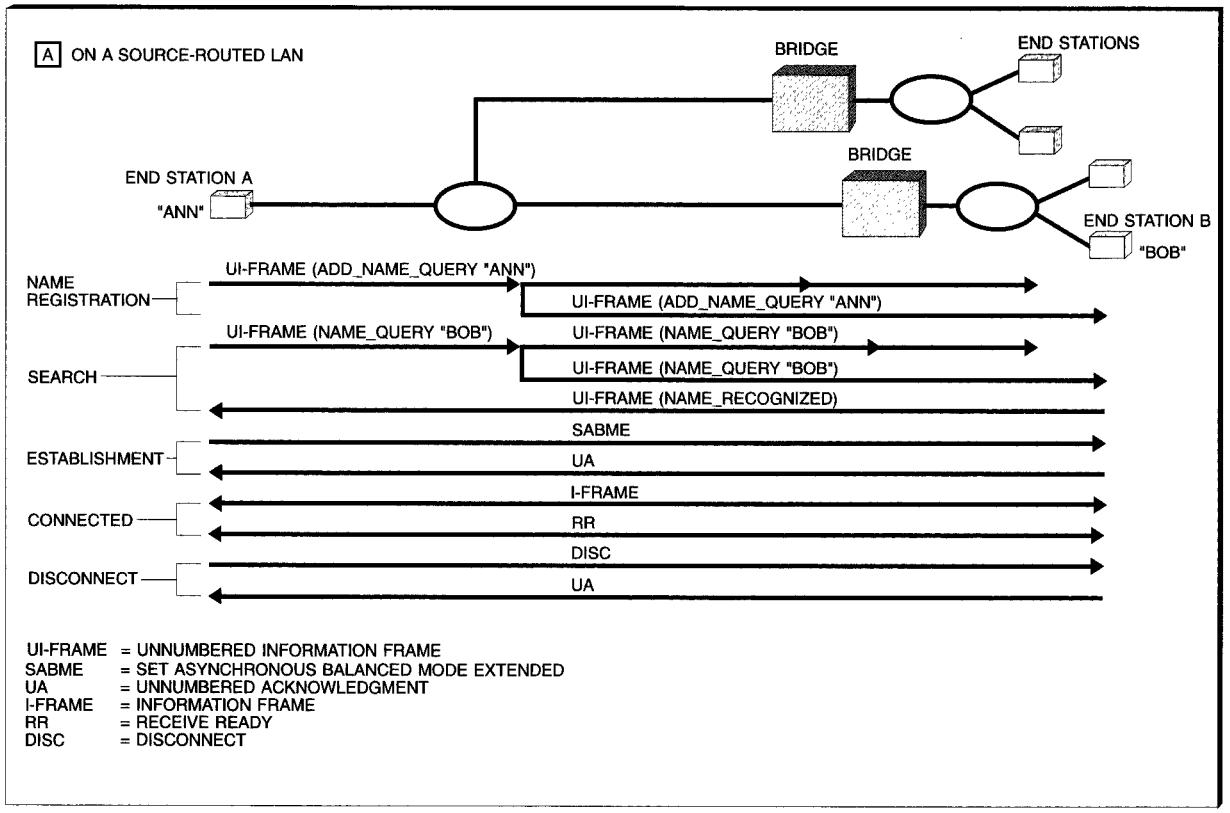
Whenever DLSw is sending to a remote partner the SSP equivalent for connectionless frames such as TEST, XID, or SABME, the origin-side DLSw discards retries of the same frame sent by the origin end station. The destination-side DLSw generates retries as required to ensure delivery on the destination data link. For example, a SABME sent four times at four-second intervals by end station A would result in a single contact SSP message from Router 1 to Router 3, and Router 3 might send its SABME three times at one-second intervals. This approach results in efficient WAN utilization and locally appropriate retry policies.

Connected phase. The search and connection establishment phases are brief, transitory phases leading to the connected phase, where an end station activates SNA sessions and transfers real end-user data. SNA control traffic and user data all flow on data links as information frames (I-frames), and are transported between DLSw partners using the SSP message infoframe. The SSP header for infoframes is considerably shorter than the one used for SSP control messages, because it can always address a circuit using an established circuit identifier.

Disconnect phase. When an LLC end station wishes to terminate one of its existing connections, it sends a disconnect (DISC) frame to the other end station, which responds with UA. The same frame types are defined for SDLC. With DLSw, these frames are reflected between partners using the SSP messages halt_dl (halt data link) and dl_halted (data link halted) and the circuit is then *disconnected*. Another DLSw disconnect scenario is the loss of a transport connection due to intermediate router failure. When a DLSw node detects such a failure, it performs a local disconnect of all connected data links that were using the failed transport connection.

NetBIOS session and circuit control. DLSw support for NetBIOS differs from its SNA support in several important respects. NetBIOS operates only on top of 802.2 LLC, so there is no need to consider other data link types such as SDLC. NetBIOS has name registration and resolution procedures that require NetBIOS-unique SSP message flows. NetBIOS applications make heavy use of datagrams in addition to connection-oriented LLC services, and frequently broadcast them to multiple NetBIOS end stations at the same time. From an implementation point of view, supporting NetBIOS is more of a challenge than supporting SNA because of these characteristics, and because there is wide

Figure 5a Example NetBIOS flows



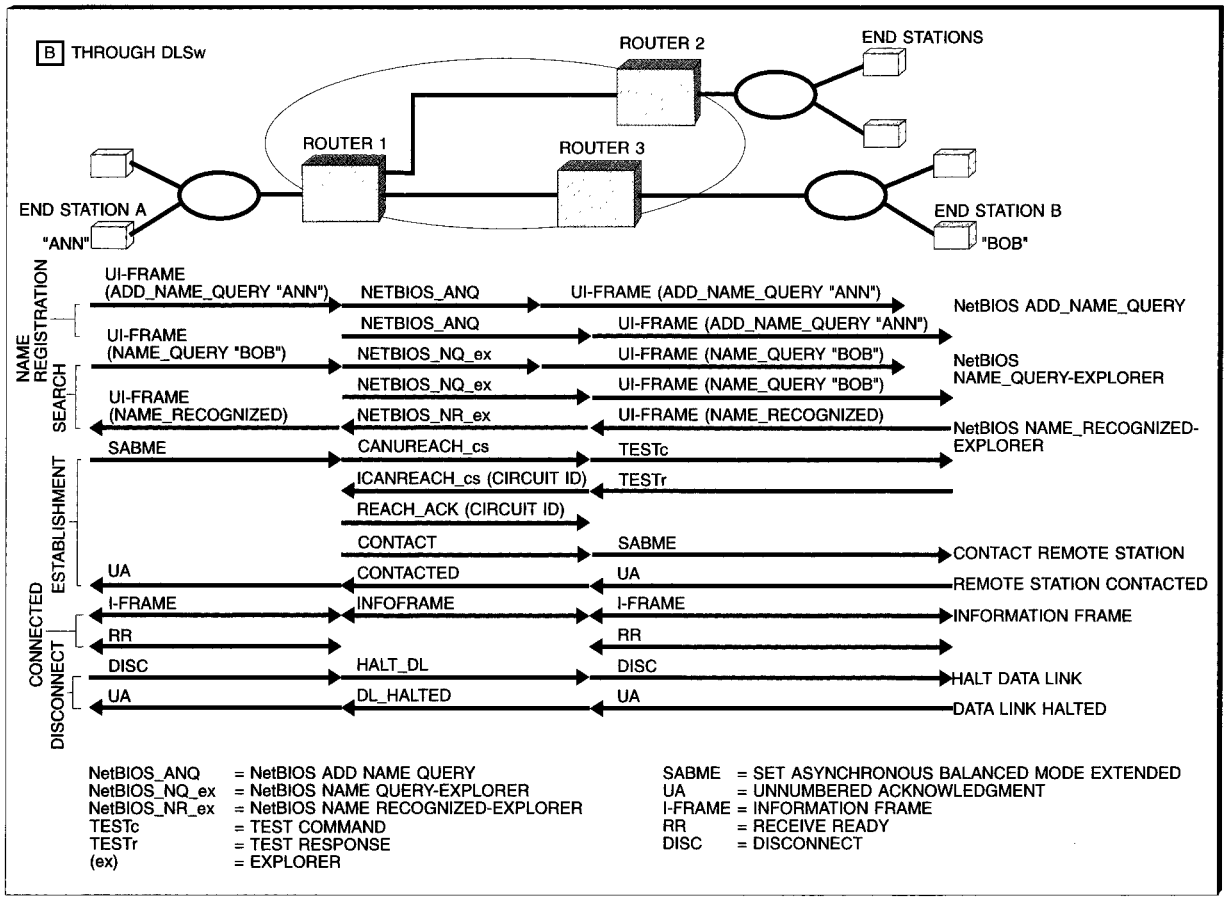
variability in how individual applications generate NetBIOS frame flows.

It will again be useful to compare normal LAN frame flows with the same sequences transported through DLSw. Before doing so, we review a few basic NetBIOS concepts and terms. NetBIOS application resources are defined by names, which are 16 bytes in length. A *unique name* may exist at only one NetBIOS end station in a network, while a *group name* may be shared by multiple end stations. An application may communicate using datagrams, or by issuing a *call* to establish a connection-oriented *session* with another application. Sessions may exist only between applications using unique names; group name traffic is datagram-based. Multiple concurrent sessions may exist on a single LLC connection between two NetBIOS end stations, and even between two applications. Only one LLC connection may carry NetBIOS traffic between any two end stations. The end stations connect it when establishing the first session between them, and disconnect it when the last session is ended.

Figures 5A and 5B show example LAN frame flows between two NetBIOS end stations, and the same flows as they are transported by DLSw. In this example, one application becomes active, calls the other to establish the only session between the two end stations, sends application data, and then ends the session. The different phases in this scenario are *name registration*, *name search*, *connection establishment*, *connected*, and *disconnect*.

Name registration phase. When the application named "Ann" becomes active, it broadcasts its name to all NetBIOS end stations in the network, to make sure that no other application is using the same name. No reply indicates that there is no name collision. The NetBIOS code in end station A sends

Figure 5b Example NetBIOS flows



an LLC Unnumbered Information frame (UI-frame) with the NetBIOS type ADD_NAME_QUERY to effect the name registration check. This frame is addressed to a special LAN group address to which all NetBIOS end stations listen. NetBIOS repeats the broadcast several times to ensure the datagram is received (retries are not pictured).

A DLSw router receiving ADD_NAME_QUERY forwards this frame to every one of its partner routers using the SSP message netbios_anq. This message goes to every partner because its purpose is to detect name collisions anywhere in the network.[28] The partner routers each broadcast the transported ADD_NAME_QUERY frame onto their destination LANs. As with datagrams used in SNA connection establishment, the origin DLSw is responsible for discarding retries of this and other NetBIOS control messages, while the destination DLSw router must generate them.

Name search phase. At some time after the name registration phase, application "Ann" does a *call* to application "Bob." NetBIOS broadcasts a UI-frame with type NAME_QUERY to find the end station in the network that hosts the unique name "Bob." It sends this frame to the NetBIOS group address so it is received by all NetBIOS stations, but only the station with "Bob" responds with the UI-frame NAME_RECOGNIZED.

DLSw treats a NAME_QUERY as a request to find a partner router that can reach the specified destination name. The origin DLSw router uses the SSP message netbios_nq_ex to forward the NAME_QUERY frame to some set of its partner routers. It chooses

this set of partners based on NetBIOS name lists it received during capabilities exchange, and on name/partner associations cached during previous name searches and name registrations. Upon receipt of the netbios_nq_ex, each of these partners broadcasts the forwarded NAME_QUERY frame on its LANs and responds to the origin router with the SSP message netbios_nr_ex if any end station responds to it with a NAME_RECOGNIZED. The origin router forwards this NAME_RECOGNIZED frame onto its LAN upon receipt of netbios_nr_ex. Unlike SNA searches, the origin router cannot construct a NAME_RECOGNIZED response to the original NAME_QUERY based on cached information; the actual NAME_QUERY frame must always flow to the destination end station because it contains session-specific data for the NetBIOS protocol layer.

Establishment phase. Once end station A has learned that "Bob" is located at end station B, it can see that it does not yet have an LLC connection with B. End station A initiates the required LLC connection by sending SABME, as described previously for SNA connection establishment. If A and B already had an LLC connection, end station A would simply start sending the I-frames that initialize a new NetBIOS session. The NetBIOS types for these I-frames are not relevant to LLC or to DLSw.

In the DLSw case, the SABME arrives at Router 1 when there is no established circuit between end stations A and B. Receipt of a SABME indicates that a circuit should not only be established but also connected. Router 1 sends canureach_cs to initiate circuit establishment and then contact after the circuit is established, to connect it. Both routers handle these flows exactly as they would for setting up an SNA circuit. If A and B were already connected by a NetBIOS circuit, the I-frames that A sends to set up a new session would simply flow as infoframes on the existing circuit.

Connected phase. While connected, end stations A and B exchange I-frames and possibly UI-frames carrying application data. On a LAN, the UI-frames do not flow on the LLC connection, but may follow a specific route between A and B. With DLSw, both of these are carried over the existing circuit; I-frames as SSP infoframes, and UI-frames as SSP dgrm-frames.

Disconnect phase. When either application decides to end the NetBIOS session between them, it sends a certain I-frame that signals session-end. In our example, application "Ann" ends the last (only) session using the LLC connection A-B, so end station A follows the session end I-frame with a DISC frame to end station B. With DLSw, these flows translate to infoframe followed by the normal halt_dl/dl_halted exchange.

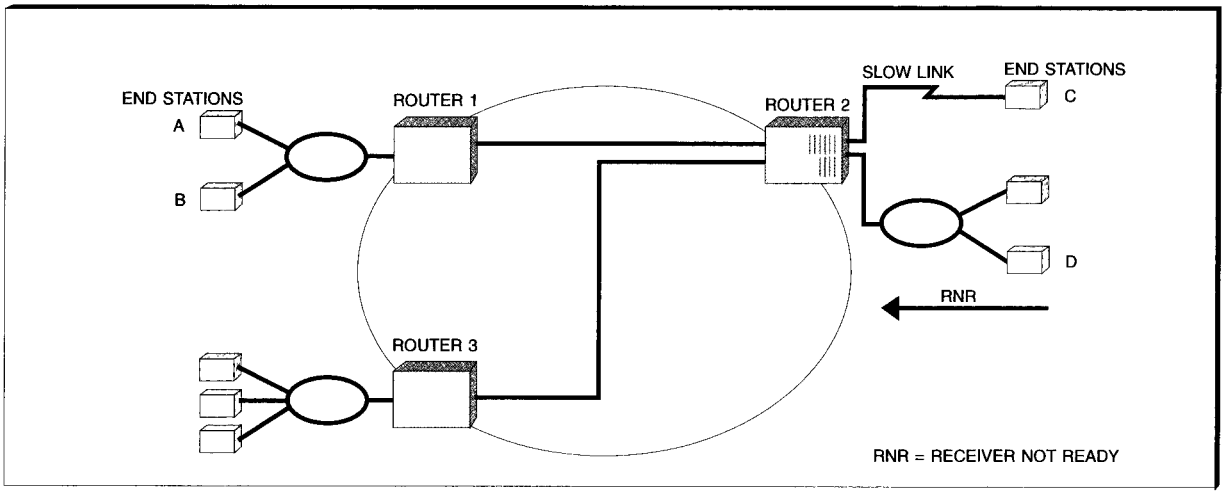
Other NetBIOS topics. The following additional topics are relevant to NetBIOS, but have no corollary in SNA.

Group name registration. When an application using a group name starts, NetBIOS broadcasts an ADD_GROUP_NAME_QUERY UI-frame to see if any NetBIOS station is using the new name as a unique name. Like ADD_NAME_QUERY, a DLSw router must forward this frame to all its partners. If another NetBIOS station detects a name collision, it broadcasts a NAME_IN_CONFLICT UI-frame. A DLSw router forwards both of these frames to its partners using dataframe, the SSP message for datagrams not being sent on an existing circuit.

Broadcast datagram traffic. As mentioned earlier, NetBIOS makes heavy use of datagrams for sending application data, especially for applications using group names. On a LAN, these datagrams are LLC UI-frames which are normally broadcast to the NetBIOS group address. They may be sent to and from both unique and group names. With DLSw, the origin router forwards these frames to some set of its partner routers using the SSP message dataframe. The origin router chooses the set of partners based on NetBIOS name lists it received during capabilities exchange, and on name/partner associations cached during previous name searches and name registrations.

Flow control. In addition to defining mechanisms for searching and setting up circuits, DLSw provides a way for partner routers to control the flow of user data on established circuits. The TCP connections that carry DLSw circuits run through devices and links that are of varying capacity and are subject to congestion. In addition, the end stations and data links being connected by a DLSw circuit may be mismatched in terms of the data rate they can support. While TCP itself contains mechanisms for flow control and congestion avoidance,[29] these affect equally all DLSw circuits sharing a TCP-based transport connection. There are several common circumstances in which it is desirable to reduce the flow of a single circuit without affecting the other circuits on the same DLSw transport connection.

Figure 6 DLSw flow control problems



Two examples of such situations are shown in Figure 6. On the circuit between end stations A and C, the high-speed token-ring data link is capable of overdriving the low-speed SDLC data link. Assuming a transport connection between Routers 1 and 2 with a greater capacity than the SDLC link from Router 2 to end station C, a heavy burst of traffic from A to C would begin to deplete the buffer space available in Router 2.[30] In another scenario, end station D is experiencing internal congestion and has signaled Router 2, using a Receiver Not Ready LLC control frame, to stop sending frames on data link B-D. Until end station D sends Router 2 a Receiver Ready (RR), Router 2 must buffer any data it receives on circuit B-D from Router 1. Depending on the buffering implementation of Router 2, congestion caused by these situations could affect other circuits sharing the transport connection between Routers 1 and 2, or even circuits carried on other transport connections of Router 2.

To address problems such as these, DLSw includes a mechanism for a router to *pace* the data flow it is receiving from a partner router on an individual circuit. DLSw circuit pacing uses many of the same concepts as Advanced Peer-to-Peer Networking (APPN) adaptive session-level pacing.[31] For each direction of data flow on a circuit, the receiving router must explicitly grant permission for its partner to send a certain number of SSP messages, called a *window*. As long as the receiver remains uncongested, it continues to grant permission for a new window as it receives each window, and its partner may continue to send messages. When the receiver becomes congested, however, it may withhold permission for a new window, and its partner must stop sending at the end of the current window. The receiver may also choose to reduce the size of the new windows it grants, or asynchronously signal its partner to stop sending immediately.

Two DLSw routers set the initial pacing window sizes (one in each direction) for a circuit during initial capability exchange—all circuits on a given transport connection use the same initial window size. Window grants, acknowledgments, and size change operators flow in both types of SSP message headers, and may flow piggybacked with user data or, under certain conditions, in an independent flow control SSP message. Both infoframes and dgrmframes are paced on a circuit. DLSw implementations are not required to vary the pacing window size in their role as a receiver, but must, as a sender, react appropriately if their partner changes the window size. In other words, a DLSw product may support either *fixed* or *adaptive* pacing as a receiver, but must support *adaptive* pacing as a sender.

Using circuit pacing, DLSw implementations can slow down the flow of data on slow or stopped circuits before congestion begins to unfairly affect other circuits. In the two examples previously given, Router 2 could reduce the window size (if it supported adaptive receive logic) or withhold a

grant of a new window to Router 1 on circuits A-C and B-D, as soon as it recognizes that the corresponding data links are overdriven or stopped. Router 1, in turn, may choose to send receiver-not-ready (RNR) frames to end stations A and C, to temporarily stop them from sending frames that Router 1 cannot immediately deliver onto its transport connection to Router 2. Because RNR does not stop an end station from sending UI-frames, Router 1 can only discard or buffer these frames if it has no permission to send them on to Router 2. DLSw defines only the pacing formats and mechanisms that operate between the routers: how to use them with a particular buffering scheme and how they interact with data link receiver ready (RR) and RNR flows is left to the router implementation.

Circuit priority. DLSw provides a simple, optional mechanism by which the two routers setting up a circuit may adopt a transmission priority—low, medium, high, or highest—for the data of that circuit. During canureach_cs/icanreach_cs exchange, each router indicates the priority it would prefer for the circuit. How a router selects this preferred value is up to the implementation; presumably, it would be based on user definition by MAC address. The origin router chooses one of the two values from the canureach_cs/icanreach_cs exchange and reports its decision in the reach_ack. The destination router is obliged to accept the choice of the origin router as the established priority for the circuit. The selected priority applies equally to all frames flowing on the circuit. Within an SNA circuit, for example, the routers do not distinguish between interactive and batch-oriented SNA session traffic.

Only the two routers on the periphery of the IP network (the two DLSw partners) can actually move high-priority traffic in a circuit ahead of data from a lower priority circuit. As discussed earlier, SSP message headers may be located at any position within a TCP segment or IP packet, so they are not visible to intermediate routers and cannot be used to trigger specialized DLSw forwarding logic such as circuit prioritization. The packet header which *is* visible allows an intermediate router to distinguish DLSw traffic from other IP traffic by virtue of its well-known TCP port numbers. Hence, an intermediate router can (and some products do) prioritize DLSw traffic with respect to the other protocols sharing a particular WAN port, but it cannot distinguish circuits within the DLSw traffic stream.

As a result, if the most congested links in a network configuration are between intermediate routers, DLSw circuit priority is effective only to the degree that TCP in the "edge" DLSw router reflects the congestion back to DLSw.

Network management. As a router-based technology, DLSw is managed using the management protocol prevalent in router networks: the Simple

Strong vendor and user interest in DLSw means that DLSw will be in use for several years.

Network Management Protocol (SNMP).[32] SNMP defines the flows between a management station, typically a workstation running a graphical network management application, and an *agent* residing in the product to be managed. In order for the application user to monitor and change the operation of the DLSw function in a router, DLSw must make available to the SNMP agent of the router, a set of managed objects that are collectively termed a Management Information Base (MIB). An AIW/IETF working group is in the process of defining a standard MIB for DLSw[33] so that a single management application written to read and write the objects in this MIB should be able to control any compliant DLSw implementation.

The DLSw MIB provides a management application with read access to extensive information about the capabilities and operational state of the DLSw function for a product. This information includes: the identity of the DLSw node and a list of the functions it supports; a list of current partner routers, along with the capabilities of each and associated traffic statistics; a list of currently cached MAC addresses and NetBIOS names, with the remote partners serving each one; and a list of all circuits, with pointers to LLC or SDLC MIBs containing their traffic statistics.

In addition, the DLSw MIB allows a management application write access to a number of objects af-

fecting how the function behaves. A user at a management station is able to perform such actions as start and stop transport connections, change lists of locally accessible MAC addresses and NetBIOS names, disconnect hung circuits, and remotely change common configuration parameters.

Finally, the DLSw MIB defines events that cause the router to asynchronously notify the management station of a potential or definite problem. Yet to be finalized in the standard MIB, these events may include the rejection of a new transport connection, any DLSw protocol violation, and the failure of a circuit.

The future of DLSw

Given the currently strong vendor and user interest in DLSw for integrating SNA and NetBIOS traffic into multiprotocol networks, DLSw is likely to be in use for several years. Even in networks where APPN provides advantages over DLSw in handling SNA node-type 2.0 and 2.1 traffic,[34] DLSw may continue to have a useful role in routing NetBIOS and subarea SNA non-2.0/2.1 traffic. Based on the DLSw vendor strategies that have been announced, it is safe to predict that DLSw will grow in a number of interesting directions from the level of the AIW Version 1 standard.

We can expect DLSw to evolve concurrently in at least three ways: first, DLSw vendors will apply the DLSw standard in new configurations that do not require any extension to the standard; second, AIW members will define new functions they wish to add to the DLSw standard; and third, individual DLSw vendors will define vendor-specific DLSw protocol extensions. In fact, Version 1 of the standard has facilitated the development of such extensions by allowing the exchange of vendor-defined capabilities between DLSw partner routers. Some vendors who develop proprietary extensions may later choose to make specifications available to the rest of the vendor community, hoping to make those extensions part of the standard.

Vendor configurations. Several new configurations are possible, where a vendor can extend DLSw beyond its typical use for connecting SDLC and LAN-based LLC end stations, to handle SNA traffic from other DLC protocols and environments. Clear candidates are: LLC running over frame relay links rather than over a LAN;[35] qualified logical link control (QLLC), for running SNA over X.25 networks; and channel data link control, for System/370* channel links.

In addition to these DLC protocols, some vendors are introducing what might be termed a "null DLC" configuration, where DLSw is actually running in an end station. In this configuration, the end station directly generates DLSw SSP messages in IP packets, instead of generating native DLC frames and relying on an external router to handle the DLC-to-DLSw mapping. Software in the end station maps existing user application communications programming interfaces to the appropriate DLSw SSP message flows. This approach begins to move DLSw from the realm of a router-based technology to that of an end-station-based technology as discussed in the beginning of this paper. Because DLSw has not yet been integrated into any large host environment, however, DLSw end stations in most networks would still require a remote DLSw-capable router to be the partner DLSw device.

The DLSw standard attempts to specify circuit state transitions and SSP messages in relation to *generic* DLC events and actions. Although LLC and SDLC are explicitly discussed, the philosophy of the standard is that implementation for DLC types other than these (e.g., all those just discussed) is simply a product choice, and does not constitute an extension of the DLSw standard.

AIW extensions. The AIW has discussed two problem areas it may address in a second version of the DLSw standard: loop prevention and transmission priority. Currently, there are a number of network configurations involving both DLSw routers and transparent bridges where frames may loop forever or be duplicated and confuse end stations. Users must design and maintain their networks in a way that prevents loops from occurring. AIW members are working to devise an algorithm that automatically detects and corrects loops as soon as they appear in a network.

Regarding transmission priority, we have previously discussed the simple circuit priority mechanism adopted for the Version 1 standard. Because this mechanism fails to distinguish between SNA sessions and cannot prioritize traffic inside the IP network, AIW members have also discussed more sophisticated approaches to DLSw traffic prioritization.

As of this writing, there is no general agreement within the AIW about the user requirements the Version 2 standard must address.

Vendor extensions. While DLSw vendor announcements are ongoing, one can identify a number of directions in which vendors might most easily and fruitfully extend the protocol. To begin with, vendors may extend DLSw to handle protocols other than SNA and NetBIOS. Examples discussed so far in the industry include IBM's LAN Network Manager (LNM) protocol for linking to and managing LAN bridges,[36] and Digital Equipment Corporation's Local Area Transport (LAT) protocol.[37] In theory, DLSw can effectively carry any protocol that involves a series of messages exchanged between two end stations. The series should be long enough to merit the overhead of DLSw circuit setup. As shown by the way DLSw carries SNA XIDs prior to *contact/contacted* exchange, the carried protocols may be purely datagram-oriented and have no explicit connection-setup frame flows. DLSw's advantages over IP tunneling for connectionless protocols include using a standard method for locating end-station resources across the WAN, and being able to reduce WAN bandwidth requirements by filtering retries of frames that are reliably delivered across the WAN.

The DLSw switch-to-switch protocol and MIB were both explicitly designed to accommodate a future transport protocol other than TCP. One candidate to replace TCP as the reliable protocol carrying SSP messages between partner nodes is LLC, specifically on point-to-point links or through fast-packet-switched networks such as frame relay or asynchronous transfer mode (ATM). Note for frame relay that this proposal is to run the DLSw SSP on top of LLC over frame relay links between DLSw routers, in the same way that the SSP runs on TCP/IP over frame relay links today. This should not be confused with the LLC-over-frame relay vendor configuration discussed earlier, where LLC is running over a frame relay link between a DLSw router and an end station.

As currently defined, DLSw's use of static transport connections to all possible search destinations limits its ability to scale to large networks. Vendors may wish to tackle this problem by decoupling network searches from transport connections, and only bringing up a transport connection between routers when it is needed to carry active user traffic. We have noted, for example, how some DLSw implementations use multicast IP to discover the identity of partner routers. This same multicast mechanism could be extended to carry searches for destination resources (e.g., MAC addresses and NetBIOS names), and to carry NetBIOS broadcast traffic. Another approach might be to introduce a two-level hierarchy of DLSw nodes, akin to the network-node/end-node APPN concept. Only network-node DLSws would maintain persistent transport connections with each other in a full mesh, and would bear the brunt of handling network search and NetBIOS broadcast traffic. End-node DLSws would need to connect only to their serving network-node DLSw, and would bring up transport connections to other end-node DLSws only as required to carry user circuits.

With the growth of ATM and other switched network technologies, DLSw is likely to find itself in routers on the boundary of a switched wide area network. Initially, the switched WAN will probably provide to boundary routers the appearance of point-to-point links, either through bare circuit emulation or by providing a frame relay service. DLSw could run its transport connections to other DLSw routers across these point-to-point links, or it could run a DLC protocol such as SDLC or LLC (over frame relay) on them, directly to an end station.

Later, router functions such as DLSw may begin to more intelligently use the capabilities of the switched WAN. For example, the scalability limitation just described could be overcome with the use of hardware-based multicast groups. The switched network infrastructure could provide DLSw with improved transport protocols, better flow control methods, and tools to provide true circuit quality-of-service selection.

When the boundaries of the switched WAN reach end-user systems, it is unclear what the role of routers will be. Assuming routing will continue to exist for such purposes as local address resolution and broadcast isolation, OSI layer-2 data link switching across new WAN transport services may continue to be a part of the router function.

Concluding remarks

Data link switching has emerged as an important router-based solution for transporting SNA and NetBIOS across wide area networks. It is being widely implemented, both because it solves key

problems of predecessor solutions and because it is the first nonproprietary method for doing so.

Networking vendors continue to announce a variety of directions in which they plan to apply and extend DLSw, suggesting that it is a basic technology with broad applicability. With its potential to support new DLC types, protocols carried, and transport protocols, data link switching promises to be a part of user networks for several years to come.

Acknowledgments

The author wishes to express appreciation to Gary Schultz and Roy Dixon for their helpful review of early drafts of this paper.

Roy Dixon and David Kushi from IBM deserve mention here for authoring the initial DLSw specification, RFC 1434. The AIW Version 1 DLSw standard itself was the collaborative effort of individuals from a number of companies. Key contributors included: Steve Klein, Gene Cox, and this author from IBM; Paul Brittain from Data Connection, Ltd.; Shannon Nix from Metaplex; Wayne Clark from Cisco Systems; and Alan Bartky (editor) from Sync Research.

Appendix: Internet document access

At the time of publication, several documents cited in this paper are available through the Internet. There are two general mechanisms for obtaining the document files, which exist in ASCII text and sometimes PostScript**-formatted versions. Users with File Transfer Protocol (FTP)-access to the Internet may link to hosts where the files reside and transfer them directly to a local host. Users with only electronic-mail-access to the Internet may request the files by sending a keyword-formatted message to a mail server host.

Requests for Comments. For detailed instructions for obtaining most Internet Engineering Task Force (IETF) Requests for Comments (RFC), including lists of FTP and mail server hosts in various countries, send an electronic-mail message to "rfc-info@isi.edu" with the message body "help: ways_to_get_rfcs".

DLSw standard. To obtain a copy of the Advanced Peer-to-Peer Networking (APPN) Implementers' Workshop (AIW) Version 1 DLSw Standard, use anonymous FTP or a World Wide Web browser to access the file with the following uniform resource locator (URL): ftp://networking.raleigh.ibm.com/pub/standards/aiw/dlsw/dlsw_v1.txt

The home page for the AIW is at URL: http://www.raleigh.ibm.com/app/aiwhome.htm

You may also request RFC 1795 using the instructions from the previous section.

*Trademark or registered trademark of International Business Machines Corporation.

**Trademark or registered trademark of Adobe Systems Incorporated.

Cited references and notes

1. *Systems Network Architecture: Technical Overview*, GC30-3073, IBM Corporation; available through IBM branch offices.
2. R. J. Cypser, *Communications for Cooperating Systems: OSI, SNA, and TCP/IP*, Addison-Wesley Publishing Co., Reading, MA (1991).
3. *LAN Technical Reference: IEEE 802.2 and NETBIOS Application Programming Interfaces*, SC30-3587, IBM Corporation; available through IBM branch offices.
4. A. S. Tanenbaum, *Computer Networks*, Second Edition, Prentice-Hall, Inc., Englewood Cliffs, NJ (1988).
5. D. Pozefsky, R. Turner, A. K. Edwards, S. Sarkar, J. Mathew, G. Bollella, K. Tracey, D. Poirier, J. Fetvedt, W. S. Hobgood, W. A. Doeringer, and D. Dykeman, "Multiprotocol Transport Networking: Eliminating Application Dependencies on Communications Protocols," *IBM Systems Journal* **34,** No. 3, 472–500 (1995, this issue).
6. RFC 1001, *Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Concepts and Methods*, and *RFC 1002, Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Detailed Specifications*, Internet Engineering Task Force (March 1987). (See Appendix for obtaining these documents.)
7. For a detailed discussion of these approaches, see A. Guruge, "Bridging the Gap Between SNA and Multiprotocol Internets" and "Merging SNA Link Traffic into LAN/WAN Internets," *Business Communications Review* supplement, *BCR's Guide to SNA Internetworking* (August 1993).
8. *Introduction to the Open Blueprint: A Guide to Distributed Computing*, G326-0395, IBM Corporation; available through IBM branch offices.
9. All WAN links under discussion may be either simple circuit-switched point-to-point connections, or logical connections through packet-switched or cell-switched networks that provide OSI layer-2 frame relay services.
10. *International Standard ISO 8802-2, IEEE Std 802.2*, The Institute of Electrical and Electronics Engineers, Inc., New York (1989).
11. For a description of different bridge types, see K. J. Christensen, L. C. Haas, F. E. Noel, and N. C. Strole, "Local Area Networks—Evolving from Shared to Switched Access," *IBM Systems Journal* **34,** No. 3, 347–374 (1995, this issue).

12. For a more detailed discussion of performance issues related to these transport methods, see C. S. Lingafelt, Sr., "Data Link Switching, SDLC Conversion, SDLC Passthrough, Etc. SNA Connection Methods Within Today's Wide Area Networks. An Overview," *Proceedings of the 19th International Conference for the Management and Performance Evaluation of Enterprise Computing Systems,* San Diego, CA (December 1993), pp. 270–280.
13. RFC 1490, *Multiprotocol Interconnect over Frame Relay*, Internet Engineering Task Force (July 1993). (See Appendix for how to obtain this document.)
14. *Multiprotocol Encapsulation Implementation Agreement*, FRF.3, Frame Relay Forum (not dated), 303 Vintage Park Dr., Foster City, CA 94404.
15. Technically, a link converter also presents a spoof of one attached end station to another. One could make a distinction between this "local spoofing" and the "remote spoofing" under discussion, but the usage we have chosen is more common. None of these terms is formally defined in a standard.
16. The DLSw router itself counts as one or sometimes two hops.
17. RFC 1434, *Data Link Switching: Switch-to-Switch Protocol*, Internet Engineering Task Force (March 1993). (See Appendix for how to obtain this document.)
18. IBM sponsors the AIW as a forum for vendors to pursue common goals in advancing APPN architecture.
19. DLSW_V1, *Data Link Switching: Switch-to-Switch Protocol, Version 1*, APPN Implementers' Workshop (December 1994) and RFC 1795, *Data Link Switching: Switch-to-Switch Protocol,* Internet Engineering Task Force (April 1995).
20. *Synchronous Data Link Control*: Concepts, GA27-3093, IBM Corporation; available through IBM branch offices.
21. In this approach, the user configures a given router as a member of one or more numbered *groups*, and defines a role—*client*, *server*, or *peer*—that the router is to play within each group. When the router starts up, it sends queries to multicast IP addresses corresponding to the complementary role within the groups of which it is a member. From the replies it receives, the router learns the IP addresses of other group members with this complementary role, and can initiate TCP connections to those IP addresses. All *peers* in a group establish a transport connection with every other peer. *Clients* in a group have a transport connection only with the *servers* in that group, and not with other *clients*.
22. In many DLSw implementations, TCP connections carry periodic "keep alive" messages during periods of inactivity, so that end stations with connected circuits can be promptly notified of any transport connection failure. With these characteristics, DLSw is usually not well-suited for use over dial-up WAN connections. To prevent a connection from being frequently dialed up, "keep alive" needs to be disabled, and a filtering regime imposed to limit the frequency of network searches.
23. The same method is used in some DLSw router implementations to route around failed WAN adapters. For this work, the DLSw IP address must belong to the product as a whole and not one particular WAN interface.
24. In the IBM 6611 Network Processor, dual simplex TCP connections simplified customer definition of partners, improved performance, and simplified transport connection bringup logic.
25. This version is normally referred to as "RFC 1434+," to indicate that the product supports several SSP messages that are not documented in RFC 1434. RFC 1434+ is not defined in any formal document, but a corresponding revised draft of the key parts of RFC 1434 is electronically available from the AIW.
26. This description is slightly oversimplified, to avoid excessive detail. Many SNA end stations first send a TEST command to only the local ring, followed by the explorer frame, which bridges copy to all rings in the bridged network. The explorer frame may be sent through all routes or along a spanning tree, and its response flows on a specific route or through all routes, respectively.
27. Unfortunately, LAN-attached SNA end stations do not support an Address Resolution Protocol (ARP) like the one defined for IP and other OSI layer-3 routed protocols. ARP in IP allows a node to dynamically resolve an IP address to the MAC address of the node that supports it. Lacking a similar function to discover the MAC address for, say, a given SNA CP name, SNA nodes must use MAC addresses to identify one another.
28. Some DLSw implementations allow this function to be disabled so that previously disjoint LAN regions with no structured naming convention may be joined by DLSw routers. Duplicate names may actually exist without causing problems, as long as these names are not the target of a NAME_QUERY.
29. D. E. Comer, *Internetworking with TCP/IP: Volume 1: Principles, Protocols, and Architecture*, Prentice-Hall, Inc., Englewood Cliffs, NJ (1991).
30. Since Router 1 has acknowledged the data to end station A and reliably delivered it to Router 2, Router 2 is required either to deliver it to end station B or to disconnect the circuit.
31. *Systems Network Architecture Advanced Peer-to-Peer Networking: Architecture Reference*, SC30-3422, IBM Corporation; available through IBM branch offices.
32. M. T. Rose, *The Simple Book: An Introduction to Management of TCP/IP-based Internets*, Prentice-Hall, Inc., Englewood Cliffs, NJ (1991).
33. The MIB is defined in a separate document from the AIW protocol standard. The most current draft of this work in progress is available from the AIW by anonymous FTP at URL: ftp://networking.raleigh.ibm.com/pub/standards/aiw/dlsw/mib/draft-letf-dlswmib-mib-05.txt
34. Because APPN routing occurs at the SNA session level instead of the data link layer, it provides a number of advantages over the DLSw SNA node-type 2.0/2.1 support in many environments. With the addition of the DLUR (dependent logical unit requester) function, APPN provides session-level class of service and transmission priority for both node-type 2.0 and 2.1 end stations. APPN computes routes on an LU-LU (logical unit to logical unit) session basis, which allows for more efficient and dynamic path selection than DLSw circuit establishment can provide. In other areas of comparison, APPN HPR (high-performance routing) may be superior to TCP/IP in areas such as flow control, and APPN currently scales to large networks more naturally than DLSw. For more information on these APPN functions, see *APPN Architecture and Product Implementations Tutorial*, GG24-3669, IBM Corporation; available through IBM branch offices.
35. As previously noted, the IETF and the Frame Relay Forum have defined frame formats for carrying SNA traffic over frame relay links using LLC. (See References 13 and 14.) These formats, generally referred to as "RFC 1490,"

have been widely implemented in IBM networking products and in products by link converter and router vendors. With only minor differences, SNA end stations supporting RFC 1490 use the same LLC frame sequences described earlier for LAN-based end stations. As a result, DLSw logic for LLC events and actions applies to either environment.
36. LNM uses type-2 LLC for reliable data transfer, but has unique type-1 frame flows for linking to a bridge. The LNM protocol is described in: *Token-Ring Network: Architecture Reference*, SC30-3374, IBM Corporation; available from IBM branch offices.
37. C. Malamud, *Analyzing DECnet/OSI Phase V,* Van Nostrand Reinhold, New York, NY (1991).

Accepted for publication March 14, 1995.

Peter W. Gayek *IBM Networking Hardware Division, P. O. Box 12195, Research Triangle Park, North Carolina 27709.* Mr. Gayek joined IBM in 1982 in Gaithersburg, Maryland, where his initial assignment was to develop voice call processing software for the SBS Satellite Communications Controller. He subsequently worked in the design and development of a number of networking functions for the 3174 Establishment Controller, including: token-ring LAN attachment and gateway, LAN Network management, and APPN. From 1990 to 1992, Mr. Gayek managed a department that developed software for bridge and bridge network management products, as well as for some of IBM's initial CallPath products. After transferring to Research Triangle Park in 1993, he began his current assignment in the design and development of Data Link Switching within the Advanced Routing Products organization. Since 1994, Mr. Gayek has served as the technical coordinator of IBM's participation in the DLSw Related Interest Group of the APPN Implementers' Workshop. Mr. Gayek received a B.S. degree in computer science from Cornell University in 1982.

Reprint Order No. G321-5575.