Internetworking Across Public ATM Networks

Ahmed Tantawy and Martina Zitterbart

IBM Research Division
Thomas J. Watson Research Center
P.O.Box 704, Yorktown Heights, NY 10598

Abstract

The emergence of public ATM networks, such as B-ISDN and SMDS, is considered to have a high impact on future internetworking environments. This paper presents a new bridging architecture designed for that purpose and discusses its feasibility in TCP/IP networks. The new architecture, called Open Bridging, uses special capabilities of public ATM networks, such as the hierarchical addressing mechanism and the routing support inside the network, to provide for efficient interconnection of remote LANs across public ATM networks. The operation of Open Bridges does not interfere with current TCP/IP protocols. However, the efficiency of Open Bridging can be obtained in the interconnection of TCP/IP networks if a slight extension to the Address Resolution Protocol (ARP) is implemented to push the routing of data packets down to the MAC level. Moreover, the architecture of Open Brouters is also proposed to enable full interoperability among domains using either Open Bridging or IP routing.

1. Introduction

For over a decade, various studies and investigations have focused on the interconnection of LANs and WANs. However, the future environment in which internetworking will be needed is likely to be different. We believe that the proliferation of LANs and the emergence of high bandwidth public networks will create a new environment, in which it will become increasingly simple and attractive to communicate among different organizations across a common public networking infrastructure.

We examine here the issues related to the interconnection of LANs through a backbone made of interconnected public MANs. In particular, we focus on MANs based on the emerging standards, namely the ATM cell-based B-ISDN and SMDS. It should be noted, however, that the new environment is far more demanding in terms of performance and flexibility than current networks (such as Internet and X.25). Given the existence of a number of different standard protocol stacks (TCP/IP, SNA, OSI, etc.), we believe that the LAN/MAN interconnection platform should be as independent as possible of the higher layer protocols running in the various interconnected environments. This means that the level of interconnection has to be kept as low as possible. Moreover, given the relatively low protocol processing power available today, compared with the increasingly high bandwidth of physical transmission media, one can conclude that it is desirable to use bridges rather than routers.

Providers of public data communication services as well as their customers independently manage and operate their corresponding networks. They actually define *independent domains* with clear boundaries. It is this very notion that we view as the basis of our scheme to interconnect private domains across a public domain, such as SMDS [1] or B-ISDN [2], at the MAC level. We call this scheme *Open Bridging* to distinguish it from current standards, where all bridged networks have to be part of one administrative domain. The Open Bridging scheme exploits some of the features of public switching networks as opposed to the current internetworks based on a mesh of point to point links between networks. One of our goals is to overcome the necessity of using network layer routing.

In the following section of this paper, we discuss TCP/IP internets and their evolution to utilize public switching networks. In section 3 we present the basic concepts of Open Bridging. Interoperability with currently available nodes and protocols is an essential requirement for the success of new schemes. As a case study, we show in section 4 that Open Bridging does not interfere with the operation of currently available TCP/IP protocols. We also show how the Address Resolution Protocol (ARP) can be slightly modified to make use of Open Bridging. The architecture of Open Brouters, proposed in section 5, enable full interoperability among domains using either Open Bridging or IP routing.

2. Internet and Emerging Public Switching Networks

2.1 Network Interconnection in Current Internets

TCP/IP is the common name for a layered communication architecture used in building an *Internet*.

0-8186-2755-7/92 $03.00

Within this architecture, IP routers are the interconnection units to the public environment. Routers are usually interconnected in the public network via point-to-point connections, such as T1 or T3 links. This means that every IP router in the public network has a direct connection to all of its neighboring IP routers. Thus, an IP router has a separate physical attachment for every connection in the public environment and it has to select the specific part on which each packets has to be forwarded.

The following discussion focuses on routing and address mapping mechanism in TCP/IP environments. Several books present a thorough explanation of the Internet and the TCP/IP protocol suite (e.g., [3], [4] and [5]).

2.1.1 Internet sockets and addresses

An application process in the Internet is uniquely identified by a *socket*, which is a pair of integers (Host, Port). *Host* is the Internet address (often called IP address) of a host and *Port* is a TCP port on that host. An IP address is a unique 32-bit value in the entire Internet. It is hierarchically structured in a way that allows for an easy identification of the network where it is located. Ports are identified by unique numbers within a host.

2.1.2 Mapping a name to a socket

To establish a connection, the TCP service user passes, among other parameters, the destination socket (i.e., the destination IP address and the destination TCP port) to the Internet layer. Therefore, the application process must know the destination socket. It can obtain it by querying a *name server*, which holds the address associated with the logical name of the remote process with which communication is sought. TCP then constructs and passes TCP segments to the IP layer along with the source and destination Internet addresses (extracted from the destination socket, given by the application).

2.1.1 IP Routing

IP is the basic protocol of the Internet layer, in addition to other protocols, such as the Internet Control Message Protocol (ICMP) and the Address Resolution Protocol (ARP). One of the main purposes of IP is the routing task. It has to determine where the destination station is located. If it is located on the same physical network, the IP datagram will be handled by the Network Access layer. Otherwise, IP must find the address of the gateway that is the next hop on the route to the desired destination. The criteria for choosing the next gateway are numerous. They include cost optimization, delay minimization, and security (by forcing routing through a prespecified path). In an environment where Ethernets or IEEE 802 LANs are used, IP has to use MAC addresses when passing packets to the network access layer [6]. Therefore, it has to map Internet addresses to MAC addresses, i.e., it has to perform the following procedure: IP extracts the network portion of the hierarchical IP address to find out whether the destination is located on the same physical network by comparing the network portion with its own network address. If the destination resides on the same network, the ARP entity is called by IP to obtain the physical address of the destination, as described in 2.1.4. Otherwise, if any entry for that specific destination exists in the routing table, the segment is routed as specified. If no such entry exists, the datagram is routed to a so-called default gateway. In all those cases, the physical address of the next hop (being either a gateway or the destination host) is specified in the packet header. Thus, a gateway is viewed as a destination by the network access layer and all lower layer internetworking units. Note that the destination Internet address specified in the IP datagram will always be the original address of the destination host and not that of the gateway. *2.1.4 Operation of ARP*

ARP is responsible for mapping IP addresses to physical addresses. Each station maintains a table of known Internet to physical address mappings. If no mapping can be found for a requested Internet address, ARP sends a broadcast frame on the physical network. This frame includes the source and destination Internet addresses as well as the source physical address. Note that if the physical network itself is a bridged LAN, all the bridges have to broadcast the ARP frame. Each station on the physical network receives this frame and compares the destination Internet address with its own Internet address. If it identifies itself as the destination, it replies with a frame addressed to the source station. This ARP response frame includes the requested physical address of the destination station.

2.2 Impact of Public Switching Networks on IP Routers

During the next few years, rapid changes in public data networks are expected due to the development of ATM switching networks, such as B-ISDN and SMDS. The use of *hierarchical 64-bit E.164 network addresses* [7] and the possibility of *maintaining several connections across a single physical point of attachment* are new features in that environment that will affect the interconnection of remote networks across those public facilities. Routing in these networks is based on the E.164 addresses and the *switches are responsible for the implementation of routing.* This means that the routing function is being moved from the network layer to the MAC layer. Moreover, point-to-point links will gradually disappear. Note

that, inside the switched network, only switches are needed and there are no IP routers. These switches route the frames through the network according to their hierarchically structured E.164 addresses. This makes connections across the switching network appear as virtual point-to-point links between every pair of nodes residing at the borders of the public network, which becomes like a virtual fully interconnected mesh.

An IP router connecting a LAN to a switched WAN has to perform some different tasks, compared to a standard IP router. An IP router connected to ATM has to provide a mapping of IP addresses onto public E.164 addresses. This address resolution has to be done at the border of the ATM network. Thus, the routing function of the IP layer is not used inside the public environment. The only main function performed is the address resolution. Furthermore, an IP router at the border does not have any influence on the path that the frame will take through the ATM network.

In general, the operation of TCP/IP across public switching networks raises some questions. Until now, such operation has only been considered for ATM networks configured as logical IP subnetworks and in such applications, the procedure is simply based on the encapsulation of IP frames (including those frames used for ARP) [8]. It should be noted here that ARP frames are broadcast and, therefore, the pure encapsulation is not practical in public environments. Another issue is routing to unknown destinations. Current IP routers use default gateways for that purpose. In the case of ATM networks, IP routers are located only at the border of the network and thus a packet forwarded to a default gateway passes the public network twice: once to the default gateway and from there to the proper gateway. It is also not clear how path costs, including subpaths across ATM networks, will be determined. Additionally, the necessity of IP addresses becomes questionable when hierarchical and universal ISDN addresses are used in the public network. The necessary address mapping steps can be reduced if ISDN addresses are substituted for Internet addresses.

The fact that bridging approaches can overcome some of the problems stated above, especially multiple address mapping, favors them as future interconnection units. In the following, we present the Open Bridging approach for LAN interconnection across ATM networks and the integration of Open Bridges in TCP/IP networks interconnected across ATM switched networks.

3. Fundamental Principles of Open Bridges

Providers of public data communication services develop networks capable of carrying information from one customer to another in a virtually transparent manner. These providers and their customers independently manage and operate their corresponding networks. They actually define *independent domains* with clear boundaries. It is this very notion that we view as the basis of our scheme to interconnect private domains across a public domain. With the increasing proliferation of LANs in customer premises, a typical network topology will consist of clusters of bridged LANs interconnected by public switched WANs or MANs.

We define a *domain* as being a *bridged network*, independently managed and having its own independent routing and addressing mechanisms. Generally, there are no geographical limitations to a domain. It could include more than one set of bridged LANs interconnected by point-to-point or switched links, using remote (or split) bridging techniques (cf. (1) in Figure 1). We define *Open Bridging* as being the concept of bridging independent domains in an efficient and transparent manner. Given that the nature and intensity of communication among domains belonging to the same organization are different from those existing in a more *open* environment, we find it preferable to distinguish between these two classes of service. Consequently, we define *a Virtual Private Network (VPN)* as a set of independent domains belonging to the same organization, but each of these domains still has the Open Bridging capability on an individual basis.

One of the objectives of Open Bridging is to allow all the stations within the same private domain to continue to communicate as specified in the current bridging standards (cf. [9] and [10]), even when these stations gain access to remote stations. Therefore, a sending station does not have to distinguish between stations belonging to its local domain and others belonging to foreign domains. It is also reasonable to believe that the operation and integrity of individually owned domains should remain independently and privately managed. This, however, goes against the current principles of bridging techniques, where the entire bridged environment is considered to be one entity, as in [9], [10], [11], and [12].

In short, it is our view that the separation between independent domains should be kept. Interdomain communications should be made through special bridges situated between domains. Each of these new

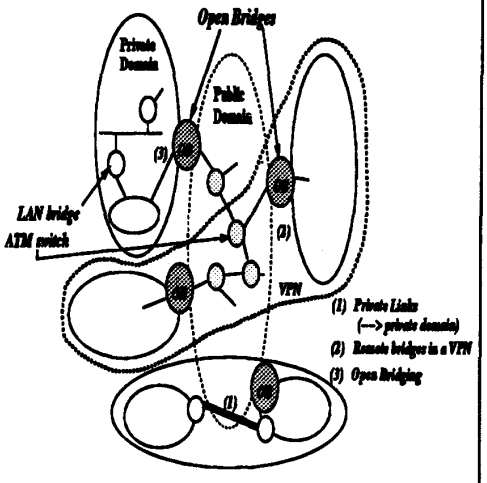

Figure 1. Open bridging and independent domains

bridges should be made capable of participating in the routing algorithms in each of the domains that it interconnects. This means that a clear distinction should be made between *local bridges* (LAN bridge or ATM switch) and *Open Bridges* (OBs), as illustrated in Figure 1.

An OB is the point of interconnection between a private and a public domain. It acts as a virtual remote bridge between distant private domains. We will call *source OB* the OB of the private domain in which the frame is generated and *destination OB* the OB of the domain in which the destination end system resides. To ensure *transparency of inter domain communication* to end systems, the OB has to interface between the routing algorithms in each of the bridged independent domains. The local routing protocols and algorithms remain unchanged and the public domain is not involved in local routing decisions. In general, the routing path between two remote stations is subdivided into three subpaths: private source subpath (source end station to source OB), public interdomain subpath (source OB to destination OB), and private destination subpath (destination OB to destination end station). The OB has also to provide routing information to enable correct and efficient forwarding of frames through the public domain. Moreover, address mapping is necessary between local and public domains. Clearly, a new protocol is needed to coordinate the operation of these OBs. We call it the *Open Bridging Protocol* (OBP).

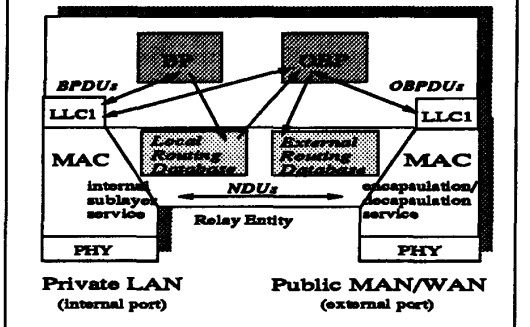

Figure 2. Architecture of the Open Bridge

The architecture of the OB is shown in Figure 2. For each *internal port* (connected to a LAN), a MAC entity has to be implemented. The *external port* (connected to the public MAN) has to provide a MAC-like service interface. For example, in case of an SMDS network, the SMDS Interface Protocol (SIP) is to be implemented.

According to the architecture of a MAC bridge [9], a Relay entity between the different MAC entities must be implemented. The OB also comprises an implementation of the LLC type 1 protocol [13] at each port. This entity serves the *Bridge Protocol (BP)* entity associated with the domain connected to the corresponding port. The Bridge Management entity and the OBP entity are also served by the LLC entity of the internal ports in addition to that of the external port.

Frame processing within the OB is based on the *normalized encapsulation* method. We prefer this method because it keeps all MAC header information until the frame reaches its destination domain and allows the use of independent address spaces in the public and private domains. It is implemented by converting the received frame into a *Normalized Data Unit (NDU)* which is known by all OBs. The NDU is then passed to the public MAC entity as simple data and thus encapsulated into the MPDU sent across the ATM network. The remote station decapsulates the frame and converts the NDU to the destination MAC format. Thus each bridge port needs only to provide conversion functions between its MAC format and the NDU format.

The NDU is simply a superframe structure containing a field for each parameter needed for the internal sublayer service primitives [9]. The existence or absence of each field is indicated by a bitmap flag.

The OB supports different routing protocols, as required by the connected private and public domains and performs the necessary mapping between them. It also keeps two types of databases. The *local routing database* contains information pertaining to routing in the local domain (e.g. spanning tree algorithm [9]), whereas the *external routing database* provides the 60-bit address of the OB leading to the remote domain containing the destination station.

4. OBs as Interconnection Units across ATM Networks

Open Bridges [14] make use of the hierarchical addressing and switching features of public switching networks, such as B-ISDN and SMDS. Thus, they seem well-suited to replace standard IP routers at the interconnection point to those networks. In the current TCP/IP environment, where neither hierarchical addressing is provided inside the public network nor MAC level switches can be used, OBs cannot be incorporated. However, the changing networking environment makes them attractive in the future. Besides simplicity and functionality, motivation may stem from performance: current bridges can relay about 4 times more packets per second than current routers and our preliminary estimates show that OBs may be able to handle twice the volume of packets handled by routers, using the same processing power.

We consider end stations in the private domain running TCP/IP protocols. Other protocol environments (such as OSI) can be similarly treated but are not discussed in this paper. We further assume a one-to-one mapping of a physical network (as defined in the IP terminology) onto a domain (as defined in the Open Bridging terminology). This means that a physical network neither extends beyond the boundaries of a private domain nor builds a subset of a private domain. Consequently, different domains have different IP network identifiers. Therefore, the ARP protocol cannot be used for address resolution among remote networks. In such cases the IP protocol entity of the source node addresses the frame to the next hop, normally an IP router (cf. section 2). This means that the generated MAC frame carries as destination MAC address the address of the IP router that is the next hop on the path to the final destination node. Thus, at the MAC-level, any information about the real MAC destination address is lost.

We propose an architecture in which an enhanced address resolution protocol is implemented in the OB to resolve the problem discussed above. Note that the OB is only required to implement this protocol and not the complete set of IP layer protocols. Thus, it still remains a bridge, because network layer routing functionality does not need to be implemented and data frames will be routed at the MAC level.

4.1 The eXtended Address Resolution Protocol

The following steps are used to establish a communication between two nodes residing in two remote physical networks and wishing to use the OB facilities. The main point is to obtain the destination MAC address so that all IP datagrams can flow from the source node to the destination node without crossing the MAC service level, i.e., only bridges will be involved in routing the datagrams between these nodes. This requires a slight extension to the ARP that is used now in IP networks. We call this extension: *XARP (eXtended Address Resolution Protocol)* [15].

The IP protocol at the source node checks the destination IP address given by TCP. If both the source and destination are located in the same domain, ARP is used to find the physical address of the destination (which will be either cached or discovered by the ARP procedure). Once address resolution is done, bridging may be used to deliver the MAC frame, which is actually an encapsulated IP datagram. On the other hand, if the source and destination belong to different networks, routing will be done through IP routers and the ARP is not used to find the physical address of the final destination node. This is where we propose to use the XARP.

To enable the OB to act as a "virtual" IP router (and thus to receive XARP frames), an entry representing the OB in the IP routing table has to be added. In the simplest case, the OB is the only connection to the public switching network. Thus, for all the frames that are not local, the OB can be designated as the "default gateway". The network administrator usually adds the entry for the default gateway to the routing table. In the case of OBs, a "fictitious" IP address is used, because the OB itself does not need to have any IP address. This special address associated with the OB will be called the *OB Identifier* (OBI). The OBI has to be unique and cannot be used as an IP address for any real node in that network. The network administrator is also responsible for that. Thus, if the IP destination address in a frame does not belong to its source network, the routing table in IP will give the OBI as IP address of the next hop. Instead of using the ARP protocol, IP uses XARP if a location in a remote IP network has to be resolved. Note that this only affects the address resolution procedure. Neither the IP protocol nor the IP routing procedure need to be modified.

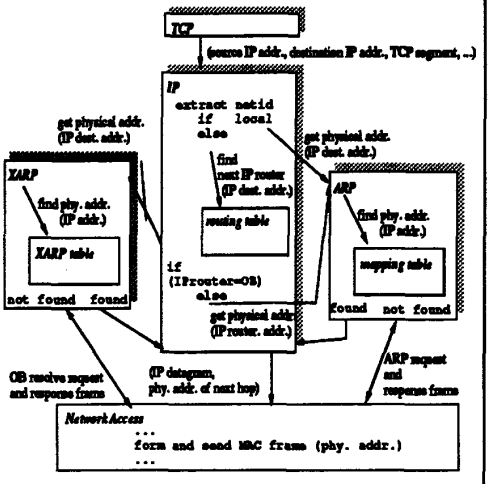

Figure 3. XARP and IP Layer Protocols

The address resolution procedure within an OB environment using XARP can be summarized in the following steps:

1. IP receives a TCP segment, extracts the network identifier from the destination IP address and proceeds as follows:
   - IP destination address is local:
     IP calls ARP and proceeds as usual.
   - IP destination address is remote:
     IP checks its routing table to find an associated IP router for that destination network or for a specific destination node. In case OBs are used, it might also find the OBI. After searching in the routing table, IP has to distinguish between the following cases:
     - Any IP router address (other than OBI):
       In this case the next hop is a regular IP router and IP can proceed as usual.
     - OBI found:
       The frame is passed across the public environment using OBs. The MAC address of the destination station (either end station or IP router) in the remote LAN is to be found. XARP generates an OB-XARP request frame to the local OB as a query. The local OB is then expected to respond with an OB-XARP response frame carrying the required MAC address. This is done after XARP entity in the OB performs the tasks described in the next step given below. Then, the destination MAC address is added to the XARP mapping table and can thus be used for subsequent frames. Note that OB resides at a known MAC address that can be inserted in the address mapping table.
2. The XARP entity of the local OB receives an OB-XARP request frame if a remote MAC address has to be resolved. If the local OB has no entry for the queried address, it sends an OB-resolve request frame to its peer in the destination OB, i.e., the OB corresponding to the network in which the destination node resides. This can be easily done because such mapping information (NetId to OB E.164 address) is needed in any case for the basic operation of the IP protocol across public networks using the ISDN addressing scheme.
3. The XARP entity at the destination OB then looks in its table, where MAC addresses corresponding to its local IP addresses are stored. If no entry can be found, the destination OB has to broadcast an XARP request frame in its local network to resolve the destination MAC address. The XARP then sends an OB-XARP response frame to the requesting XARP entity in the local OB to inform it about this address.
4. The XARP entity then relays the proper mapping (i.e., the destination MAC address) with an OB response frame to the XARP entity in the source node. This address correspondence is then cached in the source host and XARP uses it subsequently to give IP entity the MAC address of a destination node, even if that node is actually in a different network. This information is used for all datagrams that follow the first one between any two specific hosts.

The proposed scheme leads to shorter datagram latency in the network during the data transfer and the connection termination phases of TCP. On the other hand, TCP connection establishment will be slower by one round trip propagation delay, in the worst case. This penalty is paid only during the first connection established between processes in two given hosts. Subsequent connections -even between other processes on the same pair of hosts- will be faster to establish. This is particularly interesting in an environment where servers are used and to which connections are usually made. The likelihood of a "hit" in the cache is

higher due to the higher probability of connection requests with the servers.

It should be noted that XARP behaves exactly like actual ARP when an address resolution request is received by a system running ARP. This is to ensure full interoperability with current systems that do not wish to be converted to the use of Open Bridging.

To implement the XARP, each OB has to contain a XARP entity. Additionally a SNAP entity is recommended by the standard for TCP/IP protocols running on top of IEEE 802 LANs [6]. The XARP entity provides an address mapping table that associates each network identifier with the E.164 ISDN address of its OB.

XARP also uses exactly the same frame format as the existing standard for ARP frames proposes. The XARP frame is then encapsulated as data in a SNAP frame. The resulting MAC frame thus carries its MAC header, an LLC header and a SNAP header. For the use of ARP, a special SNAP value has been defined. We intend to use the same value for XARP. Frames received for ARP or XARP can be distinguished based on their operation code field. The codes from 1 to 4 are used for ARP and RARP frames. The codes from 5 to 8 can be used for XARP frames.

5. The Open Brouter

The previous sections discussed network configurations that use exclusively either IP routers (cf. section 2) or OBs (cf. section 3) to interconnect private TCP/IP domains across the public ATM network. A more sophisticated solution, based on *Open Brouters*, allows both, routed and bridged private domains, to interoperate, as if they were parts of a single network.

To allow communication between OBs and IP Routers, the "halves" of these devices that are connected to the public side (the so-called external ports, in the case of OBs) have to be able to interoperate with each other. Therefore, they must run the same encapsulation mechanism and the same protocols. For that purpose, we consider, that the NDU format is well-suited for the exchange of frames. The Open Bridging Protocol (OBP), can also be used for the exchange of address information between different OBs. If an IP router supports the NDU format and runs OBP and XARP, it can directly communicate with an OB. We call this type of OB/router *Open Brouter*, or OBR. The OBR acts as both an IP Router and an OB and, thus, does not require the implementation of the XARP extension in IP end stations. The communication between an OBR and an OB can be done as follows (cf. Figure 4).

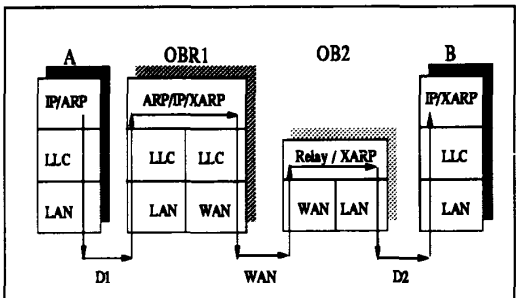

Figure 4. Communication between OB and OBR

Assume that node A in a domain D1 (using IP routing) needs to communicate with node B in a domain D2 (using open bridging). D1 must use an OBR, as mentioned above, and D2 uses an OB and the XARP protocol. The first phase in communication is the IP address resolution. In domain D1, ARP is used as described in section 2 and the IP datagram reaches OBR1. In order to locate B, OBR1 first finds out (from the IP to E.164 mapping tables) whether B is located in a bridging or routing domain. If the domain of B uses routing, the IP datagram is encapsulated to the router of D2, given its E.164 address. If D2 uses bridging, OBR1 sends an OB-resolve request to OB2. The procedure that follows is similar to the one described in section 4 of this paper with the exception that the communication is here between OBR1 (as an end node) and B, not A and B. This affects the formation of the NDU exchanged between OBR1 and OB2 and the processing of frames in OBR1. The architecture of an OBR is shown in Figure 5. The main idea here is to assume that there is a "virtual" third port in the OBR that has the same MAC address as the internal port. Any NDU addressed to that "virtual" port is decapsulated and considered as an LLC frame. On the sending side, an IP datagram sent by OBR needs to be carried in NDU format and is, therefore, passed from the LLC entity to the relay entity of the OBR.

In the aforementioned configuration, if B needs to communicate with A, a similar procedure is followed. IP datagrams generated by B are bridged across OB2 to OBR1, which routes them to A. On the other hand, if domain D1 can also use regular bridging, OBR1 will respond to the OB-resolve request frames sent by OB2 by giving the MAC address of A (assuming that it is in the internal mapping database). In this case,

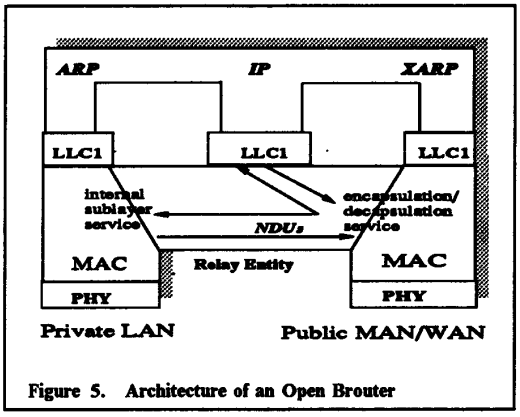

Figure 5. Architecture of an Open Brouter

B can send bridged frames all the way to A, using the bridging capabilities of OBR1 and the local bridges of D1. In this case, A is not required to use XARP and, consequently, it will not be able to use full bridging to send to B.

6. Conclusion

In this paper we have discussed the issue of LAN internetworking across ATM networks. We have presented an approach for bridging independently managed domains across such networks, using the Open Bridges, which take advantage of the special features of ATM networks. These features enable the use of MAC-level bridging for the interconnection across public networks. This is considered as an important advantage especially in the environment of emerging networks with very high bandwidth. The capability of Open Bridges to operate within TCP/IP networks is also an important advantage. We have also presented a new mechanism that allows the extension of the IP address resolution procedure across ATM networks and thus enabling the use of efficient bridging in these environments. Furthermore, we have proposed an Open Brouter architecture that provides for full interoperability between IP routed local networks and bridged networks.

References

[1] Bellcore, Technical Advisory TA-TSY-000772, Issue 3, Generic System Requirements in Support of Switched Multimegabit Data Service, October 1989.

[2] CCITT, Recommendations Drafted by Working Party XVIII/8 (General B-ISDN Aspects), June 1990.

[3] D. Comer, *Internetworking with TCP/IP*, New Jersey: Prentice Hall, 1991.

[4] W. Stallings, *Handbook of Computer Communications Standards, Vol. 3, The TCP/IP Protocol Suite, Second Edition*, Carmel: Howard W. Sams and Company, 1990.

[5] T. Socolofsky and C. Kale, A TCP/IP Tutorial, RFC 1180, January 1991.

[6] J. Postel and J. Reynolds, A Standard for the Transmission of IP Datagrams over IEEE 802 Networks, RFC 1042, February 1988.

[7] CCITT, Recommendation E.164, Numbering Plan for the ISDN Era, 1988.

[8] D. Piscitello and J. Lawrence, The Transmission of IP Datagrams over the SMDS Service, RFC 1209, March 1991.

[9] IEEE 802.1 Part D, MAC Bridges, Draft 9, July 1989.

[10] ISO/IEC JTC 1/SC6 Telecommunications and Information Exchange Between Systems, Enhancement of ISO 8802-5 Token Ring for Multi-Ring Networks, March 1989.

[11] IEEE 802.1 Part G, Remote MAC Bridging, Draft 1, January 1991.

[12] IEEE 802.6 part I, Remote LAN Bridging of Metropolitan Area Networks (MANs), Draft 1, March 8, 1991.

[13] ISO, IS 8802-2, Information Processing Systems - Local Area Networks - Part 2: Logical Link Control, 1989.

[14] A. Tantawy and M. Zitterbart, Open Bridging Across Public Networks, IBM Research Report RC-17022, July 1991.

[15] A. Tantawy and M. Zitterbart, Open Bridging in TCP/IP Networks, IBM Research Report RC-17141, August 1991.