Interworking of Addressing Schemes in an Internetwork

M. Veeraraghavan  
Polytechnic University  
mv@poly.edu

R. J. Munoz and D. M. Rouse  
Lucent Technologies  
{rjmunoz, drouse}@lucent.com

Abstract - In this paper, we examine the problem of address interworking that arises when two networks of the same type (that use the same addressing format) are interconnected by a different type of network (one that uses a different addressing format). Solutions to this problem fall broadly into four categories: (i) schemes that send address resolution messages when a call/packet arrives ("pull" information), (ii) schemes in which address resolution information is sent in routing or routing-like protocols ("push" information), (iii) schemes that use administered cross-office address translation tables at gateways, and (iv) schemes that use encapsulated/mapped addresses. Internetworking problems that are of interest today include PSTN/ATM, IP/ATM and PSTN/IP internetworking. The PSTN uses 8-byte E.164 addresses (telephone numbers), IP networks use 4-byte IPv4 addresses, and ATM networks use 20-byte ATM End System Addresses (AESAs). Current solutions to the address interworking problem for these internetworks fall into the first three categories of solutions. We propose use of the last solution category for these three internetworking problems. We provide a comparison of our proposed scheme with other solutions.

I. INTRODUCTION

Given the proliferation of different networking technologies, there is an increasing need for internetworking. Schemes for internetworking are needed to connect endpoints on two different networks (scenario 1 of Fig. 1), or to use a different network for part of the path between two endpoints that use the same networking technology (scenarios 2 or 3 of Fig. 1). For example, a telephone user may want to be connected to an Internet telephone user (scenario 1), or a call between two telephones may be routed via a packet-switched network, such as an IP or ATM network (scenarios 2 or 3). To answer the question of how to internetwork any two networks, we first need to classify network types, and identify the main attributes of networks that need to be interworked.

Networks can be classified as shown in Fig. 2. A network is characterized by its “switching mode” and “networking mode.” Switching modes are of two types: circuit-switched and packet-switched. In circuit switches, data is switched based on the “position” of arriving bits, where “position” includes space (port number), time, and/or wavelength (frequency) dimensions. In packet switches, data is switched based on some identifier in the packet headers. Networking modes are of two types: connection-oriented and connectionless. In connection-oriented networks, resources are reserved in a connection setup phase prior to data transfer and released after the transfer, whereas in connectionless networks, packets are routed based on address information in packet headers.

Examples of all three types of networks are provided in Fig. 2. ATM and X.25 networks are packet-switched connection-oriented networks. IP networks and SS7 networks are packet-switched connectionless networks. The telephony network and SONET/SDH (Synchronous Optical Network/Synchronous Digital Hierarchy) networks are time-division multiplexed circuit-switched networks. WDM (Wavelength Division Multiplexed) networks are classified as circuit-switched because they switch bits based on the incoming port and wavelength. MPLS (MultiProtocol Label Switched) and IP networks using RSVP (Resource reSerVation Protocol) are classified as belonging to both packet-switched connection-oriented and connectionless classes since they provide both types of services at the network layer.

Attributes of networks are as follows:

1. Connectionless networks typically have a datagram protocol (the user-plane protocol), a routing protocol, and an addressing scheme (addresses are carried in

1. The first author was supported by Lucent Technologies for this work.
2. PSTN: Public Switched Telephone Network; IP: Internet Protocol; ATM: Asynchronous Transfer Mode
2. Connection-oriented networks, whether packet- or circuit-switched, typically have a user-plane protocol (on which user data is carried), a routing protocol, a signaling protocol (to setup/release connections), and an addressing scheme (addresses are carried in call setup signaling messages).

To internetwork any two networks (see scenarios of Fig. 1), the user-plane protocols, routing protocols, signaling protocols, and addressing schemes of the two networks need to be interworked. This “principle” of internetworking is described in [1], and generalized solutions for the interworking of corresponding protocols are described in [2].

In this paper, we focus on the addressing scheme interworking problem, and propose solutions to this problem. We illustrate the address interworking problem using the example configuration shown in Fig. 3. In this configuration, special nodes called gateways (GWs) interconnect networks and perform protocol interworking functions. The address interworking problem can be stated as follows: when a service request (packet or call) identifies a destination by its network address, say a type 1 network address, how does a gateway to a second network, say a type 2 network, through which this service is handled, determine the type 2 network address of the egress gateway through which to reach the destination? In the example shown in Fig. 3, based on the destination end-point’s type 1 network address, gateway 1 (the ingress gateway) needs to identify which of the remaining three gateways (2, 3 or 4) to use in order to reach the destination. In other words, it needs the type 2 network address of the egress gateway that serves the destination endpoint. We refer to this as the address interworking problem.

Solutions to this problem fall broadly into four categories: (i) schemes that send address resolution queries when a call/packet arrives (“pull” information), (ii) schemes in which address resolution information is sent in routing or routing-like protocols (“push” information), (iii) schemes that use administered cross-office address translation tables, and (iv) schemes that use encapsulated/mapped addresses.

Internetworks that are of current interest include PSTN-ATM, IP-ATM, and PSTN-IP internetworks. Example scenarios include (i) A telephone-to-telephone call routed through an ATM network, (ii) An IP-endpoint-to-IP-endpoint flow routed through an intermediate ATM network, and (iii) A telephone-to-telephone call routed through an IP network. The PSTN uses 8-byte E.164 addresses (telephone numbers), IP networks use 4-byte IPv4 addresses, and ATM networks use 20-byte ATM End System Addresses (AESAs).

We review prior work on address interworking in Section II. Our proposed solution for address interworking in PSTN/ATM, IP/ATM and PSTN/IP internetworks is presented in Section III. Section IV presents a comparison of our proposed scheme with other address interworking schemes. The paper is summarized in Section V.

II. PRIOR WORK

In this section, we review address interworking solutions proposed for PSTN-ATM, IP-ATM, and PSTN-IP interworking. The solution proposed for PSTN-ATM internetworks in [3] is a combination of the administered cross-office address translation tables scheme and the “pull” scheme. In the solution described in [3], when a PSTN call arrives at an ingress gateway (say gateway 1 in the example shown in Fig. 3), two forms of interaction are needed between the ingress and egress gateways: call control and connection control.

1. To send the call control message (for terminal capability checking and value-added feature processing), the ingress gateway uses the connectionless services of the SS7 (Signaling System No. 7) network. This network uses “point code” addressing, which is a different addressing format from the PSTN E.164 format. To send the call control message from the ingress gateway to the egress gateway, the former needs the point code address of the latter. An address interworking solution is needed to translate between the called E.164 number and the point code address of the egress gateway. The expected solution is to use administered cross-office address translation tables. This data is “cross-office” because information about one gateway’s E.164 address reachability set needs to be administered into all other gateways.

2. For the connection control part, the ingress gateway needs the ATM End System Address (AESAs) of the egress gateway serving the destination endpoint. This information is “pulled” from the egress gateway as part of the call control message exchange.

Thus, the PSTN/ATM internetworking solution of [3] uses a hybrid address interworking scheme.
Next, we review prior work on IP-ATM address interworking schemes. The Classical IP over ATM [4] solution and NHRP (Next Hop Resolution Protocol) [5] [6] solution use the “pull” approach, in that address resolution information is stored in servers, and the information is extracted by gateways when needed. Another approach for address interworking in the IP/ATM internetworking context is based on the “push” approach. Here, address resolution information mapping destination IP addresses to AESAs are pushed to gateways using the IP routing protocol OSPF (Open Shortest Path). OSPF Address Resolution Advertisements (ARA) [7] are generated using the Opaque LSA (Link State Advertisement) mechanism [8] to distribute IP to ATM (or other) address resolution information to gateways. Opaque LSAs provide a generalized mechanism to allow applications or routers to distribute information within the OSPF domain, which routers can propagate without interpretation. Alternatively, the similar “transitive tags” feature of the PNNI routing protocol [9] can be used to push address resolution information via the ATM network.

Finally, a solution for the address interworking problem for PSTN/IP internetworks has been proposed in [10], [11]. Location servers connected to gateways within domains contain address translations from E.164 addresses of endpoints to IP addresses of their serving gateways. LDAP (Lightweight Directory Access Protocol) [12] is a candidate protocol for intra-domain use between gateways and location servers. TRIP [11], which describes a BGP-like inter-domain protocol, and GLP (Gateway Location Protocol) [13] have been proposed for advertising the reachability of telephony destinations between location servers (essentially address resolution information). These schemes can be classified as “push” schemes. A difference is that instead of using an existing routing protocol, such as OSPF, to “push” address resolution information as was done in the IP-ATM internetworking, a new routing-like protocol is defined that only runs at location servers (and not at routers). However issues such as how location servers know the addresses of their “neighboring” location servers need to be addressed. The TRIP framework document [10] states that this information is administered in the location servers based on service provider agreements. This issue does not arise in the OSPF ARA scheme because the network routing protocol, which runs at each router, is used to spread address resolution information.

Thus, the proposed schemes for address interworking in PSTN/ATM, IP/ATM and PSTN/IP internetworks fall into the first three categories of solutions listed in Section I: “push” schemes, “pull” schemes, or scheme that use administered cross-office address translation tables. In the next section, we present our proposed solution, which belongs to the fourth category of address interworking listed in Section 1, the use of encapsulated/mapped addresses.

III. PROPOSED SOLUTION FOR ADDRESS INTERWORKING

Consider the internetworking configuration shown in Fig. 3, where calls between two endpoints on type 1 networks are routed via a type 2 network. The problem is for the ingress gateway (gateway 1 in the example shown in Fig. 3) to determine the type 2 network address of an appropriate egress gateway through which to reach the called endpoint. Our solution to this problem consists of the following:

1. Gateways register type 1 network addresses that they serve as encapsulated or algorithmically-mapped type 2 addresses with switches in the type 2 network.
2. Network 2 switches use their routing protocols to disseminate reachability data about the registered type-1-address-encapsulated type 2 addresses.
3. When a call/packet arrives at an ingress gateway, it uses the automatic encapsulation/mapping mechanism to determine the type 2 network address corresponding to the destination’s type 1 address, and routes the call/packet with this address as the destination address. The switches/routers in the type 2 network know how to route the call/packet since their routing databases have reachability information for such addresses. When the call/packet reaches the egress gateway, it reconstructs the type 1 address of the destination and continues routing the call/packet to the destination.

Example uses of this scheme for PSTN/ATM internetworking, IP/ATM internetworking and PSTN/IP internetworking are described in the following subsections.

A. PSTN/ATM address interworking

For PSTN/ATM address interworking, our generic scheme can be applied as follows: The format of E.164-encapsulated AESAs defined in [14] is shown in Fig. 4. In the first step, gateways register encapsulated addresses of E.164 endpoints on their PSTN side using ILMI (Integrated Local Management Interface) [15] or PNNI (Private Network-to-Network Interface) routing protocol [9]. This is illustrated in the example shown in Fig. 5, where gateway GW2 registers the E.164 address sets A1 and A2 in the form of E.164-encapsulated AESAs. ILMI is used if the GW2 to ATM switch S4 interface is a user-network interface (UNI), and the PNNI routing protocol is used if this interface is a network-node interface (NNI). Gateway GW2 has its own AESA derived from the AESA of ATM switch S4. By registering the E.164-encapsu-
lated AESAs for its TDM side interfaces into the ATM network, the gateway is essentially programming its signaling processor to respond to multiple AESAs (its own plus the E.164-encapsulated AESAs it registers). This is similar to the case when an ethernet card is programmed to not only receive MAC frames for its own address, but also for multicast-IP-address-encapsulated ethernet addresses for multicast groups in which its host is a member [16].

In the second step, ATM switches use PNNI routing [9] or any other routing protocol to spread reachability information for E.164-encapsulated AESAs within the ATM network. In the example shown in Fig. 5, switch S4 generates PTSPs (PNNI Topology State Packets) announcing reachability to its E.164-encapsulated AESAs. These are flooded to the other ATM switches. Through this flooding method, ATM switch S1 now has reachability information for endpoints (e.g., telephones) addressed by E.164 addresses in sets A1 and A2. If the ATM switch S1 to GW1 interface is PNNI, then S1 will further send the reachability information to GW1.

The third step occurs when a PSTN call arrives at an ingress gateway. It determines the AESA equivalent of the called-party E.164 address and places this in the called-party address field of the ATM connection setup message. The call request is routed through the ATM network to the egress gateway that registered the destination E.164 address. If more than one gateway registered an E.164 address, any of these will be reached. The egress gateway determines the called-party E.164 address from the called-party AESA and continues call setup to the destination using PSTN signaling. In the example of Fig. 5 when GW1 receives a call setup message, assuming that the called party number is set to an address within sets A1 or A2, GW1 creates the corresponding AESA using the encapsulation mechanism shown in Fig. 4, and sends an ATM connection setup message to its switch S1. The connection is routed to the appropriate egress GW, in this case, GW2, based on routing information for the encapsulated AESA.

B. IP-ATM address interworking

Here we describe how our generic scheme, outlined at the start of Section III, is applied to IP/ATM interworking.

Since IPv4 addresses are only 4 bytes and AESAs are 20 bytes, an IP-address-encapsulated AESA format can be defined as shown in Fig. 6. This format is defined by ITU-T in [17] as an amendment to [18].

As the first step, gateways register IP-address-encapsulated AESAs using ILMI or PNNI PTSPs in the same manner as in the PSTN-ATM internetworking scenario described in Section A. The second step is for ATM switches to propagate reachability information for these encapsulated addresses using the PNNI routing protocol. In the third step, when an IP packet arrives at a gateway, it creates an IP-address-encapsulated AESA of the destination IP address, and uses it as the called-party address field in the ATM connection setup message. The connection will be routed to an appropriate egress gateway given that all intermediate ATM switches have reachability information for this AESA.

Consider an example shown in Fig. 7. IP-ATM gateway 2 reports reachability to IP subnets 128.15.11, 131.9.6 and 126.18.17 into the ATM network. This information is spread to all the ATM switches and gateways. When IP-ATM gateway 1 receives an IP datagram destined for 126.18.17.10, it encapsulates this address into an AESA and initiates ATM connection setup. The connection will be routed to IP-ATM gateway 2 since the ATM switches know that AESAs corresponding to the IP address 126.18.17 are reachable through gateway 2.

A point to note is that a gateway needs to register all IP addresses for which it is programmed to serve as “gateway,” and not just the IP addresses in its routing data table. Consider the use of default routes in gateways. Routers typically have
routing information for a few subnet addresses and a default route for all other addresses. For example, IP-ATM gateway 2 could have a routing table with explicit entries for 128.15.11 and 131.9.6 addresses, and a default entry saying that all other addresses are reachable through its 131.9.6 interface. In this case, there is no explicit routing table entry for 126.18.17. However, if gateway 2 serves as a gateway for the 126.18.17 subnet, it needs to be programmed with this information so that the gateway can then register the AESA equivalent of this address into the ATM network. Thus, note that the address resolution information table at the gateway is distinct from its own routing table information for IP addresses. This is true for all four address interworking solutions.

C. PSTN-IP address interworking

In this section, we describe how the generic solution outlined at the start of Section III can be applied to the PSTN-IP address interworking problem. First, we propose an automated address mapping scheme to determine IP addresses from E.164 addresses. Gateways register these IP addresses with the IP network whose routing protocols spread reachability. When a PSTN call setup message arrives at a gateway with a given destination E.164 address, this address is mapped into its corresponding IP address, and IP packets are generated by the ingress gateway into the IP network. IP routers route packets based on their destination IP address, which brings the packets to egress gateways that can then route the call toward the destination with the corresponding E.164 address.

While the principles of this interworking solution are the same, there are a few differences with the PSTN-ATM address interworking case: (i) IPv4 addresses (4 bytes) are smaller than E.164 addresses (8 bytes), unlike AESAs, which are larger; (ii) IPv4 address space is a scarce resource, while the AESA space is not; and (iii) IP networks are connectionless, while ATM networks are connection-oriented. The implications of these three differences are explained below.

Given the relative sizes of IPv4 addresses and E.164 addresses, address encapsulation is not possible; hence a different algorithm is needed for mapping addresses. Without the country code, E.164 addresses are 10 digits long. Since hierarchical addressing is used in the PSTN, it becomes important to only map the first six digits of E.164 addresses. Furthermore, while a digit is represented by four bits, there are only 10 possible numbers for a digit (0 to 9), while 16 numbers can be represented with 4 bits. Leveraging these facts, we propose mapping the first three digits of E.164 addresses (not counting country codes) to 10 bits, and doing the same for the next three digits. This means that only 20 bits are needed to represent summarized E.164 addresses. These 20 bits can be some part of the IP address. Reachability to these mapped IP addresses is sent as routing information.

The second issue, scarcity of IPv4 addresses, is relevant because one of the costs of address encapsulation/mapping is that address space in the second network will be lost to encapsulated/mapped addresses and cannot be used for addressing within the second network for its own endpoints. This was not an issue with AESAs, since the AESA format is large. But this does become an issue with IPv4, whose address size is only 4 bytes. Given the scarcity of public IP addresses, we propose using private IP addresses, such as class A 10.x.x.x addresses [19]. The use of private addresses implies that NAT (Network Address Translation) [20] or some such scheme is needed at routers. Scarce address space is much less of an issue with IPv6, which supports 16-byte addresses.

The third issue, i.e., that IP is connectionless as opposed to ATM, which is connection-oriented, has the following effect. If multiple gateways register reachability to the same set of E.164 addresses, different IP packets within a flow could get routed to different gateways. This does not happen in the ATM case even if multiple gateways register the same set of E.164 addresses because during connection setup one of the multiple gateways is selected, and ATM cells carrying voice data will then be routed on this connection to the same gateway. This problem can be viewed as an “anycast” problem if multiple gateways register the same IP address-encapsulated E.164 address. A solution to the connectionless packet routing problem to anycast addresses is to have devices that find each other using anycast addresses learn the native unicast address of their peer on the first exchange of datagrams [21].

IV. Comparison of Address Interworking Schemes

The four schemes compared here are (i) the “pull” scheme where address resolution information is obtained when a call or packet arrives at a gateway, (ii) the “push” scheme where address resolution information is sent in routing protocol messages as transparent information, or using a routing-like protocol between address resolution servers/gateways, (iii) the administered cross-office address translation tables scheme, in which address resolutions are administered at gateways with each gateway being programmed with addresses supported by all other gateways, and (iv) our proposed solution of using encapsulated addresses.

To compare these schemes, we state the “cost” (disadvantage) of each scheme. Advantages fall out of this listing by simply treating one scheme’s “cost” as another scheme’s “benefit.” The cost of the first scheme, the “pull” scheme, is that an address resolution message exchange is required whenever address resolutions caches are empty. This increases call setup delay or the delay incurred prior to packet transmission. The cost of the second scheme, the “push” scheme, is that routing protocol messages need to carry extra information, and routing data tables increase in size. If the routing protocol of the network is not used to “push” address resolution information, and instead a separate set of servers and new protocol is used, then the implementation of this extra set of nodes/protocol becomes its cost. The cost of the
third scheme, the administered cross-office address translation tables scheme, is that every time a new gateway is added, or new set of addresses can be reached through an existing gateway, all other gateways in the network need to have their data tables updated through some administration means. The cost of the last scheme, our proposed use of encapsulatedmapped addresses, is that address space in one network is consumed by encapsulated addresses, which means that fewer addresses are available for the network’s own endpoints. For example, consider the use of encapsulation for IP-ethercat address interworking for multicast IP addresses. The value 0x01:00:5e is reserved for the first three bytes of ethernet addresses for use by multicast-IP-addr-encapsulated ethernet addresses. This means ethernet addresses starting with this sequence of three bytes are not available for allocation as unicast ethernet addresses. We note that our proposed scheme does not create “extra” routing data or routing protocol messages unlike the push scheme. This is because the routing protocol would have spread reachability about these addresses even if encapsulation was not used since these addresses would have been allocated to the network’s own endpoints.

For PSTN-ATM and IP-ATM internetworking, we conclude that our proposed scheme enjoys all the advantages of not requiring extra signaling when calls arrive as in the “pull” schemes, not creating extra routing information or requiring some new address resolution protocol as in the “push” schemes, and not requiring any administrative support as in the administered tables schemes. Its cost, that of reducing the ATM network’s address space available for its own endpoints, is not significant given the large size of the AESA format. Hence for these two types of internetworks, PSTN-ATM and IP-ATM, we recommend our proposed scheme.

For PSTN-IP internetworks, our proposed scheme avoids the routing overheads of the “push” scheme of TRIP/GLP. Its cost however is more significant than in the ATM case. Through the use of NAT, we counter some of the effects of this cost. Our proposed solution for the PSTN/IP problem works well in situations where an IP network is deployed to “primarily” interconnect two telephone networks. This application is currently important as many telephony service providers are replacing their tandem circuit switches with packet-switched networks. On the other hand, in situations where the IP network is the public Internet, our proposed solution for PSTN/IP address interworking is limited by the use of private addresses and NAT.

V. Summary

In this paper, we examined the problem of address interworking. When a call or packet is sent from an endpoint on a type 1 network destined for an endpoint also addressed by a type 1 network address, if the call/packet is routed partly through a type 2 network, then the ingress gateway at the entry point to the type 2 network needs to determine the type 2 network address of the appropriate egress gateway to use to reach the destination endpoint. We proposed using a solution of encapsulatedmapped addresses and analyzed its advantages/disadvantages over other solutions. Our proposed solution to this problem consists of (i) encapsulating or mapping type 1 addresses into type 2 addresses, (ii) propagating reachability information for these encapsulatedmapped addresses using the routing protocol of the type 2 network, (iii) having gateways determine type 2 addresses automatically from type 1 destination addresses in incoming calls/packets to route the calls/packets through the type 2 network. This general scheme was applied to specific internetworking problems such as PSTN/ATM, IP/ATM and PSTN/IP address interworking.

REFERENCES