ADDRESSING PROBLEMS IN MULTI-NETWORK SYSTEMS

Carl A. Sunshine

University of Southern California
Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90291

Abstract

To allow uers in different networks to communicate with each other, development of powerful yet practical naming, addressing, and routing facilities is essential. Basic procedures for multi-network systems under control of a single organization have begun to be used, but a large set of more sophisticated goals remain to be addressed. This paper describes several of these more advanced problems including extendability, multihoming, network partitioning, mobile hosts, shared access, local site connections, gateway routing, and overcoming differences in heterogeneous systems.

Note:

There are three figures associated with this document which may be obtained from the author by sending a message to <SUNSHINE@ISIF>.

Introduction

The interconnection of multiple computer networks makes it possible for ever wider communities of computer users and applications to interact with each other. A basic set of problems that must be solved in accomplishing such interconnections concerns providing naming, addressing, and routing procedures that are general and convenient yet practical. These problems are particularly difficult when networks of different designs and/or operating under different authorities must be interconnected.

Current multi-network systems are fairly small (tens of networks maximum) and largely designed by and under control of a single organization. (We shall call this "homogeneous" internetworking.) Basic interconnection is supported by simple hierarchical addressing and routing procedures employed uniformly throughout the system [1,4,10,13]. Interconnections of different multi-network systems (heterogeneous internetworking) are just beginning to be made, largely by ad hoc means.

Thus, while some of the basic problems have been solved, a large set of secondary problems will soon be upon us. These include problems of scale (current methods are impractical for systems with hundreds or thousands of networks); supporting more sophisticated functions such as multihoming, network

partitioning, mobile hosts, and shared access; and overcoming the different procedures in heterogeneous systems.

This paper describes several of these interesting problems, and discusses potential solutions. The emphasis is on developing a feel for the range of problems and solutions rather than on detailed or formal treatment of any one problem. In many cases it will be clear that further research is needed to clarify the problems or to develop and evaluate better solutions.

Hierarchical Methods

A basic approach to addressing and routing in large systems is to use hierarchical methods. These methods can be applied at various levels (e.g., within networks and among networks). We give a brief summary of the basic principles involved since these form the background for many of the other problems.

As the number of subscribers or "hosts" in a single network increases, it becomes desirable to introduce a number of switches, each serving a subset of the hosts. These switches must maintain routing tables which give the best outgoing link (or set of links) for any destination. The tables are used to forward incoming packets properly toward their destination. In datagram networks, a routing decision based on final destination is made for every packet, while in virtual circuit nets only the initial call request packet

requires the full routing decision (subsequent packets of a call are forwarded over fixed routes kept in other tables).

If every switch maintained routing information for every destination individually, the routing tables would become very large. A standard approach is to introduce hierarchical addressing, where each host is assigned a particular port on a particular switch, and hence addresses take the form <switch, port>. Then routing may also be done hierarchically by sending all packets destined to a given switch over the same route, ignoring the "low order" portion of the address. Hence each switch need only maintain routes to other switches, greatly reducing the number of different destinations, and hence entries, in the routing tables.

Note that hierarchical routing is one major motivation for introducing hierarchical addresses, but these two techniques do not necessarily go together as we shall see below. Another reason for hierarchical addresses is simply to distribute the authority for assigning addresses within a large system [14].

The same techniques may be extended to multi-network systems by adding another level to the addressing hierarchy so that addresses take the form <net, switch, port>. With hierarchical routing, packets are first routed to the destination network, ignoring the rest of their address, and then routed within the final network as above. This form of

hierarchical addressing has been adopted by the public packet switching networks in CCITT Recommendation X.121, and it appears that most public networks intend to use hierarchical routing as well [13,19].

The reduction of routing table size that accompanies hierarchical routing has its price. The resulting routes may not always be optimal. If there are two ways to reach a remote network (as is often the case), one may be better for some hosts within that network and the other for other hosts. But there is by design no way to determine this from a local routing table which carries a single entry for the entire remote network. An even more serious consequence of strict hierarchical routing is discussed in the next section.

To avoid these problems, routing decisions may based on more of the address where desirable [5,14]. For example, an internetwork routing table could be augmented with entries for individual switches receiving high traffic in a remote network, while other switches in that network were covered by a single network level entry. This leads to a selective increase in the size of routing tables, and requires the ability to search the tables for variable length portions of addresses and to update tables with varying levels of detail.

Network Partitions

A network is said to be partitioned when enough links and/or switches fail so that two or more subsets of its hosts

are formed which cannot communicate with each other. In an isolated network there is no remedy for this situation until sufficient repairs are made to restore connectivity. But if the partitioned net is part of a multi-network system, there may be paths through other nets which could connect the partitions. Unfortunately, these paths are not used within the strictly hierarchical routing procedures described above. And even if a "local" packet were sent to a neighboring network by a switch, it would likely be routed right back into the same paritition by the other network.

This last point indicates another difficulty. Traffic in a remote network destined for the partitioned net will be routed into one or the other partition without consideration of its within-network switch. (Remember that other networks see a single best route to this network considered as a whole.) For some destinations, this will be the wrong partition and the destination will be unreachable by internal routes, leading to failure to deliver packets routed that way from remote nets [14,16].

One solution to this problem is to configure the system with sufficient robustness that partitions occur very rarely, and to simply tolerate the above delivery problems when they occur. This may be satisfactory for commercial systems where loads and outages are fairly predictable.

In military systems where numerous disruptions are

anticipated, some means of forcing use of any available connectivity is desirable [3]. One approach is to treat the number of networks as dynamic, and turn a partitioned network into two networks, each of which can be an explicit destination. This requires rather complex methods of updating each network's view of the overall topology, and promulgates knowledge of a partition in one network to all other networks [8]. Another approach might be to return a special error message to the neighboring router forcing it to choose another entry point to the failed network. This backup-and-try-alternate method has been implemented for call setup in Telenet [19].

"Fast Track" Routing

It is not only in case of catastrophic events like partitioning that use of external routes between two points within the same region may be desirable. If two networks cover the same geographical area, for example a store-and-forward ground net and a broadcast satellite net, performance for some types of traffic may be improved by exiting the ground net near the source, going through the satellite net, and returning to the ground net near the destination. File transfer traffic might obtain higher throughput in this fashion, for example.

To accomplish this, it is once again necessary to violate hierarchical routing. Either the network level routing must

distinguish between destinations best reached directly within the network and those best reached by going outside, or the within-network level must be made to view paths through other networks as a special kind of internal link that is available [9]. But in the latter case, the network level path status information must be brought into the internal link status maintenance procedures, probably a messy business.

Multihoming

A subscriber may want to have multiple connections to a communication system for reliability or performance reasons. In the simplest case, several independent physical lines may be managed as one logical data link to obtain greater reliability, higher throughput, or lower cost (due to the idiosyncracies of carrier tariffs). Several such multiline procedures have been developed, for example in Transpac, and in X.75. The subscriber still has a single address, and no further complications are involved.

In order to protect against node failures as well as line failures, lines to different switches must be used. In this case the user has two (or more) different addresses. The multiple addresses may be at any level in the address hierarchy: (e.g. two addresses within a network, or connected to two different nets). Multiple lines may also provide better performance by connecting directly to highly used areas

of the system and thus avoiding extra hops through the network.

In order to obtain these benefits, the ability to use both addresses and to select the optimal address must exist. This may be accomplished by the source explicitly selecting one address. But this requires the source to know that there are multiple addresses for a given destination, to select the best address for performance, and to switch to an alternate after a failure. These admittedly weighty burdens could be aided by a remote directory/routing service.

Alternatively, the packet could carry the multiple addresses explicitly, allowing each switch to pick the best of the best routes for each address. This of course adds to packet length and routing processing load.

Instead of carrying the multiple addresses, the packet might carry the name (or "logical address") of the destination [14], leaving it for the switches to lookup and select the best address at each point. This would reduce packet complexity, but increase the switch processing demands even further.

Thus we have a spectrum from high source effort to high network effort in making use of multiple addresses. In datagram nets it is probably impractical to require complex processing of the address on every packet, so more source effort will probably be required. In virtual circuit nets a

greater amount of effort can be expended by the net on the call setup request. Some public nets are already providing call forwarding facilities where a call to one inoperative or busy address will automatically be forwarded to an alternate address.

There are problems at the destination as well as the source. To obtain the benefits of multihoming, the destination must be willing to accept traffic on all addresses. In virtual circuit nets, all the traffic for a given call must flow over the same line, so a failure during the call cannot be recovered by using an alternate address. The call must be cleared with possible loss of data, and a new one requested.

Even in some datagram nets, higher level protocols are sensitive to the addresses of the local and remote hosts [3]. The source address is used to demultiplex incoming packets to the proper "connection," and packets coming from an alternate address from that used to establish the connection would not be recognized properly. To avoid this problem, the (single) name of the source could be used in the connection tables, but this would have to be carried in the packet. Alternatively, the multiple remote addresses could be stored as part of the connection table so that a packet specifying any one as source would match properly. These multiple addresses would have to be supplied as part of the connection establishment, and might

be profitably used in sending traffic if the original address failed.

Mobile Hosts

Mobile hosts represent a special case of the multiple address problem. Of course all hosts are technically mobile in the sense that they occasionally change their address due to reconfiguration and movement within the user organization, or modifications to the network topology. Hence directory information to associate the name of a host with its current address is available in most systems, either locally or via some remote server.

However, the problem of changing addresses becomes qualitatively different when the host is expected to change its network attachment point frequently, even in the midst of previously established connections. Special dynamic routing and addressing procedures have been developed for ground based mobile hosts communicating via packet radio within a single network [6]. As distances are increased and this technology is transferred to airplanes, crossing network boundaries may also be anticipated.

One method for "tracking" mobile hosts would be to maintain a specialized database of their current locations (perhaps replicated for reliability), as is done within individual packet radio nets (by the "station"). The mobile hosts would send updates to this database as needed, and users

wishing to establish communication could query the database much as any other directory service. However, they should be prepared to receive frequent address change notifications in the course of a connection, either from the mobile host itself, alternate relay points, or the database. Further details of such a scheme may be found in [18].

Assuming traffic reaches them, destinations must still be "desensitized" from the particular source address as discussed above, since this will change. But there is no fixed set of alternates to exchange at connection setup time in this case, so packets probably must carry a unique identifier (name) of the source as well as its current address. For reliability purposes, they should probably also carry the name of the destination in case it is no longer associated with the address they reach.

Mobile hosts may have multiple addresses at one moment as well as at different times (e.g., an aircraft may be in contact with two radio nets). Thus it becomes apparent that problems can interact with each other, making solutions more difficult.

Sharing Network Access

The opposite problem to one host having several access lines to the net is several hosts sharing a single access line. This may be desirable where the number of physiscal interfaces or ports to the network is limited, or to share a

long access line among nearby subscribers. Public networks provide multidrop interfaces for terminal traffic (X.28), but not for packet mode traffic (X.25). For packet level devices, the alternative to providing a fixed and hence inefficient frequency or time division multiplexor must be some sort of "intelligent" multiplexor functioning at the packet level of network access protocols.

Broadcast networks (e.g., Ethernets and ring nets) inherently provide this capability since every interface hears all traffic. Each interface is responsible for accepting appropriate traffic, and can sometimes be set to intercept traffic for multiple addresses.

Another approach is to use a higher level of protocol to provide the necessary demultiplexing. The Arpanet access (Host-IMP) protocol does not allow for shared interfaces, and the limitation of 4 host interfaces on the original IMPs has proved troublesome in some cases. The Internet Protocol (IP) is the next level above particular network access protocols in the ARPA hierarchy [10,11]. IP addresses are sufficiently long to support multiple "logical" hosts at the same physical host port on the Arpanet. The Host-IMP header indicates the same physical host address for all such packets, and the higher level IP module at the destination demultiplexes the packets to the correct logical host. An independent device to perform this function has been developed based on PDP-11/03

hardware. This "port expander" effectively turns each IMP port into 4-8 ports for hosts that use the Internet Protocol [7].

Networks vs. Gateways as Switches

In most models of hierarchical routing, networks are assumed to function as "super-switches," just as switching nodes do within one network. This view would be literally true if there were a single internet switching node in each net to which all incoming traffic from other nets was routed, and which then forwarded the traffic to another network or to a local host. Figure X shows a small example of a multi-network system and a routing table at one network/switch. The routing table gives the cost in internet hops and the best neighbor net to use to reach each other network in the system.

For efficiency, this internet switching function is usually distributed to processors called "gateways" serving each of the internet links. Instead of being sent through the net to some central point, the internet traffic can be routed immediately at its entry point to the best exit point (either another gateway or the destination host). Figure Y shows the same internet system with internet links labeled, and a routing table at the gateway located on one incoming link. Since the gateway must send packets across its net to a

particular outgoing link, the routing table now shows the name of this next link rather than the next net.

Another step in this progression leads to a single gateway located in the "middle" of each internet link rather than two separate processors in each net. The gateways take on the identity of their internet link(s). In this configuration, it is more realistic to count the network hops as the cost fucntion rather than the internet links. Hence each gateway is maintaining a distance (in network hops) between gateways, and a best next gateway to use for each destination. In this model, the gateways may be more realistically viewed as the switching nodes, and the networks as the links connecting them. This is essentially the dual of the earlier model as shown in Figure Z. But the destinations in the routing table are networks, not gateways, making this a curious sort of hybrid scheme. Hence it is not clear how to apply the "link state" type of routing procedures used in single networks (e.g., the Arpanet) to this multi-network configuration with gateways as switching nodes.

Local Site Connections

Many sites start with a single host connected to a long-haul net. As the site develops, a few more hosts are connected, also directly to the long-haul switch. As even more hosts want to join the net at that site, problems result from costly or inefficient use of network access procedures.

Some sort of port expander or intelligent multiplexor devices as discussed above become attractive.

This addresses the network connection problem, but not the local traffic requirements which are also growing, and may easily exceed traffic to remote sites. The network switch is handling a lot of traffic that never goes any further through the net. In some cases the port expanders may be capable of local switching, forming a rudimentary local net.

To handle local traffic more efficiently, an explicit local net may be desirable. A question then arises as to whether this net should be "known" to the rest of the internet system, and connected to it via one or more full-fledged gateways, or whether it should be "invisible" at the internet level with its hosts appearing as if they were directly connected to the long-haul net. In the first case, local hosts have internet addresses on an explicit local net, while in the second they have addresses on the long-haul net.

The explicit local net approach has certain advantages stemming from the explicit identification of the group of hosts at a site as a network. If the site is connectected to two or more other nets, then the internet routing mechansims will automatically choose the best path to the local hosts, which have only a single address (on their local net).

However, this participation at the internet level can also be a problem. As the number of sites with local nets

increases, so will the number of nets and hence the size of the routing tables and updates which must be propagated all over the internet system. If the growth continues at a site so that there are several local nets connected by "local" gateways, should all of these nets and the local topology be known throughout the internet system? At some point treating local nets on a par with long-haul or backbone nets breaks down.

The invisible local net approach, on the other hand, avoids problems of proliferating networks at the internet level. Many port expander or local distribution systems can perform an internal switching function, relieving the long-haul net switch of handling local traffic. But sites with connections to two or more nets will have multiple addresses for their hosts (one for each net the hosts appear "directly" connected to), and this causes some difficulties as discussed above under Multihoming.

The best solution to this tradeoff is not clear. Adding an additional level to the addressing hierarchy may be a temporary solution, but it, too, will become strained in time. This suggests allowing a variable number of levels in the addressing hierarchy, adding new levels as complexity increases in some area. But this imposes a rigid ordering of levels and hence routing, while in reality "higher" and "lower" may depend on the viewpoint of the user. Further

research is needed on how internet systems may grow and still maintain efficient addressing and routing procedures.

Multiple Domains

Most of the previous discussion has assumed a single compatible "domain" in which network addressing and routing procedures are carried out uniformly. In the real world we have already seen the growth of several large domains with different conventions, including public, mainframe manufacturer, Defense Department, and local networks. It is unrealistic and perhaps impossible that these diverse groups will ever adopt a single addressing scheme, so we must live with the problem of multiple domains for the foreseeable future.

One approach is to assume that one domain will make use of another merely as transport medium between its own homogeneous components. The used system appears merely as one of several types of media that the using system can employ via appropriate access protocols. The using system's packets will be "encapsulated" in the used system's protocols. Of course the two domains can make use of each other, achieving coexistence, if not complete interoperation, by "mutual encapsulation" [15].

To achieve full interoperability between heterogeneous systems, each system must recognize the hosts on the other.

Two basic choices are possible for crossing domain boundaries: mapping and source routing.

In the mapping approach, each domain provides a set of otherwise unused internal addresses which it maps to particular addresses in other domains. Traffic addressed to one of these "pseudo-addresses" is routed to an interface or gateway to the appropriate other domain, at which point the pseudo-address is converted into an address in the other domain. In the simplest case, this requires only bilateral agreements between domains, but it may also be extended across intermediaries with further collaboration.

A disadvantage of this approach is that the number of external addresses available is limited to those for which mappings have been previously defined and installed. Typically only a small fraction of remote parties are supported. Another disadvantage is that the same party has different addresses in different domains--the directory of names to addresses has many entries for each name, one for each domain supporting that party. The major advantage is that for those names supported, the users may address remote parties in exactly the same fashion as local ones, with no additional procedures.

In source routing [14,17,5], the source specifies a route to reach the destination consisting of the addresses of successive inter-domain gateways, and ending with the final

destination. Each address in this list is interpreted within a domain where it is meaningful, and then removed so that the next address is available in the next domain transitted.

Using this method, the full range of remote parties is accessable, and the inter-domain gateways do not have to maintain any predefined mappings or perform address conversions. The burden is shifted to the source which must know enough about the overall topology and address formats to construct a successful source route. Of course packet headers become bigger, and packet processing increases to accomodate the variable length source routes. Once again, the "address" of a given party varies from one domain to another, but it is now possible to combine this information--if the directory gives the source route to X from domain A, and a user in domain B knows a route to domain A, he can concatenate them to get a route to X from B (although it may not be an optimal route).

It is often useful to collect a return route at the same time the source route is being consumed. This allows the destination to reply. In general the return route is not simply the inverse of the source route. The return addresses are added as the packet enters each domain, while the successive destination addresses are removed as the packet exits each domain (see [17] for a detailed example).

The "network independent" transport protocol [2]

developed by the British PSS Users Forum is one of the first to explicitly deal with the problem of multiple domains. They suggest essentially a source routing mechanism. There are additional provisions for translating explicitly identified address information transmitted as data between end users. The protocol assumes a route setup procedure as part of call establishment so that the source route need only be carried in the call request packet.

The public networks have also provided for a limited form of source routing in the Call User Data field of X.25 call request packets. This field may be used by the destination DTE as additional address information for subsequent steps in a call. This mechanism was used to allow international calls between Canadian and US public networks before the hierarchical X.121 numbering plan was put into effect [12]. The Call User Data field is also beginning to be used in an ad hoc fashion to provide addressing within various private and/or local nets connected to public nets.

The Arpa Internet Protocol also supports a source routing option, but addresses within the route are all expected to be IP format addresses [11].

Conclusions

We have identified a number of problems that must be considered in going beyond the simple network interconection techniques that are in use today. The significance of these

problems is just beginning to be widely percieved. Some preliminary solutions have been proposed, but little practical experience exists. Much work remains to be done in clarifying the problems, and in developing and evaluating solutions.

Acknowledgements

Many of the concepts presented in this paper have been discussed over several years as part of the ARPA Internet project. Much of the credit for developing and clarifying these ideas belongs to my colleagues at ISI and the other sites engaged in this project.

References

Note: Several of the references listed below are Internet Experiment Notes, unpublished memos written for the ARPA Internet project.

[1] D. R. Boggs, J. F. Shoch, E. A. Taft, and R. M. Metcalfe, "Pup: An Internetwork Architecture," IEEE Trans. on Communications 28, 4, April 1980, pp. 612-623.

[2] British Post Office PSS User Forum, A Network Independent Transport Service, February 1980.

[3] V. G. Cerf, Internet Addressing and Naming in a Tactical Environment, Internet Experiment Note 110, August 1979.

[4] V. G. Cerf and P. T. Kirstein, "Issues in Packet-Network Interconnection," Proc. IEEE 66, 11, November 1978, pp. 1386-1408.

[5] D. D. Clark and D. Cohen, A Proposal for Addressing and Routing in the Internet, Internet Experiment Note 46, June 1978.

[6] R. E. Kahn, S. A. Gronemeyer, J. Burchfiel, and R. C. Kunzelman, "Advances in Packet Radio Technology," Proc. IEEE 66, 11, November 1978, pp. 1468-1496.

[7] H. A. Nelson, J. E. Mathis, and J. M. Lieb, The ARPANET IMP Port Expander, SRI Report 1080-140-1, November 1980.

[8] R. Perlman, Flying Packet Radios and Network Partitions, Internet Experiment Note 146, June 1980.

[9] R. Perlman, Utilizing Internet Routes as Expressways Through Slow Nets, Internet Experiment Note 147, June 1980.

[10] J. B. Postel, "Internetwork Protocol Approaches," IEEE Trans. on Communications 28, 4, April 1980, pp. 604-611.

[11] J. B. Postel, C. A. Sunshine, and D. Cohen, "The ARPA Internet Protocol," to appear in Computer Networks, 1981.

[12] A. M. Rybczynski, D. F. Weir, and I. M. Cunningham, "Datapac Internetworking for International Services," Proc. 4th Int. Conf. on Computer Communication, September 1978, pp. 47-56.

[13] A. M. Rybczinski, J. D. Palframan, and A. Thomas, "Design of the Datapac X.75 Internetworking Capability," Proc. 5th Int. Conf. on Computer Communication, October 1980, pp. 735-740.

[14] J. F. Shoch, "Inter-Network Naming, Addressing, and Routing," Proc. 17th IEEE Computer Society Int. Conf., September 1978, pp. 72-79.

[15] J. F. Shoch, D. Cohen, and E. A. Taft, "Mutual Encapsulation of Internetwork Protocols," to appear in Computer Networks, 1981.

[16] C. A. Sunshine, "Interconnection of Computer Networks," Computer Networks 1, 3, January 1977, pp. 175-195.

[17] C. A. Sunshine, "Source Routing in Computer Networks," ACM SIGCOMM Computer Communication Rev. 7, 1, January 1977, pp. 29-33.

[18] C. A. Sunshine and J. B. Postel, Addressing Mobile Hosts in the ARPA Internet Environment, Internet Experiment Note 135, March 1980.

[19] D. F. Weir, J. B. Holmblad, and A. C. Rothberg, "An X.75 Based Network Architecture," Proc. 5th Int. Conf. on Computer Communication, October 1980, pp. 741-750.